Doctoral Dissertation

Finite Dimensional Lattice Code Design for Wireless Communications

XUE JIAJIE

Supervisor Brian M. Kurkoski

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Information Science)

September 2025

Abstract

Lattice code is a coded-modulation scheme defined over real numbers, considered as a candidate for next generation wireless communication. Lattice codes provide error correction ability. It is also shown that lattice codes can achieve lower transmission power compared to conventional QAM modulation. Besides, lattice codes preserve the linearity of codewords, therefore can be applied for physical layer network coding. It has been shown that lattice codes achieve the capacity of the Gaussian channel if lattice decoding is performed using optimal decoding coefficient(s).

This research studies finite dimensional lattice codes for practical systems. Single user transmission and multiple access relay using compute-forward (CF) are considered as communication scenarios. The following three challenges are addressed. 1) Even with optimal coefficients, finite dimensional systems have a non-zero error rate, which gives a room on improving error rate. 2) Traditional CF relaying does not have error detection ability, potentially forwarding erroneous packets into network, for which a error detection scheme is required. 3) A lattice design is considered to achieve lower error rate than classic designs.

In this work, a retry decoding scheme is proposed for both single user scenario and CF relaying, which allows additional decoding attempts at receiver to improve error rate by adjusting value(s) of decoding coefficient(s), when errors are detected. A lower bound and an estimate on error probability are derived for single user scenario. The CRC-embedded lattice/lattice code are proposed having error detection ability for enabling retry decoding. The CRC-embedded lattices/lattice codes rely on CRC codes, with modest complexity on error detection. Besides, the error detection is applicable for CF relaying, which was not feasible in conventional systems. For a 2-user CF relay, numerical results show gains of 1.29dB, 1.31dB and 1.08dB at equal error rate 10^{-5} are achieved by $n = 64, 128, 256$ polar lattice codes at code rate $R \approx 1.6406, 1.7422, 1.8438$, respectively, where only one additional decoding attempt is required. At last, a lattice design approach is provided using construction A and binary codes with known minimum Hamming distance and codeword multiplicity, the number of minimum weight codewords. Design examples consider extended BCH codes and polar codes, where lower error rates are achieved than that by classic design rules.

Keywords: Lattices, lattice codes, finite dimensional transmission, CRC codes, compute-forward relaying, construction A.

Acknowledgment

I would like to take this opportunity to my sincerely appreciate to all who have helped and supported me during study in pursuing doctoral degree.

First, I would like to sincerely thank my supervisor, Professor Brian Kurkoski, for giving my the opportunity, and also for his insights, encouragement, advice and guidance during my study and research. Your mentorship has not only guided me on advanced topics, but also the way of thinking as a researcher and professional writing skill to finish papers and this dissertation.

I would like to sincerely thank my dissertation committee, Professor Gregory Schwartzman, Professor Fujisaki Eiichiro, Professor Lim Yuto and Professor Emanuele Viterbo, for your time to review this work and critical comments that helped me to improve this work.

Special thanks to all lab member in the BITS lab for friendship, discussions and a lot of parties we had. And thanks to JAIST to provide support to my research and an excellent environment. Also I would like to thank my friends outside of lab for their friendship and encouragement.

Finally, I would like to give my deepest thanks to my parents and my family for their encouragement, support and love, which are definitely one of the biggest part during these years.

List of Abbreviations

AWGN	Additive white Gaussian noise
BCH code	Bose–Chaudhuri–Hocquenghem code
CF	Compute-forward
CRC	Cyclical redundancy check
CVP	Closest vector problem
EBCH	extended BCH
EER	Equation error rate
ICF	Incomplete compute-forward
IF	Integer forcing
LSB	Least significant bits
MAC	Multiple access channel
ML	Maximum likelihood
MMSE	Minimum mean square error
NSM	Normalized second moment
OSD	Ordered-statistics decoding
P2P	Point-to-point
PLNC	Physical layer network coding
SC	Successive cancellation
SCL	Successive cancellation list
SIVP	Shortest independent vector problem
SNR	Signal-to-noise ratio
SU	Single user
SVP	Shortest vector problem

VNR	Volume-to-noise ratio
WER	Word error rate

List of Symbols

$\mathcal{C}_b$	binary code
$\mathcal{C}_{CRC}$	CRC code
G_{CRC}	CRC generator polynomial
d_c	minimum Hamming distance
R_b	code rate of binary code
τ_c	codeword multiplicity
$\mathbf{a}$	integer coefficient vector of CF relaying
α	scaling factor of lattice decoder
α_{MMSE}	MMSE scaling factor of lattice decoder
$\mathcal{I}$	set of index
Λ	lattice
Λ_c	coding lattice
Λ_s	shaping lattice
$\mathbf{G}$	generator matrix of lattice
$\mathcal{V}$	Voronoi region of lattice
$\mathcal{C}$	lattice code
R	code rate of lattice code
Λ'	CRC-embedded lattice
$\mathcal{C}'$	CRC-embedded lattice code
$\mathbf{b}$	lattice uncoded message
$\mathbf{n}$	additive Gaussian noise
$\mathbf{x}$	transmitted message
$\mathbf{y}$	received message
$\hat{\mathbf{x}}, \hat{\mathbf{y}}$	estimate of $\mathbf{x}, \mathbf{y}$
$\mathcal{S}_c$	coving sphere
$\mathcal{S}_e$	effective sphere
θ	lattice theta series
θ'	truncated theta series

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Background

In wireless communications, a typical scenario is that a transmitter sends a message through a noisy channel, and at the other side, a receiver aims to recover this message to understand what the transmitter said. An error correcting code adds redundancy to the message to protect against the channel noise. By sharing the knowledge of the codebook, the receiver is able to correct errors introduced by channel noise to recover the message. Shannon showed that the error correction capability of channel codes is not unlimited and reliable communication, with arbitrary small error probability, is possible if and only if the code rate R is lower than the channel capacity C, also known as $R < C = \frac{1}{2}\log(1 + \text{SNR})$, where the term SNR is the signal-to-noise ratio. In the past decades, various types of channel codes have been developed and studied to approach the channel capacity. Binary codes, as the most studied class of channel codes, use information represented using bits 0 and 1. However, physical channels are usually real-valued. Due to the restricted alphabet size of bits, the full capacity of real-valued channels is not achieved by only using binary codes, for which other techniques, such as modulation, are required to increase the transmission code rate.

The capacity-achieving analysis for channel codes requires the code length goes to infinity. However, the performance of finite length codes is more interesting for practical systems. The Ultra-Reliable Low-Latency Communications (URLLC) is one of key service categories in the 5G standard. This requires channel codes achieves very low error rate, while having very low decoding latency. Since most of decoding algorithms have time complexity depending on the block length, short block length codes are expected to achieve lower decoding latency than that of long block length codes. In practice, control channels in the 5G standard applies polar codes, because of their strong performance at short block lengths.

In a network having multiple users, the multiple access channel (MAC) allows all users having access to the network. Traditional systems use

orthogonal multiple access which designates a physical channel to two users to establish a point-to-point (P2P) communication. In order to increase throughput of the network, the non-orthogonal MAC is introduced, which allows multiple users sending their messages simultaneously. In this dissertation, we always assume the non-orthogonal MAC, for which the messages are added over the air. And at the receiver side, multi-user detection (MUD) is performed to recover individual users' messages. Transmission through MAC with MUD expands the achievable rate region compared to orthogonal transmission, with increased decoding cost due to MUD. Another way to increase the total throughput is to expand the network to involve more users. It has a high cost to build physical links having long distance. Instead, relaying techniques are considered to transfer messages to establish a connection between users separated by long distances. Routing transfers messages sequentially from a relay to another to establish the end-to-end connections. In 2000, a novel physical layer relaying technique, called network coding, is proposed, which uses multiple access nodes for transferring packets and can significantly improve the network throughput by exploring the linearity of codes [1]. With higher throughput, physical layer network coding (PLNC) is able to improve transmission latency of P2P communications with routing in a wireless network, such as a wireless relaying network.

This research studies finite dimensional channel codes for error correction using on lattices, which is defined over the real number space. Lattices share the same algebra as the real-valued physical channels and can also be considered as a coded-modulation scheme for future communications. Structure and design of short block length lattice are particularly interested to improve error performance while maintaining low decoding latency. In addition, lattices are linear, thus can be applied as a PLNC scheme to improve latency caused by routing in wireless networks.

1.1.1 What is a lattice

Lattice is a mathematical concept, which is studied in geometry and group theory. Intuitively, a lattice is an arrangement of points in n-dimensional space. A lattice can also be seen as a partition of the space using a same shape. Each area in this partition is associated with a lattice point.

1.1.2 Where we can find lattices

"Lattices are everywhere", as literally stated by Zamir in [2], lattices appear in many places in nature and in human lives. One of the most famous example in nature is that the hexagonal honeycomb has a lattice structure

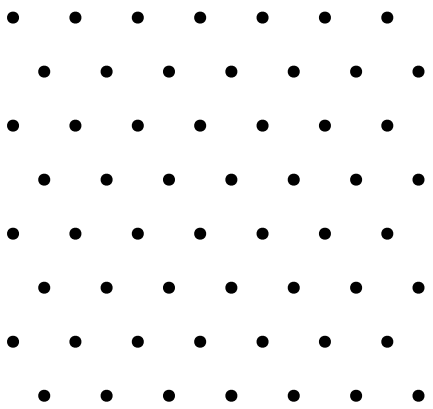


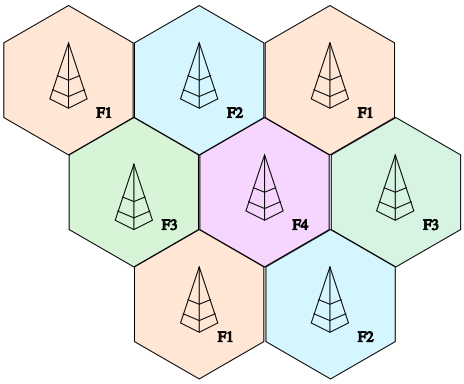
Figure 1.1: An example of lattice. From Wikipedia: Lattice(group).

in 2 dimension, as shown in Figure 1.2a. In human lives, an ideal coverage of base stations in a cellular network has the same 2-dimensional lattice structure as honeycombs, as shown in Figure 1.2b; an arrangement of cannon ball has a lattice structure in 3 dimension, as shown in Figure 1.2c.

Lattice are studied in many academic areas. In mathematics, lattices relate to various classic problems. Within a given space, the sphere packing problem (an arrangement of non-overlapping spheres within a containing space) and the sphere covering problems (minimum cost for covering a space with no empty) are studied in a long history. Conway and Sloane summarized lattices with dimension up to 24 which have the either optimal or the best known packing and covering among lattices having the same dimension [3]. Recently, a breakthrough by Viazovska and *et al.* proved that the E_8 lattice and Leech lattice, which were the best-known packing, are actually the densest packing in 8 and 24 dimension [4, 5]. In practical scenarios, the design of coverage of cellular network corresponds to the 2-dimensional covering problem. On the other hand, lattices relate to the shortest vector problem (SVP), which finds the shortest vector that connects two distinct lattice points and is known as NP-hard. Similar problems, such as the closest vector problem (CVP) and shortest independent vector problem (SIVP), are also known as NP-hard. More details related to SVP/CVP/SIVP can be found in [6].

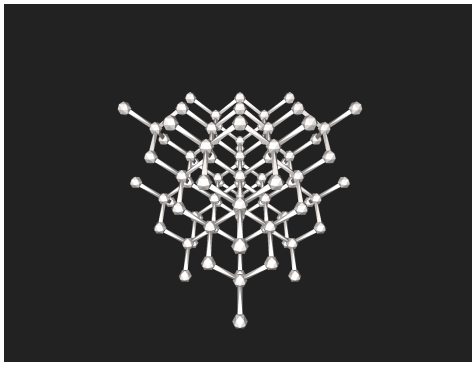
Lattice-based cryptography is considered as one of the post-quantum encryption schemes. Traditional public-key algorithms rely on mathematic problems which are difficult, such as the integer factorization problem. As the development of quantum computer, integer factorization can be solved in polynomial time using a quantum computer running Shor's algorithm.

(a) Honeycomb, by Matthew T Rader. Wikipedia: Honeycomb.



(b) Cellular network, by Andrew pmk. Wikipedia: Cellular network.

(c) Cannon ball, by Funkdooby. Wikipedia: Cannonball problem.



(d) Crystal structure of diamond, by Cmglee. Wikipedia: Bravais lattice.

Figure 1.2: Examples of lattices.

This gives a big challenge on finding a new encryption scheme, which should be quantum-safe. The NTRU encryption algorithm, first proposed at [7], is designed based on the lattice SVP, which is quantum-safe and has not yet been broken using quantum computers.

Lattices are also studied in chemistry, which can be used to represent the structure of crystals. Figure 1.2d shows the structure of diamond in 3-dimensional space, which is identical to a 3-dimensional cubic lattice.

At last, the area we are most interested in is lattices for information theory and communications. As a lattice is an arrangement of points in the real number space, it can be seen as a constellation of signal points for studying in information theory and communications. Besides the packing and covering problem described above, lattices are also studied for quantization under mean-squared error (MSE) measurement and error correcting codes

through unconstrained additive white Gaussian noise (AWGN) channel from the information theory perspective. The optimal or the best known lattices for quantization and AWGN channel coding up to 24 dimension are also summarized in [3]. The goodness of lattices for packing, covering, quantization and AWGN channel coding, respectively, are studied in [2, 8].

1.2 Related works

This section gives a historical review of lattices in information theory and communications related to this work. There is a long history of studying lattices. By definition (will be given in Chapter 2), a lattice is an infinite constellation, thus is power unconstrained. In [9], Poltyrev showed that there exists a lattice with asymptotically large dimension which achieves the capacity of power unconstrained AWGN channel. Later in [8], Erez, Litsyn and Zamir showed the existence of lattices that achieve the Poltyrev's bound by applying construction A to a random linear code over a prime size field of growing size for asymptotically large dimension (such lattices are also simultaneously good for sphere packing, covering and quantization as well). For power constrained communications, a lattice code can be constructed using a coding lattice and a shaping lattice. A lattice code is a finite constellation and satisfies a given power constraint. It was proven that lattice codes with asymptotically large dimension achieve the capacity of the AWGN channel with power constraint. This was first shown that the channel capacity is achieved using maximum likelihood (ML) decoding by choosing a shaping region as a n-dimensional thin shell [10, 11]. This result was then extended by Urbanke, where the shaping region is extended to the whole n-ball in [12]. This can be intuitively explained that in high dimensions, most of codewords of a lattice code lie in the thin shell defined in [10,11]. However, ML decoding has exponential complexity in terms of the dimension of lattices. Lattice decoder is a nearest point decoder, which may or may not output a codeword since it ignores the boundary of the codebook, and achieves lower complexity than ML decoder. In [13], it has been shown that only part of the channel capacity can be achieved with the rate $R < \frac{1}{2}\log(\text{SNR})$, where "+1" terms suggested by Shannon was dropped. More significantly, Erez and Zamir showed that the full capacity can be achieved using lattice decoder, if the received message is scaled using an minimum mean square error (MMSE) factor, denoted as $\alpha_{MMSE} = \text{SNR}/(1 + \text{SNR})$, before decoding [14], i.e. $R < C = \frac{1}{2}\log(1 + \text{SNR})$ can be achieved.

Besides the excellent results from asymptotically analysis, there are also studies related to the design of finite dimensional lattices/lattice codes. As

shown in [8], di Pietro and *et al.* proposed the low density construction A (LDA) lattices using prime size field LDPC codes with construction A, which gives $\approx$ 0.7dB gap to the Poltyrev's bound at dimension $n = 10000$ [15]. Later, it was shown that LDA lattice codes asymptotically achieve the power constrained AWGN channel capacity [16], and are good for packing and quantization [17]. Another type of lattices having excellent error performance at large dimensions is the low density lattice code (LDLC) [18]. Inspired by binary low density parity check (LDPC) codes, LDLC is defined by a sparse check matrix and applies the belief propagation for decoding, which achieves $\approx$ 0.5dB gap to the Poltyrev's bound at dimension $n = 100000$. The complexity of LDLC decoder is improved in [19] by reducing the number of Gaussian mixtures at variable nodes. For short to medium dimensions, good lattices with dimension up to 24 are summarized in [3], lattices constructed by binary codes are studied using construction D/D' based on Bose–Chaudhuri–Hocquenghem (BCH) codes [20], polar codes [21, 22] and LDPC codes [23, 24] with excellent error performance and low complexity.

Regarding lattice codes, the design consists of coding lattice and shaping lattice, which have different design purposes. The coding lattice aims to provide good error correction capability, for which the studies introduced in the previous paragraph apply to the design of coding lattices. On the other hand, codebook with lattice shaping achieves lower average power compared to the traditional quadrature amplitude modulation (QAM), which is known as the shaping gain. The shaping gain relates to the MSE of lattice quantization, with maximum value $\approx$ 1.53dB achieved by assuming dimension $n \to \infty$ and a spherical shaping region [25]. The shaping operation includes a decoding step of the shaping lattice, which might increase the complexity for encoding messages. A trade-off exists between the shaping gain and the complexity of shaping operation when designing the shaping lattice. A typical shaping strategy is to use a scaled coding lattice for shaping, known as the Voronoi shaping or self-similar shaping, for which the shaping gain of the coding lattice is achieved. However, the Voronoi shaping may not optimize the design of shaping lattice because: 1) the design purposes of coding lattice and shaping lattice are different, where coding lattice is to achieve low error rate but shaping lattice is to achieve low MSE of quantization; and 2) the Voronoi shaping might be too complex for encoding messages if the decoder of shaping lattice has high complexity. A novel technique called rectangular encoding is proposed in [26], which allows us to separate the design of coding lattice and shaping lattice. With the extra freedom on design lattice code, the shaping lattice can be selected appropriately to balance the trade-off between the shaping gain and the complexity of shaping operation to match requirements of actual systems.

This approach can be applied to design lattice codes with various shaping lattices. For example, applying E_8 lattice achieves 0.65dB shaping gain with very a low shaping complexity [25]. By allowing an increase of shaping complexity, gains with 1.03dB and up to 1.24dB can be achieved using Leech lattice [27] and convolutional code lattices [28], respectively.

Lattice-based network coding is another topic studied in this dissertation. Lattices have linearity, from which linear combinations of lattice points with integer coefficients are still in the constellation of the same lattice. The linearity of lattice codes is studied with respect to its group isomorphism property [26, 29]. The group isomorphism is important to implement compute-forward (CF) relaying [30], which is a novel lattice-based PLNC technique. For traditional relaying techniques, amplified-forward has low relaying latency while also amplifies the channel noise; decode-forward eliminates the influence of noise while having high complexity, particularly for a multiple access relay performing MUD. CF relaying assumes a multiple access relay. Instead of finding individual users' messages as decode-forward, CF relaying exploits the group structure of lattice codes and aims to decode a linear combination of users' messages with integer coefficients using a single user decoding without MUD. Similar to single user transmission, lattice decoder for CF relaying has decoding coefficients, which are a scaling factor α and an integer coefficient vector $\mathbf{a}$, denoted as $\{\mathbf{a}, \alpha\}$. The relay transfers the linear combination and the integer coefficient vector to the destination. Given sufficient number of linear combinations, the destination recovers users' messages by solving linear equations. In [30], Nazer and Gastpar showed that a network using CF relaying can transmit messages at rate as high as that indicated by the Poltyrev's bound. By exploiting the group structure of lattice codes in PLNC, CF framework gives a low complexity relaying strategy, from which relay can be designed to achieve low latency and high energy efficiency. There are extensions and variations of the CF framework, such as CF with unequal power allocation [31], compute-forward multiple access [32], MIMO compute-forward [33], integer-forcing linear receiver [34–36] and etc. Practical application design of CF framework, as well as code design, are studied in [37–40].

1.3 Motivations and contributions

1.3.1 Motivations

This research considers real-valued communications. Asymptotical analysis showed that lattice codes achieve the capacity of power constrained AWGN

channel with lattice decoding, if the received message is α_{MMSE}-scaled before decoding. However, the capacity-achieving analysis is only valid as dimension $n \to \infty$. In practical systems using finite dimensional lattice codes, the receiver would have a non-zero error probability, that is the α_{MMSE} may still fail decoding even when $R < C$. The α_{MMSE} is a real-valued continuous variable, which scales the channel noise and affects the decoding error performance, as studied in [12–14, 41].

For single user transmission using lattice codes, assume some unlucky noise realization that failed decoding using α_{MMSE}. A conventional receiver requests a retransmission, increasing the latency, which could be more significant than a local decoding process. Since α_{MMSE} is a real-valued continuous variable, an intuition is that, there might be a chance to correctly decode the message using other scaling factors. This motivates us to study a decoding strategy that allows the receiver to change value of the decoding coefficient to retry decoding process, in order to improve the error rate compared to the conventional one-shot decoder using α_{MMSE} only and potentially reduce the end-to-end latency by avoiding retransmission. This is also applied to CF relaying scenario using a coefficient set $\{\mathbf{a}, \alpha\}$. The retry decoding allows the receiver to change the value of α and/or $\mathbf{a}$ to improve the error rate of the linear combination for CF relaying.

In order to implement retry decoding, a lattice construction is studied which provides physical layer error detection ability and is applicable to both single user case and CF relaying. Another reason motivated us to study this lattice construction relies on the CF relaying. Since CF relaying works at physical layer and a linear combination contains multiple users' messages, a stand-alone relay is not able to decode individual users' messages from a single linear combination for error detection, due to the underdetermination, and may forward erroneously decoded messages into network causing decoding failure at the destination. This lattice construction is functional at physical layer to detect errors from linear combinations directly at a stand-alone relay even without the knowledge of individual users' messages.

Additionally, for future communications, low latency becomes more and more important in scenarios, such as the URLLC suggested in the 5G standard. As introduced in the background, short block codes can achieve low decoding latency so that is suitable for the URLLC scenario, e.g. the short block length polar codes for control channels in the 5G standard. The study of small to medium dimensional lattice design connects lattices to future communications with low decoding latency.

1.3.2 Contributions

The goal of this work is to give a simple design of error correcting codes and decoding scheme to **improve the error performance** for reliable communications, while **reduce the transmission latency**. The contributions of this work is divided into 3 parts and summarized in this subsection. First, the retry decoding scheme is described for power constrained communications using single user transmission scenario, which is also denoted as SU scenario in following chapters, and CF relaying scenario, respectively. In this part, a genie-aided error detection is assumed, where a genie is associated at the receiver checking the validity of decoding results. Note that the genie only tells whether the decoding result is correct or not without telling what the true message is. For the single user transmission scenario, AWGN channel is assumed and the contributions are as follows.

1. A retry decoding scheme for single user transmission is introduced using a decoding coefficient list of α's.
2. A recursive algorithm is proposed to generate a finite-length decoding coefficient list, maximizing the probability of correct decoding given all previous α's failed. So α's with high probability of correctly decode would have high priority to be tested at the receiver.
3. A lower bound and an effective sphere estimate on word error probability are derived using an exhaustive search decoder by extending the finite-length coefficient list to all real numbers, which considers the covering sphere and effective sphere respectively. The formula is simple, where only a single integral is involved.
4. The retry decoding, the lower bound and the effective sphere estimate are evaluated using low dimensional lattice codes with known covering radius. By appropriately selecting the decoding coefficient list, word error rate (WER) of the exhaustive search decoder can be approached by only using three decoding attempts so that additional complexity of retry could be low.
5. Discussions are given regarding to the tightness of the lower bound, the accuracy of the effective sphere estimate and the relationship between the benefit of retry decoding and the dimension of lattice codes.

For the CF relaying scenario, a multiple access channel with fading is assumed and the contributions are as follows.

1. Two retry decoding schemes are considered for CF relaying, that is: scheme 1 changes the decoding coefficient set $\{\mathbf{a}, \alpha\}$ for retry decoding; scheme 2 fixes $\mathbf{a}$ and only changes α. Approaches for finding the decoding coefficient list for the two schemes are also introduced.

2. The lower bound and the effective sphere estimate discovered for the SU scenario can be applied to the scheme 2.
3. A comparison is given for two retry schemes, by assuming fixed channel with different values. It is shown that, if the channel can be easily approximated by an integer vector, scheme 1 only achieves a modest gain and scheme 2 achieves the larger gain; otherwise, scheme 1 achieves the larger gain due to additional freedom on changing $\mathbf{a}$.
4. Simulation results for random channel realizations are given using polar lattice codes [22] with dimension $n = 128, 256$ to show the benefit of retry decoding in practical scenarios. Gains of 1.51dB and 1.18dB are achieved at $n = 128, 256$ respectively, with maximum two decoding attempts.

Second, a lattice construction with physical layer error detection ability is proposed to enable the retry decoding in practical systems. The proposed lattice construction and related results are summarized as follows.

1. The error detection is implemented by embedded cyclical redundancy check (CRC) code into the least significant bits (LSB) of the lattice uncoded message, called the CRC-embedded lattice.
2. It is proven that the CRC-embedded lattice is indeed a lattice, therefore forms a lattice code. A shaping lattice design is given so that errors can be detected from linear combinations of a modified CF framework, which is the ICF scheme described in [42], without requiring the knowledge of individual users' messages.
3. CRC optimization is considered to balance the cost of CRC parity bits and error detection capability, including optimizing CRC polynomial and CRC length. It is shown that optimizing the CRC polynomial requires the structure of the underlying coding lattice and does not affect as significantly as optimizing the CRC length.
4. A semi-analytical approach on optimizing the CRC length is introduced to minimize the required signal-to-noise ratio (SNR) at a given target WER. The optimization first estimates the WER after retry decoding as a function of the CRC length. Then, the optimized CRC length is obtained form curves of estimated WER of various CRC lengths.
5. Discussions regarding the implementation of retry decoding with CRC-embedded lattice codes are given for: 1) trade-offs among lattice dimension, SNR penalty, code rate and the improvement of retry decoding; 2) a comparison of a conventional one-shot decoding receiver, which requests a retransmission when decoding failed.
6. At last, implementation examples demonstrate the benefit of retry decoding using CRC-embedded lattice codes in a SU scenario and CF

relaying network, respectively, with the optimized CRC length. In particular, gains of 1.31dB and 1.08dB are achieved using 128 and 256 dimensional polar lattice codes at WER being 10^{-5}, respectively, where only one additional decoding attempt is needed.

Third, we give a study on designing finite dimensional lattices by applying construction A to binary codes at medium dimensions. Lattices with medium dimensions are expected to have low decoding latency and could be considered for low latency communication scenario, such as in URLLC scenario in 5G systems. The related results are summarized as follows.

1. Binary codes with known minimum Hamming distance and codeword multiplicity, number of codeword at minimum weight, are considered for this design.
2. A truncated series, defined in Chapter 2, of construction A lattice is explicitly given, from which the truncated union bound can be compute to estimate the WER.
3. The best lattice is obtained to minimize the required volume-to-noise ratio (VNR) at a given target WER.
4. Design examples are given for 128 dimensional lattices using extended BCH (EBCH) codes and polar codes, where EBCH code gives the best construction A lattice among design examples. A comparison with classic design rules and construction D lattice from previous research are also given.

1.4 Organization

The topics and the corresponding main contents of chapters in this dissertation are summarized as follows.

Chapter 1 gives an introduction of this dissertation, including high-level background, motivation and contributions.

Chapter 2 is about lattices, which gives a review of basic definitions, some low dimensional well-known lattices and construction A/D lattices.

Chapter 3 is about lattice codes. The basic definition, encoding/decoding technique and transmissions scheme for SU scenario and multiple access scenarios are given.

Chapter 4 proposes the retry decoding schemes for SU scenario and CF relaying. This chapter also gives algorithms on finding decoding coefficient list, an analytical bound for the SU scenario and a comparison

between two different retry schemes for CF relaying. Error detection is assumed to be genie-aided in this chapter.

Chapter 5 proposes the construction of the CRC-embedded lattice/lattice codes. This chapter also gives a CRC optimization and discussions related to practical implementation. Actual CRC codes are applied for error detection in this chapter. Numerical results are given with optimized CRC for SU scenario and CF relaying, respectively.

Chapter 6 gives a construction A lattice design approach using binary codes with known minimum Hamming distance and codeword multiplicity for medium dimensions. Design examples are provided using EBCH codes and polar codes, along with comparison with classic design rules and previous research.

Chapter 7 gives a conclusion of this dissertation and discusses potential applications as future works.

1.5 Notation

This section gives the rules of notations in this dissertation. Scalar variables are denoted using italic font, e.g. code rate R and lattice dimension n; vectors are denoted using lower-case bold, e.g. message vector $\mathbf{x}, \mathbf{y}$; matrices are denoted using upper-case bold, e.g. generator matrix $\mathbf{G}$; sets are denoted using calligraphic font, e.g. codebook $\mathcal{C}$. Vectors are column vectors, unless stated otherwise. The cardinality of a set is denoted as $|\cdot|$. The integer and real number space are denoted using $\mathbb{Z}$ and $\mathbb{R}$, respectively. Subscripts of identity matrix $\mathbf{I}$ and all-zero matrix $\mathbf{0}$ indicate the size of matrices, e.g. n-by-n identity matrix $\mathbf{I}_n$ and k-by-$(n-k)$ all-zero matrix $\mathbf{0}_{k\times(n-k)}$.

Chapter 2

Lattices

This chapter introduces lattices as the basic to this dissertation. We start from basic definitions and related properties. Then, some well-known low-dimensional lattices are introduced, which are used in later chapters. At last, lattice constructions from linear block codes are introduced.

2.1 Definitions and operations

Definition 2.1: (Lattice) An n-dimensional lattice Λ is a discrete additive subgroup of the real number space $\mathbb{R}^n$.

By the definition of lattices, the properties of groups also hold for lattices, that is, for any $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \Lambda$,

- identity element: all zero vector $\mathbf{0} \in \Lambda$,
- inverse element: $-\mathbf{a} \in \Lambda$,
- associativity: $(\mathbf{a} + \mathbf{b}) + \mathbf{c} = \mathbf{a} + (\mathbf{b} + \mathbf{c})$,
- commutativity: $\mathbf{a} + \mathbf{b} = \mathbf{b} + \mathbf{a}$,
- closure: $\mathbf{a} + \mathbf{b} \in \Lambda$.

It is straightforward that for any $m_1, m_2 \in \mathbb{Z}$, we have $m_1\mathbf{a} + m_2\mathbf{b} \in \Lambda$. A lattice can be scaled by $k \in \mathbb{R} \setminus 0$. The scaled lattice $k\Lambda$ satisfies that $k\mathbf{a} \in k\Lambda$. Lattices can also be defined for complex numbers [3], while in this dissertation, we consider the real number lattices.

2.1.1 Generator matrix and check matrix

A lattice Λ is commonly defined by a generator matrix $\mathbf{G}$, with $\text{Rank}(\mathbf{G}) = n$. Let $\mathbf{g}_i \in \mathbb{R}^n$ be a column basis vector:

$$\mathbf{g}_i = \begin{bmatrix} g_{1,i} \\ g_{2,i} \\ \vdots \\ g_{n,i} \end{bmatrix}. \tag{2.1}$$

The generator matrix is formed by n linearly independent basis vectors as $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_n]$. A lattice point $\mathbf{x} \in \Lambda$ is an integer linear combination of basis vectors as:

$$\mathbf{x} = b_1\mathbf{g}_1 + b_2\mathbf{g}_2 + \cdots + b_n\mathbf{g}_n, \tag{2.2}$$

where $b_1, b_2, \cdots, b_n \in \mathbb{Z}$. Denote the information vector $\mathbf{b}$ as:

$$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}, \tag{2.3}$$

a lattice is defined in the following form:

$$\Lambda = \{\, \mathbf{Gb} \mid \mathbf{b} \in \mathbb{Z}^n \,\}. \tag{2.4}$$

For a scaled lattice $k\Lambda$, we have $\mathbf{G}_{k\Lambda} = k\mathbf{G}$.

A lattice Λ has a check matrix, which is the inverse of the generator matrix, defined as $\mathbf{H} = \mathbf{G}^{-1}$. For $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}$ is a lattice point of Λ if and only if $\mathbf{Hx}$ is a vector of integers.

Example 2.1: Figure 2.1 illustrates an example of a 2-dimensional lattice, also known as the A_2 lattice or the hexagonal lattice, with a generator matrix

$$\mathbf{G} = \begin{bmatrix} \sqrt{3}/2 & 0 \\ 1/2 & 1 \end{bmatrix}. \tag{2.5}$$

It is also possible to define a lattice using an $m \times n$ generator matrix with $m \geq n$ and $\text{Rank}(\mathbf{G}) = n$. In this case, lattice points $\mathbf{x} \in \Lambda$ are located in an n-dimensional hyperplane inside an m-dimensional space, as shown in Example 2.2. For simplicity, we assume $m = n$ in this dissertation.

Example 2.2: A scaled A_2 lattice can be generated by

$$\mathbf{G} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & -1 \end{bmatrix}. \tag{2.6}$$

The lattice point $\mathbf{x}$ generated by (2.6) are located at a 2-dimensional plane in the 3-dimensional space satisfying $x_1 + x_2 + x_3 = 0$.

The generator matrix of a lattice is not unique. A simple example to justify this is $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_n]$ and $\mathbf{G}' = [\mathbf{g}_n, \mathbf{g}_{n-1}, \cdots, \mathbf{g}_1]$ generate the

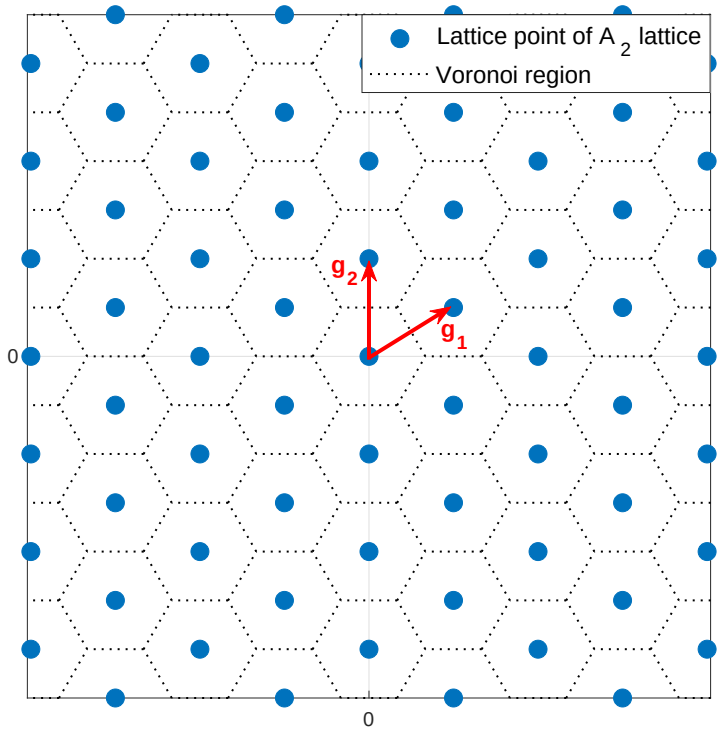


Figure 2.1: Example of A_2 lattice.

same lattice. In general, for Λ with a generator matrix $\mathbf{G}$, given an n-dimensional unimodular matrix $\mathbf{W}$, the lattice Λ_I generated by $\mathbf{G}_I = \mathbf{GW}$ is identical to Λ. We can also find an equivalent lattice Λ_O (not identical) by orthogonal transformation with generator matrix $\mathbf{G}_O = \mathbf{QG}$, where $\mathbf{Q}$ is an n-dimensional orthogonal matrix. In the 2-dimensional case, the orthogonal transformation can be viewed as rotation. For example, using

$$\mathbf{Q} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \tag{2.7}$$

the resulting lattice Λ_O is rotated from Λ by an angle θ.

In this dissertation, we prefer the generator matrix having a lower triangular structure. For lattices with a full generator matrix $\mathbf{G}$, a lower triangular generator matrix $\mathbf{G}_L$ of its equivalent lattice can be obtained by the QL decomposition $\mathbf{G} = \mathbf{QG}_L$, where $\mathbf{Q}$ is an orthogonal matrix.

2.1.2 Fundamental regions

For each lattice point $\mathbf{x} \in \Lambda$, a fundamental region $\mathcal{F}$ is associated and defined as follows.

Definition 2.2: (Fundamental region) Given a lattice Λ, a region $\mathcal{F}$ is a fundamental region of Λ if: for any distinct lattice points $\mathbf{x}_i \neq \mathbf{x}_j$, $(\mathcal{F}+\mathbf{x}_i) \cap (\mathcal{F}+\mathbf{x}_j) = \emptyset$ and the real number space can be covered as $\mathbb{R}^n = \bigcup_{\mathbf{x}\in\Lambda} \mathcal{F}+\mathbf{x}$.

From the definition, each $\mathcal{F}$ is associated to exactly one lattice point, therefore it is valid to denote it as $\mathcal{F}(\mathbf{x})$. However, the shape of the fundamental region of a lattice is not unique. Here, we introduce 2 important fundamental regions, which are well-defined and will be used in later chapters: the Voronoi region $\mathcal{V}(\mathbf{x})$ and the hypercube region $\mathcal{H}(\mathbf{x})$.

Definition 2.3: (Voronoi region) The Voronoi region $\mathcal{V}(\mathbf{x})$ is the set of $\mathbf{y} \in \mathbb{R}^n$ such that $\mathbf{y}$ is closer to $\mathbf{x}$ than any other lattice points, given as:

$$\mathcal{V}(\mathbf{x}) = \left\{ \mathbf{y} \in \mathbb{R}^n \mid \left\| \mathbf{y}-\mathbf{x} \right\|^2 \leq \|\mathbf{y}-\mathbf{x}'\|^2,\ \mathbf{x}' \in \Lambda \setminus \mathbf{x} \right\}. \tag{2.8}$$

Definition 2.4: (Hypercube region) For lattices having a triangular generator matrix with diagonal elements $g_{1,1}, g_{2,2} \cdots, g_{n,n}$, the hypercube region can be defined as:

$$\mathcal{H}(\mathbf{x}) = \left\{ \mathbf{y} \in \mathbb{R}^n \mid -\frac{g_{i,i}}{2} \leq y_i < \frac{g_{i,i}}{2}, i = 1, \cdots, n \right\}. \tag{2.9}$$

Example 2.3: The hexagonal region illustrated in Figure 2.1 is the Voronoi region of A_2 lattice and the corresponding hypercube region is illustrated in Figure 2.2.

Although there exist different shapes of fundamental region, the volume of different fundamental regions of a lattice Λ is a constant, which is also referred as the volume of the lattice as $V(\Lambda)$.

Definition 2.5: (Volume of lattice) The volume of a lattice Λ is the absolute value of determinant of its generator matrix $\mathbf{G}$:

$$V(\Lambda) = |\det(\mathbf{G})|. \tag{2.10}$$

2.1.3 Spheres related to lattices

The fundamental region defined above helps us to study the geometric properties of lattice. For example, the Voronoi region $\mathcal{V}(\mathbf{x})$ can be applied as the decoding region of a closest point lattice decoder. However, the shape of $\mathcal{V}(\mathbf{x})$ is typically irregular, thus is difficult to analyze. Instead, a series of n-spheres centered at $\mathbf{x}$ are defined for analysis.

Definition 2.6: (Packing sphere) The packing sphere $\mathcal{S}_p(\mathbf{x})$ with radius r_p and center $\mathbf{x}$ is the sphere of maximal radius that is inside the Voronoi region, i.e. $\mathcal{S}_p(\mathbf{x}) \subseteq \mathcal{V}(\mathbf{x})$.

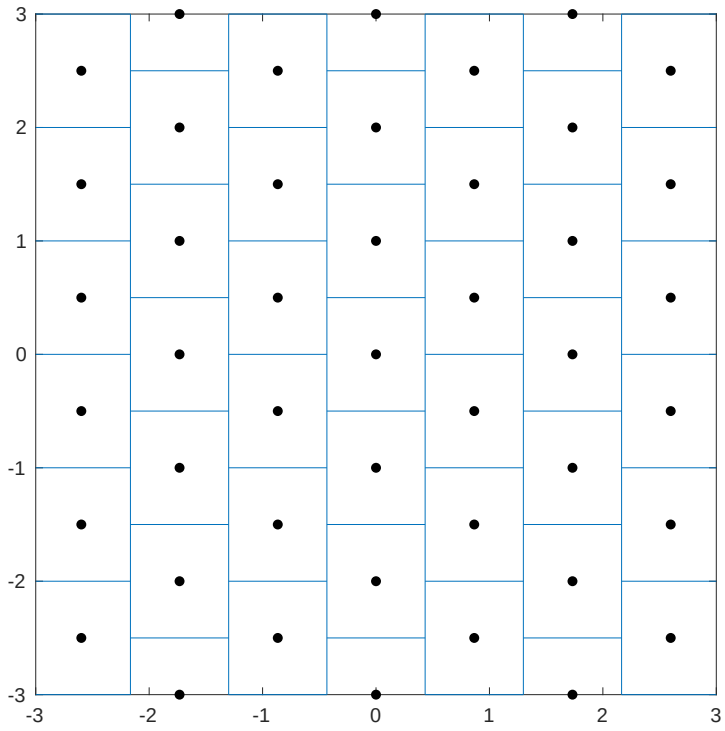


Figure 2.2: Hypercube region of A_2 lattice.

Definition 2.7: (Covering sphere) The covering sphere $\mathcal{S}_c(\mathbf{x})$ with radius r_c and center $\mathbf{x}$ is the sphere of minimal radius that covers the whole Voronoi region $\mathcal{V}$, i.e. $\mathcal{V}(\mathbf{x}) \subseteq \mathcal{S}_c(\mathbf{x})$.

Definition 2.8: (Effective sphere) The effective sphere $\mathcal{S}_e(\mathbf{x})$ with radius r_e and center $\mathbf{x}$ is the sphere with volume $V(\mathcal{S}_e(\mathbf{x}))$ being equal to the volume of the Voronoi region $\mathcal{V}(\mathbf{x})$, i.e. $V(\mathcal{S}_e(\mathbf{x})) = V(\mathcal{V}(\mathbf{x}))$. The volume of the n-sphere is given as

$$V(\mathcal{S}_e) = \frac{\pi^{n/2} r_e^n}{\Gamma(\frac{n}{2}+1)}, \tag{2.11}$$

where $\Gamma(\cdot)$ is the gamma function.

Geometrically, we have the following relationship:

- $\mathcal{S}_p(\mathbf{x}) \subseteq \mathcal{V}(\mathbf{x}) \subseteq \mathcal{S}_c(\mathbf{x})$,
- $r_p \leq r_e \leq r_c$,

When $n = 1$, equalities hold as the special case; for any finite $n > 1$, the $\subseteq$ and $\leq$ should be replaced by $\subset$ and $<$, respectively. Note that, as $n \to \infty$,

the asymptotic value of r_p is known to be strictly less than r_e, on the other hand $r_e \to r_c$ [25].

Example 2.4: Figure 2.3 gives an example to show the relationship among packing sphere $\mathcal{S}_p(\mathbf{x})$ (black solid line), covering sphere $\mathcal{S}_c(\mathbf{x})$ (black dashed line) and effective sphere $\mathcal{S}_e(\mathbf{x})$ (black dash-dotted line) with respect to a Voronoi region (red solid line) of A_2 lattice.

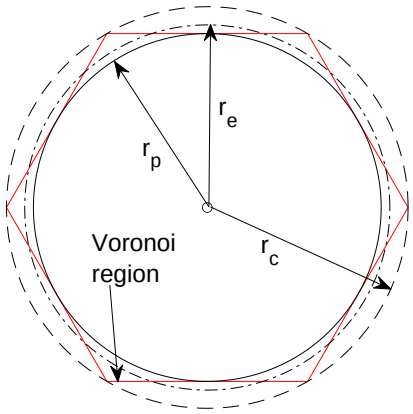


Figure 2.3: Relationship among $\mathcal{S}_p(\mathbf{x})$, $\mathcal{S}_c(\mathbf{x})$ and $\mathcal{S}_e(\mathbf{x})$.

$\mathcal{S}_p(\mathbf{x})$, $\mathcal{S}_c(\mathbf{x})$ are also related to the classic sphere packing and covering problems. These problems are not studied in this dissertation and details can be found in [3] and [25].

2.1.4 Lattice quantization and modulo

A lattice quantizer can be defined with respect to a fundamental region. In this dissertation, the Voronoi region is considered, from which the lattice quantizer $Q_\Lambda(\cdot)$ is also the lattice decoder, denoted as $DEC_\Lambda(\cdot)$.

Definition 2.9: (Lattice quantizer) Let $\mathbf{y} \in \mathbb{R}^n$. Lattice quantizer $Q_\Lambda(\cdot)$ finds the closest lattice point of $\mathbf{y}$ and is defined as

$$\hat{\mathbf{y}} = Q_\Lambda(\mathbf{y}) = \arg\min_{\mathbf{x}\in\Lambda} \|\mathbf{y} - \mathbf{x}\|^2 \tag{2.12}$$

The modulo operation for lattices is defined as follows.

Definition 2.10: (Lattice Modulo) Given $\mathbf{y} \in \mathbb{R}^n$, the modulo operation $\mathrm{mod}\Lambda$ is defined as:

$$\mathbf{y}_{mod} = \mathbf{y} \bmod \Lambda = \mathbf{y} - Q_\Lambda(\mathbf{y}) \tag{2.13}$$

The modulo operation gives the quantization error of $\mathbf{y}$ with respect to a lattice quantizer Q_Λ. This can be applied to encode lattice codes introduced in Chapter 3.

2.1.5 Lattices for communications

Lattices can be applied to communications as error correcting codes or constellation shaping. In this section, we introduce related definitions of lattices for communications.

2.1.5.1 Squared minimum distance

Since lattices are defined in the real number space, the Euclidean norm metric is considered instead of the Hamming distance metric, from which the squared minimum distance of a lattice Λ is

$$d_{min}^2 = \min_{\mathbf{x}\in\Lambda\setminus\mathbf{0}} \|\mathbf{x}\|^2. \tag{2.14}$$

It is noticed the packing radius is $r_p = d_{min}/2$.

2.1.5.2 Theta series

The theta series of a lattice is defined as the weight enumerator function of lattice points $\mathbf{x} \in \Lambda$ using the squared length in the Euclidean space.

Definition 2.11: (Theta series) Let $\tau_{d_i^2}$ be the number of $\mathbf{x} \in \Lambda$ having $\|\mathbf{x}\|^2 = d_i^2$. The theta series of Λ is

$$\theta = 1q^0 + \sum_{i=1}^{\infty} \tau_{d_i^2} q^{d_i^2}, \tag{2.15}$$

where q is regarded as a dummy variable.

Since a lattice has infinite constellation, the theta series also has infinite number of terms. In practical analysis, a truncated theta series is considered, defined as follows.

Definition 2.12: (Truncated theta series) Let $\tau_{d_i^2}$ be the number of $\mathbf{x} \in \Lambda$ having $\|\mathbf{x}\|^2 = d_i^2$. A truncated theta series of Λ with finite m terms is

$$\theta' = 1q^0 + \sum_{i=1}^{m} \tau_{d_i^2} q^{d_i^2}, \tag{2.16}$$

where q is regarded as a dummy variable.

2.1.5.3 Channel model

For communications using lattices, the additive white Gaussian noise (AWGN) channel is considered, such that the channel output is

$$\mathbf{y} = \mathbf{x} + \mathbf{n}, \tag{2.17}$$

where $\mathbf{x} \in \Lambda$ and a Gaussian noise $\mathbf{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ with equal variance σ^2 for all dimensions.

2.1.5.4 Volume-to-noise ratio

By Definition 2.1, a lattice Λ has an infinite constellation. The conventional SNR, used for power constrained communications, is not applicable, since the average power goes to infinity. Instead, the volume-to-noise ratio (VNR) is defined to measure error performance for transmission using lattices.

Definition 2.13: (VNR) Let Λ have volume $V = |\det(G)|$ and σ^2 be the per-dimensional noise power. The volume-to-noise ratio is

$$\text{VNR} = \frac{V^{2/n}}{2\pi e\sigma^2}. \tag{2.18}$$

The definition normalized the volume by the dimension n, therefore it gives a fair comparison of error performance among lattices with different dimensions. The required VNR to achieve a fixed P_e depends on the structure of lattice but is invariant to the scaling of the lattice, that is Λ and $k\Lambda$ achieve a same VNR vs. P_e performance. In this dissertation, VNR is commonly given in decibels (dB) as $\text{VNR(dB)} = 10\log_{10}\text{VNR}$.

It has been shown that lattices achieve unconstrained Gaussian channel capacity [9]. The definition of VNR can be seen as the distance to the Poltyrev limit. Asymptotically, for any fixed error rate P_e, there exists an ensemble of lattices that can achieve P_e with $\text{VNR} \to 1$ (or 0 dB) as $n \to \infty$. This is known as the goodness for AWGN (or AWGN-goodness), see also at [8, 25].

2.1.5.5 Normalized second moment

The second moment of a region $\mathcal{F}$ is defined as:

$$\frac{1}{nV(\mathcal{F})}\int_{\mathcal{F}} \|\mathbf{x}\|^2\, d\mathbf{x}. \tag{2.19}$$

The second moment can be used to find the average quantization error for a lattice by using the 0-centered Voronoi region $\mathcal{V}$. However, the value of second moment depends on the scaling of the lattice.

Definition 2.14: The normalized second moment (NSM) of lattice Λ is defined as:

$$G(\Lambda) = \frac{1}{nV(\mathcal{V})^{1+\frac{2}{n}}} \int_{\mathcal{V}} \|\mathbf{x}\|^2 \, d\mathbf{x}, \tag{2.20}$$

The value of NSM only depends on the structure of the lattice but is invariant to the scaling, which gives a scaling-independent measurement of quantization error for lattices. In general, computing the integral in (2.20) is hard due to the shape of $\mathcal{V}$. The NSM can be estimated using Monte-Carlo method. Given a sufficient large number N, generate i.i.d. samples $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$, which are uniformly distributed over the 0-centered Voronoi region $\mathcal{V}$, then the NSM is estimated by:

$$G(\Lambda) \approx \frac{1}{nNV(\mathcal{V})^{1+\frac{2}{n}}} \sum_{i=1}^{N} V(\mathcal{V}) \cdot \|\mathbf{x}_i\|^2, \tag{2.21}$$

The NSM is lower bounded by that of an n-sphere $\mathcal{B}_n$ [25, Chapter 3, 7] as

$$G(\Lambda) \geq G(\mathcal{B}) = \frac{\Gamma(n/2+1)^{2/n}}{\pi(n+2)}. \tag{2.22}$$

The value of $G(\mathcal{B})$ asymptotically approaches

$$\lim_{n\to\infty} G(\mathcal{B}) = \frac{1}{2\pi e}. \tag{2.23}$$

The inequality in (2.21) is obtained by, assuming $\mathcal{V}$ and $\mathcal{B}_n$ with same volume, for any $\mathbf{x}_1 \in \mathcal{V} \setminus \mathcal{B}_n$ and $\mathbf{x}_2 \in \mathcal{B}_n \setminus \mathcal{V}$, their Euclidean norm satisfies $\|\mathbf{x}_1\|^2 > \|\mathbf{x}_2\|^2$. The asymptotically value of $G(\mathcal{B})$ in (2.23) is obtained by the Stirling approximation. It is referred to as goodness for quantization (or a good quantizer) if the lattice satisfies $\lim_{n\to\infty} G(\Lambda) = \frac{1}{2\pi e}$, for which [8, 43] show the existence of such lattices.

Lattices can also be used for shaping the constellation to achieve a lower transmission power compared to conventional modulation schemes, such QAM. A shaping gain can be achieved and is defined as the ratio of NSM between the lattice and an integer lattice Z_n, with $G(Z_n) = 1/12$, which is a constant and independent of the dimension.

Definition 2.15: (Shaping gain) The shaping gain of a lattice Λ is defined as

$$\gamma(\Lambda)(\text{dB}) = 10\log_{10}\frac{G(Z_n)}{G(\Lambda)} = 10\log_{10}\frac{1}{12G(\Lambda)}. \tag{2.24}$$

Asymptotically, $\lim_{n\to\infty}\gamma(\Lambda)(\text{dB}) = 1.53(\text{dB})$ by considering a spherical shaping region.

2.1.5.6 Sub-lattice and coset decoding

Definition 2.16: (Sub-lattice) Given a lattice Λ, if another lattice Λ' satisfies $\Lambda' \subseteq \Lambda$, then Λ' is a sub-lattice of Λ.

Definition 2.17: (Lattice coset) Given a lattice Λ and a sub-lattice $\Lambda' \subseteq \Lambda$, a lattice coset is defined as the set $\{\, \mathbf{x} + \Lambda' \,\}$, where $\mathbf{x} \in \Lambda$ is the coset leader.

It is clear that for given Λ and a sub-lattice Λ', there only exists a finite number of distinct lattice cosets. Let $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_I\}$ be all distinct coset leaders, then Λ can be expressed as

$$\bigcup_{\mathbf{x}\in\mathcal{X}} (\mathbf{x} + \Lambda'). \tag{2.25}$$

Note that the coset decomposition of a lattice could be difficult to find in general. Fortunately, the coset decomposition for some low dimensional lattices are known as we summarize in the next section.

Coset decoding can be applied if there is no efficient lattice decoder for Λ, but exists an efficient lattice decoder of the sub-lattice Λ'. Let $\mathbf{y} \in \mathbb{R}^n$ and a lattice decoder of Λ' be $DEC_{\Lambda'}$, the coset decoding of Λ first finds

$$\hat{\mathbf{x}}_i = DEC_{\Lambda'}(\mathbf{y} - \mathbf{x}_i) + \mathbf{x}_i, \tag{2.26}$$

for $\mathbf{x}_i \in \mathcal{X}$; then gives decision by finding the $\hat{\mathbf{x}}_i$ which is the closest to $\mathbf{y}$ as:

$$\hat{\mathbf{x}} = \arg\min_i \|\mathbf{y} - \hat{\mathbf{x}}_i\|. \tag{2.27}$$

2.2 Well-known low dimensional lattices

In this section, some well-known low dimensional lattices that used in later chapters are introduced. Low dimensional lattices are well-studied, which have many known mathematical properties, such as minimum distance, covering radius and etc. A generator matrix, squared minimum distance and an optimal decoding algorithm are introduced as below. Most of the results in this section are based on [3].

2.2.1 Z_n lattice

The n-dimensional Z_n lattice, also known as the integer lattice, is generated by the identity matrix $\mathbf{I}_n$. We distinguish the integer lattice as Z_n and the n-dimensional integer space as $\mathbb{Z}^n$. Each lattice point $\mathbf{x} \in Z_n$ is a vector of integers. The squared minimum distance is $d^2_{min} = 1$. For $\mathbf{y} \in \mathbb{R}^n$, the optimal decoding algorithm is simply the component-wised rounding as:

$$\hat{\mathbf{x}} = \lfloor \mathbf{y} \rceil. \tag{2.28}$$

2.2.2 D_n lattice

The n-dimensional D^n lattice consists of points that satisfies:

$$D_n = \left\{ \mathbf{x} \in \mathbb{Z}^n \mid \sum_{i=1}^{n} x_i \bmod 2 = 0 \right\}. \tag{2.29}$$

From the definition, a generator matrix of D^n lattice can be given as:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 1 & 1 & 1 & \cdots & 1 & 2 \end{bmatrix}. \tag{2.30}$$

The squared minimum distance of the D_n lattice by (2.30) is $d^2_{min} = 2$, independent of the dimension. The optimal decoding algorithm of D_n lattices is given in [44] as shown in Algorithm 2.1.

2.2.3 E_8 lattice

The E_8 lattice is an 8-dimensional lattice with a generator matrix:

$$\mathbf{G} = \begin{bmatrix} 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}. \tag{2.31}$$

Algorithm 2.1: D_n lattice decoder, $DEC_{D_n}(\cdot)$

Input: $\mathbf{y}$
Output: $\hat{\mathbf{x}}$

$\hat{\mathbf{x}}_1 = \lfloor \mathbf{y} \rceil$;
$\hat{\mathbf{x}}_2 = \hat{\mathbf{x}}_1$;
Find $i = \arg\max |\hat{x}_{2,i} - y_i|$;
if $\hat{x}_{2,i} - y_i > 0$ **then**
  $\hat{x}_{2,i} = \hat{x}_{2,i} - 1$;
else
  $\hat{x}_{2,i} = \hat{x}_{2,i} + 1$;
end
if $\sum_i \hat{x}_{1,i} \bmod 2 = 0$ **then**
  $\hat{\mathbf{x}} = \hat{\mathbf{x}}_1$;
else
  $\hat{\mathbf{x}} = \hat{\mathbf{x}}_2$;
end

The squared minimum distance of E_8 lattice by (2.31) is $d^2_{min} = 2$.

The optimal decoding algorithm of E_8 lattice is given as follows. It has been found that E_8 lattice consists of 2 cosets of D_8 lattice with coset leaders $[0, 0, \cdots, 0]^T$ and $[1/2, 1/2, \cdots, 1/2]^T$ [3], given as

$$E_8 = D_8 \cup \left(D_8 + [1/2, 1/2, \cdots, 1/2]^T\right). \tag{2.32}$$

The optimal decoding algorithm of the E_8 lattice uses $DEC_{D_8}(\cdot)$ given in Algorithm 2.1 [44] as shown in Algorithm 2.2.

Algorithm 2.2: E_8 lattice decoder, $DEC_{E_8}(\cdot)$

Input: $\mathbf{y}$
Output: $\hat{\mathbf{x}}$

$\hat{\mathbf{x}}_1 = DEC_{D_8}(\mathbf{y})$;
$\hat{\mathbf{x}}_2 = DEC_{D_8}(\mathbf{y} - \frac{1}{2}) + \frac{1}{2}$;
if $\|\hat{\mathbf{x}}_1 - \mathbf{y}\| < \|\hat{\mathbf{x}}_2 - \mathbf{y}\|$ **then**
  $\hat{\mathbf{x}} = \hat{\mathbf{x}}_1$;
else
  $\hat{\mathbf{x}} = \hat{\mathbf{x}}_2$;
end

2.2.4 BW_{16} lattice

The Barnes-Wall lattices is class lattices with dimensional 2^k for some positive integer k [45]. In this dissertation, the 16-dimensional Barnes-Wall lattice is considered, denoted as BW_{16} lattice. A generator matrix of BW_{16} lattice is given in (A.1) at Appendix A.1. The squared minimum distance of BW_{16} lattice by (A.1) is $d^2_{min} = 8$. The BW_{16} lattice can be decomposed to 32 cosets of D_{16} lattice. The matrix $\mathbf{C}$ of coset leaders is given in (A.2) in Appendix A.1. The optimal decoding algorithm of the BW_{16} lattice is given in Algorithm 2.3.

Algorithm 2.3: BW_{16} lattice decoder, $DEC_{BW_{16}}(\cdot)$

Input: $\mathbf{y}$
Output: $\hat{\mathbf{x}}$

$e_{min} = \infty$;
for $i = 1:1:32$ **do**
  $\mathbf{c} = \mathbf{C}(:, i)$;
  $\hat{\mathbf{x}}' = DEC_{D_{16}}(\mathbf{y} - \mathbf{c}) + \mathbf{c}$;
  $e_{temp} = \|\hat{\mathbf{x}}' - \mathbf{y}\|$;
  if $e_{temp} < e_{min}$ **then**
    $e_{min} = e_{temp}$;
    $\hat{\mathbf{x}} = \hat{\mathbf{x}}_i$;
  end
end

2.2.5 Leech lattice

The Leech lattice is a 24-dimensional lattice, which can be constructed in many ways [46–48], [3, Chapter 24]. In this dissertation, the construction in [48] is considered. A generator matrix is given in (A.3) at Appendix A.2, where the $\frac{1}{\sqrt{8}}$ appeared in standard form in [3] is dropped to give an integer generator matrix. By this construction, a 24-dimensional vector of a Leech lattice point can be divided into three 8-dimensional vectors as

$$\mathbf{x} = [\mathbf{e}_1 + \mathbf{a} + \mathbf{t}, \mathbf{e}_2 + \mathbf{b} + \mathbf{t}, \mathbf{e}_3 + \mathbf{c} + \mathbf{t}]^T, \tag{2.33}$$

where $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ are arbitrary vector in the $4E_8$ lattice, $\mathbf{a}, \mathbf{b}$ are arbitrary vector from $\mathbf{A}$ given in (A.4) at Appendix A.2 and $\mathbf{t}$ is an arbitrary vector in the $\mathbf{T}$ given at (A.5) in Appendix A.2. The vector $\mathbf{c} \in \mathbf{A}$ is constrained

by

$$\mathbf{a} + \mathbf{b} + \mathbf{c} \bmod 4E_8 = 0. \tag{2.34}$$

The Leech lattice can be decoded using coset decoding based on a lattice with generator matrix

$$\mathbf{G}' = \begin{bmatrix} 4\mathbf{G}_{E8} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 4\mathbf{G}_{E8} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 4\mathbf{G}_{E8} \end{bmatrix} \tag{2.35}$$

and coset leaders specified by all possible combinations of $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and $\mathbf{t}$. An implementation of the optimal Leech lattice decoder based on this construction is given in Algorithm 2.4. There also exists other decoding algorithms with lower complexity, such as [49, 50] (optimal), [51, 52] (non-optimal, bounded distance).

2.3 Lattice constructions

In higher dimensions, it is more common to construct lattices using codes. The Construction A and Construction D are 2 efficient ways to construct lattice using linear block codes, respectively. In this chapter, we give the structure of Construction A/D lattices.

2.3.1 Construction A

Construction A [3, 25] forms a lattice by lifting a binary code to real number space.

Definition 2.18: (Construction A lattice) For an (n, k) binary linear block code $\mathcal{C}_b$, a construction A lattice consists of all vectors of the form:

$$\Lambda_a = \mathbf{c} + 2\mathbf{z}, \mathbf{c} \in \mathcal{C}_b, \mathbf{z} \in \mathbb{Z}^n, \tag{2.36}$$

or equivalently $\Lambda_a = \mathcal{C}_b + 2\mathbb{Z}^n$.

This definition can be extended to q-ary linear block code $\mathcal{C}_q$, from which the construction A lattice is given as $\Lambda_a = \mathcal{C}_q + q\mathbb{Z}^n$, as known as modulo-$q$ lattice [25].

Let the n-by-k generator matrix of binary code have the structure $\mathbf{G}_c = \begin{bmatrix} \mathbf{G}_t \\ \mathbf{P} \end{bmatrix}$, where $\mathbf{G}_t$ is a lower triangular matrix. A generator matrix of the

Algorithm 2.4: Leech lattice decoder, $DEC_{\Lambda_{24}}(\cdot)$

Input: $\mathbf{y}$
Output: $\hat{\mathbf{x}}$

Decompose the received message into 3 vectors ;
$\mathbf{y}_1 = \mathbf{y}(1:8)$;
$\mathbf{y}_2 = \mathbf{y}(9:16)$;
$\mathbf{y}_3 = \mathbf{y}(17:24)$;

Quantize using $\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3$ individually ;
for possible vectors in $\mathbf{A}$ and $\mathbf{T}$;
Initialize $\mathbf{S}$ as a 3-dimensional matrix with size $(24, 16, 16)$;
for $aa = 1:1:16$ **do**
    for $tt = 1:1:16$ **do**
        $\mathbf{a} = \mathbf{A}(:, aa)$;
        $\mathbf{t} = \mathbf{T}(:, tt)$;
        $\mathbf{s}_1 = DEC_{E_8}((\mathbf{y}_1 - \mathbf{a} - \mathbf{t})/4) * 4 + \mathbf{a} + \mathbf{t}$;
        $\mathbf{s}_2 = DEC_{E_8}((\mathbf{y}_2 - \mathbf{a} - \mathbf{t})/4) * 4 + \mathbf{a} + \mathbf{t}$;
        $\mathbf{s}_3 = DEC_{E_8}((\mathbf{y}_3 - \mathbf{a} - \mathbf{t})/4) * 4 + \mathbf{a} + \mathbf{t}$;
        $\mathbf{S}(:, aa, tt) = [\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3]^T$;
    end
end

Reconstruct estimate of codeword for possible combinations of $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and $\mathbf{t}$;
Find the one closest to the received message ;
$e_{min} = \infty$;
for $aa = 1:1:16$ **do**
    for $bb = 1:1:16$ **do**
        $cc = \mathbf{C}_{table}(aa, bb)$;
        for $tt = 1:1:16$ **do**
            $\hat{\mathbf{x}}_1 = \mathbf{S}(1:8, aa, tt)$;
            $\hat{\mathbf{x}}_2 = \mathbf{S}(1:8, bb, tt)$;
            $\hat{\mathbf{x}}_3 = \mathbf{S}(1:8, cc, tt)$;
            $\hat{\mathbf{x}}' = [\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2, \hat{\mathbf{x}}_3]^T$;
            $e_{temp} = \|\hat{\mathbf{x}}' - \mathbf{y}\|$;
            if $e_{temp} < e_{min}$ **then**
                $e_{min} = e_{temp}$;
                $\hat{\mathbf{x}} = \hat{\mathbf{x}}'$;
            end
        end
    end
end

modulo-2 construction A lattice is given as [25]:

$$\mathbf{G}_a = \left[\begin{array}{c:c} \mathbf{G}_c & \mathbf{0} \\ & 2\mathbf{I}_{n-k} \end{array} \right]. \tag{2.37}$$

From (2.37), the volume of Λ_a is

$$V(\Lambda_a) = 2^{n-k}. \tag{2.38}$$

Due to the presence of the $2\mathbb{Z}^n$ lattice, the squared minimum distance of a construction A lattice is given by

$$d_{min}^2 = \min(4, d_c), \tag{2.39}$$

where d_c is the minimum Hamming distance of $\mathcal{C}_b$.

For asymptotically large dimension n, it has been proven that there exists construction A lattices, constructed from a random linear code over a prime field of growing size, can be simultaneously good for sphere packing, sphere covering, AWGN channel coding and quantization [8].

The following are examples of finite dimensional lattices using construction A. $2E_8$ lattice can be seen as a construction A lattice formed using the $(8, 4)$ extended Hamming codes. The low-density construction A (LDA) lattices is formed using a q-ary LDPC code, which achieves excellent error performance for dimensions $n \geq 1000$ [15]. It is also shown that LDA lattices are capacity-achieving when $n \to \infty$ [16]. In communications, construction A can be seen as a shifted constellation of the widely used binary phase shift keying (BPSK) [25, Chapter 8] or pulse-amplitude modulation (PAM) scheme, from which a connection from lattice coding to conventional systems is established, see [39, 53] for implementation examples.

The encoding and decoding scheme of construction A lattices are given in Chapter 6, where lattice design based on construction A is provided.

2.3.2 Construction D

Construction D, proposed in [54], forms a lattice using a family of nested binary codes. Assume linearly independent binary basis $\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_n \in \mathbb{F}_2^n$. Let $\mathbf{G}_i = [\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_{k_i}]$ generate a binary code $\mathcal{C}_i$ for $i = 0, 1, \cdots, a$ with integer $a > 1$ and $k_0 < k_1 < \cdots < k_a = n$, from which a family of nested binary codes $\mathcal{C}_0 \subseteq \mathcal{C}_1 \subseteq \cdots \subseteq \mathcal{C}_{a-1} \subseteq \mathcal{C}_a = \mathbb{F}_2^n$ is generated from $\mathbf{G}_0, \mathbf{G}_1, \cdots, \mathbf{G}_a$.

Definition 2.19: (Construction D lattice) Let $\psi(\cdot)$ be a mapping from $\mathbb{F}_2^n$ to $\mathbb{R}^n$, such that $0 \to 0, 1 \to 1$. A construction D lattice with a layers is

defined by

$$\Lambda_d = \sum_{i=0}^{a} 2^{i-1}\psi(\mathbf{G}_i)\mathbf{b}_i, \tag{2.40}$$

where $\mathbf{b}_i \in \{0,1\}^n$.

Since $\mathcal{C}_a$ is the universal code, there exists a bijective mapping between $\psi(\mathbf{G}_a)\mathbf{b}_a$ and $\mathbf{z} \in \mathbb{Z}^n$. The construction D lattice given in (2.40) can be equivalently given as

$$\Lambda_d = \left(\sum_{i=0}^{a-1} 2^{i-1}\psi(\mathbf{G}_i)\mathbf{b}_i\right) + 2^a\mathbf{z}. \tag{2.41}$$

In [55], some expressions related to construction D lattice are given, for which the condition on equivalency among them is discussed.

Construction D is also applied to design finite dimensional lattices for practical use, such as in [20–22,56]. Additionally, Barnes-Wall lattices can be generated by construction D [45, Section III]. For example the BW_{16} lattice consists of $\mathcal{C}_0 = (16,5), \mathcal{C}_1 = (16,15)$ Reed-Muller codes as its component codes.

The encoding and decoding scheme of construction D lattices follows [20, 22].

2.4 Summary and connection to other chapters

This chapter first introduced lattices of their definitions, geometric properties, operations and the connection to communications. Then we introduced some low dimensional lattices and constructions of lattice using binary codes.

Section 2.1 is the foundation of this dissertation, from which the definitions and operations are applied in all later chapters. The lattices given in Section 2.2 are mainly used in our studies for single user transmission case in Chapter 4 and Chapter 5. Construction D lattices are used to evaluate retry decoding for CF relaying scenario in Chapter 4 and Chapter 5, where medium dimensional lattices are considered, such as $n = 128, 256$. Construction A is considered for the design of finite dimensional lattices in Chapter 6.

Chapter 3

Lattice codes

In the previous chapter, lattices with infinite constellation are introduced. This chapter introduces lattice codes with finite constellation for power-constrained communications.

3.1 Nested lattice codes

Definition 3.1: (Nested lattice codes) Let two lattice Λ_c and Λ_s satisfy $\Lambda_s \subseteq \Lambda_c$ and thus form a quotient group Λ_c/Λ_s. A nested lattice code $\mathcal{C}$ is defined as the coset leader of quotient group Λ_c/Λ_s:

$$\mathcal{C} = \{ \mathbf{x} \bmod \Lambda_s \mid \mathbf{x} \in \Lambda_c \}. \tag{3.1}$$

Equivalently, $\mathcal{C}$ also can be defined as the interaction of Λ_c and fundamental region $\mathcal{F}$ of Λ_s.

$$\mathcal{C} = \Lambda_c \cap \mathcal{F} \tag{3.2}$$

The fine lattice Λ_c is called the coding lattice and the coarse lattice Λ_s is called the shaping lattice. For a nested lattice code $\mathcal{C} = \Lambda_c/\Lambda_s$ with generator $\mathbf{G}_c$ and $\mathbf{G}_s$, the following sub-lattice condition is satisfied.

Lemma 3.1: [25, Chapter 8] A nested lattice code $\mathcal{C} = \Lambda_c/\Lambda_s$ can be formed if and only if $\mathbf{M} = \mathbf{G}_c^{-1}\mathbf{G}_s$ is a matrix of integers.

Proof. Since Λ_s is a sub-lattice of Λ_c, for any $\mathbf{x} \in \Lambda_s$, it is also a lattice point of Λ_c. Therefore, for any $\mathbf{b}_s \in \mathbb{Z}^n$, exists $\mathbf{b}_c \in \mathbb{Z}^n$, such that

$$\mathbf{x} = \mathbf{G}_c\mathbf{b}_c = \mathbf{G}_s\mathbf{b}_s \tag{3.3}$$

$$\mathbf{b}_c = \mathbf{G}_c^{-1}\mathbf{G}_s\mathbf{b}_s. \tag{3.4}$$

This is satisfied if and only if $\mathbf{M} = \mathbf{G}_c^{-1}\mathbf{G}_s$ is a matrix of integers.

□

Example 3.1: Figure 3.1 give an example of a lattice code $\mathcal{C} = A_2/4A_2$. The coding lattice is the A_2 lattice and the shaping lattice is a scaled $4A_2$ lattice, with $\mathbf{G}_c = \begin{bmatrix} \sqrt{3}/2 & 0 \\ 1/2 & 1 \end{bmatrix}$ and $\mathbf{G}_s = \begin{bmatrix} 2\sqrt{3} & 0 \\ 2 & 4 \end{bmatrix}$. The red dashed line is the Voronoi region of $4A_2$ lattice and is the shaping region of this lattice code. For $\mathbf{x} \in A_2$ at the boundary of the shaping region, only half of them are included in the codebook for the fairness of tie-breaking.

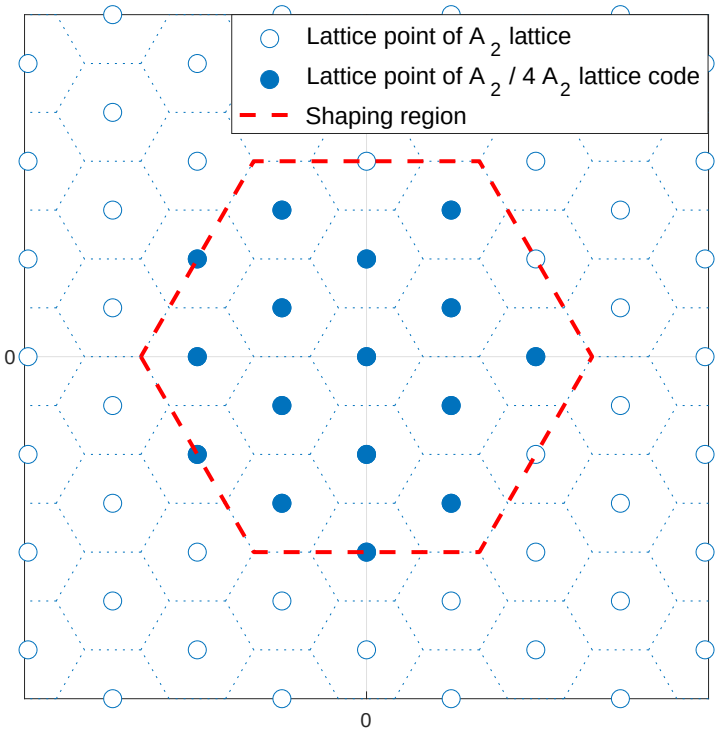


Figure 3.1: Example of A_2 lattice.

3.2 Encoding and decoding

Encoding of a lattice code $\mathcal{C} = \Lambda_c/\Lambda_s$ is to map an uncoded message $\mathbf{b} \in \mathbb{Z}^N$ to a codeword $\mathbf{x} \in \mathcal{C}$ as:

$$\mathbf{x} = ENC(\mathbf{b}) = \mathbf{G}_c\mathbf{b} \bmod \Lambda_s = \mathbf{G}_c\mathbf{b} - Q_{\Lambda_s}(\mathbf{G}_c\mathbf{b}). \tag{3.5}$$

The modΛ_s is referred as the shaping operation for encoding lattice codes and the $Q_{\Lambda_s}(\cdot)$ in (3.5) is a lattice quantizer of Λ_s.

In this dissertation, the rectangular encoding proposed in [26] are applied to design and encoding/decoding the lattice codes.

Definition 3.2: (Rectangular encoding) The lattice code $\mathcal{C}$ has a rectangular encoding if there exists $\mathbf{G}_c$ and positive integers $M_1, M_2, \cdots, M_n$ such that the function

$$\mathbf{x} = \mathbf{G}_c\mathbf{b} - Q_{\Lambda_s}(\mathbf{G}_c\mathbf{b}) \tag{3.6}$$

is a bijective mapping between the integers $b_i \in \{0, 1, \cdots, M_i - 1\}$, for $i = 1, 2, \cdots, n$, and a codeword $\mathbf{x} \in \mathcal{C}$.

Each dimension of the uncoded message $\mathbf{b}$ can be defined over different range independently by applying the rectangular encoding. The value $\log_2 M_i$ indicates the number of bits that can be carried at the i-th dimension ($M_i = 1$ means this dimension cannot carry information). A simple way to construct a lattice code using rectangular encoding is to use a diagonal matrix

$$\mathbf{M} = \text{diag}(M_1, M_2, \cdots, M_n), \tag{3.7}$$

and a generator matrix of the shaping lattice as $\mathbf{G}_s = \mathbf{G}_c\mathbf{M}$.

The message recovery process, also called indexing, is to find an estimate of the uncoded message as:

$$\hat{\mathbf{b}} = \text{Index}(\mathbf{x}). \tag{3.8}$$

From (3.5), we have

$$\begin{aligned} \hat{\mathbf{b}}' &= \mathbf{G}_c^{-1}\mathbf{x} && (3.9)\\ &= \hat{\mathbf{b}} - \mathbf{G}_c^{-1}Q_{\Lambda_s}(\mathbf{G}_c\mathbf{b}) \\ &= \hat{\mathbf{b}} - \mathbf{G}_c^{-1}\mathbf{G}_s\mathbf{s} \\ &= \hat{\mathbf{b}} - \mathbf{M}\mathbf{s}. && (3.10) \end{aligned}$$

where there exists $\mathbf{s} \in \mathbb{Z}^n$ such that $Q_{\Lambda_s}(\mathbf{G}_c\mathbf{b}) = \mathbf{G}_s\mathbf{s}$, since $Q_{\Lambda_s}(\mathbf{G}_c\mathbf{b})$ is always a lattice point of Λ_s. For both $\mathbf{G}_c$ and $\mathbf{G}_s$ with lower triangular structure (or equivalently both upper triangular structure), the matrix $\mathbf{M}$ also has a lower triangular structure (or equivalently upper triangular structure). Suppose that $\mathbf{M}$ is a lower triangular matrix, i.e. for any $i < j$, $M_{i,j} = 0$. Equation (3.10) can be written as:

$$\begin{bmatrix} \hat{b}'_1 \\ \hat{b}'_2 \\ \vdots \\ \hat{b}'_n \end{bmatrix} = \begin{bmatrix} \hat{b}_1 \\ \hat{b}_2 \\ \vdots \\ \hat{b}_n \end{bmatrix} - \begin{bmatrix} M_{1,1} & 0 & \cdots & 0 \\ M_{2,1} & M_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ M_{n,1} & M_{n,2} & \cdots & M_{n,n} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix}. \tag{3.11}$$

Then $\hat{\mathbf{b}}$ can be calculated by solving Diophantine equations for each row of (3.11) recursively from dimension 1 to n as following [26, Section IV-B], where for $i = 1$,

$$\hat{b}_1 = \hat{b}'_1 \bmod M_{1,1} \tag{3.12}$$

$$s_1 = \frac{\hat{b}_1 - \hat{b}'_1}{M_{1,1}}, \tag{3.13}$$

and for $i > 1$

$$\hat{b}_i = \left(\hat{b}'_1 + \sum_{j=1}^{i-1} M_{i,j} s_j \right) \bmod M_{i,i} \tag{3.14}$$

$$s_i = \left(\hat{b}_i - \hat{b}'_1 - \sum_{j=1}^{i-1} M_{i,j} s_j \right) / M_{i,i}. \tag{3.15}$$

Furthermore, if $\mathbf{M}$ is a diagonal matrix, i.e. for any $i \neq j$, $M_{i,j} = 0$, the indexing could be even simpler. Each dimension of $\hat{\mathbf{b}}$ can be independently expressed by

$$\hat{b}'_i = \hat{b}_i - M_i s_i, \tag{3.16}$$

for $i = 1, 2, \cdots, n$. Then $\hat{b}_i$ is obtained by

$$\hat{b}_i = \hat{b}'_i \bmod M_i. \tag{3.17}$$

3.3 Shaping scheme

Given a coding lattice Λ_c, different lattice codes can be constructed by selecting different shaping lattices Λ_s, for which the shaping gain of Λ_s is achieved. A well-known technique is self-similar shaping (or called the Voronoi shaping), where $\Lambda_s = k\Lambda_c$ with integer $k > 1$. However, the Voronoi shaping may have high complexity depending on the design of Q_{Λ_s}. Instead, another important technique called hypercube shaping is considered to have lower shaping complexity, for which Λ_s has a hypercube Voronoi region. Assuming hypercube region of Λ_s have equal length L at each side, so that $\mathbf{G}_s$ has a triangular form with diagonal elements being L. Let $\mathbf{G}_c$ have diagonal element $g_{i,i}$ for $i = 1, 2, \cdots, n$. From Lemma 3.1, integer $M_i = L/g_{i,i}$ and therefore the value of L should take

$$L = K \cdot \mathrm{lcm}(g_{1,1}, g_{2,2}, \cdots, g_{n,n}), \tag{3.18}$$

where integer $K \geq 1$ and *lcm* indicates the least common multiplier.

Hypercube shaping does not have shaping gain, however, it guarantees the equal power allocation at each dimension, specified by L. This scheme is also naturally applicable to construction A/D lattices. For lower triangular $\mathbf{G}_c$, the lcm of $g_{i,i}$ is 2 for construction A lattices and 2^a for construction D lattices, respectively. This allows us to form a lattice code that each dimension contains integer number of bits by setting $K = 2^k$, which can achieve low shaping complexity and have good compatibility with current digital systems.

Assuming a lower triangular $\mathbf{G}_s$ with element $g_{s,i,j}$ for $i, j = 1, 2, \cdots, n$, an algorithm of hypercube shaping is given in Algorithm 3.1.

Algorithm 3.1: Hypercube shaping algorithm

Input: Lower triangular $\mathbf{G}_s$, $\mathbf{x}$

Output: $\mathbf{x}_s$

1 $a_1 = \lfloor x_1/g_{s,1,1} \rceil$;
2 $e_1 = x_i - g_{s,1,1} \cdot a_i$;
3 **for** $i = 2 : 1 : n$ **do**
4 $\quad t_i = y_i - \sum_{j=1}^{i-1} g_{s,i,j} \cdot a_j$;
5 $\quad s_i = t_i / g_{s,i,i}$;
6 $\quad a_i = \lfloor s_i \rceil$;
7 $\quad e_i = t_i - g_{s,i,i} \cdot a_i$
8 **end**
9 $\mathbf{x}_s = [e_1, e_2, \cdots, e_n]^T$;

Remark 3.1: For lattice points located at boundaries of the shaping region, 0.5 appears at the decimal part of $x_i/g_{s,i,i}$. Numerical issue may occur in rounding function, with respect to step 1 and 6 in Algorithm 3.1, of some programming languages. For example, in MATLAB language, round(0.5) = 1, rounding towards the larger integer, and round(-0.5) = -1, rounding towards the smaller integer, from which some lattice points may be double-counted and the size of the codebook may be larger than expected. A simple method to avoid this issue is to shift lattice points by a small random vector, such as $(\pm 0.01, \pm 0.01, \cdots)$ before shaping. Note that it is not guaranteed to move all lattice points from boundaries, but works in most cases.

3.4 Lattice codes for communications

This section gives other related definitions and transmission scheme of lattice codes for communications, considering single user transmission and multiple access.

Definition 3.3: (Code rate) The code rate of a lattice code $\mathcal{C} = \Lambda_c/\Lambda_s$ is given as:

$$R = \frac{1}{n}\log_2 \frac{|\det(\mathbf{G}_s)|}{|\det(\mathbf{G}_c)|}. \tag{3.19}$$

For a self-similar lattice code $\Lambda_c/K\Lambda_c$, the code rate is $R = \log_2 K$. And for rectangular encoding, the code rate can be obtained from the diagonal matrix $\mathbf{M}$ as

$$R = \frac{1}{n}\log_2 \prod_{i=1}^{n} M_i. \tag{3.20}$$

By definition, it is possible to construct a lattice code with code rate $R > 1$, and it can be seen as a coded-modulation scheme with codebook size $|\mathcal{C}| = 2^{nR}$.

Definition 3.4: (Message power) The message power of a lattice code is defined as the average per-dimensional power of a codeword over the whole codebook $\mathcal{C}$ as:

$$P = \frac{1}{n \cdot 2^{nR}} \sum_{\mathbf{x}\in\mathcal{C}} \|\mathbf{x}\|^2. \tag{3.21}$$

Generally, computing exact power has high complexity if the size of the codebook is large. In practice, the message power can be approximated by the second moment, defined in (2.19), of the shaping lattice, or equivalently using the NSM of the shaping lattice by

$$P \approx V(\mathcal{V}_s)^{2/n} \cdot G(\Lambda_s), \tag{3.22}$$

where $\mathcal{V}_s$ is the Voronoi region of the shaping lattice. This estimate becomes accurate as the code rate R increases. Considering the AWGN channel given in Section 2.1.5.3, the signal-to-noise (SNR) is defined as follows.

Definition 3.5: (Signal-to-noise ratio) The SNR of a lattice-based transmission through AWGN channel is defined as:

$$\text{SNR} = \frac{P}{\sigma^2}, \tag{3.23}$$

where P is the message power given in Definition 3.4 and σ^2 is the per-dimensional variance of the Gaussian noise.

3.4.1 Single user transmission

Figure 3.2 gives a system model for a single user lattice-based transmission through the AWGN channel with random dithering, suggested in [14]. Given an uncoded message $\mathbf{b} \in \mathbb{Z}^n$, a lattice point is obtained by

$$\mathbf{x}' = \mathbf{G}_c\mathbf{b} \in \Lambda_c \tag{3.24}$$

Random dithering is applied before shaping and the transmitted message is given as

$$\mathbf{x} = (\mathbf{x}' - \mathbf{d}) \bmod \Lambda_s, \tag{3.25}$$

for which the dithering vector $\mathbf{d} \in \mathbb{R}^n$ is uniformly distributed over the shaping region $\mathcal{V}_s$ and is statically independent of $\mathbf{x}'$. Additionally, the same $\mathbf{d}$ used for encoding is shared at both the encoder and the decoder.

Lemma 3.2: [14, Lemma 1] For any $\mathbf{x}' \in \mathcal{C}$, given $\mathbf{d}$ which is uniformly distributed over $\mathcal{V}_s$ and is statically independent of $\mathbf{x}'$, we have $\mathbf{x} = (\mathbf{x}' - \mathbf{d}) \bmod \Lambda_s$ is also uniformly distributed over $\mathcal{V}_s$ and is statically independent of $\mathbf{x}'$.

Proof. Given $\mathbf{x}'$, by the arithmetic of modulo, we have

$$\mathbf{d} = (\mathbf{x}' - \mathbf{x}) \bmod \Lambda_s. \tag{3.26}$$

It can be seen that the conditional probability $\Pr(\mathbf{x}|\mathbf{x}') = \Pr(\mathbf{d})$. This implies that, for any given $\mathbf{x}'$, the probability density of a lattice codeword $\mathbf{x}$ is a constant over $\mathcal{V}_s$. □

As shown in Figure 3.1, the original lattice codebook typically has a non-zero centroid, therefore it may not match the desired power constraint. Dithering is a common randomization technique and assures average transmission power matches the desired power constraint, which is particularly important for theoretical analysis, see [8] for further details.

The AWGN channel model considered for single user transmission is given in Section 2.1.5.3. The channel output at the receiver is given as

$$\mathbf{y} = \mathbf{x} + \mathbf{n}, \tag{3.27}$$

where $\mathbf{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ is the Gaussian noise.

The decoding process is then given by:

$$\hat{\mathbf{b}} = DEC_{\Lambda_c}(\alpha\mathbf{y} + \mathbf{d}) \bmod \Lambda_s, \tag{3.28}$$

where a real-valued scaling factor $\alpha \in \mathbb{R}$ is applied at the decoder, instead of decoding the received message $\mathbf{y}$ directly. The lattice decoder is defined in Section 2.1.4, that is the closest point lattice quantizer. As we have discussed in the Chapter 1, when $n \to \infty$, there exists lattice codes that achieves the Gaussian channel capacity using lattice decoding, if the scaling factor obtained by MMSE is applied, given as:

$$\alpha_{MMSE} = \frac{P}{P + \sigma^2}. \tag{3.29}$$

Since we aim to study finite dimensional lattice codes, the random dithering vector $\mathbf{d}$ could be removed without introducing a significant performance loss. For simplicity of notation, random dithering is omitted in what follows. However, note that, it should be considered for asymptotical analysis.

3.4.2 Multiple access

Due to the linearity of lattices, lattice codes can be considered as a physical layer network coding (PLNC) technique for multiple access channels. Given a lattice code $\mathcal{C} = \Lambda_c/\Lambda_s$ and $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_L \in \mathcal{C}$, an integer linear combination

$$\hat{\mathbf{y}} = \sum_{i=i}^{l} a_i \mathbf{x}_i \in \Lambda_c, \tag{3.30}$$

for any $a_1, a_2, \cdots, a_L \in \mathbb{Z}$. For lattice codes with group isomorphism [26,29], it can be further obtained that

$$\hat{\mathbf{y}} = \sum_{i=i}^{l} a_i \mathbf{x}_i \bmod \Lambda_s \tag{3.31}$$

is a lattice codeword of $\mathcal{C}$.

Nazer and Gastpar gave a novel relaying scheme, called compute-forward (CF) relaying [30]. Instead of performing multiple user detection at a multiple relay, CF relaying uses lattice-based PLNC and aims to compute an integer linear combination of lattice codewords, which gives a low complexity relay design while having excellent achievable rate. A similar idea can also be applied to various multiple access scenarios, such as compute-forward multiple access (CFMA) [32], multiple access relay channel (MARC) [37]

and MIMO channel using integer forcing (IF) [35, 36]. Related research on implementations of lattice-base PLNC can be found in [38, 39, 53, 57, 58].

CF relaying is considered for multiple access scenario to apply the proposed retry decoding and the CRC embedded lattice codes. Details for CF relaying will be introduced in Chapter 4 with the system model of multiple access scenario.

3.5 Summary and connection to other chapters

This chapter introduced lattice codes, consisting of a coding lattice and a shaping lattice. The encoding/decoding and the shaping scheme are given. In particular, the rectangular encoding and hypercube shaping are applied for numerical evaluations and code design in later chapters. Even though hypercube shaping does not have shaping gain, an equal power allocation can be guaranteed on each symbol within a codeword. For construction A/D lattices from binary codes, rectangular encoding and hypercube shaping assure that M_i in (3.7) is a power of 2. This is a useful property to implement the proposed CRC-embedded lattices/lattice codes to CF relaying scenario, which will be introduced in Chapter 5.

A basic lattice-based transmission scheme for single user scenario is introduced in this chapter. By applying the optimal scaling factor α_{MMSE}, lattice codes are shown to achieve the capacity of the AWGN channel using lattice decoding as dimension $n \to \infty$. However, it is also noticed that a decoding error may still happen for finite dimensional lattice codes even when $R < C$ and the optimal decoding coefficient is applied. A novel scheme to improve the error rate for this situation is proposed in Chapter 4, where the decoder is allowed to retry decoding by adjusting the value of the decoding coefficient. Due to linearity, lattice codes are also suitable for PLNC in a multiple access network. Related applications to various networks and implementations were briefly introduced in this chapter. In particular, CF relaying is considered in a later chapter to implement our proposed retry decoding in Chapter 4 and CRC-embedded lattices/lattice codes in Chapter 5. The details of the system model and an overview of CF relaying scheme are given in Chapter 4.

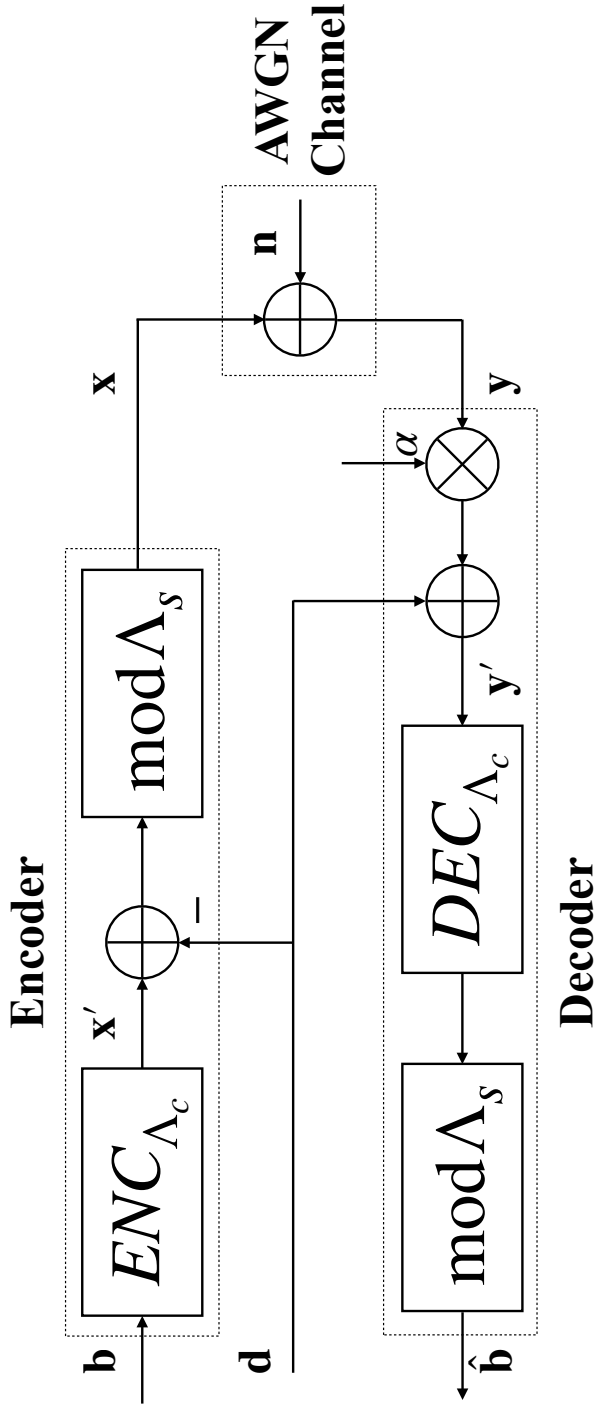


Figure 3.2: System model for a lattice-based single user transmission through the AWGN channel.

Chapter 4

Retry decoding for finite dimensional lattice codes

4.1 Introduction of this chapter

This chapter introduces a novel retry decoding for finite dimensional lattice codes. In a conventional lattice decoder, a decoding coefficient, such as α_{MMSE}, is applied to achieve optimal error performance. However, the optimal decoding may fail for finite dimensional lattice codes when $R < C$. Given an error using the optimal decoding coefficient, it is shown that the error might be corrected if the receiver can retry decoding by adjusting the value of the coefficient.

This chapter is divided into two parts: point-to-point (P2P) single user (SU) transmission and CF relaying. We start from the simple case, where the retry decoding scheme is applied to point-to-point single user transmission through AWGN channel, referred as SU scenario; then we move to multiple access scenario using compute-forward (CF) relaying through fading channel. Details of the decoding scheme and decoding coefficient search algorithm are described for SU scenario and CF relaying, respectively. For the SU scenario, we also give: 1) a lower bound on the error rate derived by allowing an infinite number of retries using exhaustive search; 2) a discussion of the relationship between the benefit of retry and the dimension of lattice code, where the Z_n lattice codes are used to illustrate such relationship; 3) a modified decoding coefficient search strategy to reduce the search cost. For CF relaying, two different schemes for retry decoding are introduced. A discussion is given to compare two schemes by assuming fixed channel and random channel, respectively. The numerical evaluations are given for SU scenario and CF relaying at the end of each section, respectively, where we evaluate the benefit of retry decoding and the discussion given for each scenario.

In this chapter, the error detection for retry decoding is assumed to be genie-aided, where a perfect genie is associated with the receiver who knows the true message and checks if the decoding result is correct. Note that the

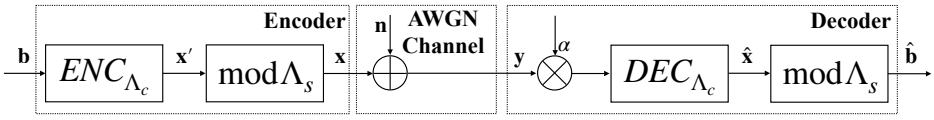


Figure 4.1: Simplified system model for single user transmission through the AWGN channel.

genie only checks the correctness of the decoding result, while not telling the decoder what the true message is. It is clear that genie is only valid for analysis. In practical implementation, a lattice construction with physical layer error detection is required to enable retry decoding, which will be given in the next chapter.

4.2 Retry decoding for single user transmission

4.2.1 System model

For the SU scenario using a lattice code $\mathcal{C} = \Lambda_c/\Lambda_s$, the point-to-point AWGN channel is assumed. Since this research studies finite dimensional lattice codes, random dithering is omitted in the transmission for simplicity of notation. The system model in Figure 3.2 is simplified as Figure 4.1. The decoding error is measured as $\hat{\mathbf{x}} \neq \mathbf{x}$, where the decoding function is

$$\hat{\mathbf{x}} = DEC_{\Lambda_c}(\alpha \cdot \mathbf{y}). \tag{4.1}$$

It is noticed that there may exist $\hat{\mathbf{x}} \neq \mathbf{x}$ such that $\hat{\mathbf{x}} \bmod \Lambda_s = \mathbf{x} \bmod \Lambda_s$, which implies $\hat{\mathbf{x}} = \mathbf{x}_s + \mathcal{C}$ with some $\mathbf{x}_s \in \Lambda_s$. However, by the Gaussian distribution of the channel noise, the probability of such event is negligible in practical decoding.

4.2.2 Decoding scheme

For the SU scenario using finite dimensional lattice codes with one-shot decoding, the receiver applies the MMSE scaling factor α_{MMSE} introduced in (3.29) to received message for lattice decoding as

$$\hat{\mathbf{x}} = DEC_{\Lambda_c}(\alpha_{MMSE} \cdot \mathbf{y}). \tag{4.2}$$

If a decoding error is detected, the proposed retry decoding scheme allows the lattice decoder to adjust the scaling factor α to achieve a lower error

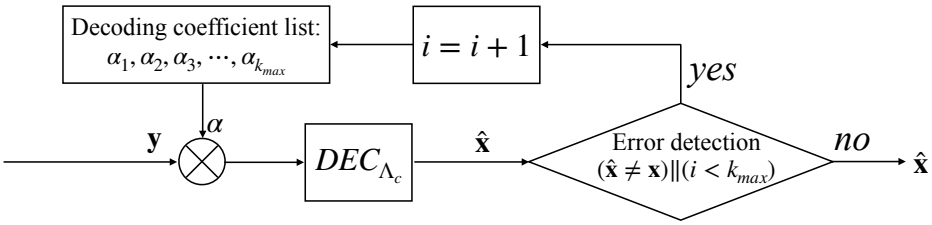


Figure 4.2: Redesigned decoder structure for retry decoding in SU scenario.

rate. Suppose the maximum number of decoding attempts is k_{max} using a decoding coefficient list $\alpha_1, \alpha_2, \cdots, \alpha_{k_{max}}$. Figure 4.2 gives a general structure of the redesigned decoder, where the decoding coefficient list is pre-stored at the decoder and the counter i is initialized to be 1 before the first decoding attempt. The value of α_1 is typically set to α_{MMSE}. This decoder can also be applied to power unconstrained communications using lattices, where the optimal scaling factor can be seen as

$$\alpha_{MMSE} = P/(P + \sigma^2) \to 1, \tag{4.3}$$

since the average power $P \to \infty$.

In this section, the decoding coefficient list is further grouped into k non-overlapping subsets $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_k$. In order to perform retry efficiently, the list is generated in the descending order based on reliability of α, so that α with higher reliability would be tested before those with lower reliability. Each subset indicates a decoding level and may contain multiple α candidates. The decoding starts from $\alpha \in \mathcal{A}_1$; then tests α candidates in $\mathcal{A}_2, \cdots, \mathcal{A}_k$ sequentially. Error detection can be performed after each decoding attempt, rather than the whole level, to terminate the decoding as soon as no error is detected. If all α's failed on error detection, the decoder may output a decoding failure to request a retransmission. The technique on generating the decoding coefficient list, including how to find reliable α's and how to group them, will be discussed in the next subsection.

4.2.3 Decoding coefficient search algorithm

An algorithm to find the decoding coefficient list is considered, which also shows how we group the list and the number of elements in each subset of $\mathcal{A}$ using genie-aided decoding and Monte-Carlo-based search. To perform efficient decoding, $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_k$ are generated in the order to maximize the probability of correct decoding given the previous candidates failed.

Let $\mathbf{x} \in \mathcal{C}$ be a non-zero, randomly generated lattice codeword. The probability of correctly decoding the received message $\mathbf{y} = \mathbf{x} + \mathbf{n}$ is a function

of α:

$$P(\alpha) = \Pr(DEC_{\Lambda_c}(\alpha \cdot \mathbf{y}) = \mathbf{x}). \tag{4.4}$$

Given a search space $[\alpha_{min}, \alpha_{max}]$, this algorithm is performed level-by-level to find a subset $\mathcal{A}$ at each level. The first step of the algorithm finds $\alpha_{1,1}$ that maximizes

$$\alpha_{1,1} = \underset{\alpha_{min} \leq \alpha \leq \alpha_{max}}{\arg\max} \; P(\alpha). \tag{4.5}$$

The result of the first step is given as $\mathcal{A}_1 = \{\alpha_{1,1}\}$. Empirically, we have $\alpha_{1,1} \approx \alpha_{MMSE}$ [14], [41], which could alternatively be used with almost no loss of error performance.

The second step of search is performed by assuming a decoding failure using $\mathcal{A}_1$, i.e. $DEC_{\Lambda_c}(\alpha_{1,1} \cdot \mathbf{y}) \neq \mathbf{x}$. The algorithm maximizes the conditional probability

$$\underset{\alpha_{min} \leq \alpha \leq \alpha_{max}}{\arg\max} \; P(\alpha | DEC_{\Lambda_c}(\alpha_{1,1} \cdot \mathbf{y}) \neq \mathbf{x}). \tag{4.6}$$

By the assumption, we already have $P(\alpha_{1,1} | DEC_{\mathcal{C}}(\alpha_{1,1} \cdot \mathbf{y}) \neq \mathbf{x}) = 0$. The search space is then divided into two part as $[\alpha_{min}, \alpha_{1,1})$ and $(\alpha_{1,1}, \alpha_{max}]$, from which two α candidates can be found in this step as:

$$\alpha_{2,1} = \underset{\alpha_{min} \leq \alpha < \alpha_{1,1}}{\arg\max} \; P(\alpha | DEC_{\Lambda_c}(\alpha_{1,1} \cdot \mathbf{y}) \neq \mathbf{x}) \tag{4.7}$$

$$\alpha_{2,2} = \underset{\alpha_{1,1} < \alpha \leq \alpha_{max}}{\arg\max} \; P(\alpha | DEC_{\Lambda_c}(\alpha_{1,1} \cdot \mathbf{y}) \neq \mathbf{x}). \tag{4.8}$$

The result of the second step is given as $\mathcal{A}_2 = \{\alpha_{2,1}, \alpha_{2,2}\}$. The algorithm is then performed recursively to generate the decoding coefficient list. It can be seen that the algorithm finds 2^{k-1} α candidates at the k-th step to generate $\mathcal{A}_k$. As a result, the number of decoding coefficients for retry in a list $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_k$ is

$$\sum_{i=1}^{k} |\mathcal{A}_i| = 2^k - 1. \tag{4.9}$$

A generalization of this algorithm is given as follows. At the $(k+1)$-th ($k \geq 1$) step, let $\mathcal{A}_1, \mathcal{A}_2 \cdots, \mathcal{A}_k$ be the decoding coefficient list found in previous steps. To find $\mathcal{A}_{k+1}$, we define:

1. an event e_k is the event that all α in $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_k$ as

$$\forall \alpha \in \bigcup_{i=1}^{k} \mathcal{A}_i, \quad DEC_{\Lambda_c}(\alpha \cdot \mathbf{y}) \neq \mathbf{x}, \tag{4.10}$$

2. a sorted list $\mathcal{A}'_k = sort(\{\alpha_{min}, \alpha_{max}\} \cup \bigcup_{i=1}^{k} \mathcal{A}_i)$, such that elements are in ascending order as

$$\alpha_{min} = \alpha'_1 < \alpha'_2 < \cdots < \alpha'_{2^k} < \alpha'_{2^k+1} = \alpha_{max}. \tag{4.11}$$

Define a conditional probability of correct decoding given the event e_k occurred as

$$P(\alpha|e_k) = \Pr(DEC_{\Lambda_c}(\alpha \cdot \mathbf{y}) = \mathbf{x}|e_k). \tag{4.12}$$

By the recursive structure, it is clear that $P(\alpha|e_k) = 0$ for any $\alpha \in \bigcup_{i=1}^{k} \mathcal{A}_i$. The search space is then split into 2^k intervals bounded by adjacent elements in $\mathcal{A}'_k$. From each interval, the algorithm finds a local optimum α that maximizes (4.12) as:

$$\alpha_{k+1,j} = \underset{\alpha'_j \le \alpha < \alpha'_{j+1}}{\arg\max}\ P(\alpha|e_k), \tag{4.13}$$

where $\alpha'_j, \alpha'_{j+1} \in \mathcal{A}'_k$ for $j = 1, 2, \cdots, 2^k$. The result of the $(k+1)$-th step of algorithm is $\mathcal{A}_{k+1} = \{\alpha_{k+1,1}, \alpha_{k+1,2}, \cdots, \alpha_{k+1,2^k}\}$.

Figure 4.3 illustrates $P(\alpha)$ and $P(\alpha|e_i)$ for $i = 1, 2$ using an E_8 lattice code, from which α candidates for $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$ are found, where

$$\begin{aligned} \alpha_{1,1} = \alpha_{MMSE} &\approx 0.9786 \\ \{\alpha_{2,1}, \alpha_{2,2}\} &\approx \{0.9103, 1.0555\} \\ \{\alpha_{3,1}, \alpha_{3,2}, \alpha_{3,3}, \alpha_{3,4}\} &\approx \{0.8676, 0.9445, 1.0128, 1.1153\}. \end{aligned}$$

It is shown that the search space in the $(k+1)$-th step is split into 2^k intervals between α's for which $P(\alpha|e_k) = 0$. It is also observed that, within each subset $\mathcal{A}_{i+1}$, the values of $P(\alpha|e_i)$ for different candidates are close. This implies that the test order within a subset $\mathcal{A}_{i+1}$ could be arbitrary for retry decoding. In particular, for $i = 1$, the two maximum values of $P(\alpha|e_1)$ are approximately 0.2477 and 0.2996 using $\alpha_{2,1}$ and $\alpha_{2,2}$, respectively, indicating a fraction of 0.5473 of messages failed decoding using α_{MMSE} can be corrected by retry decoding using $\mathcal{A}_2 = \{\alpha_{2,1}, \alpha_{2,2}\}$.

For a given lattice code $\mathcal{C}$, the decoding coefficient list only depends on SNR. Therefore, the algorithm can be performed offline to generate a look-up table of decoding coefficients for all required SNR values in advance. The practical implementation complexity can be further reduced by optimizing α candidates only for one SNR value, and then using them across all SNR values for decoding, instead of generating an SNR-dependent decoding coefficient list. This also reduces the size of look-up table which stores the decoding

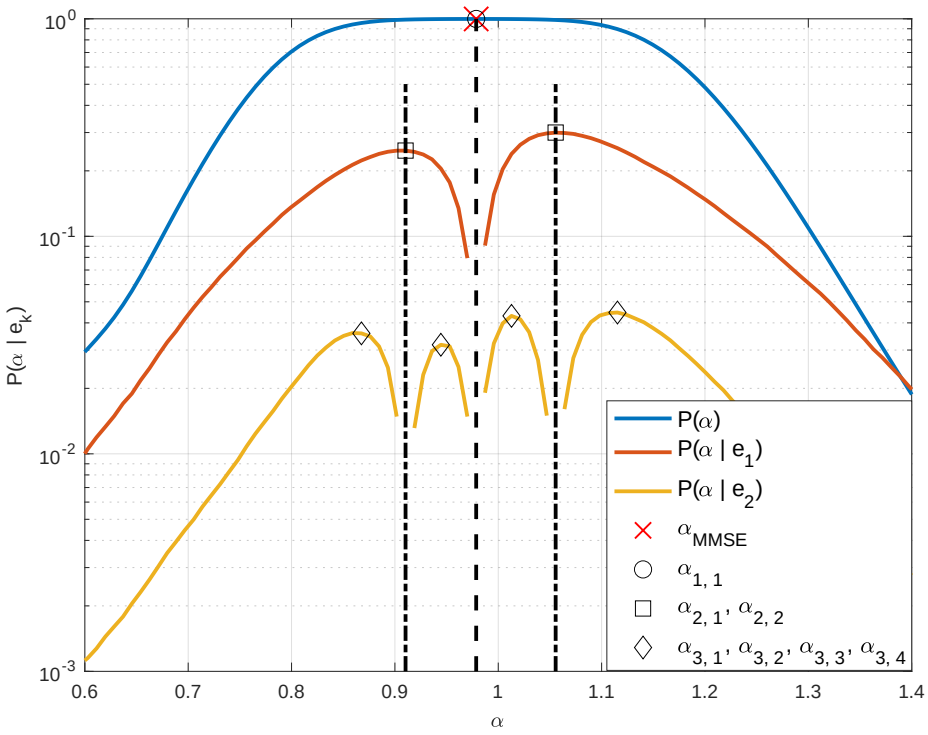


Figure 4.3: $P(\alpha)$, $P(\alpha|e_1)$ and $P(\alpha|e_2)$ curve for E_8 lattice code with hypercube shaping and code rate $R = 2$. SNR $= 17$dB so that $1 - P(\alpha_{MMSE}) \approx 10^{-3}$. The α_{MMSE} and search results $\alpha_{1,1} \cdots \alpha_{3,4}$ are marked at the corresponding curves.

coefficient list. An error performance loss may occur due to the mismatch between values of α and SNR. This will be evaluated in Section 4.2.5.3, where we observed the performance loss could be negligible in practice.

Additionally, the follow proposition is given for retry decoding using the list $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_k$ found above and genie-aided error detection.

Proposition 4.1: Suppose all $\alpha \in \bigcup_{i=1}^{k-1} \mathcal{A}_i$ failed decoding, there exists at most one $\alpha \in \mathcal{A}_k$ can correctly decode the received message.

Proof. Since the case that all $\alpha \in \mathcal{A}_k$ fail decoding is trivial, this proof is given by showing the contradiction that there does not exist distinct $\alpha_{k,a}, \alpha_{k,b} \in \mathcal{A}_k$ such that the estimate $\hat{\mathbf{x}}_a = DEC_{\Lambda_c}(\alpha_{k,a} \cdot \mathbf{y})$ and $\hat{\mathbf{x}}_b = DEC_{\Lambda_c}(\alpha_{k,b} \cdot \mathbf{y})$ satisfy $\hat{\mathbf{x}}_a = \hat{\mathbf{x}}_b = \mathbf{x}$.

Suppose existing $\alpha_{k,a} < \alpha_{k,b}$ that both of them can correctly decode the message. The lattice decoder DEC_{Λ_c} is a closest point quantizer, from which we have $\alpha_{k,a} \cdot \mathbf{y}, \alpha_{k,b} \cdot \mathbf{y} \in \mathcal{V}(\mathbf{x})$. Since the Voronoi region $\mathcal{V}$ of a lattice is a

convex polygon, it implies that, for all $0 < t < 1$ and $\alpha' = t \cdot \alpha_{k,a} + (1-t) \cdot \alpha_{k,b}$, we have:

$$\alpha' \cdot \mathbf{y} \in \mathcal{V}(\mathbf{x}), \tag{4.14}$$

i.e. the received message can be correctly decoded using any $\alpha_{k,a} < \alpha' < \alpha_{k,b}$. However, by the structure of the algorithm, there must exist at least one α candidate satisfying $\alpha_{k,a} < \alpha < \alpha_{k,b}$ belongs to $\bigcup_{i=1}^{k-1} \mathcal{A}_i$, which failed decoding by the assumption. This leads to a contradiction and concludes the proof.

□

4.2.4 Lower bound on error rate

A lower bound on error probability for retry decoding is derived by assuming a genie-aided exhaustive search decoder when the search space is extended from a finite-length list to all non-zero $\alpha \in \mathbb{R}$.

4.2.4.1 Decodable region

By the genie-aided exhaustive search decoder, a non-zeo transmitted message $\mathbf{x}$ can be correctly decoded from the received message $\mathbf{y}$ if and only if $\exists \alpha \in \mathbb{R} \setminus 0, DEC_{\Lambda_c}(\alpha \cdot \mathbf{y}) = \mathbf{x}$. This implies that the line connecting $\mathbf{y}$ and the origin $\mathbf{0}$ passes through the Voronoi region $\mathcal{V}(\mathbf{x})$. We refer to any such $\mathbf{y}$ as decodable and define the union of all decodable $\mathbf{y}$ as the decodable region with respect to $\mathbf{x}$.

Definition 4.1: (Decodable region) Given a non-zero lattice point $\mathbf{x}$, the decodable region is

$$\mathcal{D}(\mathbf{x}) = \{\frac{1}{\alpha}\mathbf{u} | \alpha \in \mathbb{R} \setminus 0, \mathbf{u} \in \mathcal{V}(\mathbf{x})\}. \tag{4.15}$$

Geometrically, $\mathcal{D}(\mathbf{x})$ forms an n-dimensional cone region with vertex at the origin $\mathbf{0}$. A 2-dimensional example is illustrated in Figure 4.4. Suppose $\mathbf{x}_1 = [5, 0]^t$ is transmitted, then $\mathbf{y}_1$ is decodable and $\mathbf{y}_1'$ is non-decodable. The decoding error probability of the genie-aided exhaustive search decoder for a given $\mathbf{x}$ is obtained as

$$P_{e,Dec} = 1 - \Pr(\mathbf{y} \in \mathcal{D}(\mathbf{x})). \tag{4.16}$$

However, the area of $\mathcal{D}(\mathbf{x})$ depends on the value of $\mathbf{x}$ (not only the underlying lattice Λ and the message power $\|\mathbf{x}\|^2$). For example, as shown in Figure 4.4,

$\mathbf{x}_1 = [5, 0]^T$ and $\mathbf{x}_2 = [4, 3]^T$ with $\|\mathbf{x}_1\|^2 = \|\mathbf{x}_2\|^2$, the area of $\mathcal{D}(\mathbf{x}_1)$ and $\mathcal{D}(\mathbf{x}_2)$ are different since the angle $\theta_1 \neq \theta_2$. Therefore $P_{e,Dec}$ is hard to find in general.

In order to give an analytical study, using the covering sphere $\mathcal{S}_c(\mathbf{x})$, we define

$$\mathcal{D}_c(\mathbf{x}) = \{\frac{1}{\alpha}\mathbf{u} | \alpha \in \mathbb{R} \setminus 0, \mathbf{u} \in \mathcal{S}_c(\mathbf{x})\}. \tag{4.17}$$

The error probability with respect to $\mathcal{S}_c$ and $\mathcal{D}_c$ can be defined as:

$$P_{e,cover} = 1 - \Pr(\mathbf{y} \in \mathcal{D}_c) \tag{4.18}$$

For finite dimensional lattices, we have $\mathcal{V}(\mathbf{x}) \subset \mathcal{S}_c(\mathbf{x})$, by which $\mathcal{D}(\mathbf{x}) \subset \mathcal{D}_c(\mathbf{x})$ and $P_{e,Dec}$ is strictly lower bounded by

$$P_{e,Dec} > P_{e,cover}. \tag{4.19}$$

The probability $\Pr(\mathbf{y} \in \mathcal{D}_c(\mathbf{x}))$ only depends on the message power $\|\mathbf{x}\|^2$ and covering radius r_c. Similarly, using the effective sphere $\mathcal{S}_e(\mathbf{x})$, we can define

$$\mathcal{D}_e(\mathbf{x}) = \{\frac{1}{\alpha}\mathbf{u} | \alpha \in \mathbb{R} \setminus 0, \mathbf{u} \in \mathcal{S}_e(\mathbf{x})\}, \tag{4.20}$$

from which an *effective sphere estimate* of error probability is given by:

$$P_{e,Dec} \approx P_{e,effc} = 1 - \Pr(\mathbf{y} \in \mathcal{D}_e(\mathbf{x})). \tag{4.21}$$

4.2.4.2 Lower bound and effective sphere estimate of error probability

An analytical form of the lower bound on error probability in (4.19) and the effective sphere estimate in (4.21) are derived for AWGN transmission in Theorem 4.1. We first introduce the following lemma, which gives a closed form to compute integral of Gaussian noise over a zero-centered sphere.

Lemma 4.1: [59] Given an n-dimensional Gaussian noise $\mathbf{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ and a zero-centered sphere $\mathcal{S}$ with radius r_e. Let $t = r_e^2/2\sigma^2$, the probability of a noise sample falls outside of the sphere has a closed form as:

$$P_e = e^{-t}\left(1 + \frac{t}{1!} + \cdots + \frac{t^{n/2-1}}{(n/2-1)!}\right) \tag{4.22}$$

for n even; while for odd n, we have:

$$P_e = \text{erfc}(t^{\frac{1}{2}}) + e^{-t}\left(\frac{t^{1/2}}{(1/2)!} + \frac{t^{3/2}}{(3/2)!} + \cdots + \frac{t^{n/2-1}}{(n/2-1)!}\right). \tag{4.23}$$

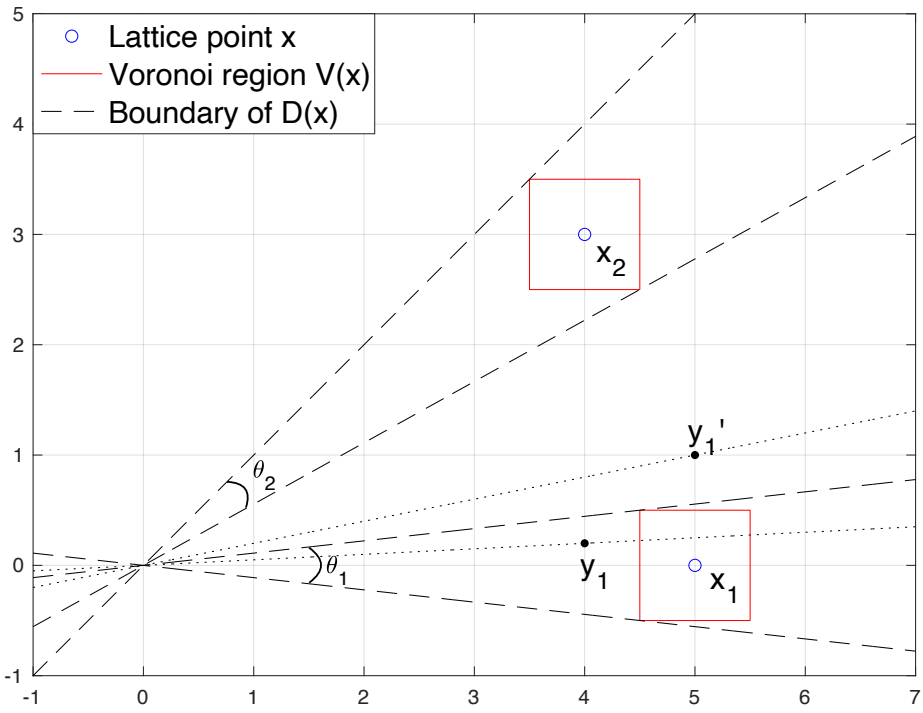


Figure 4.4: $\mathcal{D}(\mathbf{x})$ of Z_2 lattice with $\mathbf{x}_1 = [5,0]^T$ and $\mathbf{x}_2 = [4,3]^T$. With respect to $\mathbf{x}_1$, a decodable $\mathbf{y}_1$ and a non-decodable $\mathbf{y}_1'$ are plotted. The area of $\mathcal{D}(\mathbf{x}_1)$ and $\mathcal{D}(\mathbf{x}_2)$ are different since the angle $\theta_1 \neq \theta_2$.

Theorem 4.1: Let non-zero $\mathbf{x}$ be a lattice point of an $n \geq 2$ dimensional lattice Λ having covering radius r_c. Let $P = \|\mathbf{x}\|^2/n$ and σ^2 be per-dimensional message and noise power, respectively. With the restriction $r_c^2 < nP_{\mathbf{x}}$, the word error probability for retry decoding is lower bounded by:

$$P_{e,Dec} > 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{n_1^2}{2\sigma^2}} (1 - h(n_1)) dn_1 \tag{4.24}$$

where if n is odd:

$$h(n_1) = e^{-t} \left(\sum_{k=0}^{(n-3)/2} \frac{t^k}{k!} \right)$$

and if n is even:

$$h(n_1) = \text{erfc}(t^{1/2}) + e^{-t} \left(\sum_{k=1}^{(n-2)/2} \frac{t^{k-1/2}}{(k-1/2)!} \right)$$

with $t = f^2(n_1)/(2\sigma^2)$ and $f(n_1) = \left| \frac{r_c}{\sqrt{nP_{\mathbf{x}} - r_c^2}} n_1 + \sqrt{\frac{nP_{\mathbf{x}} r_c^2}{nP_{\mathbf{x}} - r_c^2}} \right|$.

Proof. Denote $g_n(\mathbf{n})$ as the density function of an n-dimensional Gaussian noise $\mathbf{n} \sim (0, \sigma^2\mathbf{I})$. The probability that $\mathbf{y}$ falls into $\mathcal{D}_c$ in (4.19) is:

$$\Pr(\mathbf{y} \in \mathcal{D}_c) = \Pr(\mathbf{n} \in \mathcal{D}_c') = \int_{\mathcal{D}_c'} g_n(\mathbf{n})\, d\mathbf{n} \tag{4.25}$$

where $\mathcal{D}_c' = \mathcal{D}_c - \mathbf{x}$. Due to the circular symmetry of the Gaussian noise and the covering sphere, it is equivalent to consider a rotated coordinate system where the z_1 axis corresponds to the line connecting the vertex of the decodable region and $\mathbf{x}$, where the coordinate of the vertex of the decodable region is $(-\sqrt{nP_{\mathbf{x}}}, 0, 0, \ldots, 0)$. (See Figure 4.5 for an example in 2 dimensions). The Euclidean distance r_n between the boundaries of decodable region and n_1 axis is a function of n_1 as

$$r_n = f(n_1) = \left| \frac{r_c}{\sqrt{nP_{\mathbf{x}} - r_c^2}} n_1 + \sqrt{\frac{nP_{\mathbf{x}} r_c^2}{nP_{\mathbf{x}} - r_c^2}} \right|. \tag{4.26}$$

Due to the independence among dimensions of Gaussian noise, (4.19) can be written as:

$$P_{e,cover} = 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{n_1^2}{2\sigma^2}} P_s(r_n) dn_1 \tag{4.27}$$

with

$$P_s(r_n) = \int_{\mathcal{S}_{n-1}(r_n)} g_{n-1}(n_2, n_3, \ldots, n_n)\, dn_2 \ldots dn_n. \tag{4.28}$$

The region $\mathcal{S}_{n-1}(r_n)$ can be considered as a truncated slice of a n dimensional Gaussian which is the intersection of the decodable region $\mathcal{D}_c'$ and a hyperplane orthogonal the n_1 axis, which gives an $n-1$ dimensional sphere with radius r_n. By Lemma 4.1, the integral of $P_s(r_n) = 1 - P_e$ has a closed form given in (4.22) and (4.23) with dimension $n-1$ as:

$$P_s(r_n) = \begin{cases} 1 - e^{-t}\left(1 + \frac{t}{1!} + \frac{t^2}{2!} + \cdots + \frac{t^{(n-3)/2}}{((n-3)/2)!}\right) & n \text{ is odd} \\ 1 - \mathrm{erfc}(t^{1/2}) - e^{-t}\left(\frac{t^{1/2}}{1/2!} + \frac{t^{3/2}}{(3/2)!} + \cdots + \frac{t^{(n-3)/2}}{((n-3)/2)!}\right) & n \text{ is even} \end{cases}, \tag{4.29}$$

where $t = f^2(n_1)/(2\sigma^2)$. Note that the integration in $P_s(r_n)$ is taken over $n-1$ dimensions, the even and odd are opposite of (4.22) and (4.23). By combining

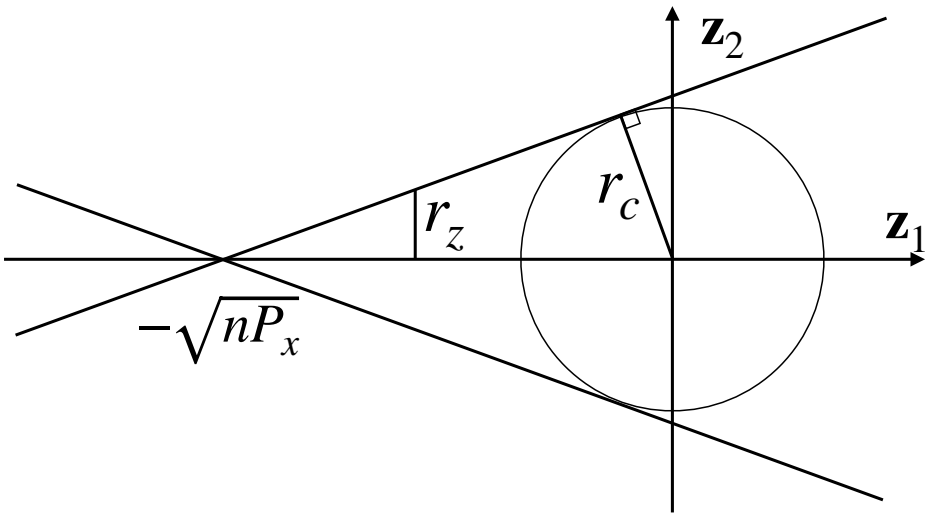


Figure 4.5: Example of rotated $\mathcal{D}_c(\mathbf{x}) - \mathbf{x}$ in 2 dimensions. The vertex of the decodable region is $(-\sqrt{nP_{\mathbf{x}}}, 0)$, where $P_{\mathbf{x}}$ is the message power. r_c is the covering radius.

(4.27) and (4.29), the probability of correct decoding $\Pr(\mathbf{n} \in \mathcal{D}_c')$ is obtained. Then the lower bound on the WER using the genie-aided exhaustive search decoder follows by $P_{e,Dec} > P_{e,cover} = 1 - \Pr(\mathbf{n} \in \mathcal{D}_c')$.

□

Corollary 4.1: The effective sphere estimate in (4.21) can be obtained using the same form as (4.24) by replacing the covering radius r_c with the effective radius r_e.

Since there is only single integral contained in (4.24), the lower bound and the effective sphere estimate can be calculated easily by numerical methods. Theorem 4.1 and Corollary 4.1 are given with respect to a lattice Λ, but the result can be extended to lattice codes $\mathcal{C} = \Lambda_c/\Lambda_s$ under lattice decoding, where the P is the average per-dimensional power of $\mathcal{C}$ and radius r_c (or r_e) is corresponding to the coding lattice Λ_c.

Note that Theorem 4.1 and Corollary 4.1 are meaningful only when $r^2 < nP_{\mathbf{x}}$ is satisfied, or equivalently $P_{\mathbf{x}} > r^2/n$. Because when $r^2 < nP_{\mathbf{x}}$, the sphere covers the origin $\mathbf{0}$, by which $\mathcal{D}_c$ is the whole real number space thus doesn't form a cone-like region and $P_{e,cover} = 0$.

4.2.4.3 Discussions

The lower bound and effective sphere estimate are derived for finite dimensional lattices/lattice codes. From a theoretical point of view, as $n \to \infty$, there exists a lattice code that achieves the Gaussian channel capacity by using α_{MMSE} [14]. Also, the existence of lattices with good for covering and quantization are demonstrated in [8, 17], which implies $P_{e,Dec} = P_{e,cover} = P_{e,effc}$ and the benefit of retry decoding approaches 0. On the other hand, for finite n, the only noise component in the SU scenario is the additive white Gaussian noise. As dimension n increases, it is known that the probability density of Gaussian noise is concentrated in a thin annulus with radius $\sqrt{n\sigma^2}$ [60], i.e. there exists $\epsilon > 0$ such that

$$\Pr\left(\sqrt{n\sigma^2} - \epsilon < \|\mathbf{n}\| < \sqrt{n\sigma^2} + \epsilon\right) \to 1. \tag{4.30}$$

By letting the Voronoi region $\mathcal{V}$ cover a sphere with radius $\sqrt{n\sigma^2}/\alpha_{MMSE}$, the improvement of retry decoding decreases as n increases. A hypothesis is considered that the benefit of retry decreases as n increases, and approaches 0 when $n \to \infty$.

In particular, the lower bound and effective sphere estimate are suitable for analyzing error probabilities of low dimensional lattices, which have known covering radius and well-studied geometric properties. Since the covering sphere only satisfies $\mathcal{V}(\mathbf{x}) \subset \mathcal{S}_c(\mathbf{x})$ for finite n, the lower bound may not be tight. The tightness depends on the lattice covering thickness defined as $\Theta(\Lambda) = V(\mathcal{S}_c)/V(\Lambda)$, from which the bound is good if a lattice is good for covering, that is $\Theta(\Lambda) \approx 1$. For finite n, an upper and a lower bound on the thinnest covering are given in [3, Chapter 2] as

$$\frac{n}{e\sqrt{e}} \lessapprox \Theta \leq n \ln n + n \ln \ln n + 5n, \tag{4.31}$$

where the lower bound, implying the best achievable covering thickness increases as n grows. However, to give an analytical description of the relationship between the covering thickness and the tightness of the lower bound is still an open question. On the other hand, the accuracy of the effective sphere estimate in Corollary 4.1 depends on the NSM, defined in (2.20), of the lattice. The estimate is good if a lattice is good for quantization, that is $G(\Lambda) \approx G(\mathcal{B})$, implying the shape of Voronoi region is close to a sphere. For such lattices, the effective sphere estimate gives an accurate estimate of WER under the genie-aided exhaustive search decoder.

4.2.5 Numerical results

Now, we give numerical evaluations for the retry decoding, the lower bound and effective sphere estimate discussed above using E_8, BW_{16} and Leech lattice codes. A numerical example is also given to illustrate the relationship between benefit of retry and dimension of lattice code using $Z_n/4Z_n$ lattice codes. We also give an evaluation by assuming an SNR-independent decoding coefficient list.

4.2.5.1 Evaluation of the lower bound

Figure 4.6-4.8 evaluates the retry decoding scheme for SU scenario, along with the lower bound in (4.19) and the effective sphere estimate in (4.21). E_8, BW_{16} and Leech lattice codes are considered where hypercube shaping is applied. For comparison, genie-aided exhaustive search decoding is evaluated to indicate the best retry decoding can achieve. A large search space of $[0.15, 1.85]$ for E_8, $[0.4, 1.6]$ for BW_{16} and Leech lattice codes are assumed, respectively, in which 200 α candidates are allocated uniformly. The SNR gains of 0.5dB, 0.4dB and 0.24dB are achieved at WER$= 10^{-5}$, respectively, compared to the one-shot decoding using α_{MMSE} only. Since E_8, BW_{16} and Leech lattices are the best-known quantizers among 8, 16 and 24 dimensional lattices [3] respectively, this implies a sphere-like Voronoi region from which (4.21) gives a relatively accurate estimate of WER of the genie-aided exhaustive search decoding. The WER performance of decoding using a finite length coefficient list $\mathcal{A}_1$ and $\mathcal{A}_2$ are also evaluated, which approaches that of exhaustive search decoding. This indicates that, by appropriately generating the decoding coefficient list, only a small number of decoding attempts is needed to approach the error performance of the exhaustive search decoding.

4.2.5.2 Relationship between benefit of retry and dimension

From Figure 4.6-4.8, it is observed that the gain of exhaustive search decoding decreases as dimension increases. However, the underlying coding lattices have different structure, which gives different coding gain in communications. A numerical example is provided to illustrate the relationship between benefit of retry decoding and dimension of lattice codes using $Z_n/4Z_n$ lattice codes for dimension $n = 2, 4, 8, 16, 24, 32, 48, 64, 80, 96, 128, 256, 500, 1000$. Although Z_n lattices are not good for communications, the structure of Z_n lattice allows us to give a fair comparison for benefit of retry decoding across various dimensions, compared to one-shot decoding. Let the target WER be 10^{-5}. The SNR gain obtained by the genie-aided exhaustive search

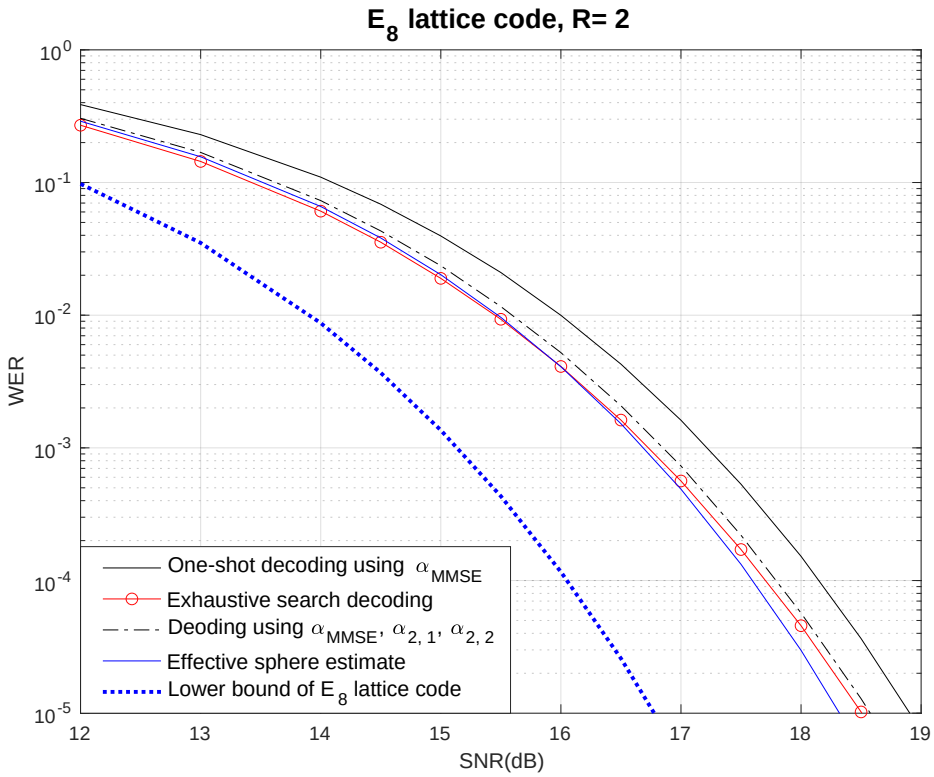


Figure 4.6: Numerical evaluation of the retry decoding using E_8 lattice code, along the corresponding lower bound and effective sphere estimate. The code rate is $R = 2$.

decoding, compared to the one-shot decoding using α_{MMSE} only, is illustrated in Figure 4.9. It is observed that the gain decreases as the dimension n increases, which agrees with the discussion given in Section 4.2.4.3.

4.2.5.3 Using SNR-independent coefficient list for decoding

Section 4.2.3 gives an offline algorithm using exhaustive search to generate an SNR-dependent decoding coefficient list for retry decoding. However, the high complexity of exhaustive search may affect code design efficiency and the SNR-dependent list requires storage space at the receiver. Instead, a modified strategy of retry decoding for SU scenario is considered, which finds one set of decoding coefficients for a specific SNR and applies to all SNRs for decoding, to reduce the search cost and required storage space at received. A WER performance loss could be expected due to the mismatch between decoding coefficient and SNR values. Figure 4.10 and Figure 4.11 show numerical examples using E_8 and BW_{16} lattice codes as used in Figure 4.6

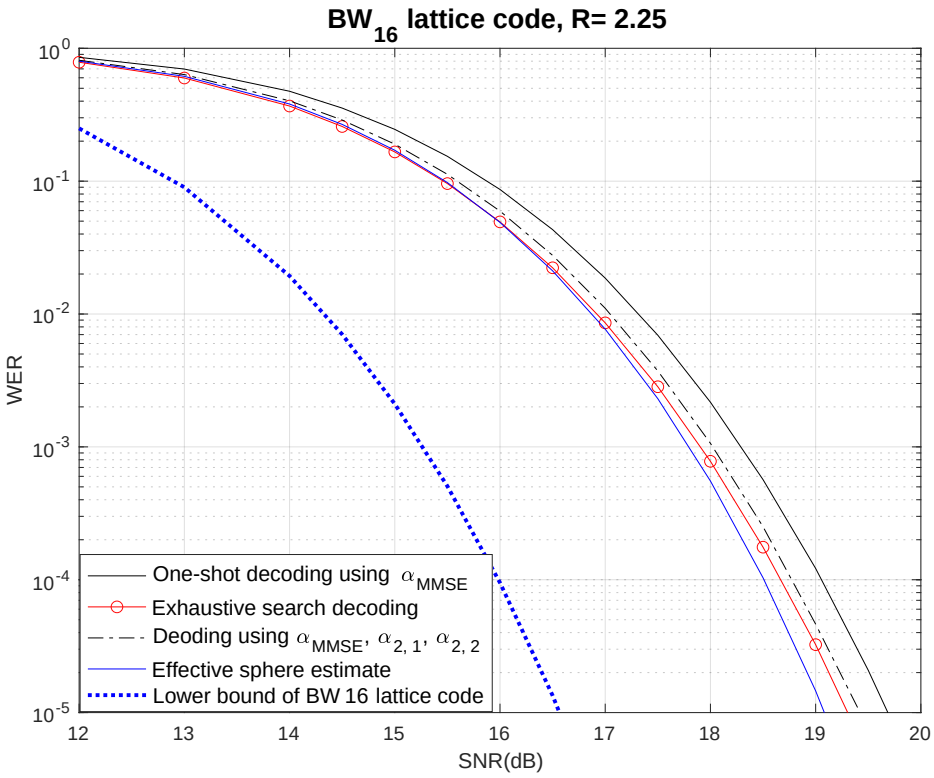


Figure 4.7: Numerical evaluation of the retry decoding using BW_{16} lattice code, along the corresponding lower bound and effective sphere estimate. The code rate is $R = 2.25$.

and Figure 4.7. It can be seen that the modified strategy only has a small performance loss. Particularly, when $\mathcal{A}_2 = \{\alpha_{2,1}, \alpha_{2,2}\}$ optimized for a high SNR is used across all SNRs, the performance loss is negligible.

4.3 Retry decoding for multiple access

In this section, we consider the multiple access scenario using CF relaying. An overview of CF relaying is first given along with the system model used in this section.

4.3.1 Overview of compute-forward

Compute-forward is a lattice-based multiple access relaying scheme [30] where one or more relays aim to decode one or more linear combinations (or called linear equations) of users' messages instead of decoding them

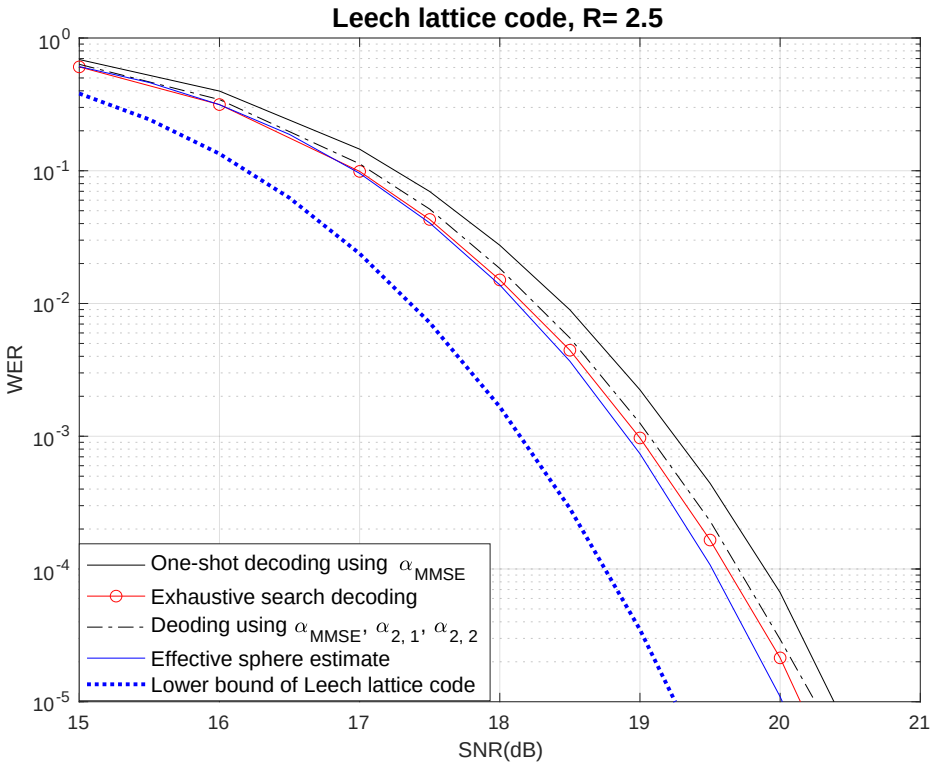


Figure 4.8: Numerical evaluation of the retry decoding using Leech lattice code, along the corresponding lower bound and effective sphere estimate. The code rate is $R = 2.5$.

individually. In [30], the uncoded message $\mathbf{b} \in \mathbb{F}_q^n$, where q is a prime number. An L-user J-relay system model, with $J \geq L$, is given in Figure 4.12 for CF relaying.

Suppose all users share the same codebook $\mathcal{C} = \Lambda_c/\Lambda_s$ with equal power allocation P. The users send lattice codewords $\mathbf{x}_i = ENC(\mathbf{b}_i)$, for $i = 1, 2, \cdots, L$, through a multiple access channel. The received message at the j-th relay is

$$\mathbf{y}_j = \sum_{i=1}^{L} h_{i,j}\mathbf{x}_i + \mathbf{n}_j, \tag{4.32}$$

where $h_{i,j} \in \mathbb{R}$ is the channel coefficient between the i-th user and the j-th relay and $\mathbf{n}_j \sim \mathcal{N}(0, \sigma^2\mathbf{I}_n)$. In the following, the vector form of the channel from users to the j-th relay is denoted by a column vector as $\mathbf{h} = [h_{1,j}, h_{2,j}, \cdots, h_{L,j}]^T$. As in the single user transmission case, random dithering for CF relaying [30] is omitted.

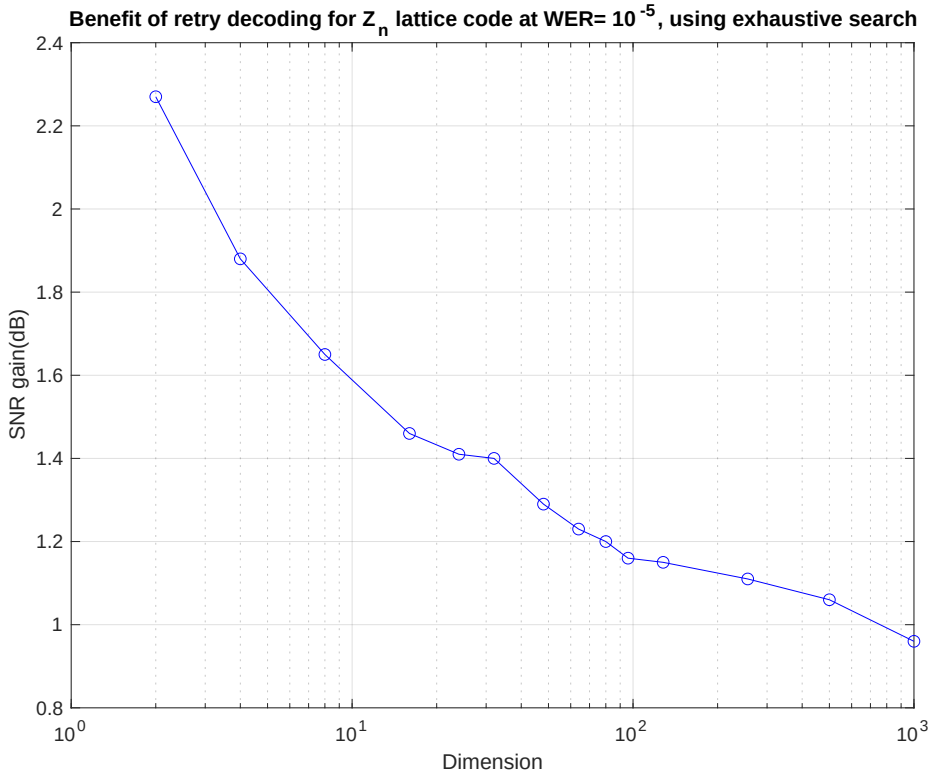


Figure 4.9: SNR gain of Z_n lattice codes at WER= 10^{-5}, between one-shot decoding using α_{MMSE} and genie-aided exhaustive search decoding.

A single-user lattice decoder estimates a desired linear combination $\sum_{i=1}^{L} a_{i,j}\mathbf{x}_i \bmod \Lambda_s$ by

$$\hat{\mathbf{y}}_j = DEC_{\Lambda_c}(\alpha_j \mathbf{y}_j) \bmod \Lambda_s, \tag{4.33}$$

with $a_{i,j} \in \mathbb{Z}$ and $\alpha_j \in \mathbb{R}$. Then relays forward $\hat{\mathbf{y}}_j$ and $\mathbf{a}_j = [a_{1,j}, a_{2,j}, \cdots, a_{L,j}]^T$, for $j = 1, 2, \cdots, J$, to the destination. The users' messages $\hat{\mathbf{b}}_1, \hat{\mathbf{b}}_2, \cdots, \hat{\mathbf{b}}_L$ can be recovered by solving linear equations, if and only if the integer coefficient matrix $([\mathbf{a}_1, \mathbf{a}_2 \cdots, \mathbf{a}_J] \bmod q)$ has rank L over $\mathbb{F}_q$.

The original CF [30] considers a power-constrained relay with mod Λ_s decoding. This restricts the uncoded message and decoding operations to a prime-size finite field $\mathbb{F}_q$, which limits flexibility on lattice code design. In [39] and [42], authors investigated an alternative CF relaying strategy, called incomplete-CF (ICF) in [42], where the relay is power unconstrained with the desired linear combination being $\sum_{i=1}^{L} a_{i,j}\mathbf{x}_i$. The linear combination is estimated without mod Λ_s in (4.33) as follows:

$$\hat{\mathbf{y}}_j = DEC_{\Lambda_c}(\alpha_j \mathbf{y}_j). \tag{4.34}$$

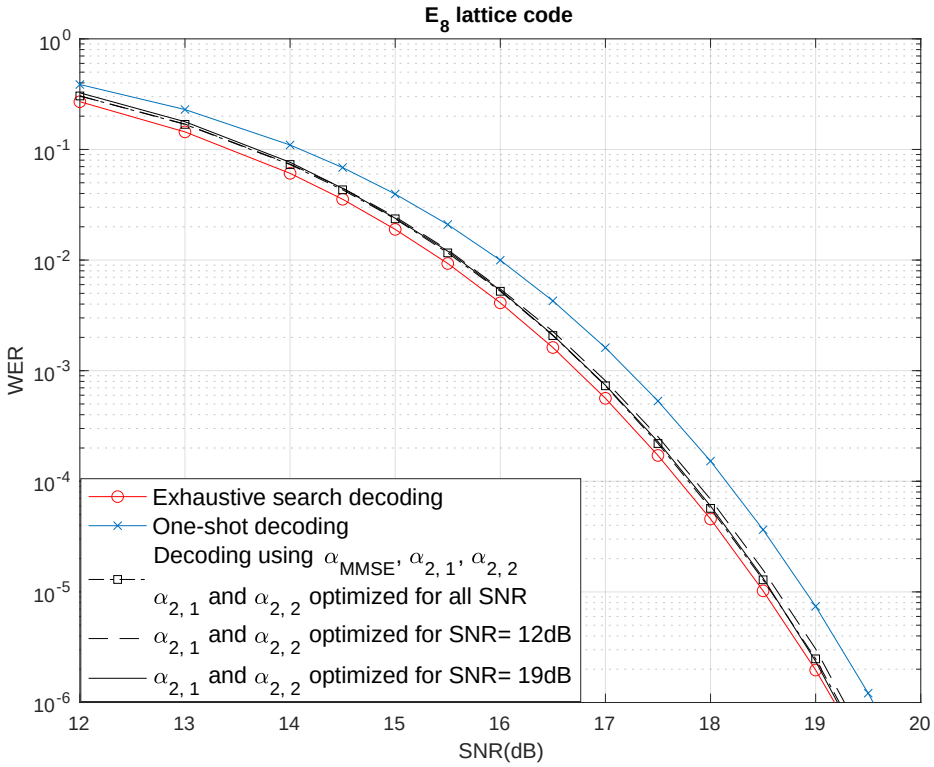


Figure 4.10: $E8$ lattice code with mismatch between SNR and decoding candidates $\alpha_{2,1}$ and $\alpha_{2,2}$. Error detection is genie-aided.

Under ICF, the destination only requires $[\mathbf{a}_1, \mathbf{a}_2 \cdots, \mathbf{a}_J]$ to have rank L in the real number field. Since the operation is over real numbers, the ICF scheme allows uncoded messages $\mathbf{b} \in \mathbb{Z}^n$ and gives more flexibility on lattice code design to match system requirements of error performance and complexity.

4.3.1.1 Coefficient search

The coefficient set $\{\mathbf{a}, \alpha\}$ of CF relaying is selected to maximize the achievable rate of the system, which is called the computation rate and is defined as follows.

Definition 4.2: (Computation rate) The computation rate is the maximum of achievable rate with given $\mathbf{h}$ and $\mathbf{a}$ over $\alpha \in \mathbb{R}$, defined as

$$R_c(\mathbf{h}, \mathbf{a}) = \max_{\alpha \in \mathbb{R}} \frac{1}{2} \log^+ \left(\frac{P}{\alpha^2 \sigma^2 + P \|\alpha \mathbf{h} - \mathbf{a}\|^2} \right), \tag{4.35}$$

where $\log^+(x) \doteq \max(\log(x), 0)$.

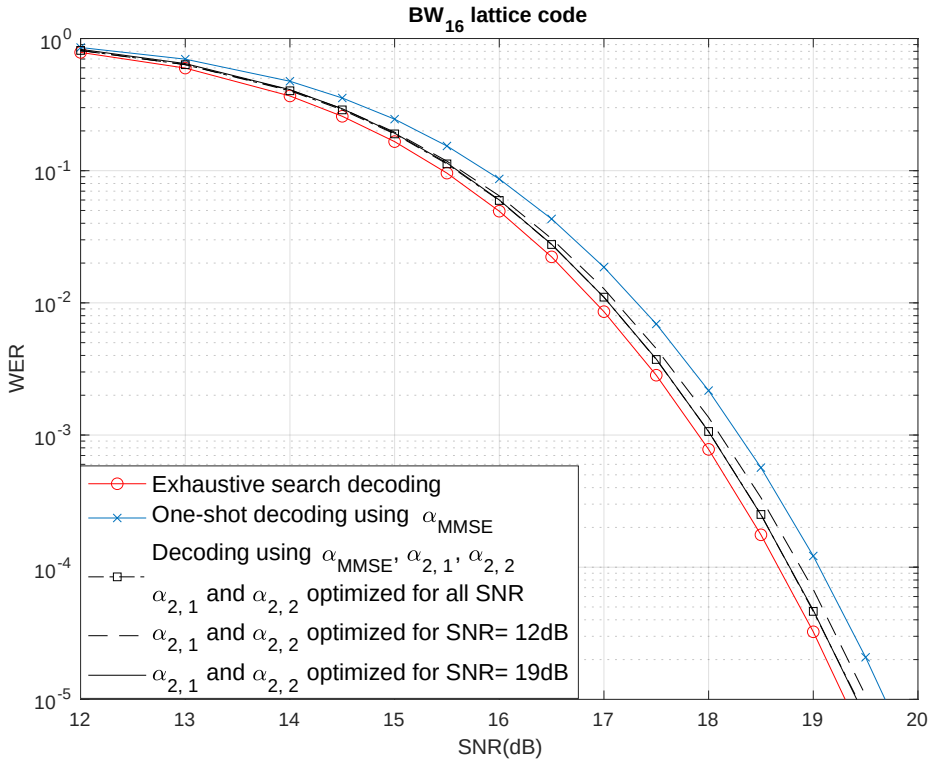


Figure 4.11: BW_{16} lattice code with mismatch between SNR and decoding candidates $\alpha_{2,1}$ and $\alpha_{2,2}$. Error detection is genie-aided.

The maximization in (4.35) can be uniquely obtained by choosing the MMSE coefficient as [30, Theorem 2]:

$$\alpha_{opt} = \frac{P\mathbf{h}^T\mathbf{a}}{\sigma^2 + P\|\mathbf{h}\|^2}. \tag{4.36}$$

Substituting (4.36) to (4.35), we have

$$R_c(\mathbf{h},\mathbf{a}) = \frac{1}{2}\log^+\left(\left(\|\mathbf{a}\|^2 - \frac{P(\mathbf{h}^T\mathbf{a})^2}{\sigma^2 + P\|\mathbf{h}\|^2}\right)^{-1}\right), \tag{4.37}$$

from which the values of $\mathbf{a}$ are selected as

$$\mathbf{a}_{opt} = \arg\max_{\mathbf{a}\in\mathbb{Z}^L} R_c(\mathbf{h},\mathbf{a}), \tag{4.38}$$

The integer coefficients $\mathbf{a}$ are restricted by

$$0 < \|\mathbf{a}\|^2 < \sigma^2 + \|\mathbf{h}\|^2 P \tag{4.39}$$

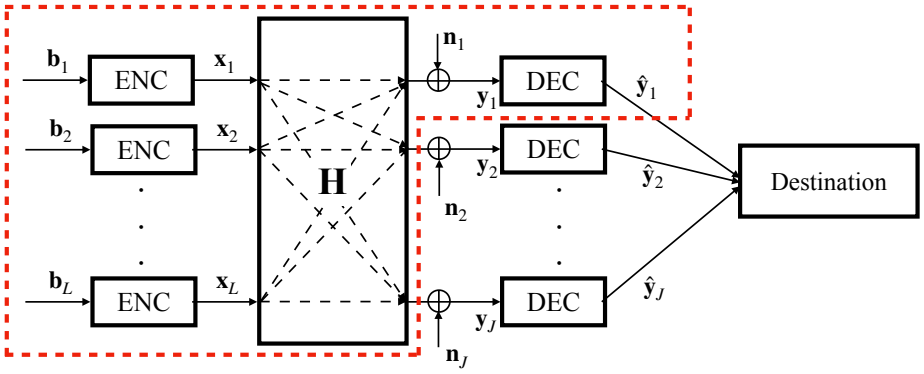


Figure 4.12: System model of multiple access network compute-forward with L users and J relays.

to have non-zero computation rate [30, Lemma 1].

The most common strategy to find the coefficient set is exhaustive search. For a given $\mathbf{h}$, a search space of $\mathbf{a}$, denoted as $\mathcal{S}_a$, is initialized according to (4.39). The receiver finds an integer coefficient vector $\mathbf{a}$ that maximizes (4.38), and compute the corresponding MMSE α using (4.36). However, the complexity of exhaustive search grows exponentially with the number of users. It has been shown that finding the optimal coefficient is equivalent to solve lattice SVP [29], which is NP-hard to find the exact solution. There exists algorithms to reduced the complexity of exhaustive search, such as applying lattice reduction [61,62], sphere decoding [63–65], quantized exhaustive search [66] and half integer [67,68].

4.3.2 System model

In this dissertation, a 2-hop network using ICF is considered. In a CF network, lattice decoding is performed at the relay node, therefore, we concentrate on the user-relay links and the error performance at one of relay nodes, as indicated by the red frame in Figure 4.12. The relay index j is omitted for simplicity in later discussions. The transmitter and channel models follow that described in the Section 4.3.1. The total channel gain is normalized to be $\|\mathbf{h}\| = 1$, by which the SNR at receiver is a constant as $\text{SNR} = P/\sigma^2$. At the receiver, the ICF scheme aims to decode a linear combination of messages as

$$\sum_{i=1}^{L} a_i \mathbf{x}_i, \tag{4.40}$$

without mod Λ_s. The equation error rate (EER) at one of relay nodes, that is $\hat{\mathbf{y}} \neq \sum_{i=1}^{L} a_i \mathbf{x}_i$, is measured for evaluation.

4.3.3 Decoding scheme

With a give coefficient set $\{\mathbf{a}, \alpha\}$, the received message in (4.32) is scaled by α at decoder as:

$$\alpha \mathbf{y} = \sum_{i=1}^{L} a_i \mathbf{x}_i + \sum_{i=1}^{L} (\alpha h_i - a_i)\mathbf{x}_i + \alpha \mathbf{n}, \tag{4.41}$$

from which the equivalent noise is defined as

$$\tilde{\mathbf{n}} = \sum_{i=1}^{L} (\alpha h_i - a_i)\mathbf{x}_i + \alpha \mathbf{n}. \tag{4.42}$$

A decoding error happens if $\tilde{\mathbf{n}} \notin \mathcal{V}(\mathbf{0})$.

Assume the current coefficient set $\{\mathbf{a}, \alpha\}$ failed decoding. Now we have freedom to change $\mathbf{a}$ and/or α for retry decoding so that the new equivalent noise $\tilde{\mathbf{n}}' \in \mathcal{V}(\mathbf{0})$ to reduce error rate. Two possible schemes are considered:

1. select a new integer coefficient $\mathbf{a}'$, compute its MMSE α' using (4.36);
2. keep integer coefficient $\mathbf{a}$ unchanged, select a new α';

4.3.3.1 Scheme 1

For scheme 1, a k-level retry decoding is considered using a length-k coefficient list as $\{\mathbf{a}_1, \alpha_1\}, \{\mathbf{a}_2, \alpha_2\}, \cdots, \{\mathbf{a}_k, \alpha_k\}$. The list is generated and sorted using computation rate (4.37) satisfying

$$R_c(\mathbf{h}, \mathbf{a}_1) \geq R_c(\mathbf{h}, \mathbf{a}_2) \geq \cdots \geq R_c(\mathbf{h}, \mathbf{a}_k). \tag{4.43}$$

For $j = 1, \cdots, k$, the scaling factor α_j is obtained by (4.36) using the corresponding $\mathbf{a}_j$. The decoding starts from $\{\mathbf{a}_1, \alpha_1\}$ and tests $\{\mathbf{a}_2, \alpha_2\}, \cdots, \{\mathbf{a}_k, \alpha_k\}$ sequentially, where error detection is performed after each decoding attempt. At the j-th attempt using $\{\mathbf{a}_j, \alpha_j\}$, the estimated linear combination $\hat{\mathbf{x}}_j = DEC_{\Lambda_c}(\alpha_j \mathbf{y})$ is obtained by the lattice decoder for Λ_c. The decoding terminates as soon as $\hat{\mathbf{x}}_j$ passes error detection or all candidates are tested. If all candidates fail, the decoder may output a decoding failure and request a retransmission without forwarding the erroneous message into the network.

4.3.3.2 Scheme 2

Since CF relaying uses a single user lattice decoder and scheme 2 only changes α for retry decoding, it is similar to the retry decoding scheme for the SU scenario in Section 4.2.3. For a k-level retry decoding, the decoding coefficient list is given as $\{\mathbf{a}, \mathcal{A}_1, \cdots, \mathcal{A}_k\}$. And the decoding steps follow that given in Section 4.2.2.

4.3.4 Decoding coefficient search algorithm

4.3.4.1 Scheme 1

The exhaustive search is considered to find a decoding coefficient list for the scheme 1. For a given channel $\mathbf{h}$, the $\mathcal{S}_a$ is first initialized according to (4.39). Differing from only finding the optimal coefficient, a size reduction of $\mathcal{S}_a$ is performed before searching.

Lemma 4.2: For any $\mathbf{a}_i \in \mathcal{S}_a$, all $\mathbf{a}_j$'s that satisfy $\mathbf{a}_j = m\mathbf{a}_i$, with integer $m = -1$ or $|m| > 1$, does not improve error performance and are eliminated from $\mathcal{S}_a$.

Proof. Recall that α is obtained by (4.36). For $m = -1$, it is trivial that $\mathbf{a}_j = -\mathbf{a}_i$ and $\alpha_j = -\alpha_i$ resulting in equal error performance. For $|m| > 1$, we have $\mathbf{a}_j = m\mathbf{a}_i$ and $\alpha_j = m\alpha_i$. Within a same transmission, messages $\mathbf{x}_i$, for $i = 1, 2, \cdots, L$, are unchanged. The equivalent noise for $\{\mathbf{a}_i, \alpha_i\}$ and $\{\mathbf{a}_j, \alpha_j\}$ satisfy $\tilde{\mathbf{n}}_j = m\tilde{\mathbf{n}}_i > \tilde{\mathbf{n}}_i$, while the direction of noise vector does not change. If $\tilde{\mathbf{n}}_i \notin \mathcal{V}(\mathbf{0})$, then $\tilde{\mathbf{n}}_j \notin \mathcal{V}(\mathbf{0})$ is also satisfied. Therefore, we have $\Pr(\tilde{\mathbf{n}}_j \notin \mathcal{V}(\mathbf{0})) \geq \Pr(\tilde{\mathbf{n}}_i \notin \mathcal{V}(\mathbf{0}))$, that is the decoding error probability using $\{\mathbf{a}_j, \alpha_j\}$ is never smaller than that using $\{\mathbf{a}_i, \alpha_i\}$. □

Using the size reduced search space $\mathcal{S}_a$, an exhaustive search is performed to first test all $\mathbf{a} \in \mathcal{S}_a$ using (4.38) and select the integer coefficients $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k$ with the k greatest computation rates. The corresponding $\alpha_1, \alpha_2, \cdots, \alpha_k$ are computed using (4.36) to generate the decoding coefficient list $\{\mathbf{a}_1, \alpha_1\}, \{\mathbf{a}_2, \alpha_2\}, \cdots, \{\mathbf{a}_k, \alpha_k\}$. It is noticed that, compared to only finding the optimal $\{\mathbf{a}_1, \alpha_1\}$, a sorting process is required to find multiple coefficients. The additional complexity is minimal compared to the exhaustive search which may have exponential complexity in the number of users.

For systems with large number of users, exhaustive search may have unacceptable complexity. The coefficient list searching could be implemented more efficiently by converting the maximization given in (4.38) into a lattice shortest independent vector problem (SIVP). It is notice that the

denominator of (4.38) can be expressed as

$$\|\mathbf{a}\|^2 - P(\mathbf{h}^T\mathbf{a})^2(\sigma^2 + P\|\mathbf{h}\|^2)^{-1} \tag{4.44}$$

$$=\mathbf{a}^T\left(\mathbf{I} - P\mathbf{h}(\sigma^2 + P\mathbf{h}^T\mathbf{h})^{-1}\mathbf{h}^T\right)\mathbf{a} \tag{4.45}$$

$$=\mathbf{a}^T\left(\mathbf{I} + \frac{P\mathbf{h}\mathbf{h}^T}{\sigma^2}\right)^{-1}\mathbf{a} \tag{4.46}$$

$$=\|\mathbf{D}^{-1/2}\mathbf{V}\mathbf{a}\|^2, \tag{4.47}$$

where $\mathbf{D}$ and $\mathbf{V}$ are obtained from the eigendecomposition $\mathbf{V}\mathbf{D}\mathbf{V}^T = \mathbf{I} + \frac{P\mathbf{h}\mathbf{h}^T}{\sigma^2}$. Since $\mathbf{a} \in \mathbb{Z}^n$, (4.47) can be seen as the Euclidean norm of a lattice point generated by $\mathbf{D}^{-1/2}\mathbf{V}$. Then (4.38) is equivalent to minimizing (4.47). The derivation technique follows [35], where integer-forcing MIMO using lattice codes are considered. Now, it can be seen that finding linearly independent $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k \in \mathcal{S}_a$ is equivalent to find k shortest linearly independent vectors from a lattice generated by $\mathbf{D}^{-1/2}\mathbf{V}$, which is also known as the lattice SIVP. Such connection is also examined at [29]. The original SIVP problem is NP-hard. However, at finite dimension, it could be implemented by lattice reduction [36, 61] or lattice list sphere decoding [63–65].

Note that the Lemma 4.2 only requires two distinct $\mathbf{a}_i$ and $\mathbf{a}_j$ are *pairwise linearly independent*. Meanwhile, solving SIVP gives up to L *fully linearly independent* vectors, which is a stronger condition than Lemma 4.2. This implies solving SIVP may eliminate $\mathbf{a}$, satisfying Lemma 4.2 but not satisfying SIVP, from the coefficient list.

4.3.4.2 Scheme 2

As we mentioned in Section 4.3.3.2, the decoding coefficient search for scheme 2 also follows that used for the SU scenario in Section 4.2.3. Note that Lemma 4.2 does not apply to scheme 2, since the direction of the equivalent noise vector changes for different α's. For example, suppose $\alpha' = m\alpha$, the equivalent noise of α' is

$$\begin{aligned}\tilde{\mathbf{n}}' &= \sum_{i=1}^{L}(\alpha' h_i - a_i)\mathbf{x}_i + \alpha'\mathbf{n} \\ &= \sum_{i=1}^{L}(m\alpha h_i - a_i)\mathbf{x}_i + m\alpha\mathbf{n} \\ &= m\tilde{\mathbf{n}} + (m-1)\sum_{i=1}^{L} a_i\mathbf{x}_i,\end{aligned} \tag{4.48}$$

where $\tilde{\mathbf{n}}'$ is not a scaled version of $\tilde{\mathbf{n}}$. The two terms in (4.48) may have different directions so that vector cancellation may happen resulting a smaller equivalent noise even for $|m| > 1$.

4.3.5 Discussions

Next, discussions related to retry decoding for CF relaying are given, including the lower bound on error probability and a comparison between scheme 1 and scheme 2.

4.3.5.1 Lower bound on error probability

It is natural to ask if we can obtain a lower bound on error probability for CF relaying as we did for the SU scenario. Unfortunately, it is still an open question to give such a lower bound for scheme 1.

However, for scheme 2 with fixed channel, it is possible to apply Theorem 4.1 in Section 4.2.4. For given channel $\mathbf{h}$ and integer coefficient $\mathbf{a}$, by Theorem 5 and Lemma 8 of [30], as dimension $N \to \infty$, the density function of the equivalent noise (4.42) is upper bounded by an i.i.d. zero-mean Gaussian distribution whose variance approaches

$$\sigma_{eq}^2 = P\|\alpha\mathbf{h} - \mathbf{a}\|^2 + \alpha\sigma^2, \tag{4.49}$$

where P and σ^2 are per-dimensional message and noise power. The scaled message in (4.41) can then be considered as an integer linear combination plus Gaussian noise:

$$\alpha\mathbf{y} = \underbrace{\sum_{i=1}^{L} a_i\mathbf{x}_i}_{\text{message}} + \underbrace{\sum_{i=1}^{L}(\alpha h_i - a_i)\mathbf{x}_i + \alpha\mathbf{n}}_{\text{additive Gaussian noise}}. \tag{4.50}$$

Theorem 4.1 can be applied with $P_{\mathbf{x}} = P\|\mathbf{a}\|^2$ and $\sigma^2 = \sigma_{eq}^2$. Note that we omitted random dithering in this dissertation for simplicity of notation. The linear combination and the equivalent noise in (4.50) is not independent due to the presence of message $\mathbf{x}_i$ in both parts. To ensure independence for analysis, random dithering, suggested in [14] and [30], must be implemented.

4.3.5.2 Scheme 1 vs scheme 2

In the previous subsection, we gave decoding schemes for CF relaying, where the decoder has freedom to change $\mathbf{a}$ and/or α. It is noticed that the optimal α obtained in (4.36) depends on the channel coefficient $\mathbf{h}$. This implies that

a decoding coefficient list for scheme 2 should be generated for each $\mathbf{h}$, which might be impractical if the channel is random and time variant. Therefore, scheme 2 is only suitable for assuming a fixed channel; while scheme 1 works for both fixed channel and random channel.

Recall the equivalent noise $\tilde{\mathbf{n}}$ in (4.42). Except the scaled Gaussian noise $\alpha\mathbf{n}$, another term $\sum_{i=1}^{L}(\alpha h_i - a_i)\mathbf{x}_i$, referred to as the integer approximation error, is nonnegligible in CF relaying. Since large α amplifies the Gaussian noise, it is desired to use a relatively small α by which $\alpha\mathbf{h}$ approximates to an integer vector $\mathbf{a}$. Retry decoding is to use new coefficient set to reduce $\tilde{\mathbf{n}}$ by reducing either $\alpha\mathbf{n}$ or $(\alpha\mathbf{h} - \mathbf{a})$.

Given a channel $\mathbf{h}$, we say a "good" approximation exists if a "small" integer approximation error can be achieved by using a "small" MMSE α. First, suppose a good approximation exists for a given $\mathbf{h}$ with a coefficient set $\{\mathbf{a}, \alpha\}$. Retry using a different $\mathbf{a}'$ may significantly increase the integer approximation error; and having no guarantee to reduce the scaled Gaussian noise to overcome the integer approximation error. If good approximation does not exist, the decoder needs to select a coefficient set $\{\mathbf{a}, \alpha\}$ with small integer approximation error but large α; or large integer approximation error but small α, where either contributes to the equivalent noise $\tilde{\mathbf{n}}$ in a different way. This simultaneously changes both the length and direction of integer approximation error term $\sum_{i=1}^{L}(\alpha h_i - a_i)\mathbf{x}_i$, and the length of the scaled Gaussian noise $\alpha\mathbf{z}$ with direction unchanged. In this case, changing $\mathbf{a}$ and α as a set explores more possibilities of $\tilde{\mathbf{n}}$ to correct more errors. It is noticed that for scheme 1, changing the direction of integer approximation error may lead to a significantly change of equivalent noise due to a possible vector cancellation against the Gaussian noise, which can be seen as a relatively aggressive strategy. On the other hand, scheme 2 is relatively conservative since the equivalent noise is changed continuously as α is continuous over the real number space.

Intuitively, we expect that scheme 2 is more suitable for retry decoding than scheme 1 if a good approximation exists for a given $\mathbf{h}$; otherwise scheme 1 is more suitable.

4.3.6 Numerical results

Numerical evaluations are given for fixed channel scenario and random channel scenario, respectively. For the fixed channel scenario, a comparison between scheme 1 and scheme 2 is given by assuming a two-user relay with different channel values using BW_{16} lattice codes as with in Figure 4.7. For the random channel scenario, the channel values are time variant and randomly generated for each message. The numerical evaluation is given

using construction D polar lattice codes with dimension $n = 128$ and 256, designed in [22]. The genie-aided error detection is assumed.

4.3.6.1 Fixed channel

Consider two-user case with normalized channel gain $\|\mathbf{h}\| = 1$. Let the fixed channel have coefficients $\mathbf{h}_1 = [0.6095, 0.7928]^T$ and $\mathbf{h}_2 = [0.4299, 0.9029]^T$, where $\mathbf{h}_1$ has a poor approximation with $h_{1,1}/h_{1,2} \approx 1/1.3$ and $\mathbf{h}_2$ has a good approximation with $h_{2,1}/h_{2,2} \approx 1/2.1$. Suppose SNR=30dB and Gaussian noise variance $\sigma^2 = 1$. The optimal and second-best coefficient set for $\mathbf{h}_1$ are $\{[3, 4]^T, 4.9946\}$ and $\{[1, 1]^T, 1.4009\}$; for $\mathbf{h}_2$ are $\{[1, 2]^T, 2.2334\}$ and $\{[2, 5]^T, 5.3688\}$, respectively. Compute the equivalent noise variance using $N_e = \alpha^2\sigma^2 + P\|\alpha\mathbf{h} - \mathbf{a}\|^2$ [30], where P is given in (3.21). For $\mathbf{h}_1$, $N_{e,1,opt} \approx 28.5230$ and $N_{e,1,sec} \approx 35.5625$; for $\mathbf{h}_2$, $N_{e,2,opt} \approx 6.8416$ and $N_{e,2,sec} \approx 147.1523$. When scheme 1 is applied, for $\mathbf{h}_1$, due to the poor integer approximation, the optimal coefficient set gives larger N_e than $\mathbf{h}_2$. However, the increase of N_e for $\mathbf{h}_1$ is less significant when the second-best coefficient set is applied for retry. Figure 4.13 and Figure 4.14 illustrate the EER for $\mathbf{h}_1$ and $\mathbf{h}_2$ using BW_{16} lattice code as used in Figure 4.7. It is observed that for $\mathbf{h}_1$, scheme 1 achieves larger gain than scheme 2; for $\mathbf{h}_2$, the gain obtained by scheme 1 is negligible, while scheme 2 achieves a gain, which is same as that of $\mathbf{h}_1$. This justifies the discussion we give in the Section 4.3.5.2.

4.3.6.2 Random channel

Figure 4.15 illustrates the EER performance at a CF relay using scheme 1, assuming random and time variant channel model. Polar lattice codes with dimension $n = 128, 256$ are applied as the coding lattices with hypercube shaping to form the lattice code, denoted as $\mathcal{C}$. The coding lattice design and lattice decoder structure follows [22]. The decoding algorithm of component polar codes are the successive cancellation (SC) decoding. The ICF scheme is applied and the number of users is 2. The messages $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$ are sent through a Rayleigh fading channel, with an additional constraint on total channel gain as $\|\mathbf{h}\| = \sqrt{h_1^2 + h_2^2} = 1$ in order to keep a constant received SNR. The received message $\mathbf{y} = h_1\mathbf{x}_1 + h_2\mathbf{x}_2 + \mathbf{n}$ is decoded using a single user lattice decoder following ICF and retry decoding using scheme 1. Since the number of users in this numerical evaluation is only 2, exhaustive search is performed to find the decoding coefficient list. The maximum number of decoding attempts is set to be 2 and 3. It is observed that gains of approximately 1.51 dB and 1.18 dB are achieved at a target EER = 10^{-5},

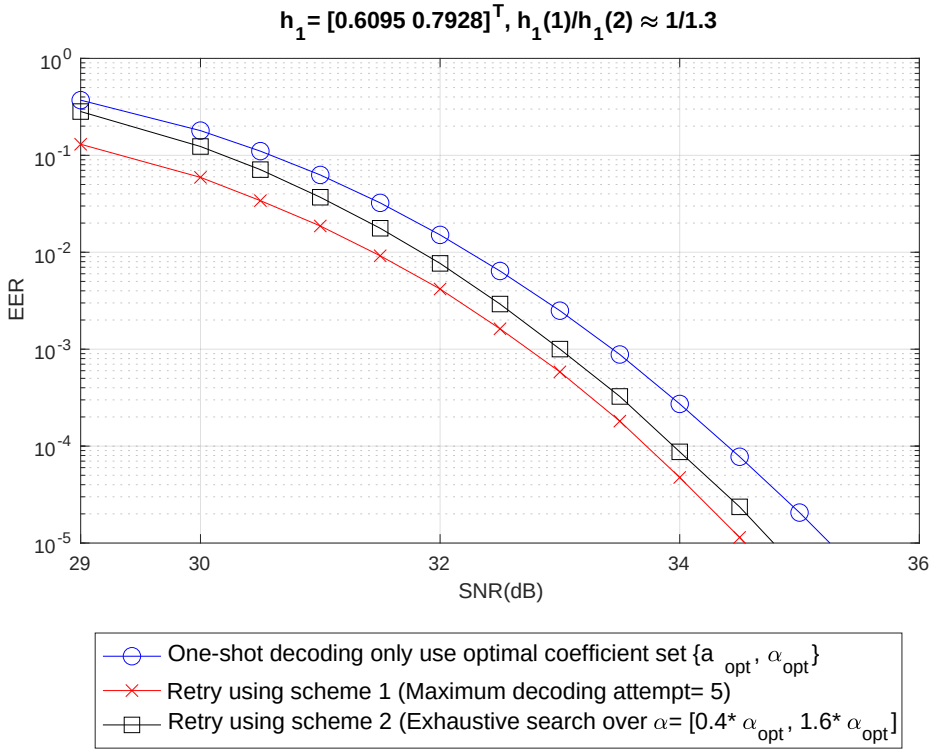


Figure 4.13: Comparison of EER performance for $\mathbf{h}_1 = [0.6095, 0.7928]^T$. Maximum number of decoding attempts is set to be sufficient large for each scheme. BW_{16} lattice code is applied as used in Figure 4.7 and error detection is genie aided.

by adding only one decoding attempt with genie-aided error detection. For lattice codes we applied here, no further gain is observed when giving the third decoding attempt.

4.4 Summary of this chapter

In this chapter, the retry decoding scheme for SU scenario and CF relaying are given, by which lower error rate can be achieved if the decoder is allowed to retry by adjusting the value of decoding coefficient(s). The decoder is assumed to be genie-aided to provide a perfect error detection.

For SU scenario, we first give procedure on retry using a decoding coefficient list $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_k$, which is sorted based on reliability. A subset $\mathcal{A}_i$ contains 2^{i-1} α's, for $i = 1, 2, \cdots, k$. An algorithm is introduced to generate such list using exhaustive search, from which the list only depends

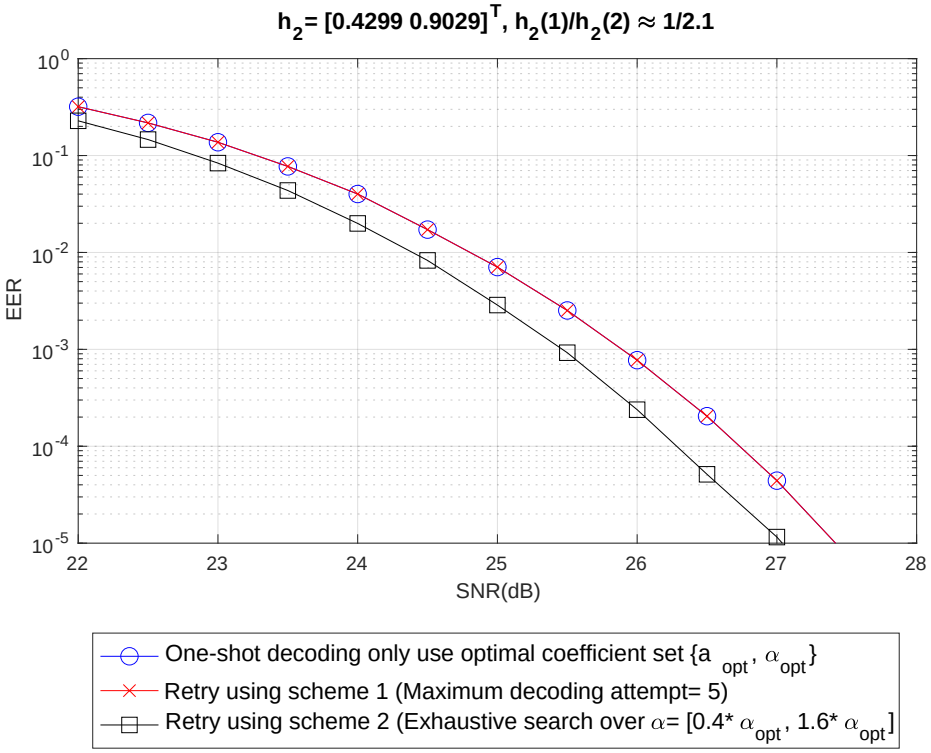


Figure 4.14: Comparison of EER performance for $\mathbf{h}_2 = [0.4299, 0.9029]^T$. Maximum number of decoding attempts is set to be sufficient large for each scheme. BW_{16} lattice code is applied as used in Figure 4.7 and error detection is genie aided.

on SNR values and could be generated offline. A lower bound on error probability is derived for the proposed retry decoding scheme by extending the decoding coefficient list to all $\alpha \in \mathbb{R}$, using covering sphere of the coding lattice. Similarly, an effective sphere estimate is considered to give an estimate on error probability using exhaustive search decoding. Discussions are given for

- the tightness of the lower bound and the accuracy of the effective sphere estimate are related to the covering thickness and NSM of the coding lattice, respectively,
- the benefit of retry decoding might decrease as the dimension of lattices increasing, due to the property of Gaussian noise.

In numerical simulations, gains are achieved using low dimensional lattice codes. Since the lattice codes for numerical simulations have the best-known quantizers among lattice having the same dimension, the effective sphere

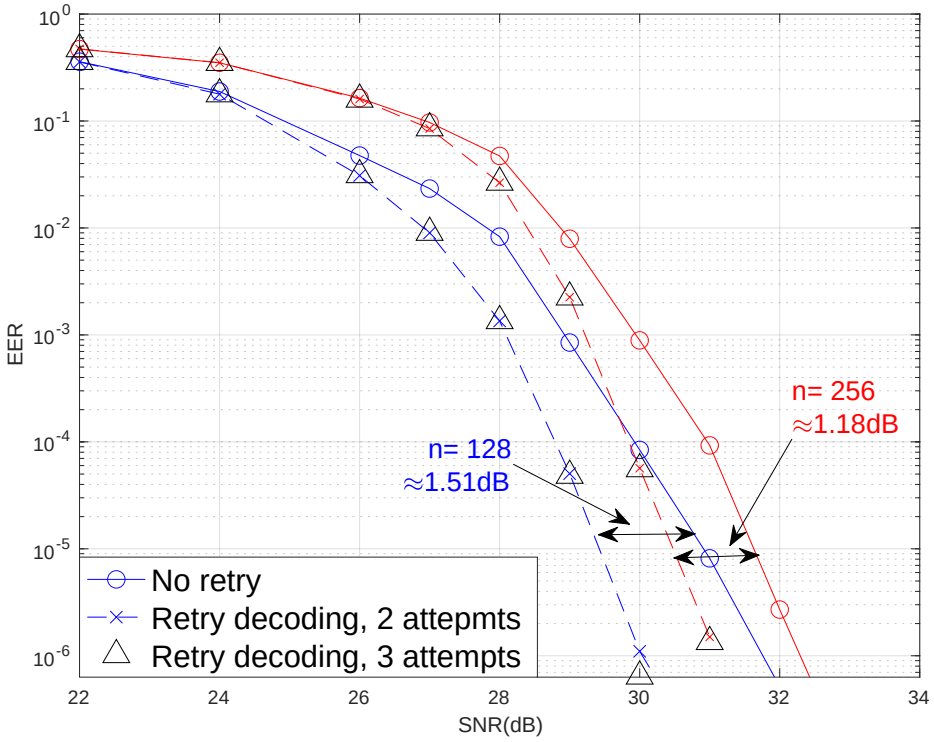


Figure 4.15: EER performance for $n = 128, 256$ polar lattice codes using retry decoding. The maximum number of decoding attempts is 2.

estimate is shown to be accurate. The WER of exhaustive search decoding can be approached by only applying a 2-level retry decoding using $\mathcal{A}_1, \mathcal{A}_2$ (with maximum 3 decoding attempts), if the decoding coefficients list is generated appropriately. A numerical example using Z_n lattice codes for n up to 1000 is given, which justifies the benefit of retry decoding decreases as the dimension increases as we discussed above. At last, a modified retry strategy is considered for SU scenario, where, instead of the SNR-dependent coefficient list, decoding coefficients are found for one SNR and applied across all SNRs for retry to reduce the searching cost, from which performance loss due to the mismatch between α's and SNRs is negligible when the α's are found for a high SNR value.

For CF relaying using ICF, 2 different schemes for retry decoding are considered

- scheme 1: change decoding coefficient $\{\mathbf{a}, \alpha\}$ as a set for retry,
- scheme 2: fix the integer coefficient $\mathbf{a}$ and only change α for retry.

Since scheme 2 is similar to what used in SU scenario, the decoding coefficient

search algorithm, the lower bound and the effective sphere estimate of SU scenario can be modified and applied to CF relaying using the scheme 2. However, finding such a lower bound for scheme 1 is still an open question. For scheme 2, since the channels are assumed to be time variant and randomly generated for each message, decoding coefficient search algorithm has to be found for each $\mathbf{h}$ individually. It might be impractical to implement scheme 2 in the random channel scenario. Meanwhile, scheme 1 works for both fixed channel and random channel. A discussion is given to compare scheme 1 and scheme 2 assuming fixed channel with different channel coefficients. An intuitive explanation is given that

- if "small" integer approximation error $\|\alpha\mathbf{h} - \mathbf{a}\|$ can be achieved by a relatively "small" α, the gain of scheme 1 becomes small compared to that of scheme 2;
- if such approximation is hard to obtain, scheme 1 gives a larger gain.

In numerical simulations, we first verified the comparison between scheme 1 and scheme 2 for fixed channel discussed above. For the random channel scenario, polar lattice codes with larger dimension as $n = 128$ and 256 are considered to show the potential of retry decoding in a practical CF relaying system, where 1.51dB and 1.18dB gain are obtained at EER= 10^{-5} by only adding at most one more decoding attempt, respectively.

Due to the extra freedom on changing integer coefficient $\mathbf{a}$, a more significant improvement of error rate is observed for CF relaying compared to the SU scenario, implying the potential use of lattice codes with retry decoding in lattice based PLNC using CF or, similarly, IF MIMO.

Chapter 5

Lattice construction for self-error detection

5.1 Introduction of this chapter

In this chapter, we give a lattice construction which has physical layer error detection ability to enable the retry decoding introduced in Chapter 4. Error detection is enabled by embedding binary CRC codes into the least significant bits (LSB) of the lattice uncoded message, referred as the CRC-embedded lattice/lattice code. In general, it will be shown that any binary linear block code (LBC) is feasible for this construction, for which it can be extended to a boarder class referred as LBC-embedded lattice/lattice code. Since the purpose of this chapter is to study lattices/lattice codes with physical layer error detection, CRC codes are considered to give lattice designs, which have low complexity and are commonly used for error detection. The basics of CRC-embedded lattices/lattice codes are introduced in this chapter, along with applications for both the SU scenario and CF relaying, where the ICF scheme is assumed for CF relaying.

This chapter is divided into three parts. In the first part, we give the construction of CRC-embedded lattices and the corresponding CRC-embedded lattice codes. The CRC codeword is embedded into the LSB of the lattice uncoded message. It is shown that this embedding ensures linearity, so that the CRC-embedded lattice indeed forms a lattice. Generator matrix and encoding/decoding schemes of the CRC-embedded lattices/lattice codes are introduced. In particular, as we discussed in the end of last chapter, a stand-alone CF relay does not have error detection ability, since one linear combination contains multiple unknown users' messages. A condition on shaping lattice design is described, by which the CF relay is able to detect errors from a linear combination directly without knowing individual users' messages. Such shaping lattices can be easily adopted to construction A/D lattices.

The second part considers CRC optimization for the CRC-embedded

lattices to balance the rate loss due to the parity bits and the error detection capability, which affects the benefit of retry decoding. Optimization is considered for: 1) the CRC generator polynomial/position of parity bits with a fixed length l and 2) the CRC length. To evaluate the optimization result, the probability of undetected error for CRC-embedded lattices is defined, along with two methods to estimate this probability. One method considers the lattice structure to give a better estimate, while the other gives the estimate only considering the CRC length, which is convenient for high dimensional lattices whose structure is unknown. It is found that increasing CRC length gives more significant improvement on the probability of undetected error than optimizing CRC polynomial. To maximize the gain obtained by retry decoding, a semi-analytical algorithm on optimizing the CRC length is given, by which simulation across all possible CRC lengths is not required.

The third part gives numerical simulations to implement the CRC-embedded lattice codes with retry decoding for SU scenario and CF relaying, where actual CRC codes are deployed for error detection. A discussion of the trade-offs related to implementation is given, including lattice dimension, code rate and the choice of lattices for each scenario. Then error performance and gain obtained by CRC-embedded lattice codes with retry decoding are illustrated for SU scenario and CF relaying with the optimized CRC length.

This chapter combines the proposed retry decoding and CRC-embedded lattice codes, which shows the potential of implementation in practical lattice based transmission. When CRC codes are applied for error detection, the gain for CF relaying is more significant than that of SU scenario, as we have shown using genie-aided error detection. The gain of retry decoding increases as the code rate increases, implying the potential use of high rate lattice codes in systems using high order modulations.

5.2 CRC-embedded lattices/lattice codes

The construction is first given with respect to a lattice and then extended to a lattice code. In this chapter, we consider the CRC bits are only embedded in the coding lattice.

The notations for CRC codes are first given. Denote $\mathcal{C}_{CRC}$ as a CRC code. In many papers (or books), CRC refers only to the padded parity bits. To define the CRC-embedded lattice, $\mathcal{C}_{CRC}$ is considered as an (n, k) binary linear block code, where the information bits are also involved. We denote the generator polynomial of a CRC code as $G_{CRC}(x)$, which may also be written in binary representation or in hexadecimal representation using the

reversed reciprocal format with the same notation, and a generator matrix of a CRC code as $\mathbf{G}_{CRC}$. Example 5.1 and Example 5.2 give examples for the format of $G_{CRC}(x)$ and $\mathbf{G}_{CRC}$ considered in this chapter.

Example 5.1: This example is about the CRC generator polynomial $G_{CRC}(x)$. Suppose a CRC-6 code has generator polynomial

$$G_{CRC}(x) = x^6 + x + 1. \tag{5.1}$$

The binary representation is the coefficient of $x^6, x^5, \cdots, 1$ given as:

$$G_{CRC}(x) = [1, 0, 0, 0, 0, 1, 1]. \tag{5.2}$$

The hexadecimal representation is obtained from the binary representation using the reversed reciprocal format. First group every 4 bits in (5.2), exclude the last bit, from less significant bit, then convert to hexadecimal value as

$$\underbrace{1, 0}_{2}, \underbrace{0, 0, 0, 1}_{1}, 1 \tag{5.3}$$

The hexadecimal representation is given as

$$G_{CRC}(x) = 0\text{x}21. \tag{5.4}$$

Equation (5.1), (5.2) and (5.4) might be used equivalently to indicate the generator polynomial of CRC codes.

Example 5.2: This example is about the CRC generator matrix $\mathbf{G}_{CRC}$ by considering a CRC-l code $\mathcal{C}_{CRC}$ as an (n, k) linear block code, with number of information sequence being $k = n - l$ bits. Suppose $\mathcal{C}_{CRC}$ has generator $G_{CRC}(x)$. Given an information sequence $\mathbf{u} \in \mathbb{F}_2^k$, CRC parity bits $\mathbf{p}$ are calculated using polynomial $G_{CRC}(x)$. The CRC codeword $\mathbf{c} \in \mathcal{C}_{CRC}$ can then be formed as:

$$\mathbf{c} = \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix}. \tag{5.5}$$

The generator matrix is formed by k linear independent CRC codeword $\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_k$ as:

$$\mathbf{G}_{CRC} = [\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_k]. \tag{5.6}$$

5.2.1 Lattice construction

Let Λ be a lattice with a generator matrix $\mathbf{G}$ and $\mathbf{b}$ be the uncoded message. The LSB vector is defined as follows.

Definition 5.1: (LSB vector) The LSB vector of a uncoded message $\mathbf{b}$ is obtained as

$$\mathbf{b}_{LSB} = \mathbf{b} \bmod 2. \tag{5.7}$$

The constellation of CRC-embedded lattice is formed by restricting the LSB vector as a CRC codeword. The lattice Λ before embedding CRC is referred as the *base lattice* to distinguish from the CRC-embedded lattice.

Definition 5.2: (CRC-embedded lattice) Let Λ be an n-dimensional *base lattice* with a generator matrix $\mathbf{G}$, and $\mathcal{C}_{CRC}$ be a binary CRC code with block length n, the constellation Λ' after embedding $\mathcal{C}_{CRC}$ into Λ is defined as:

$$\Lambda' = \{\, \mathbf{Gb} \mid \mathbf{b} \in \mathbb{Z}^n, \mathbf{b}_{LSB} \in \mathcal{C}_{CRC} \,\}. \tag{5.8}$$

Although Definition 5.2 does not explicitly show that Λ' forms a lattice, Theorem 5.1 states that Λ' is indeed a lattice by showing that Λ' forms an additive subgroup as given in Definition 2.1.

Theorem 5.1: The constellation Λ' given by Definition 5.2 is a sublattice of its base lattice Λ.

Proof. First, $\Lambda' \subseteq \Lambda$ is straightforward from (5.8) as the domain of uncoded messages of Λ' is a subset of that of Λ.

By Definition 2.1, Λ' is a lattice if Λ' forms an additive subgroup in $\mathbb{R}^n$ that has: a) identity element; b) inverse element; c) associativity; d) commutativity; e) closure.

Let $\mathbf{x} \in \Lambda'$ and the corresponding uncoded message be $\mathbf{b} = \mathbf{G}^{-1}\mathbf{x}$. The *identity element* is $\mathbf{0} \in \Lambda'$ because the LSB vector of the all-zero vector is always a codeword of $\mathcal{C}_{CRC}$. For the *inverse element*, given $\mathbf{x} \in \Lambda'$, $-\mathbf{x}$ has same LSB vector as $\mathbf{x}$ under modulo 2, by which $-\mathbf{x} \in \Lambda'$. *Associativity* and *commutativity* are trivial because the addition operation of lattice points is over the real number space. *Closure* is obtained by the linearity of $\mathcal{C}_{CRC}$. Let $\mathbf{x}_1, \mathbf{x}_2 \in \Lambda'$, $\mathbf{b}_1 = \mathbf{G}^{-1}\mathbf{x}_1$ and $\mathbf{b}_2 = \mathbf{G}^{-1}\mathbf{x}_2$. The LSB vector of $\mathbf{b}_1 + \mathbf{b}_2$ is:

$$\begin{aligned}(\mathbf{b}_1 + \mathbf{b}_2) \bmod 2 =& (\mathbf{b}_1 \bmod 2 + \mathbf{b}_2 \bmod 2) \bmod 2 \\ =& (\mathbf{b}_{1,LSB} + \mathbf{b}_{2,LSB}) \bmod 2,\end{aligned} \tag{5.9}$$

where $\mathbf{b}_{1,LSB}$ and $\mathbf{b}_{2,LSB}$ are the LSB of $\mathbf{b}_1$ and $\mathbf{b}_2$. By the linearity of $\mathcal{C}_{CRC}$, the LSB vector of $\mathbf{b}_1 + \mathbf{b}_2$ is also in $\mathcal{C}_{CRC}$. Therefore, $\mathbf{x}_1 + \mathbf{x}_2 \in \Lambda'$. This concludes the proof. □

5.2.1.1 Generator matrix of Λ'

Since Λ' is a lattice, a generator matrix $\mathbf{G}'$ is found using the generator matrix of the base lattice Λ and binary code $\mathcal{C}_{CRC}$. By Definition 5.2, the uncoded message $\mathbf{b}$ in (5.8) can be written as $\mathbf{c}_{CRC} + 2\mathbf{z}$, where $\mathbf{c}_{CRC} \in \mathcal{C}_{CRC}$ and $\mathbf{z} \in \mathbb{Z}^n$. This indicates that such $\mathbf{b}$ is a lattice point obtained by construction A using $\mathcal{C}_{CRC}$ as $\Lambda_a = \mathcal{C}_{CRC} + 2Z_n$.

Proposition 5.1: Let base lattice Λ have a generator matrix $\mathbf{G}$ and binary code $\mathcal{C}_{CRC}$ have a lower triangular generator matrix $\mathbf{G}_{CRC} = \begin{bmatrix} \mathbf{T} \\ \mathbf{P} \end{bmatrix}$, where $\mathbf{T}$ is a $k \times k$ lower triangular matrix. A generator matrix of the CRC-embedded lattice Λ' is:

$$\mathbf{G}' = \mathbf{G} \begin{bmatrix} \mathbf{T} & \mathbf{0}_{k\times(n-k)} \\ \mathbf{P} & 2\mathbf{I}_{n-k} \end{bmatrix}, \tag{5.10}$$

where $\mathbf{0}_{k\times(n-k)}$ is $k\times(n-k)$ all-zero matrix and $\mathbf{I}_{n-k}$ is the $(n-k)$-dimensional identity matrix.

Proof. For $\mathcal{C}_{CRC}$ having a lower triangular $\mathbf{G}_{CRC}$, a generator matrix of construction A lattice $\Lambda_a = \mathcal{C}_{CRC} + 2Z^n$ is given as [25]:

$$\mathbf{G}_a = \begin{bmatrix} \mathbf{T} & \mathbf{0}_{k\times(n-k)} \\ \mathbf{P} & 2\mathbf{I}_{n-k} \end{bmatrix}. \tag{5.11}$$

The uncoded message satisfying the condition in (5.8) can then be expressed as $\mathbf{b} = \mathbf{G}_a\mathbf{b}'$ with $\mathbf{b}' \in \mathbb{Z}^N$, from which the lattice point of Λ' is $\mathbf{x} = \mathbf{G}\mathbf{G}_a\mathbf{b}'$. By Definition 2.1, a generator matrix of Λ' is given as $\mathbf{G}' = \mathbf{G}\mathbf{G}_a$. □

5.2.1.2 Embedding CRC codes with block length $n' \leq n$

The CRC code considered in Definition 5.2 has block length n. It is also feasible to consider a CRC code $\mathcal{C}'_{CRC}$ with block length $n' \leq n$, so that the constraint is only given to a subsequence of the LSB vector. Denote a set of index as $\mathcal{I} = \{1, 2, \cdots, n\}$ and a subset $\mathcal{I}' \subseteq \mathcal{I}$ with cardinality $|\mathcal{I}'| = n' \leq n$. Let $\mathbf{b}_{LSB,\mathcal{I}'}$ be a subsequence of $\mathbf{b}_{LSB}$ whose indices is specified by $\mathcal{I}'$. Use $\mathcal{C}'_{CRC}$ and $\mathcal{I}'$, we can define the CRC-embedded lattice in a more general way, where only a subsequence of the LSB vector is constrained by a CRC code.

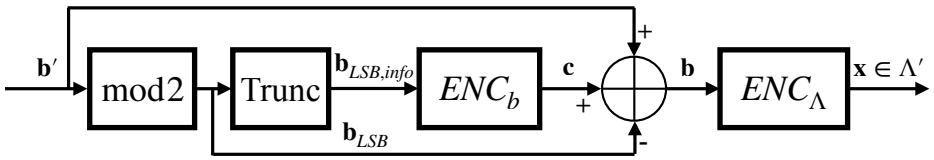


Figure 5.1: Encoder model.

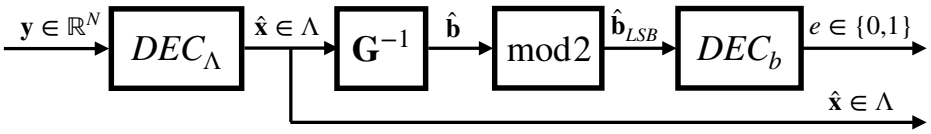


Figure 5.2: Decoder model.

Definition 5.3: (CRC-embedded lattice with partial constraint) Let $\mathcal{C}'_{CRC}$ be a binary CRC code with block length $n' \leq n$. Let $\mathcal{I}'$ be a subset of $\mathcal{I} = \{1, 2, \cdots, n\}$ indicating the indices that are constrained by $\mathcal{C}'_{CRC}$. The CRC-embedded lattice with respect to $\mathcal{C}'_{CRC}$ is given as:

$$\Lambda' = \{ \mathbf{Gb} \mid \mathbf{b} \in \mathbb{Z}^n, \mathbf{b}_{LSB,\mathcal{I}'} \in \mathcal{C}'_{CRC} \}. \tag{5.12}$$

It shall be seen that Definition 5.3 is equivalent to Definition 5.2 when $\mathcal{I}' = \mathcal{I}$. Meanwhile, it is clear that the Λ' given in Definition 5.3 also satisfies Theorem 5.1, i.e. Λ' in (5.12) is also a lattice. Without loss of generality, we consider the CRC-embedded lattices as defined in Definition 5.2 in the rest of this chapter, unless stated otherwise.

5.2.2 Encoding and decoding scheme

Next, encoding and decoding schemes are introduced to implement the CRC-embedded lattices. The encoder and decoder models are shown in Figure 5.1 and Figure 5.2. Embedding $\mathcal{C}_{CRC}$ is equivalent to removing some lattice points from the base lattice; however, the receiver uses the base lattice for decoding followed by a parity check of $\mathcal{C}_{CRC}$. ENC_Λ/DEC_Λ and $\mathbf{G}^{-1}$ corresponds to the base lattice Λ, not the CRC-embedded lattice Λ', and ENC_b/DEC_b are the binary encoder/parity check of $\mathcal{C}_{CRC}$ for error detection. Denote $\mathcal{I}_{info}$ as a set of indices of $\mathbf{b}_{LSB}$ indicating the information bits for encoding $\mathcal{C}_{CRC}$.

At the encoder side, the user's message $\mathbf{b}' = [b'_1, b'_2, \cdots, b'_n]^T$ before

embedding $\mathcal{C}_{CRC}$ satisfies

$$\begin{cases} b'_i \in \mathbb{Z}, & i \in \mathcal{I}_{info} \\ b'_i \in 2\mathbb{Z}, & i \notin \mathcal{I}_{info}. \end{cases} \tag{5.13}$$

The LSB vector $\mathbf{b}_{LSB}$ is obtained by mod 2. The *Trunc* function truncates $\mathbf{b}_{LSB}$ into $\mathbf{b}_{LSB,info}$ according to $\mathcal{I}_{info}$ to generate the input of binary encoder ENC_b. Then, an $\oplus$ operation combines $\mathbf{b}', \mathbf{b}_{LSB}$ and the binary codeword $\mathbf{c} \in \mathcal{C}_{CRC}$ to obtain the lattice uncoded message as $\mathbf{b} = \mathbf{b}' + \mathbf{c} - \mathbf{b}_{LSB}$, where the addition and subtraction are in real number space.

At the decoder side, since the lattice decoder DEC_Λ uses Λ instead of Λ', a decoding algorithm for Λ' does not need to be specified and the estimate is $\hat{\mathbf{x}} \in \Lambda$. The parity check DEC_b following gives a 1-bit pass/fail output e indicating if $\hat{\mathbf{b}}_{LSB} \in \mathcal{C}_{CRC}$ and $\hat{\mathbf{x}} \in \Lambda'$.

5.2.3 Lattice codes using CRC-embedded lattices

Forming lattice codes using CRC-embedded lattices is straightforward. In this paper, the CRC code $\mathcal{C}_{CRC}$ is only embedded into the coding lattice, that is, with respect to a base lattice code $\mathcal{C} = \Lambda_c/\Lambda_s$, the CRC-embedded lattice code is $\mathcal{C}' = \Lambda'_c/\Lambda_s$. Consider the rectangular encoding described in Section 3.2 with the diagonal matrix $\mathbf{M}$ defined in (3.7). The i-th component of uncoded message of $\mathcal{C}'$ is defined as:

$$\begin{cases} b_i \in \mathcal{M}_i, & i \in \mathcal{I}_{info} \\ b_i \in \{b | b \in \mathcal{M}_i, b \bmod 2 = 0\}, & i \notin \mathcal{I}_{info} \end{cases} \tag{5.14}$$

where $\mathcal{I}_{info}$ is set of indices of $\mathbf{b}_{LSB}$ indicating the information bits for encoding $\mathcal{C}_{CRC}$ and $\mathcal{M}_i = \{0, 1, \cdots, M_i - 1\}$ for $i = 1, 2, \cdots, n$. The encoding/decoding scheme follows Figure 5.1 and Figure 5.2 by additionally including the shaping operation.

Since CRC codes have linearity, the lattice points removed from Λ_c due to $\mathcal{C}_{CRC}$ are distributed over the whole constellation, rather than being concentrated within a certain area. This implies that the average codebook power of $\mathcal{C}'$ approximates to that of $\mathcal{C}$. The code rate is reduced since lattice codes only contain a finite number of bits per message. The loss of code rate is evaluated by the SNR penalty, defined as follows.

Definition 5.4: (SNR penalty) Let a base lattice code $\mathcal{C}$ have code rate R and $\mathcal{C}_{CRC}$ for embedding have l parity bits. The SNR penalty SNR_p of the CRC-embedded lattice code $\mathcal{C}'$ is:

$$\text{SNR}_p = 10 \log_{10} \frac{R}{R'} (\text{dB}), \tag{5.15}$$

where $R' = \frac{nR-l}{n}$ is the code rate of $\mathcal{C}'$.

5.2.3.1 An example of CRC embedded lattice/lattice code

A numerical example is given as following to illustrate the relationship among lattices and lattice codes before and after embedding CRC, respectively.

Example 5.3: Let the base lattice be A_2 lattice with

$$\mathbf{G} = \begin{bmatrix} \sqrt{3}/2 & 0 \\ 1/2 & 1 \end{bmatrix}. \tag{5.16}$$

Suppose CRC-1, which is also the single parity check code, is embedded into the base lattice. Figure 5.3 illustrates the constellation of Λ' and the corresponding lattice code with shaping lattice $4A_2$. For Λ', a generator matrix is given as:

$$\mathbf{G}' = \begin{bmatrix} \sqrt{3}/2 & 0 \\ 3/2 & 2 \end{bmatrix} \tag{5.17}$$

5.2.4 Shaping lattice design for CF relaying

This section gives a condition on design shaping lattice when applying the CRC-embedded lattice to CF relaying using ICF, which has been discussed in Section 4.3.2.

For the SU scenario, the shaping lattice does not affect the CRC embedding process, since the indexing function can recover the estimate of uncoded message $\hat{\mathbf{b}}$ using the shaping lattice. However, for CF relaying using ICF, the estimate $\hat{\mathbf{x}} = \sum_{i=1}^{L} a_i \mathbf{x}_i$ obtained by a stand-alone relay contains multiple unknowns $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_L$, which are unsolvable from a single linear combination. Additionally, the indexing function can not be applied to $\hat{\mathbf{x}}$, since $\hat{\mathbf{x}}$ is a member of coding lattice Λ_c but not necessarily a member of lattice code $\mathcal{C}$.

Instead, the estimate of the uncoded message corresponding to the linear combination $\hat{\mathbf{x}}$ is given as

$$\hat{\mathbf{b}} = \mathbf{G}_c^{-1} \hat{\mathbf{x}}, \tag{5.18}$$

and the LSB vector is obtained by $\hat{\mathbf{b}}_{LSB} = \hat{\mathbf{b}} \bmod 2$. The validity of the parity check of the users' uncoded messages $\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_L$, corresponding to $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_L$, needs to be preserved at $\hat{\mathbf{b}}_{LSB}$ after transmission.

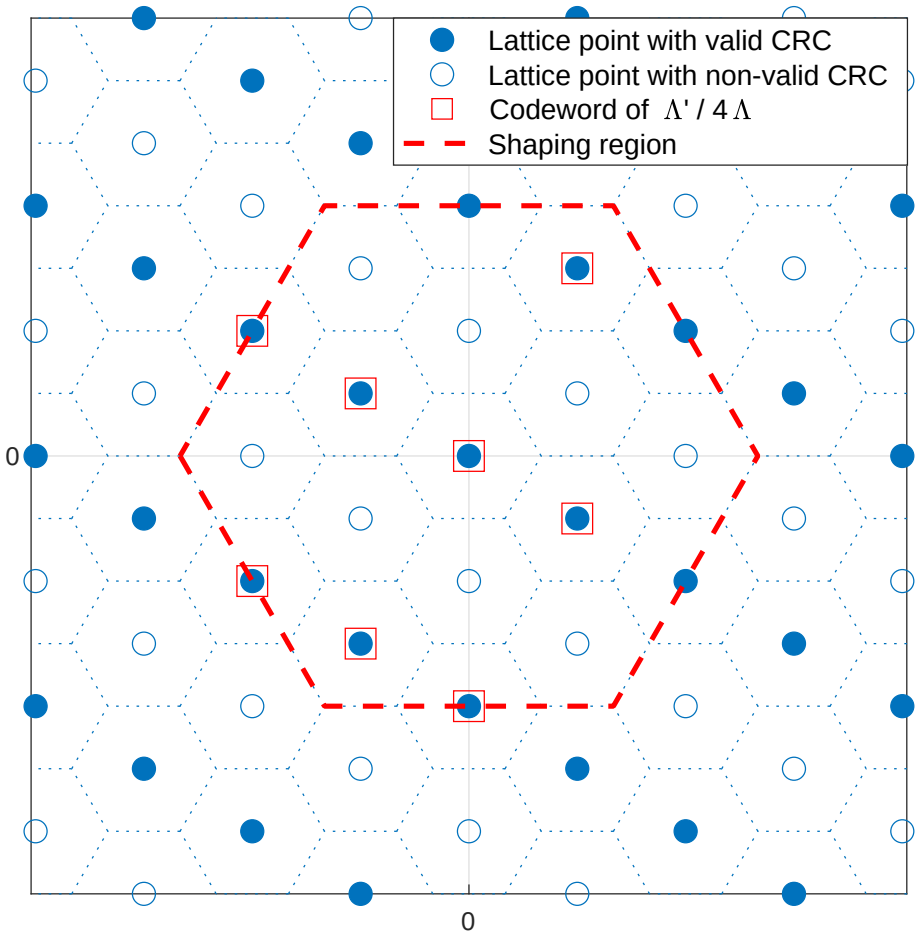


Figure 5.3: A2 lattice/lattice code with single parity check code embedded.

First, Lemma 5.1 shows the validity is preserved if the transmission is uncoded, that is linearly combining uncoded messages $\mathbf{b}_i$ directly.

Lemma 5.1: If $\mathbf{b}_i \bmod 2 \in \mathcal{C}_{CRC}$ for $i = 1, 2, \cdots, L$, then $\sum_{i=1}^{L} a_i \mathbf{b}_i \bmod 2 \in \mathcal{C}_{CRC}$ for arbitrary integers $a_1, a_2, \cdots, a_L$.

This lemma was justified for $L = 2$ and $a_1 = a_2 = 1$ when proving Theorem 5.1. The generalization to $L > 2$ and arbitrary integers $a_1, a_2, \cdots, a_L$ is straightforward.

Now, we consider the lattice coded case. Let $\mathcal{C} = \Lambda_c/\Lambda_s$, be the base lattice code before embedding binary code $\mathcal{C}_{CRC}$. Let $\mathbf{G}_c$ and $\mathbf{G}_s$ be generator matrices of Λ_c and Λ_s, respectively. By Lemma 3.1, there exists an $\mathbf{M} \in \mathbb{Z}^{n \times n}$ by which $\mathbf{G}_s = \mathbf{G}_c \mathbf{M}$.

Proposition 5.2: Let uncoded messages $\mathbf{b}_i$ have $\mathbf{b}_{i,LSB} \in \mathcal{C}_{CRC}$ and corresponding lattice codeword $\mathbf{x}_i$ for $i = 1, 2, \cdots, L$. The LSB vector $\hat{\mathbf{b}}_{LSB} = \left(\sum_{i=1}^{L} a_i \mathbf{G}_c^{-1} \mathbf{x}_i\right) \bmod 2 \in \mathcal{C}_{CRC}$ is satisfied for arbitrary $a_1, a_2, \cdots, a_L \in \mathbb{Z}$, if $\mathbf{M}$ only consists of even integers.

Proof. From (3.5), a lattice codeword $\mathbf{x}_i$ is encoded from $\mathbf{b}_i$ as:

$$\mathbf{x}_i = \mathbf{G}_c \mathbf{b}_i - Q_{\Lambda_s}(\mathbf{G}_c \mathbf{b}_i) = \mathbf{G}_c \mathbf{b}_i - \mathbf{G}_s \mathbf{s}_i, \tag{5.19}$$

where $\mathbf{s}_i \in \mathbb{Z}^n$ so that $Q_{\Lambda_s}(\mathbf{G}_c \mathbf{b}_i) = \mathbf{G}_s \mathbf{s}_i$. The LSB vector $\hat{\mathbf{b}}_{LSB}$ is:

$$\begin{aligned}
\hat{\mathbf{b}}_{LSB} &= \sum_{i=1}^{L} a_i \mathbf{G}_c^{-1} \mathbf{x}_i \bmod 2 \\
&= \left(\sum_{i=1}^{L} a_i \mathbf{b}_i - a_i \mathbf{M} \mathbf{s}_i \right) \bmod 2 \\
&= \left(\sum_{i=1}^{L} a_i \mathbf{b}_i \bmod 2 \right) \oplus \left(\sum_{i=1}^{L} a_i \mathbf{M} \mathbf{s}_i \bmod 2 \right),
\end{aligned} \tag{5.20}$$

where $\oplus$ is addition over the binary field. Lemma 5.1 shows that

$$\left(\sum_{i=1}^{L} a_i \mathbf{b}_i \bmod 2 \right) \in \mathcal{C}_{CRC}. \tag{5.21}$$

By linearity, $\hat{\mathbf{b}}_{LSB} \in \mathcal{C}_{CRC}$ if and only if

$$\left(\sum_{i=1}^{L} a_i \mathbf{M} \mathbf{s}_i \bmod 2 \right) \in \mathcal{C}_{CRC} \tag{5.22}$$

for arbitrary $a_i \in \mathbb{Z}$ and $\mathbf{s}_i \in \mathbb{Z}^N$ for $i = 1, 2, \cdots, L$. If all elements of $\mathbf{M}$ are even integers, $\left(\sum_{i=1}^{L} a_i \mathbf{M} \mathbf{s} \bmod 2\right)$ is the all-zero vector, indicating the membership of $\mathcal{C}_{CRC}$, which concludes the proof. □

Note the design of $\mathbf{M}$ given in Proposition 5.2 is valid only for ICF scheme but does not work for the original CF in [30]. It is noticed that the validity of CRC is preserved only when all elements of $\mathbf{M}$ are even integers. This implies the i-th dimension of the uncoded message is defined over $0, 1, \cdots, 2K$ for some integer K, which does not form a prime-size finite field required for the original CF.

In Section 5.2.1.2, we have discussed constructing CRC-embedded lattice using $\mathcal{C}_{CRC}$ with block length $n' < n$. For a CRC-embedded lattice Λ' given

by Definition 5.3, the constraint of the CRC code only exists at a subset of the LSB vector $\mathbf{b}_{LSB,\mathcal{I}'}$, specified by an index subset $\mathcal{I}'$. For such CRC-embedded lattice codes, only rows of $\mathbf{M}$ with index specified by $\mathcal{I}'$ affect the LSB vector. Therefore, Proposition 5.2 is modified as follows for CRC-embedded lattices constructed by Definition 5.3.

Corollary 5.1: Let $\mathcal{C} = \Lambda_c/\Lambda_s$ be the base lattice code, and Λ_c' be the CRC-embedded coding lattice obtained from Definition 5.3 with CRC code $\mathcal{C}_{CRC}'$ and index subset $\mathcal{I}'$. The LSB vector $\hat{\mathbf{b}}_{LSB,\mathcal{I}'}$ is a codeword of $\mathcal{C}_{CRC}'$ for arbitrary $a_1, a_2, \cdots, a_L \in \mathbb{Z}$, if rows of $\mathbf{M}$, with row index specified by $\mathcal{I}'$, only consists of even integers.

5.3 CRC optimization

As the key technique proposed in this chapter, CRC codes for embedding is studied in this section. It is noticed that CRC-aided error detection is not perfect. A better error detection capability, i.e. lower P_{ud}, implies more erroneous messages would be retried, from which a lower error rate could be achieved. In practice, a longer CRC usually provides better error detection capability, while resulting in a larger SNR penalty. Optimization of CRC codes is considered in this section with respect to: (1) the CRC polynomial/position of parity bits with a fixed length l, and (2) the CRC length.

5.3.1 Probability of undetected error

Error detection using finite dimensional $\mathcal{C}_{CRC}$ is imperfect and false positives may occur, that is for a transmitted $\mathbf{x} \in \Lambda'$, the event $\hat{\mathbf{x}} \in \Lambda'$ for $\hat{\mathbf{x}} \neq \mathbf{x}$. This event is referred to as an undetected error event which has probability P_{ud}. The analysis of probability P_{ud} is valid for evaluation of both CRC-embedded lattice Λ' and lattice code $\mathcal{C} = \Lambda'/\Lambda_s$, if the influence of codewords on codebook boundary could be ignored.

Definition 5.5: (Probability of undetected error event) Given a desired decoding result $\mathbf{x} \in \Lambda'$ and $\hat{\mathbf{x}} \neq \mathbf{x}$, the probability of an undetected error event is:

$$P_{ud} = \Pr(\hat{\mathbf{x}} \in \Lambda' | \hat{\mathbf{x}} \neq \mathbf{x}) = \frac{\Pr(\hat{\mathbf{x}} \in \Lambda', \hat{\mathbf{x}} \neq \mathbf{x})}{\Pr(\hat{\mathbf{x}} \neq \mathbf{x})}. \tag{5.23}$$

Due to the linearity of lattices, P_{ud} defined in (5.23) can be equivalently

expressed as:

$$P_{ud} = \frac{\sum_{\mathbf{e}\in\Lambda',\mathbf{e}\neq\mathbf{0}} p(\mathbf{e})}{\sum_{\mathbf{e}\in\Lambda,\mathbf{e}\neq\mathbf{0}} p(\mathbf{e})}, \tag{5.24}$$

where $\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$ is the quantized error vector with probability $p(\mathbf{e})$. Computing $p(\mathbf{e})$ exactly requires integrating the noise density function over $\mathcal{V}(\mathbf{e})$, which depends on the geometric properties of lattices and is impractical in most cases. Instead, we give two methods to estimate P_{ud} considering the shortest vectors of the base lattice Λ and the parity length of $\mathcal{C}_{CRC}$, respectively.

Method 1: This method considers shortest vector of a lattice and its kissing number. Denote $\mathcal{T}$ as the set of shortest non-zero vectors of Λ and $|\{\cdot\}|$ as the cardinality of set. The kissing number is the number of shortest non-zero vectors, that is $|\mathcal{T}|$. Assume the error vector $\mathbf{e}$ is uniformly distributed over $\mathcal{T}$, then P_{ud} can be estimated as:

$$P_{ud} \approx \frac{|\mathcal{T} \cap \Lambda'|}{|\mathcal{T}|}. \tag{5.25}$$

Method 2: This method considers the CRC length l to give an average value of P_{ud}. For $\mathcal{C}_{CRC}$ having l parity bits, a fraction of 2^{-l} lattice points in Λ have valid LSB vectors, which could cause an undetected error event. Then, P_{ud} can be estimated as:

$$P_{ud} \approx 1/2^l. \tag{5.26}$$

Method 1 is more suitable for low dimensional lattices for which the set $\mathcal{T}$ is either known or can be found by numerical techniques such as list sphere decoding. The assumption on the distribution of $\mathbf{e}$ in Method 1 follows the truncated union bound and circular symmetry of Gaussian noise, thus is good for medium to high SNR of the AWGN channel. For higher dimensional lattices with unknown kissing number, Method 2 gives a convenient estimate of P_{ud}.

5.3.2 Optimizing CRC with fixed length l

Given a fixed length l, this section considers CRC optimization of polynomials and the position of parity bits. Using the probability of undetected error, the Equ. (5.25) of Method 1 can be seen as a target function here. That is to find a CRC code $\mathcal{C}_{CRC}$ from the set $\mathcal{C}_{all}$, containing all possible CRC-l codes, such that:

$$\mathcal{C}_{CRC} = \arg\min_{\mathcal{C}_{all}} \frac{|\mathcal{T} \cap \Lambda'|}{|\mathcal{T}|}, \tag{5.27}$$

where the set $\mathcal{T}$ of lattice is required. Since the lattice shortest vector problem (SVP) is NP-hard in general, finding the set $\mathcal{T}$ may be impractical for high dimensional lattice. Therefore, this optimization is considered only for low dimensional lattices.

First, we consider CRC polynomial optimization. The BW_{16} lattice code with hypercube shaping, as with in Figure 4.7, is used to illustrate the relationship between CRC polynomials and P_{ud} given a fixed CRC length $l = 4, 5, 6$, as shown in Figure 5.4. BW_{16} lattice has 4320 shortest vectors, which can be found by list sphere decoding. The value of P_{ud} is estimated using Method 1, i.e. the ratio of lattice points with valid CRC among 4320 shortest vectors. Evaluation is given at SNR = 18.3dB, where the WER for one-shot decoding using α_{MMSE} is approximately 10^{-3}. All possible CRC-4, CRC-5 and CRC-6 codes are evaluated and the polynomials of CRC-5 are marked in the figure using hexadecimal representation. The average P_{ud} is calculated using Method 2 without considering the structure of the lattice, as $P_{ud} = 1/2^l$. The best CRC-l polynomial is selected to achieve the lowest P_{ud} among all possible CRC-l codes. For example, the best CRC-5 for BW_{16} lattice has polynomial $G_{CRC} = 0\text{x}15$, which has $P_{ud} \approx 0.0259$, and will be used as the CRC-5 polynomial in the CRC length optimization and implementation using BW_{16} lattices/lattice codes.

From Figure 5.4, following observations are obtained.

1. Embedding CRC-l removes a fraction of $(1 - 1/2^l)$ lattice points from the constellation, therefore CRC polynomials with a fixed length give P_{ud} close to $1/2^l$.
2. A slight lower P_{ud} can be achieved by appropriately selecting the polynomial.
3. The improvement obtained by optimizing polynomial may not be as large as to increase the CRC length by 1 bit.
4. There may exist a "good" CRC-l code that have P_{ud} as low as a "bad" CRC-$(l + 1)$ code. For example, a CRC-5 code with $G_{CRC,5} = 0\text{x}15$ and a CRC-6 code with $G_{CRC,6} = 0\text{x}2\text{C}$ have the similar $P_{ud} \approx 0.0259$.

On the other hand, changing position of parity bits also gives a different value of P_{ud} according to (5.25). However, the fraction of lattice points, which are removed from the constellation, is kept the same as $(1-1/2^l)$. Therefore, optimizing position of parity bits gives a similar result as we obtained for polynomial optimization.

To conclude above, optimizing CRC polynomial/position of parity bits requires the shortest vectors of a lattice, which may be impractical for large dimensional lattices. Therefore, the optimization in this subsection is only considered for low dimensional lattices. For large dimensional lattice,

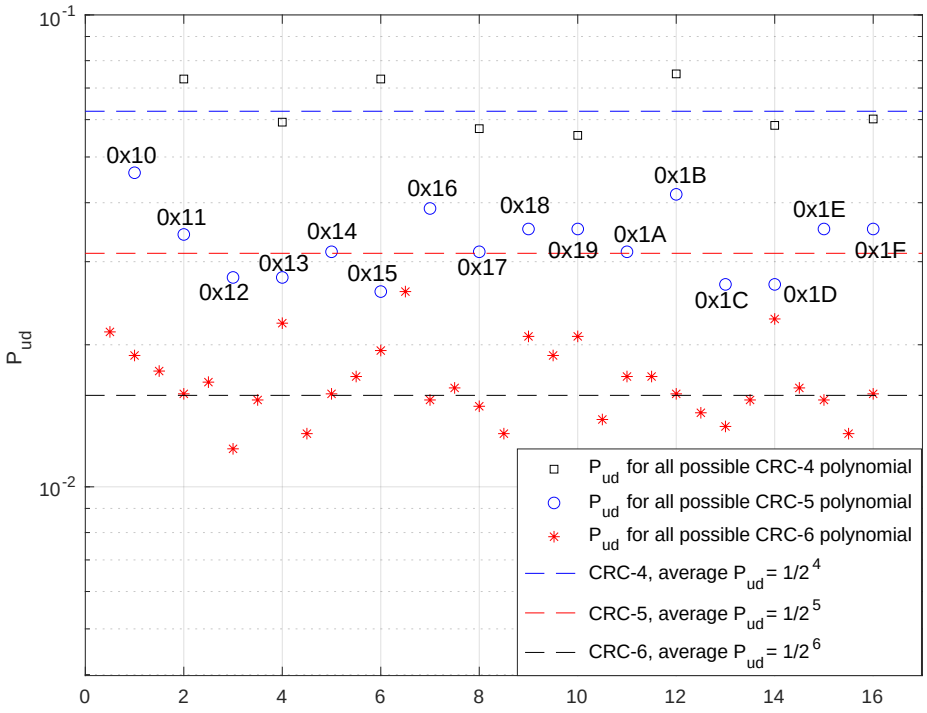


Figure 5.4: Probability of undetected error P_{ud} of all possible CRC codes with $l = 4, 5, 6$. The base lattice code is BW_{16} lattice code used in Figure 4.7.

CRC polynomials are selected randomly. In [69], authors suggested CRC polynomials for various lengths that are used in practical systems, though they are optimized for binary codes but not particularly for lattices.

5.3.3 Estimate the probability of undetected error

Table 5.1 evaluates the value of P_{ud} by Monte-Carlo simulation using (5.23) and the estimate by (5.25) and (5.26) using the same BW_{16} lattice code used in Figure 4.7, with CRC of various lengths embedded. The CRC polynomials for each length l are selected to further minimize P_{ud} by (5.27). The numerical results show that both (5.25) and (5.26) give a close estimate of P_{ud}. Since BW_{16} lattice code has known set $\mathcal{T}$, Method 1 gives a better estimate than Method 2. With the CRC polynomial optimization, a slightly lower P_{ud} is achieved compared with the average value using (5.26). However, it is observed that the value of P_{ud} is dominated by the length l, which agrees with the observation obtained from Figure 5.4.

Table 5.1: Evaluation of P_{ud} of CRC-embedded BW_{16} lattice codes and CRC length $l = 4, 5, 6, 7, 8$. All-zero codeword is assumed and SNR is set so that WER$\approx 10^{-3}$ for decoding using α_{MMSE}.

CRC length(l)		4	5	6	7	8
CRC polynomial(G_{CRC})		0xC	0x15	0x25	0x55	0x8A
P_{ud}	Monte-Carlo(5.23)	5.619e-2	2.625e-2	1.205e-2	4.201e-3	1.386e-3
	Kissing number(5.25)	5.556e-2	2.593e-2	1.204e-2	4.167e-3	1.389e-3
	Parity length(5.26)	6.250e-2	3.125e-2	1.563e-2	7.813e-3	3.906e-3

5.3.4 Optimization of CRC length

From Section 5.3.2 and Section 5.3.3, it can be seen that optimizing CRC length has more significant effect on P_{ud} comparing to optimizing CRC polynomial with a fixed length. This section gives a semi-analytic CRC length optimization to balance the trade-off between error detection capability and benefit of retry decoding to maximize the SNR gain, involving the SNR penalty, for a given target error rate. This optimization considers average probabilities of events happened during error detection and retry decoding, which is valid for both single user transmission and CF relaying.

The optimization consists of two steps: first, estimate the WER/EER after k-level retry decoding, denoted as $P_{e,total}^{(k)}$, as a function of CRC length l; then obtain the gain from the WER/EER curve using the estimated $P_{e,total}^{(k)}$ with the SNR penalty included. The optimal l gives the lowest SNR for a given target WER/EER. The $P_{e,total}^{(k)}$ is first derived as a function of the probability of undetected error P_{ud} defined in (5.23):

$$f : P_{ud} \to P_{e,total}^{(k)}. \tag{5.28}$$

Using (5.24)-(5.26), define a mapping between CRC length l and P_{ud}, $g : l \to P_{ud}$. Then $P_{e,total}^{(k)}$ given in (5.28) is further written as:

$$f \circ g : l \to P_{e,total}^{(k)}. \tag{5.29}$$

Notations for deriving $f : P_{ud} \to P_{e,total}^{(k)}$ are defined as follows. Let $P_e^{(i)}$ be the word error probability at the i-th decoding level, for $i = 1, 2, \cdots, k$. After decoding, a CRC check splits these error events into detected and undetected errors with probability $P_{e,CRC}^{(i)}$ and $P_{e,ud}^{(i)}$, respectively, where events with probability $P_{e,ud}^{(i)}$ are not retried in future decoding. The total word error

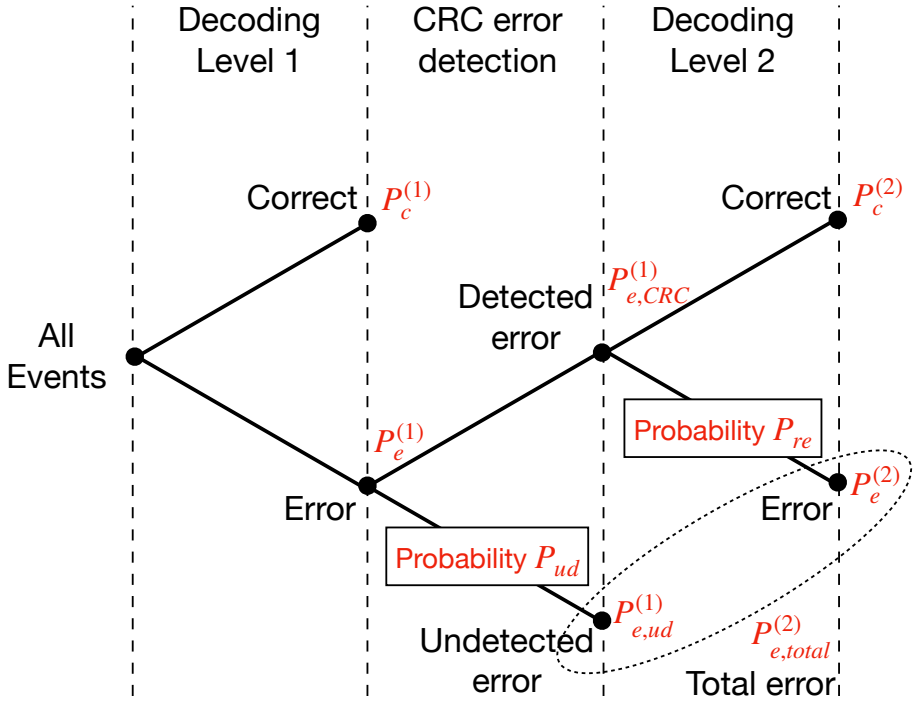


Figure 5.5: Events and probabilities for retry decoding with 2 levels.

probability $P_{e,total}^{(k)}$ for $k > 1$ is:

$$P_{e,total}^{(k)} = \sum_{i=1}^{k-1} P_{e,ud}^{(i)} + P_e^{(k)}, \tag{5.30}$$

due to the mutually exclusivity of events. For $k = 1$, $P_{e,total}^{(1)} = P_e^{(1)}$ is the word error probability of the one-shot decoder. For $i = 2, 3, \cdots, k$, another new term $P_{re}^{(i)}$ is defined to indicate the word error probability at the i-th decoding level given a detected error from the $(i-1)$-th level. Figure 5.5 illustrates an example of the structure of events and corresponding probabilities for a 2-level decoding.

Proposition 5.3: For decoding with $k > 1$ levels, given $P_e^{(1)}$ and $P_{re}^{(i)}$, for $i = 2, 3, \cdots, k$, of the CRC-embedded lattice code, the function $f : P_{ud} \to P_{e,total}^{(k)}$ is expressed as:

$$P_{e,total}^{(k)} = \sum_{i=1}^{k-1} \left(P_e^{(i)} P_{ud} \right) + P_{re}^{(k)} P_e^{(k-1)} (1 - P_{ud}), \tag{5.31}$$

where $P_e^{(i)}$ for $i > 1$ is obtained recursively as:

$$P_e^{(i)} = P_{re}^{(i)} P_e^{(i-1)} (1 - P_{ud}). \tag{5.32}$$

Proof. By the structure shown in Figure 5.5, $P_{e,CRC}^{(i)}$ can be estimated on average as $P_{e,CRC}^{(i)} = P_e^{(i)}(1 - P_{ud})$, by which $P_e^{(i)}(i > 1)$ can be recursively obtained as:

$$P_e^{(i)} = P_{re}^{(i)} P_{e,CRC}^{(i)} = P_{re}^{(i)} P_e^{(i-1)} (1 - P_{ud}). \tag{5.33}$$

Similarly, we have $P_{e,ud}^{(i)} = P_e^{(i)} P_{ud}$. Substituting $P_{e,ud}^{(i)}$ and $P_e^{(i)}$ into (5.30), it is obtained that:

$$P_{e,total}^{(k)} = \sum_{i=1}^{k-1} \left(P_e^{(i)} P_{ud} \right) + P_{re}^{(k)} P_e^{(k-1)} (1 - P_{ud}). \tag{5.34}$$

With given $P_e^{(1)}$ and $P_{re}^{(i)}$ for $i = 2, 3, \cdots, k$, we can see (5.34) as a function of P_{ud}, that is, $f : P_{ud} \to P_{e,total}^{(k)}$. □

The CRC length optimization is a semi-analytical method because estimating using f requires numerically evaluated $P_e^{(1)}$ and $P_{re}^{(i)}$, $i = 2, 3, \cdots, k$. Fortunately, the values of $P_e^{(1)}$ and $P_{re}^{(i)}$, for $i = 2, 3, \cdots, k$ depend on the value of decoding coefficients but are independent of the embedded CRC code, thus numerical evaluation over different CRC polynomials and/or length l is not needed. To justify this, recall that the lattice decoder uses the base lattice but not the CRC-embedded lattice. First, $P_e^{(1)}$ is the word error probability of the conventional one-shot decoding so that it is independent of the embedded CRC code. The lowest $P_e^{(1)}$ is achieved by selecting the optimal decoding coefficient, such as α_{MMSE} for single user transmission. Second, $P_{re}^{(i)}$ is the transition probability between *a detected error* with probability $P_{e,CRC}^{(i-1)}$ and *a decoding error of next decoding attempt* with probability $P_e^{(i)}$, see Figure 5.5 for $i = 2$. It can be seen that the measure of $P_{re}^{(i)}$ starts from an error that is already detected; passes through the retry decoding using the base lattice decoder; then ends at an error event before error detection which includes both detected and undetected errors at the i-th error detection. Since no knowledge of the CRC code is involved during this transition, the probability $P_{re}^{(i)}$ is independent of the embedded CRC codes, as well as the CRC length l. Similar to $P_e^{(1)}$, the value of $P_{re}^{(i)}$ also depends on the choice of retry decoding coefficient, see Figure 4.3 as an example of $P_{re}^{(2)}$ for the SU scenario using E_8 lattice code.

Numerically, $P_{re}^{(i)}$ can be evaluated using

$$P_{re}^{(i)} = P_{e}^{(i)} / P_{e,CRC}^{(i-1)} \tag{5.35}$$

by embedding arbitrary CRC code for error detection, or equivalently using

$$P_{re}^{(i)} = P_{e}^{(i)} / P_{e}^{(i-1)} \tag{5.36}$$

by assuming genie-aided error detection with $P_{ud} = 0$.

By letting $P_{ud} = 0$, the estimate of $P_{e,total}^{(k)} = P_{re}^{(k)} P_{e}^{(k-1)}$ indicating the word error probability using genie-aided error detection, which gives a lower bound of WER/EER for k-level retry decoding with no SNR penalty included.

5.4 Discussions

5.4.1 Trade-offs on implementing CRC-embedded lattice code and retry decoding

In the previous section, we addressed the trade-off between CRC error detection capability and SNR penalty, from which a CRC length optimization was given. Here we introduce some other trade-offs related to implementation of the proposed CRC-embedded lattices/lattice codes.

5.4.1.1 Lattice dimension n vs. SNR penalty SNR_p

The first trade-off is between lattice dimension and SNR penalty for fixed CRC length l and code rate R of the base lattice code $\mathcal{C} = \Lambda_c/\Lambda_s$. The SNR penalty in (5.15) can be written as $SNR_p = 10\log_{10}(1 + l/(nR - l))$. For lattice codes with different dimension n but the same rate R, the SNR penalty decreases when n increases. Also, from the definition of SNR_p, the increase of SNR penalty is more significant for low dimensional lattice codes when adding 1 CRC bit. This means for low dimensional lattice codes, a short CRC length is preferred to achieve small SNR penalty by sacrificing error detection capability. When n increases, longer CRC can be applied to have better error detection capability without increasing the SNR penalty too much.

5.4.1.2 Code rate R vs. improvement of retry decoding

The second trade-off is between code rate R of the base lattice code $\mathcal{C} = \Lambda_c/\Lambda_s$ and the improvement of retry decoding for fixed dimension n and

CRC length l. When code rate R increases, the ratio of the number of CRC parity bits over total number of bits decrease, so that SNR penalty is reduced. This implies that for an n-dimensional lattice code, one possible direction to increase the improvement of retry decoding is to increase the code rate by expanding the shaping region.

5.4.1.3 Lattice dimension n vs. improvement of retry decoding

Another notable trade-off is between lattice dimension and improvement of retry decoding. From theoretical perspective, as $n \to \infty$, there exists a lattice code that achieves the Gaussian channel capacity by using α_{MMSE} [14] in the SU scenario and a lattice code achieves Poltyrev's bound by using coefficient set $\{\mathbf{a}, \alpha\}$, which maximizes the computation rate [30], in the CF relaying. This implies the improvement of retry decoding asymptotically goes to 0 as $n \to \infty$.

For the SU scenario, the only noise component is the additive white Gaussian noise. For finite dimensional noise, as n increases, it is known that the probability density of Gaussian noise is concentrated in a thin annulus with radius $\sqrt{n\sigma^2}$ [60], i.e. there exists $\epsilon > 0$ such that

$$\Pr\left(\sqrt{n\sigma^2} - \epsilon < \|\mathbf{z}\| < \sqrt{n\sigma^2} + \epsilon\right) \to 1. \tag{5.37}$$

In the SU scenario, by letting the Voronoi region $\mathcal{V}$ cover a sphere with radius $(\sqrt{n\sigma^2} + \epsilon)/\alpha_{MMSE}$, the improvement of retry decoding decreases as n increases. This trade-off was illustrated in Figure 4.9 using Z_n lattices. From (4.50), since the equivalent noise of CF relaying can also be seen as a Gaussian component, applying the discussion of trade-off CF relaying is straightforward.

5.4.1.4 Summary of trade-off discussion

Overall, the SNR penalty can be reduced by increasing lattice dimension n or code rate R of the base lattice code $\mathcal{C} = \Lambda_c/\Lambda_s$. However, even though lower SNR penalty can be achieved, increasing n also reduces the improvement of retry decoding. It is important to select an appropriate dimension when implementing the CRC-embedded lattice codes to practical applications.

Numerically, for the SU scenario, gains of WER, by applying retry decoding with the CRC-embedded lattices, can be observed for low dimensional lattice codes and becomes marginal for medium dimensional lattice codes. For example, as shown in [70, Fig. 7], even though an SNR penalty is as small as 0.078dB using an $n = 128$ polar lattice code with a CRC-4 code

embedded, the gain is still marginal using $\mathcal{A}_1$ and $\mathcal{A}_2$ for retry decoding. For CF relaying, due to the extra freedom on changing integer coefficient $\mathbf{a}$, the improvement of retry decoding can still be observed for medium dimensional lattice codes.

To implement CRC-embedded lattice codes, low dimension lattice codes, such as E_8 and BW_{16} lattice codes, are considered in the SU scenario; medium dimensional lattice codes, such as $n = 64, 128, 256$ polar lattice codes, are considered in CF relaying. Since the SNR penalty is significant for low dimensional lattice codes, high code rates are considered for the SU scenario, which can be seen as high-order modulations in communications.

5.4.2 Comparing to receiver with retransmission

A discussion for comparing between the retry decoding scheme and a conventional scheme with retransmission is given for: 1) the message is successfully recovered after a decoding attempt, 2) the message is erroneous after a decoding attempt, 3) the average number of decoding attempts over all messages.

Recall that the proposed retry decoding applies to the decoder of the base lattice, rather than the CRC-embedded lattice, followed by error detection consisting of a mod 2 operation to obtain the LSB vector and a CRC check. In the first case, further retry decoding and retransmission are not needed. Error detection is additionally needed for the proposed retry decoding after the decoding attempt. However, both mod 2 operations and CRC check in error detection are low complexity, from which only a modest increase of decoding complexity is introduced. In the second case, another decoding attempt is always needed to recover the message. The conventional scheme requests a retransmission before making another attempt; while the proposed retry decoding can make another attempt immediately by using other decoding coefficient(s) in the candidate list, from which lower latency can be expected. However, it shall be noticed that the conventional scheme may achieve higher diversity gain due to the channel reuse. In the third case, at the range of WER for practical code design (e.g. $P_e = 10^{-5}$), the average number of decoding attempts becomes more important, which converges to 1 for small WER. The average complexity of retry decoding approaches that of one-shot decoding, while a lower WER can be achieved.

5.5 Implementation of CRC-embedded lattice codes

Numerical results on implementation of CRC-embedded lattice codes are given in this section for SU scenario and CF relaying. We first give the retry decoding scheme using CRC-embedded lattice codes and evaluation of P_{re}. The error performance are evaluated using E_8 and BW_{16} lattice codes for SU scenario and construction D polar lattice codes for CF relaying to:

1. verify the validity of $P_{e,total}$ estimate given in Proposition 5.3;
2. give the optimized CRC length and the maximized gain of retry decoding and CRC-embedded lattice codes based on the $P_{e,total}$ estimates.

5.5.1 Implementation in single user transmission

A k-level decoding procedure follows Section 4.2.2 using a list $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_k$. Here, CRC codes are applied for error detection instead of the genie used in the analysis, thus the SNR penalty is non-negligible. After each decoding attempt, the estimate of the uncoded message $\hat{\mathbf{b}}$ is recovered by the indexing function in (3.8) from which the LSB vector is extracted for CRC check. In simulations, a CRC check is performed after each decoding attempt, rather than after the whole level, in order to terminate the decoding process as soon as the CRC check passes to output the result. Another strategy, that might be considered, is that the decoder first finishes the whole decoding level to produce multiple estimates using a set $\mathcal{A}$ and then performs CRC check for all estimates. The decoding result is selected from the estimates having valid CRC with the maximum likelihood. Numerically, the improvement from this strategy is negligible while the decoding complexity is increased. Therefore, it is not further considered.

5.5.1.1 Estimating $P_{e,total}^{(k)}$

To estimate $P_{e,total}^{(k)}$, the value of $P_e^{(1)}$ is evaluated by one-shot decoding using α_{MMSE}. By the recursive structure of retry decoding, $P_e^{(i)}$ can be calculated using $P(\alpha|e_{i-1})$ defined in (4.12) as:

$$P_e^{(i)} = P_{e,CRC}^{(i-1)} - \sum_{j=1}^{2^{i-1}} P(\alpha_{i,j}|e_{i-1}) P_{e,CRC}^{(i-1)}. \tag{5.38}$$

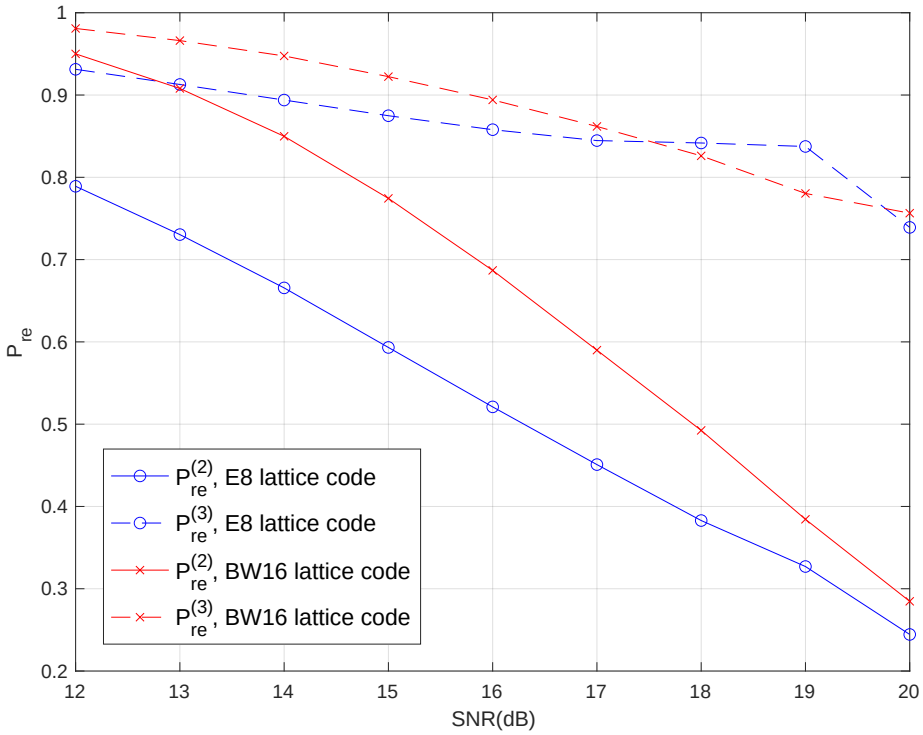


Figure 5.6: $P_{re}^{(2)}$ and $P_{re}^{(3)}$ for E_8 and BW_{16} lattice codes.

By (5.35), $P_{re}^{(i)}$ can then be obtained without requiring extra numerical evaluation as:

$$P_{re}^{(i)} = 1 - \sum_{j=1}^{2^{i-1}} P(\alpha_{i,j}|e_{i-1}). \tag{5.39}$$

Figure 5.6 illustrates numerical evaluation of $P_{re}^{(i)}$, with $i = 2, 3$, for E_8 and BW_{16} lattice codes at $R = 2$ and 2.25, respectively. It is observed that $P_{re}^{(2)} > 0.7$ in the high SNR regime, indicating the ratio of errors that can be corrected by retry decoding using $\mathcal{A}_2$. However, further retry using $\mathcal{A}_3$ only corrects errors from the second decoding level using $\mathcal{A}_2$, with a fraction of < 0.3, even though $\mathcal{A}_3$ has been optimized based on (4.12). This implies that the room left for improving WER using retry becomes small after decoding using $\mathcal{A}_2$. This agrees with the numerical evaluations shown in Figure 4.6 and Figure 4.7, where error performance of decoding using $\mathcal{A}_1 = \{\alpha_{MMSE}\}, \mathcal{A}_2 = \{\alpha_{2,1}, \alpha_{2,2}\}$ approaches that of the exhaustive search decoding.

5.5.1.2 Numerical results

The WER performance is given using E_8 and BW_{16} lattice codes with hypercube shaping. Since the shortest vector of the E_8 and BW_{16} lattices are known, for each CRC length l, a CRC polynomial optimization is first performed using (5.27) through a search over all possible CRC-l polynomials to minimize P_{ud}. After that, the CRC length is optimized to balance the error detection capability and SNR penalty using Proposition 5.3. With the SNR penalty in (5.15) included, Figure 5.7 verifies the accuracy of the estimate of $P_{e,total}^{(k)}$ using Proposition 5.3 for SU scenario. For target WER= 10^{-5} and 2-level decoding, the best achievable gain and the optimized CRC length l for E_8 and BW_{16} lattice codes are shown in Table 5.2 and Table 5.3, respectively, where R in tables are the code rates of based lattice code $\mathcal{C} = \Lambda_c/\Lambda_s$ before CRC being embedded. Regarding the first and second trade-offs discussed in Section 5.4.1, the optimized CRC lengths are short, since low dimensional lattice codes are applied. On the other hand, high rate codes are considered to reduce the SNR penalty to achieve larger gain. When $R = 2, 3, 4$ for E_8 lattice codes and $R = 2.25$ for BW_{16} lattice codes, the expected gain is less than 0 for which embedding parity bits and retry decoding are not recommended.

Table 5.2: The expected gain and optimized CRC for E_8 lattice code with hypercube shaping and 2 decoding levels.

R	Gain (dB)	Optimized l	CRC polynomial
2, 3, 4	< 0	-	-
5	0.0060	1	SPC code
6	0.0352	2	$x^2 + 1$
7	0.0621		
8	0.0845	3	$x^3 + x + 1$
9	0.1082		
10	0.1270		
11	0.1424		
-	0.3270	Upper bound on the gain.	

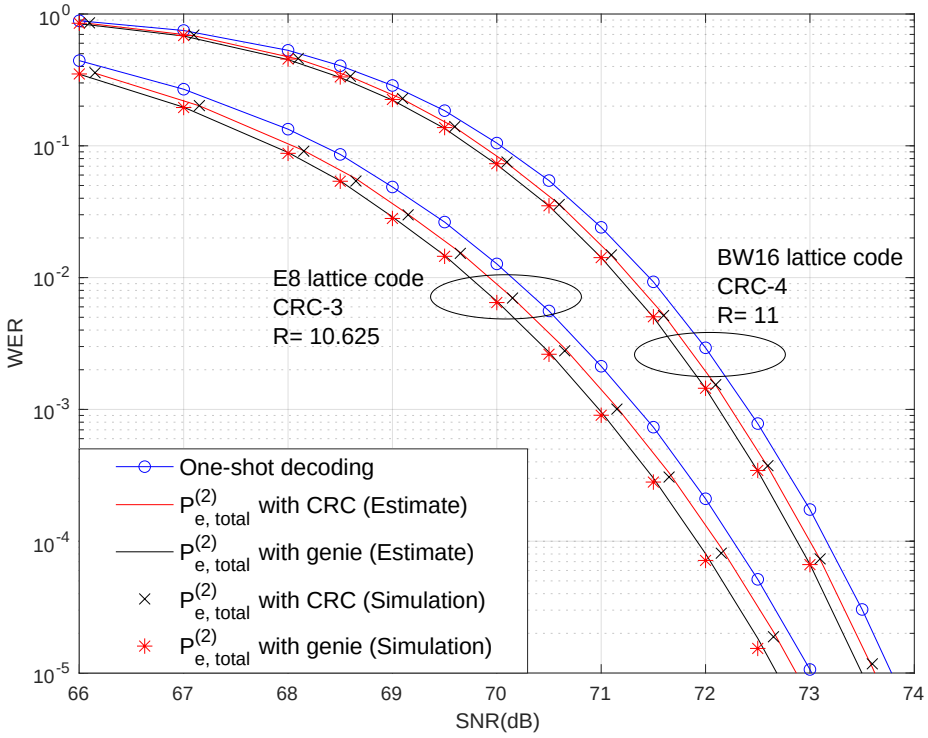


Figure 5.7: Estimated and actual $P_{e,total}^{(2)}$ for single user transmission using E_8 and BW_{16} lattice codes. The CRC polynomials are x^3+x+1 and x^4+x^3+1, respectively. The total decoding level is 2 using $\mathcal{A}_1 = \{\alpha_{MMSE}\}$, $\mathcal{A}_2 = \{\alpha_{2,1}, \alpha_{2,2}\}$.

5.5.2 Implementation in CF relaying

For implementation in CF relaying using ICF, a 2-user multiple access relay is considered. The retry decoding strategy follows scheme 1 in Section 4.3.3.1 but a CRC code is applied for error detection instead of a genie. Let $\mathcal{C} = \Lambda_c/\Lambda_s$ be the base lattice code before CRC embedding and the users' messages $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{C}$. The received message is $\mathbf{y} = h_1\mathbf{x}_1 + h_2\mathbf{x}_2 + \mathbf{z}$. The Rayleigh fading channel is assumed. The value of channel coefficients are generated randomly and time variant, where the total channel gain is normalized to $\|\mathbf{h}\| = 1$ to keep a constant received SNR. Using ICF, the estimate of the linear combination is $\hat{\mathbf{x}} \in \Lambda_c$ (not necessarily in $\mathcal{C}$). The corresponding uncoded message is recovered by $\hat{\mathbf{b}} = \mathbf{G}_c^{-1}\hat{\mathbf{x}}$, from which the LSB vector is extracted by $\bmod 2$ operation for CRC check. The EER of the linear combinations is measured at the relay, when $\hat{\mathbf{x}} \neq \sum_{i=1}^{L} a_i\mathbf{x}_i$.

Construction D polar lattice codes with $n = 64, 128, 256$ and hypercube

Table 5.3: The expected gain and optimized CRC for BW_{16} lattice code with hypercube shaping and 2 decoding levels.

R	Gain (dB)	Optimized l	CRC polynomial
2.25	< 0	-	-
3.25	0.0197	1	SPC code
4.25	0.0484	2	$x^2 + 1$
5.25	0.0741	3	$x^3 + x^2 + 1$
6.25	0.0997		
7.25	0.1182		
8.25	0.1322		
9.25	0.1431		
10.25	0.1528	4	$x^4 + x^3 + 1$
11.25	0.1624		
-	0.2880	Upper bound on the gain.	

shaping are used for channel coding. The coding lattice design follows [22], where for $n = 64$, component codes are $(64, 1), (64, 40)$ polar codes; for $n = 128$, component codes are $(128, 7), (128, 88)$ polar codes and for $n = 256$, component codes are $(256, 24), (256, 192)$ polar codes. The code rate for the base lattice codes are $R_{64} \approx 1.6406$, $R_{128} \approx 1.7422$ and $R_{256} \approx 1.8438$. The standard successive cancellation (SC) decoder is applied to decode polar codes.

5.5.2.1 Estimating $P_{e,total}^{(k)}$

To estimate $P_{e,total}^{(k)}$ in (5.31) for CRC length optimization, P_{ud} is estimated by (5.26) as $P_{ud} \approx 2^{-l}$, and both $P_e^{(1)}$ and $P_{re}^{(i)}$, for $i = 2, 3, \cdots, k$, are numerically evaluated using Monte-Carlo method. Figure 5.8 illustrates numerical evaluation of $P_{re}^{(2)}$ for the construction D polar lattice codes we considered for here.

5.5.2.2 Numerical results

Figure 5.9 verifies the accuracy of the estimate of $P_{e,total}^{(k)}$ using Proposition 5.3 for CF relaying, where CRC-4 and genie-aided error detection are considered with two decoding attempts for retry decoding using $n = 128, 256$.

Then, the expected gain for $n = 64, 128, 256$ polar lattice codes and retry decoding with two attempts are shown in Figure 5.10 for CRC length $l = 1, 2, \cdots, 16$. The value of P_{ud} is obtained using method 2 given in

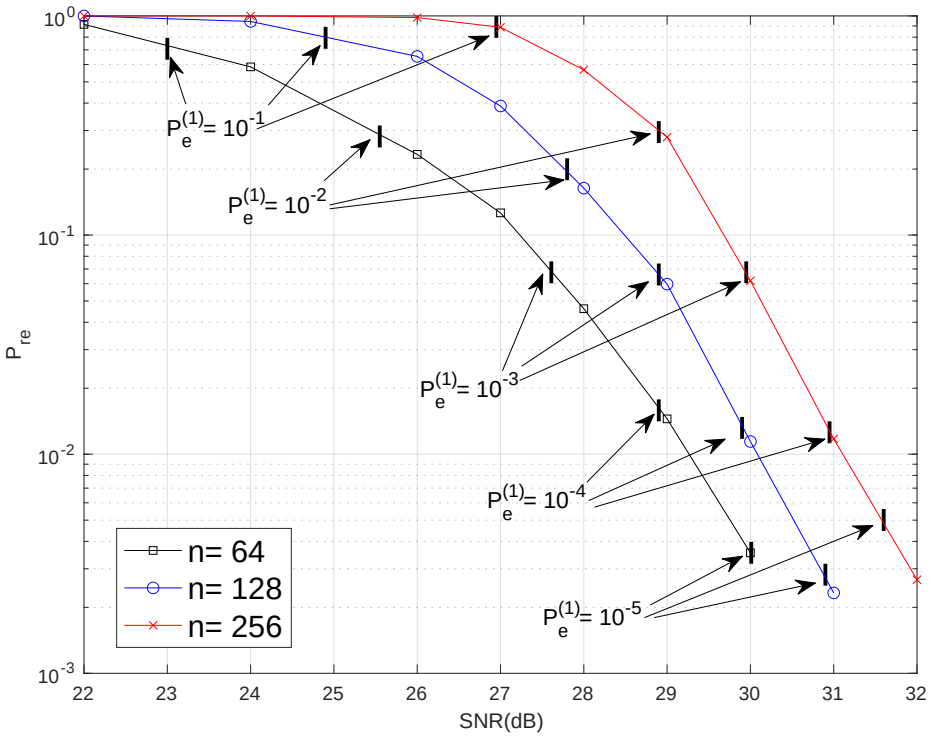


Figure 5.8: $P_{re}^{(2)}$ for $n = 64, 128, 256$ polar lattice codes.

Section 5.3.1 as $P_{ud} \approx 1/2^l$, where CRC polynomials are selected randomly without further optimization due to the high dimension of lattices. The upper bound on gains are obtained using genie-aided error detection, which has $P_{ud} = 0$ and no SNR penalty. The gaps between the actual gains to the upper bound are due to the imperfect error detection of the finite length CRC codes and the SNR penalty.

The maximum gains by CRC-embedded lattice codes are approximately 1.29dB, 1.31dB and 1.08dB, using CRC length of 7–8, 8–9 and 9–11 for $n = 64, 128, 256$, respectively. For longer CRC lengths, the gain decreases because the increasing SNR penalty overcomes the improvement on error rate. From the gain using genie-aided error detection, the improvement of retry decoding reduces as dimension increase, as we have discussed in Section 5.4.1. However, larger dimensional lattice codes also have smaller SNR penalty. Because of the smaller SNR penalty, $n = 128$ dimensional lattice codes achieves a larger gain than that of $n = 64$, when CRC is applied for error detection.

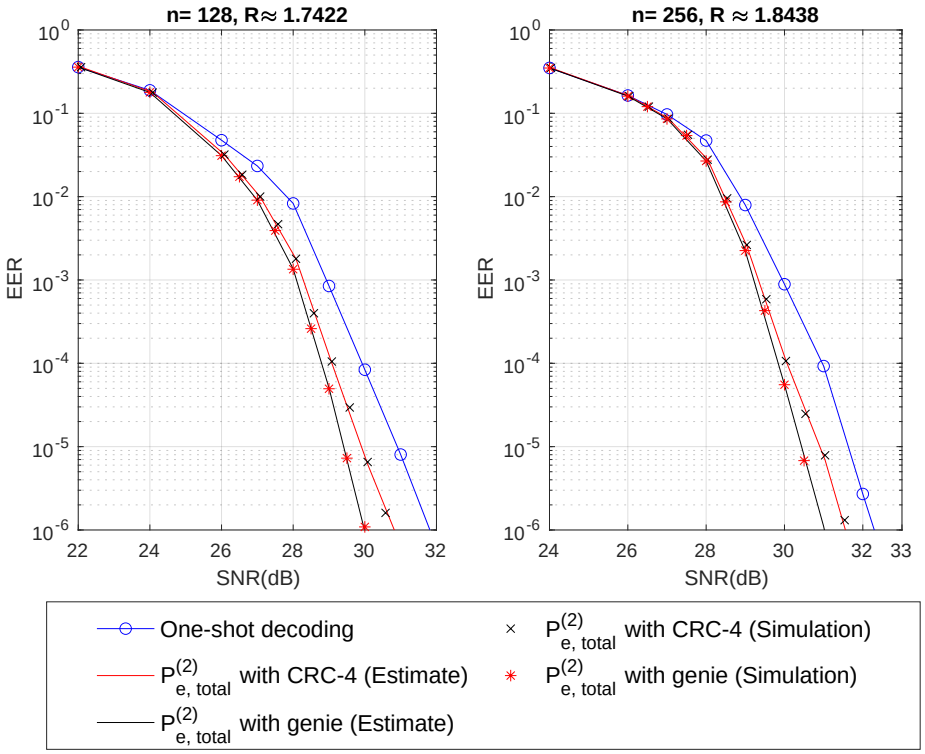


Figure 5.9: Estimate and simulation results of $P^{(2)}_{e,total}$ for 2-user CF relay using ICF with CRC-4 and genie-aided error detection. $n = 128, 256$ polar code lattice and hypercube shaping is used.

5.6 Summary of this chapter

In this chapter, a lattice construction with physical layer error detection ability was introduced. The error detection is deployed by embedding CRC codes into the LSB vector of the lattice uncoded messages, thus this lattice construction is called CRC-embedded lattices.

We first gave the definition of the construction, and showed this constriction forms a lattice in general. The encoding/decoding scheme of the CRC-embedded lattices was then given. For power constrained communications, the CRC-embedded lattice codes were introduced, which applies the CRC-embedded lattice as the coding lattice. In CF relaying using ICF, a stand-alone relay cannot detect errors from a linear combination since multiple unknowns (users' message) are included. A condition on shaping lattice design was given for the CRC-embedded lattice codes, which allows a stand-alone relay to detect error from a linear combination without requiring

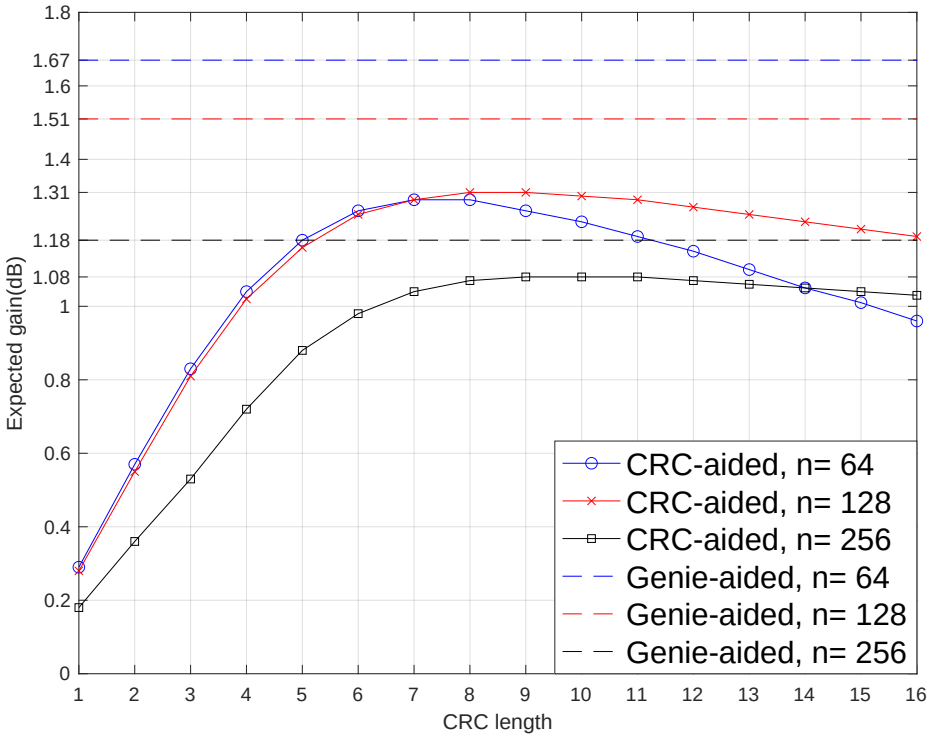


Figure 5.10: Expected gain for 2-user CF relay using $n = 64, 128, 256$ polar code lattice with CRC length from 1 to 16. The achievable gains are 1.29dB for $n = 64$ with CRC length being $7, 8$; 1.31dB for $n = 128$ with CRC length being $8, 9$; 1.08dB for $n = 256$ with CRC length being $9, 10, 11$.

individual users' messages.

CRC optimization was considered for practical code design with respect to CRC polynomial/position of parity bits and CRC length. The probability of undetected error P_{ud} was defined to evaluate the error detect capability for CRC-embedded lattices. Two methods on estimating P_{ud} were given, which consider:

1. the set of shortest vector of lattice;
2. only the CRC length regardless of the structure of lattice.

From the numerical, both methods give good estimate of P_{ud} verified by BW_{16} lattice code with various CRC lengths. In particular, Method 1 gives an more accurate estimate than Method 2 and allows us to optimize CRC polynomial/position of parity bits. However, it is only feasible for low dimensional lattices whose shortest vectors are known. On the other hand, Method 2 gives a convenient estimate of P_{ud}, which is applicable to large

dimensional lattices, only introducing a modest loss on estimation accuracy.

The CRC optimization was given for CRC polynomial and CRC length, respectively. It was shown that, optimizing CRC length affects P_{ud} more significantly than optimizing CRC polynomial. The CRC length optimization was given to maximize the gain obtained by retry decoding using CRC-embedded lattice codes. This is given as a semi-analytical method obtained by estimating the WER after retry decoding as a function of P_{ud} or, equivalently, as a function of CRC length l. In numerical evaluations of this chapter, the CRC codes for embedding have the optimized length for the underlying coding lattices, while the CRC polynomial optimization is only applied for low dimensional lattices.

Discussions were then given for trade-offs related to implementation of CRC-embedded lattices/lattice codes, and comparison to conventional receiver with retransmission. The trade-offs considered lattice code dimension, code rate, SNR penalty due to CRC bits and the benefit of retry decoding, which gives an insight on designing CRC-embedded lattice codes with appropriate lattice dimension and code rate. In the comparison with conventional receiver, given an error happened, a receiver would have to give another decoding attempt, from which a lower transmission latency can be expected using CRC-embedded lattice code with retry decoding by avoiding retransmission. However, it is also noticed that the conventional receiver achieves higher diversity gain due to the channel reuse. A comprehensive study can be considered to compare the conventional receiver with retransmission and CRC-embedded lattice code with retry decoding from a system-level perspective, involving the transmission/decoding latency analysis and/or overall system performance under same WER/latency requirement.

In numerical evaluations, we first verified the CRC length optimization to shown the accuracy on estimating $P_{e,total}^{(k)}$, the WER/EER after retry decoding. Then, by considering the estimated WER/EER as a function of CRC length l, the optimized CRC length and the maximized gain are given. According to the discussion given in Section 5.4.1, we applied low dimensional E_8 and BW_{16} lattice codes for SU scenario and medium dimensional polar lattice codes, with $n = 64, 128, 256$, for CF relaying. From numerical evaluations, we have the following observations.

1. For the SU scenario, since the effect of SNR penalty is significant when n is small, high rate lattice codes are required to achieve a gain using CRC-embedded lattice code with retry decoding, which can be considered a competing scheme with high order QAM to reduce constellation power.
2. For CF relaying, a more significant gain than that of SU scenario is

observed. At a 2-user relay, gains up to $1,29$dB, 1.31dB and 1.08dB are achieved at EER $\approx 10^{-5}$ by only adding one more decoding attempt using $n = 64, 128, 256$ construction D polar lattice codes, respectively, with optimized CRC length.

3. The trade-off between dimension and improvement of retry is demonstrated by CF relaying. Even though increasing dimension reduces the gain of retry, with the lower SNR penalty, $n = 128$ lattice codes gives a larger gain than that of a lower dimensional lattice codes with $n = 64$. This implies that, for a practical application, a range of dimensions can be found to balance the improvement of retry and SNR penalty to maximize the gain obtained by retry decoding and CRC-embedded lattice codes.

Additionally, it is noticed that polar codes applied for CF relaying have CRC-aided successive cancellation list (SCL) decoding to improve error performance compared with SC decoding [71]. The CRC for SCL decoding performs error detection at the component binary codeword level; while the proposed CRC-embedded lattice code performs error detection at the lattice codeword level. Even though SCL achieves better error performance than SC on decoding polar codes, the standard SC decoder is considered in this chapter to avoid ambiguity between two types of CRC codes. However, it is valid to implement both CRC-embedded lattice codes and CRC-aided SCL decoding to achieve lower error rate.

Chapter 6

Finite dimensional lattice design using binary code

6.1 Introduction of this chapter

This chapter considers finite dimensional lattice design. In particular, construction A using binary code is considered to design medium dimensional lattice, such as at $n = 64 \sim 256$. Binary codes with such block length have low decoding latency, which is one of key requirements for future wireless communications; while having good error performance. On the other hand, construction A has lower complexity than construction D by using only one component code. There were studies on construction A lattices using non-binary codes and for high dimensional lattices, such as the low-density construction A (LDA) lattices [15], and construction D lattices at medium dimension [20] [22]. However, there has been relatively little study using construction A at medium dimensions. Binary codes have well-studied distance properties, and have codeword Hamming weight equal to their Euclidean norm, which is not true for non-binary codes. Therefore, binary codes are considered as the component code for medium dimensional construction A lattices, which allow us to give analytic lattice designs for low latency communications. In this chapter, the square d^2 denotes the Euclidean norm and the non-square d denotes the Hamming distance, to distinguish the distance in the real number field and the binary field.

This chapter is divided into 3 parts. In the first part, we give a design method. Construction A lattices can be seen as a special case of multilevel codes, consisting of a coded layer and an uncoded layer. Various classic rules are suggested in [72] to design multilevel codes, such as the balanced distance rule and the equal error probability rule, which are considered for comparison in this chapter. A new design rule is proposed using the truncated union bound, which gives lattices to achieve the best error performance under maximum likelihood (ML) decoding. Binary codes with known minimum Hamming distance d_c and the codeword multiplicity τ_c, i.e, the number of

codeword at minimum Hamming weight, are assumed to apply the proposed design approach. A truncated theta series, introduced in Definition 2.12, is explicitly given, from which the truncated union bound is computed to estimate the WER of lattices under ML decoding.

In the second part, the design examples are given using $n = 128$ dimensional lattices. Extended BCH (EBCH) codes and polar codes are selected as the component codes, since EBCH codes have a large minimum Hamming distance and polar codes [73] are accepted as a part of the channel coding scheme in the 5G standard, respectively. In addition, the dimension $n = 128$ is almost the largest possible dimension to deploy a (near) ML decoder with reasonable complexity to evaluate design results. The ordered-statistics decoding (OSD) algorithm [74] is considered to decode binary codes. The OSD algorithm achieves a near-ML performance by letting the order $o = \lceil d_c/4 - 1 \rceil$, where d_c is the minimum Hamming distance and $\lceil \cdot \rceil$ is the ceiling function. Numerical evaluations are given using binary codes with different minimum Hamming distance and/or codeword multiplicity to find the best construction A lattice, which has the lowest VNR to achieve a target WER.

At last, a comparison of the WER between our design with classic rules and construction D are given for unconstrained lattices and lattice codes. At $n = 128$, the balanced distance rule and the equal error probability rule are considered as the classic design rules. And construction D lattices are from [20] and [22].

This chapter gives a design approach for finite dimensional lattices using binary codes, which can be considered as the base lattice of CRC-embedded lattices to implement the retry decoding. The structure of construction A can be implemented by BPSK and PAM, see [25, 39, 53]. Meanwhile, the proposed design connects the binary codes in conventional systems and the lattice codes. This implies potential applications of lattice coding in practical systems with optimized error performance under the ML decoding.

6.2 System model

Since binary codes are used to form the lattice, the mod-2 construction A, given in Definition 2.18, is applied for the lattice design. The AWGN channel is assumed as the channel model. The system model is given in Figure 6.1. The binary code is denoted as $\mathcal{C}_b$ with code parameter (n, k, d_c). The module "Enc" and "Dec" are the binary encoder and decoder of $\mathcal{C}_b$. Denote a mapping function $\psi(\cdot) : \mathbb{F}_2^n \to \mathbb{R}^n$, such that $0 \to 0, 1 \to 1$.

At the transmitter, a binary sequence $\mathbf{u} \in \mathbb{F}_2^k$ is first encoded to $\mathbf{c} =$

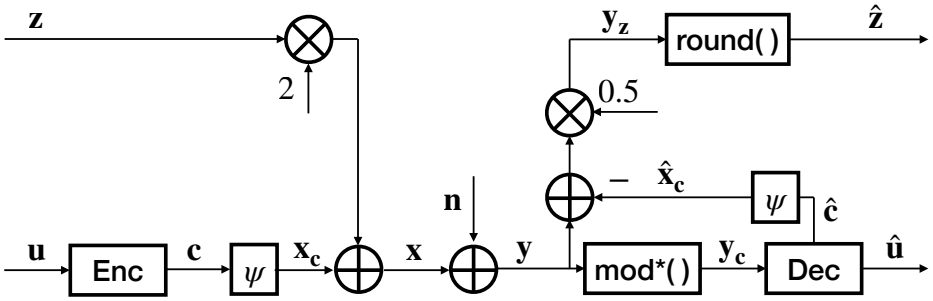


Figure 6.1: System model for transmission of mod-2 construction A lattices through AWGN channel.

$\text{Enc}(\mathbf{u}) \in \mathcal{C}_b$, then mapped into real number space using $\mathbf{x_c} = \psi(\mathbf{c})$. A lattice point is generated as $\mathbf{x} = \mathbf{x_c} + 2\mathbf{z}$, where $\mathbf{z} \in \mathbb{Z}^n$. The AWGN channel output is

$$\mathbf{y} = \mathbf{x} + \mathbf{n}, \tag{6.1}$$

where the Gaussian noise $\mathbf{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$.

At the receiver, successive cancellation decoding is applied, which is optimal to decode construction A lattices [3]. Before decoding $\mathcal{C}_b$, a modulo operation, introduced in [3] and simplified in [20], is performed to preserve distances to $(0, 1)$ as:

$$\mathbf{y_c} = \text{mod}^*(\mathbf{y}) = |(\mathbf{y} + 1) \bmod 2 - 1|, \tag{6.2}$$

where operations are component-wise. The binary decoder finds an estimate of uncoded message $\hat{\mathbf{u}}$ and the corresponding coded message $\hat{\mathbf{c}}$, which is then mapped to real number space to obtain $\hat{\mathbf{x}}_\mathbf{c} = \psi(\hat{\mathbf{c}})$. The estimate $\hat{\mathbf{x}}_\mathbf{c}$ is subtracted from the received message to obtain

$$\mathbf{y_z} = (\mathbf{y} - \hat{\mathbf{x}}_\mathbf{c})/2, \tag{6.3}$$

from which an estimate of integer $\hat{\mathbf{z}}$ is obtained by the rounding function.

6.3 Design method

This design is based on the truncated union bound. To compute it, a truncated theta series θ', introduced in Definition 2.12, is given consisting of first m non-zero term with $d_m^2 = d_c$. The lattice is designed to have the lowest VNR to achieve a given target WER.

It is noticed that, if $\mathbf{x} \in \Lambda_a$ and $\|\mathbf{x}\|^2 \leq d_c$, then $\mathbf{x}$ can be expressed as either an even integer vector $2\mathbf{z} \in 2\mathbb{Z}^n$ with $\|2\mathbf{z}\|^2 \leq d_c$ or a binary codeword $\mathbf{x_c}$ at minimum Hamming distance d_c. This can be shown that, for any $\mathbf{x}' = \mathbf{x}'_\mathbf{c} + 2\mathbf{z}$ with non-zero $\mathbf{x}'_\mathbf{c}$ and $\mathbf{z}$, its Euclidean norm $\|\mathbf{x}'\|^2 > d_c$. Therefore, the truncated theta series θ' is obtained by finding a truncated theta series of $2Z_n$ lattice with $d_i^2 \leq d_c$ and adding one term contributed by the minimum weight codeword of $\mathcal{C}_b$.

The 1-dimensional $2Z$ lattice consists of even integers and has the theta series:

$$\theta_{2Z} = 1q^0 + 2\sum_{i=1}^{\infty} q^{4i^2}. \tag{6.4}$$

The theta series of $2Z_n$ lattice can be obtained exactly by taking n-th power of θ_{2Z} as $\theta_{2Z_n} = (\theta_{2Z})^n$. Let θ_{2Z_n,d_m^2} be the truncated theta series of $2Z_n$ lattice with terms having $d_i^2 \leq d_m^2$. This can be computed by polynomial multiplication using a truncation of θ_{2Z} and dropping the terms with $d_i^2 > d_m^2$.

Example 6.1: Let $n = 4$ and $d_m^2 = 16$. Using $\theta_{2Z,16} = 1q^0 + 2q^4 + 2q^{16}$, it can be computed that

$$\begin{aligned}(\theta_{2Z,16})^4 &= (1q^0 + 2q^4 + 2q^{16})^4 &(6.5)\\ &= 1q^0 + 8q^4 + 24q^8 + 32q^{12} + 24q^{16} + 48q^{20} + \cdots . &(6.6)\end{aligned}$$

The truncated theta series is obtained by dropping the terms with $d_i > 16$ as:

$$\theta_{2Z_4,16} = 1q^0 + 8q^4 + 24q^8 + 32q^{12} + 24q^{16}. \tag{6.7}$$

Note that the truncation of θ_{2Z} should not be too short, otherwise the result of θ_{2Z_n,d_m^2} may not be accurate. It can be shown by assuming a truncation $\theta_{2Z,4} = 1q^0 + 2q^4$. In this case, we have

$$(\theta_{2Z,4})^4 = 1q^0 + 8q^4 + 24q^8 + 32q^{12} + 16q^{16}, \tag{6.8}$$

where the coefficient for q^{16} is 16, smaller than its actual value 24.

Assume the binary code $\mathcal{C}_b$ has known minimum Hamming distance d_c and codeword multiplicity τ_c. A truncated theta series of the construction A lattice $\Lambda_a = \mathcal{C}_b + 2\mathbb{Z}^n$, until $d_m^2 = d_c$, can be explicitly given as in the following proposition.

Proposition 6.1: The truncated theta series θ' of Λ_a with $d_i^2 \leq d_c$ is as follows. For $d_c \leq 4$,

$$\theta' = \begin{cases} 1q^0 + \tau_c 2^{d_c} \cdot q^{d_c} & d_c < 4 \\ 1q^0 + (\tau_c 2^{d_c} + 2n) \cdot q^{d_c} & d_c = 4. \end{cases} \tag{6.9}$$

For $d_c > 4$, let L be the integer part of $d_c/4$, then

$$\theta' = \begin{cases} \theta_{2Z_n,4L} + \tau_c 2^{d_c} \cdot q^{d_c}, & d_c \bmod 4 \neq 0 \\ \theta_{2Z_n,4(L-1)} + \left(\tau_c 2^{d_c} + \tau_{2Z_n,d_c}\right) \cdot q^{d_c}, & d_c \bmod 4 = 0. \end{cases} \tag{6.10}$$

Proof. Since lattice is defined over the real number space, a binary codeword at minimum weight corresponds to 2^{d_c} distinct lattice points by adding $+$ or $-$ sign at each position of '1'. Therefore, all minimum weight codewords contribute $\tau_c 2^{d_c}$ lattice points for θ'. And, as we justified above, θ' is obtained by adding one term contributed by the minimum weight codewords of $\mathcal{C}_b$ to $\theta_{2Z_n,4L}$. This concludes the proof. □

Use θ', the truncated union bound is computed as

$$P_e \approx \sum_{i=1}^{m} \tau_{d_i^2} \cdot Q\left(\sqrt{\frac{d_i^2}{4\sigma^2}}\right), \tag{6.11}$$

where m is the number of terms of θ'. By (2.18) and (2.38), the noise variance is

$$\sigma^2 = \frac{4^{1-R_b}}{2\pi e \cdot \mathrm{VNR}}, \tag{6.12}$$

where R_b is the code rate of $\mathcal{C}_b$. Substituting (6.12) into (6.11), we have

$$P_e \approx \sum_{i=1}^{m} \tau_{d_i^2} \cdot Q\left(\sqrt{\frac{d_i^2 \cdot 2\pi e \cdot \mathrm{VNR}}{4 \cdot 4^{1-R_b}}}\right). \tag{6.13}$$

It is noticed that P_e depends on R_b and θ' which are decided by the binary code $\mathcal{C}_b$. Therefore, P_e can be seen as a function of binary code $\mathcal{C}_b$ and VNR, denoted as

$$P_e = f(\mathcal{C}_b, \mathrm{VNR}). \tag{6.14}$$

Denote the inverse function as

$$\mathrm{VNR} = f^{-1}(\mathcal{C}_b,\ P_e). \tag{6.15}$$

Since (6.13) consists of multiple Q functions, the inverse function is obtained numerically.

Consider $\mathcal{C}_{all}$ as the search space of binary codes. The best component code $\mathcal{C}_b$ is found by minimizing the required VNR for a target P_e:

$$\mathcal{C}_b = \underset{\mathcal{C}_b' \in \mathcal{C}_{all}}{\arg\min}\, f^{-1}(\mathcal{C}_b',\ P_e), \tag{6.16}$$

Table 6.1: The code parameter of the best EBCH codes for construction A at given target P_e, with its required VNR to achieve P_e.

P_e	Required VNR(dB)	Code parameter	τ_c
10^{-4}	2.86	$(128, 106, 8)$	774192
10^{-5}	3.38		
10^{-6}	3.95		
10^{-7}	4.45	$(128, 113, 6)$	341376
10^{-8}	4.81		

or equivalently, minimizing the estimate P_e for a target VNR:

$$\mathcal{C}_b = \underset{\mathcal{C}_b' \in \mathcal{C}_{all}}{\arg\min}\, f(\mathcal{C}_b', \text{VNR}). \tag{6.17}$$

In the next section, equation (6.16) is considered as the target function to give design examples.

6.4 Design examples

The lattice design examples are given using EBCH codes and polar codes, with block length $n = 128$. The target WER to design the lattice is set to $P_e \leq 10^{-4}$, since the truncated union bound becomes accurate for small P_e. The OSD algorithm is applied to decode binary codes, which achieves a near-ML performance to evaluate the truncated union bound.

6.4.1 Design using EBCH codes

EBCH codes have fixed configurations with given n and d_c. The codeword multiplicity τ_c are found for most of EBCH codes with n up to 256 [75, 76] and are summarized in [77]. EBCH codes with $n = 128$ and $d_c = 4, 6, 8, 10$ are considered to give lattice design examples. The order of OSD algorithm is set to be 2, which is possible to give a near-ML performance for EBCH codes with $d_c \leq 10$.

Figure 6.2 verifies the truncated union bound in (6.13) using Monte-Carlo simulation. According to (6.16), Table 6.1 gives the best EBCH codes for different target P_e from which the lattices achieve the given P_e with the lowest VNR.

The numerical results show that using $(128, 120, 4)$ EBCH code as the component code, as suggested by the balanced distance rule, gives higher

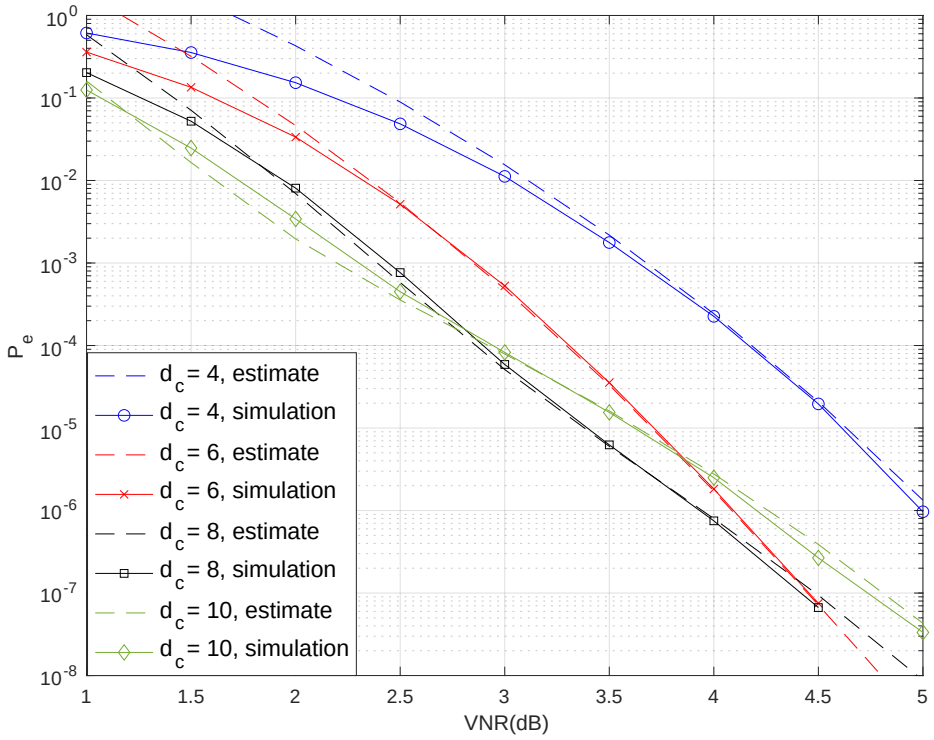


Figure 6.2: The truncated union bound estimate and numerical evaluation of P_e for EBCH code lattice with $d_c = 4, 6, 8, 10$. Order-2 OSD algorithm is used to decode EBCH codes.

P_e than other lattices using component codes with $d_c > 4$. This can be justified from the value of τ_4, which is the number of lattice points at $d^2 = 4$. Since $(128, 120, 4)$ EBCH code has $\tau_c = 85344$, then $\tau_4 = 1365760$ by (6.9); meanwhile for EBCH codes with $d_c > 4$, τ_4 is only 256, from which smaller P_e is achieved, as shown in Figure 6.2. As VNR grows, the contribution from high order terms of the truncated theta series θ' reduces for computing the truncated union bound. At a relatively high VNR, estimate P_e using lattice points with d^2_{min} becomes sufficient and, for lattices with same τ_4, the lattice with the smallest volume, equivalently the largest rate R_b of the component code, gives the lowest P_e. This can be observed from Figure 6.2 as well, that is the lattice using $(128, 113, 6)$ EBCH code achieves lower P_e than that using $(128, 106, 8)$ EBCH code when $P_e \leq 10^{-7}$. However, for very high VNR, since the Q function in (6.13) decreases exponentially as R_b increases, the component code with highest R_b, that is the $(128, 120, 4)$ EBCH code here, will eventually become the best. However, it occurs at VNR ≈ 11.8dB with $P_e \approx 10^{-47}$ computed from the truncated union bound, which is beyond

the scope of practical lattice design.

6.4.2 Design using polar codes

Polar codes have more flexibility on selecting the code parameter (n, k, d_c) than EBCH codes. The code rate R_b can be adjusted by a step as small as $1/n$ and, for given (n, k, d_c), polar codes with different τ_c can be constructed by selecting different information set. Additionally, polar codes also include the Reed-Muller codes and the extended Hamming code by properly selecting the information set.

To reduce the search space for finding good polar codes, a class of polar codes, described in [78,79], is considered. Such polar codes are defined based on the partial order of the binary representation of bit-channel indices, which is referred as *partial order property* in [78] and *decrease monomial codes* in [79,80]. The codeword multiplicity τ_c of polar codes satisfying the partial order property can be calculated analytical. Let $\mathcal{I}_p \subseteq \{0, n-1\}$ be the information set of an (n, k, d_c) polar code and a subset $\mathcal{I}'_p \subseteq \mathcal{I}_p$ indicate the rows in the polar transformation matrix with row weight d_c.

Proposition 6.2: [78] Consider an (n, k, d_c) polar code satisfying the partial order property in [78, Definition 2]. For all $i \in \mathcal{I}'_p$, let

$$\mathcal{K}_i = \{\, j \in [i+1, n-1] \mid \mathrm{wt}(\mathbf{g}_j) \geq \mathrm{wt}(\mathbf{g}_i \oplus \mathbf{g}_j) = \mathrm{wt}(\mathbf{g}_i) \,\}, \tag{6.18}$$

where $\mathrm{wt}(\mathbf{g}_i)$ is the weight of the i-th row of the polar transformation matrix. The codeword multiplicity is found by:

$$\tau_c = \sum_{i \in \mathcal{I}'_p} 2^{|\mathcal{K}_i|}. \tag{6.19}$$

Proposition 6.2 implies that τ_c only depends on the subset $\mathcal{I}'_p$. Based on this fact, information set $\mathcal{I}_p$ of a (n, k, d_c) polar code is select as follows:

- select all rows from the polar transformation matrix with row weight larger than d_c to avoid rate loss,
- select rows with weight d_c to form the information set to fit the desired code rate $R_b = k/n$ while satisfying the partial order property.

There may exist multiple polar codes having different τ_c for a given code parameter (n, k, d_c) according to the selection of $\mathcal{I}'_p$. This results in different truncated theta series θ' at the term with $d^2 = d_c$. For polar codes with small d_c, smaller τ_c may more significantly affect the P_e computed by (6.13) than that of larger d_c, due to the property of the Q function. This is evaluated

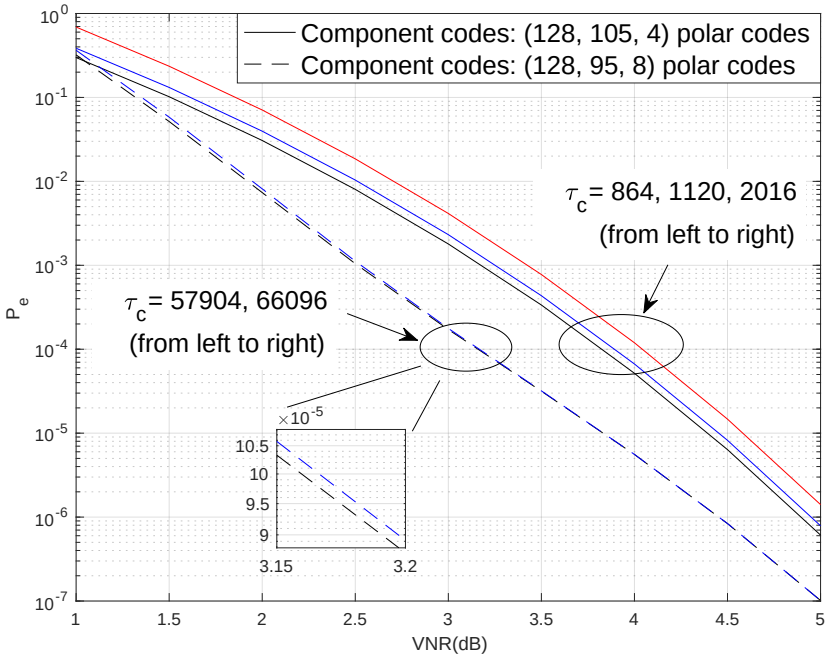


Figure 6.3: Truncated union bound of construction A lattices using polar codes with different τ_c for $d_c = 4, 8$.

for polar code lattices with component code having parameter $(128, 105, 4)$ and $(128, 95, 8)$, as illustrated in Figure 6.3. For lattices using $(128, 105, 4)$ polar codes, choosing the component code with smaller τ_c improves ≈ 0.2dB at $P_e = 10^{-4}$; however, for $(128, 95, 8)$, such improvement is negligible.

Figure 6.4 verifies the truncated union bound in (6.13) for polar code lattices with $d_c = 4, 8$ as a function of k at VNR $= 2.5, 3, 3.5$dB using the order-2 OSD, where only polar codes satisfying the partial order property are considered. For polar codes with same k, the one with the smallest τ_c is selected as the candidate code for evaluation.

At VNR $= 2$dB with $P_e \approx 10^{-3}$, a mismatch of P_e between the estimate and the simulation is observed, where the best component code has $d_c = 8$ from the simulation but not 4 from the estimate using (6.13). This is because, at low VNR regime, the lattice points with $d^2 > 4$ may still contribute to P_e, which is not involved in (6.13) for polar codes with $d_c = 4$. For error rate $P_e \leq 10^{-4}$, the contribution of lattice points with $d^2 > 4$ to P_e becomes small and the estimate becomes accurate.

Even though the $(128, 99, 8)$ polar code has very large $\tau_c = 188976$, it reduces τ_4 from 768, using the $(128, 100, 4)$ polar code, to 256 with only $1/128$ of rate loss, by which a better error performance is achieved at $P_e \leq 10^{-4}$.

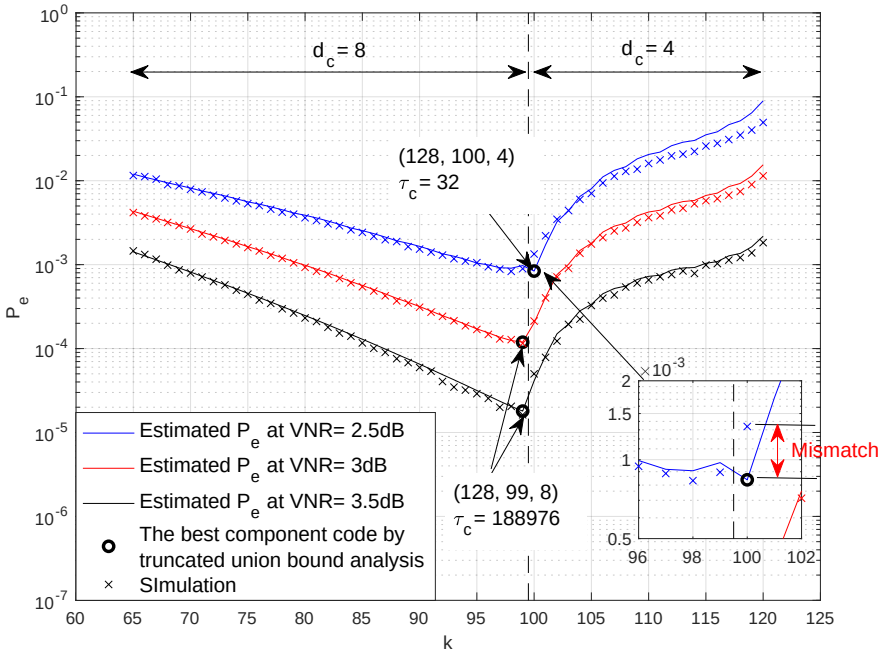


Figure 6.4: The truncated union bound estimate and numerical evaluation of P_e for polar code lattice with different code rates for $d_c = 4, 8$. Order-2 OSD algorithm is used to decode polar codes.

The best component polar code has code parameter $(128, 99, 8)$, which is also the $RM(7, 4)$ Reed-Muller code, for $P_e = 10^{-4}$ to 10^{-8} as summarized in Table 6.2.

6.4.2.1 Polar codes that do not satisfy partial order property

Using polar codes satisfying the partial order property, the $(128, 99, 8)$ polar code gives the best construction A polar code lattice. A study over polar codes not satisfying the partial order property is performed to investigate if there exists a better polar code.

Polar codes with k close to 99 are considered, which have rate close to the best polar code we found above. The value τ_c is found by numerical method. Since polar codes with k close to 99 have high code rate, the value of τ_c is obtained by: first find the weight enumerator function (WEF) for the dual code; then convert it to the WEF of the original code using the Macwilliams' theorem, see [81, Chapter 3], [82]; the value of τ_c is obtained from the coefficient of the WEF. This study is given for polar codes having $d_c = 8$ with $k = 97, 98, 99$ and $d_c = 4$ with $k = 100, 101, 102, 103$. Smaller τ_c, than that of polar codes satisfying the partial order property, are found at $k =$

Table 6.2: The code parameter of best polar codes for construction A for a given target P_e and the required VNR to achieve that P_e.

P_e	Required VNR(dB)	Code parameter	τ_c
10^{-4}	3.05	$(128, 99, 8)$	188976
10^{-5}	3.67		
10^{-6}	4.27		
10^{-7}	4.82		
10^{-8}	5.31		

$97, 101, 102, 103$. However, better component codes exceeding the $(128, 99, 8)$ polar code have not been found, as illustrated in Figure 6.5. Additionally, as discussed for Figure 6.3, smaller τ_c of polar codes with $d_c = 4$ gives a non-trivial gain of P_e at $k = 101, 102, 103$, while for $d_c = 8$, the gain is negligible at $k = 97$.

6.5 WER performance

This section compares the WER performance for construction A lattices designed by the proposed truncated union bound analysis with that by the classic balanced distance rule and the equal error probability rule, suggested in [54] and [72]. For the balanced distance rule, the component EBCH code and polar code are selected with $d_c = 4$, giving the same Euclidean norm as d_{min}^2 of the $2Z_n$ lattice. For the equal error probability rule, the binary codes are selected following [22], such that the WER of $\mathcal{C}_b$ with noise variance σ^2 is equal to that of $2Z_n$ with noise variance $\sigma^2/4$ through additive modulo Gaussian noise channel. Since EBCH codes only have fixed configurations, the equal error probability rule is applied only to polar code lattices. The order of OSD algorithm is set to 2 for decoding all binary codes, which aims to give a fair comparison by using the same decoder and also to achieve a near-ML performance.

Figure 6.6 shows WER of $n = 128$ lattices and lattice codes using EBCH codes and polar codes. Since the target is to compare the underlying coding lattice, the $8Z_n$ lattice is applied as the shaping lattice for all lattice codes, in order to avoid the influence of shaping gain caused by using the Voronoi shaping. For lattice codes, SNR_{norm} suggested in [59] and [83] is also considered as an alternative to VNR, which allows one to compare lattice

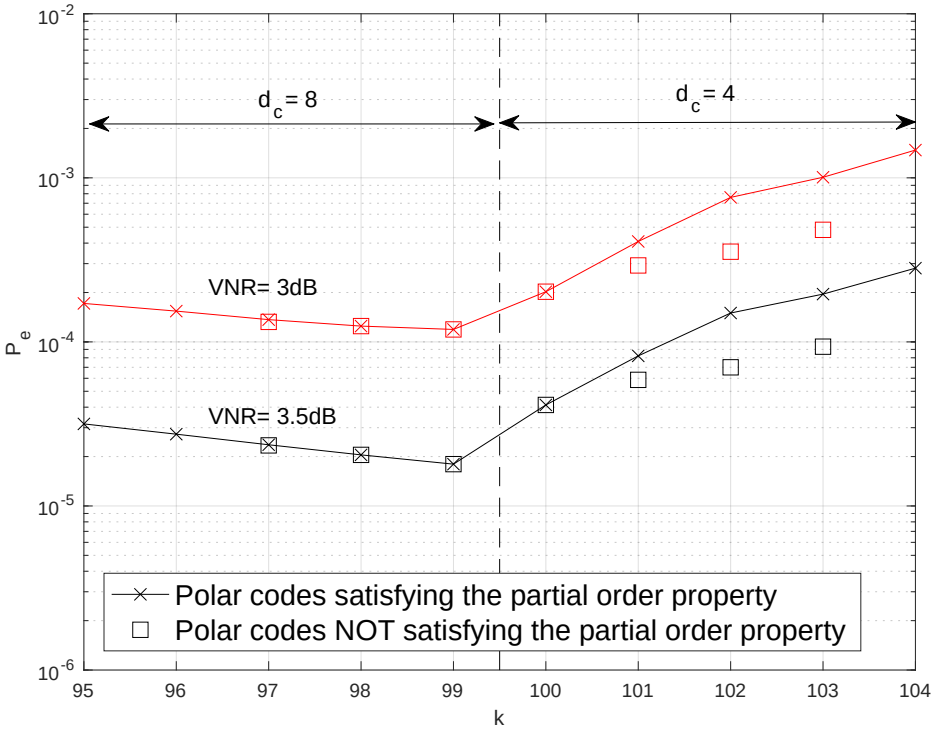


Figure 6.5: Truncated union bound for polar code not satisfying the partial order property for $k = 97$ to 103.

codes of different rates on the same scale, defined as

$$\text{SNR}_{norm} = \frac{P}{(2^{2R} - 1) \cdot \sigma^2}, \tag{6.20}$$

where P and R are the per-dimensional power and the code rate of the lattice code, respectively. One way to explain SNR_{norm} is given as follows. From the Gaussian channel capacity

$$R < C = \frac{1}{2} \log(1 + \frac{P}{\sigma^2}), \tag{6.21}$$

the definition in (6.20) implies that $P_e \to 0$ can be asymptotically achieved as $\text{SNR}_{norm} \to 1$, which is a constant independent of code rate R. However, for other definitions, such as $\text{EsN0} = P/\sigma^2$ or $\text{EbN0} = P/(R \cdot \sigma^2)$, the required SNR to achieve arbitrary small P_e depends on R. The relationship between

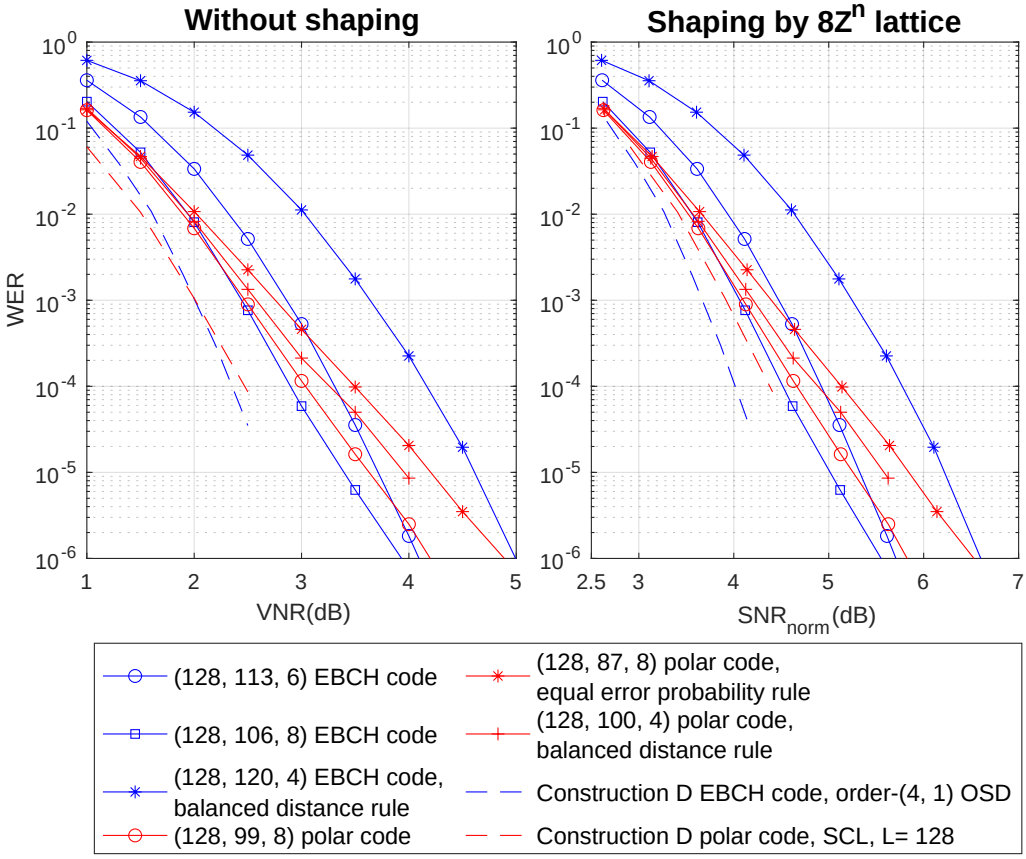


Figure 6.6: WER performance for construction A lattices with different design rules. Order-2 OSD is used to decode component codes of construction A lattices.

VNR and SNR_{norm} is given as follows by combining (2.18) and (6.20):

$$\begin{aligned}\mathrm{SNR}_{norm}(dB) =& \mathrm{VNR}(dB) + 10\log_{10}(2\pi e \cdot P) \\ &- \underbrace{10\log_{10}[(2^{2R}-1)\cdot V(\Lambda)^{2/n}]}_{\text{affected by code rate}}. \end{aligned} \tag{6.22}$$

Among construction A lattices, the proposed design approach achieves lower WER than that of the balanced distance rule or the equal error probability rule. At $\mathrm{WER} = 10^{-5}$, lattices using the $(128, 106, 8)$ EBCH code and the $(128, 99, 8)$ polar code are the best construction A lattices among design examples using EBCH codes and polar codes, respectively; and the $(128, 106, 8)$ EBCH code lattice is the best-known construction A lattice. Construction D lattices in previous studies are also plotted for comparison using: EBCH code [20], with component codes $\mathcal{C}_0 = (128, 78, 16)$ and

$\mathcal{C}_1 = (128, 120, 4)$, using order-$(4, 1)$ OSD; polar code [22], with $\mathcal{C}_0 = (128, 7)$ and $\mathcal{C}_1 = (128, 95)$, using successive cancellation list (SCL) decoding. Gaps remain from our best construction A $(128, 106, 8)$ EBCH code lattice to construction D lattices. However, our design has lower decoding complexity and design cost. For EBCH code lattices, our best lattice only needs $\sum_{i=1}^{2} \binom{106}{i} = 5671$ OSD reprocessing computations in maximum for one decoding, compared to $\sum_{i=1}^{4} \binom{78}{i} + \binom{120}{1} = 1505702$ of the corresponding construction D EBCH code lattice. On the other hand, the proposed design gives an analytic approach to optimize the component codes using the truncated union bound; while the construction D polar code lattices with SCL are optimized using the Monte-Carlo method, resulting in a higher design cost.

6.6 Summary of this chapter and discussions

This chapter considered finite dimensional lattice design using construction A and binary codes with known minimum Hamming distance d_c and codeword multiplicity τ_c. This design is targeted for coding lattices at low to medium dimensions for low latency decoding.

The proposed design is based on truncated union bound analysis, from which the lattice is expected to achieve the lowest WER under ML decoding. A truncated theta series θ' of the construction A lattice is explicitly given, consisting of a truncated theta series of $2Z_n$ lattice and a term from minimum weight codewords of binary code. The truncated union bound is computed from θ' to estimate the WER. The OSD algorithm is considered for decoding binary codes to achieve a near-ML error performance.

Design examples are given using $n = 128$ EBCH codes and polar codes, where polar codes described in [78] are considered. Numerical evaluations showed that the truncated union bound analysis gives an accurate estimate of the decoding error probability at $P_e \leq 10^{-4}$. Since the $2Z_n$ lattice dominates the error performance very quickly, high rate binary codes are suggested to form the lattice for $P_e = 10^{-4}$ to 10^{-8}. It is noticed that the all suggested codes have $d_c > 4$, which implies that the balanced distance rule is not the best for the design scenario we considered here. In particular, at dimension $n = 128$, $(128, 106, 8)$ EBCH code forms the best construction A lattice at $P_e = 10^{-5}$, among our design examples.

A comparison with the classic balanced distance rule and the equal error probability rule is given. Unconstrained lattices and lattice codes are considered for this comparison, respectively. Simulation results showed that our design achieves lower WER than that of the classic balanced distance rule

and the equal error probability rule. The best construction A lattice obtained from design examples does not achieve lower WER than that of construction D lattices in [20] and [22]. However, for EBCH code lattices, the proposed construction A lattice achieves much lower decoding complexity compared to construction D in [20]. On the other hand, for polar code lattices, the proposed design gives an analytic approach to optimize the component codes, while the construction D lattices in [22] are designed based on Monte-Carlo simulations.

Some observations and extensions for this design are discussed as follows to conclude this chapter.

1. For polar codes described in [78], τ_c can be found analytically for any valid n. The design for polar code lattices can be extended to high dimensions to apply the capacity rule, which is another classic rule suggested in [72], and considered in [21] for designing construction D polar code lattices.
2. It is observed that the best polar code is also the $RM(4, 7)$ Reed-Muller code. The study for higher dimensions may help us to understand this relationship.
3. From the numerical simulation, construction D lattices achieve lower WER than that of construction A. A valid extension is to apply this approach to construction D lattices at $n = 128$ with all component codes having known d_c and τ_c. Even though the decoding complexity might be increased using construction D, this could lead us to find the lattice achieving the lowest WER at dimension $n = 128$.
4. At last, in this chapter, only EBCH codes and polar codes are considered. There exists other good-performing binary codes, such conventional codes, LDPC codes and etc. It could be worthwhile to explore more types of component codes to give the lattice design.

Chapter 7

Conclusion and future works

7.1 Conclusion

This dissertation provided a study of finite dimensional lattice codes for wireless communications. The point-to-point single user transmission through AWGN channel and multiple access relay performing compute-forward through fading channel are considered as the communication scenarios for evaluation and implementation.

Instead of conventional one-shot lattice decoder using the optimal decoding coefficient(s), a new retry decoding scheme is proposed for finite dimensional lattice codes. It is shown that the error performance can be improved, if the receiver is allowed to retry decoding by adjusting the value(s) of decoding coefficient(s) when error is detected. CRC-embedded lattices/lattice codes are then introduced, which enable the retry decoding by adding physical layer error detection ability to lattices/lattice codes. By appropriately designing the shaping lattice, retry decoding with CRC-embedded lattice codes is feasible for PLNC using CF relaying. This can not only improve the error performance, but also prevent forwarding erroneous packets into network, since a CF relay was not possible to detect decoding errors from a single linear combination. Lattice codes design for the proposed retry decoding scheme was discussed, from which: 1) for the SU scenario, low dimensional lattice codes with high code rate are suitable; 2) for the CF relaying, medium dimension lattice codes can be implemented to achieve a non-trivial gain. More significantly, using the CRC-embedded lattice codes, gains of 1.29dB, 1.31dB and 1.08dB at a 2-user CF relay can be achieved by using $n = 64, 128, 256$ polar lattice codes with code rate $R \approx 1.6406, 1.7422, 1.8438$, respectively, where only one additional decoding attempt is required. At last, a truncated union bound based construction A lattice design was considered for future lattice-based communications, which optimizes the lattice design with respect to maximum likelihood decoding. Lower error rate can be achieved by applying the proposed design approach, compared to that by the classic balanced distance rule and the equal probability rule.

7.2 Future works

Besides the contributions we mentioned above, there are also other interesting topics and extensions, which can be considered as possible future works.

Lattice codes vs. QAM
Shaping gain is one the advantage of lattice codes, which reduces the codebook power compared to conventional QAM constellation. It could be interesting to investigate the performance of lattice codes vs. QAM constellation showing the potential of lattice codes for next-generation wireless communications.

Retry vs. retransmission
In this work, it was demonstrated that retry decoding by adjusting the value(s) of decoding coefficient(s) can improve the error performance without requesting a retransmission. Since retry decoding is a local process, a lower decoding latency can be expected compared to retransmission. However, requesting retransmission may achieve higher diversity gain due to the channel reuse. A trade-off exists between the decoding latency and error performance for receivers considering retry decoding and retransmission, respectively.

Application in other CF-like scenarios
The proposed retry decoding with CRC-embedded lattice codes is feasible for CF relaying if the shaping lattice design follows Section 5.2.4. It is valid to consider the extension to apply retry decoding with CRC-embedded lattice codes to other CF-like scenarios, such as compute-forward multiple access [32], IF MIMO receiver [35] and other lattice-based PLNC scenarios [53].

Comparing to other CRC-aided decoding algorithms
There are other CRC-aided decoding algorithms, such as CRC-aided successive cancellation list (SCL) decoding for polar codes [71, 84]. It could be interesting to study the two different CRCs. It is noted that the proposed CRC-embedded lattice codes detects errors from a lattice codeword and CRC-aided SCL detects errors from a binary codeword. It could be possible to consider new schemes, such as 1) using the CRC for SCL decoder to apply retry decoding if all the candidates output from SCL decoder failed parity check; 2) designing dual CRC-embedded lattice codes, for SCL and retry decoding individually, to provide extra error detection for binary and lattice codeword respectively.

Construction A lattice design in larger dimensions
The design of construction A lattices in Chapter 6 considered medium dimensional lattices. The codeword multiplicity of polar codes, de-

scribed in [78], can be found for any valid n. The design for larger dimensions can be considered using polar codes, at which the classic capacity rule [72] can be applied.

Learning-based coefficient selecting algorithm

In this work, retry decoding requires a coefficient list at decoder, where exhaustive search based algorithms are implemented to generate the list. For single user transmission, the coefficient list is only related to the lattice code and the SNR values; while for CF relaying, channel fading coefficients are additionally involved. Learning-based coefficient selection algorithms can be considered which explores the relationship between physical coefficients, such as SNR and fading coefficient, and decoding coefficient, in order to reduce the complexity of generating the list using exhaustive search.

Appendix A

Generator matrices and coset leaders of BW_{16} lattice and Leech lattice

A.1 BW_{16} lattice

A generator matrix of the BW_{16} is given as:

$$\mathbf{G} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\
1 & 0 & 0 & 0 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 4
\end{bmatrix}. \tag{A.1}$$

For $\mathbf{G}$ given above, the matrix of coset leaders for decoding the BW_{16} lattice is given as following, where each row $\mathbf{c}_i$ indicates the transpose of a coset leader.

$$\mathbf{C}^T = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
0 & -1 & 0 & 0 & -1 & -1 & 0 & 1 & 0 & -1 & 1 & -1 & -1 & 0 & 0 & 0 \\
-1 & 0 & 1 & -1 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 & -1 & 0 & 1 & -1 & -1 & 1 & 0 & 0 \\
-1 & -1 & 0 & -1 & -1 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 1 \\
0 & -1 & 1 & 0 & -1 & 0 & -1 & 1 & -1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\
-1 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & -1 & 1 \\
0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & -1 & 0 & -1 & -1 & 1 & -1 & 0 \\
-1 & -1 & 1 & 0 & -1 & 1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & -1 & 0 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 \\
-1 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & -1 & -1 & 0 & 0 & 1 \\
0 & 0 & 1 & -1 & 0 & 1 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & -1 & 0 \\
-1 & -1 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & 1 \\
0 & -1 & 1 & -1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & -1 & 0 & 1 & 0 \\
-1 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & -1 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & -1 & 0 & 1 & 0 & -1 & 1 & -1 & 1 \\
-1 & -1 & 1 & -1 & 0 & 1 & -1 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & -1 & 0 & -1 & 0 & 1 & -1 & 1 \\
-1 & 0 & 1 & -1 & -1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & -1 & 1 & -1 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 \\
-1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & -1 & 1 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 1 & 0 & -1 & 0 & -1 & 1 \\
-1 & 0 & 0 & -1 & -1 & -1 & 0 & 1 & -1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & -1 & -1 & 0 & -1 & 0 & -1 & -1 & 1 & -1 & 0 & 0 & 0 & 1 \\
-1 & -1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 0 \\
0 & -1 & 0 & -1 & 0 & -1 & -1 & 1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\
-1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 0 & -1 & -1 & 0 \\
0 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 1 & 0 & 1 \\
-1 & -1 & 0 & 0 & 0 & 0 & -1 & 1 & -1 & 0 & 1 & -1 & 0 & 0 & -1 & 0 \\
0 & -1 & 1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0 & -1 & 0 & 1 \\
-1 & 0 & 0 & 0 & -1 & -1 & -1 & 0 & -1 & -1 & 0 & 0 & -1 & 0 & -1 & 0
\end{bmatrix} \tag{A.2}$$

A.2 Leech lattice

A generator matrix of the Leech lattice is given as:

$$\mathbf{G} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 2 & 2 & 2 & 2 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 2 & 2 & 2 & 2 & 2 & 2 & 0 & 2 & 2 & 2 & 0 & 2 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 2 & 2 & 2 & 2 & 0 & 2 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \\
1 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\
1 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\
-3 & 0 & 0 & 0 & 2 & 2 & 2 & 4 & 2 & 2 & 2 & 4 & 2 & 4 & 4 & 4 & 2 & 4 & 4 & 4 & 4 & 4 & 4 & 8
\end{bmatrix}. \tag{A.3}$$

The transpose of matrices $\mathbf{A}$, $\mathbf{T}$ and the cross reference table on indexing $\mathbf{c}$ for implementing Leech lattice decoding proposed in [48] are given as following.

$$\mathbf{A}^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \\ -2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 2 & 2 & 0 & 0 \\ -2 & 2 & 0 & 0 & 2 & 2 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 & 2 & 2 \\ -2 & 2 & 0 & 0 & 0 & 0 & 2 & 2 \\ 2 & 0 & 2 & 0 & 2 & 0 & 2 & 0 \\ -2 & 0 & 2 & 0 & 2 & 0 & 2 & 0 \\ 2 & 0 & 2 & 0 & 0 & 2 & 0 & 2 \\ -2 & 0 & 2 & 0 & 0 & 2 & 0 & 2 \\ 2 & 0 & 0 & 2 & 2 & 0 & 0 & 2 \\ -2 & 0 & 0 & 2 & 2 & 0 & 0 & 2 \\ 2 & 0 & 0 & 2 & 0 & 2 & 2 & 0 \\ -2 & 0 & 0 & 2 & 0 & 2 & 2 & 0 \end{bmatrix}. \tag{A.4}$$

$$\mathbf{T}^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 & 2 & 0 & 0 \\ 2 & 2 & 0 & 2 & 0 & 0 & 0 & 2 \\ 2 & 0 & 2 & 2 & 0 & 0 & 2 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 2 & 0 & 2 & 0 \\ 2 & 0 & 2 & 0 & 2 & 0 & 0 & 2 \\ 2 & 0 & 0 & 2 & 2 & 2 & 0 & 0 \\ -3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & -1 & -1 & 1 & 1 & -1 & 1 & 1 \\ 3 & -1 & 1 & -1 & 1 & 1 & 1 & -1 \\ 3 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\ 3 & 1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 3 & -1 & 1 & 1 & -1 & 1 & -1 & 1 \\ 3 & 1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 3 & 1 & 1 & -1 & -1 & -1 & 1 & 1 \end{bmatrix} \tag{A.5}$$

$$\mathbf{C}_{index} = \begin{bmatrix}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\
2 & 1 & 4 & 3 & 6 & 5 & 8 & 7 & 10 & 9 & 12 & 11 & 14 & 13 & 16 & 15 \\
3 & 4 & 1 & 2 & 7 & 8 & 5 & 6 & 11 & 12 & 9 & 10 & 15 & 16 & 13 & 14 \\
4 & 3 & 2 & 1 & 8 & 7 & 6 & 5 & 12 & 11 & 10 & 9 & 16 & 15 & 14 & 13 \\
5 & 6 & 7 & 8 & 1 & 2 & 3 & 4 & 13 & 14 & 15 & 16 & 9 & 10 & 11 & 12 \\
6 & 5 & 8 & 7 & 2 & 1 & 4 & 3 & 14 & 13 & 16 & 15 & 10 & 9 & 12 & 11 \\
7 & 8 & 5 & 6 & 3 & 4 & 1 & 2 & 15 & 16 & 13 & 14 & 11 & 12 & 9 & 10 \\
8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 16 & 15 & 14 & 13 & 12 & 11 & 10 & 9 \\
9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
10 & 9 & 12 & 11 & 14 & 13 & 16 & 15 & 2 & 1 & 4 & 3 & 6 & 5 & 8 & 7 \\
11 & 12 & 9 & 10 & 15 & 16 & 13 & 14 & 3 & 4 & 1 & 2 & 7 & 8 & 5 & 6 \\
12 & 11 & 10 & 9 & 16 & 15 & 14 & 13 & 4 & 3 & 2 & 1 & 8 & 7 & 6 & 5 \\
13 & 14 & 15 & 16 & 9 & 10 & 11 & 12 & 5 & 6 & 7 & 8 & 1 & 2 & 3 & 4 \\
14 & 13 & 16 & 15 & 10 & 9 & 12 & 11 & 6 & 5 & 8 & 7 & 2 & 1 & 4 & 3 \\
15 & 16 & 13 & 14 & 11 & 12 & 9 & 10 & 7 & 8 & 5 & 6 & 3 & 4 & 1 & 2 \\
16 & 15 & 14 & 13 & 12 & 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1
\end{bmatrix} \tag{A.6}$$

For example, assuming $\mathbf{a} = [2, 2, 2, 2, 0, 0, 0, 0]^T$ and $\mathbf{b} = [-2, 2, 2, 2, 0, 0, 0, 0]$ are selected. This indicates the row index of $\mathbf{a}$ is $i_a = 3$ and the row index of $\mathbf{b}$ is $i_b = 3$, from which the row index of $\mathbf{c}$ can be referred from (A.6) as $i_c = C_{index}(i_a, i_b) = 2$ and $\mathbf{c} = [4, 0, 0, 0, 0, 0, 0, 0]^T$.

References

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on information theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[2] R. Zamir, "Lattices are everywhere," in *2009 Information Theory and Applications Workshop*. IEEE, 2009, pp. 392–421.

[3] J. H. Conway and N. J. A. Sloane, *Sphere packings, lattices and groups*, 2nd ed. Springer, 1993.

[4] M. S. Viazovska, "The sphere packing problem in dimension 8," *Annals of mathematics*, pp. 991–1015, 2017.

[5] H. Cohn, A. Kumar, S. Miller, D. Radchenko, and M. Viazovska, "The sphere packing problem in dimension 24," *Annals of mathematics*, vol. 185, no. 3, pp. 1017–1033, 2017.

[6] D. Micciancio and S. Goldwasser, *Complexity of lattice problems: a cryptographic perspective*. Springer Science & Business Media, 2002, vol. 671.

[7] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," in *International algorithmic number theory symposium*. Springer, 1998, pp. 267–288.

[8] U. Erez, S. Litsyn, and R. Zamir, "Lattices which are good for (almost) everything," *IEEE Transactions on Information Theory*, vol. 51, no. 10, pp. 3401–3416, 2005.

[9] G. Poltyrev, "On coding without restrictions for the AWGN channel," *IEEE Transactions on Information Theory*, vol. 40, no. 2, pp. 409–417, 1994.

[10] R. de Buda, "Some optimal codes have structure," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 6, pp. 893–899, 1989.

[11] T. Linder, C. Schlegal, and K. Zeger, "Corrected proof of de Buda's theorem (lattice channel codes)," *IEEE transactions on information theory*, vol. 39, no. 5, pp. 1735–1737, 1993.

[12] R. Urbanke and B. Rimoldi, "Lattice codes can achieve capacity on the AWGN channel," *IEEE transactions on Information Theory*, vol. 44, no. 1, pp. 273–278, 1998.

[13] R. de Buda, "The upper error bound of a new near-optimal code," *IEEE Transactions on Information Theory*, vol. 21, no. 4, pp. 441–445, 1975.

[14] U. Erez and R. Zamir, "Achieving 1/2 log (1 + SNR) on the AWGN channel with lattice encoding and decoding," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2293–2314, 2004.

[15] N. Di Pietro, J. J. Boutros, G. Zémor, and L. Brunel, "Integer low-density lattices based on construction A," in *2012 IEEE Information Theory Workshop*. IEEE, 2012, pp. 422–426.

[16] N. Di Pietro, G. Zémor, and J. J. Boutros, "LDA lattices without dithering achieve capacity on the gaussian channel," *IEEE Transactions on Information Theory*, vol. 64, no. 3, pp. 1561–1594, 2018.

[17] S. Vatedka and N. Kashyap, "Some "goodness" properties of LDA lattices," *Problems of Information Transmission*, vol. 53, no. 1, pp. 1–29, 2017.

[18] N. Sommer, M. Feder, and O. Shalvi, "Low-density lattice codes," *IEEE Transactions on Information Theory*, vol. 54, no. 4, pp. 1561–1585, 2008.

[19] R. A. P. Hernandez and B. M. Kurkoski, "The three/two gaussian parametric ldlc lattice decoding algorithm and its analysis," *IEEE Transactions on Communications*, vol. 64, no. 9, pp. 3624–3633, 2016.

[20] T. Matsumine, B. M. Kurkoski, and H. Ochiai, "Construction D lattice decoding and its application to BCH code lattices," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[21] L. Liu, Y. Yan, C. Ling, and X. Wu, "Construction of capacity-achieving lattice codes: Polar lattices," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 915–928, 2018.

[22] O. R. Ludwinianda, N. Liu, K. Anwar, and B. M. Kurkoski, "Design of polar code lattices of finite dimension," in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 1011–1016.

[23] F. Zhou and B. M. Kurkoski, "Construction D' lattices for power-constrained communications," *IEEE Transactions on Communications*, vol. 70, no. 4, pp. 2200–2212, 2022.

[24] S. Chen, B. M. Kurkoski, and E. Rosnes, "Construction D' lattices from quasi-cyclic low-density parity-check codes," in *2018 IEEE 10th International Symposium on Turbo Codes & Iterative Information Processing (ISTC)*. IEEE, 2018, pp. 1–5.

[25] R. Zamir, *Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation, and Multiuser Information Theory*. Cambridge University Press, 2014.

[26] B. M. Kurkoski, "Encoding and indexing of lattice codes," *IEEE Transactions on Information Theory*, vol. 64, no. 9, pp. 6320–6332, 2018.

[27] N. Di Pietro and J. J. Boutros, "Leech constellations of construction-A lattices," *IEEE Transactions on Communications*, vol. 65, no. 11, pp. 4622–4631, 2017.

[28] F. Zhou and B. M. Kurkoski, "Shaping ldlc lattices using convolutional code lattices," *IEEE Communications Letters*, vol. 21, no. 4, pp. 730–733, 2017.

[29] C. Feng, D. Silva, and F. R. Kschischang, "An algebraic approach to physical-layer network coding," *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7576–7596, 2013.

[30] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6463–6486, 2011.

[31] B. Nazer, V. R. Cadambe, V. Ntranos, and G. Caire, "Expanding the compute-and-forward framework: Unequal powers, signal levels, and multiple linear combinations," *IEEE Transactions on Information Theory*, vol. 62, no. 9, pp. 4879–4909, 2016.

[32] J. Zhu and M. Gastpar, "Gaussian multiple access via compute-and-forward," *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 2678–2695, 2017.

[33] J. Zhan, B. Nazer, M. Gastpar, and U. Erez, "MIMO compute-and-forward," in *2009 IEEE International Symposium on Information Theory*. IEEE, 2009, pp. 2848–2852.

[34] J. Zhan, B. Nazer, U. Erez, and M. Gastpar, "Integer-forcing linear receivers," in *2010 IEEE International Symposium on Information Theory*. IEEE, 2010, pp. 1022–1026.

[35] ——, "Integer-forcing linear receivers," *IEEE Transactions on Information Theory*, vol. 60, no. 12, pp. 7661–7685, 2014.

[36] A. Sakzad, J. Harshan, and E. Viterbo, "Integer-forcing MIMO linear receivers based on lattice reduction," *IEEE transactions on wireless communications*, vol. 12, no. 10, pp. 4905–4915, 2013.

[37] M. N. Hasan and B. M. Kurkoski, "Practical compute-and-forward approaches for the multiple access relay channel," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[38] E. Sula, J. Zhu, A. Pastore, S. H. Lim, and M. Gastpar, "Compute–forward multiple access (CFMA): Practical implementations," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1133–1147, 2018.

[39] O. Ordentlich, J. Zhan, U. Erez, M. Gastpar, and B. Nazer, "Practical code design for compute-and-forward," in *2011 IEEE International Symposium on Information Theory Proceedings*. IEEE, 2011, pp. 1876–1880.

[40] A. Sakzad, E. Viterbo, J. Boutros, and Y. Hong, "Phase precoded compute-and-forward with partial feedback," in *2014 IEEE International Symposium on Information Theory*. IEEE, 2014, pp. 2117–2121.

[41] N. S. Ferdinand, M. Nokleby, B. M. Kurkoski, and B. Aazhang, "MMSE scaling enhances performance in practical lattice codes," in *2014 48th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2014, pp. 1021–1025.

[42] A. Mejri and G. R.-B. Othman, "Practical physical layer network coding in multi-sources relay channels via the compute-and-forward," in *2013 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. IEEE, 2013, pp. 166–171.

[43] R. Zamir and M. Feder, "On lattice quantization noise," *IEEE Transactions on Information Theory*, vol. 42, no. 4, pp. 1152–1159, 2002.

[44] J. Conway and N. Sloane, "Fast quantizing and decoding and algorithms for lattice quantizers and codes," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 227–232, 1982.

[45] G. D. Forney, "Coset codes. i. introduction and geometrical classification," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 1123–1151, 1988.

[46] J. Leech, "Notes on sphere packings," *Canadian Journal of Mathematics*, vol. 19, pp. 251–267, 1967.

[47] J. H. Conway and N. J. Sloane, "On the voronoi regions of certain lattices," *SIAM Journal on Algebraic Discrete Methods*, vol. 5, no. 3, pp. 294–305, 1984.

[48] J. Conway and N. Sloane, "Soft decoding techniques for codes and lattices, including the golay code and the leech lattice," *IEEE Transactions on Information Theory*, vol. 32, no. 1, pp. 41–50, 1986.

[49] Y. Be'ery, B. Shahar, and J. Snyders, "Fast decoding of the leech lattice," *IEEE journal on selected areas in communications*, vol. 7, no. 6, pp. 959–967, 1989.

[50] A. Vardy and Y. Be'ery, "Maximum likelihood decoding of the leech lattice," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1435–1444, 1993.

[51] O. Amrani, Y. Be'ery, A. Vardy, F.-W. Sun, and H. C. Van Tilborg, "The leech lattice and the golay code: bounded-distance decoding and multilevel constructions," *IEEE Transactions on Information Theory*, vol. 40, no. 4, pp. 1030–1043, 1994.

[52] A. Vardy, "Even more efficient bounded-distance decoding of the hexacode, the golay code, and the leech lattice," *IEEE Transactions on Information Theory*, vol. 41, no. 5, pp. 1495–1499, 1995.

[53] T. Yang, "On lattice network coding based cell-free MIMO with uncoordinated base stations," *IEEE Transactions on Wireless Communications*, 2024.

[54] E. Barnes and N. Sloane, "New lattice packings of spheres," *Canadian Journal of Mathematics*, vol. 35, no. 1, pp. 117–130, 1983.

[55] W. Kositwattanarerk and F. Oggier, "Connections between construction D and related constructions of lattices," *Designs, codes and cryptography*, vol. 73, pp. 441–455, 2014.

[56] Y. Yan, C. Ling, and X. Wu, "Polar lattices: where arıkan meets forney," in *2013 IEEE International Symposium on Information Theory*. IEEE, 2013, pp. 1292–1296.

[57] Q. Chen, F. Yu, T. Yang, and R. Liu, "Gaussian and fading multiple access using linear physical-layer network coding," *IEEE Transactions on Wireless Communications*, vol. 22, no. 5, pp. 3099–3113, 2022.

[58] X. Qiu, T. Yang, and J. Thompson, "On lattice-based broadcasting for massive-user MIMO: Practical algorithms and optimization," *IEEE Transactions on Wireless Communications*, 2024.

[59] V. Tarokh, A. Vardy, and K. Zeger, "Universal bound on the performance of lattice codes," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 670–681, 1999.

[60] A. Blum, J. Hopcroft, and R. Kannan, *Foundations of data science*. Cambridge University Press, 2020.

[61] A. Lenstra, H. Lenstra Jr., and L. Lovász, "Factoring polynomials with rational coefficients." *Mathematische Annalen*, vol. 261, no. 4, pp. 515–534, 1982.

[62] A. Sakzad, E. Viterbo, Y. Hong, and J. Boutros, "On the ergodic rate for compute-and-forward," in *2012 International Symposium on Network Coding (NetCod)*. IEEE, 2012, pp. 131–136.

[63] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of computation*, vol. 44, no. 170, pp. 463–471, 1985.

[64] C.-P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Mathematical programming*, vol. 66, pp. 181–199, 1994.

[65] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Transactions on Information theory*, vol. 45, no. 5, pp. 1639–1642, 1999.

[66] A. Sakzad, E. Viterbo, J. J. Boutros, and Y. Hong, "Phase precoding for the compute-and-forward protocol," *arXiv preprint arXiv:1404.4157*, 2014.

[67] W. Liu and C. Ling, "Efficient integer coefficient search for compute-and-forward," *IEEE Transactions on Wireless Communications*, vol. 15, no. 12, pp. 8039–8050, 2016.

[68] S. Sahraei and M. Gastpar, "Compute-and-forward: Finding the best equation," in *2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2014, pp. 227–233.

[69] P. Koopman and T. Chakravarty, "Cyclic redundancy code (crc) polynomial selection for embedded networks," in *International Conference on Dependable Systems and Networks, 2004*. IEEE, 2004, pp. 145–154.

[70] J. Xue and B. M. Kurkoski, "Lower bound on the error rate of genie-aided lattice decoding," in *2022 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2022, pp. 3232–3237.

[71] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE transactions on information theory*, vol. 61, no. 5, pp. 2213–2226, 2015.

[72] U. Wachsmann, R. F. Fischer, and J. B. Huber, "Multilevel codes: Theoretical concepts and practical design rules," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1361–1391, 1999.

[73] 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, 2009.

[74] M. P. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Transactions on Information Theory*, vol. 41, no. 5, pp. 1379–1396, 1995.

[75] Y. Desaki, T. Fujiwara, and T. Kasami, "The weight distributions of extended binary primitive bch codes of length 128," *IEEE Transactions on Information Theory*, vol. 43, no. 4, pp. 1364–1371, 1997.

[76] T. Fujiwara, "The weight distribution of $(256, k)$ extended binary primitive bch codes with $k \leq 63$ and $k \geq 207$," *IEICE Technical Report*, pp. 29–33, 1997.

[77] J. A. M. Terada and T. Koumoto, "Weight distribution of extended BCH codes," https://https://isec.ec.okayama-u.ac.jp/home/kusaka/wd/index.html.

[78] M. Rowshan, S. H. Dau, and E. Viterbo, "On the formation of min-weight codewords of polar/pac codes and its applications," *IEEE Transactions on Information Theory*, 2023.

[79] M. Bardet, V. Dragoi, A. Otmani, and J.-P. Tillich, "Algebraic properties of polar codes from a new polynomial formalism," in *2016 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2016, pp. 230–234.

[80] ——, "Algebraic properties of polar codes from a new polynomial formalism," *arXiv preprint arXiv:1601.06215*, 2016.

[81] J. H. Van Lint, *Introduction to coding theory*, 2nd ed. Springer, 1992.

[82] N. Sloane, "Weight enumerators of codes," in *Combinatorics: Proceedings of the NATO Advanced Study Institute held at Nijenrode Castle, Breukelen, The Netherlands 8–20 July 1974*. Springer, 1975, pp. 115–142.

[83] M. V. Eyuboglu and G. D. Forney, "Lattice and trellis quantization with lattice-and trellis-bounded codebooks-high-rate theory for memoryless sources," *IEEE Transactions on Information Theory*, vol. 39, no. 1, pp. 46–59, 1993.

[84] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE communications letters*, vol. 16, no. 10, pp. 1668–1671, 2012.

Achievements

Publications

[1] J. Xue and B. M. Kurkoski, "Lower Bound on the Error Rate of Genie-Aided Lattice Decoding," 2022 IEEE International Symposium on Information Theory (ISIT), Espoo, Finland, 2022, pp. 3232-3237, doi: 10.1109/ISIT50566.2022.9834527.

[2] J. Xue and B. M. Kurkoski, "Finite Dimensional Lattice Codes with Self Error-Detection and Retry Decoding," in IEEE Transactions on Communications, early access, doi: 10.1109/TCOMM.2025.3560348.

[3] J. Xue, B. M. Kurkoski and E. Viterbo, "Construction A Lattice Design Based on the Truncated Union Bound," submitted to 2025 IEEE Information Theory Workshop, available at https://arxiv.org/abs/2502.10728.

Presentations

[1] J. Xue, B. M. Kurkoski, and E. Viterbo, In search of the best dimension 128 lattice: Construction A, Presentation at Information Theory and Applications Workshop, February 2025, University of California San Diego, USA, Feb. 2025.

[2] J. Xue and B. M. Kurkoski, CRC-enabled lattices for multiuser communication, Presentation at Information Theory and Applications Workshop, February 2023, University of California San Diego, USA, Feb. 2023.

[3] J. Xue and B. M. Kurkoski, Lattice decoding based on exhaustive search over scaling factor, IEICE Technical Report; IEICE Tech. Rep., July 2022, Okayama University of Science, Okayama, Japan, Jul. 2022.