NON-LINEAR MAPPINGS FOR GRAPH-BASED RANDOM ERROR CORRECTION

by

Mariano Eduardo Burich

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical and Computer Engineering

Summer 2022

© 2022 Mariano Eduardo Burich
All Rights Reserved

NON-LINEAR MAPPINGS FOR GRAPH-BASED RANDOM ERROR CORRECTION

by

Mariano Eduardo Burich

Approved: _____
Jamie D. Phillips, Ph.D.
Chair of the Department of Electrical and Computer Engineering

Approved: _____
Levi T. Thompson, Ph.D.
Dean of the College of Engineering

Approved: _____
Louis F. Rossi, Ph.D.
Vice Provost for Graduate and Professional Education and
Dean of the Graduate College

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Javier Garcia-Frias, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Xiang-Gen Xia, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Xiugang Wu, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Pedro M. Crespo, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Richard D. Souza, Ph.D.
Member of dissertation committee

ACKNOWLEDGEMENTS

First, I would like to thank my academic advisor, Dr. Javier Garcia-Frias, who has been my mentor, professor, and friend for the past five years. In addition to teaching me several topics such as channel coding, information theory, mathematics, and communications, he inspired me to be a better researcher and encouraged me to believe in myself. I especially would like to thank him for his suggestions and support during the development of our work and when times were tough, either in my personal life or when results were far from expected.

I would also like to thank the committee members of this dissertation, composed of Dr. Xiang-Gen Xia, Dr. Xiugang Wu, Dr. Pedro M. Crespo, and Dr. Richard D. Souza, whose research I admire and who have dedicated their time and effort to review and analyze this work.

The support of my wife, Mayara, was crucial for a healthy life during my Ph.D. journey. I am deeply grateful for her patience, optimism, and the great times we spent together. She motivated me to persist whenever I felt overwhelmed, believing that I was capable of facing any challenges.

I am very thankful to my generous parents, Daniel Burich and Marta Zarzoso, who have endured the pain of distance and encouraged me to pursue my goals. They always did their best to support me and to give me a great head start in life. Thanks to my sister, Martha Burich, who was always there for me and helped me see the bright side of all situations.

There were many people who were part of my life during my Ph.D. studies and contributed to my personal growth. I would like to thank my research colleagues, Lu Li and Yumin Li, for the productive discussions. I also had the pleasure of working with Imanol Granada, whom I would like to thank for his friendship and great insights.

In 2017, I had the privilege of starting my Ph.D. at the same time as Karelia Pena and Sergio Sepulveda, whom I would like to thank for the friendship over these years. My roommate, Juan Felipe Florez Ospina, also holds a special place in my life due to his friendship and our various discussions about science and signal processing. I am also very grateful for all the great moments and conversations with my friends Diego Roa and Jose Monsalve Díaz. I am thankful to Daniel Bortoluzzi and Matheus Barreto - my Brazilian friends in the United States who made me feel as if I were home. I am also thankful to all other students from the Hispanic Latino Graduate Student Organization (HLGSA) and the Electrical Engineering department who have contributed to make my Ph.D. journey even more special.

To my lifelong friends from UTFPR: Erik, Francisco, Guilherme, Hector, Igor, Ismael, Lucas V., and Lucas Z. - thank you for the support and discussions. Special thanks to Gabriela De Souza, my dearest friend, who has always made an effort to be part of my life by giving me valuable advice and sharing her thoughts.

This material is based upon work supported by the National Science Foundation under Grants CCF-2007754 and CIF-1618653. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

TABLE OF CONTENTS

LIST OF TABLES . . . . . **xi**
LIST OF FIGURES . . . . . **xii**
ABSTRACT . . . . . **xvii**

Chapter

1 INTRODUCTION **1**

1.1 Why Non-Linear Codes? 1
1.2 Low-Density-Parity-Check Codes 6
1.3 Rate Compatible Modulation 7
1.4 Proposed System in the Context of Coded Modulation 8
1.5 Joint Source-Channel Coding 9
1.6 Channel Model and Theoretical Limits 10
1.7 Dissertation Organization 13

2 NON-LINEAR CODES BASED ON GRAPHS **14**

2.1 Encoder 16

2.1.1 Graph Construction 16
2.1.2 Encoding Function 17
2.1.3 Constellation 18

2.2 Decoder 19

2.2.1 Initialization 19
2.2.2 Symbol Nodes 19
2.2.3 Input Nodes 21
2.2.4 Decision 22

2.3 Mapping Design 22

2.3.1 Design Parameter 23

2.3.2 Construction Algorithms . . . 23
2.3.2.1 Parity Intersect Method . . . 24
2.3.2.2 Distance Profile Method . . . 26
2.4 Results . . . 27
2.4.1 Parity Intersect Method . . . 28
2.4.2 Distance Profile Method . . . 31
2.4.3 Comparison with Existing Linear Codes . . . 34
2.5 Conclusion . . . 37
3 NON-LINEAR CODING FOR NON-UNIFORM BINARY SOURCES . . . 39
3.1 One-Dimensional Scheme . . . 40
3.1.1 Encoder . . . 40
3.1.2 Decoder . . . 43
3.1.2.1 Initialization . . . 44
3.1.2.2 Symbol Nodes . . . 44
3.1.2.3 Input Nodes . . . 46
3.1.2.4 Decision . . . 46
3.1.3 Results . . . 46
3.2 Two-Dimensional Scheme . . . 52
3.2.1 Encoder . . . 53
3.2.2 Amplitude . . . 54
3.2.3 Angle . . . 54
3.2.4 Decoder . . . 57
3.2.4.1 Initialization . . . 57
3.2.4.2 Symbol Nodes . . . 58
3.2.4.3 Input Nodes . . . 60
3.2.4.4 Decision . . . 60
3.2.5 Results . . . 60
3.3 Conclusion . . . 65

4 NON-LINEAR CODING FOR NON-BINARY SOURCES 66

4.1 Encoder 67

4.1.1 Encoding Function 67
4.1.2 Group Intersect Method 68

4.2 Decoder 72

4.2.1 Initialization 72
4.2.2 Symbol Nodes 72
4.2.3 Input Nodes 74
4.2.4 Decision 74

4.2.4.1 Decision for Non-Binary Uniform Sources 74
4.2.4.2 Decision for Quantized Analog Sources 75

4.3 Results 75

4.3.1 Non-Binary Uniform Sources 76
4.3.2 Analog Sources 79

4.4 Conclusion 82

5 NON-LINEAR CODING FOR ANALOG SOURCES 84

5.1 System Overview 86
5.2 Encoder 87

5.2.1 Proposed Non-Linear Function S 90

5.3 Decoder 93

5.3.1 Initialization 94
5.3.2 Symbol Nodes 95
5.3.3 Shifted Nodes 96
5.3.4 Decision 97

5.4 Results 97
5.5 Conclusion 102

6 CONCLUDING REMARKS AND FUTURE WORK 104

6.1 Future Work 105

BIBLIOGRAPHY 106

Appendix

A RELATIONSHIP BETWEEN HAMMING AND EUCLIDEAN DISTANCES 112
B MONOTONIC ENERGY ALLOCATION 115
C PERMISSIONS 117

LIST OF TABLES

2.1 Examples of mappings $M_i : (b_1, b_2, b_3) \rightarrow \Gamma$ by using the Parity Intersection method for $d_r = 3$, Gaussian constellation $\Gamma = G$, and a Gray code as Initial Mapping. P_i represents the parity constraint vector for mapping M_i. 25

3.1 Comparison between gaps to the theoretical limit achieved with the best linear codes in the literature and those of the proposed non-linear codes with one-dimensional constellations. 51

5.1 The proposed non-linear function S partitions the Gaussian source according to the indicated regions. 98

LIST OF FIGURES

1.1 Point-to-point digital communications system [1, 2]. 2

1.2 Each symbol x_v is generated by applying the non-linear function f over d_r randomly chosen elements of the input information. 4

1.3 Point-to-point communications based on coded modulation. 8

2.1 Proposed non-linear code encoding 6 input bits into 4 output symbols. Each symbol is generated by applying the function f to a subset of the input bits according to the underlying graph. 16

2.2 Parity Intersect method using a Gray code as initial mapping and a Gaussian constellation with parity constraints P_i, $R_c = 1$ and $d_r = 5$. 28

2.3 Parity Intersect method using an initial mapping that is random and a Gaussian constellation with parity constraints P_i, $R_c = 1$ and $d_r = 5$. 29

2.4 Parity Intersect method using an initial mapping that is random and a PAM constellation with parity constraints P_i, $R_c = 1$ and $d_r = 5$. 30

2.5 Distance profile method with $\mathbf{d_1}^{\alpha=4,\beta=5}$, $R_c = 1$ and $d_r = 5$ (adapted from [3]). 31

2.6 Distance profile method with $\mathbf{d_1}^{\alpha=4,\beta=5}$, $R_c = 1$, $d_r = 6$, and $H(\mathbf{b^k}, \mathbf{b^{k+2}}) \neq 2 \ \forall k$. 32

2.7 Distance profile method with $\mathbf{d_1}^{\alpha=4,\beta=5}$, $R_c = 2$, $d_r = 6$, and $H(\mathbf{b^k}, \mathbf{b^{k+2}}) \neq 2 \ \forall k$. 33

2.8 Distance profile method with $\mathbf{d_1}^{\alpha=6,\beta=5}$, $R_c = 3$, $d_r = 8$, and $H(\mathbf{b^k}, \mathbf{b^{k+2}}) \neq 2 \ \forall k$. 34

2.9 Distance profile method with $\mathbf{d_1}^{\alpha=6,\beta=5}$, $R_c = 4$, $d_r = 8$, and $H(\mathbf{b^k}, \mathbf{b^{k+2}}) \neq 2 \ \forall k$. 35

2.10 Comparison between our proposed non-linear codes designed using the Distance Profile method and existing linear codes in the literature when the rates are $R_c = 1$ and $R_c = 2$. 36

2.11 Comparison between our proposed non-linear codes designed using the Distance Profile method and existing linear codes in the literature when the rates are $R_c = 3$ and $R_c = 4$. 37

3.1 Proposed function f and probability of the constellation symbols when $p_0 = 0.9$. 41

3.2 For each value p_0, maximum value of d_r that guarantees a monotonic decreasing energy allocation. 43

3.3 Simulation results for the proposed non-linear codes using one-dimensional mappings for joint source-channel coding of non-uniform sources when $R_c = 1$. 48

3.4 Simulation results for the proposed non-linear codes using one-dimensional mappings for joint source-channel coding of non-uniform sources when $R_c = 2$. 49

3.5 Simulation results for the proposed non-linear codes using one-dimensional mappings for joint source-channel coding of non-uniform sources when $R_c = 3.7$ and $R_c = 5$. 50

3.6 Simulation results for the concatenation between the proposed non-linear codes using one-dimensional mappings and LDGM codes. 51

3.7 First iterations of the greedy algorithm that builds the proposed two-dimensional mappings for the case of $d_r = 5$. 56

3.8 Intermediate and final iterations of the greedy algorithm that constructs the proposed two-dimensional mappings when $d_r = 5$. . . 57

3.9 Simulation results for the proposed non-linear codes using two-dimensional mappings for joint source-channel coding of non-uniform sources when $p_0 = 0.8$. 61

3.10 Simulation results for the proposed non-linear codes using two-dimensional mappings for joint source-channel coding of non-uniform sources when $p_0 = 0.9$. 62

3.11 Simulation results for the proposed non-linear codes using two-dimensional mappings for joint source-channel coding of non-uniform sources when $p_0 = 0.9$ and $R'_c = 10$. 63

3.12 Constellation constructed by the proposed algorithm with the additional angle constraint for $d_r = 11$. Circumferences with points associated to input vectors with even Hamming weight are constrained to two quadrants, while the ones with odd Hamming weight are constrained to the other two quadrants. The shape of the constellation naturally arises as the algorithm progresses. 64

3.13 Simulation results for the proposed non-linear codes using two-dimensional mappings for joint source-channel coding of non-uniform sources when $R'_c = 4$ and the angles are constrained according to the Hamming weight of each circumference. 65

4.1 Construction of the auxiliary graphical structure for the Group Intersect algorithm. 69

4.2 Construction of the graphical structure for the mapping M_0 by using the Group Intersect method for a Q-ary source. Each circle represents a non-binary vector. 70

4.3 Group Intersect method when a non-binary uniform source with three levels ($Q = 3$) is considered for $d_r = 2$. 71

4.4 Group Intersect method when a non-binary uniform source with three levels ($Q = 3$) is considered. 77

4.5 Group Intersect method when a non-binary uniform source with three levels ($Q = 3$) is considered for $R_c = 3$ and $d_r = 6$. 78

4.6 Group Intersect method when a non-binary uniform source with four levels ($Q = 4$) is considered. 79

4.7 Group Intersect method when a non-binary uniform source with five levels ($Q = 5$) is considered for $R_c = 2$ and $d_r = 4$. 80

4.8 Transmission of a Gaussian source when optimal scalar quantization with three levels ($Q = 3$) is considered for $R_c = 2$ and $d_r = 5$. . . . 81

4.9 Transmission of a Gaussian source when optimal scalar quantization with three levels ($Q = 3$) is considered for $R_c = 3$ and $d_r = 6$. . . . 82

5.1 Optimal Performance Theoretically Attainable (OPTA) for linear and non-linear k:n analog schemes with coding rate $R_c = k/n$. Linear schemes are considerably worse than non-linear ones, and their performance for bandwidth compression ($R_c > 1$) barely improves as the signal-to-noise ratio increases. 88

5.2 Analog linear system 89

5.3 Proposed analog system that uses the non-linear function S to precode each sample produced by the analog source. 89

5.4 Proposed scheme that encodes 6 Gaussian samples into 4 output symbols. Each symbol is generated by the linear combination of a subset of the source samples modified by the function S. 90

5.5 Proposed non-linear function S for the case of $Q = 2$ before normalization. 91

5.6 Proposed non-linear function S for the case of $Q = 3$ before normalization. 92

5.7 Proposed non-linear function S for the case of $Q = 4$ before normalization. 92

5.8 Proposed non-linear function S for $Q = 2$ and $Q = 3$ after normalization so that the output has variance 1. The increase of l_0 and l_1 leads to a shrinkage of the lobes, which in the limit converge to deltas. 93

5.9 Simulation results for $d_r = 4$ and $R_c = 2$ as a function of the gap l_0. 98

5.10 Simulation results for SNR=30 dB, $Q = 4$, $d_r = 4$, and $R_c = 2$ as a function of the spacings, $\mathbf{l} = [l_0, l_1, l_2]$ ($l_0 = l_2$). The SDR of each data point is depicted inside the squares. 99

5.11 Simulation results for the proposed method with $R_c = 2$. Each data point represents the best SDR obtained after Monte Carlo optimization of the gap vector $\mathbf{l}$. 101

5.12 Simulation results for $d_r = 5$ and $R_c = 3$ as a function of the spacing l_0. 102

5.13 Simulation results for the proposed method with $R_c = 3$. Each data point represents the best SDR obtained after Monte Carlo optimization of the gap vector $\mathbf{l}$. 103

ABSTRACT

Since Claude Shannon's seminal work in 1948, it is possible to quantify the optimal performance of communication systems in terms of information rate ("speed") vs channel quality (signal-to-noise ratio). The mathematical framework developed by Shannon established that it is asymptotically possible to achieve error rates arbitrarily close to zero provided that the information rate is below the channel capacity. Indeed, as the block length goes to infinity, certain families of linear codes are able to approach the channel capacity. However, since practical systems use codes with finite block length, in practice the joint design of channel and source coding (joint source-channel coding) and the use of non-linear codes can outperform linear schemes that separately perform compression and correction.

In this work, we propose the use of non-linear codes that perform correction, compression, and modulation in one step. The proposed codes are random-like and defined by a graph and a non-linear mapping. The randomly built encoding graph allows us to design systems that are flexible in terms of rate and that are also able to transfer the source statistics to the output symbols, making the proposed schemes specially suitable for joint source-channel coding. We define design guidelines that lead to good performance for the use of graph-based non-linear codes in the transmission of binary, non-binary, and analog sources. The proposed non-linear mappings allow us to obtain good performance for the transmission of binary and non-binary sources at high rates, outperforming existing joint source-channel coding schemes for non-uniform sources, and achieving low error rates for the region under the Shannon limit, where the error cannot be arbitrarily close to zero. For the first time in the literature, we show that it is possible to approach the theoretical limits in the transmission of Gaussian

sources over Gaussian channels by using graph-based analog codes of flexible rate that are decoded using message passing.

Chapter 1

INTRODUCTION

1.1 Why Non-Linear Codes?

Since Claude Shannon's seminal work in 1948 [1], it is possible to quantify how close communication systems are from optimal performance in terms of information rate ("speed") and channel noise (signal-to-noise ratio). Intuitively, the connection between rate and signal-to-noise ratio (SNR) is clear: the faster we would like to transmit data, the better the channel should be (higher SNR). However, Shannon developed a mathematical framework which proved that the error rate could be arbitrarily close to zero as long as we sent information at a rate lower than the *channel capacity* (also called Shannon capacity). Moreover, if we transmit information at rates higher than the channel capacity, we would not be able to design systems with error rate arbitrarily close to zero.

This remarkable result was established by the *noisy-channel coding theorem* [1], which states that we can construct a mathematical function (channel code) to encode the messages prior to transmission, making them robust to channel errors and allowing the receiver to recover the original messages reliably when the block length goes to infinity and the information rate is less than the capacity. In addition to the noisy-channel coding theorem, Shannon proved the *source coding theorem* [1], which establishes the limit for data compression given the statistical model of a data source. This limit is achieved by the use of a *source code* (data compression). Finally, Shannon also proved, through the *separation theorem*, that when optimal compression (source coding) is followed by optimal protection (channel coding), the communication system is asymptotically optimal (i.e., when the block length goes to infinity). Due to these

three theorems, source and channel codes have been independently designed for many years and are considered a key part of point-to-point communications systems, as depicted in Figure 1.1.

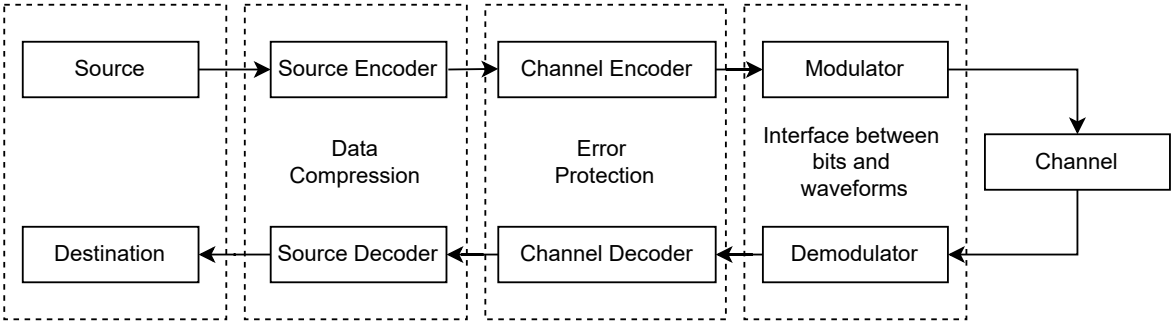


Figure 1.1: Point-to-point digital communications system [1, 2].

In light of these results, the fields of source and channel coding have become of utmost importance for the communications community, and are respectively in charge of designing codes that efficiently compress and protect the transmitted data. While Shannon was not the first to construct error correcting codes [4], the mathematical formulation of communication systems and the creation of the field of information theory caused a paradigm shift that lay the foundation for current communications technology.

Both limits, for compression and protection, can be closely approached in practice by the use of appropriate source [5] and channel codes [6–9]. While Shannon's separation theorem simplifies the system's analysis and design, it does not guarantee information rate flexibility and optimal performance for finite block lengths. With this in mind, schemes where source and channel coding are combined, in what is called *joint source-channel coding* [10–13], aim at improving the system performance and its flexibility by designing codes that exploit the natural redundancy contained in the source to improve robustness to the channel variations. Joint source-channel coding does not necessarily include the modulation block shown in the Figure 1.1, but there are also schemes that jointly perform channel coding and modulation, such as Trellis Coded

Modulation (TCM) [14]. In this work we expand and generalize a type of linear coded modulation called Rate Compatible Modulation (RCM) [15]. Originally, RCM was designed to provide seamless rate adaptation and was expanded in several works, such as [16,17], where faster RCM decoders were designed. Li and Garcia-Frias [18] proposed a parallel concatenation between RCM and Low-Density-Generator-Matrix (LDGM) codes [19, 20] to lower the system's error floor, while in [17, 21] RCM was used for the transmission of non-uniform binary sources (joint source-channel coding). Li [22] and Granada et al. [23] developed RCM-based schemes for multiple access channels, and in [24] RCM was applied to joint source-channel coding of correlated information sources transmitted over a multiple access channel. In [25], RCM was concatenated to a high-rate LDPC code and also modified to work in a rateless manner, creating what are called Analog Fountain Codes (AFC).

Rate Compatible Modulation, as well as LDPC and Turbo codes, are based on the following three concepts: (1) random-like encoding; (2) linear mapping; (3) iterative decoding. While in this work we keep the ideas of iterative decoding and random-like encoding, differently from previous work we design codes based on non-linear mappings. Specifically, the proposed non-linear codes are based on the successive repetition of a non-linear function f, which directly maps a randomly chosen subset of the input information provided by the source into an output symbol (a constellation point), as depicted in Figure 1.2.

The reason for most traditional channel codes to be linear is that their encoder is simply a matrix multiplication, which can be easily implementable in hardware through the XOR operation. Linearity also facilitates the analysis of those channel codes, since all codewords have the same distance profile. Interestingly, optimal decoding of general linear codes is NP-complete [26]. Even so, it is possible to obtain good performance in particular cases by performing low complexity optimal decoding, such as with the Viterbi decoder [27] for convolutional codes, and also by using suboptimal decoding, as the message passing [28] or belief propagation algorithm for LDPC codes. It is important to remember that using linear codes does not limit performance at

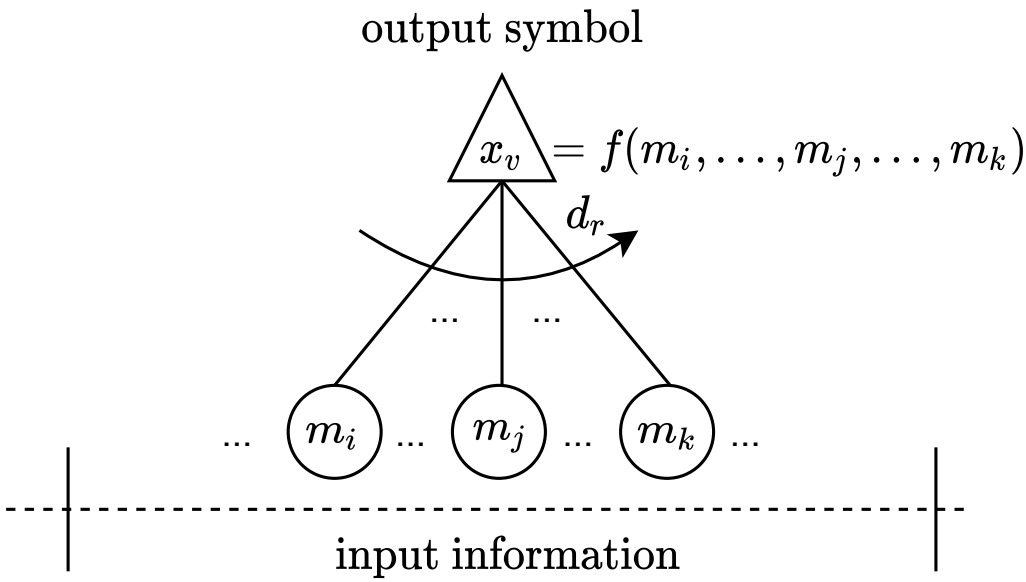


Figure 1.2: Each symbol x_v is generated by applying the non-linear function f over d_r randomly chosen elements of the input information.

the asymptotic block-length regime because there exist capacity-approaching linear codes [29].

While there are good reasons to focus on the study of linear codes, the more general family of non-linear codes is also a topic of interest, since they may outperform linear codes, specially in more practical scenarios and special applications such as short block length, flexible rate, and in the case of joint source-channel coding. It can be argued that RCM does not take considerable advantage of the linear transformations to simplify encoding, decoding, or the code design. Designing linear transformations for RCM is still a complicated task, and it is done by brute-force [15, 17], optimization [30, 31], or heuristic approaches [25]. In [32], a type of two-dimensional constellation non-linear RCM was developed with the goal of reducing the peak to average power ratio (PAPR) of the communications scheme. While it performs well, it is not a general non-linear system since one of the steps during encoding is to perform a linear transformation on the source bits. Our goal is to design non-linear codes

that directly map bits (or other type of information) to channel symbols (constellation points) by using a single block where source coding, channel coding, and modulation are all combined.

The proposed non-linear codes have many advantages when compared to traditional systems that perform linear coding and that use separate source and channel codes. In the literature, the main metric to compare channel codes is to measure how far they are from the Shannon limit for an arbitrarily low error rate ϵ. However, communication at SNRs below the Shannon limit are possible if a finite error rate is acceptable. While conventional linear codes typically exhibit very bad performance at SNRs below the Shannon limit, it is possible to design non-linear transformations that approach the theoretical error rate limits under these conditions. Specifically, most capacity-approaching linear coding schemes such as LDPC codes and Turbo codes experience what is called the waterfall effect: for low SNR, the code performs very badly and as the SNR increases it suddenly experiences a fast improvement (waterfall region) which is followed by a region of relatively slow improvement (error floor region). An advantage of the proposed non-linear codes is the flexibility in controlling the waterfall and error floor regions. Specifically, the proposed codes experience a tradeoff between error floor and waterfall, which can be adjusted in order to improve the performance at specific SNRs and to approach the theoretical limits for SNRs below the Shannon capacity. Since the proposed non-linear transformations are generalizations of linear ones such as RCM and AFC, they are also applicable for high-throughput, rateless schemes, and for short block lengths, which has received an increased interest due to the advent of Ultra-Reliable Low Latency Communications (URLLC) [33–35] .

The general framework proposed in this dissertation is based on random graphs, as in LDPC codes, except that parities are substituted by non-linear transformations. This general framework allows the encoding of sources with non-binary information by performing the encoding and the decoding processes in the non-binary domain. This approach can also be applied to the general case of non-uniform, non-binary sources to perform joint source-channel coding. Indeed, one of the most important applications

where the proposed coding scheme benefits from the non-linear transformation is in joint source-channel coding, since a direct mapping from source to channel increases the synergy between the input bits and the transmitted symbols, allowing us to finely control the power allocated to constellation points with unequal probability.

1.2 Low-Density-Parity-Check Codes

Low-Density-Parity-Check (LDPC) codes are a family of random linear error correcting codes which are defined by the sparse random structure of their parity check matrix. The matrix is randomly built by obeying predefined constraints that fix the amount of ones in its rows and columns. By the simple process of defining these general constraints and randomly building the matrix, very powerful codes can be generated.

LDPC codes were invented in [6], rediscovered in [8] and have since then been used in many applications such as satellite communications, WiFi, flash memories, digital video broadcasting, 5G, optical communications, to name a few. Together with Turbo codes [7], they were the first type of error correcting codes that were able to approach the Shannon limit, and they did so by being constructed in a random-like fashion.

The encoding process of LDPC codes is considered to be of higher computational complexity than that of Turbo codes, while Turbo codes are deemed to have a higher computational complexity at the decoder. Since LDPC codes are block codes, their encoding is performed by matrix multiplication. While maximum likelihood decoding of an LDPC code is NP-complete and intractable, its decoding can be performed approximately by a message passing (belief propagation) algorithm [28]. The use of belief propagation is possible because the sparse parity check matrix can be represented as a bipartite graph of parities and bits, whose connections represent the constraints that define the codebook.

The design of LDPC codes is not trivial and at a higher level consists on finding the distribution of the number of connections the input bits and the parities should have in the graph. In order to approach the Shannon limit, LDPC codes must be

optimized by techniques such as density evolution [36] or extrinsic information transfer (EXIT) charts [37].

While an LDPC code has a sparse parity matrix, its generator matrix is not sparse in general. A sparse generator matrix would allow LDPC codes to have a lower computational complexity at the encoder. Therefore, as a particular case of LDPC codes, Low-Density-Generator-Matrix (LDGM) codes were studied in [20, 38]. In addition to the systematic bits, LDGM codes transmit the XOR of random subsets of the input bits as the redundant information to be used for correction at the decoder. The drawback of such a simple encoder is that the error floor is high when compared to other LDPC codes. In order to reduce their error floor, LDGM codes can be concatenated [20], which leads to error rates close to Turbo and LDPC codes.

1.3 Rate Compatible Modulation

"Rate Compatible Modulation (RCM) was proposed in [15] aiming at scenarios that require channel adaptation. It generates random projections (RP) from weighted sums of input bits, similar to compressed sensing except that RCM deals with binary input symbols [16]. The generated RP symbols are used to form a constellation directly, without the need of any mapping or labeling. Channel adaptation is achieved by sending more or less RP symbols" [3, p. 1].

"As shown in [21], RCM can be used not only for pure channel coding, but also to perform joint source-channel coding for non-uniform memoryless sources or even for sources with memory. In these cases, RCM functions as a single block performing joint source-channel coding and modulation directly in a very simple manner. RCM is robust to changes in the channel quality, provides a smooth rate adaptation, and is robust to variations in the entropy of the source. RCM schemes were first designed using a brute force approach as in [21] and [18]. EXIT chart analysis was introduced in [39] for memoryless sources. In [30], we developed a density evolution analysis of RCM for uniform and non-uniform memoryless sources" [3, p. 1].

1.4 Proposed System in the Context of Coded Modulation

Coded modulation combines source coding, channel coding, and modulation into a single block, as shown in Figure 1.3.

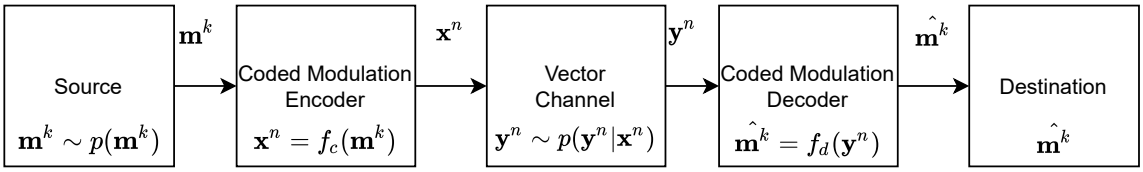


Figure 1.3: Point-to-point communications based on coded modulation.

We assume that the source produces a k-dimensional vector $\mathbf{m}^k = [m_1, ..., m_k]$ of symbols drawn from a joint probability distribution $p(\mathbf{m}^k)$. $\mathbf{m}^k$ is then encoded into an n-dimensional vector $\mathbf{x}^n = [x_1, ..., x_n]$ by the coded modulation block f_c. The sequence $\mathbf{x}^n$ is called a *codeword*, while the set of all possible codewords spanned by f_c and $\mathbf{m}^k$ is called *codebook* and mathematically defined as $\mathcal{C} = \{\, \mathbf{x}^n \,|\, \mathbf{x}^n = f_c(\mathbf{m}^k) \,\forall\, \mathbf{m}^k \in \mathcal{D}^k\}$, where $\mathcal{D}$ is the domain of the symbols produced by the information source. After encoding, a codeword is transmitted through the vector channel, which inserts noise to the transmitted codeword according to the joint probability distribution $p(\mathbf{y}^n|\mathbf{x}^n)$. The decoder block processes the channel output $\mathbf{y}^n$ and finds the sequence that minimizes a defined error metric between the estimated information vector $\hat{\mathbf{m}}^k$ and $\mathbf{m}^k$.

In communication schemes where compression is not performed and the channel has binary input and output, the data of both the source and the channel encoder are binary (i.e., $\mathbf{m}^k \in \{0,1\}^k$ and $\mathbf{x}^n \in \{0,1\}^n$), leading to a mapping $f_c : \{0,1\}^k \to \{0,1\}^n$ which is called a *binary code*. Low-Density-Parity-Check (LDPC) codes [6], Turbo codes [7], and binary BCH codes [40–42] are examples of codes that operate in the binary domain. Traditional error-correcting codes that operate in the binary domain are one-to-one transformations with $n > k$ and *coding rate* $R_c = k/n < 1$.

The described coded modulation scheme has the capability to map bits to the set of real numbers $\mathbb{R}$, and thus it is also suitable for high-throughput systems and joint source-channel coding, including RCM as a particular case.

1.5 Joint Source-Channel Coding

It is well known that a sequence of bits generated by an information source can be compressed up to its Shannon's entropy by the use of a source code [1]. In the same way, the amount of information bits per channel use that can be reliably transmitted through a noisy channel is upper bounded by the Shannon capacity, which can be approached by using a channel code [1]. While the separation theorem [1] proves that source and channel coding can be designed and implemented separately as the block length goes to infinity, when source and channel coding are designed together (joint source-channel coding), it is possible to outperform traditional coding schemes at finite block-length and also to decrease the computational complexity of the system.

Early works in joint source-channel coding focused on finding codes that use the source redundant data to increase the system robustness. Nevertheless, it is also important to design decoders that exploit the source statistics as *a priori* knowledge to provide better performance [12,13]. This dissertation considers the scenario of non-uniform memoryless binary sources, where each source bit is 1 with probability p_1 and 0 with probability $p_0 = 1 - p_1$. Joint source-channel coding for non-uniform memoryless sources was considered in [43] and [44] for Low-Density Parity-Check codes (LDPC), in [45–48] for Turbo codes, and in [17, 21, 30, 31] for Rate Compatible Modulation (RCM).

One way to approach the transmission of a non-uniform source is to design channel codes that are able to use the natural source redundancy in order to better protect the encoded message, as in [44,48]. In this case, the focus is the transformation (encoding) from k bits (message) into n bits (codeword). Another way of looking at the problem is to use conventional codes designed for uniform sources, and then to perform Unequal Energy Allocation (UEA), which assigns higher energy to channel symbols that have low probability, and low energy to channel symbols with high probability, as done in [46,47]. The main idea of UEA is to reduce the probability of error by increasing the distance between symbols while keeping the same energy usage. Different from the previous case, UEA acts upon the modulation block, which is the transformation from

the n bits codeword into m real-valued channel symbols. In [43, 45], both approaches are combined, in the sense that the source distribution is used to design better channel codes and also to improve the channel symbols for Turbo and LDPC codes. Notice that even though the source distribution is used for both error correction and modulation, both transformations are still performed separately. With this in mind, RCM is used in [17,21,30,31] with the idea of directly mapping source bits into channel symbols with a real-valued linear transformation. Since the proposed non-linear scheme performs direct mapping between bits and constellation symbols it has great potential for joint source-channel coding.

1.6 Channel Model and Theoretical Limits

In the one-dimensional case (1D), each channel input x_i is a real-valued random variable and the channel output is

$$y_i = x_i + r_i, \tag{1.1}$$

where r_i is the channel noise, with distribution $\mathcal{N}(0, N_0/2)$. For the two-dimensional case (2D), the difference is that all variables are now two-dimensional vectors

$$\mathbf{y_i} = \mathbf{x_i} + \mathbf{r_i}, \tag{1.2}$$

where each dimension of the vector $\mathbf{r_i}$ is i.i.d. channel noise with distribution $\mathcal{N}(0, N_0/2)$.

The channel capacity, C_{1D}, defined as the maximum information rate that can be transmitted through an AWGN channel for a given signal-to-noise ratio (SNR), is expressed as [49]

$$C_{1D} = \frac{1}{2}\log_2\left(1 + \frac{E_s}{\sigma_r^2}\right) = \frac{1}{2}\log_2\left(1 + \frac{2E_s}{N_0}\right), \tag{1.3}$$

where $E_s = \mathbb{E}[x^2]$ is the expected value of the energy per transmitted symbol and C_{1D} is given in bits per channel use. In the two-dimensional case we use two parallel AWGN

channels with energy $E_s/2$. Hence, the channel capacity for the two-dimensional system, denoted as C_{2D}, is expressed as

$$C_{2D} = \log_2\left(1 + \frac{E_s/2}{\sigma_r^2}\right) = \log_2\left(1 + \frac{E_s}{N_0}\right), \tag{1.4}$$

where $E_s = \mathbb{E}[\|\mathbf{x}\|^2]$.

In general, rate-distortion theory [49] establishes an asymptotically achievable lower bound of the information rate for a source to be compressed with an expected distortion of at most D. In other words, for a given expected distortion D (which can be zero), an information source can be compressed up to an information rate $R(D)$ which is a function of the acceptable distortion D. $R(D)$ is a monotonically decreasing convex function on D.

The Shannon separation theorem establishes that source and channel coding can be performed separately without loss in performance (for an infinite block length): once a source is compressed up to its rate-distortion function $R(D)$, we can transmit this information through a channel with capacity C. Therefore, arbitrarily low error rates can be achieved if and only if [49]

$$R_c R(D) < C, \tag{1.5}$$

where $R_c = k/n$ denotes the coding rate, defined by the ratio between the number of input bits, k, and number of output symbols n.

A binary non-uniform source is defined as an i.i.d. non-uniform Bernoulli random variable with probability of zero p_0 and probability of one $p_1 = 1 - p_0$. In this work, the distortion metric of binary sources is defined as the bit error rate (BER), leading to a rate-distortion function for binary sources expressed by [49]

$$R(D) = \begin{cases} H(p_0) - H(\mathrm{D}), 0 \leq D \leq \min(p_0, 1-p_0) \\ 0, \qquad\qquad\qquad\qquad D > \min(p_0, 1-p_0), \end{cases} \tag{1.6}$$

where $D = \text{BER}$ and $H(\cdot)$ is the Shannon entropy defined as

$$H(x) = -x\log_2(x) - (1-x)\log_2(1-x), \tag{1.7}$$

where $x \in (0,1)$ and $H(0) = H(1) = 0$, by definition.

Since $R_c R(D) < C$, where C is the capacity in bits per channel use, it follows that the achievable region for the one-dimensional case is

$$R_c(H(p_0) - H(\text{BER})) < \frac{1}{2}\log_2\left(1 + \frac{2E_s}{N_0}\right). \tag{1.8}$$

For the two-dimensional case,

$$R_c(H(p_0) - H(\text{BER})) < \log_2\left(1 + \frac{E_s}{N_0}\right). \tag{1.9}$$

Note that for the two-dimensional case, the transmitted symbol is actually a vector of two positions. Therefore, the average transmitted symbol energy is defined as $E_s = \mathbb{E}[\|\mathbf{x}\|^2]$.

In general, a non-binary source with Q levels, entropy $H(Q)$, and distortion measure equal to the symbol error rate (SER), has its rate-distortion function, $R(D)$, expressed in bits (generalized solution of the problem 10.5 in [49])

$$R(D) = \begin{cases} H(Q) - \text{SER}\log_2(Q-1) - H(\text{SER}), 0 \leq D \leq \min(p_0, ..., p_{Q-1}) \\ 0, \qquad\qquad\qquad\qquad\qquad\qquad D > \min(p_0, ..., p_{Q-1}), \end{cases} \tag{1.10}$$

where p_i with $i \in \{0, ..., Q-1\}$ represents the probability of the non-binary source to be equal to i. It follows that the achievable region when using one-dimensional constellation symbols is expressed as

$$R_c(H(Q) - \text{SER}\log_2(Q-1) - H(\text{SER})) < \frac{1}{2}\log_2\left(1 + \frac{2E_s}{N_0}\right). \tag{1.11}$$

Rate-distortion theory also provides a way to derive the theoretical limit (OPTA) for the transmission of a Gaussian source over an AWGN channel under the MSE metric. Since we transmit k i.i.d. Gaussian samples of variance 1 over n parallel and independent AWGN channels, we obtain

$$\begin{aligned} kR(D) &< nC, \\ \frac{k}{2}\log_2\frac{1}{\text{MSE}}, &< \frac{n}{2}\log_2\left(1+\frac{E_s}{\sigma_r^2}\right), \\ R_c\log_2\frac{1}{\text{MSE}}, &< \log_2\left(1+\frac{E_s}{\sigma_r^2}\right), \\ \text{SDR}_{\text{dB}} &< \frac{1}{R_c}10\log_{10}(1+\text{SNR}), \end{aligned} \tag{1.12}$$

where $R_c = k/n$ is the coding rate, D is the chosen distortion metric (D = MSE), $R(D)$ is the rate-distortion function of a Gaussian variable with distortion equal to its mean square error, and C is the channel capacity.

1.7 Dissertation Organization

This work explores the use of random graph-based codes to produce non-linear mappings for the transmission of several types of sources. Specifically, we consider the transmission of four types of sources and dedicate a chapter to each. In Chapter 1 we introduce the work developed in this dissertation. In Chapter 2 we discuss the proposed general framework and its application to binary uniform sources. In Chapter 3 we use the derivations and intuitive understanding from Chapter 2 to design mappings for the transmission of binary non-uniform sources. Chapter 4 generalizes one of the methods proposed in Chapter 2 for the transmission of uniform and non-uniform non-binary sources. Lastly, in Chapter 5 we expand the general framework developed in previous chapters for discrete alphabets to real-valued sources. Chapter 6 presents our conclusions.

Chapter 2

NON-LINEAR CODES BASED ON GRAPHS

From a high-level perspective, the proposed non-linear error-correcting codes are sparse graph-based codes that generate symbols by repeatedly applying a single non-linear encoding function to random subsets of the input bits. The use of a sparse graph allows us to use sub-optimal, but practical, message-passing decoders [28] such as the ones used for LDPC codes [50] and RCM [15].

Specifically, each symbol to be transmitted, x_i, is generated by a non-linear function f, whose arguments are an ordered subset of the input bits $\mathbf{m}^k = [m_1, ..., m_k]$. It is important for the subset to be defined with order since, in general, $f(x, y) \neq f(y, x)$. The codeword $\mathbf{x}^n = [x_1, ..., x_n]$ can be expressed as

$$\mathbf{x}^n = f_c(\mathbf{m}^k) = [x_1, ..., x_n] = [f(\mathcal{M}_1), ..., f(\mathcal{M}_n)], \tag{2.1}$$

where $\mathcal{M}_j$ denotes the ordered subset of input bits connected to the symbol x_j and each $x_j = f(\mathcal{M}_j)$ represents the encoding of the bits in $\mathcal{M}_j$ into the symbol x_j by the function f.

Formally, we define the undirected bipartite graph G as the tuple $(\mathbf{m}^k, \mathbf{x}^n, E)$ where, $\mathbf{x}^n = \{x_1, ..., x_n\}$ is the set of symbols (symbol nodes), $\mathbf{m}^k = \{m_1, ..., m_k\}$ is the set of input bits (input nodes), and E is the set of edges that connect the graph nodes. An edge e_{ij} connects the nodes m_i and x_j, $i \in \{1, ..., k\}, j \in \{1, ..., n\}$.

As previously stated, since $f(x, y) \neq f(y, x)$, we must define an order for the arguments of the function $x_j = f(\mathcal{M}_j)$. In order to do so, we associate an ordering number to each edge departing from the node x_j. Therefore, since f has d_r arguments, a symbol node j is connected to d_r input nodes and each edge e_{vj} has an associated

ordering number that defines the order of m_v in the generation of x_j through function f. For instance, if $w(e_{vj}) = 1$ then the input node m_v is the leftmost argument of f, if $w(e_{v',j}) = 2$ then $m_{v'}$ is the second leftmost argument of f, and so on.

The subset of all symbol nodes connected to an input node i is denoted as $\mathcal{X}_i = \{x_j \,|\, \exists(i,j)\,[e_{ij} \in E]\}$. The number of edges connected to a node is called *degree* and defined as $d_j = |\mathcal{M}_j|$ for the j-th symbol node and as $d_i = |\mathcal{X}_i|$ for the i-th input node, where $|.|$ denotes the cardinality. In this work, we fix $d_j = d_r \; \forall j$ since we use the same non-linear function f to generate each symbol node. We will explore more general possibilities in our future work.

As an example, Figure 2.1 depicts a (6,4) code. The input bits are connected by the encoding graph to the nodes that represent the symbols x_j. The coding rate $R_c = k/n$ for this system is $6/4 = 1.5$, which in other words means that the channel is used 4 times to transmit 6 bits of information. Note that for simplicity we assume the channel is one-dimensional, which means that it can only carry one symbol at each use. Higher dimensional channels can be used by grouping consecutive symbols.

The code in Figure 2.1 has degrees equal to $d_r = 3$ for the symbol nodes and $d_i = 2$ for the input nodes. It is a regular code since all input nodes have the same degree and the same happens for all the symbol nodes. As explained before, the ordering number associated to an edge e_{ij} determines the position of the input node m_i as an argument of the function f to generate the symbol node x_j. For instance, the input node m_4 is connected to x_3 by an edge with ordering number 1 and to x_1 by an edge with ordering number 3. Therefore, m_4 is the first argument of f when computing x_3, but the third argument of f when computing x_1. Notice that the (unordered) subset of symbol nodes connected to the input node m_4 is $\mathcal{X}_4 = \{x_1, x_3\}$. In contrast, the (ordered) subset of input nodes connected to the symbol node x_3 is $\mathcal{M}_3 = (m_4, m_1, m_2)$, which means $x_3 = f(\mathcal{M}_3) = f(m_4, m_1, m_2)$.

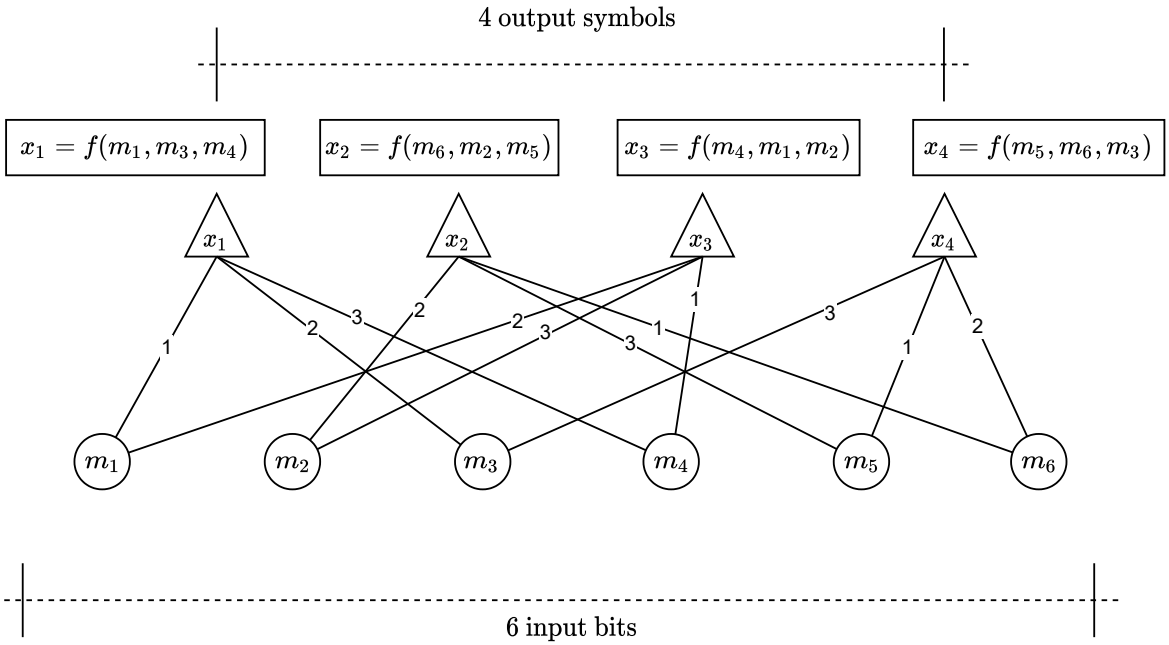


Figure 2.1: Proposed non-linear code encoding 6 input bits into 4 output symbols. Each symbol is generated by applying the function f to a subset of the input bits according to the underlying graph.

2.1 Encoder

As previously mentioned, the encoder f_c maps k input bits to n real-valued symbols by using a graph G and a function f with d_r arguments, so that all symbol nodes x_j in the graph G have a degree equal to d_r. Notice that once d_r is fixed, the function f and the graph G can be designed independently.

2.1.1 Graph Construction

As in the case of LDPC codes or RCM, the graph G is built in a random manner, but satisfying the following constraints:

1. Every symbol node, x_j, has d_r edges and each edge has a different ordering number $w(e_{ij}) \in \{1, ..., d_r\}$.

2. The degree of an input node, $d_i, i \in \{1, ..., k\}$, is either $\lfloor d_r/R_c \rfloor$ or $\lceil d_r/R_c \rceil$, where $\lceil . \rceil$ and $\lfloor . \rfloor$ are the ceiling and floor functions, respectively. This constraint makes the graph close to regular and simplifies the design. We will explore irregular structures in future work.

2.1.2 Encoding Function

"The real-valued symbol nodes generated by RCM are linear combinations of the input bits. However, there is no specific reason for the adoption of such a constrained linear mapping. Notice that in traditional binary coding schemes all operations are linear in GF(2), but not in the set of real numbers. For instance, if we consider the function $f(x, y)$ to be the binary sum (exclusive OR operation) between bits x and y, we have $f(0, 1) = f(1, 0) = 1$ and $f(0, 0) = f(1, 1) = 0$, making f non-linear in the real numbers" [3, p. 2].

"In order to consider more general schemes, we allow f to be a general non-linear function, which can be described by a lookup table that assigns a block of d_r input bits to a symbol in the desired constellation" [3, p. 2]. That is

$$f : \{0, 1\}^{d_r} \rightarrow \Gamma, \tag{2.2}$$

where Γ is the set of constellation points with cardinality $|\Gamma| = 2^{d_u}$. Notice that in general d_u does not need to be the same as d_r.

"In order to optimize the performance of the proposed scheme, we have to choose the constellation, the non-linear function f, and the underlying graphical structure" [3, p. 2]. This is a very difficult problem, and we only explore a few paths here. Since obtaining a good mapping f is a key part of the design of good codes, we dedicate a whole section to that challenge.

2.1.3 Constellation

"Any constellation of size $L = 2^{d_u}$, where $d_u \in \mathbb{N}$, can be utilized in the proposed scheme, independently of the underlying graphical structure" [3, p. 2]. For simplicity, in this work we focus on the case when $L = 2^{d_r}$.

"The simplest constellation to consider is standard PAM, consisting of constellation points $\mathcal{P} = \{p_1, ..., p_L\}$ defined, before normalization, as

$$p_n = 2n - (L + 1), n \in \{1, ..., L\}" \tag{2.3}$$

[3, p. 2].

"However, in order to achieve channel capacity in the AWGN channel, it is necessary for the channel input to follow a Gaussian distribution. Thus, we utilize a constellation that mimics a Gaussian distribution by using non-uniformly spaced points all of them occurring with the same probability [51]. Specifically, in order to choose the constellation points, we partition the Gaussian distribution into L equally likely portions, and then we choose the points that further subdivide each portion into two equally likely parts (centroids of each area). Specifically, we define the set of Gaussian constellation points $\mathcal{G} = \{g_1, ..., g_L\}$ as the ones satisfying [51]

$$\int_{-\infty}^{g_n} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx = \frac{2n-1}{2L}, \tag{2.4}$$

$n \in \{1, ..., L\}$" [3, p. 2]. (2.4) can also be written in terms of the inverse Q-function Q^{-1} as

$$g_n = -Q^{-1}\left(\frac{2n-1}{2L}\right), \tag{2.5}$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-x^2/2} dx$.

"Once the values of g_n are calculated, conventional normalization is performed in order to obtain the desired energy per symbol, E_s" [3, p. 2].

2.2 Decoder

"Message passing or belief propagation [28] is applied to perform decoding. It is important to remark that the non-linearity does not add any considerable complexity in comparison to the linear RCM decoder in [15]" [3, p. 3]. As in LDPC decoders, for a given Bernoulli variable U with $P(U = 1) = p$ and $P(U = 0) = 1 - p$, its Log Likelihood Ratio (LLR) $L(U)$ is defined as

$$L(U) = \log \frac{P(U = 1)}{P(U = 0)} = \log \frac{p}{1 - p}. \tag{2.6}$$

Below, we derive the equations for the message passing decoder by only using LLRs, reducing the risk of numerical inaccuracies due to lack of resolution when representing probabilities as floating point numbers.

2.2.1 Initialization

"We denote as $q_{ij}^{(t)}$ the message transferred from input node m_i to symbol node x_j at the t-th iteration, and as $s_{ij}^{(t)}$ the message transferred from symbol node x_j to input node m_i. The initial message from m_i to x_j for the case of uniform memoryless binary sources is

$$q_{ij}^{(1)} = \log \frac{P(m_i = 1)}{P(m_i = 0)} = 0 \tag{2.7}$$

" [3, p. 3].

2.2.2 Symbol Nodes

"The message passed from the j-th symbol node to the i-th bit at the t-th iteration is

$$s_{ij}^{(t)} = \log \frac{P(y_j|m_i = 1)}{P(y_j|m_i = 0)}. \tag{2.8}$$

Since we know that the received symbol $y_j = x_j + n_j$ depends on the AWGN noise and on the set of input bits connected to x_j, $\mathcal{M}_j$ (through x_j), we obtain

$$s_{ij}^{(t)} = \log \frac{\sum_{\mathbf{b}\in\{0,1\}^{d_r-1}} P(y_j, \mathcal{M}_{j\setminus i} = \mathbf{b}|m_i = 1)}{\sum_{\mathbf{b}\in\{0,1\}^{d_r-1}} P(y_j, \mathcal{M}_{j\setminus i} = \mathbf{b}|m_i = 0)}, \tag{2.9}$$

where $\mathcal{M}_{j\setminus i}$ denotes the subset of input bits connected to the j-th symbol node except for the i-th bit, and the sum over the binary vector $\mathbf{b} \in \{0,1\}^{d_r-1}$ expresses the evaluation of that probability over all possible binary vectors spanned by $\mathcal{M}_{j\setminus i}$" [3, p. 3].

In order to further expand (2.9) and to simplify its notation, we introduce the notation $\mathbf{b}^i = [b_1, ..., b_{i-1}, b_{i+1}, ..., b_n]$ (i.e., removing the i-th coordinate of the n-dimensional vector $\mathbf{b}$). By denoting the ordering number of edge e_{ij} as $w_{ij} = w(e_{ij})$, in the case of uniform sources $s_{ij}^{(t)}$ can be expressed as

$$\begin{aligned}
s_{ij}^{(t)} &= \log \frac{\sum_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=1}} P(y_j = f(\mathbf{b}) + n_j, \mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}}|m_i = 1)}{\sum_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=0}} P(y_j = f(\mathbf{b}) + n_j, \mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}}|m_i = 0)} \\
&= \log \frac{\sum_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=1}} \begin{aligned}[t] &P(y_j = f(\mathbf{b}) + n_j|\mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}}, m_i = 1)\times \\ &P(\mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}})\end{aligned}}{\sum_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=0}} \begin{aligned}[t] &P(y_j = f(\mathbf{b}) + n_j|\mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}}, m_i = 0)\times \\ &P(\mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}})\end{aligned}} \\
&= \log \frac{\sum_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=1}} P(y_j = f(\mathbf{b}) + n_j|\mathcal{M}_j = \mathbf{b})P(\mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}})}{\sum_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=0}} P(y_j = f(\mathbf{b}) + n_j|\mathcal{M}_j = \mathbf{b})P(\mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}})} \\
&= \log \frac{\sum_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=1}} e^{\frac{-(y_j - f(\mathbf{b}))^2}{N_0}} P(\mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}})}{\sum_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=0}} e^{\frac{-(y_j - f(\mathbf{b}))^2}{N_0}} P(\mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}})}.
\end{aligned} \tag{2.10}$$

Finally, by converting probabilities to their LLR forms, we obtain

$$s_{ij}^{(t)} = \log \frac{\sum\limits_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_i=1}} e^{\frac{-(y_j - f(\mathbf{b}))^2}{N_0} + <\mathbf{b}, \mathbf{q}_j^{(t-1)}> - q_{ij}^{(t-1)}}}{\sum\limits_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_i=0}} e^{\frac{-(y_j - f(\mathbf{b}))^2}{N_0} + <\mathbf{b}, \mathbf{q}_j^{(t-1)}>}}, \tag{2.11}$$

where $< \mathbf{a}, \mathbf{b} >$ denotes the inner product between vectors $\mathbf{a}$ and $\mathbf{b}$, and $\mathbf{q}_j^{(t-1)} = [q_{ij|w(e_{ij})=1}, ..., q_{ij|w(e_{ij})=d_r}]$ denotes the message vector ordered according to the ordering number of the respective edges that was transferred to symbol node x_j. In order to decrease the computational complexity of the decoder by avoiding recomputations, we first calculate $< \mathbf{b}, \mathbf{q}_j^{(t-1)} >$ and $(y_j - f(\mathbf{b}))^2$ for all $\mathbf{b} \in \{0,1\}^{d_r}$ and reuse those quantities for all edges departing a given symbol node j.

"Intuitively, one may think that the non-linear decoder should have much higher computational complexity than its linear counterpart. However, this is not the case. Indeed, the only difference between the RCM decoder and the proposed one is that, instead of performing convolutions (which are a consequence of linearity), here we just look for values in a mapping table. The proposed decoder utilizes more memory than traditional RCM, since we have to store 2^{d_r} entries, whereas RCM only stores d_r weights. However, this is not an important factor because the memory usage for the lookup table is still low" [3, p. 3].

2.2.3 Input Nodes

"Since we are not making any changes on the information bits, the message passing algorithm is the same as in LDPC codes" [3, p. 2]. Thus, the outgoing message from the i-th bit to the j-th symbol node is the sum of all incoming messages except the one from the j-th symbol node. That is,

$$q_{ij}^{(t)} = \sum_{v \text{ s.t. } e_{iv}\in E, v\neq j} s_{iv}^{(t-1)}. \tag{2.12}$$

2.2.4 Decision

"The decision process is also the same as in LDPC codes. Therefore, we sum all incoming messages at input node i, leading to

$$d_i^{(t)} = \sum_{v \text{ s.t. } e_{iv} \in E} s_{iv}^{(t-1)}. \tag{2.13}$$

where the i-th bit is decided to be 1 if $d_i^{(t)} > 0$ and 0 otherwise" [3, p. 3].

2.3 Mapping Design

"As previously mentioned, the mapping (f function) assigns binary vectors into single constellation points. Notice that this is not necessarily a one-to-one mapping, since more than one binary vector may be assigned to the same constellation point. Finding a good mapping is a complex combinatorial problem, and the number of possible mappings increases in a factorial manner since, in general, there are $!2^{d_r}$ different possibilities" [3, p. 2].

In general, we define the mapping by constructing a table with 2^{d_r} rows and d_r columns, so that each row contains a possible binary input vector which is mapped into one constellation point. We denote the Hamming distance between two binary vectors $\mathbf{x}$ and $\mathbf{y}$ as $H(\mathbf{x}, \mathbf{y})$.

Different from RCM, the non-linear behavior of our proposed scheme creates very pronounced waterfall and error floor regions, which are very dependent on the chosen parameters. As for LDPC and Turbo codes, the waterfall region is defined as the SNR where the code has an abrupt fall in its BER. In a similar way, the error floor region is defined as the region in which the code does not present a large improvement in BER even when the SNR increases substantially. In this section, we present two ways of constructing mappings. Both designs take into account the Hamming distances between the input binary vectors $\mathbf{b} \in \{0, 1\}^{d_r}$ and the Euclidean distances between their corresponding constellation points.

2.3.1 Design Parameter

From the perspective of the proposed framework, the proposed codes can be seen as a generalization of LDGM codes, since LDGM codes are also sparse graph-based codes with a function f that computes the modulo 2 sum (XOR operation) of all input bits and transmits $+1$ if the XOR operation is even or -1 if it is odd. In addition, LDGM codes can be seen as a specific case of LDPC codes where the systematic bits are transmitted and the redundancy bits are connected by only one edge to a check node. While this considerably simplifies the encoder process, the aforementioned degree one leads to error floors much higher than for general LDPC codes, which is not a problem since these error floors can be reduced by concatenation with another error correcting code. Since the proposed scheme can be seen as a generalization of LDGM codes, it will also have high error floors, which can be further understood by studying the error floor of LDGM codes.

In [52], the error floor of an LDGM code is studied by considering the BER when the incoming messages at a check node are perfect. By applying the same method to our proposed scheme, if we had a symbol node, x_i, where all the connected input bits were known, all incoming messages to the i-th symbol would be equal to positive or negative infinity (0 or 1 when converted to the probability domain), making the message calculated in (2.10) a function of the Euclidean distance between input binary vectors with Hamming distance 1. Thus, the methods we propose for mapping construction, aiming at different tradeoffs between the waterfall and the error floor region, focus on varying the Euclidean distance of constellation symbols whose corresponding input vectors are at Hamming distance 1 from each other (more details are provided in Appendix A).

2.3.2 Construction Algorithms

In this section we present the proposed design guidelines for the non-linear function f, which as explained before is replicated in each symbol node. The two proposed methods are based on varying the Euclidean distance of the constellation

symbols whose corresponding input binary vectors have Hamming distance 1. However, this is a complex combinatorial and optimization problem without a known solution, and these methods do not attempt to guarantee optimality, but rather propose simple ways of achieving good performance for a range of rates.

2.3.2.1 Parity Intersect Method

The parity of a binary sequence $\mathbf{x}$, $P(\mathbf{x})$ is defined as the modulo 2 sum of its elements (XOR operation). Recall that function f maps $\{0,1\}^{d_r}$ to the constellation set Γ. We arrange Γ in ascending order, to define the assignments in a systematic way. Notice that two binary vectors that differ in one position (Hamming distance 1), also have different parities. In the Parity Intersect method, we use this property to create parity constraint vectors [1] that indirectly control the Hamming distances of the mapping, consequently changing the error floor and waterfall of the code. Specifically, we generate parity constraints vectors by intersecting in different ways vectors that have different parities. For a mapping M_i, we specify the parity constraint vector of length 2^{d_r} as

$$P_i = \left[\underbrace{0, \cdots, 0}_{2^{d_r-1}-i}, \underbrace{1, 0, 1, 0, \cdots, 1, 0, 1, 0}_{2i}, \underbrace{1, \cdots, 1}_{2^{d_r-1}-i} \right]. \tag{2.14}$$

Note that as i increases the zeros and ones in P_i become more entangled, allowing the constellation symbols associated to input vectors with Hamming distance 1 to get closer (in Euclidean distance) to each other.

In order to generate a mapping with a desired parity constraint vector, P_i, we start with an initial mapping as indicated in Table 2.1. Then, starting from the top, we fill each row of the desired mapping by transferring the first (topmost) available binary vector of the initial mapping that has not been yet transferred and that satisfies P_i.

[1] The parity constraint vector for a given mapping specifies the parity of the input vector associated to each constellation symbol, where the constellation symbols are ordered in an ascending manner.

Table 2.1: Examples of mappings $M_i : (b_1, b_2, b_3) \to \Gamma$ by using the Parity Intersection method for $d_r = 3$, Gaussian constellation $\Gamma = G$, and a Gray code as Initial Mapping. P_i represents the parity constraint vector for mapping M_i.

Initial Mapping		
b_1	b_2	b_3
0	0	0
0	0	1
0	1	1
0	1	0
1	1	0
1	1	1
1	0	1
1	0	0

Mapping M_0				
b_1	b_2	b_3	Γ=G	P_0
0	0	0	-1.53	0
0	1	1	-0.88	0
1	1	0	-0.48	0
1	0	1	-0.15	0
0	0	1	+0.15	1
0	1	0	+0.48	1
1	1	1	+0.88	1
1	0	0	+1.53	1

Mapping M_1				
b_1	b_2	b_3	Γ=G	P_1
0	0	0	-1.53	0
0	1	1	-0.88	0
1	1	0	-0.48	0
0	0	1	-0.15	1
1	0	1	+0.15	0
0	1	0	+0.48	1
1	1	1	+0.88	1
1	0	0	+1.53	1

Mapping M_2				
b_1	b_2	b_3	Γ=G	P_2
0	0	0	-1.53	0
0	1	1	-0.88	0
0	0	1	-0.48	1
1	1	0	-0.15	0
0	1	0	+0.15	1
1	0	1	+0.48	0
1	1	1	+0.88	1
1	0	0	+1.53	1

Notice that the generated mapping will depend on both the desired parity constraint vector and the initial mapping.

In order to study the effects of both the initial mapping and the parity constraint vector P_i, we will choose two different initial mappings. The first one is the binary reflected Gray code, which we will choose due to its widespread use and simple construction. The second is a mapping shuffled at random. Since i in (2.14) ranges from 0 to $2^{d_r-1} - 1$, the mapping $M_{2^{d_r-1}-1}$ for the initial Gray code will be the Gray code itself, which is not expected to produce good results since the input vectors corresponding to all neighboring constellation symbols will be at Hamming distance 1 from each other. An initial random mapping is expected to have a better performance for $i = 2^{d_r-1} - 1$.

2.3.2.2 Distance Profile Method

In this method, the mappings M_i are constructed according to the Hamming distance of the input vectors associated to the neighboring constellation symbols. A distance profile is designed and a mapping that fits the desired Hamming distance constraints is randomly constructed.

As in the Parity Intersect method, the mapping function $f : \{0,1\}^{d_r} \rightarrow \Gamma$ is sorted in ascending order according to its constellation set $\Gamma = [\Gamma_0, \Gamma_1, ..., \Gamma_{2^{d_r}-1}]$ where $\Gamma_k < \Gamma_{k+1}\ \forall k$. Moreover, the binary vectors $\mathbf{b^k} \in \{0,1\}^{d_r},\ k \in \{0, ..., 2^{d_r} - 1\}$ are defined such that $f(\mathbf{b^k}) = \Gamma_k$[2]. By defining the desired Hamming distance between binary vectors that correspond to constellation symbols that are next to each other, we define the distance profile constraint that will be used when constructing a mapping. Specifically, we define the first-order Hamming distance profile $\mathbf{d_1}(M)$ for a mapping M as

$$\begin{aligned}\mathbf{d_1}(M) &= \left[H(\mathbf{b^0}, \mathbf{b^1}), H(\mathbf{b^1}, \mathbf{b^2}), \cdots, H(\mathbf{b^{2^{d_r}-2}}, \mathbf{b^{2^{d_r}-1}})\right] \\ &= \left[H(\mathbf{b^{k-1}}, \mathbf{b^k})\right] \quad k \in \{1, ..., 2^{d_r} - 1\}.\end{aligned}$$

In order to construct a mapping M, we first specify its first-order Hamming distance profile $\mathbf{d_1}^{\alpha,\beta}(M) = \left[H(\mathbf{b^{k-1}}, \mathbf{b^k})\right]$, where its k-th position $H(\mathbf{b^{k-1}}, \mathbf{b^k})$ is defined as

$$H^i(\mathbf{b^{k-1}}, \mathbf{b^k}) = \begin{cases} \beta, & \text{if } k = 2^{d_r-1} + 2j\ \forall j \in \mathbb{Z} \text{ s.t.} -i \leq j \leq i \\ \alpha, & \text{otherwise,} \end{cases} \tag{2.15}$$

where β is a positive odd integer, α is a positive even integer, and i is a non-negative

[2] Notice that $\mathbf{b^k}$ can be any binary vector. $\mathbf{k}$ is used to indicate that it will be associated to constellation symbol Γ_k

integer. Thus,

$$\begin{aligned}\mathbf{d_1}^{\alpha,\beta}(M_i) &= \left[H(\mathbf{b^0},\mathbf{b^1}), H(\mathbf{b^1},\mathbf{b^2}), \cdots, H(\mathbf{b^{2^{d_r}-2}},\mathbf{b^{2^{d_r}-1}})\right] \\ &= \left[\underbrace{\alpha,\cdots,\alpha}_{2^{d_r-1}-1-2i},\underbrace{\beta,\alpha,\cdots,\beta,\alpha,\beta,\alpha,\beta,\cdots,\alpha,\beta}_{1+4i}\ \underbrace{\alpha,\cdots,\alpha}_{2^{d_r-1}-1-2i}\right].\end{aligned} \tag{2.16}$$

The idea of generating $\mathbf{d_1}^{\alpha,\beta}(M_i)$ in this manner is to produce mappings that look like random but with different distance profiles, allowing us to explore the tradeoff between waterfall and convergence threshold. If α takes the largest possible value, which is $\alpha = d_r - 1$ for d_r odd and $\alpha = d_r - 2$ for d_r even, we force the binary vectors associated to consecutive constellation points to change greatly.

Similarly to the Parity Intersect method, when the Hamming distance between two binary sequences $\mathbf{x}$ and $\mathbf{y}$, $H(\mathbf{x},\mathbf{y})$, is odd, their parities will be different. Hence, when $H^i(\mathbf{b^{k-1}},\mathbf{b^k}) = \beta$ (odd), the parities of $\mathbf{b^{k-1}}$ and $\mathbf{b^k}$ will flip, while $\mathbf{b^{k-1}}$ and $\mathbf{b^k}$ will have the same parity when $H^i(\mathbf{b^{k-1}},\mathbf{b^k}) = \alpha$ (even).

2.4 Results

In this section, we present Monte Carlo simulations to assess the performance of the proposed codes. For each data point, we transmit 200 blocks of fixed length by encoding $k = 30000$ input bits into $n = k/R_c$ channel symbols, which are then transmitted through an AWGN channel. Moreover, whenever not indicated otherwise, all simulations are performed by using the Gaussian constellation defined in (2.5). When the first 20 blocks present an average bit error rate higher than 0.1, the remaining blocks are not simulated for that SNR. As previously mentioned, the proposed codes present large error floors, which in practice does not lead to problems since the error floor can be reduced by using concatenated structures such as LDPC or LDGM codes.

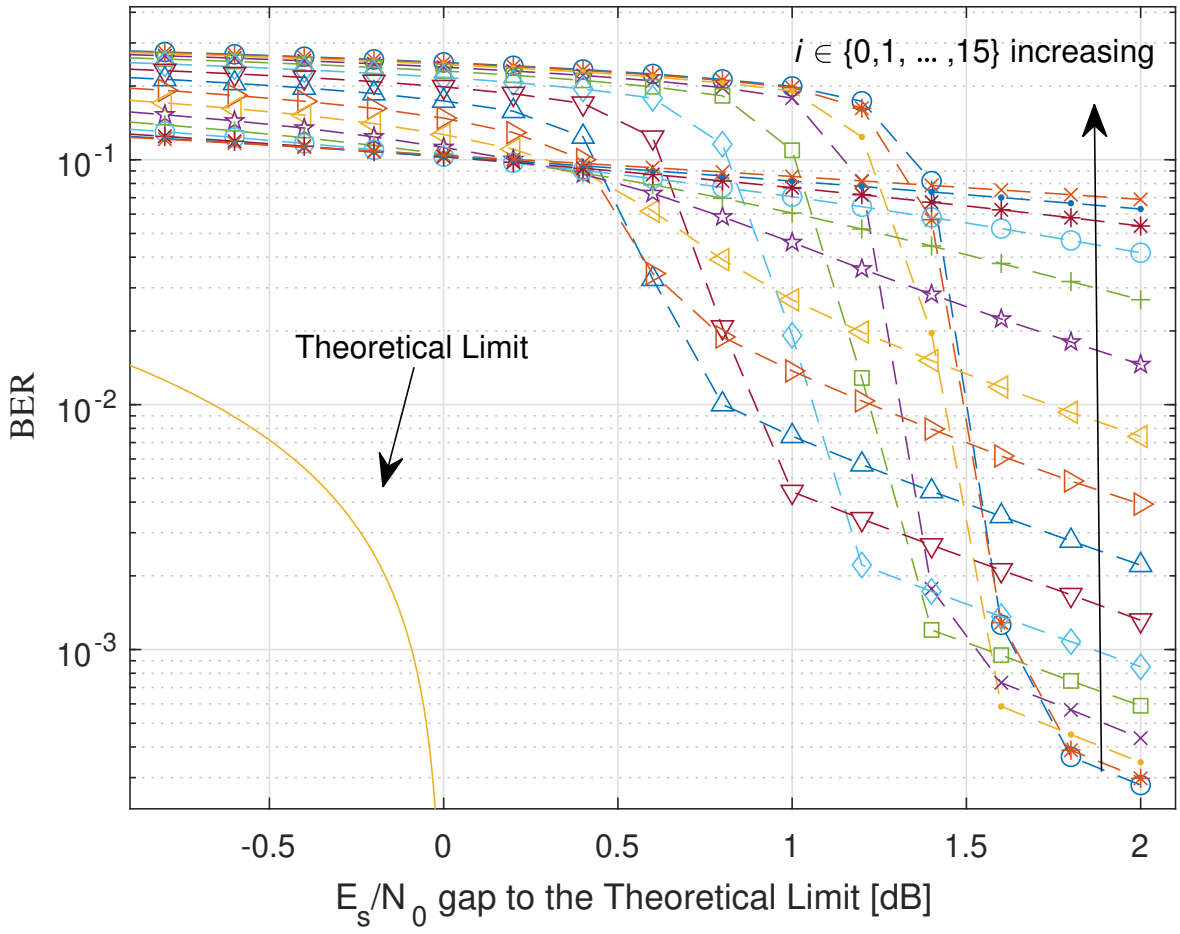


Figure 2.2: Parity Intersect method using a Gray code as initial mapping and a Gaussian constellation with parity constraints P_i, $R_c = 1$ and $d_r = 5$.

2.4.1 Parity Intersect Method

We start by presenting results for the Parity Intersect method. Figure 2.2 shows the results of the Monte Carlo simulations when a Gray code is used as initial mapping for $d_r = 5$ and $R_c = 1$. As expected, by varying i we can control the tradeoff between error floor and waterfall. As i increases, the parities become more intertwined, leading to a smaller Euclidean distance between input vectors at Hamming distance 1 from each other, which consequently rises the error floor. Conversely, as the error floor increases, the waterfall improves by moving to the left, closer to the theoretical limit.

Motivated by the pseudo-randomness of capacity-approaching error correcting

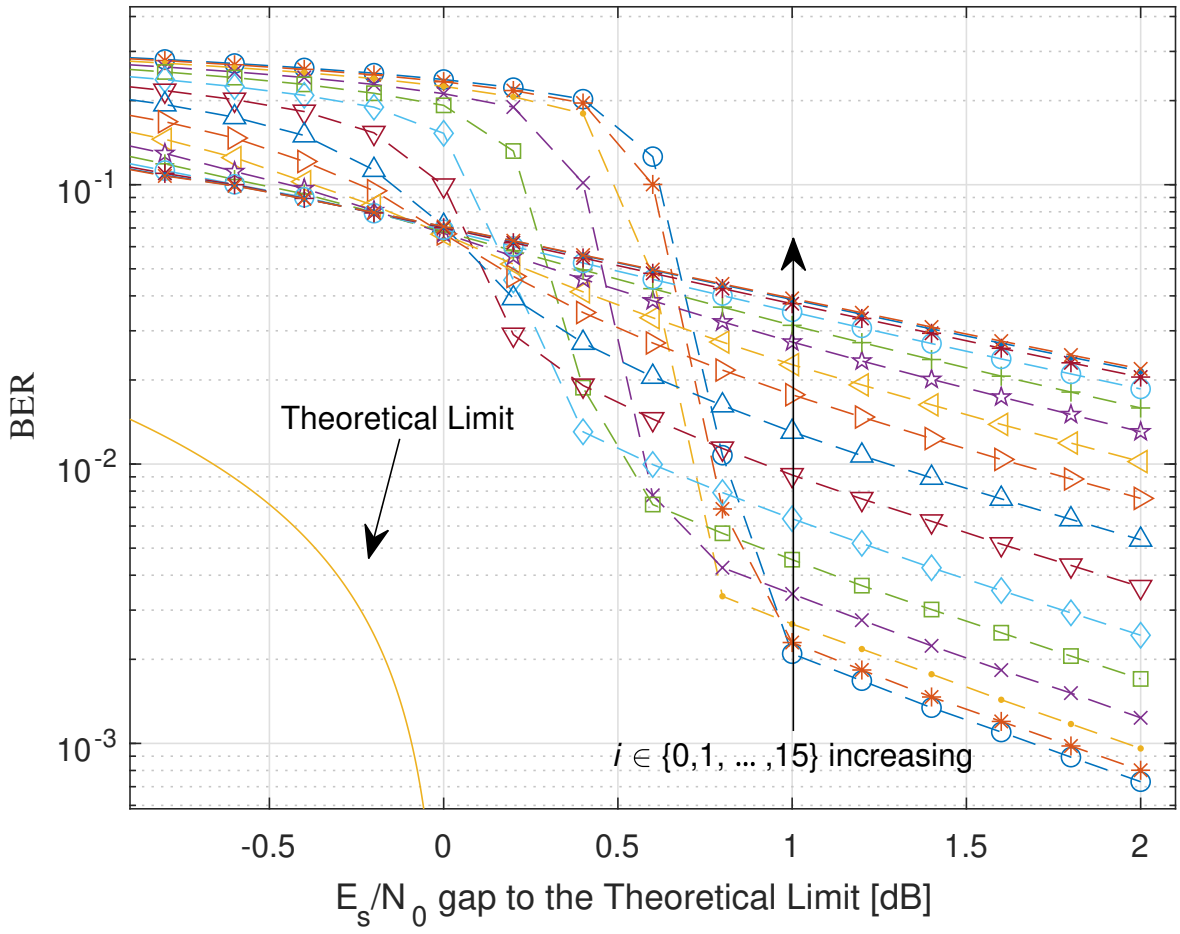


Figure 2.3: Parity Intersect method using an initial mapping that is random and a Gaussian constellation with parity constraints P_i, $R_c = 1$ and $d_r = 5$.

codes, we present in Figure 2.3 the simulation results for the same setup as Figure 2.2, except that the initial mapping is random. As expected, the tradeoff between error floor and waterfall is still controlled by the intersection between parities according to i. However, the random structure present in the mapping considerably improves the waterfall, while the error floor is slightly degraded. Remarkably, BER $< .1$ for SNRs below the Shannon limit ($E_s/N_{0[gap]} < 0$ dB), which is something that does not happen for conventional codes, AFC, or RCM. It is worth noting that even without optimizing the initial mappings in order to improve the waterfall or the error floor, the Parity Intersect method provides a very simple way of constructing functions that are able to

provide very good performance and flexibility in terms of the waterfall and error floor tradeoff. These results corroborate the discussion in section 2.3.1 of how the Euclidean distance of input vectors at Hamming distance 1 affects the error floor.

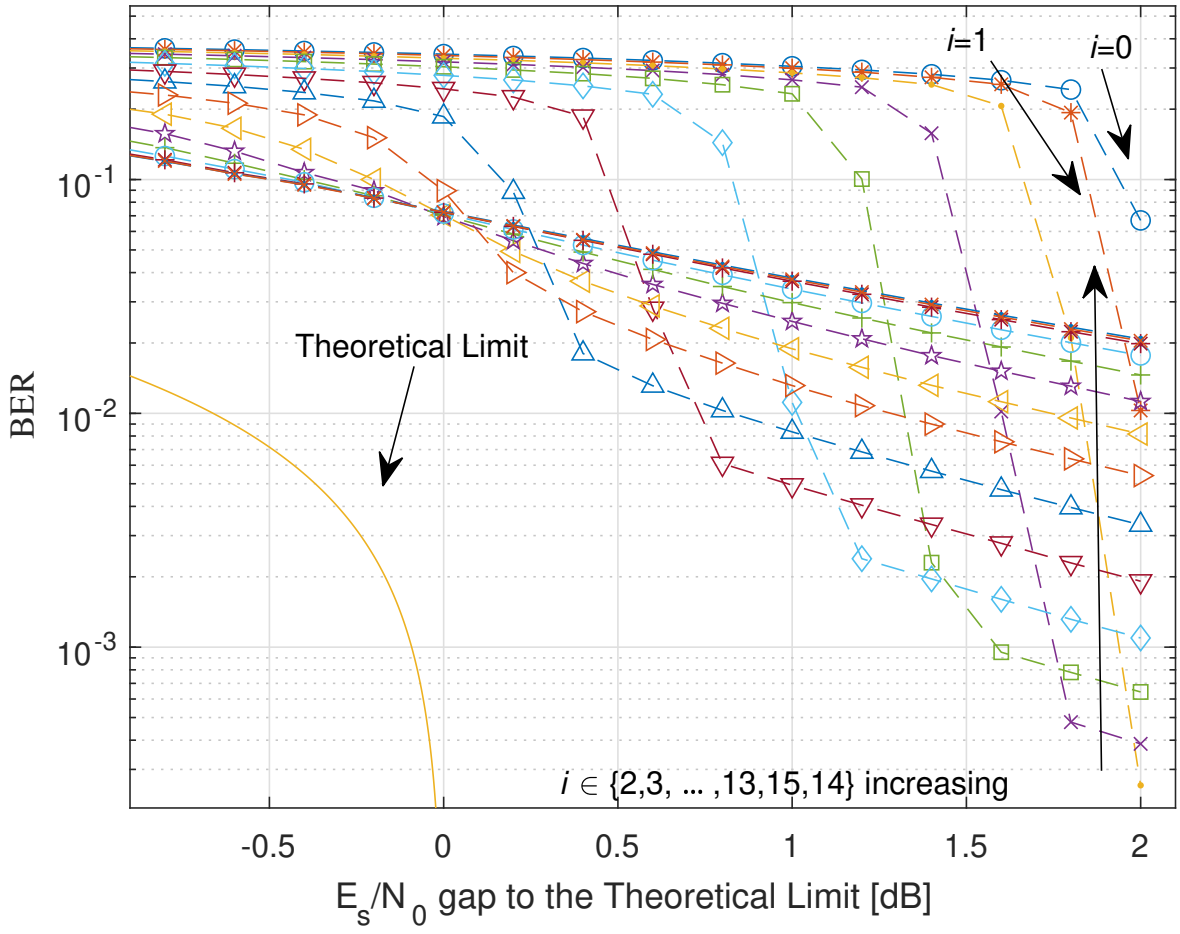


Figure 2.4: Parity Intersect method using an initial mapping that is random and a PAM constellation with parity constraints P_i, $R_c = 1$ and $d_r = 5$.

As it is well known, the channel symbols must follow a Normal distribution to achieve capacity, which led us to use the Gaussian constellation from (2.5) in the previous figures. In order to study how the constellation affects the results of the proposed methods, we present in Figure 2.4 results for the same setup as in Figure 2.3, except that we now use a PAM constellation as defined in (2.3). On the one hand, it is easy to notice that the use of a PAM constellation leads to a worsened waterfall for

all mappings. On the other hand, the error floor improves for all mappings, once again demonstrating the tradeoff between error floor and waterfall.

2.4.2 Distance Profile Method

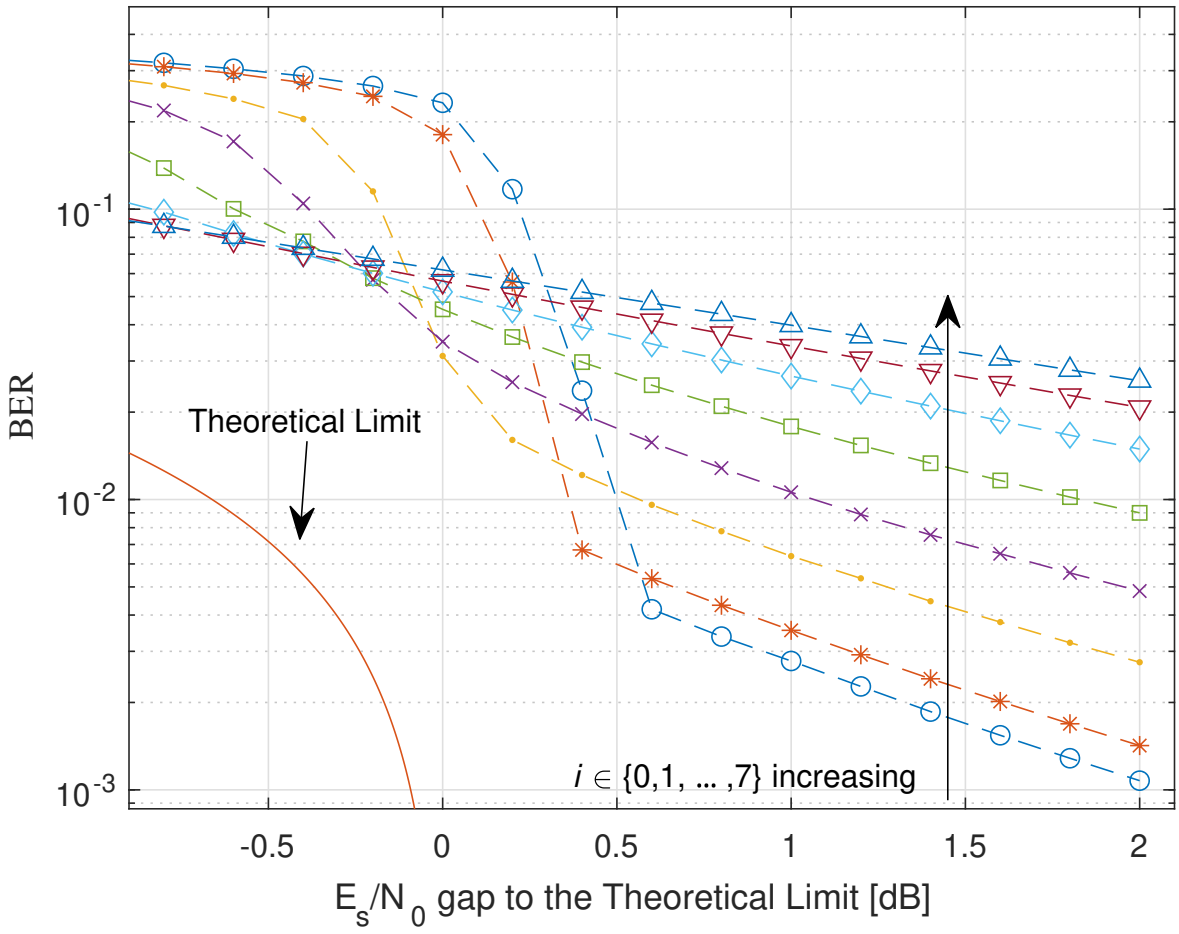


Figure 2.5: Distance profile method with $\mathbf{d_1}^{\alpha=4,\beta=5}$, $R_c = 1$ and $d_r = 5$ (adapted from [3]).

Next, we present results for the Distance Profile method. Figure 2.5 depicts the simulation results for $d_r = 5$ and the family of mappings with distance profile $\mathbf{d_1}^{\alpha=4,\beta=5}(M_i)$. As previously mentioned, every time two neighboring input vectors are at an odd Hamming distance, their parities flip, inducing a parity profile (similar to the one from the Parity Intersect method). However, in addition to the parities, we also control the Hamming distance between input vectors associated to neighboring

constellation symbols. Figure 2.5 shows that the Distance Profile method outperforms the Parity Intersect method for the region below the Shannon Limit ($E_s/N_{0[gap]} < 0$ dB), remarkably achieving an error rate lower than .1 (BER = 0.04 at $E_s/N_{0[gap]} = 0$ dB).

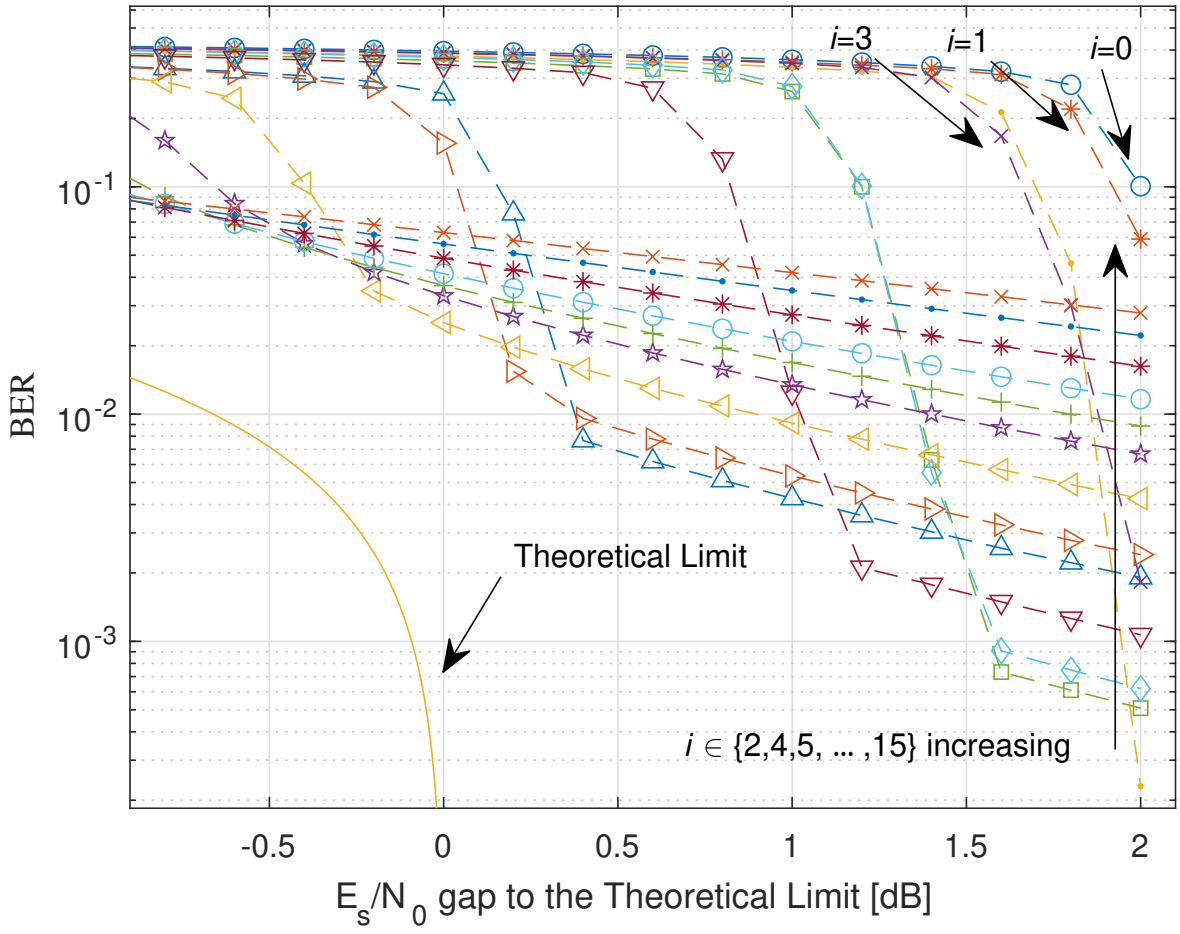


Figure 2.6: Distance profile method with $\mathbf{d_1}^{\alpha=4,\beta=5}$, $R_c = 1$, $d_r = 6$, and $H(\mathbf{b^k}, \mathbf{b^{k+2}}) \neq 2\ \forall k$.

Following the results for the Distance Profile method with $\mathbf{d_1}^{\alpha=4,\beta=5}$, $d_r = 5$ and $R_c = 1$, we present in Figure 2.6 the results for the same setup but with $d_r = 6$ instead. It is remarkable that the proposed codes approach the theoretical limit closely when the signal-to-noise ratio is below the Shannon limit.

Next, we present results for $R_c = 2$, $R_c = 3$ and $R_c = 4$ showing that our

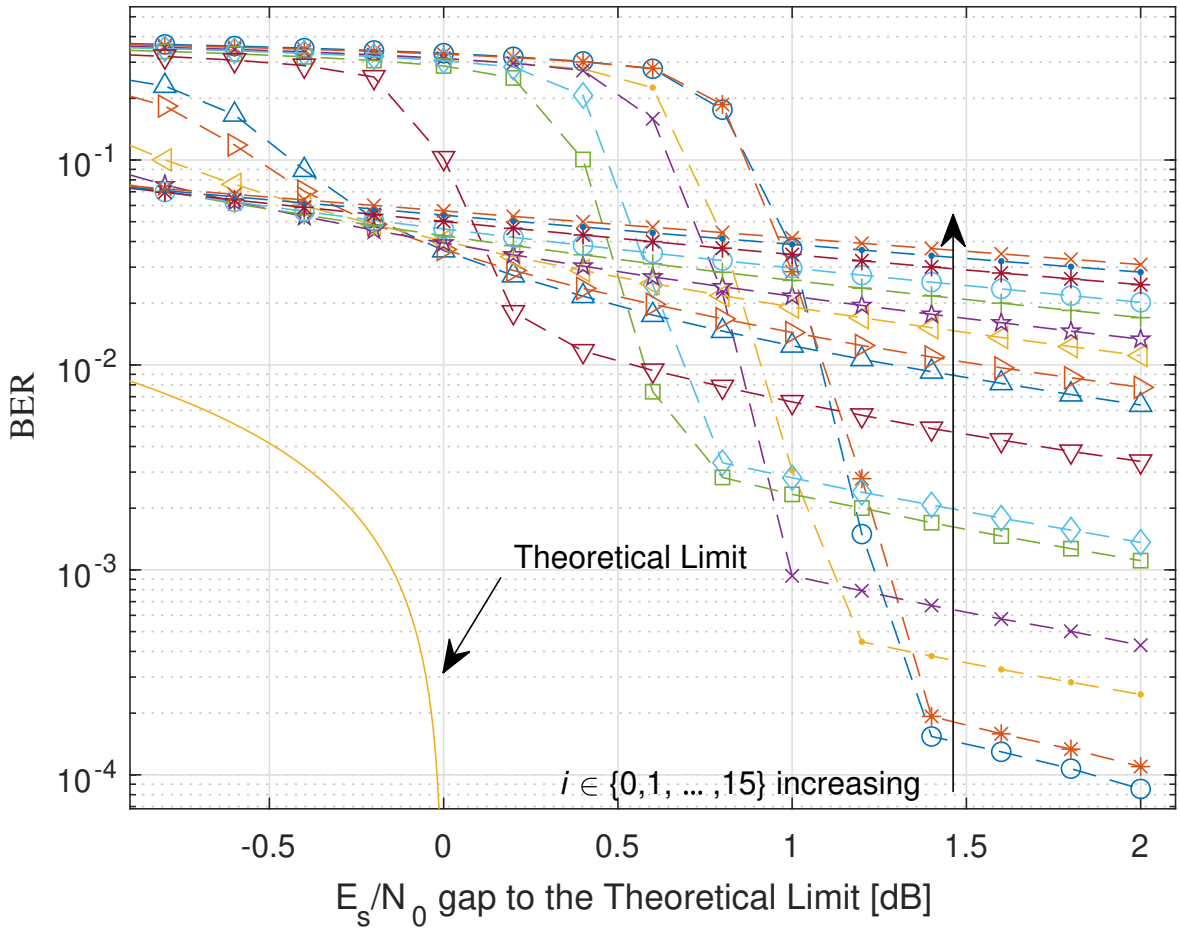


Figure 2.7: Distance profile method with $\mathbf{d_1}^{\alpha=4,\beta=5}$, $R_c = 2$, $d_r = 6$, and $H(\mathbf{b^k}, \mathbf{b^{k+2}}) \neq 2\ \forall k$.

methods can be also used for communications at higher rates. Figure 2.7 shows the results for the same setup as Figure 2.6 but for $R_c = 2$. For the cases of $R_c = 3$ and $R_c = 4$, in order to keep a good connectivity of the input nodes we set $d_r = 8$. Figures 2.8 and 2.9 show simulation results for the Distance Profile method for $R_c = 3$ and $R_c = 4$, respectively, for the family of mappings constructed with first order Hamming profile $\mathbf{d_1}^{\alpha=6,\beta=5}$. In Figure 2.8 we see that for $R_c = 3$ the performance at $E_s/N_{0[gap]} < 0$ dB is similar to that of conventional codes. However, as the SNR increases, we see a steep improvement, reaching values of BER$= 10^{-4}$ without any sort of concatenation. For the case of $R_c = 4$ in Figure 2.9, we see an excellent

performance at low SNR where $E_s/N_{0[gap]} < 0$ dB, reaching a value of BER= 0.03 at $E_s/N_{0[gap]} = 0$ dB. Moreover, we are also able to reach error floors around 10^{-4} for $E_s/N_{0[gap]} = 2$ dB.

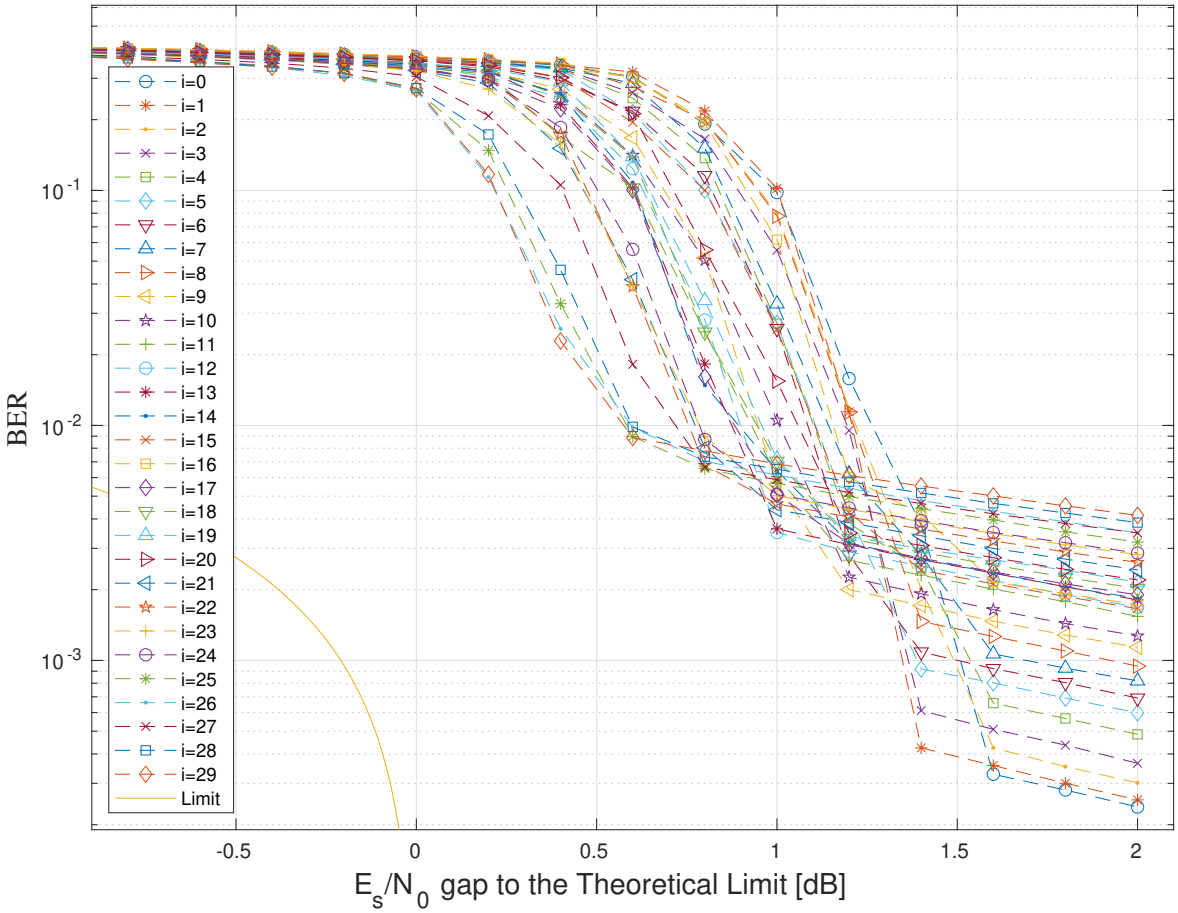


Figure 2.8: Distance profile method with $\mathbf{d_1}^{\alpha=6,\beta=5}$, $R_c = 3$, $d_r = 8$, and $H(\mathbf{b^k}, \mathbf{b^{k+2}}) \neq 2\ \forall k$.

2.4.3 Comparison with Existing Linear Codes

Finally, in Figures 2.10 and 2.11 we compare the performance between the Distance Profile method and linear codes utilized in the literature for $d_r = 8$. For the linear schemes, we construct the graph as for the other mappings in this work and repeatedly use the function f constructed by the linear mappings found in the literature with equation $\mathbf{x} = \langle \mathbf{w}, \mathbf{b} \rangle$, where $\mathbf{x}$ are the channel symbols, $\mathbf{w}$ is the weight

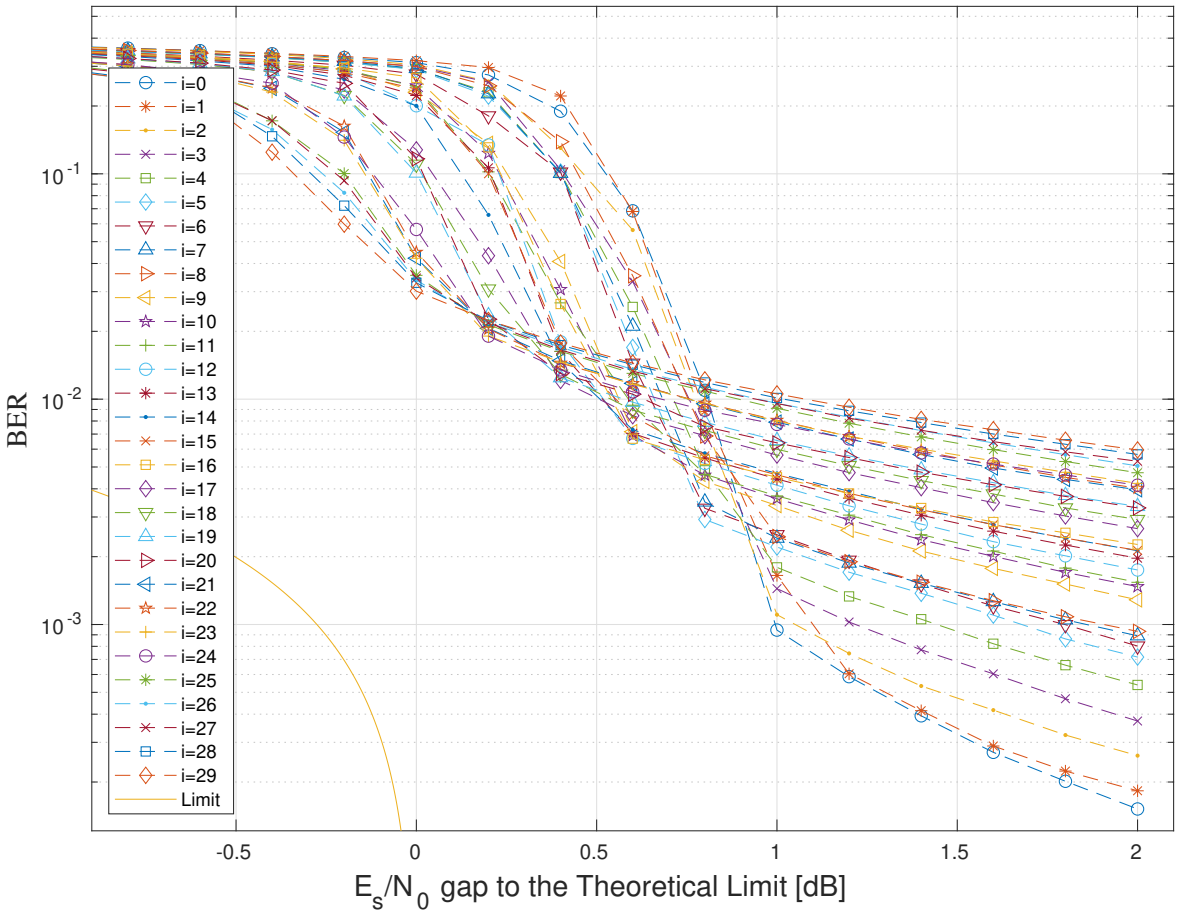


Figure 2.9: Distance profile method with $\mathbf{d_1}^{\alpha=6,\beta=5}$, $R_c = 4$, $d_r = 8$, and $H(\mathbf{b^k}, \mathbf{b^{k+2}}) \neq 2\ \forall k$.

set, and $\mathbf{b}$ are the values of the input bits connected to that symbol node, represented as -1 when 0 and $+1$ when 1. The weight set used for AFC is $W_{AFC} = \{1/2, -1/3, 1/5, -1/7, 1/11, -1/13, 1/17, -1/19\}$, which was slightly modified from [25] in order to improve the convergence of the decoder since we are not using it in a concatenated setting. Moreover, note that AFC originally has different mappings in each symbol node, which are drawn with repetition from W_{AFC}. However, for comparison purposes, we choose a fixed W_{AFC} as shown above since we are considering just one function in our non-linear scheme. In the same manner, we use the following weight sets for RCM schemes: $W_{RCM_A} = \{1, 2, 4, 4, -1, -2, -4, -4\}$ [15] and $W_{RCM_B} = \{2, 3, 4, 8, -2, -3, -4, -8\}$ [18]. Different from AFC, in the RCM case each

element of $\mathbf{b}$ is a 0 or 1, as typically done for RCM.

Figure 2.10 compares the linear schemes from the literature with the proposed non-linear codes with mappings $\mathbf{d_1}^{\alpha=4,\beta=5}$ for $d_r = 6$ and rates $R_c = 1$ and $R_c = 2$. For $R_c = 1$, our proposed code with $i = 12$ performs considerably better than AFC and RCM for all SNRs. Moreover, we see that both AFC and RCM have very high error floors, much higher than our proposed non-linear scheme. For $R_c = 2$, the mapping with $i = 9$ outperforms AFC and RCM for all SNRs. Our proposed code with $i = 12$ is much better than the linear schemes for a wide SNR range.

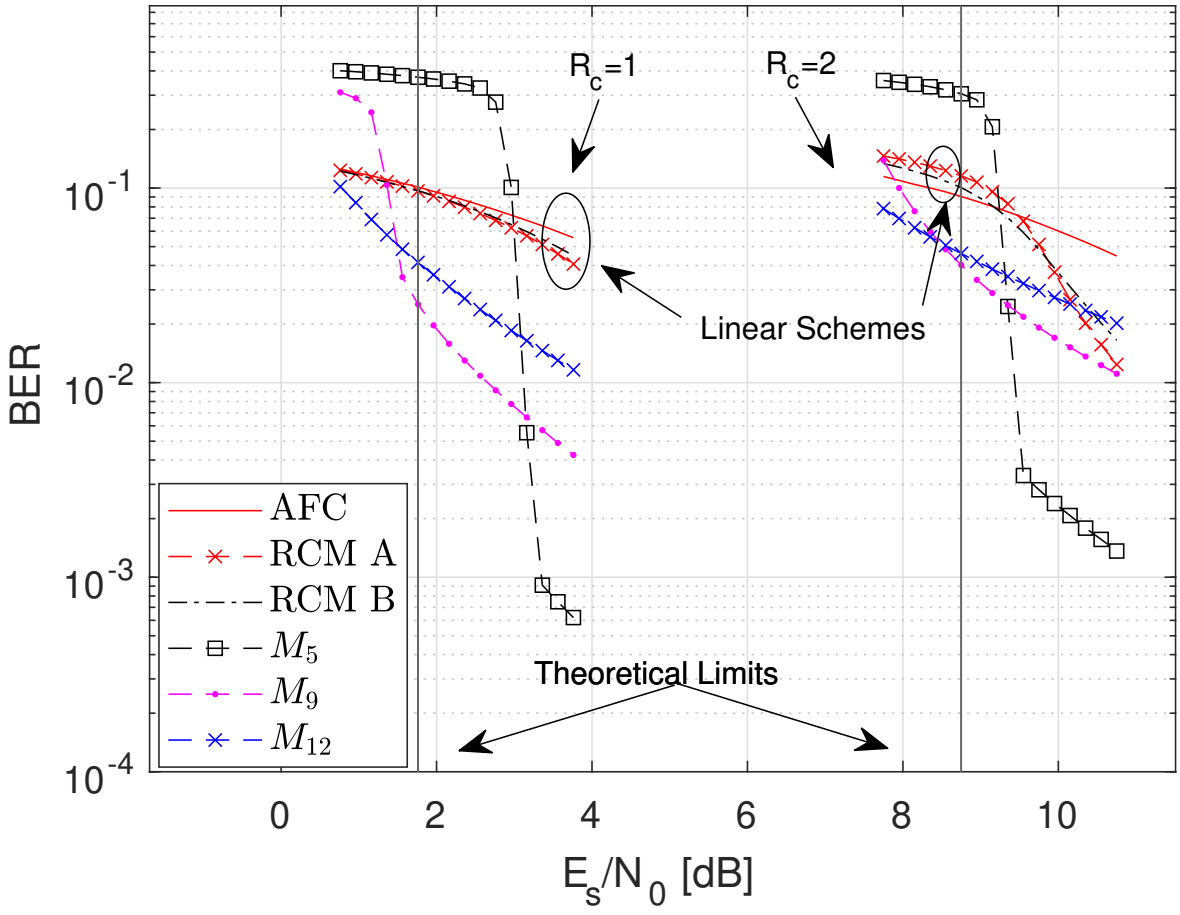


Figure 2.10: Comparison between our proposed non-linear codes designed using the Distance Profile method and existing linear codes in the literature when the rates are $R_c = 1$ and $R_c = 2$.

It is also of interest to perform the same comparison for higher rates. The

proposed non-linear codes use the mappings $\mathbf{d_1}^{\alpha=6,\beta=5}$ with $d_r = 8$. Figure 2.11 shows that for $R_c = 3$ we are able to considerably reduce the error floor with respect to the linear schemes. In the case of $R_c = 4$, we find non-linear codes that considerably outperform the existing linear codes by lowering the BER from 0.1 to 0.01 at signal-to-noise ratios below the theoretical limit.

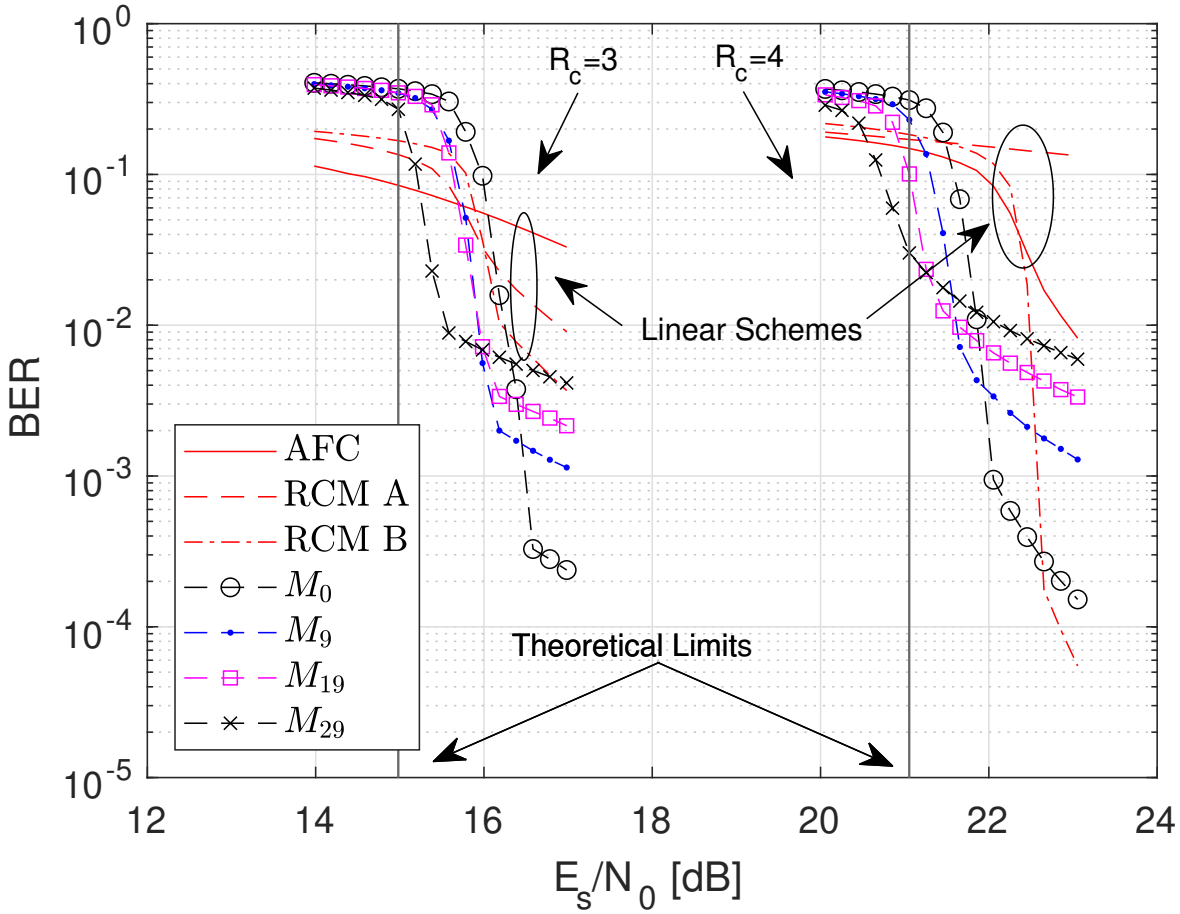


Figure 2.11: Comparison between our proposed non-linear codes designed using the Distance Profile method and existing linear codes in the literature when the rates are $R_c = 3$ and $R_c = 4$.

2.5 Conclusion

We have proposed two methods to construct non-linear codes that perform random-like encoding through a graph structure. Both methods are related to each

other and to LDGM codes, since we vary how the parities of the input sequences are mapped onto the channel symbols, allowing us to explore a tradeoff between waterfall and error floor. The first method, Parity Intersect, shows how varying the parities allows us to explore this tradeoff, whereas the second method, Distance Profile, varies the parities and includes additional constraints, leading to results that outperform the best linear codes existing in the literature. Different from conventional systems that perform modulation and error correction in a separate fashion, the proposed methods perform these tasks jointly, presenting more flexibility. Remarkably, the proposed non-linear codes are capable to approach the theoretical limit at signal-to-noise ratios below the Shannon limit, which is of great interest since this does not occur in conventional linear error correcting codes.

Chapter 3

NON-LINEAR CODING FOR NON-UNIFORM BINARY SOURCES

In this chapter, we apply the proposed non-linear codes to perform joint source-channel coding. As shown in [21, 24, 53], RCM achieves good performance for joint source-channel coding due to its ability to assign energy to the constellation points in an efficient way. Intuitively, the key idea is to provide higher energy to symbols that occur with lower probability. There are three main ideas behind energy allocation for joint source-channel coding of a non-uniform source. The first is related to how costly it is to send a symbol. If a symbol occurs with low probability, it will not have a considerable weight on the overall energy consumption. Therefore, there is a low cost of associating a low probability symbol to a high energy constellation point. The second is that less likely symbols actually carry more information about the source than symbols that occur frequently. Since the iterative decoder uses the source statistics to decode the received symbols, it will start by assuming that most symbols are part of the most likely sequences. While this is true, if the low probability symbols are neglected due to their low energy, they will have a very low chance of being successfully decoded, significantly increasing the error rate. The third is that a Gaussian-like constellation is necessary to achieve channel capacity. While assigning high energy to low probability symbols does not necessarily mean the constellation is Gaussian-like, the resulting constellation somewhat resembles Gaussianity. The non-linear mappings proposed in this dissertation are very well suited to perform energy allocation as explained above.

In this dissertation, we present two schemes for joint source-channel coding of non-uniform binary sources. The first one is a generalization of LDGM codes but without transmitting the systematic bits. The second scheme designs a transformation

that maps the non-uniform input bits onto two-dimensional constellation symbols using a graphical structure similar to the one defined in Chapter 2.

3.1 One-Dimensional Scheme

As in Chapter 2, we repeatedly apply the function f to randomly chosen subsets of the input bits, denoted as $\mathcal{M}_j$, which will in turn generate the output symbol x_j. The scheme presented in this section is one-dimensional since the output of f is a scalar, as in the case of uniform sources. The codeword to be transmitted $\mathbf{x}^n = [x_1, ..., x_n]$ is expressed as

$$\mathbf{x}^n = f_c(\mathbf{m}^k) = [x_1, ..., x_n] = [f(\mathcal{M}_1), ..., f(\mathcal{M}_n)]. \tag{3.1}$$

Formally, the one-dimensional mapping f has the following structure

$$f : \{0, 1\}^{d_r} \rightarrow \Gamma, \tag{3.2}$$

where Γ is the set of constellation points.

3.1.1 Encoder

The Hamming weight of a sequence $\mathbf{b} = [b_1, ..., b_n]$ is defined as the number of non-zero components and denoted as $W(\mathbf{b})$. Note that if the Hamming weight of a binary sequence is odd, the XOR sum of the sequence $\mathbf{b}$ is one, and it is zero otherwise. Hence, if an LDGM code (without systematic bits) is considered, all coded bits connected to a subset of input bits with odd (even) weight are equal to +1 (-1). This type of mapping achieves good performance for uniform sources, where the decoding process starts with initial knowledge of the transmitted (noisy) systematic bits. In the case of non-uniform sources, the prior knowledge of p_0 can be used to initialize the decoding process, but for high information rates LDGM codes experience performance degradation because each output symbol can transmit at most one information bit. This is the rationale of generalizing the XOR operation by increasing the constellation size.

We define the transformation $f : \{0,1\}^{d_r} \to \Gamma \subset \mathbb{R}$ as

$$f(\mathbf{b}) = \begin{cases} W(\mathbf{b}) & \text{if } W(\mathbf{b}) \text{ is odd} \\ -(W(\mathbf{b})+1) & \text{if } W(\mathbf{b}) \text{ is even,} \end{cases} \tag{3.3}$$

where Γ is the not yet normalized constellation, $|\Gamma| = d_r + 1$ and $\mathbf{b}$ is the vector of input bits connected to a symbol node. Figure 3.1 depicts the proposed transformation for $d_r = 3$, as well as the probability of symbol $f(\mathbf{b})$ occurring when $p_0 = 0.9$. As mentioned, the function f is a generalization of the XOR operation in which vectors of input bits are assigned to a constellation symbol according to their Hamming weight. Notice from (3.3) that as the probability of an input vector decreases (i.e., the Hamming weight increases if we assume $p_0 > 0.5$), the energy of its corresponding constellation symbol increases[1]. Before transmission, the constellation is normalized to have zero mean and desired average energy per symbol E_s.

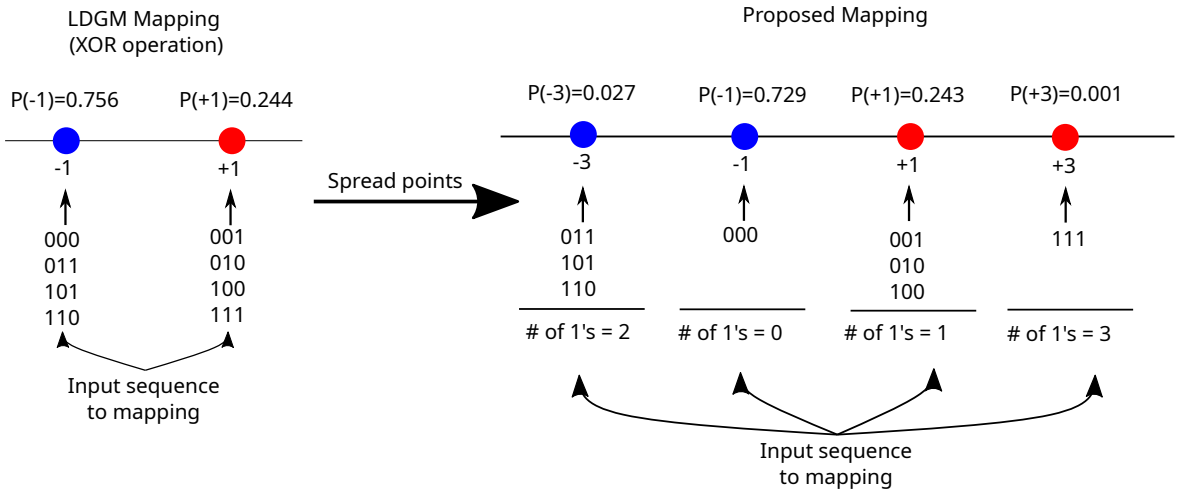


Figure 3.1: Proposed function f and probability of the constellation symbols when $p_0 = 0.9$.

As in the uniform case, each symbol node x_i is generated using the same function

[1] While the location of constellation points could be adjusted to improve performance, this does not lead to substantial improvements.

f (i.e., $x_i = f(\mathcal{M}_i)$, where $\mathcal{M}_i$ is the subset of information bits connected to x_i). Notice that the assigned constellation symbol is a function of the number of ones in $\mathcal{M}_i$ and it does not depend on their order. Thus, different from the case of uniform sources, ordering numbers are not required.

We assume $p_0 > 0.5$ so that the proposed mapping in (3.3) assigns less energy to more likely input vectors, as previously discussed. For the case of $p_0 < 0.5$ all we have to do is to invert the bits in $\mathbf{b}$.

Since $\mathbf{b}$ is composed of i.i.d. non-uniform Bernoulli variables, $W(\mathbf{b})$ is a binomially distributed random variable with probability mass

$$P(W(\mathbf{b}) = K) = \binom{d_r}{K} p_0^{d_r - K} (1 - p_0)^K. \tag{3.4}$$

For low d_r and fixed p_0, the probability of $W(\mathbf{b})$ is monotonically decreasing, providing appropriate energy allocation for the symbols x_i. As d_r increases, or p_0 decreases, $P(W(\mathbf{b}) = K)$ reaches a point at which it stops being monotonic. As shown in Appendix B, the necessary and sufficient condition for $P(W(\mathbf{b}))$ in (3.4) to provide a monotonic energy allocation when $p_0 > 0.5$ is given by

$$d_r < \frac{1 + \sqrt{1 + 8\left(\frac{p_0}{1-p_0}\right)^2}}{2}. \tag{3.5}$$

Figure 3.2 provides, for each value of p_0, the maximum value of d_r that generates a monotonic decreasing energy allocation. When p_0 increases, $P(W(\mathbf{b}))$ can have a monotonic distribution for higher values of d_r. Intuitively, we expect that the optimal value for d_r will be higher in the case of joint source-channel coding, since for non-uniform sources each input bit contains less information than in the uniform case. Thus, using a higher value for d_r allows an output symbol to be connected to more input bits so that the output symbol is able to carry the same amount of information as in the uniform case.

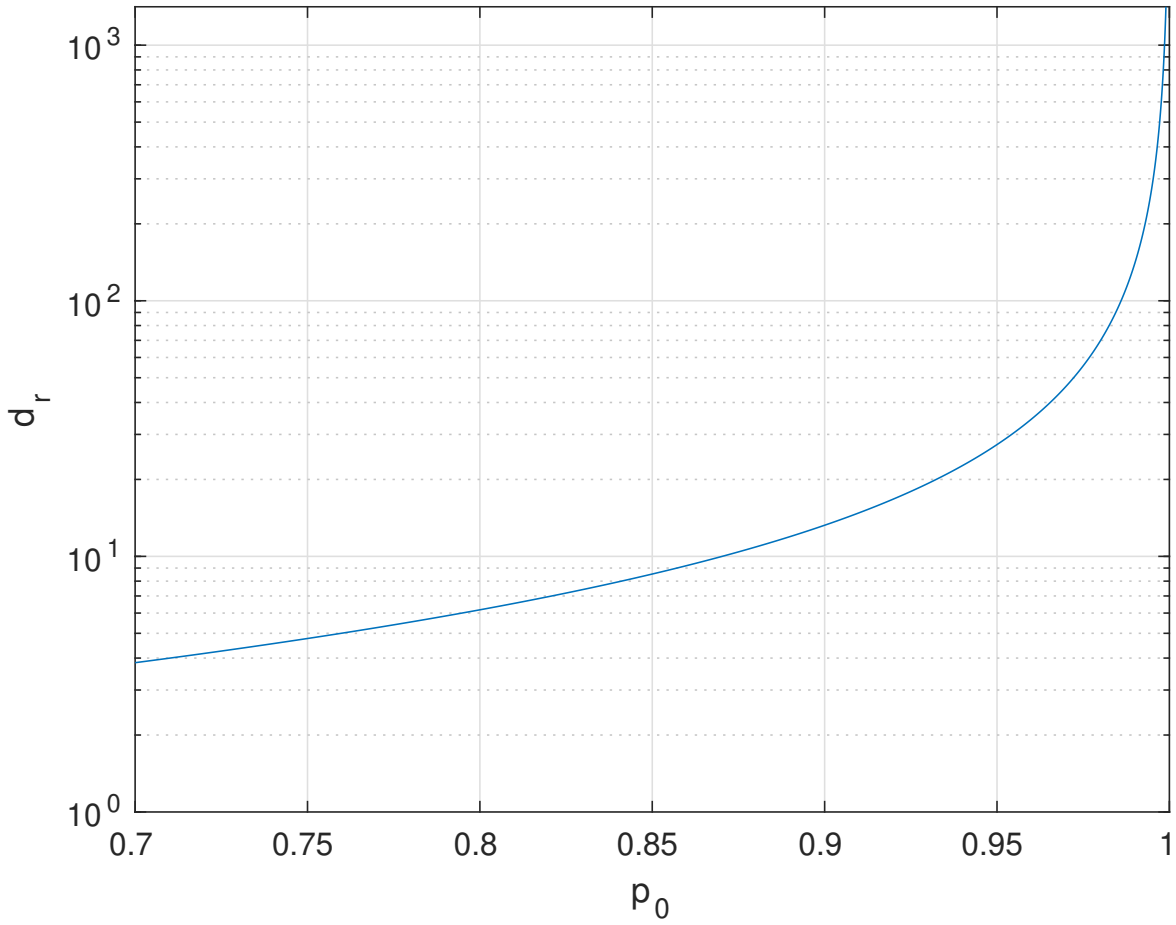


Figure 3.2: For each value p_0, maximum value of d_r that guarantees a monotonic decreasing energy allocation.

3.1.2 Decoder

As in Chapter 2, we apply the belief propagation decoder [28], except that now we include the prior knowledge of the source, p_0. While in the sequel we assume that p_0 is known, in a practical scenario p_0 could also be estimated by the source in an adaptive manner.

While the general decoder can be directly used, we are interested in decreasing its computational complexity, since d_r is higher than for the typical uniform case. Therefore, we present a particular implementation of the general decoder that takes advantage of the structure existing in the proposed mapping to reduce the necessary

amount of calculations. The proposed simplification uses the Discrete Fourier Transform to calculate the probabilities of a Poisson binomial Distribution, as derived in [54].

The Poisson binomial distribution is a general case of the binomial distribution. While the binomial distribution is defined by $P(\sum_{i=1}^{n} X_i = k)$ where the X_i's are i.i.d. Bernoulli variables with $P(X_i = 0) = p_0$, in the Poisson binomial distribution the Bernoulli random variables are independent but not identically distributed, i.e., $P(X_i = 0) = p_0^i$. In order to compute $P(\sum_{i=1}^{n} X_i = k)$ for the Poisson binomial distribution we could just simply count all possibilities, but it is possible to reduce the computational cost by using the Fast Fourier Transform (FFT) method presented in [54].

3.1.2.1 Initialization

As in Chapter 2, in order to reduce numerical inaccuracies, the messages are represented as log-likelihood ratios (LLR). The message transferred from input node m_i to symbol node x_j at the t-th iteration is denoted as $q_{ij}^{(t)}$, which for $t = 0$ is expressed as

$$q_{ij}^{(1)} = \log \frac{P(m_i = 1)}{P(m_i = 0)} = \log \frac{1 - p_0}{p_0}. \tag{3.6}$$

3.1.2.2 Symbol Nodes

We denote as $s_{ij}^{(t)}$ the message passed from the j-th symbol node to the i-th bit node at the t-th iteration

$$\begin{aligned} s_{ij}^{(t)} &= \log \frac{P(y_j|m_i = 1)}{P(y_j|m_i = 0)} \\ &= \log \frac{\sum\limits_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=1}} e^{\frac{-(y_j - f(\mathbf{b}))^2}{N_0}} P(\mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}})}{\sum\limits_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=0}} e^{\frac{-(y_j - f(\mathbf{b}))^2}{N_0}} P(\mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}})}. \end{aligned} \tag{3.7}$$

Notice that since we have $d_r + 1$ constellation points and all vectors of the same Hamming weight are mapped into the same constellation point, we can group (3.7) according to the Hamming weight of the input sequences. That is,

$$
\begin{aligned}
s_{ij}^{(t)} &= \log \frac{\sum\limits_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=1}} e^{\frac{-(y_j - f(\mathbf{b}))^2}{N_0}} P(\mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}})}{\sum\limits_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=0}} e^{\frac{-(y_j - f(\mathbf{b}))^2}{N_0}} P(\mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}})} \\
&= \log \frac{\sum\limits_{w=1}^{d_r} e^{\frac{-(y_j - g(w))^2}{N_0}} \left(\sum\limits_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=1 \\ \text{and } W(\mathbf{b})=w}} P(\mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}}) \right)}{\sum\limits_{w=0}^{d_r-1} e^{\frac{-(y_j - g(w))^2}{N_0}} \left(\sum\limits_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=0 \\ \text{and } W(\mathbf{b})=w}} P(\mathcal{M}_{j\setminus i} = \mathbf{b}^{w_{ij}}) \right)} \\
&= \log \frac{\sum\limits_{w=1}^{d_r} e^{-\frac{(y_j - g(w))^2}{N_0}} P(W(\mathcal{M}_{j\setminus i}) = w-1)}{\sum\limits_{w=0}^{d_r-1} e^{-\frac{(y_j - g(w))^2}{N_0}} P(W(\mathcal{M}_{j\setminus i}) = w)},
\end{aligned}
\tag{3.8}
$$

where $g(w)$ denotes the constellation point associated with the input vector with Hamming weight w, and $P(W(\mathcal{M}_{j\setminus i}))$ is calculated by using the messages $q_{ij}^{(t)}$. Instead of computing $P(W(\mathcal{M}_{j\setminus i}) = w)$ in a naive manner, the algorithm presented in [54] allows us to efficiently compute that distribution by using the FFT, which considerably reduces the decoding complexity. Note that this is an exact method and not an approximation. However, some performance loss with respect to the general non-linear decoder can be expected, since the calculations are not performed in the LLR domain.

3.1.2.3 Input Nodes

By denoting the edge that connects the source bit m_i to the symbol node m_v as e_{iv}, we compute the outgoing message $q_{ij}^{(t)}$ as

$$q_{ij}^{(t)} = \sum_{v \text{ s.t. } e_{iv} \in E, v \neq j} s_{iv}^{(t-1)} + \log \frac{1-p_0}{p_0}. \tag{3.9}$$

3.1.2.4 Decision

In order to decide if a decoded bit is a 1 or 0, we compute the quantity $d_i^{(t)}$, which is expressed as

$$d_i^{(t)} = \sum_{v \text{ s.t. } e_{iv} \in E} s_{iv}^{(t-1)} + \log \frac{1-p_0}{p_0}, \tag{3.10}$$

where, as before, the i-th bit is decided to be 1 if $d_i^{(t)} > 0$ and 0 otherwise.

As for RCM, it is important to consider the computational complexity due to the high number of operations performed at the decoder. This is especially true for the case of non-uniform sources because, as the source non-uniformity increases, a higher d_r is necessary to obtain good performance. The computational complexity of the general non-linear decoder increases exponentially with d_r. Therefore, the proposed simplified decoder is critical for the implementation of the proposed scheme. Even though the complexity of the simplified decoder is still not linear on d_r, it is computationally efficient since the FFT performed at the symbol nodes has complexity $\mathcal{O}(d_r \log d_r)$.

3.1.3 Results

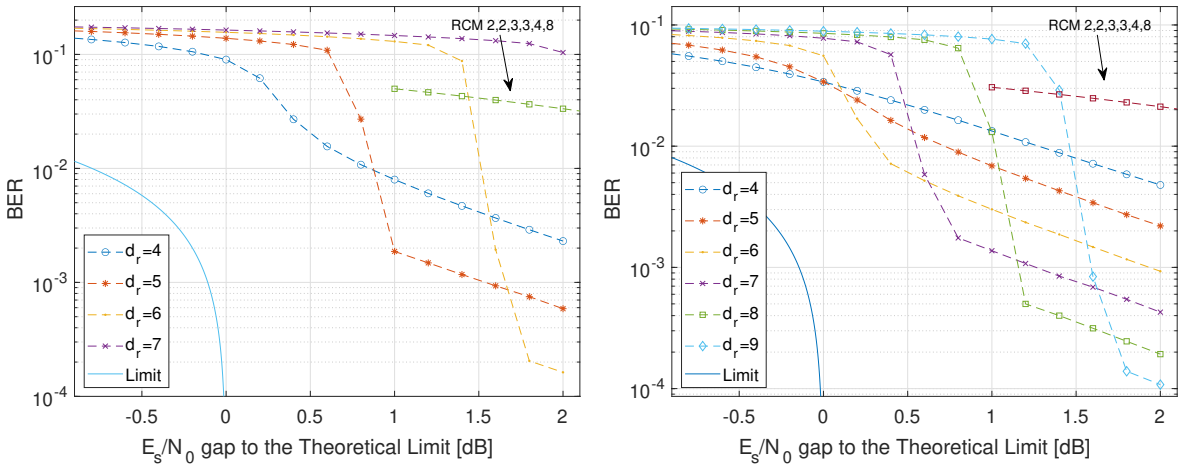
In this section we present the simulation results of the proposed non-linear codes for joint source-channel coding of non-uniform binary sources using one dimensional constellations. Unless otherwise indicated, we transmit 200 blocks with 30000 input bits and $30000/R_c$ output symbols over an AWGN channel. Decoding is performed using the belief propagation decoder described in Section 3.1.2, using 200 iterations at most. It is worth mentioning that while R_c determines the coding rate, the information

rate per channel use is actually much lower, and equal to $R_c H(p_0)$, which explains why it is possible to successfully use high values for R_c with only $d_r + 1$ constellation points.

Figure 3.3 shows the results of the proposed non-linear codes for $R_c = 1$ and several values of p_0. As in the uniform case, it is possible to control the tradeoff between waterfall and error floor. As d_r increases the error floor decreases, but the codes require a higher SNR to reach the error floor. Note that this characteristic holds for all values of p_0 in Figure 3.3. By comparing the results for several values of d_r with the bound provided in Figure 3.2, we can notice that for values of d_r that satisfy the bound, the waterfall is good. For instance, when $p_0 = 0.8$, the condition for monotonically decreasing energy allocation is $d_r \leq 6$, which as seen in Figure 3.3a, lead to convergence within 2 dB from the Shannon limit. For $p_0 = 0.9$, $p_0 = 0.95$, and $p_0 = 0.99$, all values of d_r in the figures satisfy the bound as well, which is $d_r \leq 13$, $d_r \leq 27$, and $d_r \leq 140$, respectively. Figures 3.3a and 3.3b also show the bit error rate of the RCM scheme with weights $\pm 2, \pm 2, \pm 3, \pm 3, \pm 4, \pm 8$ used in [31]. While this linear scheme presents good performance for high rates and $p_0 = 0.8$, in the case of $R_c = 1$ it is greatly outperformed by the proposed non-linear codes.

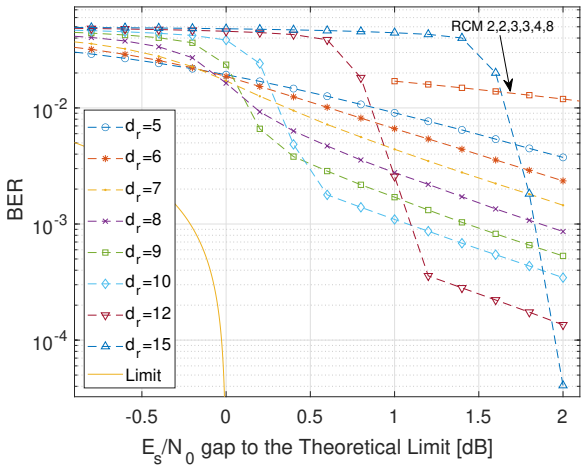
Similar to RCM and AFC, the results in Figure 3.3 present high error floors. This is not a problem, since the error floor can be easily reduced by concatenating the proposed codes with a high rate code, as in [21, 25, 31]. Once the proposed non-linear code lowers the BER below a certain threshold, the concatenated structure will be able to significantly reduce the number of remaining bits in error.

Figure 3.4 shows simulation results for $R_c = 2$ when $p_0 = 0.9$ and $p_0 = 0.99$. As previously mentioned, the information rate of the proposed joint source-channel coding scheme is more constrained than in the case of one-to-one mappings with 2^{d_r} points, typical of uniform sources, since now the constellation size of the proposed scheme is just $d_r + 1$. The drawback is that for high information rates, R, where $R = H(p_0)R_c$, the code might not provide good performance. Indeed, the results in Figure 3.4a, where R is higher, show waterfall degradation when compared to Figure 3.3b, while Figures 3.3d and 3.4b perform similarly. Still, the results in Figure 3.4 show that in spite of

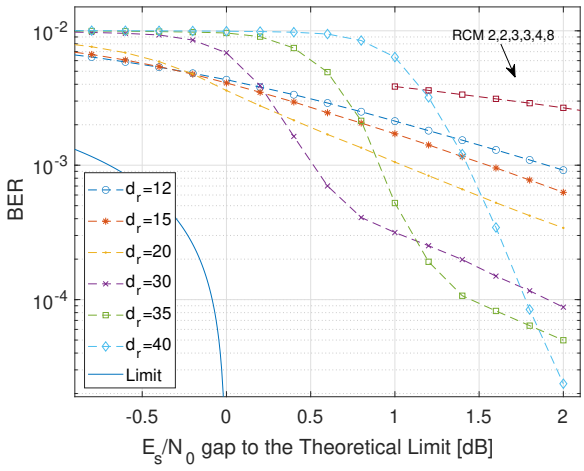


(a) $p_0 = 0.8$

(b) $p_0 = 0.9$

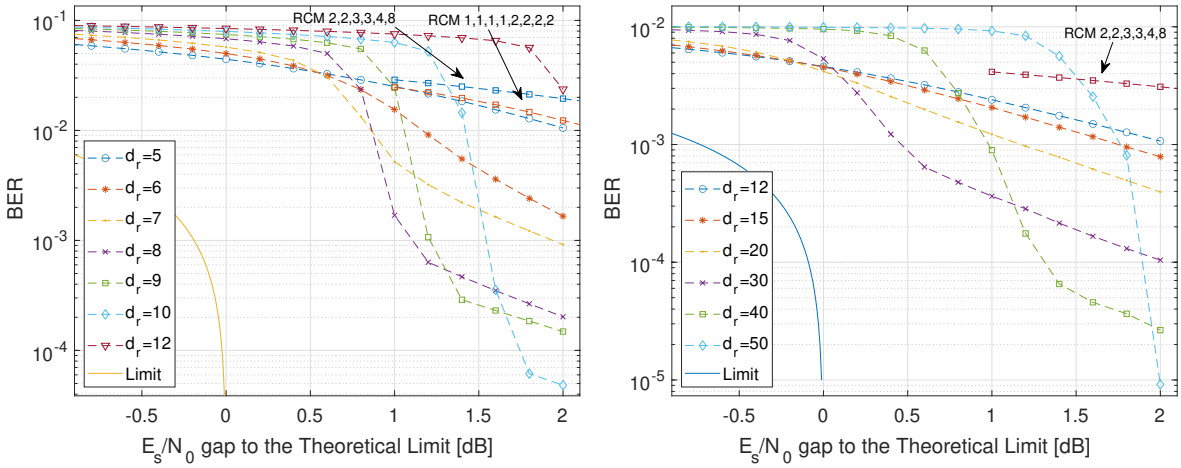


(c) $p_0 = 0.95$



(d) $p_0 = 0.99$

Figure 3.3: Simulation results for the proposed non-linear codes using one-dimensional mappings for joint source-channel coding of non-uniform sources when $R_c = 1$.



(a) $p_0 = 0.9$ (b) $p_0 = 0.99$

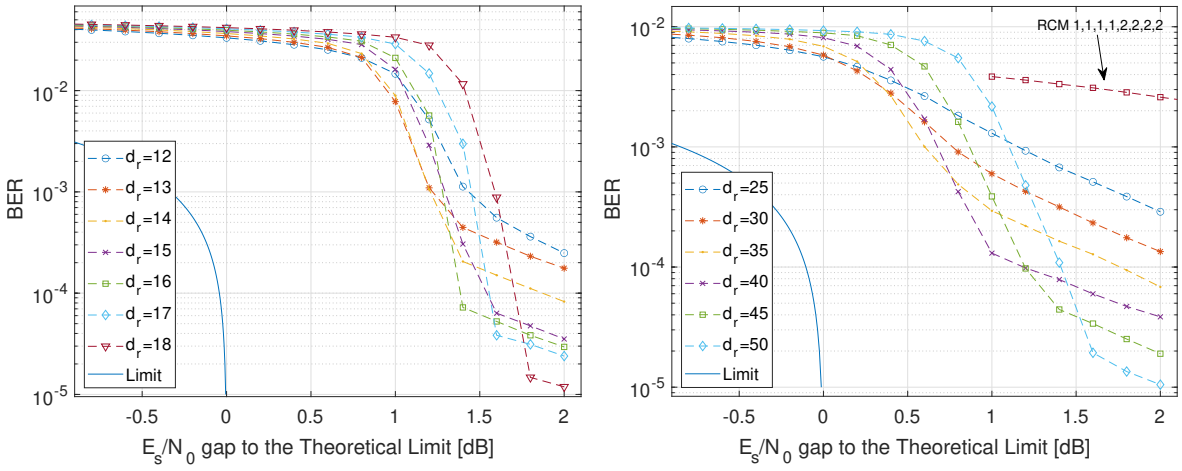
Figure 3.4: Simulation results for the proposed non-linear codes using one-dimensional mappings for joint source-channel coding of non-uniform sources when $R_c = 2$.

this constraint, the performance is good for $R_c > 1$ when $p_0 \geq 0.9$. Figure 3.4a also depicts the performance of two RCM codes with weight set $\pm 2, \pm 2, \pm 3, \pm 3, \pm 4, \pm 8$ [31] and $\pm 1, \pm 1, \pm 1, \pm 1, \pm 2, \pm 2, \pm 2, \pm 2$ [21]. It can be seen that the proposed scheme outperforms these RCM codes in both waterfall and error floor.

Next, we further increase R_c in order to compare the performance of the proposed scheme with the linear codes in [31] and [21], which have rates $R_c = 3.7$ for $p_0 = 0.95$ and $R_c = 5$ for $p_0 = 0.99$. We present the results in Figure 3.5 (different from other rates, for $R_c = 3.7$ we consider 37000 input bits and 10000 output symbols)[2]. As before, notice that for all values of d_r, waterfall occurs within 2 dB of the Shannon limit.

Finally, in order to reduce the error floor of the proposed schemes, we present results for their parallel concatenation with LDGM codes. Rather than transmitting all n symbols by using the proposed mapping, we transmit $n(1 - r_L)$ of them, while the remainder nr_L are transmitted using an LDGM mapping. Note that R_c is kept constant. Due to the concatenation and the fact that R_c is fixed, the waterfall of

[2] Note that the literature also presents results for high rate linear codes with $p_0 = 0.9$. However, the proposed one-dimensional method does not work well in these cases, which motivates the development of the two-dimensional mappings in the next section.

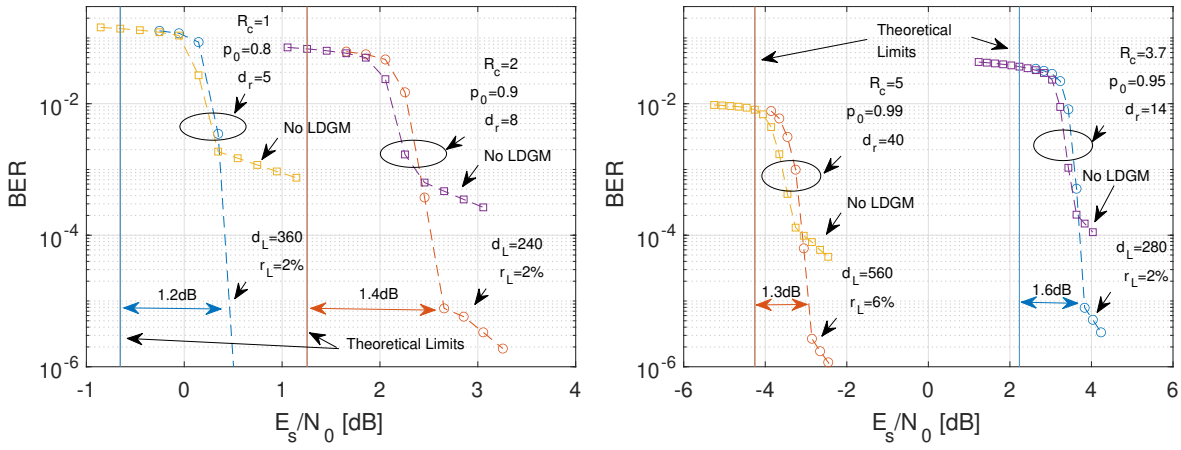


(a) $p_0 = 0.95$ - $R_c = 3.7$ (b) $p_0 = 0.99$ - $R_c = 5$

Figure 3.5: Simulation results for the proposed non-linear codes using one-dimensional mappings for joint source-channel coding of non-uniform sources when $R_c = 3.7$ and $R_c = 5$.

the concatenated scheme gets worse while the error floor improves (if r_L is small, the location of the waterfall region does not change much). Figure 3.6 shows the simulation results for the aforementioned concatenation. We simulate 600 blocks of 30000 input bits for all values of R_c except for $R_c = 3.7$, when we transmit 37000 input bits per block. We choose cases where the benefits of the proposed non-linear codes are more significant, which is for high values of R_c and high values of p_0 (Figure 3.6b), or for intermediate values of R_c and p_0 (Figure 3.6a).

Compared with linear codes, the proposed non-linear codes present remarkable performance when both non-uniformity and R_c are high. The optimized linear scheme proposed in [31] is 2.3 dB away from the theoretical limit for $p_0 = 0.95$ when $R_c = 3.7$, while for the proposed method the gap is 1.6 dB, as shown in Figure 3.6b. For the case of $p_0 = 0.99$ and $R_c = 5$, the gap from the theoretical limit in [21] is 2.2 dB, while the gap for the proposed scheme is 1.3 dB (see Figure 3.6b). Table 3.1 shows the performance comparison between the proposed non-linear codes and the best linear codes in the literature for cases with high non-uniformity.



(a) $p_0 = 0.8$ and $p_0 = 0.9$ (b) $p_0 = 0.95$ and $p_0 = 0.99$

Figure 3.6: Simulation results for the concatenation between the proposed non-linear codes using one-dimensional mappings and LDGM codes.

System	Gap to the theoretical limit	p_0	R_c
Proposed scheme	1.6 dB	0.95	3.7
[31]	2.3 dB	0.95	3.7
Proposed scheme	1.3 dB	0.99	5
[21]	2.2 dB	0.99	5

Table 3.1: Comparison between gaps to the theoretical limit achieved with the best linear codes in the literature and those of the proposed non-linear codes with one-dimensional constellations.

3.2 Two-Dimensional Scheme

While the proposed one-dimensional (1D) mappings work well for certain combinations of R_c and p_0, their performance is not very good when high information rates, $H(p_0)R_c$, are desired. As previously mentioned, the 1D mappings are rate-constrained because rather than using all possible 2^{d_r} constellation points, their constellations possess only $d_r + 1$ points. There are benefits from using such a compact constellation, such as a simple decoder and good performance when the information rate is not high. However, the results are not satisfactory when low values of p_0 are used with high values of R_c. With this in mind, we propose a scheme where the transformation f generates two-dimensional (2D) constellation points that are directly transmitted through the AWGN channel. Notice that for joint source-channel coding energy allocation is key, and the allocation can be done in a simple manner by using a two-dimensional constellation.

The rationale for the proposed energy allocation is the same as for one-dimensional mappings: to provide equal energy to all output symbols that correspond to input vectors that have the same probability, while keeping apart symbols whose input vectors are at a Hamming distance 1 of each other. Notice that in the 1D case the energy of a symbol mapped to the one-dimensional point c is equal to c^2, and thus there are only two possibilities where that point can lie ($-c$ or c). On the other hand, for the two-dimensional case, where energy is defined as the squared amplitude of the symbol, there are infinite possibilities on a circumference of radius c where that symbol can lie. Therefore, we position symbols corresponding to input vectors that have the same probability over the same circumference, and then adjust their angles to maximize the Euclidean distance between symbols that correspond to input vectors that differ in one bit. By doing this, we are able to achieve energy allocation and at the same time we also lower the error floor.

One of the main issues when using two-dimensional mappings is the computational complexity at the decoder. The general non-linear decoder has complexity that is exponential on d_r. Moreover, since we are exploring general two-dimensional

mappings, it is not possible to provide an exact decoder for a particular case, as we did for one-dimensional mappings. However, due to the non-uniformity of the source, it is possible to provide an approximate decoder that achieves good performance, as we will see in the sequel.

3.2.1 Encoder

As in the previous chapter, the encoding process is performed by replicating the transformation $f(\mathcal{M}_j)$ over and over again, until it generates the n output symbols from the k input bits. Since in the two-dimensional case f is a general transformation, we associate ordering numbers to the links between each input bit, m_i, and output symbol, $\mathbf{x_j}$, to indicate the position where each bit is used as an argument of f to generate $\mathbf{x_j}$.

Each output symbol, $\mathbf{x_j} \in \Gamma$, is generated by the mapping $f : \{0,1\}^{d_r} \rightarrow \Gamma \subset \mathbb{C}$, where Γ is the set with all possible constellation points. For simplicity, polar coordinates will be used. That is, a point $\mathbf{v} = (x, y) \in \mathbb{C}$ is represented in polar coordinates as the tuple (r, θ), such that $\mathbf{v} = re^{j\theta}$.The energy of $\mathbf{v}$ is equal to r^2, which can be written in terms of the squared Euclidean norm as $\|\mathbf{v}\|^2$. Hence, the expected energy per symbol, E_s, is expressed as $E_s = \mathbb{E}[\|\mathbf{x}\|^2]$. The idea of the proposed scheme is to use the amplitude to perform energy allocation, and then to adjust the angles so that the error floor is minimized. For simplicity, we split the function f into two functions, f_r and f_θ, which will, respectively, assign amplitudes and angles according to the vector of input bits connected to each output symbol. We will first define f_r as a function of the probability of each binary sequence, and then define f_θ to maximize, in a greedy manner, the minimum Euclidean distance between symbols whose input vectors differ in one bit. Formally, the j-th two-dimensional output node, $\mathbf{x_j} = f(\mathbf{b})$, with amplitude r_j and angle θ_r is expressed as

$$(r_j, \theta_j) = (f_r(\mathbf{b}), f_\theta(\mathbf{b})), \tag{3.11}$$

where f_r and f_θ are transformations that respectively assign the amplitude and angle to an input sequence $\mathbf{b}$.

3.2.2 Amplitude

Since the main idea of using a two-dimensional mapping is to perform energy allocation in an easy manner, we assign all equally likely input vectors to symbols that have the same amplitude. That is, let $\mathbf{b} \in \{0,1\}^{d_r}$. Then, the probability $P(\mathbf{b} = \mathbf{b}')$ is given by $(1-p_0)^{W(\mathbf{b}')}p_0^{d_r - W(\mathbf{b}')}$, where $W(\mathbf{b}')$ denotes the Hamming weight of $\mathbf{b}'$. Hence, all $\binom{d_r}{i}$ binary sequences with Hamming weight equal to i are equally likely and will be mapped to symbols with the same amplitude r_i. Notice that for $p_0 > 0.5$, $P(\mathbf{b} = \mathbf{b}')$ is monotonically decreasing over $W(\mathbf{b}')$, allowing us to perform proper energy allocation by defining f_r as

$$f_r(\mathbf{b}) := W(\mathbf{b}). \tag{3.12}$$

Hence, before normalization all output symbols lie on circumferences of radius $\{0, 1, ..., d_r\}$, according to the Hamming weight of their corresponding input vectors.

3.2.3 Angle

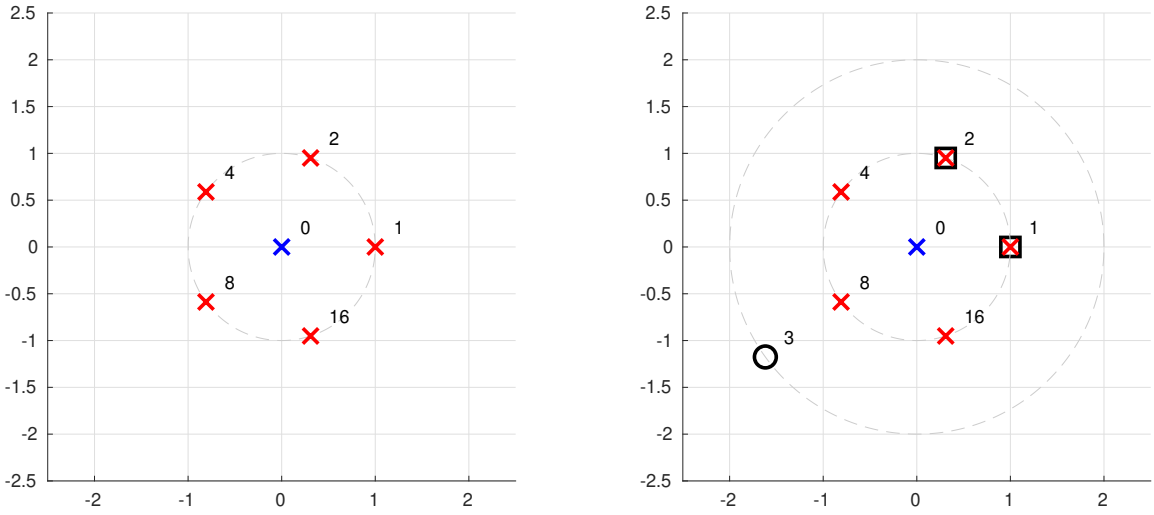
Assigning efficient angles to the constellation symbols is as important as providing proper energy allocation. As previously discussed, the error floor is mostly impacted by constellation symbols close in Euclidean distance whose associated input vectors are at a Hamming distance 1 from each other. In the previous section, we specified that all input vectors with the same probability are mapped to constellation symbols that lie on the same circumference. Note that since an input vector of Hamming weight W lies on a circumference of radius W, the symbols corresponding to binary sequences that differ in one bit are located on the circumferences of radius either $W-1$ or $W+1$. Importantly, even though the symbols associated to input vectors that differ in one bit are located in neighboring circles, it is possible to position them apart from each other to optimize the code performance. Since our design guidelines focus on energy

allocation and on separating symbols associated to input vectors at Hamming distance 1 from each other, constellation points can overlap.

We propose a greedy algorithm to design the mapping $f_\theta(\mathbf{b}) : \{0,1\}^{d_r} \rightarrow [0, 2\pi)$. The idea is to maximize the minimum Euclidean distance between symbols that correspond to input vectors that are at Hamming distance 1 from each other, while also satisfying the amplitude constraint provided by f_r. The greedy method sequentially finds the optimized angles for a given circumference assuming that the angles for the neighboring circumference of smaller radius is known. In this way, it is possible to find angles that satisfy the energy allocation requirements and result in constellation points separated as indicated before.

The algorithm starts by assigning angles to the input vectors of Hamming weight zero and one. The constellation points are assigned one at a time, following the decimal order of their corresponding input binary sequence. Since the all-zero sequence has amplitude equal to zero, that angle is not relevant. Next, all sequences with Hamming weight one are already at the maximum Euclidean distance from the point $(0, 0)$ because all of them lie on the circumference of radius one. Therefore, we position all d_r binary sequences with weight one uniformly separated over the circumference of radius one, as depicted in Figure 3.7a for $d_r = 5$. Once all sequences with Hamming weight zero and one are positioned, we start to assign the symbols to the circumferences of radius 2 up to d_r, in that order. The first constellation point to be positioned is the one generated by $\mathbf{b} = [0, ..., 0, 1, 1]$, which in decimal representation is equal to three. The algorithm searches for sequences in the neighboring inner circle that are at Hamming distance equal to one from the vector $\mathbf{b}$, which in this case are the constellation points that correspond to the decimals one and two, marked as squares in Figure 3.7b. By using the decimal representation of $\mathbf{b}$ as the argument of f_θ, we set $f_\theta(3)$ such that the minimum Euclidean distance between those points is maximized, which happens when $f_\theta(3) = \pi + (f_\theta(1) + f_\theta(2))/2$.

We repeat this process until all points of a circumference are assigned and we then repeat the same process for circumferences of higher radius. Notice that while the

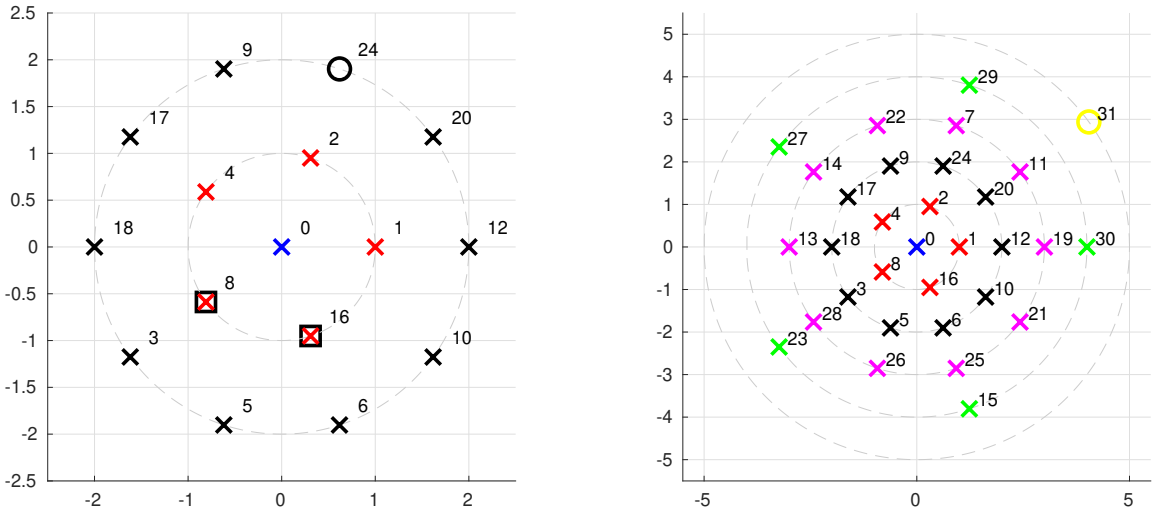


(a) Initial setup (b) Assignment of constellation point "3"

Figure 3.7: First iterations of the greedy algorithm that builds the proposed two-dimensional mappings for the case of $d_r = 5$.

process might be straightforward when there are only two symbols at Hamming distance 1 from the one to be assigned, the process is more complicated for circumferences of higher radius. In addition, there are situations where multiple angles maximize the minimum Euclidean distance. In these cases, we need a tie-breaking criteria to design the proposed mapping: given multiple choices, we position the constellation point at the location with the minimum number of points already assigned [3]. Figure 3.8a shows the constellation symbols resulting from the proposed algorithm for the circumferences of Hamming weight two and less when $d_r = 5$. Notice that no overlaps occurred, which is typical when d_r is small. In Figure 3.8b, we show the final two-dimensional constellation for $d_r = 5$. Since the algorithm provides a constellation with mean close, but not equal, to zero, we shift it in order to force a zero mean. Normalization is then performed to obtain the desired value of E_s.

[3] Note that some overlaps will naturally happen. However, with this criteria we will try to keep them at a minimum.



(a) Assignment of symbol "24" (b) Complete two-dimensional mapping

Figure 3.8: Intermediate and final iterations of the greedy algorithm that constructs the proposed two-dimensional mappings when $d_r = 5$.

3.2.4 Decoder

As expected, the general decoder for one-dimensional constellations also works for two-dimensional constellations with very few changes. Unfortunately, the degree of the symbol nodes, d_r, is now quite high, dramatically increasing the computational complexity of the decoder. Therefore, in this section we present the general decoder and an approximate decoding method, which is crucial for the practical implementation of the proposed non-linear codes with two dimensional constellations. The approximate method provides good performance and takes advantage of the non-uniformity of the source, which means that some messages will be much more likely than others. This will allow us to avoid the computation of constellation symbols that have low probability at a given iteration.

3.2.4.1 Initialization

All operations computed by the decoder are in the log-likelihood ratio (LLR) domain. As in the one-dimensional non-uniform case, we denote as $q_{ij}^{(t)}$ the outgoing message from input node m_i to symbol node x_j at the t-th iteration. The first outgoing

message $q_{ij}^{(1)}$ conveys the prior knowledge of m_i to the output symbol x_j and is expressed as

$$q_{ij}^{(1)} = \log \frac{P(m_i = 0)}{P(m_i = 1)} = \log \frac{p_0}{1 - p_0}. \tag{3.13}$$

3.2.4.2 Symbol Nodes

Similarly to the general one-dimensional decoder, the outgoing message from symbol node x_j to bit m_i, at the t-th iteration, is denoted by $s_{ij}^{(t)}$ and expressed as

$$s_{ij}^{(t)} = \log \frac{\sum_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w(e_{ij})}=0}} e^{\frac{-\|y_j - f(\mathbf{b})\|^2}{N_0} + <\mathbf{b}, \mathbf{q}_j^{(t-1)}>}}{\sum_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w(e_{ij})}=1}} e^{\frac{-\|\mathbf{y_j} - f(\mathbf{b})\|^2}{N_0} + <\mathbf{b}, \mathbf{q}_j^{(t-1)}> - q_{ij}^{(t-1)}}}, \tag{3.14}$$

where, as before, $< \mathbf{a}, \mathbf{b} >$ is the inner product between vectors $\mathbf{a}$ and $\mathbf{b}$, and $\mathbf{q}_j^{(t-1)} = [q_{ij|w(e_{ij})=1}\ , ..., q_{ij|w(e_{ij})=d_r}]$ is the message vector ordered according to the position of each input bit as an argument of f.

By using the notation $\mathbf{b}^i = [b_1, ..., b_{i-1}, b_{i+1}, ..., b_n]$ (i.e., removing the i-th coordinate of the n-dimensional vector $\mathbf{b}$), we have

$$s_{ij}^{(t)} = \log \frac{\sum_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w(e_{ij})}=0}} e^{\frac{-\|\mathbf{y_j} - f(\mathbf{b})\|^2}{N_0} + <\mathbf{b}^{w(e_{ij})}, \mathbf{q}_j^{w(e_{ij}),(t-1)}>}}{\sum_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w(e_{ij})}=1}} e^{\frac{-\|\mathbf{y_j} - f(\mathbf{b})\|^2}{N_0} + <\mathbf{b}^{w(e_{ij})}, \mathbf{q}_j^{w(e_{ij}),(t-1)}>}}, \tag{3.15}$$

where $\mathbf{q}_j^{w(e_{ij}),(t-1)}$ denotes the vector of ordered messages $\mathbf{q}_j^{(t-1)}$ without its $w(e_{ij})$-th coordinate.

Up to this point, all equations for $s_{ij}^{(t)}$ are exact. In order to develop a simplified decoder, we notice that the equation $\log\left(\sum_{k=1}^{K} e^{c_k}\right)$ can be approximated to $\max_k c_k$. Unfortunately, using the maximum is an oversimplified representation of the equation

and does not provide good performance in this case. However, instead of choosing the maximum, we can compute $\log\left(\sum_{k=1}^{K} e^{c_k}\right)$ by using the N largest c_k's. By choosing $N = 1$, the approximation is equivalent to computing the maximum, while for $N = K$ it is the equivalent of computing the exact equation. Hence, we can vary N as a tradeoff between performance and computational cost.

To apply the simplified decoder we need to find the N highest values of c_k in (3.15) in an efficient way, where c_k represents the k-th term in the sum of exponentials in (3.15), i.e.,

$$c_k = \frac{-\|\mathbf{y_j} - f(\mathbf{b})\|^2}{N_0} + < \mathbf{b}^{w(e_{ij})}, \mathbf{q}_j^{w(e_{ij}),(t-1)} >, \tag{3.16}$$

where $b_{w(e_{ij})} = 0$ when c_k is in the numerator and $b_{w(e_{ij})} = 1$ otherwise. Note that $\mathbf{b} \in \{0,1\}^{d_r}$ spans $2^{d_r} - 1$ terms due to the constraint in the sum. Hence, $k \in \{1, ..., 2^{d_r-1}\}$.

Since the left term of (3.16) is based on the general transformation f, the only way to find its largest values is to perform a search over all the constellation space, which is computationally expensive. However, since $p_0 > 0.5$, the vector $\mathbf{q}_j^{(t-1)}$ has most of its components greater than zero during the first iteration, making the right term of (3.16) considerably larger than the left one, since the latter is always less than or equal to zero. Therefore, it is reasonable to assume that the right terms of c_k will dominate. In this way, the approximated outgoing message can be written as

$$s_{ij}^{(t)} = \log \frac{\sum\limits_{\substack{\mathbf{b}\in\mathbf{B} \\ \text{s.t. } b_{w(e_{ij})}=0}} e^{\frac{-\|\mathbf{y_j}-f(\mathbf{b})\|^2}{N_0} + <\mathbf{b}^{w(e_{ij})}, \mathbf{q}_j^{w(e_{ij}),(t-1)}>}}{\sum\limits_{\substack{\mathbf{b}\in\mathbf{B} \\ \text{s.t. } b_{w(e_{ij})}=1}} e^{\frac{-\|\mathbf{y_j}-f(\mathbf{b})\|^2}{N_0} + <\mathbf{b}^{w(e_{ij})}, \mathbf{q}_j^{w(e_{ij}),(t-1)}>}}, \tag{3.17}$$

where $\mathbf{B}$ is the set of elements $\mathbf{b} \in \{0,1\}^{d_r}$ that generate the N largest inner products $< \mathbf{b}^{w(e_{ij})}, \mathbf{q}_j^{w(e_{ij}),(t-1)} >$.

The calculation of inner products between the incoming messages and the binary vectors has complexity $\mathcal{O}(2^{d_r})$ in the exact case. The simplified approach reduces the complexity to $\mathcal{O}(2^{d_r-d_c})$, where $d_c \in \mathbb{N}$, by computing the $2^{d_r-d_c-1}$ largest terms

$< \mathbf{b}^{w(e_{ij})}, \mathbf{q}_j^{w(e_{ij}),(t-1)} >$ in an approximate manner.

3.2.4.3 Input Nodes

The message from source bit m_i to symbol node x_j is computed as

$$q_{ij}^{(t)} = \sum_{v \text{ s.t. } e_{iv} \in E, v \neq j} s_{iv}^{(t-1)} + \log \frac{p_0}{1-p_0}. \tag{3.18}$$

3.2.4.4 Decision

The decision for each bit is calculated as

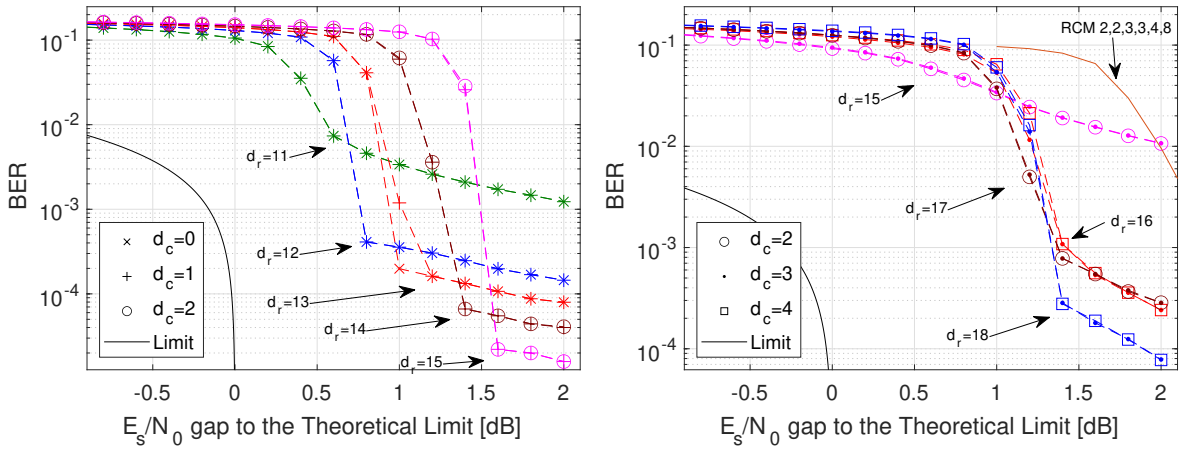
$$d_i^{(t)} = \sum_{v \text{ s.t. } e_{iv} \in E} s_{iv}^{(t-1)} + \log \frac{p_0}{1-p_0}. \tag{3.19}$$

The decision for bit m_i will be 0 if $d_i^{(t)} > 0$ and 1 otherwise.

3.2.5 Results

In this section we present simulation results showing the performance of the proposed non-linear codes using two-dimensional mappings for joint source-channel coding of non-uniform sources. We simulate 80 blocks of 30000 bits by transmitting $30000/R_c'$ two-dimensional constellation symbols over an AWGN channel with noise variance $N_0/2$ per dimension, where R_c' denotes the coding rate for the two-dimensional system. The information rate in terms of information bits per dimension is $H(p_0)R_d$, where R_d is the coding rate per dimension ($R_d = R_c$ for one dimensional systems and $R_c' = 2R_d$ for two dimensional systems). All mappings in this section are designed using the algorithm proposed in section 3.2.1, and decoding is performed applying the approximate method for different values of d_c. Note that when $d_c = 0$, the approximate decoder is equivalent to the exact decoding scheme.

First, we present results for $p_0 = 0.8$ with $R_c' = 4$ and $R_c' = 8$, as depicted in Figure 3.9. For $R_c' = 4$, it can be seen that, as d_r increases, the waterfall degrades and the error floor improves, similarly to the one-dimensional case. Different from the

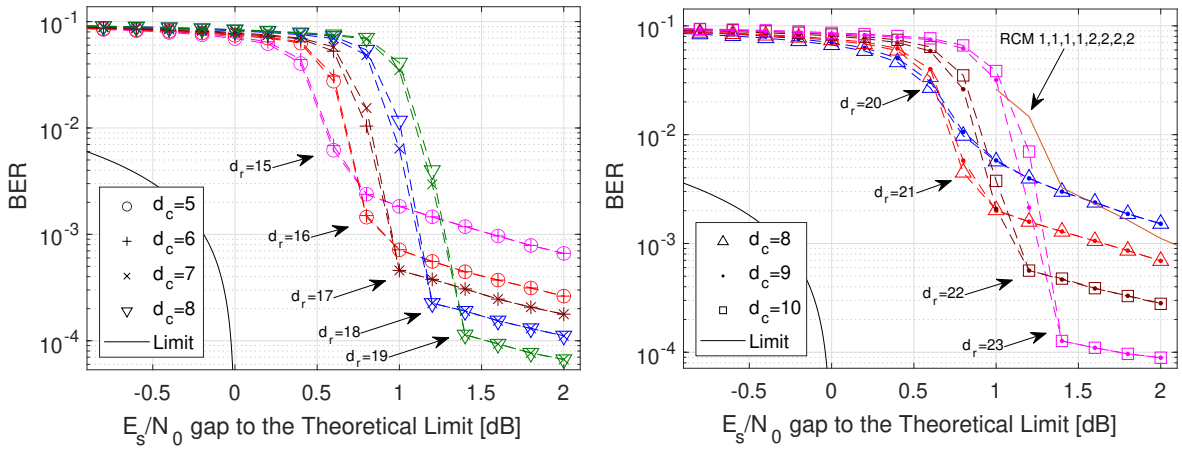


(a) $R'_c = 4$ (b) $R'_c = 8$

Figure 3.9: Simulation results for the proposed non-linear codes using two-dimensional mappings for joint source-channel coding of non-uniform sources when $p_0 = 0.8$.

one-dimensional mapping, the proposed two-dimensional scheme is able to transmit at high rates with low error floors when $p_0 = 0.8$. The figure shows that the approximated decoder works well. By comparing $d_c = 0$ (exact decoder) and $d_c = 1$, we can see that there is only a small degradation of 0.2 dB in the waterfall when $d_r = 13$, while the computational complexity is substantially reduced. Because of the high computational complexity of the exact decoder, we do not simulate it for $d_r > 13$. For $d_r = 14$ and $d_r = 15$, we use the approximate decoder with $d_c = 1$ and $d_c = 2$, and both cases result in identical performance, showing that the approximate decoder is a good solution to the computational complexity problem. Figure 3.9b focuses on the case of $R'_c = 8$ and compares the proposed scheme to the RCM code with weight set $\pm 2, \pm 2, \pm 3, \pm 3, \pm 4, \pm 8$ utilized in [31], which is clearly outperformed by the proposed scheme.

Figures 3.10a and 3.10b show results for $p_0 = 0.9$ with $R'_c = 4$ and $R'_c = 8$. The proposed two-dimensional scheme is able to transmit information at higher rates than the one-dimensional case. For $R_d = 2$, the two-dimensional system ($R'_c = 4$) presents a better waterfall (BER $< 10^{-3}$ at 1 dB from the theoretical limit) than the one-dimensional scheme. Notice that the degree of the output symbols, d_r, is higher than in typical RCM systems, especially for Figure 3.10b, causing a drastic increase



(a) $R'_c = 4$ (b) $R'_c = 8$

Figure 3.10: Simulation results for the proposed non-linear codes using two-dimensional mappings for joint source-channel coding of non-uniform sources when $p_0 = 0.9$.

in the number of operations performed by the exact decoder. Due to the increased computational complexity, Figure 3.10 only depicts results that use the approximate decoder, which is able to achieve good performance while dramatically reducing the number of operations performed at the receiver. Figure 3.10b includes the RCM code proposed in [21], which has a worse waterfall and higher error floor when compared to the proposed scheme.

Figure 3.11 shows results for $p_0 = 0.9$ when $R'_c = 10$ ($R_d = 5$), which is also the scenario in [21]. The approximate decoder is definitely necessary in this case, since all d_r's are greater than 22 and that leads to a highly computationally complex decoder. As before, when d_r increases the waterfall degrades and the error floor improves. The proposed codes clearly outperform the RCM code in [21], which has weights $\pm1, \pm1, \pm1, \pm1, \pm2, \pm2, \pm2, \pm2$.

Finally, with the purpose of reducing the error floor of the proposed scheme, we make a slight adjustment to the algorithm that optimizes the angle assigned to each input vector. In the unaltered version of the algorithm, each sequence $\mathbf{b}$ is encoded by $f_\theta(\mathbf{b})$ into the angle $\theta \in [0, 2\pi)$. As previously mentioned, the assignment performed by the algorithm maximizes, in a greedy manner, the minimum Euclidean distance

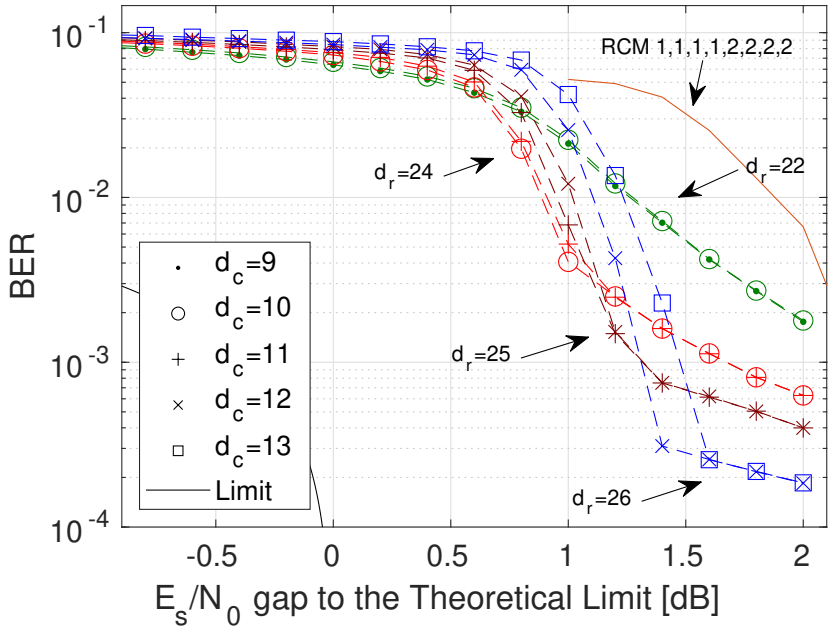


Figure 3.11: Simulation results for the proposed non-linear codes using two-dimensional mappings for joint source-channel coding of non-uniform sources when $p_0 = 0.9$ and $R'_c = 10$.

between points that are associated to input vectors that differ in one position. While this already provides good performance, the error floor might be improved if neighboring circles (points associated to input vectors at Hamming distance 1) are set even farther apart. For that purpose, we use the same algorithm as before, with the added constraint that the input vectors with even Hamming weight must have $\theta \in [0, \pi)$ and the ones with odd Hamming weight must have $\theta \in [\pi, 2\pi)$. The proposed assignment corresponds to mappings where input vectors with even Hamming weight are in the first and second quadrants while the ones with odd Hamming weight are in the third and forth quadrants, in a way that resembles the one-dimensional mappings proposed in the previous section.

Figure 3.12 shows the resulting constellation (rotated) for the case of $d_r = 11$. By constraining the first circumference to be only in two quadrants, the other assignments naturally obey the same constraint due to the maximization of the minimum Euclidean distance between constellation points associated to input vectors that differ in one bit. The same shape holds for other values of d_r. Notice that if the angles are

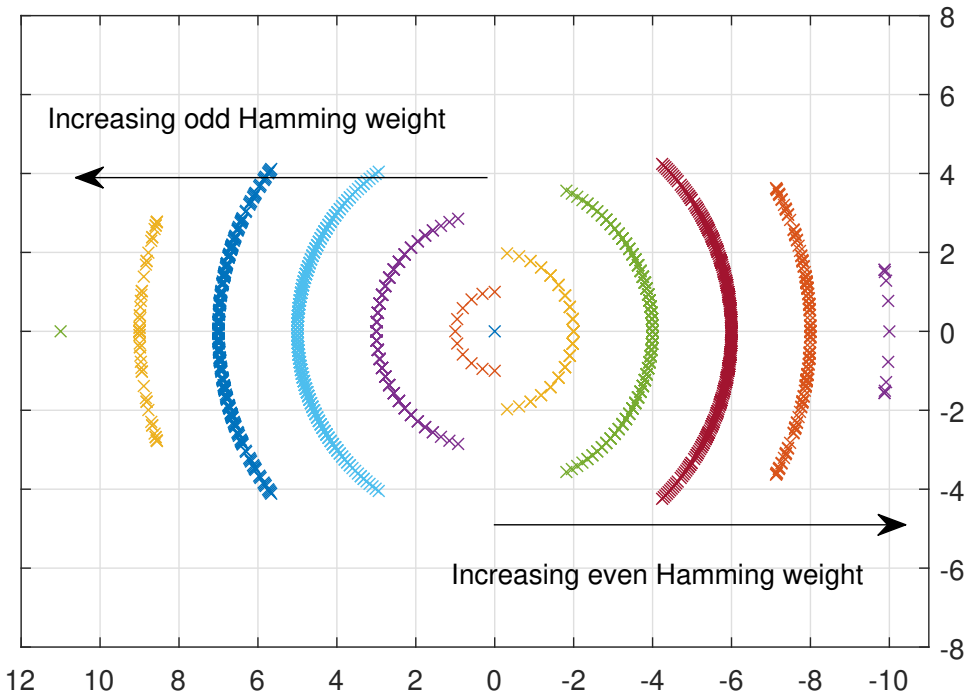
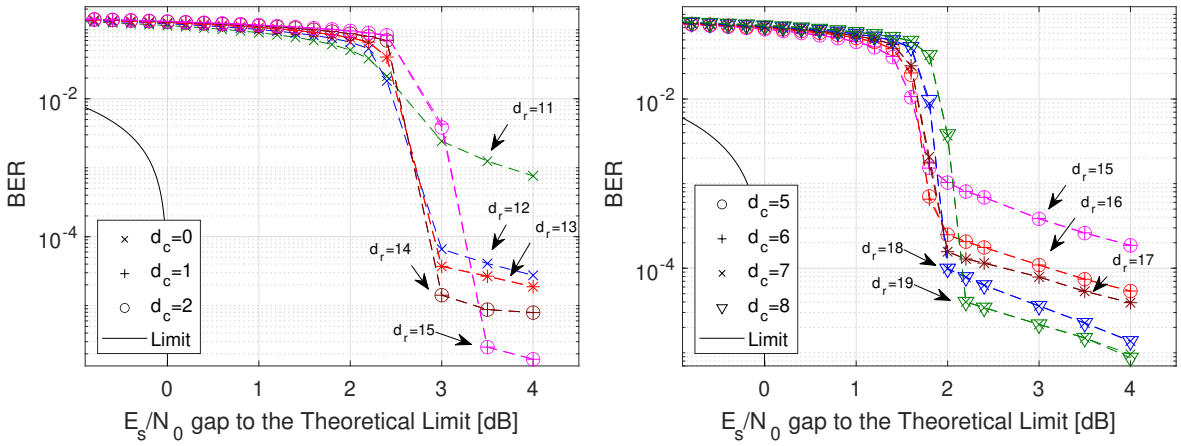


Figure 3.12: Constellation constructed by the proposed algorithm with the additional angle constraint for $d_r = 11$. Circumferences with points associated to input vectors with even Hamming weight are constrained to two quadrants, while the ones with odd Hamming weight are constrained to the other two quadrants. The shape of the constellation naturally arises as the algorithm progresses.

constrained to be either 0 or π, the proposed two-dimensional mapping will resemble the one-dimensional mappings from Section 3.1.

Figure 3.13 shows results for the proposed scheme with constrained angles for $p_0 = 0.8$ and $p_0 = 0.9$ when $R'_c = 4$. While the waterfall of Figures 3.13a and 3.13b is severally impacted by the added constraint, there is an improvement in the error floor when compared to their unconstrained counterparts from Figures 3.9a and 3.10a. The waterfall for all other values of R'_c were impacted even more severely.



(a) $p_0 = 0.8$ (b) $p_0 = 0.9$

Figure 3.13: Simulation results for the proposed non-linear codes using two-dimensional mappings for joint source-channel coding of non-uniform sources when $R_c' = 4$ and the angles are constrained according to the Hamming weight of each circumference.

3.3 Conclusion

In this chapter, we have first introduced one-dimensional non-linear mappings for joint source-channel coding of non-uniform memoryless sources. The proposed scheme is able to achieve excellent performance by properly allocating the energy resources. In addition to the general decoder, we introduce a simplified (but exact) decoder that is derived as a particular case of the general decoder. Different from RCM, where weights must be optimized for optimal performance, we provide a clear and simple rationale of how to find good mappings for joint source-channel coding. Finally, the limitations of one-dimensional mappings motivate the development of two-dimensional ones in order to further increase the rate at which information can be transmitted. In this context, we also introduced an approximate decoder, which performs quite close to the general one but is able to greatly reduce the computational complexity by taking advantage of the non-uniformity of the source. The proposed non-linear codes are able to outperform the best linear codes existing in the literature.

Chapter 4

NON-LINEAR CODING FOR NON-BINARY SOURCES

Modern channel codes, such as LDPC [6, 8], Turbo [7], and Polar codes [9] are typically designed for binary sources and implemented by operations in GF(2). Nevertheless, some members of the classical family of BCH codes [40,41] (Reed-Solomon codes [55] in particular) have relied on the design and encoding of non-binary symbols. In addition, the non-binary counterparts of modern binary channel codes have also been studied, which led to the development of non-binary variants of LDPC [56], Turbo [57, 58], and Polar codes [59, 60].

It can be argued that the motivation behind the study of non-binary codes is similar to that of non-linear codes. That is, since there are linear binary codes that approach the channel capacity as the block-length goes to infinity, and non-binary symbols can be encoded by binary alphabets, it is possible to approach the Shannon limit for non-binary sources. However, non-binary codes, as non-linear codes, may outperform linear binary codes in the case of finite block-length. In addition, our work is also motivated by the potential gains of non-binary codes for joint source-channel coding of Gaussian sources. It is clear that non-linear schemes and coded modulation provide great benefits to practical schemes that perform joint compression and correction. Therefore, by following the same rationale, we propose a joint source-channel coding scheme that transmits non-binary sources by using a non-linear mapping.

Specifically, in this chapter we propose the use of non-linear graph-based codes for the compression and correction of non-binary sources. We expand the previously presented non-linear binary codes to their non-binary counterpart by developing a general method of designing non-linear mappings over a non-binary alphabet with Q

levels. The proposed codes are used to the transmission of uniform non-binary sources and of quantized real-valued sources.

4.1 Encoder

Each non-binary input symbol, $m_i \in \{0, ..., Q-1\}$, belongs to a Q-ary alphabet and is an element of the k-dimensional non-binary input block $\mathbf{m}^k$, which is encoded by the general function f_c into n one-dimensional constellation symbols, denoted by $\mathbf{x}^n = [x_1, ..., x_n]$. In other words, $\mathbf{x}^n = f_c(\mathbf{m}^k)$. The general function f_c is implemented by a graph that connects the output symbols to the input non-binary symbols by successively replicating a lower-dimensional function, denoted by f.

The proposed scheme generates each symbol x_i by repeatedly applying the function f over random subsets of the k non-binary input symbols. This process is performed by a random-like graph that connects non-binary input symbols to output symbols. The encoding function $f : \{0, ..., Q-1\}^{d_r} \rightarrow \Gamma$ takes as input d_r non-binary symbols that are realizations of the input source and encodes them into an element of the countable set $\Gamma \subset \mathbb{R}$. As before, to enforce regularity in the graph, we constrain the difference in the number of edges of the most and least connected symbols to be less than or equal to one.

4.1.1 Encoding Function

In the general non-binary case, each output symbol is generated by the function f that maps the connected non-binary input symbols into real-valued symbols that belong to the set Γ. As in the binary case, we design f as a one-to-one function with Γ resembling a Gaussian distribution by using non-uniformly spaced points. Hence, Γ has Q^{d_r} elements that follow [51] (2.5), as presented in Chapter 2.

Once the constellation set Γ is fixed, the function f should be designed to properly assign the connected non-binary input symbols to elements of Γ. Note that as for the binary case, the edges connecting the output symbols x_i and the non-binary input symbols m_j have ordering numbers that are used to specify the argument ordering

of f. In other words, the edge between x_i and m_j, denoted by e_{ij}, has an ordering number $w(e_{ij}) \in \{1, ..., d_r\}$, which indicates that m_j is the $w(e_{ij})$-th argument of f from left to right. For simplicity, we also denote $w(e_{ij})$ as w_{ij}.

4.1.2 Group Intersect Method

We proceed by presenting the Group Intersect method, which is a generalization of the Parity Intersect method from section 2.4.1. In general, the function f can be seen as a mapping from an input vector with d_r components, where each component is an element from the set $\{0, ..., Q-1\}$. Since in the binary case the error floor can be controlled by the Euclidean distance between output symbols whose corresponding input vectors differ in one bit, we carry the same idea to the non-binary case. In the binary case ($Q = 2$), we separate the 2^{d_r} input vectors according to their parities (modulus 2 sum), which in turn generates two groups - one where all elements have parity equal to zero and another with parity equal to one. Then, as the groups are intersected, we are able to control the tradeoff between error floor and waterfall.

The rationale for the non-binary case is the same. We map, in a random fashion, input vectors that differ in one non-binary symbol into output symbols that are located farther away in terms of Euclidean distance from each other. For that purpose, we separate the Q^{d_r} input vectors into Q disjoint groups (sets), which are expressed by

$$\mathcal{G}_i = \left\{ \mathbf{c} = [c_0, ..., c_{d_r-1}] \in \{0, ..., Q-1\}^{d_r} \quad \middle| \sum_{k=0}^{d_r-1} c_k = i \ (\text{mod Q}) \right\}, i \in \{0, ..., Q-1\} \tag{4.1}$$

where $\mathcal{G}_i$ denotes the set of Q^{d_r-1} non-binary vectors whose modulus Q sum of its components is equal to i. In this way, we are able to build groups where any two non-binary vectors that are in the same group differ in two or more positions, i.e., their Hamming distance is greater than one.

Next, we group the sets $\mathcal{G}_i$ into $\mathcal{G} = [\mathcal{G}_0, ..., \mathcal{G}_{Q-1}]$, in that order, and perform a one-to-one correspondence between the vectors in $\mathcal{G}$ and the constellation symbols in

Γ. That is, by denoting as $\mathbf{c}_k^i$ the k-th non-binary vector of $\mathcal{G}_i$, we obtain $\mathcal{G}$ as

$$\mathcal{G} = [\mathbf{c}_0^0, ..., \mathbf{c}_{Q^{d_r-1}-1}^0, \mathbf{c}_0^1, ..., \mathbf{c}_{Q^{d_r-1}-1}^1, ..., \mathbf{c}_0^{Q-1}, ..., \mathbf{c}_{Q^{d_r-1}-1}^{Q-1}], \tag{4.2}$$

and then we assign each $\mathbf{c}_k^i$ to a constellation symbol in Γ as follows

$$\begin{aligned} \mathcal{G} = [&\mathbf{c}_0^0, \mathbf{c}_1^0, ..., \mathbf{c}_{Q^{d_r-1}-1}^0, \mathbf{c}_0^1, ..., \quad \mathbf{c}_{Q^{d_r-1}-1}^1, ..., \ \mathbf{c}_0^{Q-1}, ..., \quad \mathbf{c}_{Q^{d_r-1}-1}^{Q-1}] \\ &\downarrow \ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ \Gamma = [&\Gamma_0, \Gamma_1, ..., \Gamma_{Q^{d_r-1}-1}, \Gamma_{Q^{d_r-1}}, ..., \Gamma_{2Q^{d_r-1}-1}, ..., \Gamma_{(Q-1)Q^{d_r-1}}, ..., \Gamma_{Q^{d_r}-1}] \end{aligned} \tag{4.3, 4.4}$$

The random mapping defined by the aforementioned assignment is denoted as M_0. The proposed method successively intersects neighboring groups of $\mathcal{G}$, allowing us to systematically vary the Euclidean distance between output symbols associated to input vectors that differ in one position. Note that while Γ is fixed for all mappings, $\mathcal{G}$ changes. Therefore, we express the $\mathcal{G}$ associated to mapping M_i as $\mathcal{G}^i$, where $i \in \{0, ..., Q^{d_r-1} - 1\}$ and $\mathcal{G}^0$ denotes the initial mapping presented in (4.2).

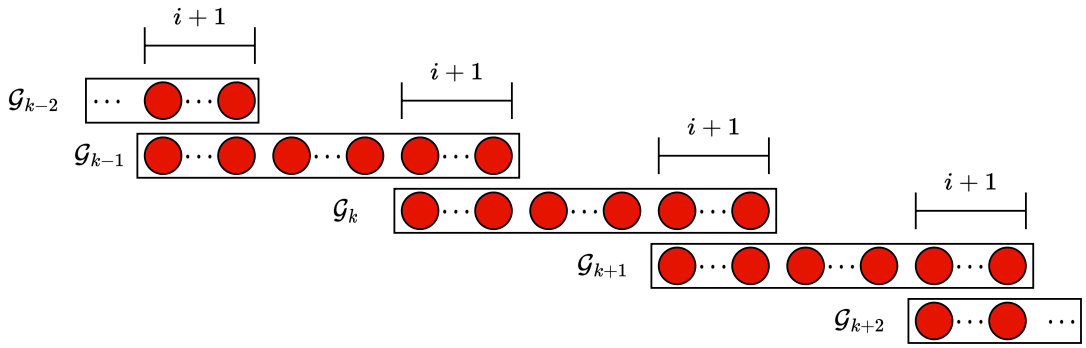


Figure 4.1: Construction of the auxiliary graphical structure for the Group Intersect algorithm.

The algorithm that constructs each mapping proceeds in two steps. In the first step it starts by creating an auxiliary graphical structure with circles positioned in a grid with rows and columns, where each circle represents a non-binary input vector.

Then, in the second step, it traverses the previously created structure so that the input vectors (circles) are mapped to their respective constellation symbols.

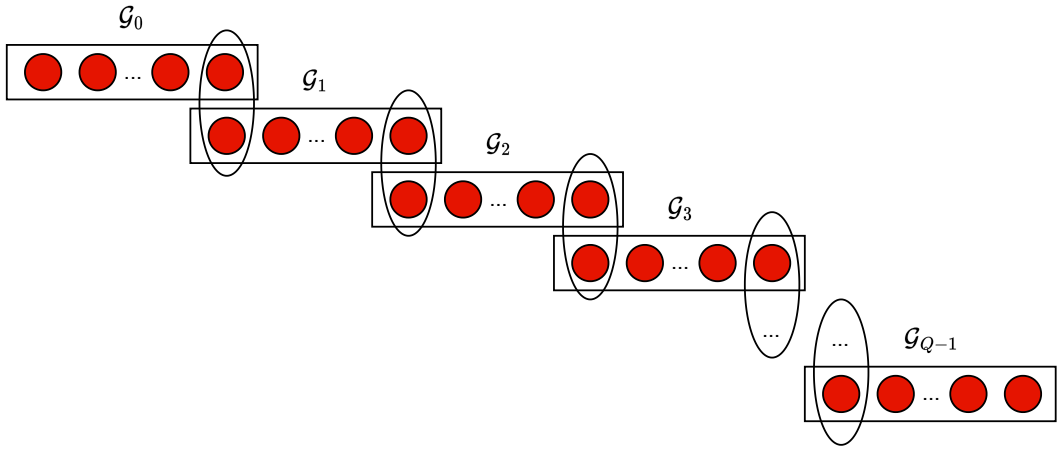
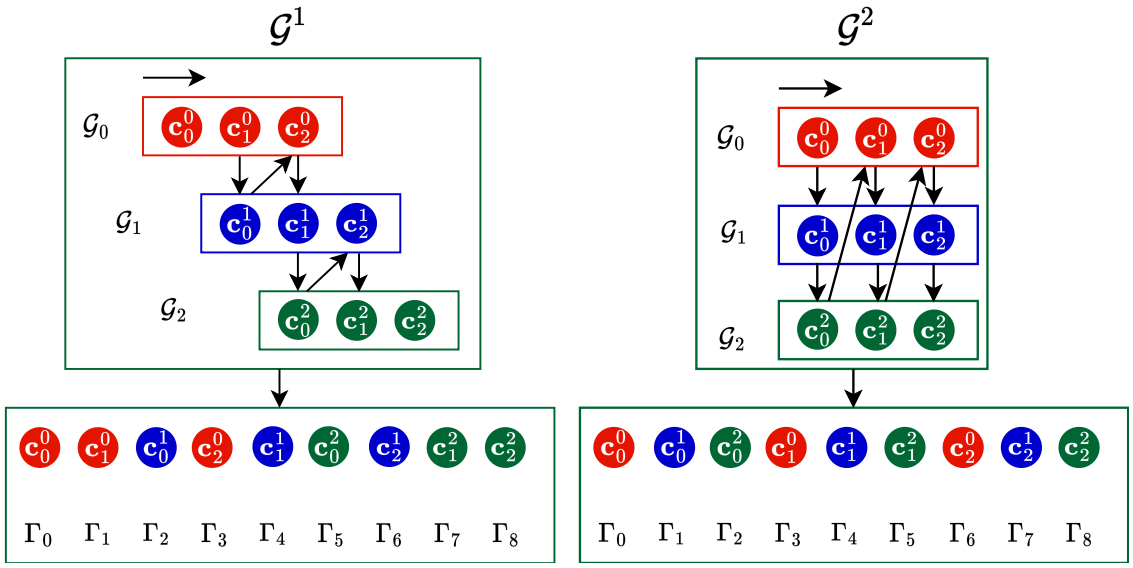


Figure 4.2: Construction of the graphical structure for the mapping M_0 by using the Group Intersect method for a Q-ary source. Each circle represents a non-binary vector.

As previously mentioned, the first step of the algorithm is to construct the auxiliary graphical structure. In order to build $\mathcal{G}^i$, and consequently, M_i, the graphical structure is organized so that each group $\mathcal{G}_k$ is positioned in the k-th row and the $i+1$ rightmost circles of each group share the same column as the $i+1$ leftmost circles of the next group, as depicted in Figure 4.1. Note that an overlap between more than two groups will occur as i increases. Figure 4.2 shows the particular case of $i = 0$, where each group shares only one column with the previous.

The second step of the algorithm is to traverse the auxiliary graphical structure to map each circle (non-binary input vector) to its correspondent element of Γ. For that, we introduce the iterator v and map the v-th visited circle to Γ_v. The leftmost circle of $\mathcal{G}_0$ is always set as the starting point ($v = 0$), and, therefore, mapped to Γ_0. The iterator v traverses the auxiliary graphical structure as follows: if there is a circle below the current one, $v + 1$ is set to that circle. Otherwise the iterator moves to the topmost circle of the column at the right of the current circle. The order at which v traverses the graphical structure defines the ordered set of input vectors, $\mathcal{G}^i$, which are



(a) M_1 (b) M_2

Figure 4.3: Group Intersect method when a non-binary uniform source with three levels ($Q = 3$) is considered for $d_r = 2$.

then mapped to the set of constellation symbols Γ ordered in an increasing manner, resulting in the mapping M_i. Note that the mapping M_0 shown in Figure 4.2 follows (4.4) and provides no intersection.

As an example, the process of building mappings M_1 and M_2 for $Q = 3$ with $d_r = 2$ is shown in Figure 4.3[1]. Note that as i increases, so does the intersection between groups, which leads to a decrease in the Euclidean distance between constellation symbols generated by input vectors that differ in one position.

Finally, to obtain the mappings M_i, $i \in \{0, ..., Q^{d_r-1} - 1\}$, the function f performs a one-to-one assignment of the non-binary input vectors from $\mathcal{G}^i$ to the constellation set Γ, following their respective ordering. Note that before the transmission of the output symbols through the AWGN channel, we normalize each output node to be zero mean with energy equal to E_s.

[1] Although $d_r = 2$ is not used for practical codes, here it illustrates how the mapping is generated.

4.2 Decoder

In addition to being used during encoding, the randomly built graph is also an important tool for the decoding process. In this section we describe the message passing algorithm [28] used to decode the received symbols and reconstruct the original non-binary source. The decoder is a direct generalization of the decoder typically used for non-binary LDPC codes [56] and RCM [15]. A similar version of this process can be found in [24], where it is applied to the transmission of correlated information sources using RCM.

The belief propagation decoder relies on passing messages through the edges of the encoding graph. Each message corresponds to local estimates that are propagated through the graph with the goal of approximately reconstructing the posterior probability mass function of the source. In the binary case, a message is a real number that is passed from node to node. In the non-binary case, instead of passing just a real value, we must pass probability mass functions with Q points (actually, $Q-1$ points, since the sum must be one). Since the proposed scheme can be used for non-binary non-uniform sources, we denote the *a priori* probability mass function of the input nodes m_i as $P(m_i)$.

4.2.1 Initialization

As in the binary case, the message passed from the input node m_i to the symbol node x_j is denoted by $q_{ij}^{(t)}$, whereas the message exchanged in the opposite direction is denoted by $s_{ij}^{(t)}$. Also, as previously mentioned, all messages are in $\mathbb{R}^Q$.

We initialize the algorithm by setting $q_{ij}^{(0)} = P(m_i)\ \forall i$ and passing $q_{ij}^{(0)}$ to the corresponding symbol nodes. In general, $P(m_i)$ is a probability mass function of the form $P(m_i) = [P(m_i = 0), P(m_i = 1), ..., P(m_i = Q-1)]$.

4.2.2 Symbol Nodes

The message exchanged from the j-th symbol node to the i-th input node is a generalization of the conditional density function for RCM decoders, which is expressed

as

$$s_{ij}^{(t)} \propto P(y_j|m_i), \tag{4.5}$$

where we use the proportional symbol ($\propto$) since $s_{ij}^{(t)}$ is normalized to have unit sum.

The previous expression is expanded in the same manner as for binary sources. We denote the subset of input nodes connected to the j-th symbol node, except for the i-th input node, as $\mathcal{M}_{j\backslash i}$, and then expand (4.5) as follows

$$\begin{aligned}
s_{ij}^{(t)}(X) &\propto P(y_j|m_i = X), \\
&= \sum_{\mathbf{b}\in\{0,\ldots,Q-1\}^{d_r-1}} P(y_j, \mathcal{M}_{j\backslash i} = \mathbf{b}|m_i = X) \\
&= \sum_{\substack{\mathbf{b}\in\{0,\ldots,Q-1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=X}} P(y_j = f(\mathbf{b}) + n_j, \mathcal{M}_{j\backslash i} = \mathbf{b}^{w_{ij}}|m_i = X) \\
&= \sum_{\substack{\mathbf{b}\in\{0,\ldots,Q-1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=X}} P(y_j = f(\mathbf{b}) + n_j|\mathcal{M}_{j\backslash i} = \mathbf{b}^{w_{ij}}, m_i = X) P(\mathcal{M}_{j\backslash i} = \mathbf{b}^{w_{ij}})
\end{aligned} \tag{4.6}$$

$$\begin{aligned}
&= \sum_{\substack{\mathbf{b}\in\{0,\ldots,Q-1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=X}} P(y_j = f(\mathbf{b}) + n_j|\mathcal{M}_j = \mathbf{b}) P(\mathcal{M}_{j\backslash i} = \mathbf{b}^{w_{ij}}) \\
&= \sum_{\substack{\mathbf{b}\in\{0,\ldots,Q-1\}^{d_r} \\ \text{s.t. } b_{w_{ij}}=X}} e^{\frac{-(y_j - f(\mathbf{b}))^2}{N_0}} P(\mathcal{M}_{j\backslash i} = \mathbf{b}^{w_{ij}}),
\end{aligned}$$

where $\mathbf{b}^i$ and b_i are used to remove and access the i-th coordinate of $\mathbf{b} = [b_1, b_2, ..., b_{d_r}]$, respectively.

The probability $P(\mathcal{M}_{j\backslash i})$ is calculated by multiplying the incoming messages at the j-th symbol node, as follows

$$P(\mathcal{M}_{j\backslash i} = \mathbf{b}^{w_{ij}}) = \prod_{v \text{ s.t. } e_{vj}\in E, v\neq i} q_{vj}(b_{w_{vj}}), \tag{4.7}$$

where E denotes the set of edges in the graph and e_{vj} denotes the edge between input

node m_v and symbol node x_j.

4.2.3 Input Nodes

The message exchanged from the i-th input node to the j-th symbol node is denoted as $q_{ij}^{(t)}$ and computed as

$$q_{ij}^{(t)}(X) \propto P(m_i = X) \prod_{v \text{ s.t. } e_{iv} \in E, v \neq j} s_{iv}(X), \tag{4.8}$$

where $q_{ij}^{(t)}(X)$ expresses the current estimate of $P(m_i = X)$ without taking s_{ij} into account. Once again, we use the proportional symbol ($\propto$) because the probability mass function, $q_{ij}^{(t)}$, must be normalized after its computation.

4.2.4 Decision

After a certain number of iterations, we estimate the posterior mass probability of each m_i, denoted as $P(m_i|\mathbf{y}^n)$, by calculating the point-wise product of the incoming messages at each input node as

$$\tilde{P}(m_i = X) \propto P(m_i = X) \prod_{v \text{ s.t. } e_{iv} \in E} s_{iv}(X), \tag{4.9}$$

where $\tilde{P}(m_i = X)$ is the estimate of the posterior mass function of m_i (i.e., $\tilde{P}(m_i = X) \approx P(m_i|\mathbf{y}^n)$).

Once we obtain an estimate of the posterior mass function, we must define a distortion metric that will dictate how to reconstruct the symbol for each input node. In this work, we apply the proposed scheme to two types of sources, each with its own distortion metric.

4.2.4.1 Decision for Non-Binary Uniform Sources

We first consider uniform non-binary sources (Q equally likely symbols), where the goal is to minimize the symbol error rate. In this case, the reconstructed source is

equal to the most likely symbol of $\tilde{P}(m_i)$, formally defined as

$$\tilde{m}_i = \arg\max_{X \in \{0,\ldots,Q-1\}} \tilde{P}(m_i = X). \tag{4.10}$$

4.2.4.2 Decision for Quantized Analog Sources

We also consider Gaussian real-valued sources with zero mean and unit variance. Since we are using a non-binary code, we first perform scalar quantization as proposed in [61], and then encode the quantized symbols as in the proposed scheme. The optimal regions and outputs for quantization presented by [61] will be used for encoding and decoding, respectively. We denote the Gaussian variable associated to the i-th input node as m'_i, leading to $m_i = \mathcal{Q}(m'_i)$, where $\mathcal{Q}(\cdot)$ denotes the process of scalar quantization.

In the encoding process we use the regions from [61] to transform the real-valued source into an index (or symbol) from a Q-ary alphabet. Then, we transmit the indexes through the AWGN channel. During the decoding process, we use optimal quantization output values from [61] to estimate the real-valued source. Since the goal is to minimize the mean square error of the original Gaussian source, and not the symbol error rate of m_i, the estimate of m_i, denoted by $\tilde{m}_i$, is expressed as

$$\tilde{m}_i = \sum_{X \in \{0,\ldots,Q-1\}} \tilde{P}(m_i = X)\mathbf{v}_X, \tag{4.11}$$

where $\mathbf{v}_X$ is the quantization output value of the corresponding region.

4.3 Results

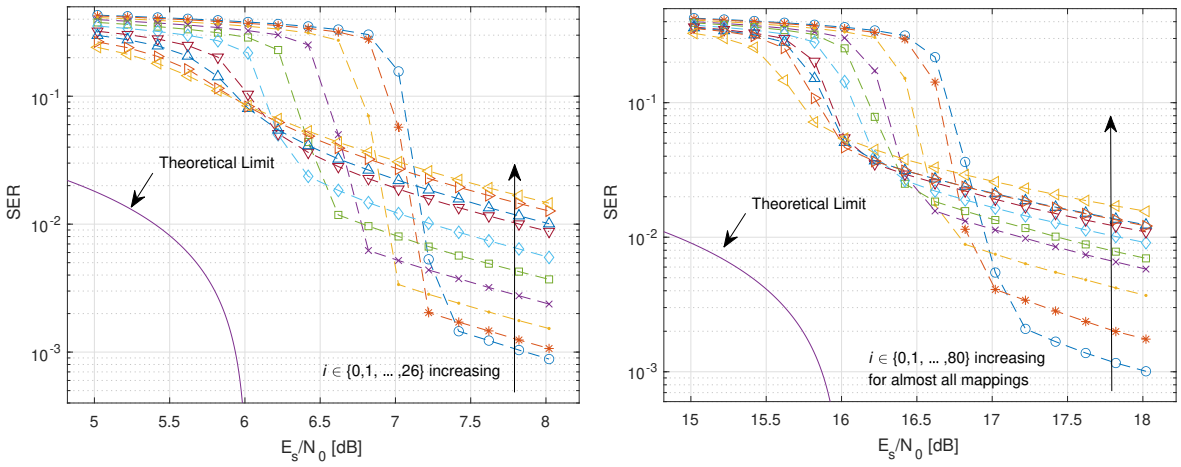
In this section, we present results of the proposed Group Intersect method for the transmission of non-binary sources through the AWGN channel. We transmit 80 blocks that are produced by encoding 30000 non-binary input symbols into $30000/R_c$ one-dimensional output symbols, where R_c is the coding rate of the encoding graph.

We apply the proposed scheme for two types of sources: 1) Non-binary uniform sources with Q equally likely symbols; 2) Gaussian sources with zero mean and unit variance which are optimally quantized (scalar quantization) into Q levels as in [61]. In general, the proposed method is able to accommodate both types of sources without any change. We maintain the same exact encoding graphs and mappings M_i for both sources and adjust the new normalization values (zero mean and energy equal to E_s). We also consider the non-uniform *a priori* density for the Gaussian source. In the non-binary uniform scenario (1), we focus on the symbol error rate (SER) as the distortion metric, whereas in the Gaussian case (2) we focus on the mean square error (MSE). We denote the average energy of the one-dimensional output symbols as E_s and consider an AWGN channel with variance equal to $N_0/2$, where N_0 is the spectral power density of the channel. As typical in analog codes, we express the signal-to-noise ratio (SNR) for the case (2) as the ratio between average symbol energy and the variance of the communication channel. Hence, SNR $= 2E_s/N_0$.

4.3.1 Non-Binary Uniform Sources

Figure 4.4 shows the results for the transmission of a non-binary uniform source with three levels ($Q = 3$) for $R_c = 1$ and $R_c = 2$. For both cases we consider the mappings M_i, $i \in \{0, ..., Q^{d_r-1} - 1\}$, and show how the performance varies depending on the mapping. The proposed method successfully generalizes the Parity Intersect method previously developed for binary sources. Our simulation results exhibit good performance at both low and high SNR, as well as similar characteristics as its binary counterpart: as i increases, the error floor gets higher while the waterfall improves.

When $i = 0$, the corresponding mapping provides a high Euclidean distance between constellation symbols whose input vectors differ in one position (Hamming distance 1). As i increases, the Euclidean distance decreases, improving the waterfall and worsening the error floor. While this rationale is not able to fully characterize the designed mappings, it is a simple and effective method to explore the tradeoff between waterfall and error floor. In Figure 4.4a, the SER at high SNR sequentially



(a) $R_c = 1$ and $d_r = 4$ (b) $R_c = 2$ and $d_r = 5$

Figure 4.4: Group Intersect method when a non-binary uniform source with three levels ($Q = 3$) is considered.

decreases as i increases, whereas in Figure 4.4b this happens for the vast majority of mappings with a few exceptions when SER $< 10^{-2}$. This indicates that there are other factors affecting performance, which will be investigated in future work. Nevertheless, it also demonstrates than input vectors at Hamming distance 1 have major importance when designing mappings, whether binary or non-binary. Notice that the envelope of the curves is able to reasonably follow that of the theoretical limit, providing good performance for regions close to the Shannon limit, and also beyond that limit (in the case of high SER).

Figure 4.5 presents simulation results for a ternary source ($Q = 3$) transmitted through a system with $R_c = 3$. As in Figure 4.4, we see an improvement in the waterfall and a degradation of the error floor as i increases. Note that the system rate is high ($R_c \log_2 Q = 3 \log_2 3 = 4.76$ bits per channel use) and the system shows good performance, specially for high SNR where it reaches values under 10^{-4} without the need of using concatenation.

Next, we present results for non-binary sources with $Q = 4$. In Figures 4.6a and 4.6b we show simulation results for $R_c = 1$ and $R_c = 2$, respectively. Both Figures are results of the simulation of the same mappings M_i for different values of R_c, showing

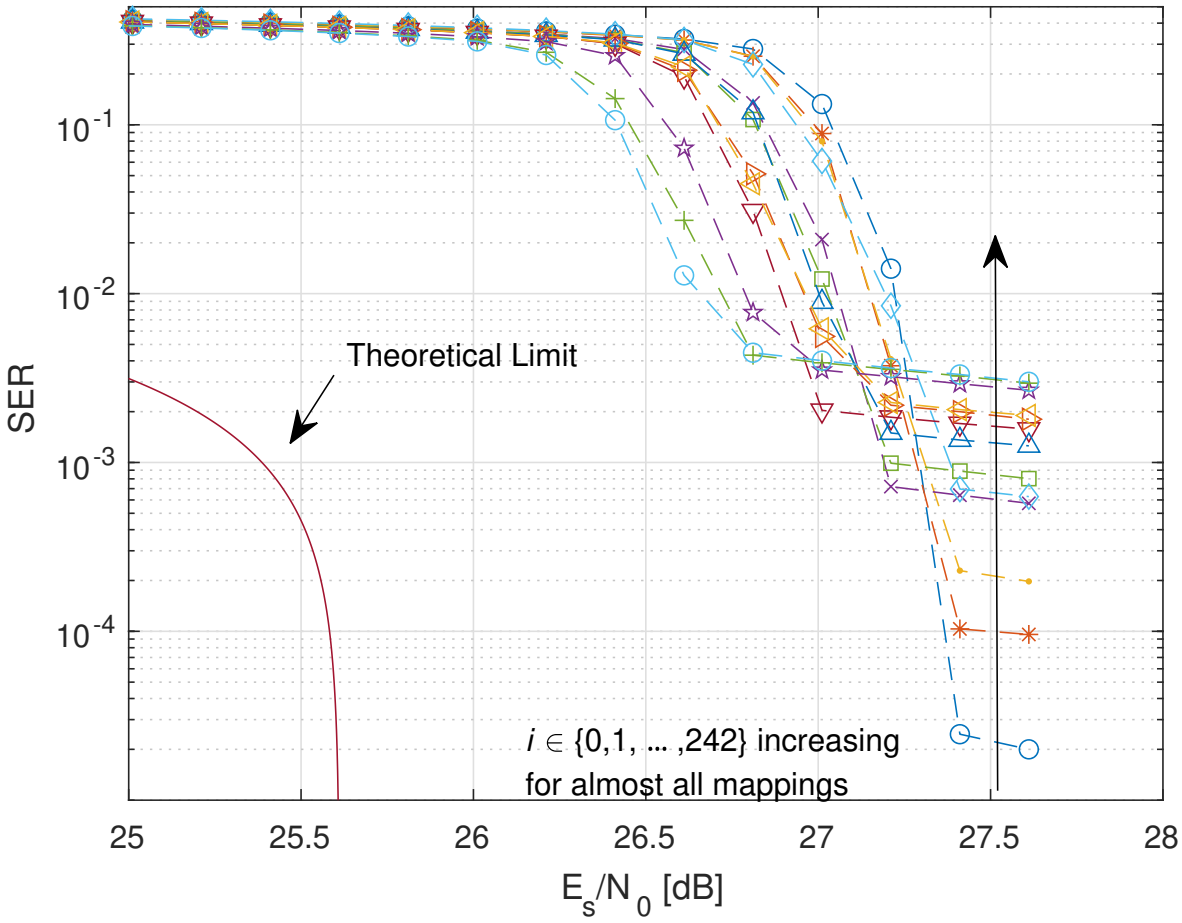
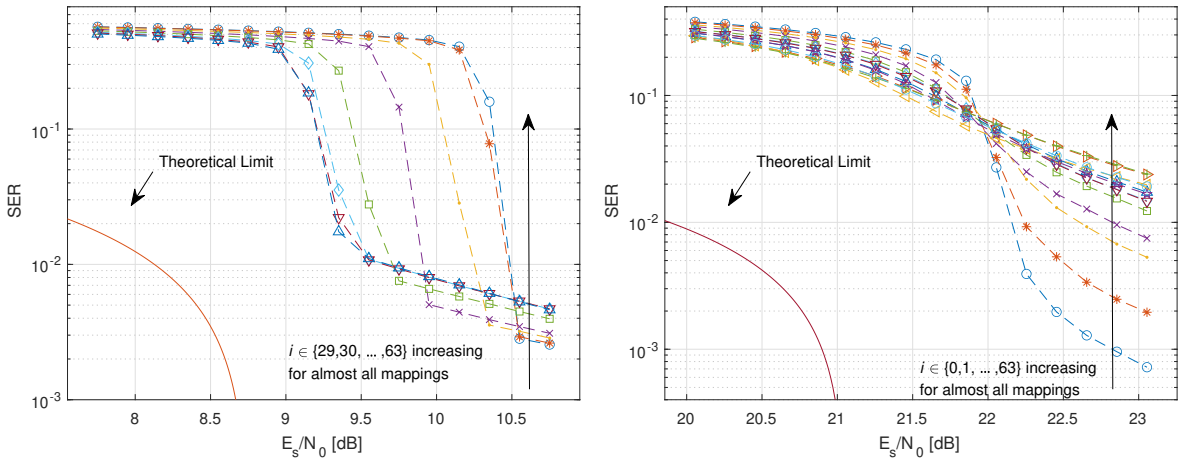


Figure 4.5: Group Intersect method when a non-binary uniform source with three levels ($Q = 3$) is considered for $R_c = 3$ and $d_r = 6$.

how the proposed scheme also presents relative flexibility in terms of R_c. In Figure 4.6a we present results for the mappings M_i only when $i \geq 29$, since convergence to the error floor region for $i \leq 28$ happens only at higher SNRs. The information rate of the system, in bits per channel use, is equal to 2 when $R_c = 1$ and 4 when $R_c = 2$.

Finally, in Figure 4.7 we present results for $Q = 5$ with $R_c = 2$ and $d_r = 4$, which corresponds to an information rate of $2 \log_2 5 = 4.64$ bits per channel use. Again, we see that the proposed method is able to achieve good performance for non-binary sources with different values of Q, obtaining different tradeoff between waterfall and error floor depending on the applied mapping. As i increases, the error floor gradually worsens, whereas the waterfall region approaches the theoretical limit.



(a) $R_c = 1$ and $d_r = 4$ (b) $R_c = 2$ and $d_r = 4$

Figure 4.6: Group Intersect method when a non-binary uniform source with four levels ($Q = 4$) is considered.

4.3.2 Analog Sources

We use the same non-binary codes developed using the Group Intersect method to transmit a Gaussian source. Each non-binary input node is a quantized Gaussian i.i.d. sample, quantized in an optimal scalar fashion as indicated in [61]. Note that by construction the non-binary quantized source has a non-uniform density function. Therefore, the output symbols must be re-normalized so that they have zero mean and average energy equal to E_s. In addition, as previously mentioned, the *a priori* density function is applied at the decoder so that we can achieve a better approximation of the posterior density function, $P(m_i|\mathbf{y}^n)$. Once the approximate posterior is inferred, we measure both symbol error rate (SER) of the non-binary symbols as well as the mean square error (MSE) of the Gaussian i.i.d. samples. In this way we can provide a better analysis of the code performance.

Since the considered system transmits a Gaussian source, the goal in this section is to minimize the MSE of the real-valued samples, instead of the SER of the quantized version. Note that it is suboptimal to use the non-binary value that minimizes the SER when the goal is to obtain the optimal MSE. While the SER is minimized by the most likely value of the posterior density function, the MSE is minimized by calculating its

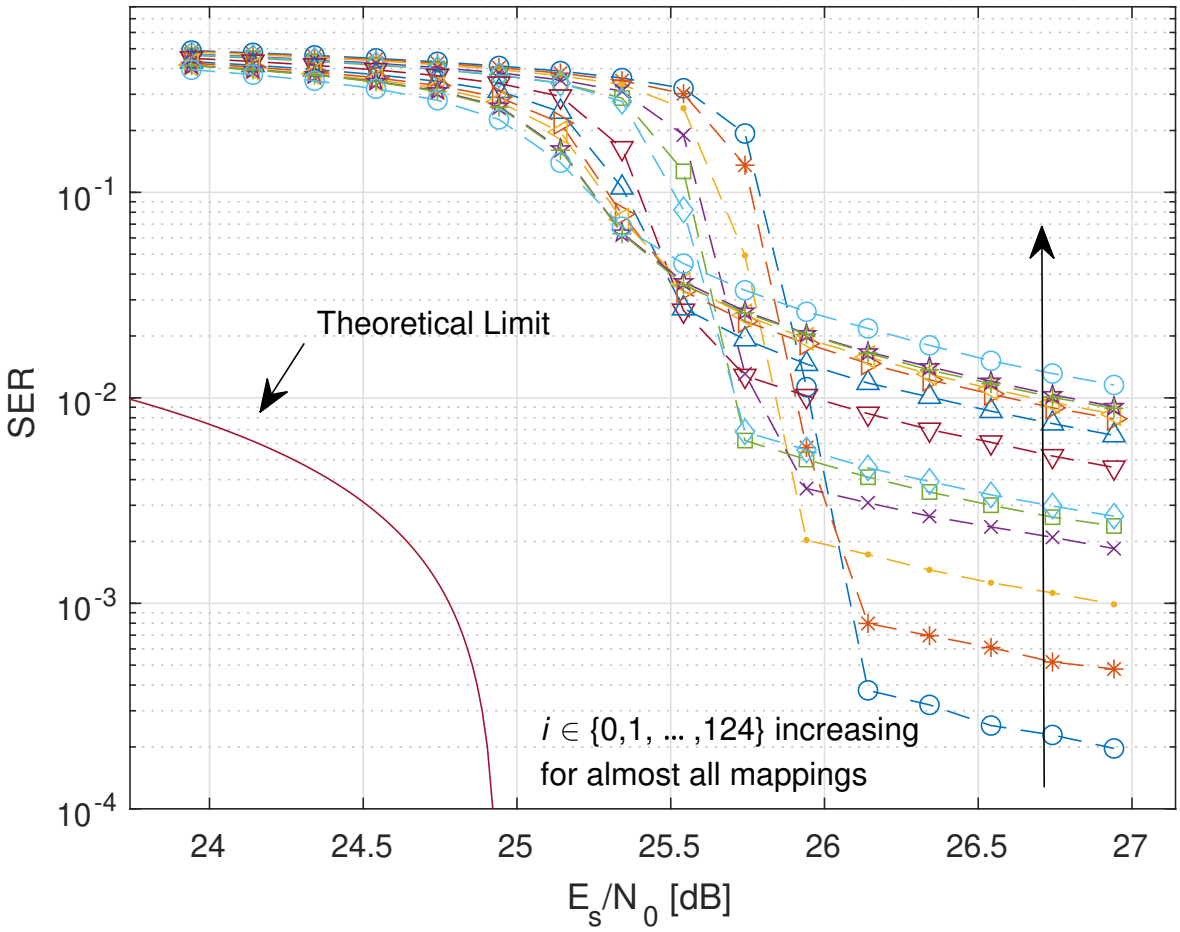


Figure 4.7: Group Intersect method when a non-binary uniform source with five levels ($Q = 5$) is considered for $R_c = 2$ and $d_r = 4$.

expectation.

Since $R_c = 1$ is a trivial case for the transmission of a Gaussian analog source over a Gaussian channel (uncoded scheme is optimal), we focus on $R_c = 2$ and $R_c = 3$. Figure 4.8 shows simulation results when the Gaussian source is quantized with $Q = 3$ levels and $R_c = 2$. We present the same set of simulations with different error metrics: symbol error rate (SER), which is relevant for the quantized system, and signal-to-distortion-ratio, expressed by $\text{SDR} = 10\log_{10}(1/\text{MSE})$, which is relevant for the end-to-end analog system. Notice in Figure 4.8a that a capacity-approaching digital system

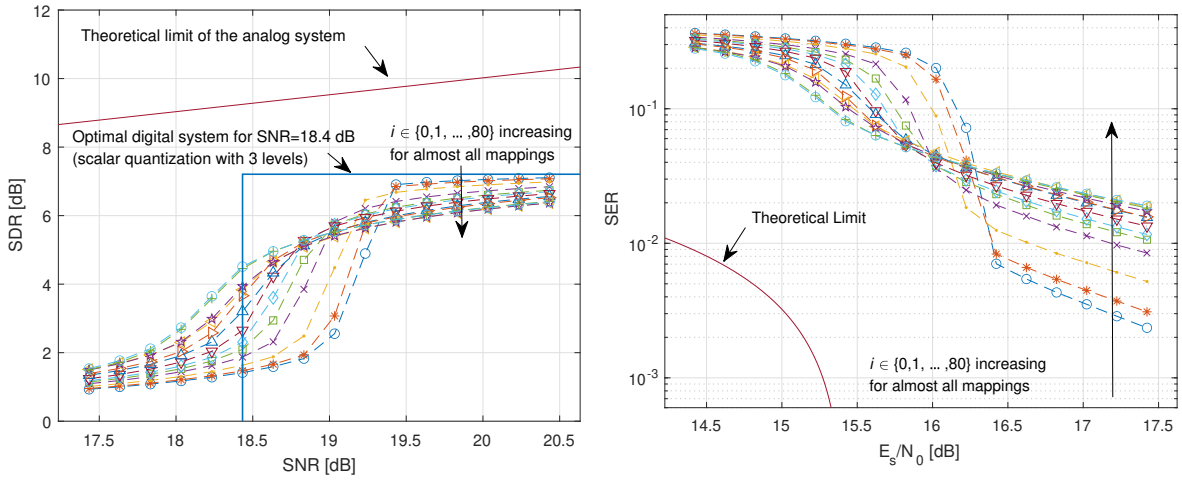
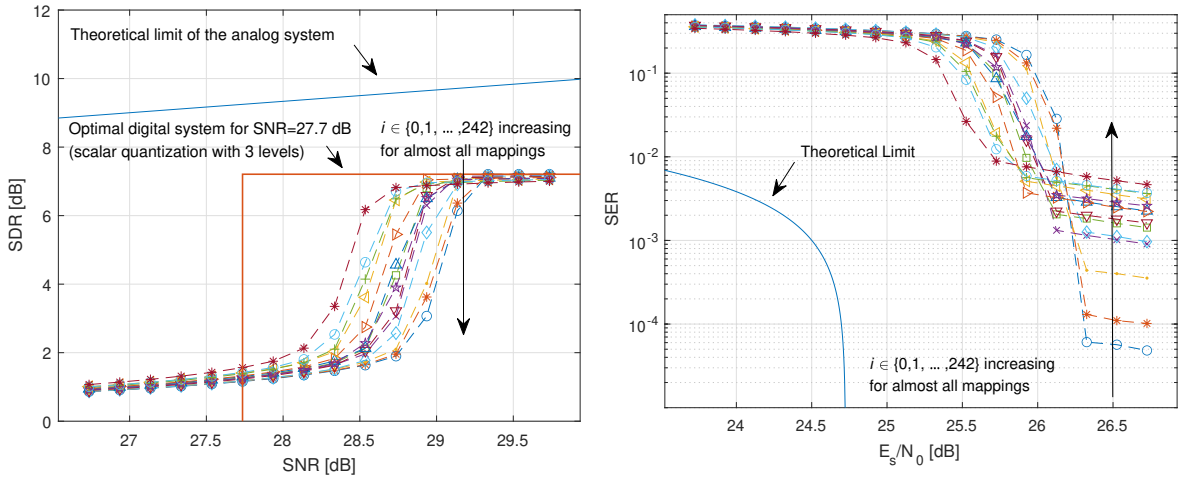


Figure 4.8: Transmission of a Gaussian source when optimal scalar quantization with three levels ($Q = 3$) is considered for $R_c = 2$ and $d_r = 5$.

does not achieve the theoretical limit of the analog system. The reason is that compressing (quantizing) a Gaussian source one sample at a time (scalar quantization) is inefficient. In order to achieve the rate-distortion function of a Gaussian source, several samples must be jointly quantized (vector quantization). However, we perform scalar quantization on the non-binary symbols and depict that limit for comparison purposes, leaving vector quantization for future work. Notice that by optimal digital system we mean one that is perfect at the Shannon limit for a given SNR and catastrophic below it. Traditional digital channel codes experience this catastrophic behavior beyond the Shannon limit. However, the proposed non-linear codes are capable of following the "real" theoretical limit, which explains the non-catastrophic behavior in Figure 4.8a.

Figure 4.8a shows that the mappings M_i approach the scalar quantization limit as the SNR increases. Indeed, mappings with lower i are able to steeply get closer to the limit than the ones with higher i. Figure 4.8b provides the rationale for this phenomenon, as it shows that mappings with low i have a steeper waterfall and lower error floor, as typical digital systems. In contrast, mappings with high values of i present good performance under the Shannon limit, which leads to the graceful degradation noticed in the previous paragraph.



(a) (b)

Figure 4.9: Transmission of a Gaussian source when optimal scalar quantization with three levels ($Q = 3$) is considered for $R_c = 3$ and $d_r = 6$.

Figure 4.9 presents results for $Q = 3$ with $R_c = 3$. Different from the previous case, as shown in Figure 4.9b the performance below the Shannon limit is not good, which explains why graceful degradation is not observed in Figure 4.9a.

Note that the proposed Group Intersect method is based on constellation symbols that are generated by input vectors that differ in one non-binary symbol, making it more suitable for SER minimization. In the case of analog coding, however, it does not suffice to take into account only symbols that differ in one position, but also how close they are. In other words, instead of focusing on perfect correction (low error floor), an efficient code for analog sources must force errors to be close to the original symbol, even if mistakes are made. Thus, in order to improve performance for the case of analog sources, other techniques need to be developed. We will consider them in the next chapter.

4.4 Conclusion

In this section we have presented a novel and general method for transmitting non-binary sources by using random-like graph-based channel codes. The resulting symbol error rate for the proposed schemes is around 10^{-3} at SNRs around 1-1.5 dBs

away from the Shannon limit, which is similar or better when compared with codes that operate in the binary domain. The proposed scheme provides a systematic and general manner to explore the tradeoff between waterfall and error floor, and allows the encoding of non-binary sources in a flexible manner (easily increasing/decreasing the overall system rate). We have also applied the proposed method to joint source-channel coding of Gaussian sources, providing reasonable performance for cases where bandwidth compression (increase in rate) is sought. As future work, it would be interesting to explore the use of vector quantization and the design of a mapping construction method that takes into account the different distortions caused by symbol errors, rather than just the cases where the input vectors differ in one component.

Chapter 5

NON-LINEAR CODING FOR ANALOG SOURCES

Conventional digital communication systems rely on symbols that belong to discrete alphabets, which allows the receiver to perfectly recover the transmitted sequence by using traditional error-correcting techniques. While the modular framework designed for digital communications has led to remarkable information transmission quality and is the foundation of modern systems, the added complexity and delay are unnecessary in particular cases, as described below.

One of the cases when digital systems are not necessary is when a source generates i.i.d. discrete-time Gaussian samples which are to be transmitted through an AWGN channel, one sample at a time. Instead of performing the traditional tasks of quantization, compression, correction, and modulation, Gaussian sources can be transmitted by using symbols with the same value as the samples generated by the source. In addition to drastically reducing the complexity of the system and achieving optimality, the straightforward analog system is also zero-delay since each source sample is decoded instantly at the receiver, which is not the case for digital systems that must receive many symbols to start the decoding process. Each channel symbol generated by the aforementioned analog system carries the information generated by a single source sample, leading into what is called a system with bandwidth 1:1, where the left-most integer denotes the number of source samples that are mapped to the number of channel symbols denoted by the right-most integer.

While the transmission of a Gaussian source through an AWGN channel in the 1:1 case is optimally solved, the same cannot be said for more general bandwidths. By considering general k:n systems, where k is the number of source samples that are

encoded into n real-valued output symbols, analog codes are classified by their ability to either expand ($k < n$) or reduce ($n < k$) the overall bandwidth.

In 1949, Shannon [62] discussed the idea of designing noise-resilient analog transformations by presenting an efficient way to map a line into a square (1:2 system), which introduced the idea of space-filling curves for analog coding. Later in 1976, [63] developed bounds for analog linear systems, proving that linear transformations have mediocre performance when used for bandwidth expansion and very bad performance when bandwidth reduction is sought, motivating the development of non-linear mappings. Several works in the literature rely on the use of space-filling curves (non-linear) to design short block-length analog systems, which are then optimized by brute-force or by algorithms such as Power Constrained Channel Optimized Vector Quantizers (PCCOVQ) [64,65]. While the schemes proposed in the literature have low delay and good performance, they must be redesigned if the desired bandwidth changes. Moreover, as k or n increase, searching for mappings becomes intractable due to the computational complexity of the optimization algorithms.

Motivated by the lack of flexibility of current analog schemes and the high complexity of their design, in this chapter we expand the previously proposed non-linear graph-based scheme for the transmission of analog sources. As before, the system design can be separated into encoder and decoder. Since it is not a simple task to find good mappings for analog coding (space-filling curves), we reevaluate the use of linear mappings to transmit Gaussian sources. While a linear scheme would greatly simplify the design, as mentioned before, it would also lead to a very significant loss in performance. Instead, the proposed scheme first performs a non-linear transformation to each source sample, which is then followed by a linear encoder. Thus, while part of the scheme is linear, the proposed code is non-linear when analyzed from end-to-end.

The rationale of the proposed mapping is that while pure linear codes perform poorly in analog coding, linear codes for binary sources (even in the real domain, e.g., Rate Compatible Modulation) are able to approach near-optimal performance. Thus, quantization followed by digital linear coding achieves good performance for the

transmission of analog sources. However, quantization, which is a particular example of non-linear transformation, introduces a (desired) distortion and limits the performance of the system if vector quantization is not used. In the context of analog coding, the idea in this chapter is to develop a procedure that resembling quantization introduces non-linearities. Hence, prior to transmitting the information through an analog linear system, the proposed scheme encodes each source sample by a function that introduces discontinuities to the source sample while keeping its information. Due to the discontinuities (non-linearities) that are introduced, the performance of the overall analog system is not anymore bounded by either the OPTA for linear schemes or by the quantization distortion.

In order to decode the received signal, we continue to use an iterative decoder that applies the belief propagation algorithm to the continuous random variables on the encoding graph. The sum-product algorithm for a linear analog system was initially presented in [66,67]. It can be derived by generalizing the previously presented decoder from the discrete to the continuous domain.

5.1 System Overview

In this chapter, the analog source produces normally distributed samples with unit variance that are sequentially grouped into blocks of k samples, generating the vector $\mathbf{m}^k$. Next, by applying the general function $f_c : \mathbb{R}^k \rightarrow \mathbb{R}^n$, the encoder transforms the k source samples into n real-valued output symbols, which are denoted by the vector $\mathbf{x}^n = [x_1, ..., x_n]$. The output symbols are normalized to have an average energy E_s and transmitted through an Additive White Gaussian Noise (AWGN) channel of zero mean and variance σ_r^2. Finally, the decoder processes the received noisy sequence and generates the estimate vector $\hat{\mathbf{m}}^k$. Different from the binary case, where the distortion is measured by the bit-error-rate (BER), since we now deal with continuous variables, the mean square error (MSE) is chosen as the distortion metric. We express the MSE in its logarithmic form, which is called signal-to-distortion rate (SDR) and

defined in dBs by

$$\mathrm{SDR}_{\mathrm{dB}} = 10\log_{10}\frac{1}{\mathrm{MSE}}, \tag{5.1}$$

where the mean square error (MSE) is expressed by $\mathrm{MSE} = \mathbb{E}[\|\hat{\mathbf{m}}^k - \mathbf{m}^k\|^2]/k$.

As for traditional digital communications, the signal-to-noise ratio, $\mathrm{SNR}_{\mathrm{dB}}$, is defined in dBs by

$$\mathrm{SNR}_{\mathrm{dB}} = 10\log_{10}\mathrm{SNR} = 10\log_{10}\frac{E_s}{\sigma_r^2}, \tag{5.2}$$

where σ_r^2 is the AWGN channel variance, $E_s = \mathbb{E}[\|\mathbf{x}^n\|^2]/n$ is the average symbol energy, and $\mathrm{SNR} = E_s/\sigma_r^2$ denotes the signal-to-noise ratio.

The motivation for the use of non-linear schemes for the transmission of analog sources is based on the fact that analog linear codes perform badly when compared to the OPTA from (1.12). In [63], the theoretical limit for analog coding using a linear transformation is obtained as

$$\mathrm{SDR}_{\mathrm{dB}} < \begin{cases} 10\log_{10}\left(\frac{\mathrm{SNR}}{R_c}+1\right) & \text{if } R_c \leq 1 \\ 10\log_{10}\left(\frac{1}{1-\frac{\mathrm{SNR}}{R_c(\mathrm{SNR}+1)}}\right) & \text{if } R_c > 1 \end{cases} \tag{5.3}$$

Note that in the case of bandwidth compression ($R_c > 1$), the theoretical limit for a linear scheme is upper bounded by $10\log_{10}\frac{R_c}{R_c-1}$ independently of how high the SNR is. This motivates our interest in finding alternative schemes for bandwidth compression. In Figure 5.1 we present the OPTA for both linear and non-linear schemes, respectively, from (5.3) and (1.12). It is clear that when $R_c \neq 1$ any linear scheme is far from reaching the theoretical limits, specially when bandwidth compression is desired ($R_c > 1$).

5.2 Encoder

As previously mentioned, different from binary codes, linear analog schemes are far from reaching the optimal performance theoretically attainable (OPTA). However, linear systems have low encoding complexity and simpler designs than general non-linear systems. With this in mind, we propose the use of a transformation, S, which

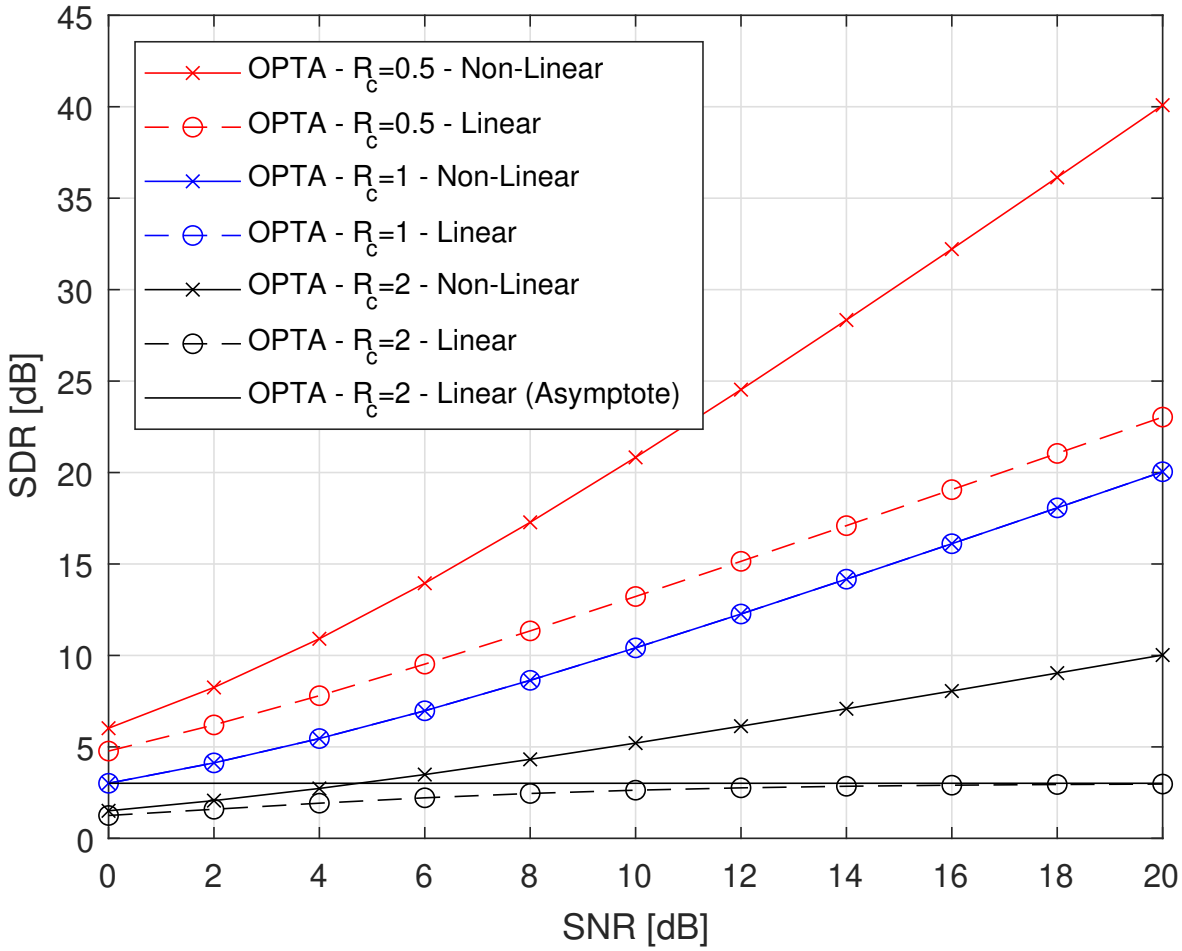


Figure 5.1: Optimal Performance Theoretically Attainable (OPTA) for linear and non-linear k:n analog schemes with coding rate $R_c = k/n$. Linear schemes are considerably worse than non-linear ones, and their performance for bandwidth compression ($R_c > 1$) barely improves as the signal-to-noise ratio increases.

will be used before the linear encoder, substituting the quantization step that would be performed in Rate Compatible Modulation. Figure 5.2 depicts an analog linear system which is encoded by a matrix $\mathbf{E}$ in the real domain, whose performance is bounded by (5.3). In traditional coding the analog source is quantized into 2 or more levels, which are then encoded by $\mathbf{E}$.

Figure 5.3 introduces the proposed scheme, where the non-linear function S is inserted between the analog source and the analog linear encoder to modify each source sample m_i. The function $S : \mathbb{R} \to \mathbb{R}$ is designed to introduce discontinuities without

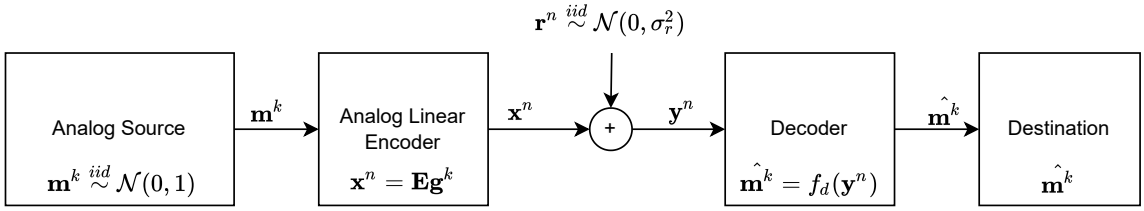


Figure 5.2: Analog linear system

performing quantization. After each sample m_i is transformed into $g_i = S(m_i)$, we encode the k dimensional vector, $\mathbf{g}^k$, by using a linear system with matrix $\mathbf{E}$. The encoding matrix $\mathbf{E}$ is sparse and represented as a graph so that belief propagation decoding can be performed [28, 66, 67].

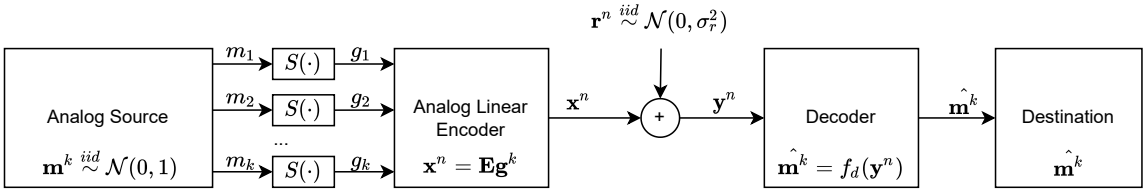


Figure 5.3: Proposed analog system that uses the non-linear function S to precode each sample produced by the analog source.

Figure 5.4 depicts an example of the proposed system. The example shown performs bandwidth compression with $R_c = 6/4 = 1.5$. Each output symbol, $x_j \in \mathbb{R}$, is generated by repeatedly applying the same linear combination over a random subset of the nodes that represent the modified source samples, $g_i = S(m_i)$. Formally, an output symbol x_j is expressed as

$$x_j =< \mathbf{w}, \mathbf{g}_j >= \sum_{k=1}^{d_r} w_k g_{i|w(e_{ij})=w_k}, \tag{5.4}$$

where d_r is the number of connections of each output symbol x_j, $w(e_{ij})$ is the weight of the edge that connects g_i and x_j, and $\mathbf{g}_j = [g_{i|w(e_{ij})=w_1}, ..., g_{i|w(e_{ij})=w_{d_r}}]$ is the set of nodes

connected to the output symbol x_j, which are ordered according to the weight vector $\mathbf{w} = [w_1, ..., w_{d_r}]$. Note that each node g_i is generated by S from its corresponding source sample as $g_i = S(m_i)$.

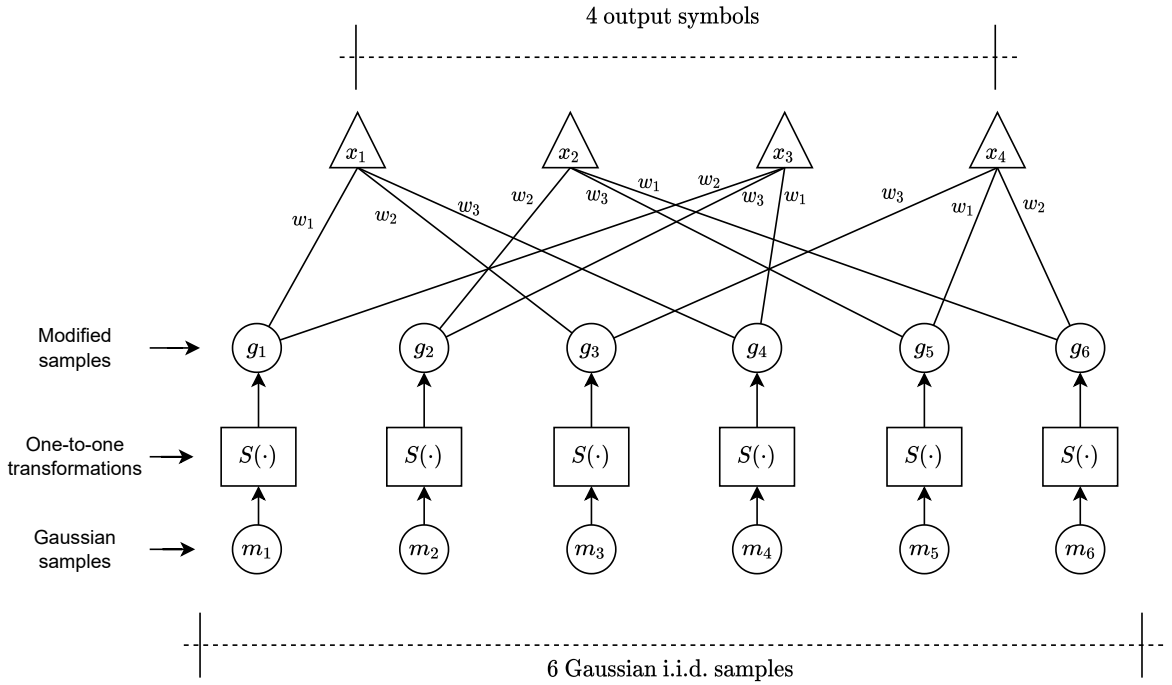


Figure 5.4: Proposed scheme that encodes 6 Gaussian samples into 4 output symbols. Each symbol is generated by the linear combination of a subset of the source samples modified by the function S.

5.2.1 Proposed Non-Linear Function S

The design of the function S is key in order to obtain good performance. The rationale is to design S by creating discontinuities (non-linearities) at each source sample m_i. These discontinuities will be reflected on the output symbols, which is necessary because otherwise the scheme would be linear, and thus its performance would be poor. Notice that space-filling curves and digital systems introduce discontinuities when encoding analog sources, which allows the encoded symbols to be more resilient to noise. By discontinuities, we mean that two points that are close on the source space $\mathbb{R}^k$ are mapped to output symbols that are not always close on the channel space $\mathbb{R}^n$.

We design S by introducing regions of zero support (gaps) in the probability density function of the Gaussian samples generated by the source. First, we define a partition of $\mathbb{R}$ in Q segments, denoted by the set $\mathbf{s}_i = (s_i, s_{i+1}) \subset \mathbb{R}$ where $i \in \{0, ..., Q-1\}$. Then, instead of assigning a single value for all $m \in \mathbf{s}_i$ as in scalar quantization [61], we introduce a gap of zero support between two consecutive regions $\mathbf{s}_i$ and $\mathbf{s}_{i+1}$ by spreading both regions apart from each other[1].

Figure 5.5 depicts the aforementioned process for $Q = 2$. The Gaussian distribution of the source is split into two regions, $\mathbf{s}_0 = (-\infty, 0)$ and $\mathbf{s}_1 = (0, \infty)$, which are separated by a segment of length l_0. The same process can be carried out for higher values of Q, as in Figures 5.6 and 5.7, which depict the function S for $Q = 3$ and $Q = 4$, respectively. As Q increases, so does the complexity of S, since more parameters must be adjusted. Different from the linear case, gaps between regions generate discontinuities in the linear combinations transmitted through the channel, which help the decoder distinguish the encoded samples.

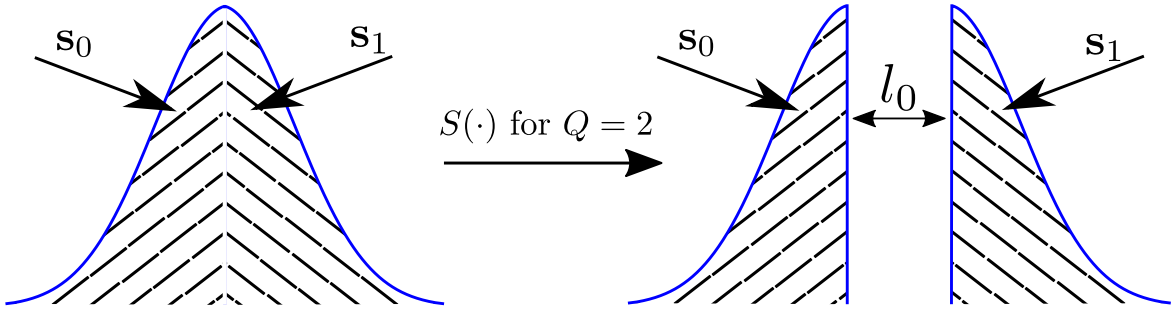


Figure 5.5: Proposed non-linear function S for the case of $Q = 2$ before normalization.

Since the last point of the region $\mathbf{s}_i$ and the first point of the region $\mathbf{s}_{i+1}$ coincide (disjoint sets whose union fill $\mathbb{R}$), we can define the vector $\mathbf{s} = [s_1, ..., s_{Q-1}]$, where $s_i \in \mathbb{R}$ are the points that delimit consecutive regions. Note that we omit s_0 and s_Q because $s_0 = -\infty$ and $s_Q = \infty$. The width of the gap introduced between segments $\mathbf{s}_i$

[1] The function S is called a one-to-one into function, since its domain is mapped to unique points in the image (one-to-one) but not all points in the image have a preimage (into function). That is, all points of the image that are not within the gaps have an unique correspondence to the input domain, whereas the points within the gaps do not have corresponding points in the input domain.

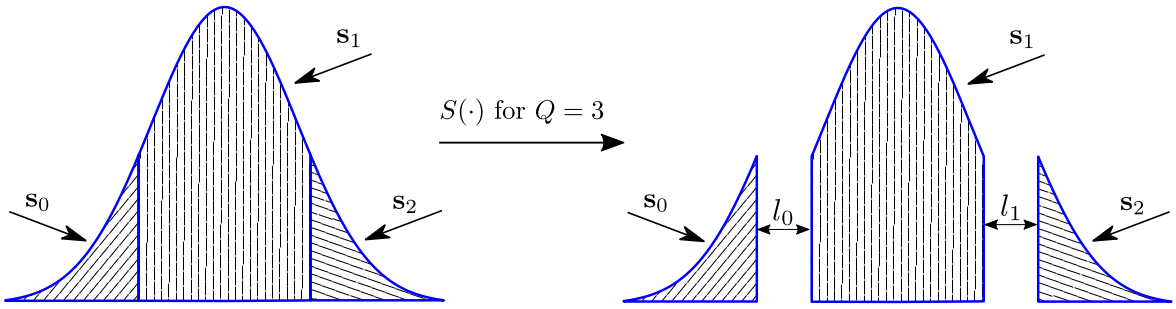


Figure 5.6: Proposed non-linear function S for the case of $Q = 3$ before normalization.

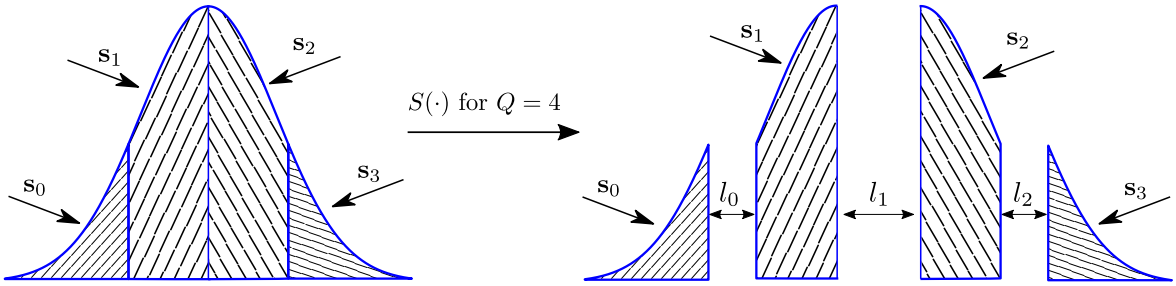
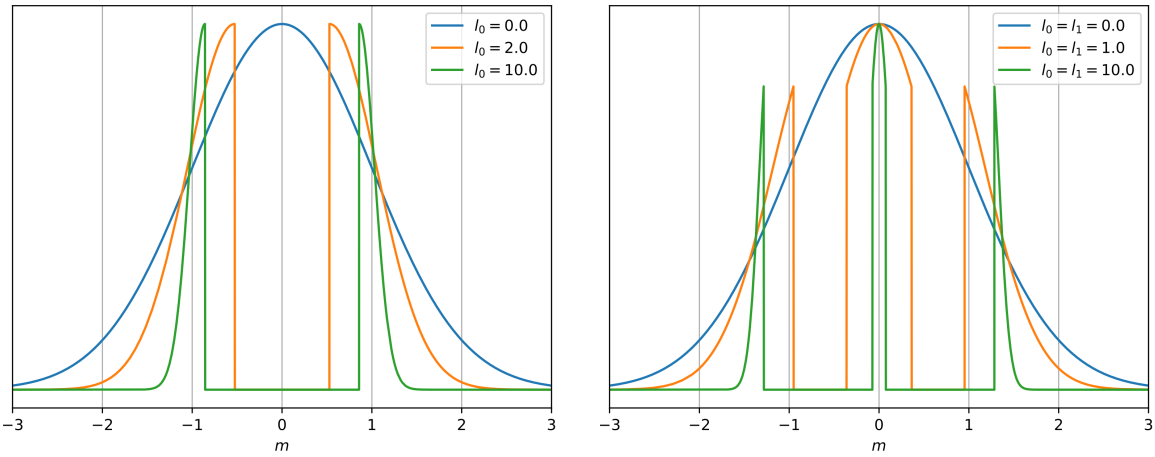


Figure 5.7: Proposed non-linear function S for the case of $Q = 4$ before normalization.

and $\mathbf{s}_{i+1}$ is defined as $l_i \in \mathbb{R}$. The widths are grouped into the vector $\mathbf{l} = [l_0, ..., l_{L-1}]$, where $L = Q/2$ for even Q and $L = (Q-1)/2$ otherwise. Note that this process does not force the output of S to have unit variance, which is desired since the weight vector $\mathbf{w}$ can be kept constant to generate symbols with the targeted E_s. Therefore, we normalize the output of S for unit variance.

In order to simplify the design, we constraint our parameters according to some guidelines. First, since the input distribution is symmetric, we set S to generate a symmetric distribution, which allows us to decrease the number of parameters in $\mathbf{s}$ and $\mathbf{l}$ since they now become anti-symmetric and symmetric, respectively[2]. Then, we set the vector defining the partition, $\mathbf{s}$, according to the optimal values for scalar quantization under mean square error distortion, as provided in [61]. While this does not guarantee optimality, since we are not exactly performing scalar quantization, it

[2] Symmetric vectors are of the form $[a, ..., b, c, c, b, ..., a]$ or $[a, ..., b, c, b, ..., a]$, while anti-symmetric are of the form $[-a, ..., -b, -c, c, b, ..., a]$ or $[-a, ..., -b, 0, b, ..., a]$.



(a) $Q = 2$ (b) $Q = 3$

Figure 5.8: Proposed non-linear function S for $Q = 2$ and $Q = 3$ after normalization so that the output has variance 1. The increase of l_0 and l_1 leads to a shrinkage of the lobes, which in the limit converge to deltas.

considerably simplifies the design.

After the gaps are created, we normalize the resulting distribution to have unit variance. Since the gaps spread out the distribution, the variance is increased, which requires each region to be shrunk so that unit variance is achieved. Figures 5.8a and 5.8b depict the distributions when S is applied to a Gaussian distribution for $Q = 2$ and $Q = 3$, respectively. Notice that the width of each region shrinks as l_0 and l_1 increase and in the limit it converges to delta functions. In the case of $Q = 2$, the limit actually results in the binary scheme performed by Rate Compatible Modulation.

Finally, the normalized function S is used to encode each source sample, m_i, into $g_i = S(m_i)$, which are linearly encoded by the random-like graph.

5.3 Decoder

As typically done for graph-based channel codes, we use the belief propagation algorithm [28] to decode the received observations, $\mathbf{y}^n$, after the transmission of $\mathbf{x}^n$ through the AWGN channel. The belief propagation algorithm is used to estimate the posterior probability density function of the source samples $p^p_{m_i}$. The proposed

decoder is a generalization of the algorithm used for graph-based codes in the digital domain. The messages passed from node to node are real numbers for the binary case, probability mass functions for the non-binary case, and probability density functions in the analog case.

A belief propagation decoder was derived to retrieve the noisy information encoded by an analog linear code in [66, 67]. Note that in the case of binary and non-binary sources, the measure of error is the bit-error-rate and symbol-error-rate, respectively, which are minimized by maximum-a-posteriori estimation. In the case of analog codes, we aim at minimizing the mean square error, which is minimized by computing the expectation of the posterior probability density function. In other words, the decoder estimates the posterior probability density function, $p^p_{m_i}$, for all symbols m_i, $i \in \{0, ..., k-1\}$, for both the discrete case and the continuous case in a similar manner. However, when the decoder makes a decision, it uses the mode of the posterior probability density function for the discrete case and the mean of the posterior probability density function for the continuous case.

The decoder in this chapter resembles the one from [66,67] with two differences. First, the non-linearity introduced by S. Second, [66,67] use check nodes instead of the projections we utilize here. Both differences will be addressed in the following sections.

5.3.1 Initialization

We start by passing the messages from the shifted nodes, g_i, to the symbol nodes x_j. The iteration number is denoted by t, where $t = 0$ during the first iteration. The only information at the input nodes during the first iteration is the prior knowledge from the source, which is the density $p^a_{m_i}(x) \sim \mathcal{N}(0, 1)$. Since message passing is performed between the variables g_i and x_j, and S is a one-to-one into function, we obtain the prior, $p^a_g(x)$, as

$$p^a_g(x) \propto \begin{cases} e^{\frac{-y^2}{2}}, & \text{for } x = S(y),\ y \in \mathbb{R} \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

Hence, the initial message transferred from the shifted node, g_i, to the symbol node, x_j, denoted by $q_{ij}^{(0)}(x)$, is expressed as $q_{ij}^{(0)}(x) = p_g^a(x)$.

5.3.2 Symbol Nodes

The message passed from the symbol node x_j to the shifted node g_i is denoted as $s_{ij}^{(t)}(x)$. As previously mentioned, the equation of the received symbol, y_j, is expressed as

$$y_j =< \mathbf{w}, \mathbf{g}_j > + n_j = \sum_{k=1}^{d_r} w_k g_{v|w(e_{vj})=w_k} + n_j, \tag{5.6}$$

where n_j is the AWGN realization over the symbol x_j and $\mathbf{g}_j = [g_{v|w(e_{vj})=w_1}, ..., g_{v|w(e_{vj})=w_{d_r}}]$ is the vector of shifted nodes connected to x_j ordered according to their weight index.

In order to compute $p(y_j|g_i)$, we start by merging all variables, except g_i, into the auxiliary variable, $z_{j\backslash i}$, expressed as

$$z_{j\backslash i} = \sum_{\substack{k=1 \\ v\neq i}}^{d_r} w_k g_{v|w(e_{vj})=w_k} + n_j, \tag{5.7}$$

and with probability density function

$$p_{z_{j\backslash i}} \propto \left(\circledast_{\substack{k=1 \\ v\neq i}}^{d_r} p(w_k g_{v|w(e_{vj})=w_k}) \right) \circledast \mathcal{N}(0, \sigma_r^2), \tag{5.8}$$

where $\circledast$ is the convolution operator and $p(w_k g_{v|w(e_{vj})=w_k})$ is the probability density function of the product of shifted node g_v and weight w_k. Note that the estimates $p(g_{v|w(e_{vj})=w_k})$ are obtained from the previous iterations, therefore leading to

$$p_{z_{j\backslash i}} \propto \left(\circledast_{\substack{k \text{ s.t. } e_{kj}\in E_j \\ k\neq i}} q_{kj}^{\prime(t-1)} \right) \circledast \mathcal{N}(0, \sigma_r^2), \tag{5.9}$$

where $q_{kj}^{\prime(t-1)}$ is the properly scaled message (probability density function) passed from shifted node g_k to symbol node x_j, E_j is the set of edges of the graph departing from

symbol node x_j, e_{kj} is the edge connecting g_k and x_j, and $w(e_{kj})$ corresponds to the weight of the edge between nodes g_k and x_j.

Next, we use the auxiliary variable from (5.7) in (5.6) obtaining

$$y_j = z_{j\backslash i} + w(e_{ij})g_i, \tag{5.10}$$

and then we calculate

$$p(y_j|g_i) \propto \int_{\infty}^{\infty} p(y_j|g_i, z_{j\backslash i})p(z_{j\backslash i})dz_{j\backslash i} \tag{5.11}$$

$$\propto \int_{\infty}^{\infty} \delta(y_j - w(e_{ij})g_i - z_{j\backslash i})p(z_{j\backslash i})dz_{j\backslash i} \tag{5.12}$$

$$\propto p_{z_{j\backslash i}}(y_j - w(e_{ij})g_i), \tag{5.13}$$

where $\delta(\cdot)$ is the delta function and $p(z_{j\backslash i})$ denotes the probability density function, $p_{z_{j\backslash i}}(\cdot)$, calculated by (5.9).

Finally, the outgoing message from the symbol node, x_j, to the shifted node, g_i, denoted by $s_{ij}^{(t)}(x)$ is given by

$$s_{ij}^{(t)}(x) \propto \left(\circledast_{\substack{k \text{ s.t. } e_{kj}\in E_j \\ k\neq i}} q_{kj}^{\prime(t-1)} \right) \circledast \mathcal{N}(0, \sigma_r^2)(y_j - w(e_{ij})x). \tag{5.14}$$

where the right-most parenthesis indicates that $s_{ij}^{(t)}$ is a function of x scaled by $-w(e_{ij})$ and shifted by y_j.

5.3.3 Shifted Nodes

The message passed from the shifted node g_i to the symbol node x_j at the t-th iteration is denoted by $q_{ij}^{(t)}$. As in [66, 67], the outgoing message $q_{ij}^{(t)}$ is equal to the

point-wise product of the incoming messages, which includes the prior p_g^a, leading to

$$q_{ij}^{(t)}(x) \propto p_g^a(x) \times \left(\prod_{v \text{ s.t. } e_{iv} \in E^i, v \neq j} s_{iv}^{(t-1)}(x) \right), \tag{5.15}$$

where E^i is the set of edges departing from node g_i.

5.3.4 Decision

Different from previous schemes, the decision process is not focused on estimating g_i, but rather its associated source sample m_i. For this purpose, we first estimate the posterior $p_{g_i}^p$ by using the messages incoming at an shifted node, allowing us to obtain

$$p_{g_i}^p(x) \propto p_g^a(x) \times \left(\prod_{v \text{ s.t. } e_{iv} \in E^i} s_{iv}^{(t-1)}(x) \right). \tag{5.16}$$

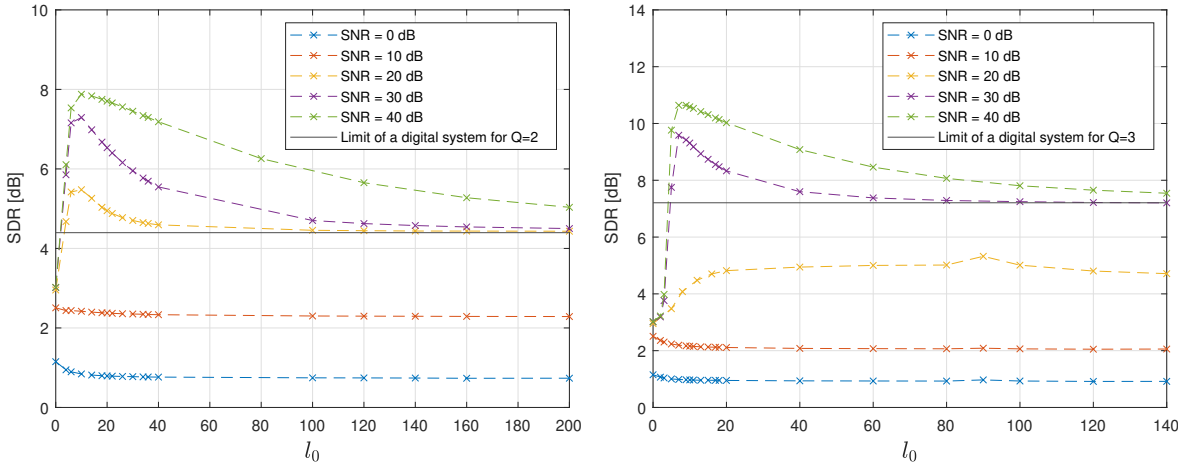
Then, we obtain the posterior probability density function of input node m_i by applying the inverse to transformation S, so that

$$p_{m_i}^p(x) \propto p_{g_i}^p(y) \text{ for } y = S(x), x \in \mathbb{R}. \tag{5.17}$$

The source sample estimate, $\hat{m}_i$, is obtained by computing the expectation of this posterior probability.

5.4 Results

In this section we perform Monte Carlo simulations to evaluate the performance of the proposed scheme over the AWGN channel. The weight vector of the proposed scheme, $\mathbf{w}$, is obtained from [25] and equal to the d_r leftmost components of the vector $[1/2, 1/3, 1/5, 1/7, 1/11, 1/13]$. Moreover, each messaged passed (probability density function) is quantized on the real line from -50 to 50 with 50001 points (resolution of 0.002) and the decoder processes 15 iterations per block. As previously mentioned, the function S generates gaps in the distribution of the source samples, which are defined



(a) $Q = 2$ (b) $Q = 3$

Figure 5.9: Simulation results for $d_r = 4$ and $R_c = 2$ as a function of the gap l_0.

by vectors that represent the separated regions and the width of the generated gaps, denoted by $\mathbf{s}$ and $\mathbf{l}$, respectively. The partition vector, $\mathbf{s}$, is defined using the optimal points for scalar quantization obtained from [61] and presented in Table 5.1. The vector of widths, $\mathbf{l} = [l_0, ..., l_{Q-1}]$, is optimized by Monte Carlo, and its size varies according to the number of segments in the partition, Q. Since we constraint the output distribution of S to be symmetric, $l_0 = l_1$ for $Q = 3$ and $l_0 = l_2$ for $Q = 4$.

Number of segments (Q)	Partition vector ($\mathbf{s}$) [61]
2	$[-\infty, 0.0, \infty]$
3	$[-\infty, -0.6120, 0.6120, \infty]$
4	$[-\infty, -0.9816, 0.0, 0.9816, \infty]$
5	$[-\infty, -1.244, -0.3823, 0.3823, 1.244, \infty]$

Table 5.1: The proposed non-linear function S partitions the Gaussian source according to the indicated regions.

First, we evaluate the proposed method by simulating 80 blocks for a 300:150 analog system ($R_c = 2$). Figures 5.9a and 5.9b depict the performance of the proposed system for several SNRs as a function of the gap generated by the function S, for $Q = 2$ and $Q = 3$, respectively. In addition, we also depict the limit of an optimal digital system, where scalar quantization with Q levels is performed at each source sample, as in [61]. Note that a purely linear scheme is obtained when $l_0 = 0$ since non-linearities

are not produced by S in that case. As l_0 increases in both Figures, the performance improves initially, except for the scenarios where SNR = 0 dB and SNR = 10 dB, which is expected because the performance of a linear system is good in those channel conditions. The explanation of this behavior is that the width of each partition has fine-grain information of the encoded values, whereas the separation introduces coarse-grain information (discontinuities). This creates a tradeoff because as the separation increases the width of each segment decreases.

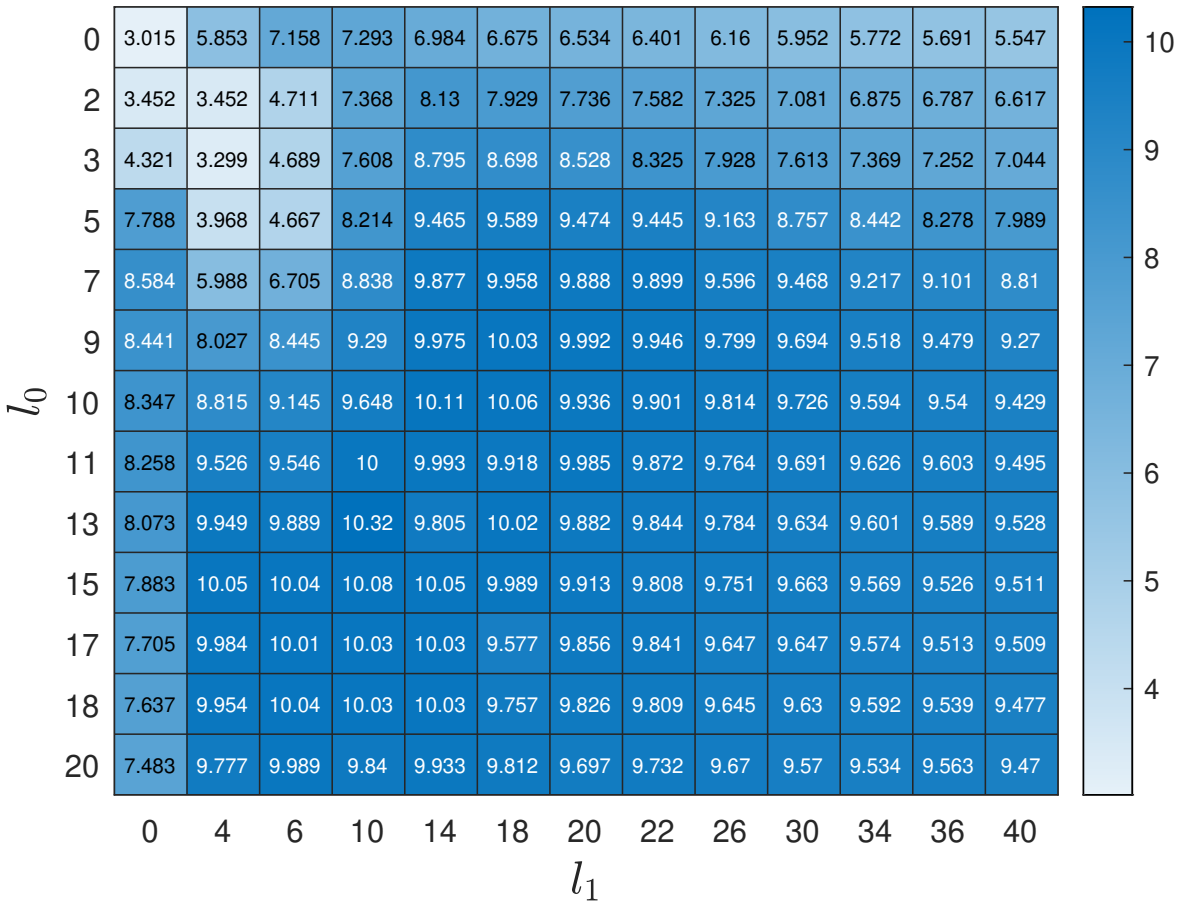


Figure 5.10: Simulation results for SNR=30 dB, $Q = 4$, $d_r = 4$, and $R_c = 2$ as a function of the spacings, $\mathbf{l} = [l_0, l_1, l_2]$ ($l_0 = l_2$). The SDR of each data point is depicted inside the squares.

When SNR is high, the SDR improves when the gap increases until it reaches a maximum. Then, the SDR decreases up to the SDR limit for the corresponding Q of

the digital system. Since the output distribution of S converges to deltas as the gaps go to infinity, and since the regions are set by the optimal scalar quantizer, it is natural for the performance at high SNR to converge to the quantization error caused by the scalar quantization. Remarkably, the proposed system is able to perform better than the linear system and it is not bounded by the SDR limit of scalar quantization. Figure 5.10 depicts the grid search performed to optimize the vector of widths, $\mathbf{l} = [l_0, l_1, l_2]$, for the case with $Q = 4$ and SNR $= 30$ dB ($l_0 = l_2$). Note that a behavior similar to that of $Q = 2$ and $Q = 3$ can be observed: as l_0 and l_1 increase, the performance suddenly increases and then slowly decreases once a maximum is reached.

Figure 5.11 compares the proposed scheme to the OPTA and to the limits for scalar quantization (optimal digital system). The SDR was optimized for each channel condition by performing a Monte Carlo search over the vector $\mathbf{l}$. In the low SNR case, the non-linearity provided by the function S is not able to outperform the linear case, which is not a problem since the linear system provides reasonable performance. As the SNR increases, the proposed scheme rapidly outperforms the theoretical limit using analog linear codes, and for a fixed value of Q it is also able to provide SDRs higher than those achievable by (digital) systems using scalar quantization with Q values. Note that each curve in the figure scheme flattens as the channel conditions improve, but the envelope of the curves is parallel to the OPTA. This is an interesting result, as it is the first time it has been observed in the literature for a graph-based analog code decoded using message passing.

Next, we simulate 80 blocks of the proposed scheme for a 300:100 analog system ($R_c = 3$) transmitted over an AWGN channel. The same optimization procedure for the gap vector $\mathbf{l}$ is followed. The resulting SDR is depicted in Figures 5.12a and 5.12b for $d_r = 5$ with $Q = 2$ and $Q = 3$, respectively. In the case of Figure 5.12a, we again notice that linear systems are better than the proposed method for low SNR. However, for higher SNRs we are able to obtain gains when compared to a quantized system. Notice that, in the case of $R_c = 2$, the SDR also converges to the limit imposed by scalar quantization as l_0 increases, since in that case the function S converges to deltas

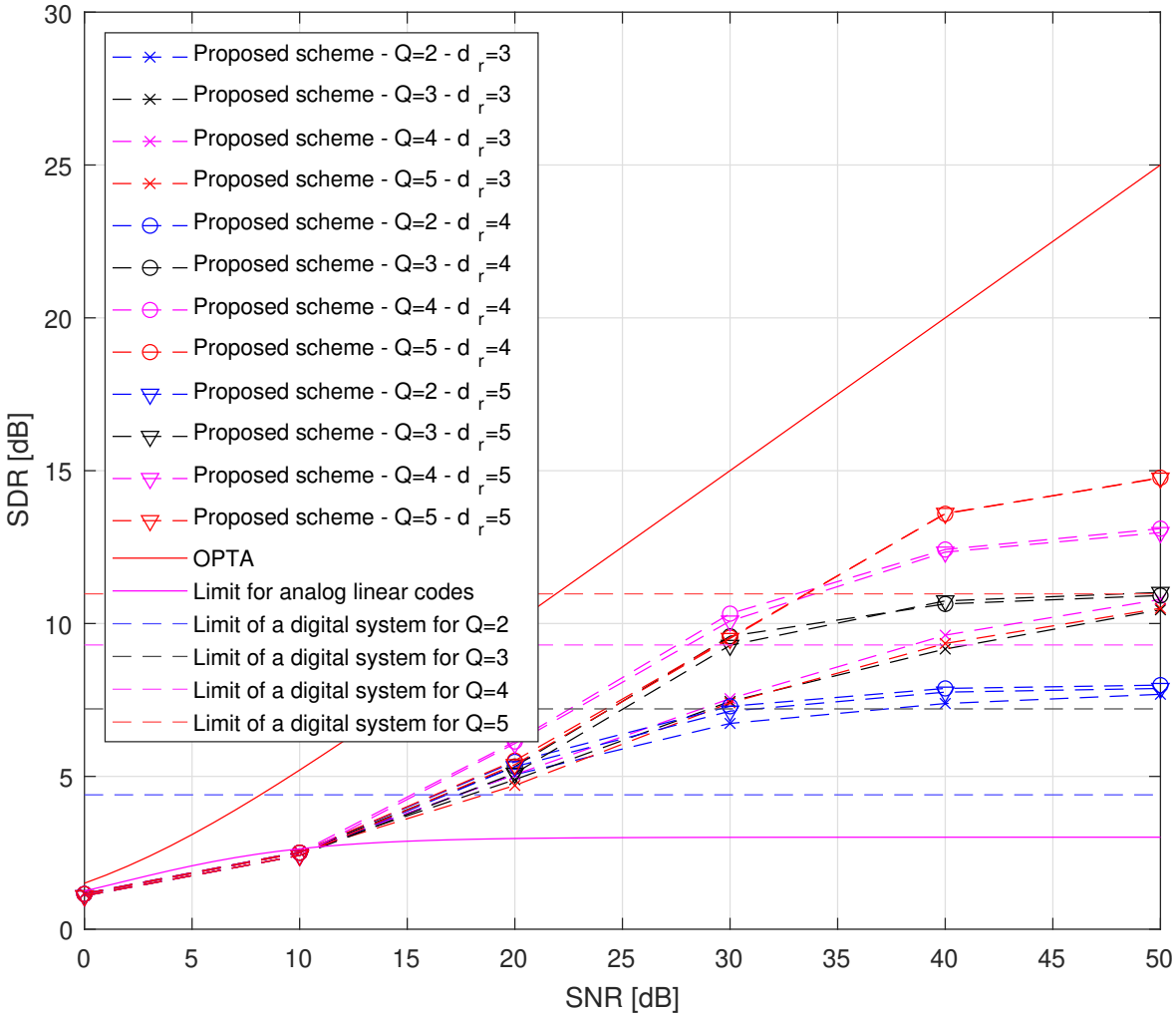
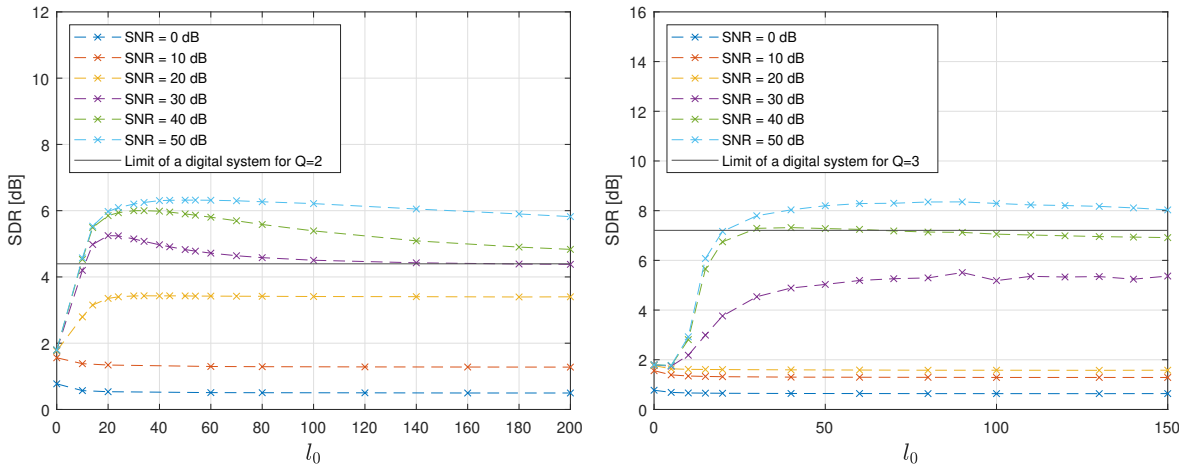


Figure 5.11: Simulation results for the proposed method with $R_c = 2$. Each data point represents the best SDR obtained after Monte Carlo optimization of the gap vector $\mathbf{l}$.

(binary system). Figure 5.12b shows a similar behavior. Notice that the sensitivity of the SDR as l_0 increases is small for low and medium SNRs, showing that adjusting l_0 is more important for the high SNR case.

Figure 5.13 presents simulation results when $R_c = 3$. Different from the previous case where $R_c = 2$, Q plays a more important role here, since $Q = 2$ clearly outperforms $Q = 3$ for low SNR, which is not the case for $R_c = 2$.



(a) $Q = 2$ (b) $Q = 3$

Figure 5.12: Simulation results for $d_r = 5$ and $R_c = 3$ as a function of the spacing l_0.

5.5 Conclusion

In this chapter we have presented a novel analog coding scheme that transmits Gaussian i.i.d. samples through the AWGN channel by using a random-like graph. The motivation is to provide rate flexibility in a simple way, which is not achieved by traditional schemes for joint source-channel coding of analog sources. The novelty of the proposed scheme is the insertion of non-linearities into each input sample prior to linear coding. In this way, the overall system is non-linear and, therefore, not limited by the OPTA for linear systems. The non-linearities are designed so that it is possible to transition between an analog and a digital system by simply adjusting some system parameters. The proposed analog graphical codes are the first ones in the literature that achieve performance close to the theoretical limits for the transmission of Gaussian sources over Gaussian channels, easily outperforming analog linear codes and surpassing the limits that would be imposed by scalar quantization.

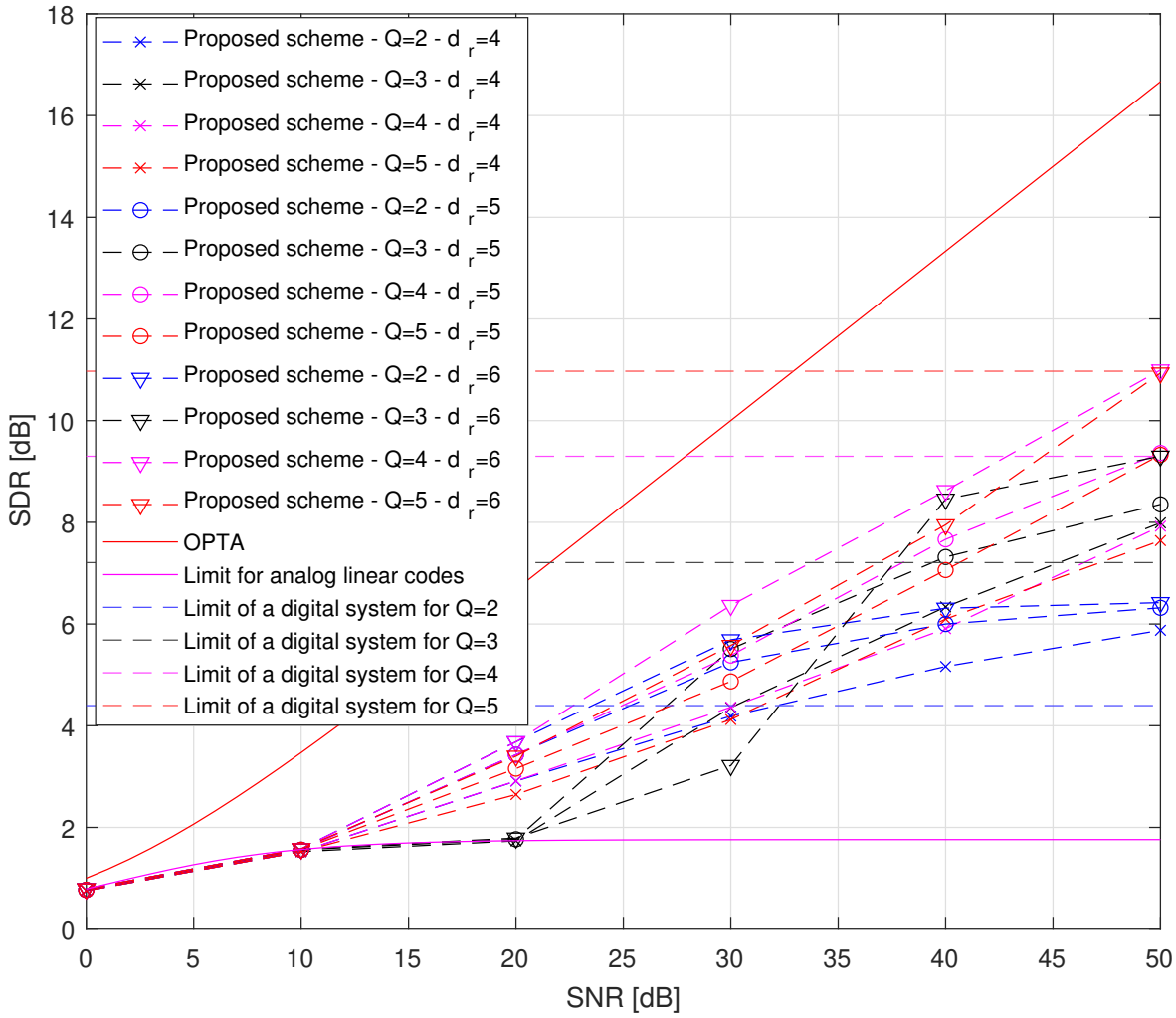


Figure 5.13: Simulation results for the proposed method with $R_c = 3$. Each data point represents the best SDR obtained after Monte Carlo optimization of the gap vector $\mathbf{l}$.

Chapter 6

CONCLUDING REMARKS AND FUTURE WORK

In this dissertation we have studied and developed a general framework for non-linear graph-based random error correcting codes. In addition to the intrinsic rate flexibility of these graph-based codes, they are very well suited for joint source-channel coding.

The main characteristic of all the proposed schemes is the use of non-linear mappings as constituent blocks of a random-like graph used in the encoding and decoding processes. In general, non-linear mappings have the capability of outperforming their linear counterparts, which serves as motivation for our work as a whole. Moreover, the random-like encoding graph allows the source statistics to be directly transferred to the output symbols, which increases the synergy between source and channel, strongly motivating the use of the proposed schemes for joint source-channel coding.

We started our work by designing non-linear mappings for uniform sources, which was crucial for our understanding of good design guidelines for the non-linear mappings. Then, focusing on joint source-channel coding, we applied these ideas to binary non-uniform sources, where we were able to achieve good performance with respect to the Shannon limit. Due to the design flexibility of the graph-based scheme, we expanded it to the transmission of non-binary sources, which allowed us to also transmit quantized symbols of a Gaussian source. Finally, we showed that the proposed framework can be once again generalized to transmit a Gaussian source without any type of quantization. In that case, the proposed concatenation of linear and non-linear schemes was able to easily outperform the linear scenario for medium and high SNRs.

6.1 Future Work

As shown in this work, the link between input vectors at Hamming distance 1 and the Euclidean distance of their associated constellation points is crucial. However, this is not enough to fully characterize the behavior of the proposed codes, weather binary or non-binary. Therefore, as future work, it would be interesting to further study the role of other distances, specially at higher rates, aiming at developing design methods that systematically take into account a profile of distances. In addition, by studying the distance profile of the code, it might also be possible to better understand the tradeoff between error floor and waterfall, allowing for the design of better mappings in the region under the Shannon limit. The expansion of this work to fading channels is also of great interest.

Regarding joint source-channel coding, while the proposed non-binary codes perform well, their performance in the region below the Shannon limit is worse than that of their binary counterparts. Good performance in that region is of interest for systems that do not require perfect correction. Therefore, the design of codes that perform better for SNRs below the Shannon limit is an interesting research topic.

Finally, the use of graph-based codes for the transmission of analog sources is very novel and many research directions are possible, such as studying the type of non-linearity that should be included, investigating the existence of simple mappings without linear components, and developing simplified decoders.

BIBLIOGRAPHY

[1] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[2] J. Proakis, *Digital Communications*. Boston: McGraw-Hill, 2008.

[3] M. E. Burich, R. D. Souza, and J. Garcia-Frias, "Non-Linear Rate Compatible Modulation : Invited Presentation," in *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2019, pp. 1–5.

[4] T. M. Thompson, *From Error-Correcting Codes through Sphere Packings to Simple Groups*. Mathematical Association of America, 1983.

[5] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic Coding for Data Compression," *Commun. ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.

[6] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.

[7] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Proceedings of ICC '93 - IEEE International Conference on Communications*, vol. 2, May 1993, pp. 1064–1070 vol.2.

[8] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, no. 18, p. 1645, 1996.

[9] E. Arikan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[10] R. V. Cox, J. Hagenauer, N. Seshadri, and C.-W. Sundberg, "Subband speech coding and matched convolutional channel coding for mobile radio channels," *IEEE Transactions on Signal Processing*, vol. 39, no. 8, pp. 1717–1731, Aug. 1991.

[11] F. I. Alajaji, N. C. Phamdo, and T. E. Fuja, "Channel codes that exploit the residual redundancy in CELP-encoded speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 5, pp. 325–336, Sep. 1996.

[12] J. Hagenauer, "Source-controlled channel decoding," *IEEE Transactions on Communications*, vol. 43, no. 9, pp. 2449–2457, Sep. 1995.

[13] J. Garcia-Frias and J. D. Villasenor, "Combining hidden Markov source models and parallel concatenated codes," *IEEE Communications Letters*, vol. 1, no. 4, pp. 111–113, Jul. 1997.

[14] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. 28, no. 1, pp. 55–67, Jan. 1982.

[15] H. Cui, C. Luo, K. Tan, F. Wu, and C. W. Chen, "Seamless rate adaptation for wireless networking," in *Proceedings of the 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '11. New York, NY, USA: Association for Computing Machinery, Oct. 2011, pp. 437–446.

[16] M. Wang, J. Wu, S. F. Shi, C. Luo, and F. Wu, "Fast Decoding and Hardware Design for Binary-Input Compressive Sensing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 591–603, Sep. 2012.

[17] H. Cui, C. Luo, J. Wu, C. W. Chen, and F. Wu, "Compressive Coded Modulation for Seamless Rate Adaptation," *IEEE Transactions on Wireless Communications*, vol. 12, no. 10, pp. 4892–4904, Oct. 2013.

[18] L. Li and J. Garcia-Frias, "Hybrid analog-digital coding scheme based on parallel concatenation of linear random projections and LDGM codes," in *2014 48th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2014, pp. 1–6.

[19] T. Oenning and J. Moon, "A low-density generator matrix interpretation of parallel concatenated single bit parity codes," *IEEE Transactions on Magnetics*, vol. 37, no. 2, pp. 737–741, Mar. 2001.

[20] J. Garcia-Frias and W. Zhong, "Approaching Shannon performance by iterative decoding of linear codes with low-density generator matrix," *IEEE Communications Letters*, vol. 7, no. 6, pp. 266–268, Jun. 2003.

[21] L. Li and J. Garcia-Frias, "Hybrid analog-digital coding for nonuniform memoryless sources," in *2015 49th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2015, pp. 1–5.

[22] L. Li, "A hybrid analog-digital coding scheme for digital sources," Ph.D. dissertation, University of Delaware, 2018.

[23] I. Granada, P. M. Crespo, and J. Garcia-Frías, "Rate Compatible Modulation for Non-Orthogonal Multiple Access," *IEEE Access*, vol. 8, pp. 224 246–224 259, 2020.

[24] I. Granada, P. M. Crespo, M. E. Burich, and J. Garcia-Frías, "Rate Compatible Modulation for Correlated Information Sources," *IEEE Access*, vol. 9, pp. 65 449–65 465, 2021.

[25] M. Shirvanimoghaddam, Y. Li, and B. Vucetic, "Near-Capacity Adaptive Analog Fountain Codes for Wireless Channels," *IEEE Communications Letters*, vol. 17, no. 12, pp. 2241–2244, Dec. 2013.

[26] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems (Corresp.)," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 384–386, May 1978.

[27] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[28] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.

[29] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms*. USA: Cambridge University Press, 2002.

[30] M. E. Burich, R. D. Souza, and J. Garcia-Frias, "Discretized Density Evolution for Rate Compatible Modulation : Invited Presentation," in *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2019, pp. 1–6.

[31] I. Granada, P. M. Crespo, and J. Garcia-Frías, "Asymptotic BER EXIT chart analysis for high rate codes based on the parallel concatenation of analog RCM and digital LDGM codes," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, p. 11, Dec. 2019.

[32] R. Duan, R. Liu, M. Shirvanimoghaddam, Y. Li, and C. W. Chen, "A Low PAPR Constellation Mapping Scheme for Rate Compatible Modulation," *IEEE Communications Letters*, vol. 20, no. 2, pp. 256–259, Feb. 2016.

[33] M. Shirvanimoghaddam, M. S. Mohammadi, R. Abbas, A. Minja, C. Yue, B. Matuz, G. Han, Z. Lin, W. Liu, Y. Li, S. Johnson, and B. Vucetic, "Short Block-Length Codes for Ultra-Reliable Low Latency Communications," *IEEE Communications Magazine*, vol. 57, no. 2, pp. 130–137, Feb. 2019.

[34] K. Zhang, J. Jiao, Z. Huang, S. Wu, and Q. Zhang, "Finite Block-Length Analog Fountain Codes for Ultra-Reliable Low Latency Communications," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1391–1404, Mar. 2020.

[35] W. J. Lim, R. Abbas, Y. Li, B. Vucetic, and M. Shirvanimoghaddam, "Analysis and Design of Analog Fountain Codes for Short Packet Communications," *IEEE*

Transactions on Vehicular Technology, vol. 70, no. 12, pp. 12 662–12 674, Dec. 2021.

[36] Sae-Young Chung, G. D. Forney, T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Communications Letters*, vol. 5, no. 2, pp. 58–60, Feb. 2001.

[37] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[38] J.-f. Cheng and R. J. Mceliece, "Some High-Rate Near Capacity Codecs for the Gaussian Channel," in *34th Allerton Conference on Communications, Control and Computing*, 1996.

[39] J. Wu, Z. Teng, H. Cui, C. Luo, X. Huang, and H. Chen, "Arithmetic-BICM for Seamless Rate Adaptation for Wireless Communication Systems," *IEEE Systems Journal*, vol. 10, no. 1, pp. 228–239, Mar. 2016.

[40] A. Hocquenghem, "Codes Correcteurs d'Erreurs," *Chiffres*, vol. 2, pp. 147–156, 1959.

[41] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error correcting binary group codes," *Information and Control*, vol. 3, no. 1, pp. 68–79, Mar. 1960.

[42] ——, "Further results on error correcting binary group codes," *Information and Control*, vol. 3, no. 3, pp. 279–290, Sep. 1960.

[43] I. Ochoa, P. M. Crespo, and M. Hernaez, "LDPC Codes for Non-Uniform Memoryless Sources and Unequal Energy Allocation," *IEEE Communications Letters*, vol. 14, no. 9, pp. 794–796, Sep. 2010.

[44] G. I. Shamir and J. J. Boutros, "Non-systematic low-density parity-check codes for nonuniform sources," in *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.*, Sep. 2005, pp. 1898–1902.

[45] R. D. Souza, G. I. Shamir, J. Garcia-Frias, and Kai Xie, "Non-systematic turbo coding with unequal energy allocation for nonuniform memoryless sources," in *Proceedings. International Symposium on Information Theory, 2005. ISIT 2005.*, Sep. 2005, pp. 1893–1897.

[46] F. Cabarcas, R. D. Souza, and J. Garcia-Frias, "Turbo coding of strongly nonuniform memoryless sources with unequal energy allocation and PAM signaling," *IEEE Transactions on Signal Processing*, vol. 54, no. 5, pp. 1942–1946, May 2006.

[47] ——, "Source-controlled turbo coding of nonuniform memoryless sources based on unequal energy allocation," in *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, Jun. 2004, pp. 166–.

[48] G.-c. Zhu, F. Alajaji, J. Bajcsy, and P. Mitran, "Nonsystematic Turbo codes for non-uniform i.i.d. sources over AWGN channels," in *Proc. 2002 Conf. on Inform. Sciences and Systems*, 2002.

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

[50] D. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[51] D. Sommer and G. P. Fettweis, "Signal Shaping by Non-Uniform QAM for AWGN Channels and Applications Using Turbo Coding," in *Proceedings ITF Conference Source and Channel Coding*, 2000, pp. 81–86.

[52] K. Liu and J. García-Frías, "Error floor analysis in LDGM codes," in *2010 IEEE International Symposium on Information Theory*, Jun. 2010, pp. 734–738.

[53] I. Granada, P. M. Crespo, and J. Garcia-Frías, "Combining the Burrows-Wheeler Transform and RCM-LDGM Codes for the Transmission of Sources with Memory at High Spectral Efficiencies," *Entropy*, vol. 21, no. 4, p. 378, Apr. 2019.

[54] Y. Hong, "On computing the distribution function for the Poisson binomial distribution," *Computational Statistics & Data Analysis*, vol. 59, pp. 41–51, Mar. 2013.

[55] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the society for industrial and applied mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[56] M. Davey and D. MacKay, "Low density parity check codes over GF(q)," in *1998 Information Theory Workshop*, Jun. 1998, pp. 70–71.

[57] J. Berkmann, "Symbol-by-Symbol MAP Decoding of Nonbinary Codes," 1998.

[58] C. Berrou, M. Jezequel, C. Douillard, and S. Kerouedan, "The advantages of non-binary turbo codes," in *Proceedings 2001 IEEE Information Theory Workshop*, Sep. 2001, pp. 61–63.

[59] E. Şaşoğlu, E. Telatar, and E. Arikan, "Polarization for arbitrary discrete memoryless channels," in *2009 IEEE Information Theory Workshop*, Oct. 2009, pp. 144–148.

[60] R. Mori and T. Tanaka, "Non-binary polar codes using Reed-Solomon codes and algebraic geometry codes," in *2010 IEEE Information Theory Workshop*, Aug. 2010, pp. 1–5.

[61] J. Max, "Quantizing for minimum distortion," *IRE Transactions on Information Theory*, vol. 6, no. 1, pp. 7–12, Mar. 1960.

[62] C. Shannon, "Communication in the Presence of Noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, Jan. 1949.

[63] Kyong-Hwa Lee and D. Petersen, "Optimal Linear Coding for Vector Channels," *IEEE Transactions on Communications*, vol. 24, no. 12, pp. 1283–1290, Dec. 1976.

[64] V. A. Vaishampayan, "Combined Source-Channel Coding for Bandlimited Waveform Channels," Ph.D. dissertation, University of Maryland, 1989.

[65] A. Fuldseth and T. Ramstad, "Bandwidth compression for continuous amplitude channels based on vector approximation to a continuous subset of the source signal space," in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, Apr. 1997, pp. 3093–3096 vol.4.

[66] F. Hu and W. Henkel, "An Analysis of the Sum-Product Decoding of Analog Compound Codes," in *2007 IEEE International Symposium on Information Theory*, Jun. 2007, pp. 1886–1890.

[67] F. Hu, "Analog codes for analysis of iterative decoding and impulse noise correction," Thesis, Aachen : Shaker, 2009.

Appendix A

RELATIONSHIP BETWEEN HAMMING AND EUCLIDEAN DISTANCES

Assume that the source bits connected to the symbol node x_i are the components of vector $\mathbf{v}$. If we also assume that all messages arriving at symbol node x_i are correctly decoded (perfect messages), we obtain

$$
\begin{aligned}
s_{ik}^{(t)} &= \log \frac{\sum\limits_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ik}}=1}} e^{\frac{-(y_k - f(\mathbf{b}))^2}{N_0}} P(\mathcal{M}_{k\setminus i} = \mathbf{b}^{w_{ik}})}{\sum\limits_{\substack{\mathbf{b}\in\{0,1\}^{d_r} \\ \text{s.t. } b_{w_{ik}}=0}} e^{\frac{-(y_k - f(\mathbf{b}))^2}{N_0}} P(\mathcal{M}_{k\setminus i} = \mathbf{b}^{w_{ik}})} \\
&= \log \frac{e^{\frac{-(y_k - f(\mathbf{v}^1_{w_{ik}}))^2}{N_0}}}{e^{\frac{-(y_k - f(\mathbf{v}^0_{w_{ik}}))^2}{N_0}}} \\
&= \frac{-(y_k - f(\mathbf{v}^1_{w_{ik}}))^2}{N_0} - \frac{-(y_k - f(\mathbf{v}^0_{w_{ik}}))^2}{N_0} \\
&= \frac{2y_k(f(\mathbf{v}^1_{w_{ik}}) - f(\mathbf{v}^0_{w_{ik}})) - f(\mathbf{v}^1_{w_{ik}})^2 + f(\mathbf{v}^0_{w_{ik}})^2}{N_0},
\end{aligned}
\tag{A.1}
$$

where the notation $\mathbf{v}_i^h$ means we insert the value h in the vector $\mathbf{v}$ at the position i. Since we are studying the probabilistic behavior of the whole graph, we take advantage of the fact that y_k is actually a random variable with conditional PDF $f(y_k|\mathbf{v}) \sim \mathcal{N}(f(\mathbf{v}), N_0/2)$, which makes $s_{ik}^{(t)}$ to also be Gaussian distributed with mean $\mathbb{E}[s_{ik}^{(t)}]$ and variance $\sigma^2[s_{ik}^{(t)}]$. Without loss of generality (only possible change of sign), we assume

$\mathbf{v}^1_{w_{ik}} = \mathbf{v}$, allowing us to derive the mean and variance of $s^{(t)}_{ik}$ as follows

$$\begin{aligned}
\mathbb{E}[s^{(t)}_{ik}] &= \mathbb{E}\left[\frac{2y_k(f(\mathbf{v}^1_{w_{ik}}) - f(\mathbf{v}^0_{w_{ik}})) - f(\mathbf{v}^1_{w_{ik}})^2 + f(\mathbf{v}^0_{w_{ik}})^2}{N_0}\right] \\
&= \frac{2\mathbb{E}[y_k](f(\mathbf{v}^1_{w_{ik}}) - f(\mathbf{v}^0_{w_{ik}})) - f(\mathbf{v}^1_{w_{ik}})^2 + f(\mathbf{v}^0_{w_{ik}})^2}{N_0} \\
&= \frac{2f(\mathbf{v}^1_{w_{ik}})(f(\mathbf{v}^1_{w_{ik}}) - f(\mathbf{v}^0_{w_{ik}})) - f(\mathbf{v}^1_{w_{ik}})^2 + f(\mathbf{v}^0_{w_{ik}})^2}{N_0} \\
&= \frac{(f(\mathbf{v}^1_{w_{ik}}) - f(\mathbf{v}^0_{w_{ik}}))^2}{N_0}
\end{aligned} \tag{A.2}$$

$$\begin{aligned}
&\sigma^2[s^{(t)}_{ik}] = \\
&= \sigma^2\left[\frac{2y_k(f(\mathbf{v}^1_{w_{ik}}) - f(\mathbf{v}^0_{w_{ik}})) - f(\mathbf{v}^1_{w_{ik}})^2 + f(\mathbf{v}^0_{w_{ik}})^2}{N_0}\right] \\
&= \sigma^2\left[\frac{2y_k(f(\mathbf{v}^1_{w_{ik}}) - f(\mathbf{v}^0_{w_{ik}}))}{N_0}\right] \\
&= \frac{4(f(\mathbf{v}^1_{w_{ik}}) - f(\mathbf{v}^0_{w_{ik}}))^2\sigma^2[y_k]}{{N_0}^2} = \frac{4(f(\mathbf{v}^1_{w_{ik}}) - f(\mathbf{v}^0_{w_{ik}}))^2 N_0}{2{N_0}^2} \\
&= \frac{2(f(\mathbf{v}^1_{w_{ik}}) - f(\mathbf{v}^0_{w_{ik}}))^2}{N_0}.
\end{aligned} \tag{A.3}$$

Finally, we evaluate the probability, P_e, for the message $s^{(t)}_{ik}$ to provide a wrong estimate of the i-th source bit, which occurs when the message is negative (since we are assuming that the source bit i is equal to 1). Note that this is not a definite measure of error, since the source bit i is connected to other edges which will hopefully correct an erroneous message. However, this metric can still be used to understand how Hamming

distances affect the decoder.

$$
\begin{aligned}
P_e &= \frac{1}{\sqrt{2\pi\sigma^2[s_{ik}^{(t)}]}} \int_{-\infty}^{0} e^{-(x-\mathbb{E}[s_{ik}^{(t)}])^2/2\sigma^2[s_{ik}^{(t)}]} dx \\
&= Q\left(\frac{\mathbb{E}[s_{ik}^{(t)}]}{\sigma[s_{ik}^{(t)}]}\right) = Q\left(\frac{\frac{(f(\mathbf{v}_{w_{ik}}^1)-f(\mathbf{v}_{w_{ik}}^0))^2}{N_0}}{\sqrt{\frac{2(f(\mathbf{v}_{w_{ik}}^1)-f(\mathbf{v}_{w_{ik}}^0))^2}{N_0}}}\right) \\
&= Q\left(\frac{|f(\mathbf{v}_{w_{ik}}^1) - f(\mathbf{v}_{w_{ik}}^0)|}{\sqrt{2N_0}}\right).
\end{aligned}
\tag{A.4}
$$

Since $Q(x)$ is a monotonically decreasing function, (A.4) states that the larger the Euclidean distance between $f(\mathbf{v}_{w_{ik}}^1)$ and $f(\mathbf{v}_{w_{ik}}^0)$, the more accurate the message $s_{ik}^{(t)}$ will be, given that the incoming messages at the k-th symbol node are perfect. This provides an intuitive explanation of the role P_e plays when the decoder is close to convergence (e.g., in the error floor region).

Notice that vectors $\mathbf{v}_{w_{ik}}^1$ and $\mathbf{v}_{w_{ik}}^0$ are binary vectors that are equal in all coordinates except w_{ik}, which means that their Hamming distance is equal to 1. This leads us to design the mapping f by controlling the Euclidean distance of the constellation symbols corresponding to input vectors at Hamming distance 1 from each other. In this way, we can increase and decrease the error floor, which also impacts the waterfall region.

Appendix B

MONOTONIC ENERGY ALLOCATION

To find the condition required by the proposed non-linear mappings to allocate power in a monotonic fashion we must find $p_0 > 0.5$ and d_r such that

$$P(W = i) > P(W = i + 2)\, \forall i \in \{0, ..., d_r - 2\}, \tag{B.1}$$

where W is binomially distributed as

$$P(W = i) = \binom{d_r}{i} p_0^{d_r - i}(1 - p_0)^i. \tag{B.2}$$

By expanding (B.1), we get

$$\binom{d_r}{i} p_0^{d_r - i}(1 - p_0)^i > \binom{d_r}{i + 2} p_0^{d_r - i - 2}(1 - p_0)^{i+2}$$

$$\binom{d_r}{i} \Big/ \binom{d_r}{i + 2} > \frac{(1 - p_0)^2}{p_0^2}.$$

Since $\binom{d_r}{i}/\binom{d_r}{i+1} = (i + 1)/(d_r - i)$ is monotonically increasing on i, so is $\binom{d_r}{i}/\binom{d_r}{i+2}$. Therefore, the following conditions must be satisfied

$$\binom{d_r}{i} \Big/ \binom{d_r}{i + 2} > \frac{(1 - p_0)^2}{p_0^2} \;\; \forall i \in \{0, ..., d_r - 2\}$$

$$\binom{d_r}{d_r - 2} \Big/ \binom{d_r}{d_r} > \binom{d_r}{d_r - 3} \Big/ \binom{d_r}{d_r - 1} > \cdots > \binom{d_r}{0} \Big/ \binom{d_r}{2} > \frac{(1 - p_0)^2}{p_0^2}.$$

Finally leading to

$$\binom{d_r}{0} \Big/ \binom{d_r}{2} > \frac{(1-p_0)^2}{p_0^2}$$

$$\frac{d_r(d_r-1)}{2} < \left(\frac{p_0}{1-p_0}\right)^2,$$

which for a fixed $p_0 > 0.5$ implies

$$d_r < \frac{1+\sqrt{1+8\left(\frac{p_0}{1-p_0}\right)^2}}{2}. \tag{B.3}$$

Appendix C

PERMISSIONS

Parts of Chapters 1 and 2 were extracted verbatim from an article presented at the 53rd Annual Conference on Information Sciences and Systems (CISS) © 2019 IEEE. Reprinted, with permission, from Mariano Eduardo Burich, Richard Demo Souza, and Javier Garcia-Frias, Non-Linear Rate Compatible Modulation: Invited Presentation, March 2019.

In reference to IEEE copyrighted material which is used with permission in this dissertation, the IEEE does not endorse any of University of Delaware's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.