On Optimal Codes for Additive Noise Channels

by

Al Mahdi Moufid

A thesis submitted to the
Department of Mathematics and Statistics
in conformity with the requirements for
the degree of Master of Applied Science

Queen's University
Kingston, Ontario, Canada
September 2020

Abstract

One of the fundamental goals of coding theory is to find optimal codes, in terms of achieving the largest probability of correct decoding. Most of the foundational works on optimal codes use three assumptions. First, optimality is defined relative to a code's rate and minimum Hamming distance [19, Section 2.4], and not its probability of correct decoding. The problem is that the former conditions are not always good indicators for the latter. Second, for practicality only optimal codes with some kind of algebraic structure are considered, the most common being linearity. While linear codes have practical advantages, it is known that often the best codes are not linear [20, Page 259]. Thirdly, codes are usually proven optimal only for memoryless channels. However, there are many real-life channels which exhibit memory [19]. Attempts to extend these results often make use of interleaving. This technique renders some of these channels with memory equivalent to a memoryless one with respect to the decoder. However, this comes with two disadvantages namely: we fail to exploit the channel's memory and we add latency to the communication system.

In this thesis, we seek optimal codes which require milder conditions than those aforementioned. Specifically, we describe a class of optimal q-ary block codes over additive noise channels, which are a generalization of a class of optimal codes from [16].

We prove these codes are optimal using our novel method of analyzing the probability of correct decoding of block codes based on so-called error indexed decoder regions. Our method allows us to prove that these codes must be either a linear code or a coset of one.

Acknowledgments

I would like to express my profound gratitude to Professors Fady Alajaji and David Wehlau, whose support, expertise, and patience made my masters studies fruitful. I am fortunate to have had them as supervisors. I am also incredibly grateful to my family and friends whose unconditional love and support guided me throughout my graduate studies.

Contents

List of Tables

List of Figures

List of Important Symbols

Φ	A generalized syndrome decoder
$\mathcal{C}$	An $[n, k]$-code
$\mathcal{E}_{\mathcal{C}}(\boldsymbol{y})$	The error set of $\boldsymbol{y}$, with respect to $\mathcal{C}$
$\boldsymbol{\mathcal{E}}(\mathcal{C})$	The set of all possible error sets of $\mathcal{C}$
$\boldsymbol{e_L}(\boldsymbol{y})$	The error set leader of $\boldsymbol{y}$
$\mathcal{D}(\mathcal{C})$	The decoding set of $\mathcal{C}$
$\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})$	The decoder region of $\boldsymbol{e}$, with respect to $\mathcal{C}$
$\boldsymbol{\mathcal{R}}(\mathcal{C})$	The set of all possible decoder regions of $\mathcal{C}$

List of Acronyms

BFMNC Binary First-Order Markov Noise Channel

BSC Binary Symmetric Channel

MD Minimum (Hamming) Distance

MLD Maximum Likelihood Decoder

GSD Generalized Syndrome Decoder

PCD Probability of Correct Decoding

Chapter 1

Introduction

1.1 Problem Description

In a landmark paper [24], Shannon presented his *channel coding theorem* which proved that every channel has an upper limit on the rate of transmission, called capacity. Information can only be transmitted reliably below this rate. He also proved the existence of channel codes which can achieve this capacity with an arbitrarily small decoding error probability. His results provide a mathematical basis for coding theory. However, from a practical standpoint, the channel coding theorem left much unanswered. His proof for the existence of good codes used a random coding (probabilistic) argument and did not provide an explicit method for their construction. In this work, we seek to identify and analyze such good codes.

Specifically, we are interested in finding optimal codes for a class of noisy channels.

An optimal code is defined as one achieving the largest probability of correct decoding (PCD) among all codes of the same length and dimension. Prior fundamental analytical work on the subject uses a mix of three assumptions which greatly limit the applicability of the results. The first, and perhaps the most limiting, is defining optimality relative to a code's rate and minimum Hamming distance.

Classically, an *optimum*[1] code is defined as one with the largest number of codewords for a fixed length and minimum distance (MD) [19, Page 58]. In other words, an optimum code is one which has the largest possible rate for a given amount of error correction. This definition is quite limiting for our goal of finding codes with maximal PCD. For the binary symmetric channel (BSC), the MD can be used to lower bound a code's error correction [19, Section 1.3], and hence also its PCD [19, Section 1.5]. Thus, it is natural to use the MD to guide our search for codes with maximal PCD. This leads us to seek codes with *maximal* MD.

For some special cases, it has been proven that codes with maximal MD also achieves the maximal PCD. One such special case, is *nearly perfect codes*. A well known example being Preparata codes [21] which are also non-linear.[2] Nearly perfect codes, a subset of the larger family of quasi-perfect codes,[3] are optimal over the BSC [13, Page 164] [19, Section 17.3]. The search for nearly perfect codes was thorough, and all parameters for which they exist are known [18]. However, these types of codes only exist for a handful of parameters, and so we must seek alternative families of codes.

We can attempt to find optimal codes in general by searching for codes with

[1] We will use the word 'optimum' to distinguish codes satisfying the classical definition of optimality, and 'optimal' for those which satisfies ours.

[2] See [19] for a good treatment of important non-linear codes.

[3] Quasi-perfect codes form part of the class of *uniformly packed codes* which have similar nice properties [29].

maximal MD. However in general, codes with maximal MD are not guaranteed to have the largest PCD. In fact, there exists codes with the *minimal* MD which achieve the largest PCD.[4] In other words, a large MD *does not imply* better performance than codes with a smaller MD. Furthermore, apart from our carefully chosen example, most works on optimum codes focus on linear codes.

In an attempt to find codes which are practical to implement it is natural to impose some kind of structure on them. For practical applications, linear codes are the most important class of codes, with specifically, cyclic codes being the most studied of all codes [19]. However, it is known that the best codes are often not even linear even for the BSC [20, Section 11.2]. Furthermore, some non-linear codes are quite practical, in the sense that they admit efficient encoding and decoding algorithms. For example, Preparata codes admit such encoding and decoding algorithms [21]. Furthermore, there has been some recent work on developing general encoding and decoding algorithms for non-linear codes which can be represented as the union of cosets [5, 31]. Therefore, in this work we will consider both linear and non-linear codes. We also consider a larger class of channels than just memoryless ones.

Classically, optimality is defined relative to memoryless channels. However, many real life channels exhibit memory or statistical dependence in their noise process [19], particularly wireless communications channels. Radio based communication systems experience noise due to multi-path fading [22, Chapter 14]. These fading channels are modelled using channels with memory. For example, the finite state markov channel (FSMC) [23, 32] and the queue based channel (QBC) [34, 35] are two channels with memory well suited to modeling fading. The analysis of codes over such channels often assume the use of interleaving before transmission. Interleaving is a technique

[4]We provide a concrete example of this counter intuitive phenomenon in Appendix A.

which distributes transmission errors uniformly over the codewords. This allows the decoder to operate as it would over a memoryless channel. This solution comes at the price of increased transmission delay and failure to exploit channel memory. The latter is especially important, since in several instances, channels with memory have shown an increased capacity over their memoryless (interleaved) counterparts [3, 8].

Our work substitutes these three assumptions for milder ones. First, our definition for optimality is not based on a code's rate and MD but solely on its PCD. Second, we consider a general block code. Third, we consider any additive noise channel. This large group of channels includes many with and without memory such as: the BSC, the Gilbert-Elliot channel [9, 14], the infinite memory Polya-contagion channel (IMCC) and its finite version (FMCC) both introduced in [3], the FSMC [32], and the QBC [34].

1.2 Literature Review

Related works on optimal codes is based on a mix of the three aforementioned assumptions. The most significant body of work is on optimal (non-)linear codes for memoryless channels. However, most of the classical works in this area do not seek optimal codes directly, i.e., codes with maximal PCD. Instead for practicality and tractability, they define optimum codes as those with the largest rate for a given MD. For the BSC, the MD can be used to lower bound a code's error correction. Thus, a code's MD serves as a convenient approximation for its PCD.

However, as elaborated in Section 1.1, it is difficult and cumbersome to find optimum codes which also have maximal PCD. One natural, but flawed, methodology

would be to seek optimum codes with maximal MD. This methodology suffers from three issues. First even over the BSC, the relationship between a code's MD and its PCD is not monotonically increasing. In other words, a code with a larger MD does not necessarily have a larger PCD. Indeed, even a code with maximal MD is not guaranteed to be optimal. Only in special cases, e.g., quasi-perfect, is a code with the largest MD guaranteed to have the largest PCD [13, Section 5.8]. This leads into the second issue, rarity. The problem of constructing quasi-perfect codes is a hard combinatorial problem with only a few known solutions [28, Section 5.1]. Furthermore, the classical definition of optimality requires codes to have largest rate, leaving us without results for most code parameters. Therefore, these works and their methodology is of limited use in finding or describing optimal codes in general, even for the BSC. Furthermore, the above methodology is less tenable for channels with memory.

Errors over channels with memory are not independent. If certain error bursts (or sequences) are more likely than others, then the Hamming distance is not a good distance function to use [28, Section 2.1]. Therefore some authors have sought generalizations to quasi-perfect codes with alternative metrics [16,30]. We will return to these works later. First, we present some recent work on finding optimal (non-)linear codes for memoryless channels which do not use a metric.

In [7], the authors present *weak flip codes* which are a family of linear and non-linear codes. For ultra-small code sizes, they are proven optimal over various binary memoryless channels. These codes are defined constructively by their codebook matrix [7, Definition 10] and have properties similar to linear codes [6]. While optimal only for some memoryless channels, in some cases, codes optimal over memoryless

channels can also be optimal over channels with memory.

In [1], the authors study the problem of maximum likelihood (ML) decoding of binary perfect and quasi-perfect codes over a binary channel with additive Markov noise. It is shown that for a range of channel conditions strict ML (SML) decoding is *near equivalent* to strict minimum distance (SMD) decoding. And for perfect codes SML *is equivalent* to SMD [1, Lemma 4.4]. The authors observe that under these conditions their results, combined with work from [15, Theorem 7.1], allows us to conclude that some Hamming codes are optimal over the Markov noise channel [1, Section 4]. These results are significantly extended in a follow up work, discussed next.

In [4], the authors study ML decoding over three channels with memory: IMCC, FMCC, and the QBC. For each of these channels, they derive sufficient conditions on a general code and the channel parameters such that ML and MD decoding are equivalent. This allows them to provide sufficient conditions for quasi-perfect and perfect codes to be optimal over these channels [4, Lemma 3.2 and 3.3]. Furthermore, they also present equivalent conditions for generalized quasi-perfect codes, studied in [15], to be optimal.

As mentioned before, the Hamming distance is not an effective metric to use for channels with memory. Therefore, some work has been done on generalizing the notion of (quasi-)perfect codes using other metrics. In [15], Hamada presents such a generalization of quasi-perfect codes based on a variant of the Fano metric. He proves that these codes are optimal over the class of channels which are homogenous with respect to the metric [15, Theorem 4.1]. He also presents a sufficient condition for optimality [15, Theorem 7.3] under a milder assumption, namely removing the

requirement that the function for the variant of the Fano metric is a distance measure. This latter idea is extended in [30], where a more general variant of the Fano metric is used to define another class of generalized quasi-perfect codes. These generalized quasi-perfect codes [30, Definition 3.2], includes Hamada's [15, Definition 4.1] as a special case and are optimal over symmetric channels [30, Theorem 3.1].

In a subsequent paper [16], Hamada explores a class of optimal linear codes over additive noise channels. However, unlike the previous work, these codes are not directly related to quasi-perfect codes. Instead, he defines them based on properties of their decoding set [16, Section 4]. He presents a set of conditions on the decoding set, which make them 'ideal', and proves that linear codes with an ideal decoding set are optimal [16, Section 5] over an arbitrary additive noise channel.

1.3 Thesis Contribution and Overview

One of our main contributions is the novel treatment of *error indexed decoder regions* (Chapter 4), and *partially correctable errors* (Section 4.2). These concepts allow for a unified analysis of both linear and non-linear codes over additive noise channels. They also lead to an interesting closed form expression for a code's PCD (Chapter 5). These results allow us to derive a class of optimal codes (Theorem 18) which include Hamada's optimal linear codes [16, Section 5]. Furthermore, we are able to characterize these optimal codes as being either linear or the translate of a linear code. The rest of this thesis is organized as follows.

Chapter 2 starts with a brief overview of additive noise channels, with explicit description of one such channel: the binary first-order Markov noise channel (BFMNC).

This channel will be used for most of the motivating examples we present. We then cover block codes, and maximum likelihood decoding. Chapter 3 is intended as a unified treatment of channel coding over additive noise channels for a general code. We derive from first principles several generalizations of classical coding theory concepts. We also show the limitations of some of these concepts for analyzing block codes (Section 3.5).

An alternative method of analysis is introduced in Chapter 4. It relies on the novel, to the best of our knowledge, treatment of *error indexed decoder regions* or simply decoder regions. We show how a code's decoder regions relate to error correction and the probability of correct decoding. This connection reveals the importance of both *correctable* and *partially correctable* errors (Section 4.2). Finally in Chapter 5, we express a code's PCD in terms of these decoder regions using the errors which indexes them and their size. We also prove that any linear code has the same PCD as any of its translates. Finally, we present a class of optimal codes, which includes the class of optimal linear codes from [16]. Furthermore, we prove that this class of codes contains only linear codes and translates of linear codes. Finally, Chapter 6 summarizes our results and explores future avenues of research.

Chapter 2

Preliminaries: Channel Models and Block Codes

2.1 Channel Models

A communication system consists of three main parts: the sender, the receiver and the connection which allows them to communicate. This connection or channel can take many forms such as a telegraph, a two way radio, a digital subscriber line, etc. However, we model all these channels in the same way. We treat a channel as a black box with an input and output with certain statistical properties.

With the advent of the digital revolution, most channel inputs consists of some symbol in a fixed and finite set called an alphabet. The output is also assumed to consist of some symbol in an alphabet. This type of channel is called a *discrete* channel. By convention, input and output alphabets are denoted by $\mathcal{X}$ and $\mathcal{Y}$, respectively.

These two alphabets need not be the same, but we will assume they are.[1]

We view both the input and output of the channel as outcomes of a pair of random variables: (X, Y). Typically in communication systems, the input X follows the uniform distribution over $\mathcal{X}$. We make this assumption throughout this work. However, the relationship between the input and output of the channel is not one-to-one.

We do not always receive what was intended, as the channel suffers from interference or noise. The effect of this noise is characterized by a probability distribution between the channel's input and output. This probability distribution may depend not only on the current input symbol but also on previous ones. If a channel's output depends on more than the current input, then we describe this channel as having *memory*.

The inputs and outputs to our channel occur over *discrete* time. For convenience, we examine the inputs and outputs over a finite window of time of length n. Each of these windows is called a *word*. These input and output words are represented by n-tuples $\boldsymbol{x} \in \mathcal{X}^n$ and $\boldsymbol{y} \in \mathcal{Y}^n$, respectively. The words sent and received at time k are denoted by $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$, respectively. Finally due to our channel's noise, if we send $\boldsymbol{x}$ we will receive $\boldsymbol{y}$ only with some probability $P_{Y^n|X^n}(\boldsymbol{y}|\boldsymbol{x})$. So formally, a channel is defined by the sequence $\{\mathcal{X}^n, P_{Y^n|X^n}(\cdot|\cdot), \mathcal{Y}^n\}_{n=1}^{\infty}$. However for some channels, the effect of the channel noise can be describe more explicitly.

[1] In this digital age, the channel's alphabets are often taken as $\mathcal{X} = \mathcal{Y} = \{0, 1\}$.

2.1.1 Additive Noise Channels

An *additive noise* channel is one whose output at time k is $y_k = x_k + z_k$ where $z_k \in \mathcal{Z}$ is the noise symbol at time k. This noise symbol is generated by some random process $\{Z_i\}_{i \in \mathbb{N}^*}$ where $\mathbb{N}^*$ are the natural numbers excluding zero. We will assume henceforth that the noise process is both stationary and independent from the channel's input. All of these properties results in a tractable expression for the channel output. In other words, for $\boldsymbol{x} \in \mathcal{X}^n$ with a resulting output $\boldsymbol{y} \in \mathcal{Y}^n$, we have

$$P_{Y^n|X^n}(\boldsymbol{y}|\boldsymbol{x}) = P_{Y^n|X^n}(X^n + Z^n = \boldsymbol{y}|\boldsymbol{x}) = P_{Z^n|X^n}(Z^n = \boldsymbol{y} - \boldsymbol{x}|X^n = \boldsymbol{x})$$

where addition and subtraction of the n-tuples is component wise as defined between the sets $\mathcal{X}$ and $\mathcal{Y}$. In this work, we will only consider modulo q additive noise channels where $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \mathbb{F}_q$, where $\mathbb{F}_q$ is the finite field of order q. Channel words are then n-tuples over $\mathbb{F}_q$ with addition between these tuples defined as component wise addition modulo q. A well known result of abstract algebra is that a finite field of order q exists if and only if q is a power of a prime. However it is conventional to instead consider words not as n-tuples, but as vectors from $\mathbb{F}_q^n$, the n-dimensional vector space over $\mathbb{F}_q$, with addition between vectors being the typical vector addition.

Note that additive noise channels with memory are "symmetric" in the sense that uniformly distributed input n-tuples X^n yield uniformly distributed output n-tuples Y^n and maximize the channel's normalized input-output mutual information $\frac{1}{n} I(X^n; Y^n)$. This yields that for well behaved noise processes (such as stationary ergodic noise), the channel capacity is achieved by a uniformly distributed input

process and satisfies the following expression (e.g., see [2, Problem 4.27]):

$$C = \lim_{n\to\infty} \frac{1}{n} I(X^n; Y^n) = \log_2 q - H(\mathcal{Z}) \qquad \text{(in bits/channel use)},$$

where $H(\mathcal{Z}) = \lim_{n\to\infty} \frac{1}{n} H(Z^n)$ is the noise entropy rate. We next describe an interesting instance of such additive noise channels with memory, when the noise process is a first-order Markov chain.

2.1.2 Binary First-Order Markov Noise Channel

The binary first-order Markov noise channel (BFMNC) is one of the simplest additive noise channels with memory. The channel alphabets are all identical given by $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \mathbb{F}_2$, and its noise process forms a discrete time first-order Markov chain. In other words, at any time $k > 1$, we have $P_{Z_k|Z^{k-1}}(z_k|z_{k-1}, \cdots, z_1) = P_{Z_k|Z_{k-1}}(z_k|z_{k-1})$. Therefore, the probability distribution for any error word $\boldsymbol{z} \in \mathbb{F}_2^n$ is given by:

$$\begin{aligned} P_{Z^n}(\boldsymbol{z}) &= P(Z_1 = z_1) \prod_{i=2}^{n} P(Z_i = z_i \mid Z_{i-1} = z_{i-1}, \cdots, Z_1 = z_1) \\ &= P(Z_1 = z_1) \prod_{i=2}^{n} P(Z_i = z_i \mid Z_{i-1} = z_{i-1}) \\ &= P(Z_1 = z_1) \prod_{i=2}^{n} P_{Z_i|Z_{i-1}}(z_i|z_{i-1}). \end{aligned}$$

Furthermore, as the noise process is stationary by assumption, we can associate with the channel the following transition matrix:

$$\boldsymbol{P} := [P_{ij}] = \begin{bmatrix} 1-\lambda & \lambda \\ 1-\gamma & \gamma \end{bmatrix}$$

where $P_{ij} := P_{Z_{t+1}|Z_t}(j|i)$ for any t and $i, j \in \mathbb{F}_2$. In other words, the $(i,j)^{th}$ entry of $\boldsymbol{P}$ represents the probability that the next noise bit will be j given the last one was i. If $\lambda, \gamma \in (0,1)$, then our Markov chain is irreducible and has a unique stationary distribution $\boldsymbol{\pi}$ which is given by:

$$\boldsymbol{\pi} = \Big[1-p, p\Big] = \Big[\tfrac{1-\gamma}{1-\gamma+\lambda}, \tfrac{\lambda}{1-\gamma+\lambda}\Big]$$

where $p = P(Z_t = 1) \in (0,1), t = 1, 2, \cdots, n$. We can also express $\boldsymbol{P}$ in terms of the noise correlation coefficient defined as:

$$\epsilon := \frac{Cov(Z_k, Z_{k-1})}{Var(Z_k)} \in (-1, 1)$$

which gives us an equivalent transition matrix of

$$\boldsymbol{P} = \begin{bmatrix} \epsilon + (1-\epsilon)(1-p) & (1-\epsilon)p \\ (1-\epsilon)(1-p) & \epsilon + (1-\epsilon)p \end{bmatrix}.$$

Note that if $\lambda = \gamma$, then $\epsilon = 0$ and the BFMNC reduces to the BSC with crossover probability (or bit error rate) $p = \lambda$. Without loss of generality, we mostly limit ourselves to instances of the BFMNC where the bit error rate $p \in (0, 1/2)$. Furthermore, we will also mostly limit ourselves to cases where $\gamma > \lambda$ or equivalently $\epsilon > 0$. This

results in the phenomenon that once an error has occurred, it is more likely to happen again, a behaviour found in real-life channels.

2.2 Channel Coding

> The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.
>
> - Claude Shannon [24]

The framework developed in Shannon's landmark paper [24], set coding theory's fundamental goal as finding codes which minimize the decoding error probability. Our work fits into this framework by seeking to define, but not construct, a set of optimal codes. However, unlike the asymptotic results presented by Shannon we will limit ourselves to finding optimal codes with a fixed and finite length.

All codes try to reduce the decoding error probability in fundamentally the same way. Before transmission, a code "intelligently" adds extra information into the original message in a process called *encoding*. This extra information is designed to be exploited by the receiver to detect, and hopefully correct, any errors introduced during transit. The reverse process performed by the receiver is called *decoding*. In general, the more extra information the encoder adds, the better the decoder performs. So it is important to only compare codes which add the same amount of extra information.

The messages sent over the channel can be of arbitrary and varying lengths. However in the real world, communication systems employ networking protocols which

send and receive data in fixed finite sized chunks, or packets. This naturally leads to the creation of codes which both operate on and generate such chunks of data called *block codes*.[2]

2.2.1 Block Codes

A chunk of data to be sent is called a *message word*. The number of distinct message words a code can protect is called the code *size* denoted by $M \in \mathbb{N}^*$. We can view a message word as a distinct k-tuple over $\mathcal{X}$, $k = \left\lceil log_{|\mathcal{X}|}(M) \right\rceil$. In general, $M \leq |\mathcal{X}^k|$, however, in this work we make the common assumption that $M = |\mathcal{X}^k|$. In this case the constant k is called the code's *dimension*. Once encoded, a message word becomes a *codeword*, with the set of all codewords denoted by $\mathcal{C}$. A codeword is a n-tuple $\boldsymbol{c} \in \mathcal{X}^n$ with $n \in \mathbb{N}^*$ being called the code's *length*. A code's encoder is a bijective map between $\mathcal{X}^k$ and $\mathcal{C}$. Under our assumptions for additive noise channels, this gives a map between $\mathbb{F}_q^k$ and $\mathcal{C} \subset \mathbb{F}_q^n$. This leads us to the following definition for a q-ary block code.

Definition 1. *An $[n, k]$-code, $\mathcal{C}$, is the image of a bijective function:*

$$f_{\mathcal{C}} : \mathbb{F}_q^k \mapsto \mathcal{C} \subset \mathbb{F}_q^n$$

where $n, k \in \mathbb{N}^*$ with $n > k$.

Definition 2 (Linear Code). *A linear $[n, k]$-code $\mathcal{C}$ is a subgroup of the additive*

[2]There exists other types of codes, such as *convolutional codes*, which can operate on arbitrary large messages. For a unified introduction to these code, see the series of papers by Forney [10], [11] [12] or the textbook [17].

group $(\mathbb{F}_q^n, +)$ *where* $+$ *denotes vector addition.*

Remark 1. *Our definition for a linear code is based on the older but more general concept of a group code given by Slepian [25]. Alternatively, a linear code can be described as a subspace of the vector space* $\mathbb{F}_q^n$*. This definition allows one to leverage linear algebra to describe a practical method of encoding and decoding these codes [20, Chapter 7]. However, this definition is more restrictive as an additive subgroup of a vector space over an arbitrary finite field is not necessarily a subspace.*

The tuple (n, k), is called the code's *parameters*. We will often use the conventional shorthand, (n, k)-code, to refer to any code of length of n and dimension k. This shorthand also provides us a nice way of comparing fixed length codes, as we do in this work.

Recall that we can only fairly compare the performance of codes which add the same amount of information. The parameter k allows us to measure this feature precisely. However, we will not group codes based solely on their k value. There is another important practical consideration.

In real-life communications systems, both fidelity and *latency* are important. The larger a code's length, the more data a system must buffer before making a transmission, which means added latency. Therefore, the value of n is also important. Hence in practice, it is of interest to only compare the performance of one code relative to others with the same parameters, i.e., to other $[n, k]$-codes.

2.2.2 Channel Decoding

Once we have encoded and sent a message through a channel, we are left with the task of decoding it.

Definition 3. *A decoder for an $[n, k]$-code $\mathcal{C}$ is a deterministic function:*

$$\mathbb{D} : \mathbb{F}_q^n \mapsto \mathcal{C} \ \cup \ \{\emptyset\}$$

where $\emptyset$ represents a decoding failure.

Remark 2. *For simplicity of analysis, the decoders in this work perform block by block decoding. In other words, it disregards all information about previous blocks when decoding the current one.*

The decoder maps every channel output word to a codeword, or declares a decoding failure. A complete decoder is one which never produces a decoding failure. On the other hand, an incomplete decoder is one where there exists $\boldsymbol{y} \in \mathbb{F}_q^n$ such that $\mathbb{D}(\boldsymbol{y}) = \emptyset$.

A decoder's goal is to recover the initial message sent. In other words, we wish to decode a received word to *the most likely sent message*. We have assumed that our channel's input, the message words, are equiprobable. Therefore all codewords are also equiprobable. Under this condition the optimal decoder, in terms of maximizing the PCD, is the maximum likelihood decoder (MLD).

Definition 4. *For an $[n, k]$-code $\mathcal{C}$, an MLD is a deterministic onto function ϕ:*

$$\begin{aligned} \phi : \ & \mathbb{F}_q^n \to \mathcal{C} \\ & \boldsymbol{y} \mapsto \hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c} \in \mathcal{C}} P_{Y^n|X^n}(\boldsymbol{y}|\boldsymbol{c}) \end{aligned} \tag{2.1}$$

where $P_{Y^n|X^n}$ is the channel's transition probability distribution.

Remark 3. *The choice for* $\arg\max$ *in (2.1) may not be unique. In other words, there may be ties. How ties are broken does not effect a code's PCD. However, Definition 4 does require that any ties be broken in a deterministic manner. For example, one could impose an arbitrary labelling on the codewords, i.e., $\mathcal{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_M\}$, and in case of ties, pick the codeword, $\boldsymbol{c}_i$, with the smallest index i.*

The above decoder definition is not amenable both to analysis and efficient implementation, particularly for large values of the dimension k. Thankfully, we can leverage properties of additive noise channels to define an equivalent but more tractable decoder.

Chapter 3

Coding for Additive Noise Channels

3.1 Generalized Syndrome Decoder

The output of an additive noise channel can always be described by $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}$ where $\boldsymbol{c}$ is the codeword originally sent and $\boldsymbol{e}$ the error word introduced by the channel. But the decoder does not have knowledge of $\boldsymbol{e}$, and must deduce it from $\boldsymbol{y}$. Given a received word $\boldsymbol{y}$, $\boldsymbol{e}$ may be any vector in $\mathbb{F}_q^n$ satisfying the property $\exists \hat{\boldsymbol{c}} \in \mathcal{C}$ such that $\hat{\boldsymbol{c}} + \boldsymbol{e} = \boldsymbol{y}$. We call the set of all possible errors for a received word its *error set*.

Definition 5 (**Error Set**). *For an $[n, k]$-code $\mathcal{C}$, the error set $\mathcal{E}_{\mathcal{C}}(\boldsymbol{y})$ for a received word $\boldsymbol{y} \in \mathbb{F}_q^n$ is defined as:*

$$\mathcal{E}_{\mathcal{C}}(\boldsymbol{y}) := \{\ \boldsymbol{e} \in \mathbb{F}_q^n \mid \boldsymbol{y} - \boldsymbol{e} \in \mathcal{C}\}.$$

Definition 6. *The set of all possible error sets $\boldsymbol{\mathcal{E}}$, for an $[n,k]$-code $\mathcal{C}$, is defined as*

$$\boldsymbol{\mathcal{E}}(\mathcal{C}) := \{\mathcal{E}_{\mathcal{C}}(\boldsymbol{y}) \mid \boldsymbol{y} \in \mathbb{F}_q^n\}.$$

Definition 7. *For an error set $\mathcal{E} \in \boldsymbol{\mathcal{E}}(\mathcal{C})$, we call a word $\boldsymbol{y} \in \mathbb{F}_q^n$ such that $\mathcal{E}_{\mathcal{C}}(\boldsymbol{y}) = \mathcal{E}$ a representative of $\mathcal{E}$.*

A code's error sets are *independent* of the channel on which the code is used. It is purely a result of the code's structure. We next use this observation to derive some universal properties.

Lemma 1. *The error sets of an $[n,k]$-code $\mathcal{C}$ cover $\mathbb{F}_q^n$.*

Proof. Let $\mathcal{C}$ be an $[n,k]$-code. For any $\boldsymbol{y} \in \mathbb{F}_q^n$, set $\boldsymbol{y}' = \boldsymbol{y} + \boldsymbol{c}$ for some $\boldsymbol{c} \in \mathcal{C}$; then we have $\boldsymbol{y}' - \boldsymbol{y} \in \mathcal{C} \implies \boldsymbol{y} \in \mathcal{E}_{\mathcal{C}}(\boldsymbol{y}')$. □

Further properties are derived from the base structure of error sets.

Definition 8 (**Translation**)**.** *The translation of a subset $\mathcal{S} \subseteq \mathbb{F}_q^n$ by a word $\boldsymbol{v} \in \mathbb{F}_q^n$ is denoted and defined as:*

$$\boldsymbol{v} + \mathcal{S} = \{\boldsymbol{v} + \boldsymbol{s} \mid \boldsymbol{s} \in \mathcal{S}\}$$

and the set $\boldsymbol{v} + \mathcal{S}$ is called a translate of $\mathcal{S}$.

Remark 4. *By definition, cosets of a linear code are each a translate of the code.*

Definition 9 (**Set Inverse**)**.** *Let X be a subset of $\mathbb{F}_q^n$, its (additive) inverse is denoted and defined as:*

$$-X := \{-\boldsymbol{x} \mid \boldsymbol{x} \in X\}.$$

Theorem 1. *Every error set for an $[n,k]$-code $\mathcal{C}$ is a translate of the set $-\mathcal{C}$.*

Proof. Let $\mathcal{E} \in \boldsymbol{\mathcal{E}}(\mathcal{C})$ with some representative $\boldsymbol{y} \in \mathbb{F}_q^n$, then we have:

$$\begin{aligned}\mathcal{E} &= \{\ \boldsymbol{e} \in \mathbb{F}_q^n \mid \boldsymbol{y} - \boldsymbol{e} \in \mathcal{C}\} && \text{(by Definition 5)}\\ &= \{\ \boldsymbol{y} - \boldsymbol{c} \mid \boldsymbol{c} \in \mathcal{C}\} = \boldsymbol{y} + \{-\boldsymbol{c} \mid \boldsymbol{c} \in \mathcal{C}\} = \boldsymbol{y} + (-\mathcal{C}).\end{aligned}$$

□

For linear codes, the (additive) inverse of the code is itself.

Lemma 2. *Let X be a additive subgroup of $\mathbb{F}_q^n$, then $-X = X$.*

Proof. Be definition $-X = \{-\boldsymbol{x} \mid \boldsymbol{x} \in X\}$. But since X is a subgroup, it is closed under inverses, i.e., for all $\boldsymbol{x} \in X$ we also have $-\boldsymbol{x} \in X$, so $-X \subseteq X$. But $|-X| = |X|$, and hence $-X = X$. □

Corollary 1. *Every error set for a linear $[n,k]$-code $\mathcal{C}$, is a translate (i.e., coset) of $\mathcal{C}$.*

Proof. Follows directly from Theorem 1 and Lemma 2. □

We can use error sets to define our generalized syndrome decoder (GSD).

Definition 10 (**Generalized Syndrome Decoder**)**.** *Given an $[n,k]$-code $\mathcal{C}$ used over an additive noise channel, a GSD is defined as a deterministic onto function Φ:*

$$\begin{aligned}\Phi :\ & \mathbb{F}_q^n \to \mathcal{C}\\ & \boldsymbol{y} \mapsto \hat{\boldsymbol{c}} = \boldsymbol{y} - \arg\max_{\boldsymbol{e} \in \mathcal{E}_{\mathcal{C}}(\boldsymbol{y})} P_{Z^n}(\boldsymbol{e}),\end{aligned} \tag{3.1}$$

where ties are broken in a deterministic manner.

Remark 5. *As the name implies, the GSD is based on the syndrome decoder [20, Section 7.5], also known as decoding using a standard array [19, Section 1.4]. However, unlike a syndrome decoder it does not use at its base a code's cosets, but its error sets. This allows the GSD to handle decoding of non-linear codes. Furthermore, it uses error set leaders instead of coset leaders. These must be chosen based on its error probability, and not the Hamming weight, as to accommodate channels with memory. For those channels, the Hamming weight of an error word is not a good approximation of its error probability.*

Another way the GSD deviates from a syndrome decoder is the lack of syndromes. Unfortunately, this makes the GSD totally impractical. Syndrome decoders are 'practical' because they can calculate a syndrome for a received word, which is in one-to-one correspondence with its coset leader, i.e., the most probable error to have occurred. In its current form, the GSD requires us to explicitly construct the error set for a received word and find the error set leader by exhaustive search. This may be impractical except for small-size codes. Despite this flaw, this decoder proves useful for analyzing the performance of block codes.

Theorem 2. *For an $[n, k]$-code $\mathcal{C}$ used over an additive noise channel, a GSD is an MLD.*

Proof. An MLD is defined by (2.1), but since the channel has additive noise, we have:

$$\arg\max_{\boldsymbol{c}\in\mathcal{C}} P_{Y^n|X^n}(\boldsymbol{y}|\boldsymbol{c}) = \arg\max_{\boldsymbol{c}\in\mathcal{C}} P_{Z^n}(\boldsymbol{y}-\boldsymbol{c}) = \boldsymbol{y} - \arg\max_{\boldsymbol{e}\in\mathcal{E}_{\mathcal{C}}(\boldsymbol{y})} P_{Z^n}(\boldsymbol{e}). \tag{3.2}$$

□

In the remainder of this work, we will assume that all our block codes are used

with a GSD. We can safely make this assumption as Theorem 2 proves that GSD and MLD are equivalent for additive noise channels. Finally, some errors within an error set are particularly important.

Definition 11 (**Error Set Leader**). *For an $[n, k]$-code $\mathcal{C}$ with GSD Φ, the error set leader for a received word $\boldsymbol{y} \in \mathbb{F}_q^n$ is defined as:*

$$\boldsymbol{e_L}(\boldsymbol{y}) := \boldsymbol{y} - \Phi(\boldsymbol{y}) = \arg \max_{\boldsymbol{e} \in \mathcal{E}_{\mathcal{C}}(\boldsymbol{y})} P_{Z^n}(\boldsymbol{e}). \tag{3.3}$$

In other words, the error set leader is the most likely error pattern (under the channel noise distribution) which could have occurred given the received word.

Remark 6. *Note that the choice of* $\arg\max$ *in Definition 11 must coincide with that of Φ. Hence ties must be broken in the same deterministic manner.*

Remark 7. *Corollary 1 shows equivalence between error sets and cosets. This equivalence has an important effect in classical coding theory. Indeed, when discussing linear codes, the terms coset and coset leader are often used interchangeably with error set and error set leader respectively.*

3.2 Error Detection and Correction

Definition 12 (**Error Detection**). *For an $[n, k]$-code $\mathcal{C}$ and error word $\boldsymbol{e} \in \mathbb{F}_q^n$, we say that we can detect $\boldsymbol{e}$ if there exists some $\boldsymbol{y} \in \mathbb{F}_q^n$ such that*

$$\boldsymbol{e} \in \mathcal{E}_{\mathcal{C}}(\boldsymbol{y});$$

i.e., $\boldsymbol{y} - \boldsymbol{e} \in \mathcal{C}$.

Remark 8. *Our definition for error correction is derived from (3.2). The action of an MLD can be described as picking the most likely error to have occurred from the error set of a received word.*

Hence the decoder can only *detect* error patterns in the error set of the received word. But detection is not always correction. Unlike error detection which is a binary event, error correction occurs on a spectrum. Our decoder correctly decodes a received word, if it outputs the codeword which was originally sent, i.e., $\Phi(\boldsymbol{y}) = \boldsymbol{c}$ where $\boldsymbol{y}$ is the channel output and $\boldsymbol{c}$ the codeword sent. Error correction is simply correct decoding conditioned on a specific error pattern.

Definition 13 (**Error Correction**). *Given an $[n, k]$-code $\mathcal{C}$, we say that an error pattern is corrected if*

$$\Phi(\boldsymbol{c} + \boldsymbol{e}) = \boldsymbol{c}$$

where $\boldsymbol{c} \in \mathcal{C}$ is the codeword sent, and $\boldsymbol{e} \in \mathbb{F}_q^n$ is error pattern which has occurred.

It is of interest to find which specific error patterns can be corrected. For a fixed error pattern $\boldsymbol{e}_f$, there are $|\mathcal{C}|$ cases to consider. Each case corresponds to a channel output of the form $\boldsymbol{y} = \boldsymbol{e}_f + \boldsymbol{c}$ for some $\boldsymbol{c} \in \mathcal{C}$. We can say our error pattern, $\boldsymbol{e}_f$, can be corrected, if there exists *at least* one codeword such that $\Phi(\boldsymbol{c} + \boldsymbol{e}_f) = \boldsymbol{c}$.

Definition 14 (**Partially Correctable**). *Given an $[n, k]$-code $\mathcal{C}$, we say that $\boldsymbol{e} \in \mathbb{F}_q^n$ is partially correctable, if*

$$\forall \boldsymbol{c} \in \mathcal{C}' \text{ we have } \Phi(\boldsymbol{c} + \boldsymbol{e}) = \boldsymbol{c},$$

for some strict non-empty subset $\mathcal{C}' \subsetneq \mathcal{C}$.

If we can correct an error pattern in all cases, i.e., $\forall \boldsymbol{c} \in \mathcal{C}$, we call this error word correctable.

Definition 15 (**Correctable**). *Given an* $[n, k]$*-code* $\mathcal{C}$*, we say that* $\boldsymbol{e} \in \mathbb{F}_q^n$ *is correctable, if*

$$\forall \boldsymbol{c} \in \mathcal{C} \text{ we have } \Phi(\boldsymbol{c} + \boldsymbol{e}) = \boldsymbol{c}.$$

Since the distinction between partially correctable and correctable errors may be foreign to the reader, we will provide an example. Definition 14 defines an error as partially correctable, if there exists *at least one* codeword that the decoder can recover if it is corrupted by this error. So it is possible that we can correct an error if it corrupts one codeword but not another. In other words, we could have the case where $\exists \boldsymbol{e} \in \mathbb{F}_q^n$ and $\boldsymbol{c} \neq \boldsymbol{c}' \in \mathcal{C}$ such

$$\Phi(\boldsymbol{c} + \boldsymbol{e}) = \boldsymbol{c} \text{ and } \Phi(\boldsymbol{c}' + \boldsymbol{e}) \neq \boldsymbol{c}'.$$

Let us examine a concrete example. Consider the following binary $[5, 2]$-code: $\mathcal{C}_W = \{\boldsymbol{c}_0, \boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3\} := \{(0\,0\,0\,0\,0), (0\,0\,0\,0\,1), (1\,0\,0\,0\,0), (1\,1\,0\,1\,1)\}$. We use $\mathcal{C}_W$ over the BFMNC($\lambda = 0.05, \gamma = 0.1$), and send the codeword $\boldsymbol{c}_2$. Unfortunately, the error $\boldsymbol{e}_0 := (0\,1\,1\,1\,0)$ occurs during transmission, and we receive the word $(1\,1\,1\,1\,0)$. A GSD (or MLD) Φ would then decode it as follows:

$$\begin{aligned}\Phi((1\,1\,1\,1\,0)) &= (1\,1\,1\,1\,0) - \arg\max_{\boldsymbol{e} \in \mathcal{E}_{\mathcal{C}_W}(\boldsymbol{y})} P_{Z^n}(\boldsymbol{e}) \\ &= (1\,1\,1\,1\,0) - (0\,0\,1\,0\,1) \qquad \text{(by Table 3.1)}\end{aligned}$$

$$= \boldsymbol{c}_3 \neq \boldsymbol{c}_2.$$

Hence we have incorrectly decoded the received word. However, if we send $\boldsymbol{c}_0$, and once again the error pattern $\boldsymbol{e}_0$ occurs; we receive $(0\,1\,1\,1\,0)$. Then we decode it as:

$$\begin{aligned}
\Phi((0\,1\,1\,1\,0)) &= (0\,1\,1\,1\,0) - \arg\max_{\boldsymbol{e}\in\mathcal{E}_{\mathcal{C}}(\boldsymbol{y})} P_{Z^n}(\boldsymbol{e}) \\
&= (0\,1\,1\,1\,0) - (0\,1\,1\,1\,0) \qquad \text{(by Table 3.1)} \\
&= \boldsymbol{c}_0.
\end{aligned}$$

So surprisingly, we are able to correct $\boldsymbol{e}_0$ if it corrupts $\boldsymbol{c}_0$ but not $\boldsymbol{c}_2$. Therefore, this error pattern is partially correctable for $\mathcal{C}_W$, but not correctable.

Table 3.1: Select error sets of $\mathcal{C}_W$ and noise probabilities

$\mathcal{E}_{\mathcal{C}_W}(\boldsymbol{y})$	$\boldsymbol{e} \in \mathcal{E}_{\mathcal{C}_W}(\boldsymbol{y})$	$P_{Z^n}(\boldsymbol{e})$ for BFMNC($\lambda = 0.05, \gamma = 0.1$)
$\mathcal{E}_{\mathcal{C}_W}((1\,1\,1\,1\,0))$	$(0\,0\,1\,0\,1)$	1.82e-3
	$(0\,1\,1\,1\,0)$	4.05e-4
	$(1\,1\,1\,1\,0)$	2.37e-6
	$(1\,1\,1\,1\,1)$	5.26e-7
$\mathcal{E}_{\mathcal{C}_W}((0\,1\,1\,1\,0))$	$(0\,1\,1\,1\,0)$	4.05e-4
	$(0\,1\,1\,1\,1)$	4.26e-5
	$(1\,0\,1\,0\,1)$	1.07e-5
	$(1\,1\,1\,1\,0)$	2.37e-6

Remark 9. *In classical coding theory, error correction is defined relative to the MD of a code. A code is t-error correcting if $d_{\min}(\mathcal{C}) \geq 2t+1$ where $d_{\min}(\cdot)$ is the minimum distance of the code [28, Section 2.1]. This is due to a fundamental result for the BSC; a code with minimum distance, d, can correct all error patterns of Hamming weight*

less than or equal to $\lceil \frac{d-1}{2} \rceil$ *[19, Theorem 1.2]. Hence, the number of errors a code can correct can be estimated using its MD. But estimating error correction solely using the MD focuses our attention on correctable errors.*

A t*-error correcting code has the property that Hamming spheres of radius* t *centred around* ***every*** *codeword are disjoint. Since these spheres have the same radius and are around every codeword, it means all errors of weight* $\leq \lceil \frac{d-1}{2} \rceil$ *can be corrected no matter which codeword is corrupted. In other words, all errors of weight* $\leq \lceil \frac{d-1}{2} \rceil$ *are correctable. However, as we shall see it is important to also consider partially correctable errors, especially for non-linear codes.*

3.3 The Decoding Set

When it comes to finding correctable and partially correctable error patterns we can focus our attention on a single set.

Definition 16 (**Decoding Set**)**.** *The decoding set of an* $[n, k]$*-code* $\mathcal{C}$ *with GSD* Φ *is the set of all distinct error set leaders:*

$$\mathcal{D}(\mathcal{C}) := \{\boldsymbol{e}_{\boldsymbol{L}}(\boldsymbol{y}) \mid \boldsymbol{y} \in \mathbb{F}_q^n\}.$$

Remark 10. *The decoding set for a code is unique as the GSD is deterministic.*

Our definition for a decoding set differs from the one in [16]. As we will build directly on work from [16], it is important to discuss why we adopted a different definition. To make our discussion clearer, we will refer to sets which satisfy Hamada's definition as H-decoding sets.

H-decoding sets are introduced in the context of establishing the existence of a syndrome decoder based on an arbitrary weight function. Of course, such a decoder exists for linear codes. However, Hamada remarks that his decoder can exist under a milder assumption on the code, i.e., the existence of an H-decoding set for the code.

Definition 17 (**H-Decoding Set** [16])**.** *Given an $[n,k]$-code $\mathcal{C}$, if there exists a subset of $\mathcal{H} \subset \mathcal{X}^n$ such that*

$$\bigcup_{\boldsymbol{c}\in\mathcal{C}} (\boldsymbol{c}+\mathcal{H}) = \mathcal{X}^n$$

and

$$(\boldsymbol{c}+\mathcal{H}) \bigcap (\boldsymbol{c}'+\mathcal{H}) = \emptyset \text{ for } \boldsymbol{c},\boldsymbol{c}' \in \mathcal{C}, \boldsymbol{c} \neq \boldsymbol{c}';$$

then $\mathcal{H}$ is called a H-decoding for $\mathcal{C}$.

The existence of this set guarantees that Hamada's syndrome decoder, which we denote by Φ_H, where $\Phi_H(\boldsymbol{y}) = \boldsymbol{y} - \boldsymbol{L}_{\mathcal{H}}(\boldsymbol{y})$, is well defined by letting $\boldsymbol{L}_{\mathcal{H}}(\boldsymbol{y}) = \boldsymbol{z}$ be the unique[1] word $\boldsymbol{z} \in \mathcal{H}$ such that $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{z}$ for some $\boldsymbol{c} \in \mathcal{C}$ [16, Section 4]. Furthermore, we have that $\Phi_H^{-1}(\boldsymbol{c}) := \{\boldsymbol{x} \in \mathcal{X}^n \mid \Phi_H(\boldsymbol{x}) = \boldsymbol{c}\} = \boldsymbol{c} + \mathcal{H}$. But H-decoding sets do not exist for all codes. In fact, their existence implies the code is either linear or the translate of one.

Lemma 3. *Given an $[n,k]$-code $\mathcal{C}$, if the code admits an H-decoding set, then we can partition $\mathbb{F}_q^n$ with translates of $\mathcal{C}$.*

Proof. Let $\mathcal{C}$ be an $[n,k]$-code admitting a H-decoding set, $\mathcal{H}$. By Definition 17 we

[1] By Definition 17, translates of $\mathcal{H}$ by codewords partition $\mathcal{X}^n$, i.e., for any $\boldsymbol{x} \in \mathcal{X}^n$ we have that $\exists!(\boldsymbol{c},\boldsymbol{h}) \in \mathcal{C} \times \mathcal{H}$ such that $\boldsymbol{x} = \boldsymbol{c} + \boldsymbol{h}$

have that

$$\bigcup_{\boldsymbol{c}\in\mathcal{C}}(\boldsymbol{c}+\mathcal{H})=\mathbb{F}_q^n \implies \forall \boldsymbol{y}\in\mathbb{F}_q^n \text{ we have } \boldsymbol{y}=\boldsymbol{c}+\boldsymbol{h} \text{ for some } \boldsymbol{c}\in\mathcal{C} \;\&\; \boldsymbol{h}\in\mathcal{H}$$
$$\implies \bigcup_{\boldsymbol{h}\in\mathcal{H}}(\boldsymbol{h}+\mathcal{C})=\mathbb{F}_q^n.$$

Therefore, the translates of the code by elements of $\mathcal{H}$ cover the space. Now to prove this cover is also a partition, we proceed by contradiction.

Suppose that the translates of $\mathcal{C}$ by elements of $\mathcal{H}$ do not form a partition. Then there exists elements $\boldsymbol{h}\neq\boldsymbol{h}'\in\mathcal{H}$ such that $(\boldsymbol{h}+\mathcal{C})\cap(\boldsymbol{h}'+\mathcal{C})$ is non-empty. Therefore,

$$\exists \boldsymbol{y}\in(\boldsymbol{h}+\mathcal{C})\cap(\boldsymbol{h}'+\mathcal{C}) \text{ such that } \boldsymbol{y}=\boldsymbol{h}+\boldsymbol{c}=\boldsymbol{h}'+\boldsymbol{c}' \text{ for some } \boldsymbol{c}\neq\boldsymbol{c}'\in\mathcal{C}.$$
$$\implies \boldsymbol{y}\in(\boldsymbol{c}+\mathcal{H}) \text{ and } \boldsymbol{y}\in(\boldsymbol{c}'+\mathcal{H})$$
$$\implies (\boldsymbol{c}+\mathcal{H})\cap(\boldsymbol{c}'+\mathcal{H})\neq\emptyset.$$

But this contradicts Definition 17. We conclude that the translates of the code by elements of $\mathcal{H}$ must form a partition. □

Theorem 3. *Given an $[n,k]$-code $\mathcal{C}$, if $\mathcal{C}$ admits an H-decoding set, $\mathcal{H}$, then $\mathcal{C}$ is linear or a translate of a linear code.*

Proof. Suppose our code admits an H-decoding set. Lemma 3 tells us that the translates of the code by elements of the H-decoding set form a partition of $\mathbb{F}_q^n$. Finally by Theorem 5 (which will be shown in Section 3.4), we conclude that $\mathcal{C}$ must be an additive subgroup of $\mathbb{F}_q^n$ or the coset of one. □

Despite these shortcoming, we still wish to use a syndrome decoder and decoding sets in our analysis as they provide a tractable expression for maximum likelihood

decoding. But we also want to include *all non-linear codes* in our analysis. Therefore, we sought alternative definitions for both, arriving at our definition for the GSD and decoding set.

Returning to our decoding set, there is a direct connection between its elements and error correction.

Theorem 4. *For an $[n, k]$-code $\mathcal{C}$, an error word $\boldsymbol{e} \in \mathbb{F}_q^n$ is partially correctable or correctable if and only if $\boldsymbol{e} \in \mathcal{D}(\mathcal{C})$.*

Proof. Take any $\boldsymbol{e} \in \mathcal{D}(\mathcal{C})$. By Definitions 11 and 16, there exists a $\boldsymbol{y} \in \mathbb{F}_q^n$ such that $\boldsymbol{e} = \boldsymbol{e_L}(\boldsymbol{y}) = \boldsymbol{y} - \Phi(\boldsymbol{y})$. From this result, we can derive that $\Phi(\boldsymbol{y}) = \boldsymbol{y} - \boldsymbol{e}$ and $\boldsymbol{y} = \boldsymbol{e} + \boldsymbol{c}$ for some $\boldsymbol{c} \in \mathcal{C}$, which implies that $\Phi(\boldsymbol{c} + \boldsymbol{e}) = \boldsymbol{c}$. Therefore, $\boldsymbol{e}$ is partially correctable or correctable.

Next suppose $\boldsymbol{e}^* \in \mathbb{F}_q^n$ is partially correctable or correctable. In either case, there exists at least one $\boldsymbol{c} \in \mathcal{C}$ such that $\Phi(\boldsymbol{c} + \boldsymbol{e}^*) = \boldsymbol{c}$. Let $\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{e}^*$, then we have that

$$
\begin{aligned}
&\Phi(\boldsymbol{y}) = \boldsymbol{c} \\
&\implies \boldsymbol{e}^* = \boldsymbol{y} - \Phi(\boldsymbol{y}) \\
&\implies \boldsymbol{e}^* = \boldsymbol{e_L}(\boldsymbol{y}) \in \mathcal{D}(\mathcal{C}) \qquad \text{(by Definitions 11 and 16)}
\end{aligned}
$$

□

We can use the above one-to-one correspondence to describe how many unique error words a code can correct.

Corollary 2. *An $[n, k]$-code $\mathcal{C}$ corrects exactly $|\mathcal{D}(\mathcal{C})|$ error words.*

Proof. Follows directly from Theorem 4 □

The cardinality of the above set can be easily bounded from above.

Lemma 4. *For an $[n, k]$-code $\mathcal{C}$, we have that $|\mathcal{D}(\mathcal{C})| \leq |\boldsymbol{\mathcal{E}}(\mathcal{C})|$.*

Proof. By Definitions 11 and 16, every error set leader $\boldsymbol{e} \in \mathcal{D}(\mathcal{C})$ originates from an error set $\mathcal{E}_{\mathcal{C}}(\boldsymbol{y}) \in \boldsymbol{\mathcal{E}}(\mathcal{C})$ for some $\boldsymbol{y} \in \mathbb{F}_q^n$. □

For linear codes, and their translates, we can precisely calculate the cardinality of the decoding set.

Corollary 3. *A linear $[n, k]$-code $\mathcal{C}$, or a translate of one, corrects exactly q^{n-k} error words.*

Proof. Since our code $\mathcal{C}$ is linear, or a translate of one, then it can be expressed as $\boldsymbol{t} + H$ for some base subgroup $H \leqslant \mathbb{F}_q^n$ and $\boldsymbol{t} \in \mathbb{F}_q^n$. By Theorem 1 the error set for a received word $\boldsymbol{y}$ is equal to

$$\begin{aligned} \mathcal{E}_{\mathcal{C}}(\boldsymbol{y}) &= \boldsymbol{y} - \mathcal{C} \\ &= \boldsymbol{y} - (\boldsymbol{t} + H) = (\boldsymbol{y} - \boldsymbol{t}) - H \\ &= (\boldsymbol{y} - \boldsymbol{t}) + H. \qquad \text{(by Lemma 2)} \end{aligned} \tag{3.4}$$

Therefore the error sets are equivalent to cosets of the base subgroup. We can determine the number of distinct cosets using Lagrange's theorem:

$$\begin{aligned} |\mathbb{F}_q^n| &= [\mathbb{F}_q^n \ : \ H] \cdot |H| \\ \text{where } [\mathbb{F}_q^n \ : \ H] &= \frac{q^n}{q^k} \qquad \text{(since } |H| = |\mathcal{C}|\text{).} \\ \text{Thus } |\boldsymbol{\mathcal{E}}(\mathcal{C})| &= q^{n-k}. \end{aligned}$$

Lemma 4 gives us that $|\mathcal{D}(\mathcal{C})| \leq |\boldsymbol{\mathcal{E}}(\mathcal{C})|$. Furthermore, the cosets of H partition $\mathbb{F}_q^n$, and so $\boldsymbol{\mathcal{E}}(\mathcal{C})$ also forms a partition. Therefore, each error set leader originates from a unique error set, i.e., $|\mathcal{D}(\mathcal{C})| = |\boldsymbol{\mathcal{E}}(\mathcal{C})|$. We combine that relation with Corollary 2 to conclude that our code corrects $|\boldsymbol{\mathcal{E}}(\mathcal{C})| = q^{n-k}$ error words. □

3.4 Error Sets and the Partitioning Property

Unfortunately Corollary 3, cannot be applied to a large class of non-linear codes. The error sets for most non-linear codes do not partition the space. Theorem 1 proves that all error sets for the code $\mathcal{C}$ are translations of $-\mathcal{C}$. In [27], Tugger proved only cosets of a subgroup can partition the space in such a manner.

Theorem 5 (**The Partitioning Property** [27])**.** *Let X be a subset of a group G. Then the left translates of X partition G if and only if X is a (left or right) coset of a subgroup of G.*

Remark 11. *The additive group $(\mathbb{F}_q^n, +)$ is commutative and hence both left and right cosets, or translates, are equivalent. Hence in this work we simply use the term translate and coset.*

Corollary 4. *Given an $[n, k]$-code $\mathcal{C}$, its error sets partition the space if and only if the code is linear or it is a translate of a linear code.*

Proof. First suppose that the code's error sets partition the space. By Theorem 1, for any error set $\mathcal{E} \in \boldsymbol{\mathcal{E}}(\mathcal{C})$ with representative $\boldsymbol{y} \in \mathbb{F}_q^n$ we have $\mathcal{E} = \boldsymbol{y} + (-\mathcal{C})$. Hence

translates of the set $-\mathcal{C}$ partition the space. Then by Theorem 5, the set $-\mathcal{C}$ is a coset of a subgroup of $\mathbb{F}_q^n$, i.e., $-\mathcal{C} = \boldsymbol{t} + H$ for some subgroup $H \leqslant \mathbb{F}_q^n$ and some choice of representative $\boldsymbol{t} \in \mathbb{F}_q^n$. This results in two cases depending on if $\boldsymbol{t} \in H$ or $\boldsymbol{t} \notin H$.

If $\boldsymbol{t} \in H$, then $-\mathcal{C} = \boldsymbol{t} + H = H$. By Lemma 2, the set inverse of the code is equal to itself, i.e, $-(-\mathcal{C}) = -\mathcal{C}$. But by the definition of set inverse we get $-(-\mathcal{C}) = \{-(-\boldsymbol{c}) \mid \boldsymbol{c} \in \mathcal{C}\} = \mathcal{C}$. Therefore, $\mathcal{C} = -\mathcal{C} = H \leqslant \mathbb{F}_q^n$, and so $\mathcal{C}$ is a linear code.

If $\boldsymbol{t} \notin H$, then $-\mathcal{C} = \boldsymbol{t} + H$ implies that for all $\boldsymbol{c} \in \mathcal{C}$ we have $-\boldsymbol{c} = \boldsymbol{t} + \boldsymbol{h}$ for some $\boldsymbol{h} \in H$. Hence,

$$\begin{aligned} &\forall \boldsymbol{c} \in \mathcal{C} \text{ we have } \boldsymbol{t} + \boldsymbol{c} = -\boldsymbol{h} \in H \\ &\implies \boldsymbol{t} + \mathcal{C} \subseteq H. \end{aligned}$$

But $|\mathcal{C}| = |H|$, and so we must have $\boldsymbol{t} + \mathcal{C} = H$. In other words, $\mathcal{C}$ is a translate of the subgroup, or linear code, H.

Now suppose that the code is linear or it is a translate of a linear code. Then $\mathcal{C} = \boldsymbol{t} + H$ for some subgroup $H \leqslant \mathbb{F}_q^n$ and choice of representative $\boldsymbol{t} \in \mathbb{F}_q^n$. An error set for this code $\mathcal{E} \in \boldsymbol{\mathcal{E}}(\mathcal{C})$ with representative $\boldsymbol{y} \in \mathbb{F}_q^n$ can be expressed as:

$$\begin{aligned} \mathcal{E} &= \boldsymbol{y} + (-\mathcal{C}) \\ &= \boldsymbol{y} - (\boldsymbol{t} + H)(\boldsymbol{y} - \boldsymbol{t}) - H \\ &= (\boldsymbol{y} - \boldsymbol{t}) + H \qquad \text{(by Lemma 2)} \end{aligned}$$

Therefore every error sets of the code is a coset of the subgroup H. Conversely, every

coset of the subgroup H, i.e., $\boldsymbol{r} + H$ with some representative $\boldsymbol{r} \in \mathbb{F}_q^n$, is equivalent to an error set of $\mathcal{C}$ by:

$$\begin{aligned} \boldsymbol{r} + H &= (\boldsymbol{r} + \boldsymbol{t}) - (\boldsymbol{t} - H) \\ &= (\boldsymbol{r} + \boldsymbol{t}) - (\boldsymbol{t} + H) && \text{(by Lemma 2)} \\ &= \mathcal{E}_{\mathcal{C}}(\boldsymbol{r} + \boldsymbol{t}) \in \boldsymbol{\mathcal{E}}(\mathcal{C}). \end{aligned}$$

Therefore the code's error sets form the same partition of the space as the cosets of H. □

We are at an impasse. For error sets, the partitioning property is too closely tied to linearity. We might be tempted to simply forge ahead and try some other method to link error sets and error correction for a general code. However, this seems unlikely to be fruitful as the inability to form a partition has dramatic consequences.

3.5 Limitation

Lemma 1 shows that a code's error sets cover the space. If they form a cover, but not a partition, then these sets must overlap. And the more they overlap, the greater the chances that the following occurs:

$$\exists \mathcal{E}_1 \neq \mathcal{E}_2 \in \boldsymbol{\mathcal{E}}(\mathcal{C}) \text{ with representatives } \boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathbb{F}_q^n \text{ s.t. } \boldsymbol{e_L}(\boldsymbol{y}_1) = \boldsymbol{e_L}(\boldsymbol{y}_2). \tag{3.5}$$

If the above is true, then there are more unique error sets than unique error sets leaders. This puts into question the usefulness of error set analysis.

This type of analysis depends on using error sets to determine either exactly or approximately other important properties of a code. For our purposes, the number of error set leaders is an important property. It is tied directly to both the number of correctable error words (Corollary 2) and the probability of correct decoding (to be seen in (5.1)). So let us then briefly examine some empirical evidence to gauge how well error sets can approximate this property.

We conduct the following experiment. For some choice of small n_0, k_0, we survey all $[n_0, k_0]$-codes and measure the distribution of the following quantity.

Definition 18 (**Error Delta**). *For an $[n, k]$-code $\mathcal{C}$, its error delta is defined as*

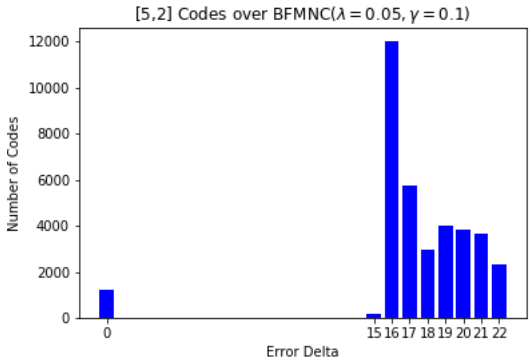
$$\Delta(\mathcal{C}) := |\boldsymbol{\mathcal{E}}(\mathcal{C})| - |\mathcal{D}(\mathcal{C})|$$

where $\Delta(\mathcal{C}) \in \{0, 1, \cdots, |\boldsymbol{\mathcal{E}}(\mathcal{C})|\}$ by Lemma 4.

This quantity serves as proxy to measure the extent that the situation in (3.5) occurs within a given code. For a linear code, $\Delta(\mathcal{C}) = 0$, indicating the number of error sets corresponds *exactly* to the number of error set leaders. This correspondence weakens as the value of $\Delta(\mathcal{C})$ increases. Specifically, if $\Delta(\mathcal{C}) \gg 0$ then there are many more error sets than error set leaders.

So as it relates to our experiment, if $\Delta(\mathcal{C}) \approx 0$ for *most* codes, then the number of error sets serves as a good approximation for the number of error set leaders. On the contrary, if many codes display a large $\Delta(\mathcal{C})$ value, then error sets are a bad approximation. For our experiment, we surveyed *all* $\binom{2^5}{2^2} = 35960$ possible binary $[5, 2]$-codes over the BFMNC with parameters $\lambda = 0.05$ and $\gamma = 0.1$. The results are summarized in Figure 3.1. While far from exhaustive, the breadth and skewness

Figure 3.1: Exhaustive survey of error delta for binary $[5, 2]$-codes over the BFMNC.



of the distribution is striking. These results show how for a general code, there is little promise of approximating the number of error set leaders using error sets. This limitation lead us to develop an alternative and more powerful analysis method.

Chapter 4

Error Indexed Decoder Regions

Error sets are based on the description of Φ given in (3.1). On the other hand, decoder regions, which we define next, are based on the action of Φ as shown in (3.3).

Definition 19 (**Error Indexed Decoder Region**). *For an $[n, k]$-code $\mathcal{C}$ with GSD Φ, the decoder region indexed by the error word $\boldsymbol{e} \in \mathbb{F}_q^n$ is defined as:*

$$\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}) := \{\boldsymbol{y} \in \mathbb{F}_q^n \mid \boldsymbol{y} - \Phi(\boldsymbol{y}) = \boldsymbol{e}\}. \tag{4.1}$$

Remark 12. *For some fixed code $\mathcal{C}$, an error word $\boldsymbol{e} \in \mathbb{F}_q^n$ indexes a non-empty decoder region if and only if $\boldsymbol{e} \in \mathcal{D}(\mathcal{C})$.*

These decoder regions are closely related to the translates of a code.

Corollary 5 (**Decoder Regions and Code Translates**). *Given an $[n, k]$-code $\mathcal{C}$, for any $\boldsymbol{e} \in \mathbb{F}_q^n$ we have $\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}) \subseteq \boldsymbol{e} + \mathcal{C}$.*

Proof. Let $\boldsymbol{e} \in \mathbb{F}_q^n$. By Definition 19, for all $\boldsymbol{y} \in \mathcal{R}_{\mathcal{C}}(\boldsymbol{e})$, we have $\boldsymbol{y} - \Phi(\boldsymbol{y}) = \boldsymbol{e}$ which

gives $\boldsymbol{y} = \boldsymbol{e} + \Phi(\boldsymbol{y}) \in \boldsymbol{e} + \mathcal{C}$. So we can restate our decoder region as

$$\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}) = \{\boldsymbol{y} \in \boldsymbol{e} + \mathcal{C} \mid \boldsymbol{y} - \Phi(\boldsymbol{y}) = \boldsymbol{e}\} \subseteq \boldsymbol{e} + \mathcal{C}.$$

□

Definition 20. *For an $[n, k]$-code $\mathcal{C}$, we denote the set of all non-empty decoder regions by $\boldsymbol{\mathcal{R}}(\mathcal{C})$.*

There is an easy way to find all non-empty decoder regions.

Lemma 5. *For an $[n, k]$-code $\mathcal{C}$, all non-empty decoding regions are indexed by some error word in the code's decoding set.*

Proof. Let $\mathcal{C}$ be any $[n, k]$-code. We define the following indexing function:

$$\begin{aligned} I : \mathcal{D}(\mathcal{C}) &\to \boldsymbol{\mathcal{R}}(\mathcal{C}) \\ \boldsymbol{e} &\mapsto \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}). \end{aligned}$$

From Remark 12, it follows that I is surjective. □

4.1 Properties of Decoder Regions

These decoder regions posses some very nice properties similar to those of cosets.

Corollary 6. *The union of all possible decoder regions is the union of all decoder*

regions centred on correctable errors, i.e.,

$$\bigcup_{\boldsymbol{e}\in\mathbb{F}_q^n} \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}) = \bigcup_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}).$$

Proof. Follows directly from Remark 12 □

These sets have similar properties to cosets.

Theorem 6 (**Cover**). *The decoder regions of a $[n,k]$-code $\mathcal{C}$ cover $\mathbb{F}_q^n$, i.e.,*

$$\mathbb{F}_q^n = \bigcup_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}). \tag{4.2}$$

Proof. For any $\boldsymbol{y} \in \mathbb{F}_q^n$, we have

$$\begin{aligned} &\boldsymbol{e}_{\boldsymbol{L}}(\boldsymbol{y}) = \boldsymbol{y} - \Phi(\boldsymbol{y}) && \text{(by Definition 11)} \\ &\implies \boldsymbol{y} \in \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}_{\boldsymbol{L}}(\boldsymbol{y})). \end{aligned}$$

Now we can build the trivial cover for a finite space.

$$\mathbb{F}_q^n = \bigcup_{\boldsymbol{y}\in\mathbb{F}_q^n} \{\boldsymbol{y}\} = \bigcup_{\boldsymbol{y}\in\mathbb{F}_q^n} \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}_{\boldsymbol{L}}(\boldsymbol{y})) = \bigcup_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}).$$

The last equality following from Corollary 6. □

Theorem 7 (**Pair-Wise Disjoint**). *For an $[n,k]$-code $\mathcal{C}$, given any pair of $\boldsymbol{e} \neq \boldsymbol{e}' \in \mathcal{D}(\mathcal{C})$ the following holds*

$$\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}) \ \cap \ \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}') = \emptyset.$$

Proof. By contradiction, suppose $\exists \boldsymbol{e} \neq \boldsymbol{e}' \in \mathcal{D}(\mathcal{C})$ but

$$\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}) \ \cap \ \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}') \neq \emptyset.$$

But $\forall \boldsymbol{y} \in \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}) \ \cap \ \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}')$ we have

$$\begin{aligned} &\boldsymbol{y} - \Phi(\boldsymbol{y}) = \boldsymbol{e} \text{ and } \boldsymbol{y} - \Phi(\boldsymbol{y}) = \boldsymbol{e}' \\ &\qquad \implies \boldsymbol{e} = \boldsymbol{e}'. \qquad\qquad\qquad\qquad \text{(since } \Phi \text{ is deterministic)} \end{aligned}$$

This is a contradiction, and so we conclude that these decoder regions must be pairwise disjoint. □

With these two results, we achieve the very useful partitioning property.

Theorem 8 (**Partition**). *The decoder regions of an $[n, k]$-code $\mathcal{C}$ partition $\mathbb{F}_q^n$, i.e.,*

$$\mathbb{F}_q^n = \bigsqcup_{\boldsymbol{e} \in \mathcal{D}(\mathcal{C})} \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}) \tag{4.3}$$

where $\bigsqcup$ denotes the union of disjoint sets.

Proof. By Theorem 6 we have that the decoder regions cover the space. And by Theorem 7 we have that these sets are pair wise disjoint. □

Corollary 7. *For any $[n, k]$-code $\mathcal{C}$ the following holds:*

$$\sum_{\boldsymbol{e} \in \mathcal{D}(\mathcal{C})} |\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| = q^n.$$

Proof. Follows directly from Theorem 8. □

Unlike the partition induced by cosets, each part need not be equal.

Theorem 9 (**Uneven Partition**). *The decoder regions of an $[n, k]$-code $\mathcal{C}$ has cardinality (or size), that is bounded as follows:*

$$\forall \boldsymbol{e} \in \mathcal{D}(\mathcal{C}),\ \ 1 \leq |\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| \leq |\mathcal{C}|.$$

Proof. We will prove the above, by showing that our decoder is injective when its domain is restricted to a fixed decoder region. Let us proceed by contradiction. Suppose that for some $\boldsymbol{e} \in \mathcal{D}(\mathcal{C})$ there exists distinct $\boldsymbol{y}, \boldsymbol{y}' \in \mathcal{R}_{\mathcal{C}}(\boldsymbol{e})$ such that

$$\begin{aligned}
&\Phi(\boldsymbol{y}) = \Phi(\boldsymbol{y}') \\
&\implies \boldsymbol{y} - \Phi(\boldsymbol{y}) = \boldsymbol{y}' - \Phi(\boldsymbol{y}') \qquad (\text{since } \boldsymbol{y} - \Phi(\boldsymbol{y}) = \boldsymbol{e} \text{ and } \boldsymbol{y}' - \Phi(\boldsymbol{y}') = \boldsymbol{e}) \\
&\implies \boldsymbol{y} = \boldsymbol{y}'.
\end{aligned}$$

This is a contradiction. Therefore Φ is an injective function into $\mathcal{C}$ over the domain $\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})$, and so

$$|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| \leq |\mathcal{C}|.$$

Furthermore by Remark 12, we have that all decoder regions for errors in $\mathcal{D}(\mathcal{C})$ are non-empty, and so $\forall \boldsymbol{e} \in \mathcal{D}(\mathcal{C})$ we have $|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| \geq 1$. □

For linear codes and their translates, all these decoder regions are of the same size.

Lemma 6. *For an $[n, k]$-code $\mathcal{C}$, if $|\mathcal{D}(\mathcal{C})| = q^{n-k}$, then*

$$\forall \boldsymbol{e} \in \mathcal{D}(\mathcal{C}) \;\; |\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| = |\mathcal{C}| = q^k.$$

Proof. Follows from Theorem 9 and Corollary 7. □

Theorem 10. *For a linear $[n, k]$-code $\mathcal{C}$, all its non-empty decoder regions have cardinality $|\mathcal{C}|$, i.e.,*

$$|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| = |\mathcal{C}| \quad \forall \boldsymbol{e} \in \mathcal{D}(\mathcal{C}).$$

Proof. Let $\mathcal{C}$ be a linear $[n, k]$-code, or a translation of one. By Corollary 3, the code corrects q^{n-k} error words, i.e., $|\mathcal{D}(\mathcal{C})| = q^{n-k}$ (Corollary 2). By Lemma 6, this forces $|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| = |\mathcal{C}|$ for all $\boldsymbol{e} \in \mathcal{D}(\mathcal{C})$. □

4.2 Relation to Error Correction

What makes decoder regions incredibly useful is their tight coupling with error correction.

Theorem 11. *For an $[n, k]$-code $\mathcal{C}$, an error word $\boldsymbol{e} \in \mathcal{D}(\mathcal{C})$ is correctable if and only if*

$$|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| = |\mathcal{C}|.$$

Proof. From Corollary 5, we know that for any $\boldsymbol{e} \in \mathcal{D}(\mathcal{C})$, we have $\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}) \subseteq \boldsymbol{e} + \mathcal{C}$. And so if $|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| \neq |\boldsymbol{e} + \mathcal{C}| = |\mathcal{C}|$ then $|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| < |\mathcal{C}|$.

Let $\boldsymbol{e} \in \mathcal{D}(\mathcal{C})$. Suppose $|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| < |\mathcal{C}|$, then there exists a codeword $\boldsymbol{c} \in \mathcal{C}$ such that $(\boldsymbol{e}+\boldsymbol{c}) \in (\boldsymbol{e}+\mathcal{C}) \setminus \mathcal{R}_{\mathcal{C}}(\boldsymbol{e})$. Since $(\boldsymbol{e}+\boldsymbol{c}) \notin \mathcal{R}_{\mathcal{C}}(\boldsymbol{e})$ we have:

$$\begin{aligned}(\boldsymbol{e}+\boldsymbol{c}) - \Phi(\boldsymbol{e}+\boldsymbol{c}) &\neq \boldsymbol{e} \\ \iff \Phi(\boldsymbol{e}+\boldsymbol{c}) &\neq \boldsymbol{c}.\end{aligned}$$

Therefore, $\boldsymbol{e}$ is not correctable. Now suppose that $\boldsymbol{e}$ is not correctable, then there exists a codeword such that

$$\begin{aligned}&\Phi(\boldsymbol{c}+\boldsymbol{e}) \neq \boldsymbol{c} \\ \iff &(\boldsymbol{e}+\boldsymbol{c}) - \Phi(\boldsymbol{e}+\boldsymbol{c}) \neq \boldsymbol{e} \\ \iff &(\boldsymbol{c}+\boldsymbol{e}) \notin \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}) \text{ but } (\boldsymbol{c}+\boldsymbol{e}) \in \boldsymbol{e}+\mathcal{C} \\ \implies &|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| < |\boldsymbol{e}+\mathcal{C}| = |\mathcal{C}|.\end{aligned}$$

□

Now we have a strong condition for when an error is correctable. On the other hand, if an error is partially correctable, we can also determine for which codewords we are able to correct the error.

Corollary 8. *A $[n,k]$-code $\mathcal{C}$ can correct an error word $\boldsymbol{e} \in \mathbb{F}_q^n$ if and only if $\boldsymbol{e}$ corrupts a codeword from $(-\boldsymbol{e}+\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})) \subseteq \mathcal{C}$.*

Proof. Let $\boldsymbol{e} \in \mathbb{F}_q^n$. Suppose we have some $\boldsymbol{c} \in \mathcal{C}$ such that $\boldsymbol{c} \notin (-\boldsymbol{e}+\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}))$, then

$$\begin{aligned}(\boldsymbol{c}+\boldsymbol{e}) \notin \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}) \iff (\boldsymbol{c}+\boldsymbol{e}) - \Phi(\boldsymbol{c}+\boldsymbol{e}) &\neq \boldsymbol{e} \\ \Phi(\boldsymbol{c}+\boldsymbol{e}) &\neq \boldsymbol{c}.\end{aligned}$$

Therefore $\boldsymbol{e}$ cannot be corrected if it corrupts $\boldsymbol{c}$. Conversely, suppose that we have $\boldsymbol{c} \in (-\boldsymbol{e} + \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}))$ then we have

$$\begin{aligned}(\boldsymbol{c}+\boldsymbol{e}) \in \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}) &\iff (\boldsymbol{e}+\boldsymbol{c}) - \Phi(\boldsymbol{c}+\boldsymbol{e}) = \boldsymbol{e} \\ &\iff \Phi(\boldsymbol{c}+\boldsymbol{e}) = \boldsymbol{c}.\end{aligned}$$

□

We can now use decoding regions other properties to establish bounds on error correction in block codes.

4.3 Bounds on Error Correction

Lemma 7. *For an $[n,k]$-code $\mathcal{C}$, we have*

$$|\mathcal{D}(\mathcal{C})| = |\boldsymbol{\mathcal{R}}(\mathcal{C})|.$$

Proof. We will prove this by showing that the function I is bijective, where I is defined as

$$\begin{aligned}I :&\mathcal{D}(\mathcal{C}) \to \boldsymbol{\mathcal{R}}(\mathcal{C}) \\ &\boldsymbol{e} \mapsto \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}).\end{aligned}$$

In Lemma 5, we proved that this function is surjective, its injectivity follows directly from Theorem 7, □

Corollary 9. *An $[n, k]$-code $\mathcal{C}$ corrects exactly $|\boldsymbol{\mathcal{R}}(\mathcal{C})|$ error words.*

Proof. Follows directly from Lemma 7 and Corollary 2. □

More concretely, the number of correctable errors is bounded as follows.

Corollary 10 (**Bound on Error Correction**)**.** *An $[n, k]$-code $\mathcal{C}$ corrects between q^{n-k} and q^n unique error words inclusively.*

Proof. Corollary 9 proves that $\mathcal{C}$ corrects exactly $|\mathcal{D}(\mathcal{C})|$ error words. We will prove our claim by bounding that set's cardinality using two relations. The first is given by Corollary 7:

$$\sum_{\boldsymbol{e} \in \mathcal{D}(\mathcal{C})} |\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| = q^n. \tag{4.4}$$

Furthermore, Theorem 9 bounds each summand by:

$$1 \leq |\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| \leq |\mathcal{C}| \quad \forall \boldsymbol{e} \in \mathcal{D}(\mathcal{C}). \tag{4.5}$$

Noting that $|\mathcal{C}| = q^k$, Equations (4.4) and (4.5) yield that

$$q^{n-k} \leq |\mathcal{D}(\mathcal{C})| \leq q^n \tag{4.6}$$

where the minimum and maximum are each reached when:

$$|\mathcal{D}(\mathcal{C})| = \begin{cases} q^{n-k} & \text{if } |\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| = |\mathcal{C}| \ \ \forall \boldsymbol{e} \in \mathcal{D} \\ q^n & \text{if } |\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| = 1 \ \ \forall \boldsymbol{e} \in \mathcal{D}. \end{cases}$$

□

A more explicit bound can be found for correctable errors.

Corollary 11 (**Upper Bound on Correctable Errors**). *An $[n, k]$-code $\mathcal{C}$ has at most q^{n-k} correctable errors.*

Proof. Theorem 11 proves that an error is completely correctable if and only if $|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| = |\mathcal{C}|$. But by Corollary 7, the total cardinality of all decoding regions is:

$$\sum_{\boldsymbol{e} \in \mathcal{D}(\mathcal{C})} |\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| = q^n.$$

Therefore, we can have at most $q^n/|\mathcal{C}| = q^{n-k}$ decoding regions of that size. □

Remark 13. *From Theorem 11, we see that to maximize the number of correctable errors, we must maximize the size of our decoding regions. But doing so, according to Corollary 10, will reduce the total number of error words we can correct.*

It turns out that linear codes do not have any partially correctable errors.

Theorem 12 (**Linear Codes and Correctable Errors**). *For an $[n, k]$-code $\mathcal{C}$, all error words $\boldsymbol{e} \in \mathcal{D}(\mathcal{C})$ are correctable if and only if the code is linear or a translate of one.*

Proof. Suppose a code $\mathcal{C}$ has no partially correctable errors in its decoding set. Theorem 11 shows that for all $\boldsymbol{e} \in \mathcal{E}(\mathcal{C})$ we have $|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| = |\mathcal{C}|$. Furthermore, decoder regions are subsets of error sets (Corollary 5), and so if they achieve maximal cardinality it must be that $\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}) = \boldsymbol{e} + \mathcal{C}$. So for this code, decoder regions are equivalent to error sets. However Theorem 8, tell us that decoder regions form a partition. Then

this code's error sets must also form a partition. But this only occurs in linear codes or their translations (Corollary 4).

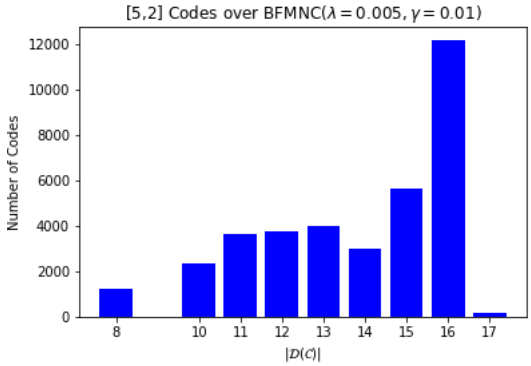
Now suppose if $\mathcal{C}$ is linear, or a translation of one. By Theorem 10, we have that $|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})| = |\mathcal{C}|$ for all $\boldsymbol{e} \in \mathcal{D}(\mathcal{C})$. Lastly, Theorem 11 shows this restriction makes all these error words correctable. □

4.4 Numerical Results

Corollary 10 reflects badly on linear codes, for combined with Corollary 3, they show that *all* linear codes can only correct the least amount of error words. And so we seek some empirical evidence to see how other non-linear codes fare. The result is shown in the following experiment.

For some choice of small n_0, k_0, we survey all $[n_0, k_0]$-codes and measure the distribution of $|\mathcal{D}(\mathcal{C})|$, which is equivalent to the number of correctable errors. Figure 4.1 summarizes the results for a survey of all $\binom{2^5}{2^2} = 35960$ binary $[5, 2]$ codes over the BFMNC with parameters $\lambda = 0.005$ and $\gamma = 0.01$. From these results we see that most codes correct a far greater number of errors than linear codes, which can correct eight error words in this example. But as we noted in Remark 13, there is a cost to correcting more than the minimum. The more error words you can correct, the less often you can correct each one. In other words, a code which can correct many errors, will have most of these errors be partially correctable.

Figure 4.1: Exhaustive survey of decoding set size for binary $[5, 2]$-codes over the BFMNC.



Chapter 5

Probability of Correct Decoding

As stated above, our goal is to find codes with maximal PCD. We will denote a code's PCD by $P_C(\mathcal{C})$. This leads us to our central theorem.

Theorem 13. *The PCD for an $[n, k]$-code $\mathcal{C}$ used over an additive noise channel with maximum likelihood decoder Φ satisfies:*

$$P_C(\mathcal{C}) = \sum_{\boldsymbol{e} \in \mathcal{D}(\mathcal{C})} \frac{|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})|}{|\mathcal{C}|} P(Z^n = \boldsymbol{e}). \tag{5.1}$$

Proof.

$$\begin{aligned} P_C(\mathcal{C}) &= \sum_{\boldsymbol{c} \in \mathcal{C}} \sum_{\boldsymbol{y}:\Phi(\boldsymbol{y})=\boldsymbol{c}} \frac{1}{|\mathcal{C}|} P(Y^n = \boldsymbol{y} | X^n = \boldsymbol{c}) \\ &= \sum_{\boldsymbol{y} \in \mathbb{F}_q^n} \frac{1}{|\mathcal{C}|} P(Y^n = \boldsymbol{y} | X^n = \Phi(\boldsymbol{y})) \\ &= \sum_{\boldsymbol{y} \in \mathbb{F}_q^n} \frac{1}{|\mathcal{C}|} P(Z^n = \boldsymbol{y} - \Phi(\boldsymbol{y}) | X^n = \Phi(\boldsymbol{y})) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} \sum_{\boldsymbol{y}\in\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})} \frac{1}{|\mathcal{C}|} P(Z^n = \boldsymbol{y} - \Phi(\boldsymbol{y}) | X^n = \Phi(\boldsymbol{y})) && \text{(by Theorem 8)} \\
&= \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} \sum_{\boldsymbol{y}\in\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})} \frac{1}{|\mathcal{C}|} P(Z^n = \boldsymbol{e} | X^n = \Phi(\boldsymbol{y})) && \text{(by Definition 19)} \\
&= \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} \frac{|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})|}{|\mathcal{C}|} P(Z^n = \boldsymbol{e}).
\end{aligned}
$$

□

Remark 14. *If we set* $w_{\boldsymbol{e}} = {}^{|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})|}/_{|\mathcal{C}|}$*, (5.1) becomes*

$$
P_C(\mathcal{C}) = \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} w_{\boldsymbol{e}} P(Z^n = \boldsymbol{e}) \text{ where } 0 < w_{\boldsymbol{e}} \leq 1 \text{ and } \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} w_{\boldsymbol{e}} = \frac{q^n}{|\mathcal{C}|}. \tag{5.2}
$$

The restrictions on $w_{\boldsymbol{e}}$ *is a direct result of Theorem 9 and Corollary 7 respectively. This is our preferred form of expressing a code's PCD.*

For our purposes, we will label a code as optimal if it achieves the highest possible PCD within its class.

Definition 21 (**Optimal Code**)**.** *An* $[n, k]$*-code* $\mathcal{C}$ *is said to be optimal if for any other* $[n, k]$*-code* $\mathcal{C}'$*, we have*

$$
P_C(\mathcal{C}) \geq P_C(\mathcal{C}').
$$

An important concept in coding theory is that of code equivalence.

5.1 Equivalence

Definition 22 (**Equivalent Codes** [19])**.** *We say two $[n,k]$-codes $\mathcal{C}$ and $\mathcal{C}'$ are equivalent if there exist n permutations $\pi_1 \cdots \pi_n$ of the q elements and a permutation σ of the n coordinate positions such that*

$$\text{if } (c_1, \cdots, c_n) \in \mathcal{C} \text{ then } \sigma(\pi_1(c_1), \cdots, \pi_n(c_n)) \in \mathcal{C}'.$$

The definition of equivalence is meant to preserves important properties of the code; as there may be very good engineering reasons for preferring one code over another equivalent one [19]. One important property of codes to preserve is the PCD. For the BSC, two equivalent binary linear codes have the same PCD [26]. For channels with memory, this does not hold in general. However, we can derive a similar guarantee not for equivalent linear codes, but for linear codes and their translates.

Lemma 8. *Given an $[n,k]$ code $\mathcal{C}$, the decoding set of $\mathcal{C}$ and any translate of $\mathcal{C}$ are equal, i.e.,*

$$\mathcal{D}(\mathcal{C}) = \mathcal{D}(\boldsymbol{t} + \mathcal{C})$$

for any vector $\boldsymbol{t} \in \mathbb{F}_q^n$.

Proof. Let $\mathcal{C}$ be an $[n,k]$-code. For any vector $\boldsymbol{t} \in \mathbb{F}_q^n$, we can convert an error set of $\mathcal{C}$ into an error set of $\boldsymbol{t} + \mathcal{C}$, i.e.,

$$\begin{aligned} \mathcal{E}_{\mathcal{C}}(\boldsymbol{y}) &= \boldsymbol{y} - \mathcal{C} \\ &= \boldsymbol{y} + \boldsymbol{t} - \boldsymbol{t} - \mathcal{C} = (\boldsymbol{y} + \boldsymbol{t}) - (\boldsymbol{t} + \mathcal{C}) \end{aligned}$$

$$= \mathcal{E}_{\boldsymbol{t}+\mathcal{C}}(\boldsymbol{y}+\boldsymbol{t}). \tag{5.3}$$

With this result, we can show that their decoding sets are equal.

$$\begin{aligned}
\mathcal{D}(\mathcal{C}) &= \left\{ \boldsymbol{e}_{\boldsymbol{L}}(\boldsymbol{y}) \;\middle|\; \boldsymbol{y} \in \mathbb{F}_q^n \right\} \\
&= \left\{ \arg \max_{\boldsymbol{e} \in \mathcal{E}_{\mathcal{C}}(\boldsymbol{y})} P_{Z^n}(\boldsymbol{e}) \;\middle|\; \boldsymbol{y} \in \mathbb{F}_q^n \right\} && \text{(by Definition 11)} \\
&= \left\{ \arg \max_{\boldsymbol{e} \in \mathcal{E}_{\boldsymbol{t}+\mathcal{C}}(\boldsymbol{y}+\boldsymbol{t})} P_{Z^n}(\boldsymbol{e}) \;\middle|\; \boldsymbol{y} \in \mathbb{F}_q^n \right\} && \text{(by (5.3))} \\
&= \left\{ \arg \max_{\boldsymbol{e} \in \mathcal{E}_{\boldsymbol{t}+\mathcal{C}}(\boldsymbol{z})} P_{Z^n}(\boldsymbol{e}) \;\middle|\; \boldsymbol{z} \in \boldsymbol{t} + \mathbb{F}_q^n \right\} \\
&= \left\{ \arg \max_{\boldsymbol{e} \in \mathcal{E}_{\boldsymbol{t}+\mathcal{C}}(\boldsymbol{z})} P_{Z^n}(\boldsymbol{e}) \;\middle|\; \boldsymbol{z} \in \mathbb{F}_q^n \right\} && \text{(since } \boldsymbol{t} \in \mathbb{F}_q^n \text{ and } \mathbb{F}_q^n \text{ is a group)} \\
&= \mathcal{D}(\boldsymbol{t}+\mathcal{C}).
\end{aligned}$$

□

Remark 15. *Lemma 8 shows that translates of a code can correct the exact same error patterns as the original code. However in general, the size of the error indexed decoding regions for the code and its translate may change. In other words, a correctable error for a code might be only partially correctable for its translate or vice versa. This does not occur for linear codes and their translates, as all corrected error patterns are always correctable (Theorem 12).*

Theorem 14. *The PCD of a linear $[n,k]$-code $\mathcal{C}$ is equal to any of its translates, i.e.,*

$$P_C(\mathcal{C}) = P_C(\boldsymbol{t}+\mathcal{C})$$

for any vector $\boldsymbol{t} \in \mathbb{F}_q^n$.

Proof. Let $\mathcal{C}$ be a linear $[n,k]$-code, and $\boldsymbol{t}$ some vector in $\mathbb{F}_q^n$. The PCD of $\mathcal{C}$ can be expressed using (5.2) as:

$$\begin{aligned} P_C(\mathcal{C}) &= \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} w_{\boldsymbol{e}} P(Z^n = \boldsymbol{e}) \\ &= \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} P(Z^n = \boldsymbol{e}) && \text{(by Theorem 10)} \\ &= \sum_{\boldsymbol{e}\in\mathcal{D}(\boldsymbol{t}+\mathcal{C})} P(Z^n = \boldsymbol{e}) && \text{(by Lemma 8)} \\ &= P_C(\boldsymbol{t}+\mathcal{C}). \end{aligned}$$

The last equality follows from the fact that for all $\boldsymbol{e} \in \mathcal{D}(\boldsymbol{t}+\mathcal{C})$ we have $w_{\boldsymbol{e}} = 1$ by Theorem 10. □

5.2 A Class of Optimal Codes

In [16, Section 5], Hamada presented some sufficient conditions for linear codes to be optimal. These conditions are based on the concept of an ideal decoding set.

Definition 23 (**Ideal Decoding Set** [16])**.** *For an additive noise channel and parameters $(n,k) \in \mathbb{N}^* \times \mathbb{N}^*$ where $n > k$, a subset $\mathcal{D}^* \subset \mathbb{F}_q^n$ is called an ideal decoding set if the following two conditions hold:*

Condition 1. *$|\mathcal{D}^*| = q^{n-k}$;*

Condition 2. *$\forall(\boldsymbol{e},\boldsymbol{e}') \in \mathcal{D}^* \times (\mathbb{F}_q^n \setminus \mathcal{D}^*)$, we have $P(Z^n = \boldsymbol{e}) \geq P(Z^n = \boldsymbol{e}')$.*

Definition 24. *An $[n,k]$-code $\mathcal{C}$ has an ideal decoding set if its decoding set $\mathcal{D}(\mathcal{C})$ satisfies Definition 23.*

Note that Condition 1 for ideal decoding sets has an important implication, see Lemma 6. Before, proving the optimality of codes with ideal decoding sets we provide some examples of such codes. First, we give an example for the BSC.

Definition 25 (**Perfect Code** [28])**.** *If a t-error correcting binary $[n,k]$-code $\mathcal{C}$ has the property*

$$\bigcup_{\boldsymbol{c}\in\mathcal{C}} \mathcal{S}_t(\boldsymbol{c}) = \mathbb{F}_q^n \text{ where } \mathcal{S}_t(\boldsymbol{c}) = \{\boldsymbol{x} \in \mathbb{F}_q^n \mid d_H(\boldsymbol{x},\boldsymbol{c}) \leq t\}, \tag{5.4}$$

then $\mathcal{C}$ is called perfect.

Lemma 9. *A t-error correcting binary perfect $[n,k]$-code $\mathcal{C}$ over the BSC corrects an error word $\boldsymbol{e} \in \mathbb{F}_2^n$ if and only if $w_H(\boldsymbol{e}) \leq t$, where $w_H(\cdot)$ is the Hamming weight.*

Proof. Follows from Definition 25 and [19, Theorem 1.2]. □

Theorem 15 (**Hamming Bound** [19])**.** *A t-error correcting binary perfect code of length n containing M codewords must satisfy*

$$M\Big[\sum_{i=0}^{t}\binom{n}{i}\Big] = 2^n$$

Theorem 16. *A t-error correcting binary perfect $[n,k]$-code $\mathcal{C}$ has an ideal decoding set over the BSC(p).*

Proof. Without loss of generality, we assume that the BSC's cross over probability $p < 1/2$.

Let $\mathcal{C}$ be a t-error correcting binary perfect $[n, k]$-code. By Lemma 9 and Theorem 4, we have that its decoding set is equal to:

$$\mathcal{D}(\mathcal{C}) = \{\boldsymbol{e} \in \mathbb{F}_2^n \mid w_H(\boldsymbol{e}) \leq t\}. \tag{5.5}$$

Hence, we can determine the size of the decoding set by counting all possible vectors of length n with Hamming weight $i \leq t$, i.e.,

$$|\mathcal{D}(\mathcal{C})| = \sum_{i=0}^{t} \binom{n}{i} \leq 2^{n-k}. \qquad \text{(by Theorem 15)}$$

But by Corollary 10, we have $|\mathcal{D}(\mathcal{C})| \geq 2^{n-k}$. Therefore, $|\mathcal{D}(\mathcal{C})| = 2^{n-k}$, which fulfills Condition 1 for an ideal decoding set.

Next take any pair of vectors $(\boldsymbol{e}, \boldsymbol{e}') \in \mathcal{D}(\mathcal{C}) \times (\mathbb{F}_2^n \setminus \mathcal{D}(\mathcal{C}))$. By (5.5), we know that $w_H(\boldsymbol{e}) \leq t$ and $w_H(\boldsymbol{e}') > t$, and so

$$P_{Z^n}(\boldsymbol{e}) > P_{Z^n}(\boldsymbol{e}') \qquad \text{(by (A.1))}.$$

Therefore $\mathcal{D}(\mathcal{C})$ also satisfies Condition 2 for an ideal decoding set. □

Furthermore, binary perfect codes with an MD of three also have an ideal decoding set for certain parameters of the BFMNC [16, Theorem 7.1]. However, no matter the channel, all codes with an ideal decoding set have a very particular structure.

Lemma 10. *If an $[n, k]$-code $\mathcal{C}$ has $|\mathcal{D}(\mathcal{C})| = q^{n-k}$, then $\mathcal{C}$ is either linear or a translate of a linear code.*

Proof. Let $\mathcal{C}$ be some $[n, k]$-code with $|\mathcal{D}(\mathcal{C})| = q^{n-k}$. By Lemma 6 and Theorem 11 , all errors in $\mathcal{D}(\mathcal{C})$ are correctable. Theorem 12 shows that a code only has correctable

errors if and only if it is linear, or a translate of one. □

Theorem 17. *If an $[n,k]$-code $\mathcal{C}$ has an ideal decoding set, then the code is either linear or a translate of a linear code.*

Proof. If a code possess an ideal decoding set, then $|\mathcal{D}(\mathcal{C})| = q^{n-k}$. Then by Lemma 10 the code must be linear or a translate of a linear code. □

We now prove that any code with an ideal decoding set is optimal.

Theorem 18. *If an $[n,k]$-code $\mathcal{C}$ has an ideal decoding set, then it is optimal.*

Proof. Let $\mathcal{C}$ be an $[n,k]$-code with an ideal decoding set. $\mathcal{C}$ is optimal if for any other $[n,k]$ code $\mathcal{C}'$ we have $P_C(\mathcal{C}) \geq P_C(\mathcal{C}')$ which is equivalent to:

$$\begin{aligned}
&\iff \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} w_{\boldsymbol{e}} P_{Z^n}(\boldsymbol{e}) \geq \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C}')} w'_{\boldsymbol{e}} P_{Z^n}(\boldsymbol{e}) \\
&\iff \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} P_{Z^n}(\boldsymbol{e}) \geq \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C}')} w'_{\boldsymbol{e}} P_{Z^n}(\boldsymbol{e}) \qquad \text{(by Lemma 6)} \\
&\iff \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} (1-w'_{\boldsymbol{e}}) P_{Z^n}(\boldsymbol{e}) \geq \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C}')\setminus\mathcal{D}(\mathcal{C})} w'_{\boldsymbol{e}} P_{Z^n}(\boldsymbol{e}). \qquad (5.6)
\end{aligned}$$

We will prove that (5.6) holds by constructing the inequality using the properties of an ideal decoding set and the restrictions given in (5.2). This construction needs to be broken into two cases depending on if $\mathcal{D}(\mathcal{C}) = \mathcal{D}(\mathcal{C}')$ or $\mathcal{D}(\mathcal{C}) \neq \mathcal{D}(\mathcal{C}')$. In the former case, (5.6) becomes:

$$\sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} (1-w'_{\boldsymbol{e}}) P_{Z^n}(\boldsymbol{e}) \geq 0$$

which is trivially true, since $\forall \boldsymbol{e}$ we have $0 \leq w'_{\boldsymbol{e}}, P_{Z^n}(\boldsymbol{e}) \leq 1$. So now we proceed with the more difficult case.

Suppose $\mathcal{D}(\mathcal{C}) \neq \mathcal{D}(\mathcal{C}')$. We know that both codes have the same dimension, and so the following holds:

$$\begin{aligned}
\sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} w_{\boldsymbol{e}} = \frac{q^n}{|\mathcal{C}|} = \frac{q^n}{|\mathcal{C}'|} = \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C}')} w'_{\boldsymbol{e}} \qquad & \text{(by (5.2))} \\
\implies \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} 1 - \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} w'_{\boldsymbol{e}} = \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C}')\setminus\mathcal{D}(\mathcal{C})} w'_{\boldsymbol{e}} \qquad & \text{(by Lemma 6)} \\
\implies \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} (1 - w'_{\boldsymbol{e}}) = \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C}')\setminus\mathcal{D}(\mathcal{C})} w'_{\boldsymbol{e}}. \qquad & (5.7)
\end{aligned}$$

We then pick two vectors $\boldsymbol{z}_0$, and $\boldsymbol{z}'_0$ such that

$$\boldsymbol{z}_0 := \arg\min_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} P_{Z^n}(\boldsymbol{e}) \;\text{ and }\; \boldsymbol{z}'_0 := \arg\max_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C}')\setminus\mathcal{D}(\mathcal{C})} P_{Z^n}(\boldsymbol{e}).$$

Since $\mathcal{D}(\mathcal{C})$ is an ideal decoding set, we have that $P_{Z^n}(\boldsymbol{z}_0) \geq P_{Z^n}(\boldsymbol{z}'_0)$. This fact allow us to transform (5.7) into an inequality by multiplying the right and left side by $P_{Z^n}(\boldsymbol{z}_0)$ and $P_{Z^n}(\boldsymbol{z}'_0)$ respectively, giving:

$$\sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} (1 - w'_{\boldsymbol{e}}) P_{Z^n}(\boldsymbol{z}_0) \geq \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C}')\setminus\mathcal{D}(\mathcal{C})} w'_{\boldsymbol{e}} P_{Z^n}(\boldsymbol{z}'_0). \tag{5.8}$$

By construction, we have chosen $\boldsymbol{z}_0$ so that $P_{Z^n}(\boldsymbol{z}_0) \leq P_{Z^n}(\boldsymbol{e})\ \forall \boldsymbol{e} \in \mathcal{D}(\mathcal{C})$, and so we have:

$$\sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} (1 - w'_{\boldsymbol{e}}) P_{Z^n}(\boldsymbol{e}) \geq \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C})} (1 - w'_{\boldsymbol{e}}) P_{Z^n}(\boldsymbol{z}_0). \tag{5.9}$$

Similarly, we have chosen $\boldsymbol{z}'_0$ so that $P_{Z^n}(\boldsymbol{z}'_0) \geq P_{Z^n}(\boldsymbol{e})\ \forall \boldsymbol{e} \in \mathcal{D}(\mathcal{C}') \setminus \mathcal{D}(\mathcal{C})$ and so we

have:

$$\sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C}')\setminus\mathcal{D}(\mathcal{C})} w'_{\boldsymbol{e}} P_{Z^n}(\boldsymbol{z}'_0) \geq \sum_{\boldsymbol{e}\in\mathcal{D}(\mathcal{C}')\setminus\mathcal{D}(\mathcal{C})} w'_{\boldsymbol{e}} P_{Z^n}(\boldsymbol{z}'_0). \tag{5.10}$$

Then (5.6) follows by combining (5.9), (5.10), and (5.8). □

5.3 Numerical Results

Theorem 18 provides us with a class of optimal linear and non-linear codes. A natural question is: does this class describe most optimal codes? We attempt to address this question with the following experiment.

For some choice of small n_0, k_0, we survey all optimal[1] $[n_0, k_0]$-codes grouping the optimal codes by the size of the code's decoding set. We group codes by the size of their decoding set so we can easily determine if they do not have an ideal decoding set. By Definition 23, an ideal decoding set must have q^{n-k} elements. So if a code's decoding set has more elements, it cannot have an ideal decoding set.

Specifically, we will examine all possible binary $[4, k]$-codes, i.e. $k \in \{1, 2, 3\}$,[2] over both the BFMNC and BSC. These parameters are quite small, and this is due to the curse of dimensionality. For a choice of code parameters (n, k), the number of possible binary $[n, k]$-codes is $\binom{2^n}{2^k}$. Therefore, there is a total of $\sum_{i=1}^{3}\binom{2^4}{2^i} = 14,810$ possible binary $[4, k]$-codes, but if we increase the code length by just one, there are $\sum_{i=1}^{4}\binom{2^5}{2^i} = 611,635,146$ possible binary $[5, k]$-codes. Furthermore, our examples use

[1] Since the code parameters are small, we are able to calculate a code's exact PCD using (5.1). Therefore our experiment does not suffer from the approximation errors for the PCD as are common in simulations.

only a single choice of parameters for each channel. Hence, surveying the behaviour of optimal codes even for small parameters is not trivial and out of scope of this work. Therefore, we will not, and warn against, extrapolating from the handful of examples we present. As with the other examples in this work, they are meant solely as counterexamples to guide our general analytical work.

Figures 5.1 and 5.2 summarize the results for a survey of all possible binary $[4, k]$-codes over the BFMNC with parameters $\lambda = 0.05$ and $\gamma = 0.1$, and the BSC with cross over probability $p = \frac{\lambda}{1-\gamma+\lambda} \approx 0.05$ respectively.[3] In these examples, there are two important observations.

First, our grouping of optimal codes by decoding set size allows us to distinguish between two important classes of codes. All linear codes and their translates form the group with $|\mathcal{D}(\mathcal{C})| = q^{n-k}$ (Lemma 10), and all other non-linear codes[4] have some $|\mathcal{D}(\mathcal{C})| > q^{n-k}$. This grouping also allows us to distinguish which optimal codes *cannot* have ideal decoding sets, i.e., those with $|\mathcal{D}(\mathcal{C})| > q^{n-k}$ (Theorem 17). Unfortunately, we cannot use the size of a code's decoding set alone to conclude that it has an ideal decoding set. The size of the decoding set is only one of the two conditions needed for it to be ideal (Definition 23).

Secondly, each grouping of codes is labeled with a percentage indicating the proportion of optimal codes which belong to that group. For example, in Figure 5.1 97.1% of all optimal binary $[4, 3]$-codes are non-linear codes with $|\mathcal{D}(\mathcal{C})| = 4$, which cannot have ideal decoding sets.

In both examples, we observe an interesting trend. There are only a few optimal

[2]By Definition 1, all block codes in this work have dimension k satisfying $n > k \geq 1$.

[3]We choose the crossover probability so that the BSC and the BFMNC have the same bit error rate, i.e., $P(Z_i = 1)$.

[4]Those non-linear codes which are not a translate of a linear code.

Figure 5.1: Exhaustive survey of optimal binary $[4, k]$-codes over the BFMNC.

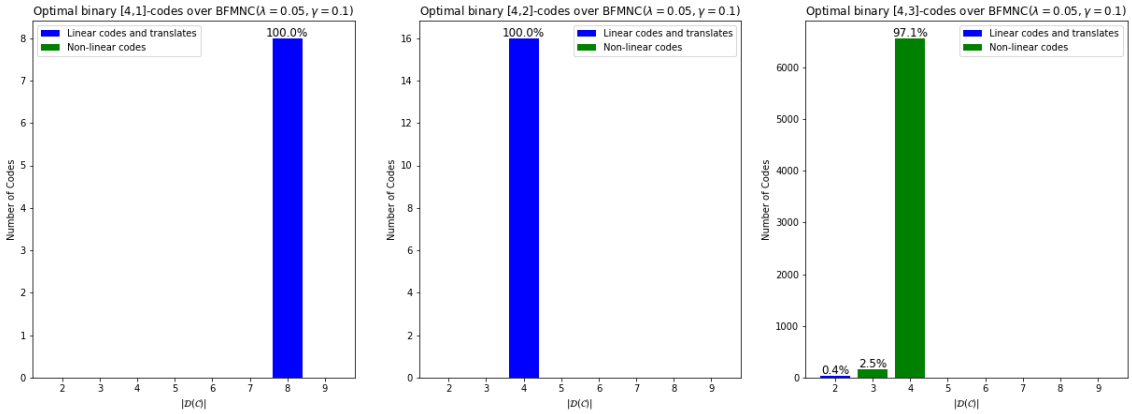
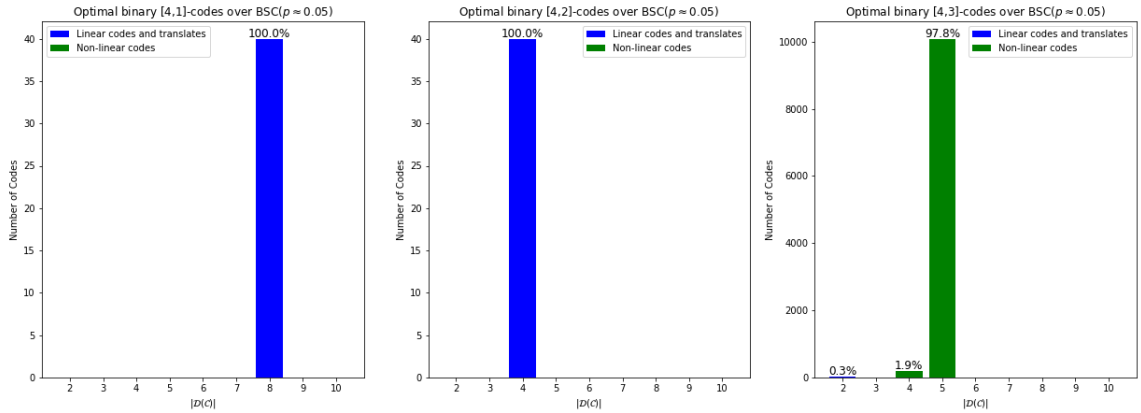


Figure 5.2: Exhaustive survey of optimal binary $[4, k]$-codes over the BSC.



binary $[4, 1]$ and $[4, 2]$-codes, and they are exclusively a translate of a linear code. This is a small number even relative to the total number of codes with those parameters, i.e., 120 binary $[4, 1]$-codes and 1820 binary $[4, 2]$-codes.

However for the higher dimensions, the situation radically shifts in favour of non-linear codes. In Figure 5.1, more than 6500 of 12870 binary $[4, 3]$-codes are optimal, with 97.1% of them being non-linear. A similar situation occurs over the BSC . The

results in Figure 5.2 confirm the common knowledge that classically optimal[5] codes are often non-linear [20, Page 259]. This also highlights the importance of our earlier warning; we should be extremely cautious when extrapolating from these examples. As we have just seen, even for the simple case of $[4, k]$-codes the nature of optimal codes can quickly and radically shift even for a fixed choice of channel parameters.

However, there is still value in examining these examples further, particularly in regards to codes with ideal decoding sets. Tables 5.1 and 5.2 gives a handful of optimal $[4, 3]$-codes with varying decoding set sizes. What is interesting is that all these codes' decoding sets satisfy Condition 2 for an ideal decoding set. In other words, the decoding sets for codes in Tables 5.1 and 5.2 consist of some combination of the most likely channel errors. Hence, removing or modifying Condition 1 may yield a larger class of optimal codes.

Table 5.1: Select optimal binary $[4, 3]$-codes for BSC($p \approx 0.05$)

$\mathcal{C}$	$\lvert\mathcal{D}(\mathcal{C})\rvert$	Ideal Decoding Set
$\{(0\,0\,0\,0), (0\,0\,0\,1), (0\,0\,1\,0), (0\,0\,1\,1),$ $(0\,1\,0\,0), (0\,1\,0\,1), (0\,1\,1\,0), (0\,1\,1\,1)\}$	2	True
$\{(0\,1\,1\,0), (0\,1\,1\,1), (1\,0\,0\,0), (1\,0\,0\,1),$ $(1\,1\,0\,0), (1\,1\,0\,1), (1\,1\,1\,0), (1\,1\,1\,1)\}$	4	False
$\{(0\,0\,0\,1), (0\,1\,1\,0), (0\,1\,1\,1), (1\,0\,0\,1),$ $(1\,0\,1\,0), (1\,1\,0\,0), (1\,1\,0\,1), (1\,1\,1\,0)\}$	5	False

[5]Classically, an optimal code is one with the highest rate (k/n) for a given MD [19, Page 58].

Table 5.2: Select optimal binary $[4, 3]$-codes for BFMNC($\lambda = 0.05, \gamma = 0.1$)

$\mathcal{C}$	$\lvert\mathcal{D}(\mathcal{C})\rvert$	Ideal Decoding Set
$\{(0\,0\,0\,0), (0\,0\,0\,1), (0\,0\,1\,0), (0\,0\,1\,1),$ $(1\,0\,0\,0), (1\,0\,0\,1), (1\,0\,1\,0), (1\,0\,1\,1)\}$	2	True
$\{(0\,1\,0\,0), (0\,1\,0\,1), (0\,1\,1\,0), (0\,1\,1\,1),$ $(1\,0\,1\,0), (1\,0\,1\,1), (1\,1\,1\,0), (1\,1\,1\,1)\}$	3	False
$\{(0\,1\,0\,0), (0\,1\,0\,1), (0\,1\,1\,0), (0\,1\,1\,1),$ $(1\,0\,1\,1), (1\,1\,0\,0), (1\,1\,1\,0), (1\,1\,1\,1)\}$	4	False

5.4 Discussion

5.4.1 Relaxing Optimality Conditions

Section 5.3 showed how removing Condition 1 of an ideal decoding set may yield a larger class of optimal codes. We saw some examples in Tables 5.1 and 5.2 where codes do not satisfy this condition, yet are optimal. This hints that this condition is restrictive. In fact, Theorem 17 shows how Condition 1 implies a code is a coset of a linear code. Another perspective is that this condition forces a codes to have only correctable errors (Lemma 6 and Theorem 11). So there is value in examining an optimal code which has partially correctable errors.

From the experiment for Figure 5.1, we found an optimal code which satisfies Condition 2, but not Condition 1. We denote our selected code as $\mathcal{C}_s$ and it is defined by the following codebook:

$$\mathcal{C}_s := \{(0\,1\,0\,0), (0\,1\,0\,1), (0\,1\,1\,0), (0\,1\,1\,1), (1\,0\,1\,1), (1\,1\,0\,0), (1\,1\,1\,0), (1\,1\,1\,1)\}.$$

With regards to the structure of the code itself, the size of its decoding set implies

it is not a coset of a linear code (Corollary 3). As for the decoding set, we know it is not ideal as a consequence of Theorem 17, since $|\mathcal{D}(\mathcal{C}_s)| = 4 > 2^{4-3}$. However, it does satisfy Condition 2 for an ideal decoding set. To make our discussion clearer, we label the errors in the decoding set. Our decoding set becomes $\mathcal{D}(\mathcal{C}_s) = \{\boldsymbol{e}_0, \boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$ where:

$$\boldsymbol{e}_0 = (\,0\,0\,1\,0\,),\ \boldsymbol{e}_1 = (\,0\,0\,0\,0\,),\ \boldsymbol{e}_2 = (\,0\,0\,0\,1\,),\ \boldsymbol{e}_3 = (\,0\,1\,0\,0\,).$$

We measure how completely each of these error words are corrected using the following metric.

Definition 26 (**Completeness Ratio**). *For an $[n, k]$-code $\mathcal{C}$, the completeness ratio for an error word $\boldsymbol{e} \in \mathcal{D}(\mathcal{C})$ is defined as:*

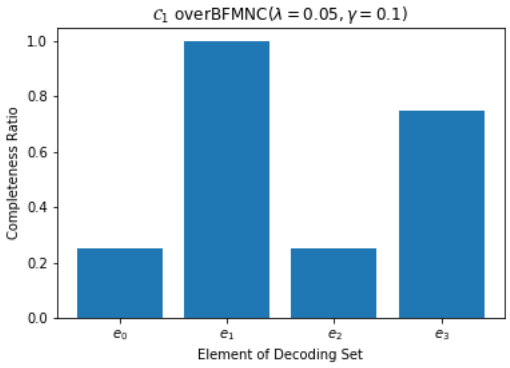
$$\Theta(\boldsymbol{e}) := \frac{|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})|}{|\mathcal{C}|}$$

where $0 < \Theta(\boldsymbol{e}) \leq 1$ by Theorem 9.

This metric measures how completely a code can correct a given error. Given a code and some error $\boldsymbol{e} \in \mathcal{D}(\mathcal{C})$, if $\Theta(\boldsymbol{e}) \approx 0$ then we have a partially correctable error, which can be corrected only in a few cases, i.e, when the channel output is $\boldsymbol{y} \in (-\boldsymbol{e} + \mathcal{R}_{\mathcal{C}}(\boldsymbol{e}))$ (Corollary 8). As $\Theta(\boldsymbol{e}) \to 1$, we can correct $\boldsymbol{e}$ for a growing number of cases, and when $\Theta(\boldsymbol{e}) = 1$ our error is correctable.

Figure 5.3 shows the distribution of $\Theta(\cdot)$ over $\mathcal{D}(\mathcal{C}_s)$. We see that *only* the all zero error word is correctable, all the rest are partially correctable. In fact, most errors are correctable in only 25% of cases. The only exception is $\boldsymbol{e}_3$ which can be corrected in 65% of possible cases. Yet $\mathcal{C}_s$ is an optimal code. Hence, a code with

Figure 5.3: Completness Ratio for $\mathcal{C}_s$



mostly partially correctable errors can be optimal. This is a much milder condition than a code possessing an ideal decoding set, which requires all errors be correctable. This observation leads us to the following conjecture.

Conjecture 5.1. *If an $[n, k]$-code $\mathcal{C}$ decoding set $\mathcal{D}(\mathcal{C})$ satisfies:*

$$\forall(\boldsymbol{e}, \boldsymbol{e}') \in \mathcal{D}(\mathcal{C}) \times (\mathbb{F}_q^n \setminus \mathcal{D}(\mathcal{C})), \text{ we have } P(Z^n = \boldsymbol{e}) \geq P(Z^n = \boldsymbol{e}'),$$

then $\mathcal{C}$ is optimal.

5.4.2 A New Perspective

Our new expression for a code's PCD, (5.1), makes it easier to see the connection between maximizing the PCD and (linear) integer programming.

Definition 27 (**Linear Integer Program** [33])**.** *A linear integer program consists of the following constrained maximization:*

$$\begin{aligned} &\max \boldsymbol{s}\boldsymbol{x} \\ &A\boldsymbol{x} \leq \boldsymbol{b} \\ &\boldsymbol{x} \geq 0 \text{ and } \boldsymbol{x} \in \mathbb{Z}^n \end{aligned}$$

where A is a m by n matrix, $\boldsymbol{s}$ an n-dimensional row vector, $\boldsymbol{b}$ an m-dimensional column vector, and $\boldsymbol{x}$ an n-dimensional column vector of variables or unknowns.

Remark 16. *Integer programs only restrict the vector $\boldsymbol{x}$ to be integer valued, not the other program parameters, $\boldsymbol{s}$, $\boldsymbol{b}$, and A, which can have elements belonging to another set, e.g., $\mathbb{R}$.*

Many real-life problems can be expressed in such a form. These problems appear in variety of domains such as production planning, telecommunications, and molecular biology [33, Chapter 1]. In general, integer programming is an NP-complete problem, but there exists polynomial time solutions for many special cases [33, Chapter 6]. We herein show how maximizing the PCD can be framed as a (linear) integer program. We leave analyzing the complexity of this formulation to future works. We achieve our reframing using (5.1) mixed with a few key observations.

Definition 28. *Let (n, k) be a fixed pair of code parameters, and $\mathbb{F}_q^n = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_L\}$ be an arbitrary labeling of the vector space, where $L = q^n$. We define the the function $g_{(n,k)}$ as follows:*

$$g_{(n,k)} : \boldsymbol{\mathcal{C}}(n, k) \to \mathbb{N}^L$$

$$\mathcal{C} \mapsto \boldsymbol{a} := \Big[|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}_1)|, |\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}_2)|, \cdots, |\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}_L)|\Big],$$

where $\boldsymbol{\mathcal{C}}(n,k) := \{\mathcal{C} \subset \mathbb{F}_q^n \mid |\mathcal{C}| = q^k\}$ *is the set of all* $[n,k]$*-codes.*

Corollary 12. *Let* (n,k) *denote a fixed pair of code parameters,* $L = q^n$*, and* $\mathbb{F}_q^n = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_L\}$ *be an arbitrary labeling of the vector space. The maximal PCD for all* $[n,k]$*-codes over an additive noise channel with noise distribution* P_{Z^n} *is given by:*

$$\begin{aligned}\max_{\forall \mathcal{C} \in \boldsymbol{\mathcal{C}}(n,k)} P_C(\mathcal{C}) &= \max_{\boldsymbol{a} \in g_{(n,k)}(\boldsymbol{\mathcal{C}}(n,k))} \left[\sum_{i=1}^{L} \frac{a_i}{q^k} P(Z^n = \boldsymbol{e}_i)\right] \\ &= \max_{\boldsymbol{a} \in g_{(n,k)}(\boldsymbol{\mathcal{C}}(n,k))} \frac{1}{q^k} \left[\boldsymbol{p}^T \cdot \boldsymbol{a}\right], \end{aligned} \tag{5.11}$$

where $\boldsymbol{p} \in [0,1]^L$ *is the vector whose* i^{th} *element* $p_i = P_{Z^n}(\boldsymbol{e}_i)$*, and* $\cdot$ *represents the dot product of vectors.*

Proof. Theorem 13 gives the PCD of an $[n,k]$-code $\mathcal{C}$ as:

$$\begin{aligned} P_C(\mathcal{C}) &= \sum_{\boldsymbol{e} \in \mathcal{D}(\mathcal{C})} \frac{|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})|}{|\mathcal{C}|} P(Z^n = \boldsymbol{e}) \\ &= \sum_{\boldsymbol{e} \in \mathbb{F}_q^n} \frac{|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})|}{|\mathcal{C}|} P(Z^n = \boldsymbol{e}). \end{aligned} \tag{5.12}$$

The last equality follows directly from Remark 12. Next, we make bookkeeping easier using our arbitrary labelling of $\mathbb{F}_q^n$ rewriting (5.12) as:

$$P_C(\mathcal{C}) = \sum_{\boldsymbol{e} \in \mathbb{F}_q^n} \frac{|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e})|}{|\mathcal{C}|} P(Z^n = \boldsymbol{e}) = \sum_{i=1}^{L} \frac{|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}_i)|}{|\mathcal{C}|} P(Z^n = \boldsymbol{e}_i). \tag{5.13}$$

We then leverage the function $g_{(n,k)}$ to associate an integer vector with $\mathcal{C}$, allowing us

to rewrite (5.13) as:

$$P_C(\mathcal{C}) = \sum_{i=1}^{L} \frac{|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}_i)|}{|\mathcal{C}|} P(Z^n = \boldsymbol{e}_i) = \sum_{i=1}^{L} \frac{a_i}{|\mathcal{C}|} P(Z^n = \boldsymbol{e}_i) \tag{5.14}$$

where a_i is the i^{th} element of the vector $\boldsymbol{a} = g_{(n,k)}(\mathcal{C})$. Now we can see that maximizing the PCD is almost equivalent to finding a vector of non-negative integers, $\boldsymbol{a} \in \mathbb{N} \cup \{0\}$, maximizing (5.14). However, this vector must be induced by some $[n,k]$-code $\mathcal{C}$, i.e., $\boldsymbol{a} \in g_{(n,k)}(\boldsymbol{\mathcal{C}}(n,k))$. In other words, the maximum of $P_C(\cdot)$ over all $[n,k]$-codes is given by:

$$\max_{\forall \mathcal{C} \in \boldsymbol{\mathcal{C}}(n,k)} P_C(\mathcal{C}) = \max_{\boldsymbol{a} \in g_{(n,k)}(\boldsymbol{\mathcal{C}}(n,k))} \left[\sum_{i=1}^{L} \frac{a_i}{q^k} P(Z^n = \boldsymbol{e}_i) \right]. \tag{5.15}$$

Finally, if we let $\boldsymbol{p}$ be the vector whose i^{th} element $p_i = P_{Z^n}(\boldsymbol{e}_i)$, then we can replace the summation in (5.15) with a vector dot product, i.e,

$$\begin{aligned} \max_{\boldsymbol{a} \in g_{(n,k)}(\boldsymbol{\mathcal{C}}(n,k))} \left[\sum_{i=1}^{L} \frac{a_i}{q^k} P(Z^n = \boldsymbol{e}_i) \right]. &= \max_{\boldsymbol{a} \in g_{(n,k)}(\boldsymbol{\mathcal{C}}(n,k))} \left[\sum_{i=1}^{L} \frac{1}{q^k} a_i p_i \right] \\ &= \max_{\boldsymbol{a} \in g_{(n,k)}(\boldsymbol{\mathcal{C}}(n,k))} \frac{1}{q^k} \left[\boldsymbol{p}^T \cdot \boldsymbol{a} \right]. \end{aligned}$$

□

Corollary 12 shows how maximizing the PCD could be stated as a linear integer program. However, it is unclear what our are our constraints. Finding constraints for (5.12) is equivalent to finding conditions on $\boldsymbol{x} \in \mathbb{N}^L$ such that $\boldsymbol{x} \in g_{(n,k)}(\boldsymbol{\mathcal{C}}(n,k))$. This appears to be an extremely difficult combinatorial problem. So instead we can optimize on a superset of $g_{(n,k)}(\boldsymbol{\mathcal{C}}(n,k))$, whose membership is easier to define. We accomplish this task using the following lemmas.

Lemma 11. *Let $\mathcal{C}$ be an $[n,k]$-code. The vector $\boldsymbol{a} := g_{(n,k)}(\mathcal{C})$ satisfies the following three conditions:*

Condition 1. $\sum_{i=1}^{L} a_i = q^n$,

Condition 2. $\forall i, 0 \leq a_i \leq q^k$,

Condition 3. $q^{n-k} \leq |\{a_i \mid a_i \neq 0\}| \leq q^n$.

Proof. By Definition 28, the i^{th} element of the vector $\boldsymbol{a}$ is equal to $|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}_i)|$, and therefore we have:

$$\begin{aligned}\sum_{i=1}^{L} a_i &= \sum_{i=1}^{L} |\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}_i)| \\ &= q^n \qquad\qquad \text{(by Corollary 7)},\end{aligned}$$

which results in Condition 1. Condition 2 follows from Theorem 9, i.e, $0 \leq a_i = |\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}_i)| \leq q^k$, noting that if $\boldsymbol{e}_i \notin \mathcal{D}(\mathcal{C})$ then $|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}_i)| = 0$. Finally, from Remark 12 we have that $|\mathcal{R}_{\mathcal{C}}(\boldsymbol{e}_i)| \neq 0$ if and only if $\boldsymbol{e}_i \in \mathcal{D}(\mathcal{C})$, and so:

$$|\{a_i \mid a_i \neq 0\}| = |\mathcal{D}(\mathcal{C})|.$$

Therefore, Condition 3 follows directly from the above and (4.6). □

For linear codes, these conditions simplify as follows:

Lemma 12. *Let $\mathcal{C}$ be a linear $[n,k]$-code. The vector $\boldsymbol{a} := g_{(n,k)}(\mathcal{C})$ satisfies the following three conditions:*

- $\sum_{i=1}^{L} a_i = q^n$,

- $\forall i, a_i \in \{0, q^k\}$,
- $|\{a_i \mid a_i \neq 0\}| = q^{n-k}$.

Proof. The first condition is given by Lemma 11. The second follows directly from Theorem 10. The final condition is a result of Corollary 2 and Corollary 3. □

Now we can construct a superset using the constraints in Lemma 11 or Lemma 12. However, since we are maximizing over a superset we cannot determine the exact maximal PCD, but only an upper bound. If we use Lemma 11, for example, we get the following inequality:

$$\max_{\forall \mathcal{C} \in \boldsymbol{\mathcal{C}}(n,k)} P_C(\mathcal{C}) = \max_{\boldsymbol{a} \in g_{(n,k)}(\boldsymbol{\mathcal{C}}(n,k))} \frac{1}{q^k} \left[\boldsymbol{p}^T \cdot \boldsymbol{a}\right] \leq \max_{\boldsymbol{a} \in G} \frac{1}{q^k} \left[\boldsymbol{p}^T \cdot \boldsymbol{a}\right]$$

where $G := \{\boldsymbol{a} \in \mathbb{N}^L \mid \text{Condition 1,2, and 3}\} \supset g_{(n,k)}(\boldsymbol{\mathcal{C}}(n,k))$. In other words, we are maximizing over the set of vectors which satisfy the necessary, but not sufficient, conditions (Lemma 11) for a vector to be in $g_{(n,k)}(\boldsymbol{\mathcal{C}}(n,k))$.

Chapter 6

Conclusion

We have constructed a class of codes optimal based on work from [16]. This result relied heavily on our novel treatment of error indexed decoder regions, and partially correctable errors. These concepts allow for a unified analysis of both linear and non-linear codes over additive noise channels. They also lead to an interesting new expression for a code's PCD. Surprisingly, it allowed us to characterize these optimal codes as being either linear or the translate of a linear code.

There are various avenues for future work. One avenue would be to find milder optimality conditions for non-linear codes. As we have seen, Condition 1 for an ideal decoding set seems too strong a condition for optimality, forcing codes to be linear or a translate. Hopefully, our techniques and particularly our expression for a code's PCD might prove useful in relaxing these conditions, and finding a larger class of optimal codes.

Perhaps the most fruitful avenue of research would be the analysis of (5.11), which reformulates the PCD maximization problem as an integer program. Future work

could leverage the tools of integer programming to establish new effective bounds on the maximal PCD. Furthermore, the analysis of maximizing solutions of (5.11) in terms of decoder regions may give insight into the structure of optimal codes.

Lastly, an important avenue of research would be verifying the robustness of our results. Many common assumptions used for modelling channels are only approximately true in practice. For example, the inputs to a channel are not always equiprobable. And it is not clear how well (5.1) approximates a code's PCD in such a case. Any analytical, or empirical, results on the accuracy of (5.1) under alternate assumptions would be of great value.

Appendix A

Minimum Hamming Distance and Probability of Correct Decoding

In Section 1.2, we stated that there exists cases where a code with the smallest MD is optimal, and one with the largest is suboptimal. This fact is true even for BSC. In this section, we present one such a case. We assume the reader is familiar with the material on additive noise channels covered in Chapter 2, and our definition of optimality (Definition 21). For the sake of completeness we briefly describe the BSC.

The BSC(p) is a memoryless binary additive noise channel with crossover probability p. This parameter describes the noise probability which is $P(Z_i = 1) = p$ and $P(Z_i = 0) = 1 - p$ for all i. Hence the noise probability for any error word $\boldsymbol{z} \in \mathbb{F}_2^n$ is given by:

$$P_{Z^n}(\boldsymbol{z}) = p^{w_H(\boldsymbol{z})}(1-p)^{n-w_H(\boldsymbol{z})} \tag{A.1}$$

where $w_H(\cdot)$ is the Hamming weight of a binary vector. Without loss of generality we can assume that $p < 1/2$. If $p > 1/2$, we can apply a permutation to the channel input, i.e., π such that $\pi(1) = 0$ and $\pi(0) = 1$, and obtain an equivalent channel with $p < 1/2$. We consider the BSC($p = 0.4$) and the following two linear $[4, 2]$-codes:

$$\mathcal{C}_L = \{(0\,0\,0\,0), (0\,0\,1\,1), (0\,1\,0\,1), (0\,1\,1\,0)\} \text{ with } d_{\min}(\mathcal{C}_L) = 2$$

and

$$\mathcal{C}_S = \{(0\,0\,0\,0), (0\,0\,0\,1), (1\,1\,1\,0), (1\,1\,1\,1)\} \text{ with } d_{\min}(\mathcal{C}_S) = 1$$

where $d_{\min}(\cdot)$ is the minimum Hamming distance of a code. For any $[4, 2]$-code $\mathcal{C}$ we have $d_{\min}(\mathcal{C}) \in \{1, 2\}$.[1] Therefore, $\mathcal{C}_L$ has the *largest* MD while $\mathcal{C}_S$ has *smallest* MD among all $[4, 2]$-codes. Hence, from a classical coding theory perspective we expect $\mathcal{C}_L$ to have a greater PCD than $\mathcal{C}_S$. We can use (5.1) to calculate each code's PCD, which gives us:

$$P_C(\mathcal{C}_L) \approx 0.36 < P_C(\mathcal{C}_S) \approx 0.38.^{2}$$

This is a surprising result. Even for the BSC, if we select a code with the *maximal* MD for a given size and dimension, we can still end up with a code which is not only suboptimal, but also has inferior performance to one with a smaller MD. But perhaps more surprising is that $\mathcal{C}_S$ is optimal.[3] In conclusion, we cannot rely on the minimum Hamming distance to find or describe optimal codes in general, even for the BSC.

---

[1] This was confirmed through brute force, calculating $d_{\min}$ for all $\binom{2^4}{2^2} = 1820$ possible $[4, 2]$-codes.

[2] We have confirmed this strict inequality holds for a range of channel parameters, i.e., $P_C(\mathcal{C}_L) < P_C(\mathcal{C}_S)$ over the BSC(p) with $p \in \{0.4, 0.2, 0.1, 10^{-2}, 10^{-3}\}$.

[3] This was confirmed once again through brute force, by calculating $P_C(\cdot)$ using (5.1) for all possible $[4, 2]$-codes over the $BSC(p = 0.4)$.

Bibliography

[1] H. Al-Lawati and F. Alajaji. On decoding binary perfect and quasi-perfect codes over Markov noise channels. *IEEE Transactions on Communications*, 57(4):873–878, 2009.

[2] F. Alajaji and P.-N. Chen. *An Introduction to Single-User Information Theory*. Springer, 2018.

[3] F. Alajaji and T. Fuja. A communication channel modeled on contagion. *IEEE Transactions on Information Theory*, 40(6):2035 –2041, Nov 1994.

[4] G. Azar and F. Alajaji. On the equivalence between maximum likelihood and minimum distance decoding for binary contagion and queue-based channels with memory. *IEEE Transactions on Communications*, 63(1):1–10, 2015.

[5] H. Bauer, B. Ganter, and F. Hergert. Algebraic techniques for nonlinear codes. *Combinatorica*, 3(1):21–33, 1983.

[6] P.-N Chen, H. Lin, and S. M. Moser. Weak flip codes and applications to optimal code design on the binary erasure channel. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 160–167, 2012.

[7] P.-N. Chen, H. Lin, and S. M. Moser. Optimal ultrasmall block-codes for binary discrete memoryless channels. *IEEE Transactions on Information Theory*, 59(11):7346–7378, 2013.

[8] R. L. Dobrushin and M. S. Pinsker. Memory increases transmission capacity. *Problemy Peredachi Informatsii*, 5(1):94–95, 1969.

[9] E. O. Elliott. Estimates of error rates for codes on burst-noise channels. *Bell Syst. Tech. J*, 42(9):1977–1997, 1963.

[10] G. D. Forney. Convolutional codes I: Algebraic structure. *IEEE Transactions on Information Theory*, 16(6):720–738, 1970.

[11] G. D. Forney. Convolutional codes II: Maximum-likelihood decoding. *Information and Control*, 25(3):222 – 266, 1974.

[12] G. D. Forney. Convolutional codes III: Sequential decoding. *Information and Control*, 25(3):267 – 297, 1974.

[13] R. G. Gallager. *Information Theory and Reliable Communication*, volume 2. Springer, 1968.

[14] E. N. Gilbert. Capacity of a burst-noise channel. *Bell Syst. Tech. J*, 39(9):1253–1265, 1960.

[15] M Hamada. A sufficient condition for a code to achieve the minimum decoding error probability–generalization of perfect and quasi-perfect codes. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 83(10):1870–1877, 2000.

[16] M. Hamada. Near-optimality of subcodes of hamming codes on the two-state Markovian additive channel. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 84(10):2383–2388, 2001.

[17] R. Johannesson and K. S. Zigangirov. *Fundamentals of Convolutional Coding*. John Wiley and Sons, 2015.

[18] K. Lindström. All nearly perfect codes are known. *Information and Control*, 35(1):40–47, 1977.

[19] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. Number pt. 2 in Mathematical Library. North-Holland Publishing Company, 1977.

[20] R. J. McEliece. *The Theory of Information and Coding: A Mathematical Framework for Communication*. Addison-Wesley Publishing Company, 1977. From Encyclopedia of Mathematics and its Applications.

[21] F. P. Preparata. A class of optimum nonlinear double-error-correcting codes. *Information and Control*, 13(4):378–400, 1968.

[22] J. Proakis. *Digital Communications*. McGraw-Hill series in electrical and computer engineering. McGraw-Hill, 4th ed edition, 2000.

[23] P. Sadeghi, R. A. Kennedy, P. B. Rapajic, and R. Shams. Finite-state Markov modeling of fading channels - a survey of principles and applications. *IEEE Signal Processing Magazine*, 25(5):57–80, 2008.

[24] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[25] D. Slepian. A class of binary signaling alphabets. *The Bell System Technical Journal*, 35(1):203–234, 1956.

[26] D. Slepian. Some further theory of group codes. *The Bell System Technical Journal*, 39(5):1219–1252, 1960.

[27] R. T. Tugger. Subgroups and the partitioning property. *Mathematics Magazine*, 66(2):114–115, 1993.

[28] J. H. Van Lint. *Coding Theory*, volume 201. Springer, 1971.

[29] H. C. A. van Tilborg. *Uniformly Packed Codes*. PhD thesis, Technische Hogeschool, 1976.

[30] G. Vazquez-Vilar, A. Guillén i Fàbregas, and S. Verdú. The error probability of generalized perfect codes via the meta-converse. *IEEE Transactions on Information Theory*, 65(9):5705–5717, 2019.

[31] M. Villanueva, F. Zeng, and J. Pujol. Efficient representation of binary nonlinear codes: constructions and minimum distance computation. *Designs, Codes and Cryptography*, 76(1):3–21, 2015.

[32] H. S. Wang and N. Moayeri. Finite-state Markov channel-a useful model for radio communication channels. *IEEE Transactions on Vehicular Technology*, 44(1):163–171, 1995.

[33] L. A. Wolsey. *Integer Programming*, volume 52. John Wiley and Sons, 1998.

[34] L. Zhong, F. Alajaji, and G. Takahara. A binary communication channel with memory based on a finite queue. *IEEE Transactions on Information Theory*, 53(8):2815–2840, 2007.

[35] L. Zhong, F. Alajaji, and G. Takahara. A model for correlated Rician fading channels based on a finite queue. *IEEE Transactions on Vehicular Technology*, 57(1):79–89, 2008.