國立交通大學

電信工程研究所

博士論文

二元無記憶通道的最佳極小區塊碼設計

Optimal Ultra-Small Block-Codes for Binary Input Discrete Memoryless Channels

研究生：林玄寅

指導教授：Stefan M. Moser 博士

陳伯寧 博士

中華民國一百零二年六月

二元無記憶通道的最佳極小區塊碼設計

Optimal Ultra-Small Block-Codes for Binary Input Discrete Memoryless Channels

研究生：林玄寅　　Student: Hsuan-Yin Lin

指導教授：莫詩台方 博士　　Advisors: Dr. Stefan M. Moser

陳伯寧 博士　　Dr. Po-Ning Chen

國立交通大學

電信工程研究所

博士論文

A Dissertation
Submitted to Institute of Communication Engineering
College of Electrical and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in
Communication Engineering
Hsinchu, Taiwan

2013 年 六 月

Information Theory Laboratory

Insititute of Communication Engineering
National Chiao Tung University

Doctor Project

Optimal Ultra-Small Block-Codes for Binary Input Discrete Memoryless Channels

Hsuan-Yin Lin

Advisors:	Prof. Dr. Stefan M. Moser	National Chiao Tung University, Taiwan
	Prof. Dr. Po-Ning Chen	National Chiao Tung University, Taiwan
Graduation Committee:	Prof. Dr. Ying Li	Yuan Ze University, Taiwan
	Prof. Dr. Mao-Chao Lin	National Taiwan University, Taiwan
	Prof. Dr. Chung-Chin Lu	National Tsing Hua University, Taiwan
	Prof. Dr. Hsiao-Feng (Francis) Lu	National Chiao Tung University, Taiwan
	Prof. Dr. Yu Ted Su	National Chiao Tung University, Taiwan
	Prof. Guu-Chang Yang	National Chung Hsing University, Taiwan

Optimal Ultra-Small Block-Codes for Binary Input Discrete Memoryless Channels

Student: Hsuan-Yin Lin

Advisors: Dr. Stefan M. Moser
Dr. Po-Ning Chen

Institute of Communications Engineering
National Chiao Tung University

Abstract

Optimal block-codes with a very small number of codewords are investigated for the binary input discrete memoryless channels. Those channels are the binary asymmetric channel (BAC), including the two special cases of the binary symmetric channel (BSC) and the Z-channel (ZC). The binary erasure channel (BEC) is a common used channel with ternary output. For the asymmetric channels, a general BAC, it is shown that so-called *flip codes* are optimal codes with two codewords. The optimal (in the sense of minimum average error probability, using maximum likelihood decoding) code structure is derived for the ZC in the cases of two, three, and four codewords and an arbitrary finite blocklength. For the symmetric channels, the BSC and the BEC, the optimal code structure is derived with at most three codewords and an arbitrary finite blocklength, a statement for linear optimal codes with four codes is also given.

The derivation of these optimal codes relies heavily on a new approach of constructing and analyzing the codebook matrix not row-wise (codewords), but *column-wise*. This new tool allows an elegant definition of interesting code families that is recursive in the blocklength n and admits their *exact* analysis of error performance that is not based on the union bound or other approximations.

二元無記憶通道的最佳極小區塊碼設計

研究生：林玄寅　　指導教授：莫詩台方 博士

陳伯寧 博士

國立交通大學電信工程研究所

摘要

在這篇論文中，我們探討數種二元無記憶通道模式下的極小區塊碼 (ultra-small block code) 的最佳設計 (optimal design)，所探討的二元無記憶通道模式包含：二元非對稱通道 (binary asymmetric channel or BAC) 與二元輸入三元輸出的二元抹除通道 (binary erasure channel or BEC)。針對前者，我們另將特別著重兩個特例，即二元對稱通道 (binary symmetric channel or BSC) 與Z-通道 (Z-channel or ZC)。本研究中所謂的最佳碼，指的是在最大概度解碼 (maximum-likelihood decoding) 法則下，可達最低平均錯誤率的區塊碼設計。而所謂的極小區塊碼指的是碼字個數極小的情況，例如2、3或4。

針對二元非對稱通道 (BAC)，我們證明了當碼字個數爲2時，相對碼 (flip codes) 爲最佳區塊碼設計。另外，針對二元非對稱通道 (BAC) 的特例Z-通道，我們在碼字個數爲2、3、4時，也提出給定任意碼長 (block length) 的最佳設計。而對於對稱式的通道，例如二元對稱通道 (BSC) 與二元抹除通道 (BEC)，我們針對碼字個數爲2與3的情況下，找到給定任意碼長的最佳碼的設計規律。此外針對這兩個對稱式的通道，在碼字個數增加爲4時，我們也設計了最佳的線性區塊碼 (linear block code)。

我們證明所設計的區塊碼可達最低的最大概度解碼錯誤率的主要關鍵技巧，乃是我們使用新的區塊碼建構觀點。簡言之，我們不用傳統碼字 (codeword) 爲基準的分析法則，而是針對區塊碼矩陣採用以直列組合方式進行分析。這種新的分析方式可以精巧的定義出必要的區塊碼類別, 使我們可用區塊碼的碼長遞迴的方式來建構碼，同時也讓我們可以推導出區塊碼的平均錯誤率的精確公式，而不需依賴傳統的聯集上限 (union bound) 或是其他所謂的錯誤率近似值的分析技巧。

Acknowledgments

First and foremost, I would like to express my deep and sincerest gratitude to Prof. Stefan M. Moser and Prof. Po-Ning Chen, my research advisers, for their insightful comments, patient guidance, enthusiastic encouragement and suggestions on any matter related this thesis. Without their help and support, this thesis would not be completed. Prof. Moser, who has supported me throughout my thesis with his patience and knowledge while allowing me the space to work in my own way. He always give me right instruction and impressed intuition guidance for the research path. He helped me get on the road to LATEX and provided an experienced ear for my doubts about writing a thesis. Prof. Chen's strong insights in information theory and mathematics leads me in a right way for proofs in my research, and his plenty of experiences for proving mathematical theorems indicate me how to do theoretical research by perseverance. I have learned numerous principal things from them, not only doing research, but also the attitude in life.

I also want to thank all the members and alumni in our Information Theory LAB. They have helped me a lot in my oral presentation and oral defense. We do have a great time during my Ph.D. time.

I am very grateful to my girlfriend Yi-Wen Wu, she always encourage and accompany me when the depress time, without her support, this thesis can not be completed and I would not continue my dream to finish my Ph.D. thesis.

Finally I would also like to extend my thanks to my parents, they never push me to do what I don't want to do. I am greatly indebted to my family.

Hsinchu, Taiwan 28 June 2013

Lin Hsuan-Yin

Contents

Chapter 1

Introduction

1.1 Introduction

Shannon proved in his ground-breaking work [1] that it is possible to find an information transmission scheme that can transmit messages at arbitrarily small error probability as long as the transmission rate in bits per channel use is below the so-called *capacity* of the channel. However, he did not provide a way on how to find such schemes. In particular, he did not tell us much about the design of codes apart from the fact that good codes may need to have a large blocklength.

For many practical applications, exactly this latter constraint is rather unfortunate as we often cannot tolerate too much delay (e.g., in inter-human communication, in time-critical control and communication, etc.). Moreover, the system complexity usually grows exponentially in the blocklength, and in consequence having large blocklength might not be an option and we have to restrict the codewords to some reasonable size. The question now arises what can theoretically be said about the performance of communication systems with such restricted block size.

During the last years, there has been an renewed interest in the theoretical understanding of finite-length coding [2]–[5]. There are several possible ways on how one can approach the problem of finite-length codes. In [2], the authors fix an acceptable error probability and a finite blocklength and then find bounds on the maximal achievable transmission rate. This parallels the method of Shannon who set the acceptable error probability to zero, but allowed infinite blocklength, and then found the maximum achievable transmission rate (the capacity). A typical example in [2] shows that for a blocklength of 1800 channel uses and for an error probability of 10^{-6}, one can achieve a rate of approximately 80 percent of the capacity of a binary symmetric channel of capacity 0.5 bits.

In another approach, one fixes the transmission rate and studies how the error probability depends on the blocklength n (i.e., one basically studies error exponents, but for relatively small n [6]). For example, [5] introduces new random coding bounds that enable a simple numerical evaluation of the error probability for finite blocklengths.

All these results have in common that they are related to Shannon's ideas in the sense

that they try to make fundamental statements about what is possible and what not. The exact manner how these systems have to be built is ignored on purpose.

Our approach in this thesis is different. Based on the insight that for very short blocklength, one has no big hope of transmitting much information with acceptable error probability, we concentrate on codes with a small *fixed* number of codewords: so-called *ultra-small block-codes*. By this reduction of the transmission rates, our results are directly applicable even for very short blocklengths. In contrast to [2] that provide bounds on the best possible theoretical performance, we try to find a *best* possible *design* that minimizes the average error probability. Hence, we put a big emphasis on finding insights in how to actually build an optimal system. In this respect, this thesis could rather be compared to [7]. There the authors try to describe the empirical distribution of good codes (i.e., of codes that approach capacity with vanishing error probability) and show that for a large enough blocklength, the empirical distribution of certain good codes converges in the sense of divergence to a set of input distributions that maximize the input-output mutual information. Note, however, that [7] again focuses on the asymptotic regime, while our focus lies on finite blocklength.

There are interesting applications for ultra-small block-codes, e.g., in the situation of establishing an initial connection in a wireless link: the amount of information that needs to be transmitted during the setup of the link is very limited, usually only a couple of bits, but these bits need to be transmitted in very short time (e.g., blocklength in the range of $n = 20$ to $n = 30$) with the highest possible reliability [8]. Another important application for ultra-small block-codes is in the area of *quality of service (QoS)*. In many delay-sensitive wireless systems like, e.g., voice over IP (VoIP) and wireless interactive and streaming video applications, it is essential to comply with certain limitations on queuing delays or buffer violation probabilities [3]–[4]. A further area where the performance of short codes is relevant is proposed in [9]: effective rateless short codes can be used to transmit some limited feedback about the channel state information in a wireless link or in some other latency-constrained application. Hence, it is of significant interest to conduct an analysis of (and to provide predictions for) the performance levels of practical finite-blocklength systems. Note that while the motivation of this work focuses on rather smaller values of n, our results nevertheless hold for arbitrary finite n.

The study of ultra-small block-codes is interesting not only because of the above mentioned direct applications, but because their analytic description is a first step to a better fundamental understanding of optimal *nonlinear* coding schemes (with ML decoding) and of their performance based on the *exact* error probability rather than on an upper bound on the achievable error probability derived from the union bound or the mutual information density bound and its statistics [10], [11].

To simplify our analysis, we have restricted ourselves for the moment to binary input and output discrete memoryless channels, that we call in their general form *binary asymmetric channels (BAC)*. The two most important special cases of the BAC, the *binary symmetric channel (BSC)* and the *Z-channel (ZC)*, are then investigated more in detail. The other channel we focus on more is the binary input and ternary output channel, which

is called *binary erasure channel (BEC)*.

Our main contributions are as follows:

- We provide first fundamental insights into the performance analysis of *optimal nonlinear code design* for the BAC. Note that there exists a vast literature about linear codes, their properties and good linear design (e.g., [12]). Some Hamming-distance related topics of nonlinear codes are addressed in [13].[1]
- We provide new insights in the optimal code construction for the BAC for an arbitrary finite blocklength n and for $\mathsf{M} = 2$ codewords.
- We provide optimal code constructions for the ZC for an arbitrary finite blocklength n and for $\mathsf{M} = 2$, 3 and 4 codewords. For the BSC, we show an achievable best code design for $\mathsf{M} = 2, 3$. We have also found the linear optimal codes for $\mathsf{M} = 4$. For the ZC we also conjecture an optimal design for $\mathsf{M} = 5$.
- We provide optimal code constructions for the BEC for an arbitrary finite blocklength n and for $\mathsf{M} = 2$, 3 codewords. We have also found the linear optimal codes for $\mathsf{M} = 4$. We also conjecture an optimal design for $\mathsf{M} = 5, 6$. For some certain blocklength, a optimal code structure is conjectured with arbitrary M.
- For the ZC, BSC, and BEC these channels, we can derive its exact performance for comparison. Some known bounds for a finite blocklegnth with fixed number of codewords are introduced.
- We propose a new approach to the design and analysis of block-codes: instead of focusing on the codewords (i.e., the rows in the codebook matrix), we look at the codebook matrix in a *column-wise* manner.

The remainder of this thesis is structured as follows: after some comments about our notation we will introduce some common definitions and our channel models in Chapter 2 and Chapter 3. After some more preliminaries in Chapter 4. Chapter 5 contains a very short example showing that the analysis of even such simple channel models is nontrivial and often nonintuitive. Chapter 6 then presents new code definitions that will be used for our main results. In Chapter 7, we review some important previous work. Chapter 8–11 then contain our main results. In Chapter 8 we analyze the BAC only for two codewords, Chapter 9 takes a closer look at the ZC. In Chapter 10 and Chapter 11, we investigate the BSC and BEC, respectively. Many of the lengthy proofs have been moved to the appendix.

As is common in coding theory, vectors (denoted by bold face Roman letters, e.g., $\mathbf{x}$) are row-vectors. However, for simplicity of notation and to avoid a large number of transpose-signs, we slightly misuse this notational convention for one special case: any vector $\mathbf{c}$ is a column-vector. It should be always clear from the context because these

[1] Note that some of the code designs proposed in this thesis actually have interesting "linear-like" properties and can be considered as generalizations of linear codes with 2^k codewords to codes with a general number of codewords M. For more details see [14].

vectors are used to build codebook matrices and are therefore also conceptually quite different from the transmitted codewords $\mathbf{x}$ or the received sequence $\mathbf{y}$. Otherwise our used notation follows the main stream. We use capital letters for random quantities and small letters for realizations; sets are denoted by a calligraphic font, e.g., $\mathcal{D}$; and constants are depicted by Greek letters, small Romans or a special font, e.g., M.

Chapter 2

Definitions

2.1 Discrete Memoryless Channel

The probably most fundamental model describing communication over a noisy channel is the so-called *discrete memoryless channel (DMC)*. A DMC consists of a

- a finite input alphabet $\mathcal{X}$;
- a finite output alphabet $\mathcal{Y}$; and
- a conditional probability distribution $P_{Y|X}(\cdot|x)$ for all $x \in \mathcal{X}$ such that

$$P_{Y_k|X_1,X_2,\ldots,X_k,Y_1,Y_2,\ldots,Y_{k-1}}(y_k|x_1,x_2,\ldots,x_k,y_1,y_2,\ldots,y_{k-1}) = P_{Y|X}(y_k|x_k) \qquad \forall\, k. \quad (2.1)$$

Note that a DMC is called *memoryless* because the current output Y_k depends only on the current input x_k. Moreover also note that the channel is *time-invariant* in the sense that for a particular input x_k, the distribution of the output Y_k does not change over time.

Definition 2.1 *We say a DMC is used* without feedback*, if*

$$P(x_k|x_1,\ldots,x_{k-1},y_1,\ldots,y_{k-1}) = P(x_k|x_1,\ldots,x_{k-1}) \quad \forall\, k, \qquad (2.2)$$

i.e., X_k depends only on past inputs (by choice of the encoder), but not on past outputs. Hence, there is no feedback link from the receiver back to the transmitter that would inform the transmitter about the last outputs.

Note that even though we assume the channel to be memoryless, we do *not* restrict the encoder to be memoryless! We now have the following theorem.

Theorem 2.2 *If a DMC is used without feedback, then*

$$P(y_1,\ldots,y_n|x_1,\ldots,x_n) = \prod_{k=1}^{n} P_{Y|X}(y_k|x_k) \quad \forall\, n \geq 1. \qquad (2.3)$$

Proof: See, e.g., [15]. □

2.2 Coding for DMC

Definition 2.3 *A* (M, n) coding scheme *for a DMC* $(\mathcal{X}, \mathcal{Y}, P_{Y|X})$ *consists of*

- *the* message set $\mathcal{M} = \{1, \ldots, \mathsf{M}\}$ *of* M *equally likely random messages* M*;*
- *the* (M, n) codebook *(or simply* code*) consisting of* M *length-*n *channel input sequences, called* codewords*;*
- *an* encoding function $f\colon \mathcal{M} \to \mathcal{X}^n$ *that assigns for every message* $m \in \mathcal{M}$ *a codeword* $\mathbf{x} = (x_1, \ldots, x_n)$*; and*
- *a* decoding function $g\colon \mathcal{Y}^n \to \hat{\mathcal{M}}$ *that maps the received channel output* n*-sequence* $\mathbf{y}$ *to a guess* $\hat{m} \in \hat{\mathcal{M}}$*. (Usually, we have* $\hat{\mathcal{M}} = \mathcal{M}$*.)*

Note that an (M, n) code consist merely of a unsorted list of M codewords of length n, whereas an (M, n) coding scheme additionally also defines the encoding and decoding functions. Hence, the same code can be part of many different coding schemes.

Definition 2.4 *A code is called* linear *if the sum of any two codewords again is a codeword.*

Note that a linear code always contains the all-zero codeword.

The two main parameters of interest of a code are the number of possible messages M (the larger, the more information is transmitted) and the blocklength n (the shorter, the less time is needed to transmit the message):

- we have M equally likely messages, i.e., the entropy is $H(M) = \log_2 \mathsf{M}$ bits and we need $\log_2 \mathsf{M}$ bits to describe the message in binary form;
- we need n transmissions of a channel input symbol X_k over the channel in order to transmit the complete message.

Hence, it makes sense to give the following definition.

Definition 2.5 *The* rate[2] *of a* (M, n) *code is defined as*

$$\mathsf{R} \triangleq \frac{\log_2 \mathsf{M}}{n} \text{ bits/transmission.} \tag{2.4}$$

It describes what amount of information (i.e., what part of the $\log_2 \mathsf{M}$ bits) is transmitted in each channel use.

However, this definition of a rate makes only sense if the message really arrives at the receiver, i.e., if the receiver does not make a decoding error!

[2] We define the rate here using a logarithm of base 2. However, we can use any logarithm as long as we adapt the units accordingly.

Definition 2.6 *An* (M, n) *coding scheme for a DMC consists of a codebook* $\mathscr{C}^{(\mathsf{M},n)}$ *with* M *codewords* $\mathbf{x}_m$ *of length* n *(*$m = 1, \ldots, \mathsf{M}$*), an encoder that maps every message* m *into its corresponding codeword* $\mathbf{x}_m$*, and a decoder that makes a decoding decision* $g(\mathbf{y}) \in \{1, \ldots, \mathsf{M}\}$ *for every received binary* n*-vector* $\mathbf{y}$*.*

We will always assume that the M possible messages are equally likely.

Definition 2.7 *Given that message* m *has been sent, let* $\lambda_m\big(\mathscr{C}^{(\mathsf{M},n)}\big)$ *be the* probability of a decoding error *of an* (M, n) *coding scheme with blocklength* n*:*

$$\lambda_m\big(\mathscr{C}^{(\mathsf{M},n)}\big) \triangleq \Pr[g(\mathbf{Y}) \neq m | \mathbf{X} = \mathbf{x}_m] \tag{2.5}$$

$$= \sum_{\mathbf{y}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m)\ \mathrm{I}\{g(\mathbf{y}) \neq m\}, \tag{2.6}$$

where $\mathrm{I}\{\cdot\}$ *is the indicator function*

$$\mathrm{I}\{\text{statement}\} \triangleq \begin{cases} 1 & \text{if statement is true,} \\ 0 & \text{if statement is wrong.} \end{cases} \tag{2.7}$$

The maximum error probability $\lambda\big(\mathscr{C}^{(\mathsf{M},n)}\big)$ *of an* (M, n) *coding scheme is defined as*

$$\lambda\big(\mathscr{C}^{(\mathsf{M},n)}\big) \triangleq \max_{m \in \mathcal{M}} \lambda_m\big(\mathscr{C}^{(\mathsf{M},n)}\big). \tag{2.8}$$

The average error probability $P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big)$ *of an* (M, n) *coding scheme is defined as*

$$P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) \triangleq \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \lambda_m\big(\mathscr{C}^{(\mathsf{M},n)}\big). \tag{2.9}$$

Moreover, sometimes it will be more convenient to focus on the probability of not making any error, denoted success probability $\psi_m\big(\mathscr{C}^{(\mathsf{M},n)}\big)$*:*

$$\psi_m\big(\mathscr{C}^{(\mathsf{M},n)}\big) \triangleq \Pr[g(\mathbf{Y}) = m | \mathbf{X} = \mathbf{x}_m] \tag{2.10}$$

$$= \sum_{\mathbf{y}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m) \mathrm{I}\{g(\mathbf{y}) = m\}. \tag{2.11}$$

The definition of minimum success probability $\psi\big(\mathscr{C}^{(\mathsf{M},n)}\big)$ *and the average success probability*[3] $P_{\mathrm{c}}\big(\mathscr{C}^{(\mathsf{M},n)}\big)$ *are accordingly.*

Definition 2.8 *For a given* (M, n) *coding scheme, we define the* decoding region $\mathcal{D}_m^{(\mathsf{M},n)}$ *as the set of* n*-vectors* $\mathbf{y}$ *corresponding to the* m*-th codeword* $\mathbf{x}_m$ *as follows:*

$$\mathcal{D}_m^{(\mathsf{M},n)} \triangleq \{\mathbf{y} \colon g(\mathbf{y}) = m\}. \tag{2.12}$$

[3]The subscript "c" stands for "correct."

Note that we will always assume that the M possible messages are equally likely and that the decoder g is a *maximum likelihood (ML) decoder*:

$$g(\mathbf{y}) \triangleq \arg \max_{1 \le m \le \mathsf{M}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m) \tag{2.13}$$

that minimizes the average error probability $P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big)$ among all possible decoders.

Hence, we are going to be lazy and directly concentrate on the set of codewords $\mathscr{C}^{(\mathsf{M},n)}$, called (M, n) *codebook* or usually simply (M, n) *code*. Sometimes we follow the custom of traditional coding theory and use three parameters: $\big(\mathsf{M}, n, d\big)$ *code*, where the third parameter d denotes the *minimum Hamming distance*, i.e., the minimum number of components in which any two codewords differ.

Moreover, we also make the following definitions.

Definition 2.9 *By $d_{\alpha,\beta}(\mathbf{x}_m, \mathbf{y})$ we denote the number of positions j, where $x_{m,j} = \alpha$ and $y_j = \beta$. For $m \neq m'$, the* joint composition *$q_{\alpha,\beta}(m, m')$ of two codewords $\mathbf{x}_m$ and $\mathbf{x}_{m'}$ is defined as*

$$q_{\alpha,\beta}(m, m') \triangleq \frac{d_{\alpha,\beta}(\mathbf{x}_m, \mathbf{x}_{m'})}{n}. \tag{2.14}$$

Note that $d_{\mathrm{H}}(\cdot,\cdot) \triangleq d_{0,1}(\cdot,\cdot) + d_{1,0}(\cdot,\cdot)$ and $w_{\mathrm{H}}(\mathbf{x}) \triangleq d_{\mathrm{H}}(\mathbf{x}, \mathbf{0})$ denote the commonly used Hamming distance and Hamming weight, respectively.

The following remark deals with the way how codebooks can be described. It is not standard, but turns out to be very important and is actually the clue to our derivations.

Remark 2.10 *Usually, the codebook $\mathscr{C}^{(\mathsf{M},n)}$ is written as an* $\mathsf{M} \times n$ codebook matrix *with the M rows corresponding to the M codewords:*

$$\mathscr{C}^{(\mathsf{M},n)} = \begin{pmatrix} — \, \mathbf{x}_1 \, — \\ \vdots \\ — \, \mathbf{x}_{\mathsf{M}} \, — \end{pmatrix} = \begin{pmatrix} | & | & & | \\ \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_n \\ | & | & & | \end{pmatrix}. \tag{2.15}$$

However, it turns out to be much more convenient to consider the codebook column-wise *rather than row-wise! So, instead of specifying the codewords of a codebook, we actually specify its (length-M) column-vectors $\mathbf{c}_j$.*

Remark 2.11 *Since we assume equally likely messages, any permutation of rows only changes the assignment of codewords to messages and has no impact on the performance. We consider two codes with permuted rows as being* equal, *i.e., a code is actually a* set *of codewords, where the ordering of the codewords is irrelevant.*

Furthermore, since we are only considering memoryless channels, any permutation of the columns of $\mathscr{C}^{(\mathsf{M},n)}$ will lead to another codebook that is equivalent to the first in the sense that it has the exact same error probability. We say that such two codes are equivalent. *We would like to emphasize that two codebooks being equivalent is not the same as two codebooks being equal. However, as we are mainly interested in the performance of*

a codebook, we usually treat two equivalent codes as being the same. In particular, when we speak of a unique code design*, we do not exclude the always possible permutations of columns.*

In spite of this, for the sake of clarity of our derivations, we usually will define a certain fixed order of the codewords/codebook column vectors.

The most famous relation between code rate and error probability has been derived by Shannon in his landmark paper from 1948 [1].

Theorem 2.12 (The Channel Coding Theorem for a DMC) *Define*

$$\mathsf{C} \triangleq \max_{P_X(\cdot)} I(X;Y) \tag{2.16}$$

where X and Y have to be understood as input and output of a DMC and where the maximization is over all input distributions $P_X(\cdot)$.

Then for every $\mathsf{R} < \mathsf{C}$ there exists a sequence of $(2^{n\mathsf{R}}, n)$ coding schemes with maximum error probability $\lambda\big(\mathscr{C}^{(\mathsf{M},n)}\big) \to 0$ as the blocklength n gets very large.

Conversely, any sequence of $(2^{n\mathsf{R}}, n)$ coding schemes with maximum error probability $\lambda\big(\mathscr{C}^{(\mathsf{M},n)}\big) \to 0$ must have a rate $\mathsf{R} \leq \mathsf{C}$.

So we see that C denotes the maximum rate at which reliable communication is possible. Therefore C is called ***channel capacity****.*

Note that this theorem considers only the situation of n tending to infinity and thereby the error probability going to zero. However, in a practical system, we cannot allow the blocklength n to be too large because of delay and complexity. On the other hand it is not necessary to have zero error probability either.

So the question arises what we can say about "capacity" for finite n, i.e., if we allow a certain maximal probability of error, what is the smallest necessary blocklength n to achieve it? Or, vice versa, fixing a certain short blocklength n, what is the best average error probability that can be achieved? And, what is the optimal code structure for a given channel?

Chapter 3

Channel Models

We consider a discrete memoryless channel (DMC) with both a binary input and a binary output alphabets. The most general such binary DMC is the so-called *binary asymmetric channel (BAC)* and is specified by two parameters: ϵ_0 denotes the probability that a 0 is flipped into a 1, and ϵ_1 denotes the probability that a 1 is flipped into a 0, see Fig. 3.1.

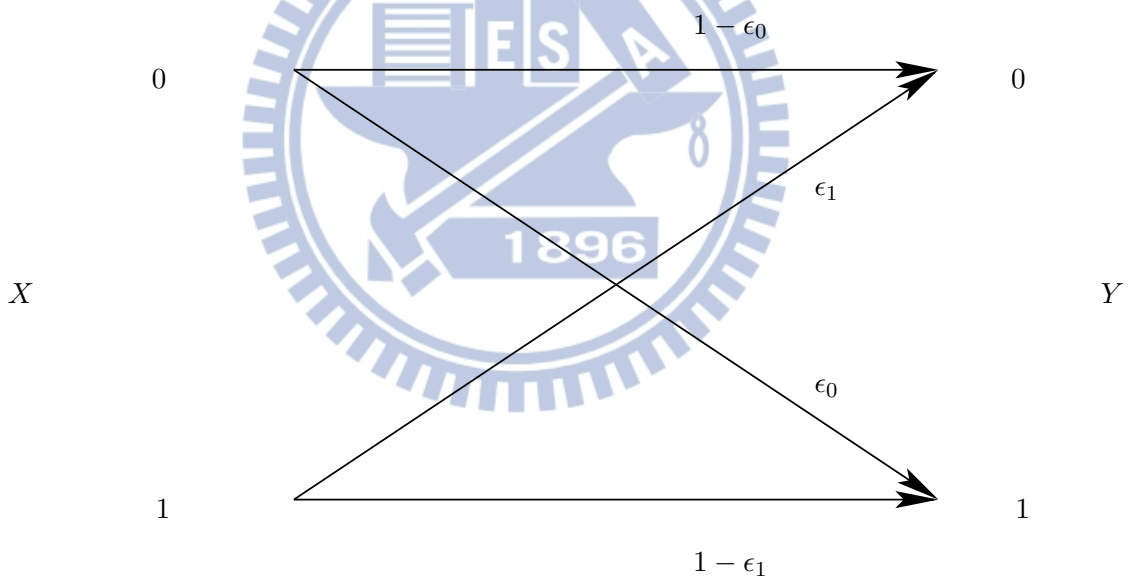


Figure 3.1: The binary asymmetric channel (BAC).

For symmetry reasons and without loss of generality, we can restrict the values of these parameters as follows:

$$0 \leq \epsilon_0 \leq \epsilon_1 \leq 1 \tag{3.1}$$

$$\epsilon_0 \leq 1 - \epsilon_0 \tag{3.2}$$

$$\epsilon_0 \leq 1 - \epsilon_1. \tag{3.3}$$

Note that in the case when $\epsilon_0 > \epsilon_1$, we simply flip all zeros to ones and vice versa to get

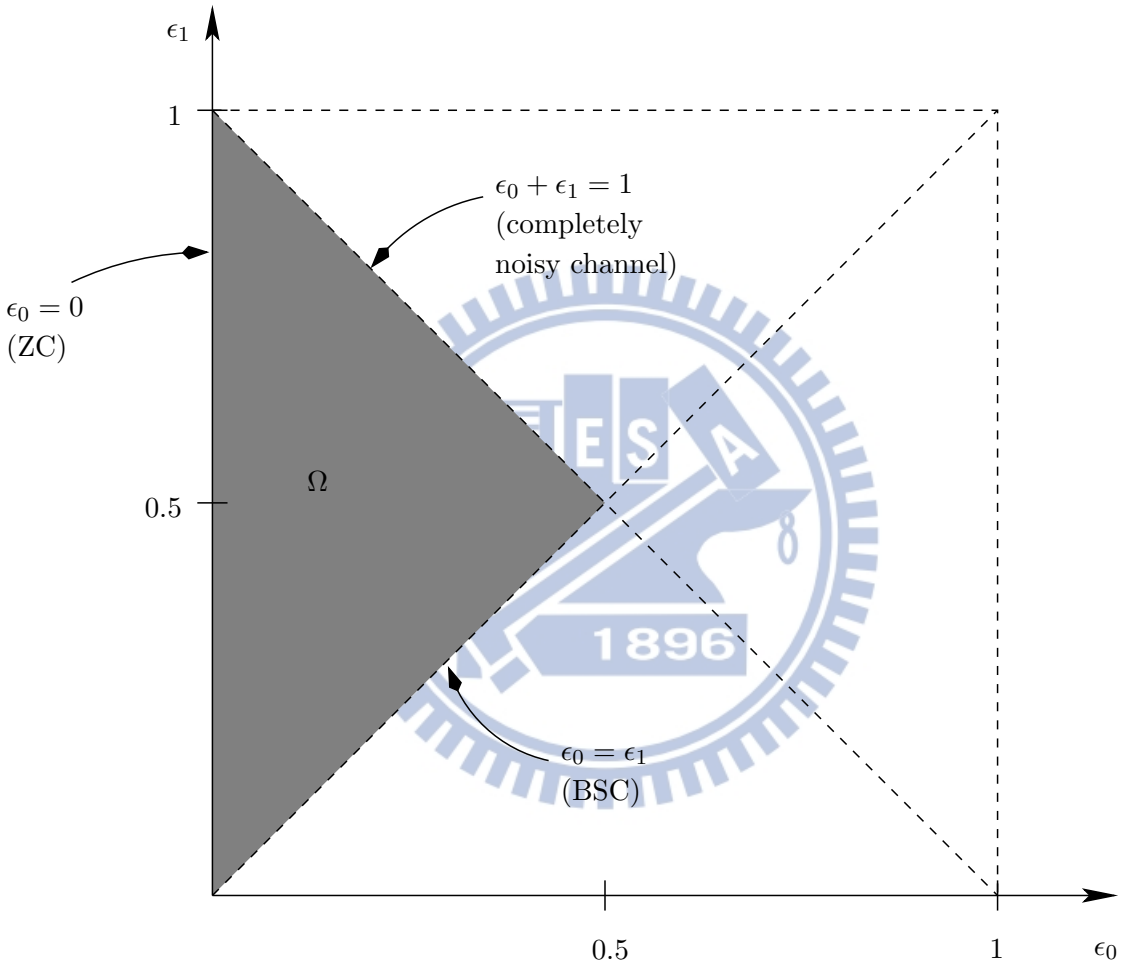


Figure 3.2: Region of possible choices of the channel parameters ϵ_0 and ϵ_1 of a BAC. The shaded area corresponds to the interesting area according to (3.1)–(3.3).

an equivalent channel with $\epsilon_0 \leq \epsilon_1$. For the case when $\epsilon_0 > 1 - \epsilon_0$, we flip the output Y, i.e., change all output zeros to ones and ones to zeros, to get an equivalent channel with $\epsilon_0 \leq 1 - \epsilon_0$. Note that (3.2) can be simplified to $\epsilon_0 \leq \frac{1}{2}$ and is actually implied by (3.1) and (3.3). And for the case when $\epsilon_0 > 1 - \epsilon_1$, we flip the input X to get an equivalent channel that satisfies $\epsilon_0 \leq 1 - \epsilon_1$.

We have depicted the region of possible choices of the parameters ϵ_0 and ϵ_1 in Fig. 3.2. The region of interesting choices given by (3.1)–(3.3) is denoted by Ω.

Note that the boundaries of Ω correspond to three special cases: The *binary symmetric channel (BSC)* (see Fig. 3.3) has equal cross-over probabilities $\epsilon_0 = \epsilon_1 = \epsilon$. According to (3.2), we can assume without loss of generality that $\epsilon \leq \frac{1}{2}$.

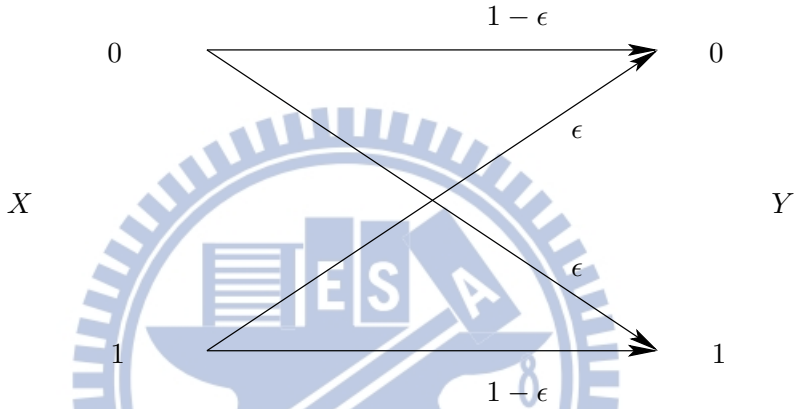


Figure 3.3: The binary symmetric channel (BSC).

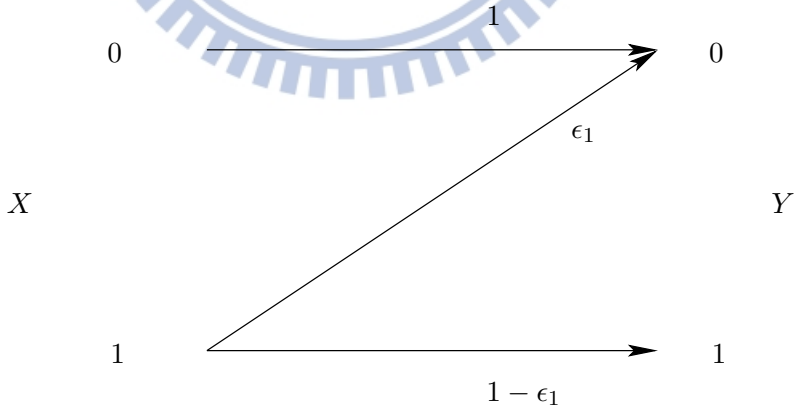


Figure 3.4: The Z-channel (ZC).

The *Z-channel (ZC)* (see Fig. 3.4) will never distort an input 0, i.e., $\epsilon_0 = 0$. An input 1 is flipped to 0 with probability $\epsilon_1 < 1$.

Finally, the case $\epsilon_0 = 1 - \epsilon_1$ corresponds to a completely noisy channel of zero capacity: given $Y = y$, the events $X = 0$ and $X = 1$ are equally likely, i.e., $X \perp\!\!\!\perp Y$.

A special case of the binary input and ternary output channel is BEC, which is not belong the special case of BAC, we have the transition probability, δ, from zero to one. The output alphabets of BEC are $\{0, 1, 2\}$, which is not binary. Here is the channel model defined as the following figure:

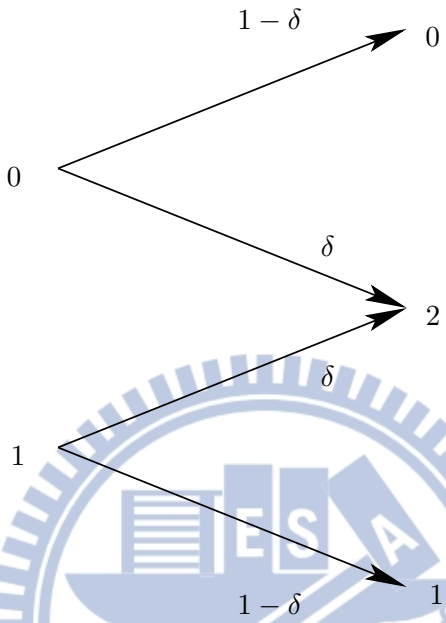


Figure 3.5: BEC

Due to the symmetry of the BSC and BEC, we have an additional equivalence in the codebook design.

Lemma 3.1 *Consider an arbitrary code $\mathscr{C}^{(\mathsf{M},n)}$ to be used on the BSC or BEC and consider an arbitrary M-vector $\mathbf{c}$. Now construct a new length-$(n+1)$ code $\mathscr{C}^{(\mathsf{M},n+1)}$ by appending $\mathbf{c}$ to the codebook matrix of $\mathscr{C}^{(\mathsf{M},n)}$ and a new length-$(n+1)$ code $\overline{\mathscr{C}}^{(\mathsf{M},n+1)}$ by appending the flipped vector $\bar{\mathbf{c}} = \mathbf{c} \oplus \mathbf{1}$ to the codebook matrix of $\mathscr{C}^{(\mathsf{M},n)}$. Then the performance of these two new codes is identical:*

$$P_{\mathrm{e}}^{(n+1)}\big(\mathscr{C}^{(\mathsf{M},n+1)}\big) = P_{\mathrm{e}}^{(n+1)}\Big(\overline{\mathscr{C}}^{(\mathsf{M},n+1)}\Big). \tag{3.4}$$

We remind the reader that our ultimate goal is to find the structure of an optimal code $\mathscr{C}^{(\mathsf{M},n)*}$ that satisfies

$$P_{\mathrm{e}}^{(n)}\big(\mathscr{C}^{(\mathsf{M},n)*}\big) \le P_{\mathrm{e}}^{(n)}\big(\mathscr{C}^{(\mathsf{M},n)}\big) \tag{3.5}$$

for any code $\mathscr{C}^{(\mathsf{M},n)}$.

Chapter 4

Preliminaries

4.1 Error Probability of the BAC

The conditional probability of the received vector $\mathbf{y}$ given the sent codeword $\mathbf{x}_m$ of the BAC can be written as

$$P_{Y|X}^n(\mathbf{y}|\mathbf{x}_m) = (1-\epsilon_0)^{d_{0,0}(\mathbf{x}_m,\mathbf{y})} \cdot \epsilon_0^{d_{0,1}(\mathbf{x}_m,\mathbf{y})} \cdot \epsilon_1^{d_{1,0}(\mathbf{x}_m,\mathbf{y})} \cdot (1-\epsilon_1)^{d_{1,1}(\mathbf{x}_m,\mathbf{y})} \tag{4.1}$$

where we use $P_{Y|X}^n$ to denote the product distribution

$$P_{Y|X}^n(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{n} P_{Y|X}(y_j|x_j). \tag{4.2}$$

Considering that

$$n = d_{0,0}(\mathbf{x}_m,\mathbf{y}) + d_{0,1}(\mathbf{x}_m,\mathbf{y}) + d_{1,0}(\mathbf{x}_m,\mathbf{y}) + d_{1,1}(\mathbf{x}_m,\mathbf{y}) \tag{4.3}$$

the average error probability of a coding scheme $\mathscr{C}^{(\mathsf{M},n)}$ over a BAC can now be written as

$$P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) = \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})\neq m}} P_{Y|X}^n(\mathbf{y}|\mathbf{x}_m) \tag{4.4}$$

$$= \frac{(1-\epsilon_0)^n}{\mathsf{M}} \sum_{\mathbf{y}} \sum_{\substack{m=1 \\ m\neq g(\mathbf{y})}}^{\mathsf{M}} \left(\frac{\epsilon_0}{1-\epsilon_0}\right)^{d_{0,1}(\mathbf{x}_m,\mathbf{y})} \cdot \left(\frac{\epsilon_1}{1-\epsilon_0}\right)^{d_{1,0}(\mathbf{x}_m,\mathbf{y})} \left(\frac{1-\epsilon_1}{1-\epsilon_0}\right)^{d_{1,1}(\mathbf{x}_m,\mathbf{y})} \tag{4.5}$$

where $g(\mathbf{y})$ is the ML decision (2.13) for the observation $\mathbf{y}$.

4.1.1 Capacity of the BAC

Without loss of generality, we can only consider BACs with $0 \le \epsilon_0 \le \epsilon_1 \le 1$ and $0 \le \epsilon_0 + \epsilon_1 \le 1$.

Proposition 4.1 *The capacity of a BAC is given by*

$$\mathrm{C}_{\mathrm{BAC}} = \frac{\epsilon_1}{1-\epsilon_0-\epsilon_1} \cdot H_{\mathrm{b}}(\epsilon_0) - \frac{1-\epsilon_0}{1-\epsilon_0-\epsilon_1} \cdot H_{\mathrm{b}}(\epsilon_1) + \log_2\left(1 + 2^{\frac{H_{\mathrm{b}}(\epsilon_1)-H_{\mathrm{b}}(\epsilon_0)}{1-\epsilon_0-\epsilon_1}}\right) \tag{4.6}$$

bits, where $H_{\mathrm{b}}(\cdot)$ is the binary entropy function defined as

$$H_{\mathrm{b}}(p) \triangleq -p\log_2 p - (1-p)\log_2(1-p). \tag{4.7}$$

The input distribution $P_X^*(\cdot)$ that achieves this capacity is given by

$$P_X^*(0) = 1 - P_X^*(1) = \frac{z - \epsilon_1(1+z)}{(1+z)(1-\epsilon_0-\epsilon_1)} \tag{4.8}$$

with

$$z \triangleq 2^{\frac{H_{\mathrm{b}}(\epsilon_1)-H_{\mathrm{b}}(\epsilon_0)}{1-\epsilon_0-\epsilon_1}}. \tag{4.9}$$

4.2 Error (and Success) Probability of the ZC

In the special case of a ZC, the average success probability can be expressed as follows:

$$\begin{aligned}
&P_{\mathrm{c}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) \\
&= \frac{1}{\mathsf{M}} \sum_{\mathbf{y}} \sum_{\substack{m=1 \\ g(\mathbf{y})=m}}^{\mathsf{M}} \mathrm{I}\left\{d_{01}(\mathbf{x}_m,\mathbf{y})=0\right\} \epsilon_1^{d_{10}(\mathbf{x}_m,\mathbf{y})} (1-\epsilon_1)^{d_{11}(\mathbf{x}_m,\mathbf{y})} && (4.10)\\
&= \frac{1}{\mathsf{M}} \sum_{\mathbf{y}} \sum_{\substack{m=1 \\ g(\mathbf{y})=m}}^{\mathsf{M}} \mathrm{I}\left\{d_{0,1}(\mathbf{x}_m,\mathbf{y})=0\right\} \left(\frac{\epsilon_1}{1-\epsilon_1}\right)^{d_{1,0}(\mathbf{x}_m,\mathbf{y})} \cdot (1-\epsilon_1)^{d_{1,1}(\mathbf{x}_m,\mathbf{y})+d_{1,0}(\mathbf{x}_m,\mathbf{y})} && (4.11)\\
&= \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})=m}} \mathrm{I}\left\{d_{0,1}(\mathbf{x}_m,\mathbf{y})=0\right\} \left(\frac{\epsilon_1}{1-\epsilon_1}\right)^{d_{1,0}(\mathbf{x}_m,\mathbf{y})} \cdot (1-\epsilon_1)^{w_{\mathrm{H}}(\mathbf{x}_m)}. && (4.12)
\end{aligned}$$

The error probability formula is accordingly

$$P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) = \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})\neq m}} \mathrm{I}\left\{d_{0,1}(\mathbf{x}_m,\mathbf{y})=0\right\} \cdot \left(\frac{\epsilon_1}{1-\epsilon_1}\right)^{d_{1,0}(\mathbf{x}_m,\mathbf{y})} (1-\epsilon_1)^{w_{\mathrm{H}}(\mathbf{x}_m)}. \tag{4.13}$$

From (B.52), note that the capacity-achieving distribution for $\epsilon_1 = \frac{1}{2}$ is

$$P_X^*(1) = \frac{2}{5}. \tag{4.14}$$

The capacity-achieving distribution is strongly depends on the cross-over probability ϵ_1.

4.3 Error (and Success) Probability of the BSC

In the special case of a BSC, (4.5) simplifies to

$$P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) = \frac{(1-\epsilon)^n}{\mathsf{M}} \sum_{\mathbf{y}} \sum_{\substack{m=1 \\ g(\mathbf{y})\neq m}}^{\mathsf{M}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_{\mathrm{H}}(\mathbf{x}_m,\mathbf{y})}. \tag{4.15}$$

The success probability is accordingly

$$P_{\mathrm{c}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) = \frac{(1-\epsilon)^n}{\mathsf{M}} \sum_{\mathbf{y}} \sum_{\substack{m=1 \\ g(\mathbf{y})=m}}^{\mathsf{M}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_{\mathrm{H}}(\mathbf{x}_m,\mathbf{y})}. \tag{4.16}$$

4.3.1 Capacity of the BSC

From (C.97), the capacity of a BSC is given by

$$\mathsf{C}_{\mathrm{BSC}} = 1 - H_{\mathrm{b}}(\epsilon) \tag{4.17}$$

bits. The input distribution $P_X^*(\cdot)$ that achieve the capacity is the uniform distribution given by

$$P_X^*(0) = 1 - P_X^*(1) = \frac{1}{2}, \tag{4.18}$$

which is irrelevant to the cross-over prbability ϵ.

4.4 Error (and Success) Probability of the BEC

The only difference of BEC is the output turn out to be ternary.

Definition 4.2 *To make the conditional probability express shortly, we defined the number of times the symbol a occurs in one received vector $\mathbf{y}$ by $\mathsf{N}(a|\mathbf{y})$. By $\mathrm{I}(a|\mathbf{y})$ we denote the set of indices i such that $y_i = a$, hence $\mathsf{N}(a|\mathbf{y}) = |\mathrm{I}(a|\mathbf{y})|$, i.e., $\mathbf{x}_{\mathrm{I}(a|\mathbf{y})}$ is a vector of length $\mathsf{N}(a|\mathbf{y})$ containing all x_i where $i \in \mathrm{I}(a|\mathbf{y})$.*

It is often easier to maximize the success probability instead of minimizing the error probability. For the convenience of later derivations, we are going to derive its error and success probabilities:

$$P_{\mathrm{c}}(\mathcal{C}^{(\mathsf{M},n)}) = \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})=m}} (1-\epsilon)^{n-\mathsf{N}(2|\mathbf{y})} \cdot \epsilon^{\mathsf{N}(2|\mathbf{y})} \cdot \mathrm{I}\left\{d_{\mathrm{H}}\left(\mathbf{x}_{m\,\mathrm{I}(b|\mathbf{y})}, \mathbf{y}_{\mathrm{I}(b|\mathbf{y})}\right) = 0\right\}, \tag{4.19}$$

where $b \in \{0, 1\}$. The error probability formula is accordingly

$$P_{\mathrm{e}}(\mathcal{C}^{(\mathsf{M},n)}) = \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y}) \neq m}} (1-\epsilon)^{n-\mathsf{N}(2|\mathbf{y})} \cdot \epsilon^{\mathsf{N}(2|\mathbf{y})} \cdot \mathrm{I}\left\{ d_{\mathrm{H}}\left(\mathbf{x}_{m\,\mathrm{I}(b|\mathbf{y})}, \mathbf{y}_{\mathrm{I}(b|\mathbf{y})} \right) = 0 \right\} \tag{4.20}$$

4.4.1 Capacity of the BEC

The capacity of a BEC is given by

$$\mathsf{C}_{\mathrm{BEC}} = 1 - \delta \tag{4.21}$$

bits. The input distribution $P_X^*(\cdot)$ that achieve the capacity is the uniform distribution given by

$$P_X^*(0) = 1 - P_X^*(1) = \frac{1}{2}, \tag{4.22}$$

which is also irrelevant to the cross-over probability δ.

4.5 Pairwise Hamming Distance

The minimum Hamming distance is a well-known and often used quality criterion of a codebook [12], [13]. [13, Ch. 2] discusses the maximum minimum Hamming distance for a given code $\mathscr{C}^{(\mathsf{M},n)}$, e.g., the Plotkin bound and Levenshtein's theorem. (For discussions of the upper and lower bounds to average error probability, see Chapter 7.) Unfortunately, a design based on the minimum Hamming distance can fail even for linear codes and even for a very symmetric channel like the BSC, whose error probability performance is completely specified by the Hamming distances between codewords and received vectors.

We therefore define a slightly more general and more concise description of a codebook: the *pairwise Hamming distance vector*.

Definition 4.3 *Given a codebook $\mathscr{C}^{(\mathsf{M},n)}$ with codewords $\mathbf{x}_m$, $1 \le m \le \mathsf{M}$, we define the length $\frac{1}{2}(\mathsf{M}-1)\mathsf{M}$* pairwise Hamming distance vector

$$\begin{aligned} &\mathbf{d}\big(\mathscr{C}^{(\mathsf{M},n)}\big) \\ &\quad \triangleq \Big(d_{\mathrm{H}}(\mathbf{x}_1, \mathbf{x}_2), \\ &\qquad d_{\mathrm{H}}(\mathbf{x}_1, \mathbf{x}_3),\ d_{\mathrm{H}}(\mathbf{x}_2, \mathbf{x}_3), \\ &\qquad d_{\mathrm{H}}(\mathbf{x}_1, \mathbf{x}_4),\ d_{\mathrm{H}}(\mathbf{x}_2, \mathbf{x}_4),\ d_{\mathrm{H}}(\mathbf{x}_3, \mathbf{x}_4), \\ &\qquad \ldots, \\ &\qquad d_{\mathrm{H}}(\mathbf{x}_1, \mathbf{x}_{\mathsf{M}}),\ d_{\mathrm{H}}(\mathbf{x}_2, \mathbf{x}_{\mathsf{M}}), \ldots,\ d_{\mathrm{H}}(\mathbf{x}_{\mathsf{M}-1}, \mathbf{x}_{\mathsf{M}}) \Big). \end{aligned} \tag{4.23}$$

The minimum Hamming distance $d_{\min}\big(\mathscr{C}^{(\mathsf{M},n)}\big)$ *is then defined as the minimum component of the pairwise Hamming distance vector* $\mathbf{d}\big(\mathscr{C}^{(\mathsf{M},n)}\big)$.

Chapter 5

A Counterexample

To show that the search for an optimal (possibly nonlinear) code is neither trivial nor intuitive even in the symmetric BSC case, we would like to start with a simple example before we summarize our main results.

Assume a BSC with cross probability $\epsilon = 0.4$, $\mathsf{M} = 4$, and a blocklength $n = 4$. Then consider the following codes:[4]

$$\mathscr{C}_1^{(4,4)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \qquad \mathscr{C}_2^{(4,4)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}. \tag{5.1}$$

We observe that while both codes are linear (i.e., any sum of two codewords is also a codeword), the first code has a minimum Hamming distance 1, and the second has a minimum Hamming distance 2. It is quite common to believe that $\mathscr{C}_2^{(4,4)}$ shows a better performance. This intuition is based on Gallager's famous performance bound [6, Exercise 5.19]:

$$P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) \le (\mathsf{M}-1)e^{-d_{\min}(\mathscr{C}^{(\mathsf{M},n)})\log\frac{1}{\sqrt{4\epsilon(1-\epsilon)}}}. \tag{5.2}$$

However, the exact average error probability as given in (4.15) actually can be evaluated as $P_{\mathrm{e}}\big(\mathscr{C}_1^{(4,4)}\big) \approx 0.6112$ and $P_{\mathrm{e}}\big(\mathscr{C}_2^{(4,4)}\big) = 0.64$. Hence, even though the minimum Hamming distance of the first codebook is smaller, its overall performance is superior to the second codebook!

Our goal is to find the structure of an optimal code $\mathscr{C}^{(\mathsf{M},n)*}$ that satisfies

$$P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)*}\big) \le P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) \tag{5.3}$$

for any code $\mathscr{C}^{(\mathsf{M},n)}$.

[4] We will see in Chapter 6 that both codes are *weak flip codes*. In this example, $\mathscr{C}_1^{(4,4)} = \mathscr{C}_{1,0}^{(4,4)}$ and $\mathscr{C}_2^{(4,4)} = \mathscr{C}_{2,0}^{(4,4)}$ according to Definition 6.5 given later.

Chapter 6

Flip Codes, Weak Flip Codes and Hadamard Codes

We next introduce some special families of binary codes. We start with a family of codes with two codewords.

Definition 6.1 *The* flip code of type t *for* $t \in \{0, 1, \ldots, \lfloor \frac{n}{2} \rfloor\}$ *is a code with* $\mathsf{M} = 2$ *codewords defined by the following codebook matrix* $\mathscr{C}_t^{(2,n)}$*:*

$$\mathscr{C}_t^{(2,n)} \triangleq \begin{pmatrix} \mathbf{x} \\ \bar{\mathbf{x}} \end{pmatrix} = \begin{pmatrix} 0 & \cdots & \overbrace{0 & 1 & \cdots & 1}^{t \text{ columns}} \\ 1 & \cdots & 1 & 0 & \cdots & 0 \end{pmatrix}. \tag{6.1}$$

Defining the column vectors

$$\left\{ \mathbf{c}_1^{(2)} \triangleq \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \mathbf{c}_2^{(2)} \triangleq \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}, \tag{6.2}$$

we see that a flip code of type t *is given by a codebook matrix that consists of* $n-t$ *columns* $\mathbf{c}_1^{(2)}$ *and* t *columns* $\mathbf{c}_2^{(2)}$*.*

We again remind the reader that due to the memorylessness of the BEC, other codes with the same columns as $\mathscr{C}_t^{(2,n)}$, but in different order are equivalent to $\mathscr{C}_t^{(2,n)}$. Moreover, we would like to point out that while the flip code of type 0 corresponds to a repetition code, the general flip code of type t with $t > 0$ is neither a repetition code nor is it even linear.

We have shown in [16] that for any blocklength n and for a correct choice[5] of t, the flip codes are optimal on *any* binary-input binary-output channel for arbitrary channel parameters. In particular, they are optimal for the BSC and the ZC [16].

The columns given in the set in (6.2) are called *candidate columns*. They are flipped versions of each other, therefore also the name of the code.

[5] We would like to emphasize that the optimal choice of t for many binary channels is not 0, i.e., the linear repetition code is not optimal!

The definition of a flip code with one codeword being the flipped version of the other cannot be easily extended to a situation with more than two codewords. Hence, for $\mathsf{M} > 2$, we need a new approach. We give the following definition.

Definition 6.2 *Given an* $\mathsf{M} > 2$*, a length-*M *candidate column* $\mathbf{c}$ *is called a* weak flip column *if its first component is 0 and its Hamming weight equals to* $\lfloor \frac{\mathsf{M}}{2} \rfloor$ *or* $\lceil \frac{\mathsf{M}}{2} \rceil$*. The collection of all possible weak flip columns is called* weak flip candidate columns set *and is denoted by* $\mathcal{C}^{(\mathsf{M})}$*.*

We see that a weak flip column contains an almost equal number of zeros and ones. The restriction of the first component to be zero is based on the insight of Lemma 3.1. For the remainder of this work, we introduce the shorthand

$$\ell \triangleq \left\lceil \frac{\mathsf{M}}{2} \right\rceil . \tag{6.3}$$

Lemma 6.3 *The cardinality of a weak flip candidate columns set is*

$$\left| \mathcal{C}^{(\mathsf{M})} \right| = \binom{2\ell - 1}{\ell}. \tag{6.4}$$

Proof: If $\mathsf{M} = 2\ell$, then we have $\binom{2\ell-1}{\ell}$ possible choices, while if $\mathsf{M} = 2\ell - 1$, we have $\binom{2\ell-2}{\ell-1} + \binom{2\ell-2}{\ell} = \binom{2\ell-1}{\ell}$ choices. □

We are now ready to generalize Definition 6.1.

Definition 6.4 *A* weak flip code *is a codebook that is constructed only by weak flip columns.*

Concretely, for $\mathsf{M} = 3$ or $\mathsf{M} = 4$, we have the following.

Definition 6.5 *The* weak flip code of type (t_2, t_3) *for* $\mathsf{M} = 3$ *or* $\mathsf{M} = 4$ *codewords is defined by a codebook matrix* $\mathscr{C}_{t_2,t_3}^{(\mathsf{M},n)}$ *that consists of* $t_1 \triangleq n - t_2 - t_3$ *columns* $\mathbf{c}_1^{(\mathsf{M})}$*,* t_2 *columns* $\mathbf{c}_2^{(\mathsf{M})}$*, and* t_3 *columns* $\mathbf{c}_3^{(\mathsf{M})}$*, where*

$$\left\{ \mathbf{c}_1^{(3)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \mathbf{c}_2^{(3)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \mathbf{c}_3^{(3)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\} \tag{6.5}$$

or

$$\left\{ \mathbf{c}_1^{(4)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \mathbf{c}_2^{(4)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \mathbf{c}_3^{(4)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \right\}, \tag{6.6}$$

respectively. We often describe the weak flip code of type (t_2, t_3) *by its* code parameters

$$[t_1, t_2, t_3] \tag{6.7}$$

where t_1 can be computed from the blocklength n and the type (t_2, t_3) as $t_1 = n - t_2 - t_3$. Moreover, we use

$$\mathcal{D}_{t_2,t_3;m}^{(\mathsf{M},n)} \triangleq \{\mathbf{y} \colon g(\mathbf{y}) = m\} \tag{6.8}$$

to denote the decoding region of the mth codeword of $\mathscr{C}_{t_2,t_3}^{(\mathsf{M},n)}$.

An interesting subfamily of weak flip codes of type (t_2, t_3) for $\mathsf{M} = 3$ or $\mathsf{M} = 4$ is defined as follows.

Definition 6.6 *A* fair weak flip *code of type (t_2, t_3), $\mathscr{C}_{t_2,t_3}^{(\mathsf{M},n)}$, with $\mathsf{M} = 3$ or $\mathsf{M} = 4$ codewords satisfies that*

$$t_1 = t_2 = t_3. \tag{6.9}$$

Note that the fair weak flip code of type (t_2, t_3) is only defined provided that the blocklength satisfies $n \bmod 3 = 0$. In order to be able to provide convenient comparisons for every blocklength n, we define a *generalized fair weak flip code* for every n, $\mathscr{C}_{\lfloor \frac{n+1}{3} \rfloor, \lfloor \frac{n}{3} \rfloor}^{(\mathsf{M},n)}$, where

$$t_2 = \left\lfloor \frac{n+1}{3} \right\rfloor, \quad t_3 = \left\lfloor \frac{n}{3} \right\rfloor. \tag{6.10}$$

If $n \bmod 3 = 0$, the generalized fair weak flip code actually is a fair weak flip code.

The following lemma follows from the respective definitions in a straightforward manner. We therefore omit its proof.

Lemma 6.7 *The pairwise Hamming distance vector of a weak flip code of type (t_2, t_3) can be computed as follows:*

$$\begin{aligned}
\mathbf{d}^{(3,n)} &= (t_2 + t_3, t_1 + t_3, t_1 + t_2), \\
\mathbf{d}^{(4,n)} &= (t_2 + t_3, t_1 + t_3, t_1 + t_2, t_1 + t_2, t_1 + t_3, t_2 + t_3).
\end{aligned}$$

A similar definition can be given also for larger M, however, one needs to be aware that the number of weak flip candidate columns is increasing fast. For $\mathsf{M} = 5$ or $\mathsf{M} = 6$ we have ten weak flip candidate columns:

$$\left\{ \mathbf{c}_1^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \mathbf{c}_2^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \mathbf{c}_3^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \right.$$

$$\left. \mathbf{c}_4^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \mathbf{c}_5^{(5)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \mathbf{c}_6^{(5)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \mathbf{c}_7^{(5)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \right.$$

$$\mathbf{c}_8^{(5)} \triangleq \begin{pmatrix}0\\1\\1\\0\\0\end{pmatrix}, \mathbf{c}_9^{(5)} \triangleq \begin{pmatrix}0\\1\\1\\0\\1\end{pmatrix}, \mathbf{c}_{10}^{(5)} \triangleq \begin{pmatrix}0\\1\\1\\1\\0\end{pmatrix}\Bigg\}, \tag{6.11}$$

and

$$\Bigg\{\mathbf{c}_1^{(6)} \triangleq \begin{pmatrix}0\\0\\0\\1\\1\\1\end{pmatrix}, \mathbf{c}_2^{(6)} \triangleq \begin{pmatrix}0\\0\\1\\0\\1\\1\end{pmatrix}, \mathbf{c}_3^{(6)} \triangleq \begin{pmatrix}0\\0\\1\\1\\0\\1\end{pmatrix},$$

$$\mathbf{c}_4^{(6)} \triangleq \begin{pmatrix}0\\0\\1\\1\\1\\0\end{pmatrix}, \mathbf{c}_5^{(6)} \triangleq \begin{pmatrix}0\\1\\0\\0\\1\\1\end{pmatrix}, \mathbf{c}_6^{(6)} \triangleq \begin{pmatrix}0\\1\\0\\1\\0\\1\end{pmatrix}, \mathbf{c}_7^{(6)} \triangleq \begin{pmatrix}0\\1\\0\\1\\1\\0\end{pmatrix},$$

$$\mathbf{c}_8^{(6)} \triangleq \begin{pmatrix}0\\1\\1\\0\\0\\1\end{pmatrix}, \mathbf{c}_9^{(6)} \triangleq \begin{pmatrix}0\\1\\1\\0\\1\\0\end{pmatrix}, \mathbf{c}_{10}^{(6)} \triangleq \begin{pmatrix}0\\1\\1\\1\\0\\0\end{pmatrix}\Bigg\}, \tag{6.12}$$

respectively.

We will next introduce a generalized fair weak flip codes, as we will see in Section 6.1, possess particularly beautiful properties.

Definition 6.8 *A weak flip code is called* fair *if it is constructed by an equal number of all possible weak flip candidate columns in* $\mathcal{C}^{(\mathrm{M})}$*. Note that by definition the blocklength of a fair weak flip code is always a multiple of* $\binom{2\ell-1}{\ell}$*,* $\ell \geq 2$*.*

Fair weak flip codes have been used by Shannon *et al.* [17] for the derivation of error exponents, although the codes were not named at that time. Note that the error exponents are defined when the blocklength n goes to infinity, but in this work we consider finite n.

Related to the weak flip codes and the fair weak flip codes are the families of *Hadamard codes* [13, Ch. 2].

Definition 6.9 *For an even integer* n*, a (*normalized*)* Hadamard matrix H_n *of order* n *is an* $n \times n$ *matrix with entries* $+1$ *and* -1 *and with the first row and column being all* $+1$*, such that*

$$\mathsf{H}_n \mathsf{H}_n^{\mathsf{T}} = n\mathsf{I}_n, \tag{6.13}$$

if such a matrix exists. Here I_n *is the identity matrix of size* n*. If the entries* $+1$ *are replaced by* 0 *and the entries* -1 *by* 1*,* H_n *is changed into the* binary Hadamard matrix A_n*.*

Note that a necessary (but not sufficient) condition for the existence of H_n (and the corresponding A_n) is that n is a 1, 2 or multiple of 4 [13, Ch. 2].

Definition 6.10 *The binary Hadamard matrix* A_n *gives rise to three families of Hadamard codes:*

1. *The* $\left(n, n-1, \frac{n}{2}\right)$ Hadamard code $\mathscr{H}_{1,n}$ *consists of the rows of* A_n *with the first column deleted. The codewords in* $\mathscr{H}_{1,n}$ *that begin with* 0 *form the* $\left(\frac{n}{2}, n-2, \frac{n}{2}\right)$ Hadamard code $\mathscr{H}'_{1,n}$ *if the initial zero is deleted.*
2. *The* $\left(2n, n-1, \frac{n}{2}-1\right)$ Hadamard code $\mathscr{H}_{2,n}$ *consists of* $\mathscr{H}_{1,n}$ *together with the complements of all its codewords.*
3. *The* $\left(2n, n, \frac{n}{2}\right)$ Hadamard code $\mathscr{H}_{3,n}$ *consists of the rows of* A_n *and their complements.*

Further Hadamard codes can be created by an arbitrary combinations of the codebook matrices of different Hadamard codes.

Example 6.11 *Consider a* $(6, 10, 6)$ $\mathscr{H}'_{1,12}$ *code:*

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{6.14}$$

From this code, see the candidate columns (6.12) *for* $\mathsf{M} = 6$*, it is identical to the fair weak flip code for* $\mathsf{M} = 6$*. Since the fair weak flip code already used up all the possible weak flip candidate columns, hence, there is only one* $(6, 10, 6)$ $\mathscr{H}'_{1,12}$ *in column-wise respect.*

Example 6.12 *Consider an* $(8, 7, 4)$ $\mathscr{H}_{1,8}$ *code:*

$$\mathscr{H}^1_{1,8} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{6.15}$$

and the other $(8, 7, 4)$ $\mathscr{H}_{1,8}^2$ *code:*

$$\mathscr{H}_{1,8}^2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}. \tag{6.16}$$

From these codes, an $(8, 35, 20)$ *Hadamard code can be constructed by simply concatenating* $\mathscr{H}_{1,8}^1$ *five times, or concatenating* $\mathscr{H}_{1,8}^1$ *three times and* $\mathscr{H}_{1,8}^2$ *two times.*

Note that since the rows of H_n are orthogonal, any two rows of A_n agree in $\frac{1}{2}n$ places and differ in $\frac{1}{2}n$ places, i.e., they have a Hamming distance $\frac{1}{2}n$. Moreover, by definition the first row of a binary Hadamard matrix is the all-zero row. Hence, we see that all Hadamard codes are weak flip codes, i.e., the family of weak flip codes is a superset of Hadamard codes.

On the other hand, every Hadamard code of parameters (M, n), for which fair weak flip codes exist, is not necessarily equivalent to a fair weak flip code. We also would like to remark that the Hadamard codes rely on the existence of Hadamard matrices. So in general, it is very difficult to predict whether for a given pair (M, n), a Hadamard code will exist or not. This is in stark contrast to weak flip codes (which exist for all M and n) and fair weak flip codes (which exist for all M and all n being a multiple of $\binom{2\ell-1}{\ell}$).

Example 6.13 *We continue with Example 6.12 and note that the* $(8, 35, 20)$ *Hadamard code that is constructed by five repetitions of the matrix given in* (6.15) *is actually not a fair weak flip code, since we have to use up all possible weak flip candidate columns to get a* $(8, 35, 20)$ *fair weak flip code.*

Note that two Hadamard matrices can be *equivalent* if one can be obtained from the other by permuting rows and columns and multiplying rows and columns by -1. In other words, Hadamard codes can actually be constructed from weak candidate columns. This also follows directly from the already mentioned fact that Hadamard codes are weak flip codes.

6.1 Characteristics of Weak Flip Codes

In conventional coding theory, most results are restricted to so called *linear codes* that possess very powerful algebraic properties. For the following definitions and proofs see, e.g., [12], [13].

Definition 6.14 *Let* $\mathsf{M} = 2^k$*, where* $k \in \mathbb{N}$*. The binary code* $\mathscr{C}_{\text{lin}}^{(\mathsf{M},n)}$ *is* linear *if its codewords span a* k*-dimensional subspace of* $\{0,1\}^n$*.*

One of the most important property of a linear code is as follows.

Proposition 6.15 *Let* $\mathscr{C}_{\text{lin}}$ *be linear and let* $\mathbf{x}_m \in \mathscr{C}_{\text{lin}}$ *be given. Then the code that we obtain by adding* $\mathbf{x}_m$ *to each codeword of* $\mathscr{C}_{\text{lin}}$ *is equal to* $\mathscr{C}_{\text{lin}}$*.*

Another property concerns the column weights.

Proposition 6.16 *If an* (M, n) *binary code is linear, then each column of its codebook matrix has Hamming weight* $\frac{\mathsf{M}}{2}$*, i.e., the code is a weak flip code.*

Hence, linear codes are weak flip codes. Note, however, that linear codes only exist if $\mathsf{M} = 2^k$, where $k \in \mathbb{N}$, while weak flip codes are defined for any M. Also note that the converse of Proposition 6.16 does not hold, i.e., even if $\mathsf{M} = 2^k$ for some $k \in \mathbb{N}$, a weak flip code $\mathscr{C}^{(\mathsf{M},n)}$ is not necessarily linear. It is not even the case that a fair weak flip code for $\mathsf{M} = 2^k$ is necessarily linear!

Now the question arises as to which of the many powerful algebraic properties of linear codes are retained in weak flip codes.

Theorem 6.17 *Consider a weak flip code* $\mathscr{C}^{(\mathsf{M},n)}$ *and fix some codeword* $\mathbf{x}_m \in \mathscr{C}^{(\mathsf{M},n)}$*. If we add this codeword to all codewords in* $\mathscr{C}^{(\mathsf{M},n)}$*, then the resulting code* $\tilde{\mathscr{C}}^{(\mathsf{M},n)} \triangleq \left\{\mathbf{x}_m \oplus \mathbf{x} \,\middle|\, \forall\, \mathbf{x} \in \mathscr{C}^{(\mathsf{M},n)}\right\}$ *is still a weak flip code, however, it is not necessarily the same one.*

Proof: Let $\mathscr{C}^{(\mathsf{M},n)}$ be according to Definition 6.4. We have to prove that

$$\begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_{\mathsf{M}} \end{pmatrix} \oplus \begin{pmatrix} \mathbf{x}_m \\ \mathbf{x}_m \\ \vdots \\ \mathbf{x}_m \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1 \oplus \mathbf{x}_m \\ \vdots \\ \mathbf{x}_m \oplus \mathbf{x}_m = \mathbf{0} \\ \vdots \\ \mathbf{x}_{\mathsf{M}} \oplus \mathbf{x}_m \end{pmatrix} \triangleq \tilde{\mathscr{C}}^{(\mathsf{M},n)} \tag{6.17}$$

is a weak flip code. Let $\mathbf{c}_i$ denote the column vectors of $\mathscr{C}^{(\mathsf{M},n)}$. Then $\tilde{\mathscr{C}}^{(\mathsf{M},n)}$ has the column vectors

$$\tilde{\mathbf{c}}_i = \begin{cases} \mathbf{c}_i & \text{if } x_{m,i} = 0, \\ \bar{\mathbf{c}}_i & \text{if } x_{m,i} = 1, \end{cases} \tag{6.18}$$

for $1 \leq i \leq n$. Since $\mathbf{c}_i$ is a weak flip column, either $w_{\text{H}}(\mathbf{c}_i) = \left\lfloor \frac{\mathsf{M}}{2} \right\rfloor$ and therefore $w_{\text{H}}(\bar{\mathbf{c}}_i) = \left\lceil \frac{\mathsf{M}}{2} \right\rceil$, or $w_{\text{H}}(\mathbf{c}_i) = \left\lceil \frac{\mathsf{M}}{2} \right\rceil$ and therefore $w_{\text{H}}(\bar{\mathbf{c}}_i) = \left\lfloor \frac{\mathsf{M}}{2} \right\rfloor$. So we only need to interchange the first codeword of $\tilde{\mathscr{C}}$ and the all-zero codeword in the mth row in $\tilde{\mathscr{C}}$ (which is always possible, see discussion after Definition 2.7), and we see that $\tilde{\mathscr{C}}$ is also a weak flip code. □

Theorem 6.17 is a beautiful property of weak flip codes; however, it still represents a considerable weakening of the powerful property of linear codes given in Proposition 6.15. This can be fixed by considering the subfamily of fair weak flip codes.

Theorem 6.18 (Quasi-Linear Codes) *Let $\mathscr{C}$ be a **fair** weak flip code and let $\mathbf{x}_m \in \mathscr{C}$ be given. Then the code $\tilde{\mathscr{C}} = \{\mathbf{x}_m \oplus \mathbf{x} \,|\, \forall\, \mathbf{x} \in \mathscr{C}^{(\mathrm{M},n)}\}$ is equivalent to $\mathscr{C}$.*

Proof: We have already seen in Theorem 6.17 that adding a codeword will result in a weak flip code again. In the case of a fair weak flip code, however, all possible candidate columns will show up again with the same equal frequency. It only remains to rearrange some rows and columns. □

If we recall Proposition 6.16 and the discussion after it, we realize that the definition of the quasi-linear fair weak flip code is a considerable enlargement of the set of codes having the property given in Theorem 6.18.

The following corollary is a direct consequence of Theorem 6.18.

Corollary 6.19 *The Hamming weights of each codeword of a fair weak flip code are all identical except the all-zero codeword $\mathbf{x}_1$. In other words, if we let $w_{\mathrm{H}}(\cdot)$ be the Hamming weight function, then*

$$w_{\mathrm{H}}(\mathbf{x}_2) = w_{\mathrm{H}}(\mathbf{x}_3) = \cdots = w_{\mathrm{H}}(\mathbf{x}_{\mathrm{M}}). \tag{6.19}$$

Before we next investigate the minimum Hamming distance for the quasi-linear fair weak flip codes, we quickly recall an important bound that holds for any (M, n, d) code.

Lemma 6.20 (Plotkin Bound [13]) *The minimum distance of an (M, n) binary code $\mathscr{C}^{(\mathrm{M},n)}$ always satisfies*

$$d_{\min}\big(\mathscr{C}^{(\mathrm{M},n)}\big) \leq \begin{cases} \frac{n \cdot \frac{\mathrm{M}}{2}}{\mathrm{M}-1} & \mathrm{M} \text{ even}, \\ \frac{n \cdot \frac{\mathrm{M}+1}{2}}{\mathrm{M}} & \mathrm{M} \text{ odd}. \end{cases} \tag{6.20}$$

Proof: We show a quick proof. We sum the Hamming distance over all possible pairs of two codewords apart from the codeword with itself:

$$\mathrm{M}(\mathrm{M}-1) \cdot d_{\min}(\mathscr{C}^{(\mathrm{M},n)}) \leq \sum_{\mathbf{u} \in \mathscr{C}^{(\mathrm{M},n)}} \sum_{\substack{\mathbf{v} \in \mathscr{C}^{(\mathrm{M},n)} \\ \mathbf{v} \neq \mathbf{u}}} d_{\mathrm{H}}(\mathbf{u}, \mathbf{v}) \tag{6.21}$$

$$= \sum_{j=1}^{n} 2b_j \cdot (\mathrm{M} - b_j) \tag{6.22}$$

$$\leq \begin{cases} n \cdot \frac{\mathrm{M}^2}{2} & \text{if M even (achieved if } b_j = \mathrm{M}/2), \\ n \cdot \frac{\mathrm{M}^2-1}{2} & \text{if M odd (achieved if } b_j = (\mathrm{M} \pm 1)/2). \end{cases} \tag{6.23}$$

Here in (A.33) we rearrange the order of summation: instead of summing over all codewords (rows), we approach the problem column-wise and assume that the jth column of $\mathscr{C}^{(\mathrm{M},n)}$ contains b_j zeros and $\mathrm{M} - b_j$ ones: then this column contributes $2b_j(\mathrm{M} - b_j)$ to the sum. □

Note that from the proof of Lemma 6.20 we can see that a necessary condition for a codebook to meet the Plotkin-bound is that the codebook is composed by weak flip

candidate columns. Furthermore, Levenshtein [13, Ch. 2] proved that the Plotkin bound can be achieved, provided that Hadamard matrices exist.

Theorem 6.21 *Fix some* M *and a blocklength* n *with* $n \bmod \binom{2\ell-1}{\ell} = 0$*. Then a fair weak flip code* $\mathscr{C}^{(\mathsf{M},n)}$ *achieves the largest minimum Hamming distance among all codes of given blocklength and satisfies*

$$d_{\min}\big(\mathscr{C}^{(\mathsf{M},n)}\big) = \frac{n \cdot \ell}{2\ell - 1}. \tag{6.24}$$

Proof: For $\mathsf{M} = 2\ell$, we know that by definition the Hamming weight of each column of the codebook matrix is equal to ℓ. Hence, when changing the sum from column-wise to row-wise, where we can ignore the first row of zero weight (from the all-zero codeword $\mathbf{x}_1$), we get

$$n \cdot \ell = \sum_{j=1}^{n} w_{\mathrm{H}}(\mathbf{c}_j) = \sum_{m=2}^{2\ell} w_{\mathrm{H}}(\mathbf{x}_m) \tag{6.25}$$

$$= \sum_{m=2}^{2\ell} d_{\min}\big(\mathscr{C}^{(\mathsf{M},n)}\big) \tag{6.26}$$

$$= (2\ell - 1) \cdot d_{\min}\big(\mathscr{C}^{(\mathsf{M},n)}\big). \tag{6.27}$$

Here, (B.42) follows from Theorem 6.18 and from Corollary 6.19. For $\mathsf{M} = 2\ell - 1$, the Hamming distance remains the same due to the fair construction.

It remains to show that a fair weak flip code achieves the largest minimum Hamming distance among all codes of given blocklength. From Corollary 6.19 we know that (apart from the all-zero codeword) all codewords of a fair weak flip code have the same Hamming weight. So, if we flip an arbitrary 1 in the codebook matrix to become a 0, then the corresponding codeword has a decreased Hamming weight and is therefore closer to the all-zero codeword. If we flip an arbitrary 0 to become a 1, then the corresponding codeword is closer to some other codeword that already has a 1 in this position. Hence, in both cases we have reduced the minimum Hamming distance. Finally, based on the concept of looking at the code in column-wise, it can be seen that whenever we change more than one bit, we either get back to a fair weak flip code or to another code who is worse. □

Chapter 7

Previous Work

7.1 SGB Bounds on the Average Error Probability

In [17], Shannon, Gallager, and Berlekamp derive upper and lower bounds on the average error probability of a given code used on a DMC. We next quickly review their results.

Definition 7.1 *For* $0 < s < 1$ *we define*

$$\mu_{\alpha,\beta}(s) \triangleq \ln \sum_{y} P_{Y|X}(y|\alpha)^{1-s} P_{Y|X}(y|\beta)^{s}. \tag{7.1}$$

Therefore, the generalized $\mu(s)$ *for blocklength* n *between* $\mathbf{x}_m$ *and* $\mathbf{x}_{m'}$ *can be defined and expressed in terms of* (7.1) *by*

$$\mu(s) \triangleq \ln \sum_{\mathbf{y}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m)^{1-s} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_{m'})^{s} = n \sum_{\alpha} \sum_{\beta} q_{\alpha,\beta}(m,m') \mu_{\alpha,\beta}(s), \tag{7.2}$$

and the discrepancy $\mathsf{D}^{(\mathrm{DMC})}(m,m')$ *between* $\mathbf{x}_m$ *and* $\mathbf{x}_{m'}$ *is defined as*

$$\mathsf{D}^{(\mathrm{DMC})}(m,m') \triangleq -\min_{0 \le s \le 1} \sum_{\alpha} \sum_{\beta} q_{\alpha,\beta}(m,m') \mu_{\alpha,\beta}(s) \tag{7.3}$$

with $q_{\alpha,\beta}(m,m')$ *given in Def. 2.9.*

Note that the discrepancy is a generalization of the Hamming distance, however, it depends strongly on the channel cross-over probabilities. We use a superscript "(DMC)" to indicate the channel which the discrepancy refers to.

Definition 7.2 *The* minimum discrepancy $\mathsf{D}^{(\mathrm{DMC})}_{\min}(\mathscr{C}^{(\mathrm{M},n)})$ *for a codebook is the minimum value of* $\mathsf{D}^{(\mathrm{DMC})}(m,m')$ *over all pairs of codewords. The* maximum minimum discrepancy *is the maximum value of* $\mathsf{D}^{(\mathrm{DMC})}_{\min}(\mathscr{C}^{(\mathrm{M},n)})$ *over all possible* $\mathscr{C}^{(\mathrm{M},n)}$ *codebooks:* $\max_{\mathscr{C}^{(\mathrm{M},n)}} \mathsf{D}^{(\mathrm{DMC})}_{\min}(\mathscr{C}^{(\mathrm{M},n)})$.

Theorem 7.3 (Lower Bounds to Conditional Error Probability [17]) *If* $\mathbf{x}_m$ *and* $\mathbf{x}_{m'}$ *are pair of codewords in a code of blocklength* n*, then either*

$$\lambda_m > \frac{1}{4}\exp -n\left[\mathsf{D}^{(\mathrm{DMC})}(m,m') + \sqrt{\frac{2}{n}}\ln\left(1/P_{\min}\right)\right] \tag{7.4}$$

or

$$\lambda_{m'} > \frac{1}{4}\exp -n\left[\mathsf{D}^{(\mathrm{DMC})}(m,m') + \sqrt{\frac{2}{n}}\ln\left(1/P_{\min}\right)\right], \tag{7.5}$$

where $P_{\min}$ *is the smallest nonzero transition probability for the channel.*

Conversely, one can also show that

$$\lambda_m \le (\mathsf{M}-1)\exp -n\left(\mathsf{D}_{\min}^{(\mathrm{DMC})}\big(\mathscr{C}^{(\mathsf{M},n)}\big)\right), \quad \text{for all } m. \tag{7.6}$$

Theorem 7.4 (SGB Bounds on Average Error Probability [17]) *For an arbitrary DMC, the average error probability* $P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big)$ *of a given code* $\mathscr{C}^{(\mathsf{M},n)}$ *with* M *codewords and blocklength* n *is upper- and lower-bounded as follows:*

$$\begin{aligned}\frac{1}{4\mathsf{M}}e^{-n\left(\mathsf{D}_{\min}^{(\mathrm{DMC})}\left(\mathscr{C}^{(\mathsf{M},n)}\right)+\sqrt{\frac{2}{n}}\log\frac{1}{P_{\min}}\right)}&\\ \le P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) \le (\mathsf{M}-1)e^{-n\mathsf{D}_{\min}^{(\mathrm{DMC})}\left(\mathscr{C}^{(\mathsf{M},n)}\right)}&\end{aligned} \tag{7.7}$$

where $P_{\min}$ *denotes the smallest nonzero transition probability of the channel.*

Note that these bounds are specific to a given code design (via $\mathsf{D}_{\min}^{(\mathrm{DMC})}$). Therefore, the upper bound is a generally valid upper bound on the optimal performance, while the lower bound only holds in general if we apply it to the optimal code or to a suboptimal code that achieves the optimal $\mathsf{D}_{\min}$.

The bounds (7.7) are tight enough to derive the *error exponent* of the DMC (for a fixed number M of codewords).

Theorem 7.5 ([17]) *The error exponent of a DMC for a fixed number* M *of codewords*

$$\mathsf{E}_{\mathsf{M}} \triangleq \varlimsup_{n\to\infty}\max_{\mathscr{C}^{(\mathsf{M},n)}}\left\{-\frac{1}{n}\log P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big)\right\} \tag{7.8}$$

is given as

$$\mathsf{E}_{\mathsf{M}} = \lim_{n\to\infty}\max_{\mathscr{C}^{(\mathsf{M},n)}}\mathsf{D}_{\min}^{(\mathrm{DMC})}\big(\mathscr{C}^{(\mathsf{M},n)}\big). \tag{7.9}$$

Unfortunately, in general the evaluation of the error exponent is very difficult. For some cases, however, it can be done. For example, for $\mathsf{M}=2$, we have

$$\mathsf{E}_2 = \max_{\mathscr{C}^{(2,n)}}\mathsf{D}_{\min}^{(\mathrm{DMC})}\big(\mathscr{C}^{(2,n)}\big) = \max_{\alpha,\beta}\left\{-\min_{0\le s\le 1}\mu_{\alpha,\beta}(s)\right\}. \tag{7.10}$$

Also for the class of so-called *pairwise reversible channels*, the calculation of the error exponent turns out to be uncomplicated.

Definition 7.6 *A* pairwise reversible channel *is a DMC that has* $\mu'_{\alpha,\beta}(\frac{1}{2}) = 0$ *for any inputs* α, β.

Clearly, the BSC and BEC are pairwise reversible channels.

Note that it is easy to compute the pairwise discrepancy of a linear code on a pairwise reversible channel, so linear codes are quite suitable for computing (7.7).

Theorem 7.7 ([17]) *For* pairwise reversible channels *with* $\mathsf{M} > 2$,

$$\mathsf{E}_{\mathsf{M}} = \frac{1}{\mathsf{M}(\mathsf{M}-1)} \max_{\substack{\mathsf{M}_x \text{ s.t.} \\ \sum_x \mathsf{M}_x = \mathsf{M}}} \sum_{\substack{\text{all input} \\ \text{letters } x}} \sum_{\substack{\text{all input} \\ \text{letters } x'}} \mathsf{M}_x \mathsf{M}_{x'} \cdot \left(-\ln \sum_y \sqrt{P_{Y|X}(y|x) P_{Y|X}(y|x')} \right) \tag{7.11}$$

where M_x *denotes the number of times the channel input letter* x *occurs in a column. Moreover,* E_{M} *is achieved by* fair weak flip codes.[6]

We would like to emphasize that while Shannon *et al.* proved that fair weak flip codes achieve the error exponent, they did not investigate the error performance of fair weak flip codes for finite n. As we will show later, fair weak flip might be strictly suboptimal codes for finite n (see also [18]).

7.2 Gallager Bound

Another famous bound is by Gallager [6].

Theorem 7.8 ([6]) *For an arbitrary DMC, there exists a code* $\mathscr{C}^{(\mathsf{M},n)}$ *with* $\mathsf{M} = \lfloor e^{n\mathsf{R}} \rfloor$ *such that*

$$P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) \leq e^{-n\mathsf{E}_{\mathrm{G}}(\mathsf{R})} \tag{7.12}$$

where $\mathsf{E}_{\mathrm{G}}(\cdot)$ *is the Gallager exponent and is given by*

$$\mathsf{E}_{\mathrm{G}}(\mathsf{R}) = \max_{Q(\cdot)} \max_{0 \leq \rho \leq 1} \big\{ \mathsf{E}_0(\rho, Q) - \rho \mathsf{R} \big\} \tag{7.13}$$

with

$$\mathsf{E}_0(\rho, Q) \triangleq -\log \left(\sum_y \left(\sum_x Q(x) P_{Y|X}(y|x)^{\frac{1}{1+\rho}} \right)^{1+\rho} \right). \tag{7.14}$$

[6] While throughout we only consider binary inputs and $\mathsf{M} = 3$ or $\mathsf{M} = 4$, the definitions of our fair weak flip codes can be generalized to nonbinary inputs and larger M. Also these generalized fair weak flip codes will achieve the corresponding error exponents. Note that Shannon *et al.* did not actually name their exponent-achieving codes.

7.3 PPV Bounds for the BSC

In [2], Polyanskiy, Poor, and Verdú present upper and lower bounds on the optimal average error probability for finite blocklength for the BSC. The upper bound is based on *random coding*.

Theorem 7.9 (PPV Upper Bound [19, Theorem 2], [2, Theorem 32]) *If the codebook $\mathbb{C}^{(\mathsf{M},n)}$ is created at random based on a uniform distribution, the expected average error probability (averaged over all codewords and all codebooks) satisfies*

$$\mathsf{E}\left[P_{\mathrm{e}}\left(\mathbb{C}^{(\mathsf{M},n)}\right)\right] = 1 - 2^{n-n\mathsf{M}} \sum_{i=0}^{n} \binom{n}{i} \epsilon^i (1-\epsilon)^{n-i} \cdot \left(\sum_{m=0}^{\mathsf{M}-1} \frac{1}{m+1} \binom{\mathsf{M}-1}{m} \binom{n}{i}^m \left(\sum_{j=i+1}^{n} \binom{n}{j} \right)^{\mathsf{M}-1-m} \right). \tag{7.15}$$

Note that there must exist a codebook whose average error probability achieves (7.15), so Theorem 7.9 provides a general achievable upper bound, although we do not know its concrete code structure.

Polyanskiy, Poor, and Verdú also provide a new general converse for the average error probability: the so-called *meta-converse*, which is based on binary hypothesis testing. For a BSC, the meta-converse lower bound happens to be equivalent to Gallager's sphere-packing bound.

Theorem 7.10 (PPV Lower Bound [6, p. 163, Eq. (5.8.19)], [2, Theorem 35]) *Any codebook $\mathscr{C}^{(\mathsf{M},n)}$ satisfies*

$$P_{\mathrm{e}}\left(\mathscr{C}^{(\mathsf{M},n)}\right) \geq \left(\binom{n}{\mathsf{N}} - \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \mathsf{A}_{m,\mathsf{N}} \right) \epsilon^{\mathsf{N}} (1-\epsilon)^{n-\mathsf{N}} + \sum_{j=\mathsf{N}+1}^{n} \binom{n}{j} \epsilon^j (1-\epsilon)^{n-j} \tag{7.16}$$

where for $m \in \{1, \ldots, \mathsf{M}\}$ and for $j \in \{1, \ldots, \mathsf{N}-1, \mathsf{N}+1, \ldots, n\}$

$$\mathsf{A}_{m,j} = \begin{cases} \binom{n}{j} & 0 \leq j \leq \mathsf{N}-1 \\ 0 & \mathsf{N}+1 \leq j \leq n \end{cases} \tag{7.17}$$

and where the positive integer N and coefficients $\mathsf{A}_{m,\mathsf{N}}$ are chosen such that

$$\mathsf{M} \sum_{j=0}^{\mathsf{N}-1} \mathsf{A}_{m,j} + \sum_{m=1}^{\mathsf{M}} \mathsf{A}_{m,\mathsf{N}} = 2^n \tag{7.18}$$

$$0 < \sum_{m=1}^{\mathsf{M}} \mathsf{A}_{m,\mathsf{N}} \leq \mathsf{M} \binom{n}{\mathsf{N}}. \tag{7.19}$$

7.4 PPV Bounds for the BEC

In [2], Polyanskiy, Poor, and Verdú present upper and lower bounds on the optimal average error probability for finite blocklength for the BEC. The upper bound is based on *random coding*.

Theorem 7.11 *For the BEC with crossover probability δ, the average error probability for an random code is given by*

$$\begin{aligned}&\mathsf{E}\left[P_{\mathrm{e}}\big(\mathbb{C}^{(\mathsf{M},n)}\big)\right]\\&\quad=1-\sum_{j=0}^{n}\binom{n}{j}(1-\delta)^{j}\delta^{n-j}\sum_{\ell=0}^{\mathsf{M}-1}\frac{1}{\ell+1}\binom{\mathsf{M}-1}{\ell}(2^{-j})^{\ell}(1-2^{-j})^{\mathsf{M}-1-\ell}.\end{aligned}\tag{7.20}$$

Note that there must exist a codebook whose average error probability achieves (C.16), so Theorem 7.11 provides a general achievable upper bound, although we do not know its concrete code structure.

Polyanskiy, Poor, and Verdú also provide a new general converse for the average error probability for a BEC.

Theorem 7.12 *For the BEC with erasure probability δ, the average error probability of a $\mathscr{C}^{(\mathsf{M},n)}$ code satisfies*

$$P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big)\geq\sum_{\ell=\lfloor n-\log_2\mathsf{M}\rfloor+1}^{n}\binom{n}{\ell}\delta^{\ell}(1-\delta)^{n-\ell}\left(1-\frac{2^{n-\ell}}{\mathsf{M}}\right).\tag{7.21}$$

Note that this lower bound is not derived by the method: meta-converse, it is from other technique.

Chapter 8

Analysis of the BAC

We start with results that hold for the general BAC. In this section we will restrict ourselves to two codewords $\mathsf{M} = 2$. One can show that the BAC is a pairwise reversible channel, however, in this analysis we do not focus on its bounds on its performance, but put a special emphasis on the optimal code design.

8.1 Optimal Codes

Theorem 8.1 *Consider a BAC and a blocklength n. Then, irrespective of the channel parameters ϵ_0 and ϵ_1, there exists a choice of t, $0 \leq t \leq \lfloor \frac{n}{2} \rfloor$, such that the flip code of type t, $\mathscr{C}_t^{(2,n)}$, is optimal in the sense that it minimizes the average error probability.*

Proof: Consider an arbitrary code with $\mathsf{M} = 2$ codewords and a blocklength $n+j$, and assume that this code is not a flip code, but that it has a number j of positions where both codewords have the same symbol. An optimal decoder will ignore these j positions completely. Hence, the performance of this code will be identical to a flip code of length n. Now, change this code in the j positions with identical symbol such that the code becomes a flip code. If we use a suboptimal decoder that ignores these j positions we still keep the same performance. However, an ML decoder can potentially improve the performance, i.e., we have

$$P_{\mathrm{e}}\big(\mathscr{C}_{\text{not flip}}^{(\mathsf{M},n+j)}\big)_{\text{ML decoder}} = P_{\mathrm{e}}\big(\mathscr{C}_{\text{flip}}^{(\mathsf{M},n+j)}\big)_{\text{suboptimal decoder}} \tag{8.1}$$

$$\geq P_{\mathrm{e}}\big(\mathscr{C}_{\text{flip}}^{(\mathsf{M},n+j)}\big)_{\text{ML decoder}}. \tag{8.2}$$

An alternative proof is shown in Appendix A.3. While this proof is more elaborate, it turns out to be very useful for the derivation of Theorem 8.3. □

This result is intuitively very pleasing because it seems to be a rather bad choice to have two codewords with the same symbol in a particular position, i.e., $\mathbf{x}_{1,j} = \mathbf{x}_{2,j} = 0$ in the same position j. However, note that the theorem does not exclude the possibility that another code might exist that also is optimal and that has an identical symbol in both codewords at a given position.

We would like to point out that the exact choice of t is not obvious and depends strongly on n, ϵ_0, and ϵ_1. As an example, the optimal choices of t are shown in Fig. 8.6 for $n = 7$. We see that depending on the channel parameters, the optimal value of t changes.

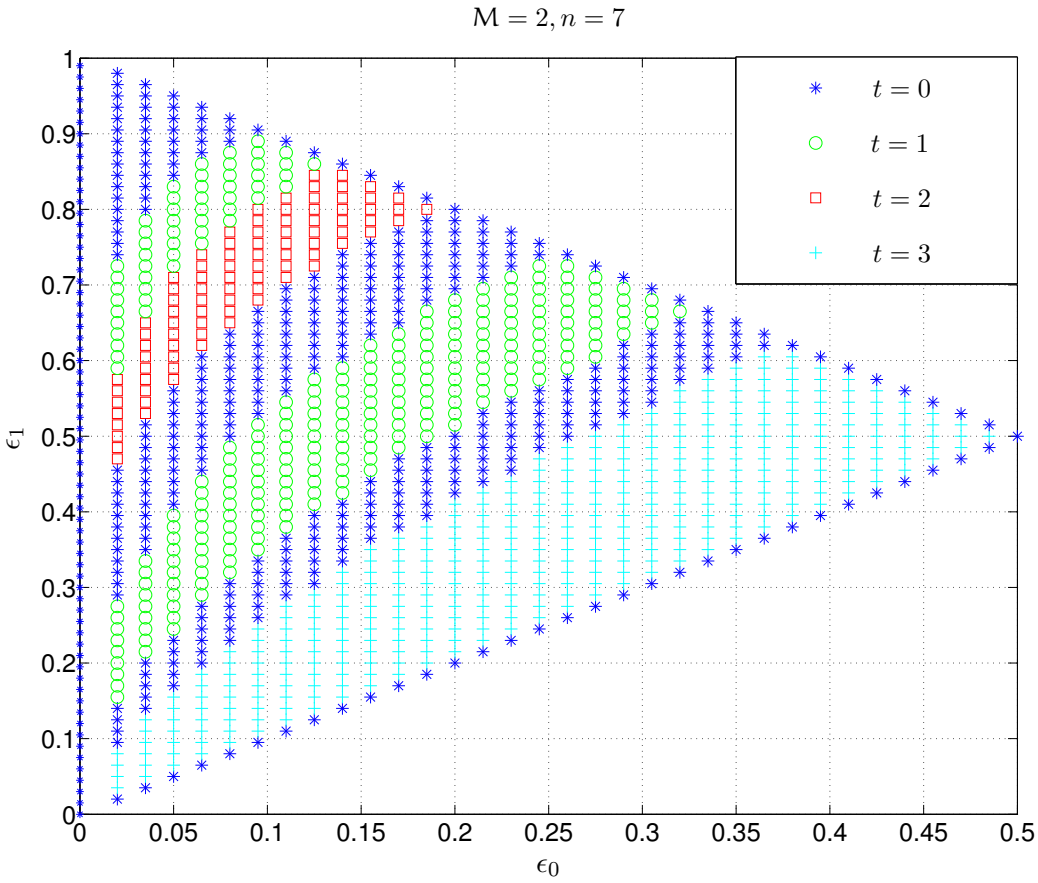


Figure 8.6: Optimal codebooks on a BAC: the optimal choice of the parameter t for different values of ϵ_0 and ϵ_1 for a fixed blocklength $n = 7$.

Note that for a completely noisy channel ($\epsilon_1 = 1 - \epsilon_0$), the choice of t is irrelevant since the probability of error is $\frac{1}{2}$ for any code. Moreover, in Theorem 9.1 it will be shown that the flip codes of type 0 is optimal on the ZC; and in Theorem 10.1 it will be shown that the flip codes are optimal on the BSC for any choice of t. We defer the exact treatment of the ZC and the BSC to Chapter 9 and 10, respectively.

8.2 The Optimal Decision Rule for Flip Codes

Having only two codewords, the ML decision rule can be expressed using the *log-likelihood ratio (LLR)*. For the flip code of type t, $\mathscr{C}_t^{(2,n)}$ (see Def. 6.1), the LLR is given as

$$\log\left(\frac{P_{Y|X}^n(\mathbf{y}|\mathbf{x}_1)}{P_{Y|X}^n(\mathbf{y}|\mathbf{x}_2)}\right)$$

$$
\begin{aligned}
&= \log\left(\frac{\left(\frac{\epsilon_0}{1-\epsilon_0}\right)^{d_{01}(\mathbf{x}_1,\mathbf{y})}\left(\frac{\epsilon_1}{1-\epsilon_0}\right)^{d_{10}(\mathbf{x}_1,\mathbf{y})}}{\left(\frac{\epsilon_0}{1-\epsilon_0}\right)^{t-d_{10}(\mathbf{x}_1,\mathbf{y})}\left(\frac{\epsilon_1}{1-\epsilon_0}\right)^{n-t-d_{01}(\mathbf{x}_1,\mathbf{y})}}\right. \\
&\qquad \left. \cdot \frac{\left(\frac{1-\epsilon_1}{1-\epsilon_0}\right)^{t-d_{10}(\mathbf{x}_1,\mathbf{y})}}{\left(\frac{1-\epsilon_1}{1-\epsilon_0}\right)^{d_{01}(\mathbf{x}_1,\mathbf{y})}}\right) \qquad (8.3)\\
&= \big(t - d_{01}(\mathbf{x}_1,\mathbf{y}) - d_{10}(\mathbf{x}_1,\mathbf{y})\big)\log\left(\frac{1-\epsilon_1}{\epsilon_0}\right) \\
&\quad + \big(n - t - d_{01}(\mathbf{x}_1,\mathbf{y}) - d_{10}(\mathbf{x}_1,\mathbf{y})\big)\log\left(\frac{1-\epsilon_0}{\epsilon_1}\right) \qquad (8.4)\\
&= (t-d)\log\left(\frac{1-\epsilon_1}{\epsilon_0}\right) + (n-t-d)\log\left(\frac{1-\epsilon_0}{\epsilon_1}\right) \qquad (8.5)\\
&\triangleq \mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d) \qquad (8.6)
\end{aligned}
$$

where we have defined

$$d \triangleq d_{01}(\mathbf{x}_1,\mathbf{y}) + d_{10}(\mathbf{x}_1,\mathbf{y}) = d_{\mathrm{H}}(\mathbf{x}_1,\mathbf{y}) \qquad (8.7)$$

to be the Hamming distance of the received sequence to the *first* codeword.

Hence we now express the ML decision rule for the flip code of type t as

$$\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d) \begin{cases} \geq 0 & \Longrightarrow\ g(\mathbf{y}) = 1, \\ < 0 & \Longrightarrow\ g(\mathbf{y}) = 2. \end{cases} \qquad (8.8)$$

Recall that ϵ_0 and ϵ_1 are parameters describing the channel (BAC), t and n describe the codebook (flip code $\mathscr{C}_t^{(2,n)}$), and $0 \leq d \leq n$ describes the received vector $\mathbf{y}$ (with respect to the first codeword). As an example, Fig. 8.7 depicts the log-likelihood ratio $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$ as a function of ϵ_0 (with $\epsilon_1 = 1-2\epsilon_0$) for the flip code $\mathscr{C}_1^{(2,n)}$ in the cases of $n=6$ and $n=7$. We see that for some integer ℓ, $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$ is always larger than 0 for $d \leq \ell$ and smaller than 0 for $d > \ell$.

More properties of $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$ are summarized in Appendix A.2.

From Prop. A.1 in Appendix A.2, it follows directly that the ML decision rule for a flip code is a *threshold rule*, i.e., there exists an integer ℓ such that for $d \leq \ell$, the received vector $\mathbf{y}$ is decoded to $\mathbf{x}_1$, and for $d > \ell$, $\mathbf{y}$ is decoded to $\mathbf{x}_2$.

Corollary 8.2 (Threshold Rule) *For every flip code $\mathscr{C}_t^{(2,n)}$ and every BAC $(\epsilon_0,\epsilon_1) \in \Omega$, there exists a* threshold ℓ, $t \leq \ell \leq \lfloor \frac{n-1}{2} \rfloor$, *such that the ML decision rule can be stated as*

$$g(\mathbf{y}) = \begin{cases} 1 & \text{if } 0 \leq d \leq \ell \\ 2 & \text{if } \ell+1 \leq d \leq n. \end{cases} \qquad (8.9)$$

The threshold ℓ depends on (ϵ_0,ϵ_1). The region of channel parameters with identical threshold ℓ (for given n and t) is then defined as follows:

$$\begin{aligned}\Omega_{\ell,t}^{(n)} \triangleq \big\{(\epsilon_0,\epsilon_1)\colon\ &\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,\ell) \geq 0 \text{ and} \\ &\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,\ell+1) \leq 0\big\}. \qquad (8.10)\end{aligned}$$

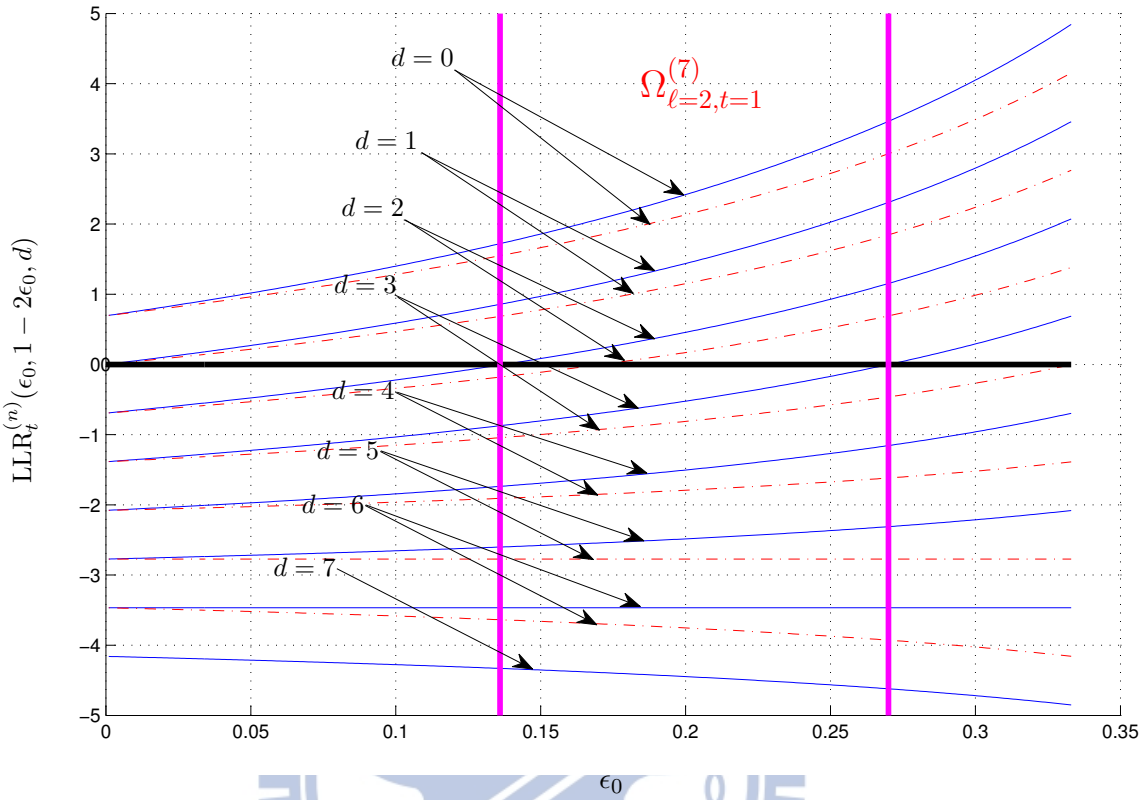


Figure 8.7: The log-likelihood ratio $\mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1 = 1 - 2\epsilon_0, d)$ for $\mathscr{C}_1^{(2,n)}$ (i.e., $t = 1$) as a function of ϵ_0 for different values of d. The solid blue lines correspond to $n = 7$, the dashed red lines to $n = 6$. Observe that for $n = 7$ and $\epsilon_0 \in [0.136, 0.270]$ (i.e., the region between the two vertical purple lines), the threshold for the optimal ML decision rule is $\ell = 2$, see Cor. 8.2.

8.3 Best Codes for a Fixed Decision Rule

Our original goal was to find the optimal code for a given channel (ϵ_0, ϵ_1). We have shown that this is equivalent in finding an optimal t. Unfortunately, this search is difficult because the borders between the regions of different optimal t (see, e.g., Fig. 8.6) are defined by the combined influences of two different forces: when varying (ϵ_0, ϵ_1), either the optimal code $\mathscr{C}_t^{(2,n)}$ changes, but the optimal threshold ℓ remains the same, or the optimal choice of ℓ changes, too. Hence, a joint optimization of t and ℓ is necessary.

We now simplify the problem by fixing the decision rule (i.e., the threshold ℓ) and then search for the *best* code $\mathscr{C}_t^{(2,n)}$ for the given threshold ℓ and the given channel (ϵ_0, ϵ_1). This turns out to be easier, but unless we happen to have chosen the optimal ℓ for the given BAC (ϵ_0, ϵ_1), this will result in a suboptimal solution.

We start with the following interesting result that has some important consequences.

Theorem 8.3 *Fix a blocklength n, a code parameter $0 \leq t \leq \lfloor \frac{n}{2} \rfloor$, and a decision rule threshold ℓ. Then the roots (ϵ_0, ϵ_1) of*

$$2P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_t^{(2,n)}\big) - 2P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_{t+1}^{(2,n)}\big) = 0 \tag{8.11}$$

are identical to the roots of

$$\mathrm{LLR}_t^{(n-1)}(\epsilon_0, \epsilon_1, \ell) = 0 \tag{8.12}$$

where $P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_t^{(2,n)}\big)$ denotes the error probability of code $\mathscr{C}_t^{(2,n)}$ decoded under the decision threshold ℓ. Moreover, for a fixed $\epsilon_0 \in \Omega$, there exists at most one $\epsilon_1 \in \Omega$ such that (8.11) holds; and for a fixed $\epsilon_1 \in \Omega$, there exists at most one $\epsilon_0 \in \Omega$ such that (8.11) holds. This means that if (8.11) has a solution, then this solution is unique for a fixed ϵ_0 or ϵ_1.

Proof: See Appendix A.4. □

Using Theorem 8.3 and Prop. A.1, we can now state conditions on t such that $\mathscr{C}_t^{(2,n)}$ is best under a fixed decision rule ℓ.

Corollary 8.4 *Fix a blocklength n and a decision rule ℓ. Then the flip code of type t, $\mathscr{C}_t^{(2,n)}$, is best for a fixed decision rule ℓ if, and only if, (ϵ_0, ϵ_1) belongs to*

$$\begin{aligned}\big\{(\epsilon_0, \epsilon_1) \colon \mathrm{LLR}_t^{(n-1)}(\epsilon_0, \epsilon_1, \ell) > 0 \text{ and} \\ \mathrm{LLR}_{t-1}^{(n-1)}(\epsilon_0, \epsilon_1, \ell) < 0\big\}.\end{aligned} \tag{8.13}$$

If the region is empty, then t is not best for any channel.

Proof: From (A.47) in the proof of Theorem 8.3 in Appendix A.4 and from assumption (3.1) it follows that

$$\mathrm{LLR}_t^{(n-1)}(\epsilon_0, \epsilon_1, \ell) > 0 \iff P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_t^{(2,n)}\big) < P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_{t+1}^{(2,n)}\big). \tag{8.14}$$

As we know from Prop. A.1 that $\mathrm{LLR}_t^{(n-1)}(\epsilon_0, \epsilon_1, \ell)$ is increasing in t, this means that if both (8.14) and

$$\mathrm{LLR}_{t-1}^{(n-1)}(\epsilon_0, \epsilon_1, \ell) < 0 \tag{8.15}$$

are satisfied, the code $\mathscr{C}_t^{(2,n)}$ is best for the given channel (ϵ_0, ϵ_1), for the given blocklength n, and for the fixed decision rule ℓ. □

We illustrate Cor. 8.4 by an example. We fix $n = 7$, $\ell = 2$, $\epsilon_1 = 0.5$, and let ϵ_0 increase from 0 to $\min\{\epsilon_1, 1 - \epsilon_1\} = 0.5$, see Fig. 8.8. Starting with $t = 3$, we check that

$$\mathrm{LLR}_2^{(6)}(\epsilon_0, 0.5, 2) > 0 \tag{8.16}$$

for all ϵ_0, i.e., $P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_2^{(2,7)}\big) < P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_3^{(2,7)}\big)$. Next, we check $t = 2$:

$$\mathrm{LLR}_1^{(6)}(\epsilon_0, 0.5, 2) < 0 \tag{8.17}$$

for small ϵ_0, i.e., the code $\mathscr{C}_2^{(2,7)}$ is best for those ϵ_0. When increasing ϵ_0, as soon as $\mathrm{LLR}_1^{(6)}(\epsilon_0, 0.5, 2) = 0$, there is a change and $\mathscr{C}_1^{(2,7)}$ becomes best. Further increasing ϵ_0 while keeping $t = 1$ then finally reveals the last change that happens at the root of $\mathrm{LLR}_0^{(6)}(\epsilon_0, 0.5, 2)$. So there are three best codes for $(\epsilon_0, 0.5) \in \Omega$:

- $\mathscr{C}_2^{(2,7)}$ is best in $\{\epsilon_0 \colon \mathrm{LLR}_2^{(6)}(\epsilon_0, 0.5, 2) > 0 \quad \text{and} \quad \mathrm{LLR}_1^{(6)}(\epsilon_0, 0.5, 2) < 0\}$;
- $\mathscr{C}_1^{(2,7)}$ is best in $\{\epsilon_0 \colon \mathrm{LLR}_1^{(6)}(\epsilon_0, 0.5, 2) > 0 \quad \text{and} \quad \mathrm{LLR}_0^{(6)}(\epsilon_0, 0.5, 2) < 0\}$;
- $\mathscr{C}_0^{(2,7)}$ is best in $\{\epsilon_0 \colon \mathrm{LLR}_0^{(6)}(\epsilon_0, 0.5, 2) > 0\}$.

In Fig. 8.8 the error probabilities of the various flip codes are shown as a function of ϵ_0. The best choices of t for all values of $(\epsilon_0, \epsilon_1) \in \Omega$ for $n = 7$ and $\ell = 2$ are shown in Fig. 8.9.

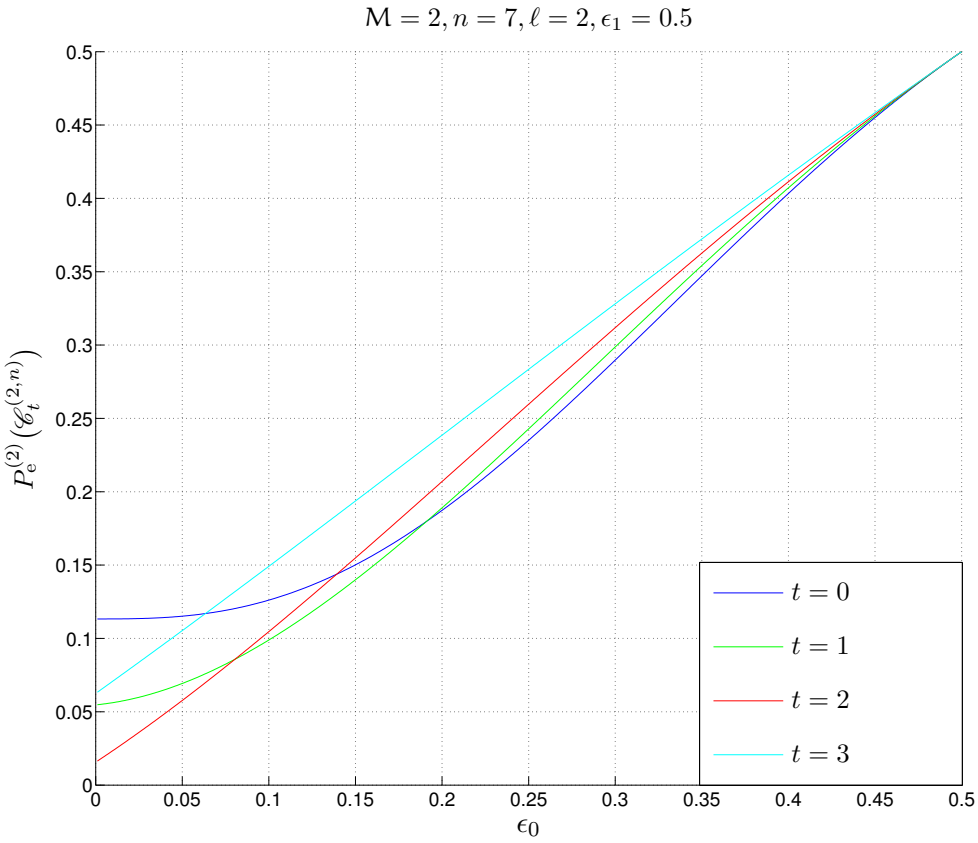


Figure 8.8: The error probabilities of all possible flip codes $\mathscr{C}_t^{(2,n)}$ as a function of the channel parameter ϵ_0, for a fixed blocklength $n = 7$, $\epsilon_1 = 0.5$, and a fixed decision rule $\ell = 2$. For any ϵ_0, the best code is the one with the smallest error probability value.

Cor. 8.4 shows that for a fixed decision rule ℓ, the choice of the best code parameter t depending on the given parameters n, ϵ_0, and ϵ_1 is much easier than the choice of the jointly optimal t and ℓ for a globally optimal code. In particular, we have the following regular structure.

Corollary 8.5 *Fix a blocklength n and a decision rule ℓ, and consider a BAC. If we increase ϵ_0 or decrease ϵ_1, then the best value of t is nonincreasing.*

More sloppily we can say that when we are moving inside of Ω (see Fig. 3.2) to the right or downwards, the best t will either remain the same or be reduced by 1. This means that the picture of the regions of best codes is much more regular without seemingly random jumps between different t. For an illustration compare the best codes for a fixed decision rule $\ell = 2$ in Fig. 8.9 with the corresponding globally optimal regions of Fig. 8.6.

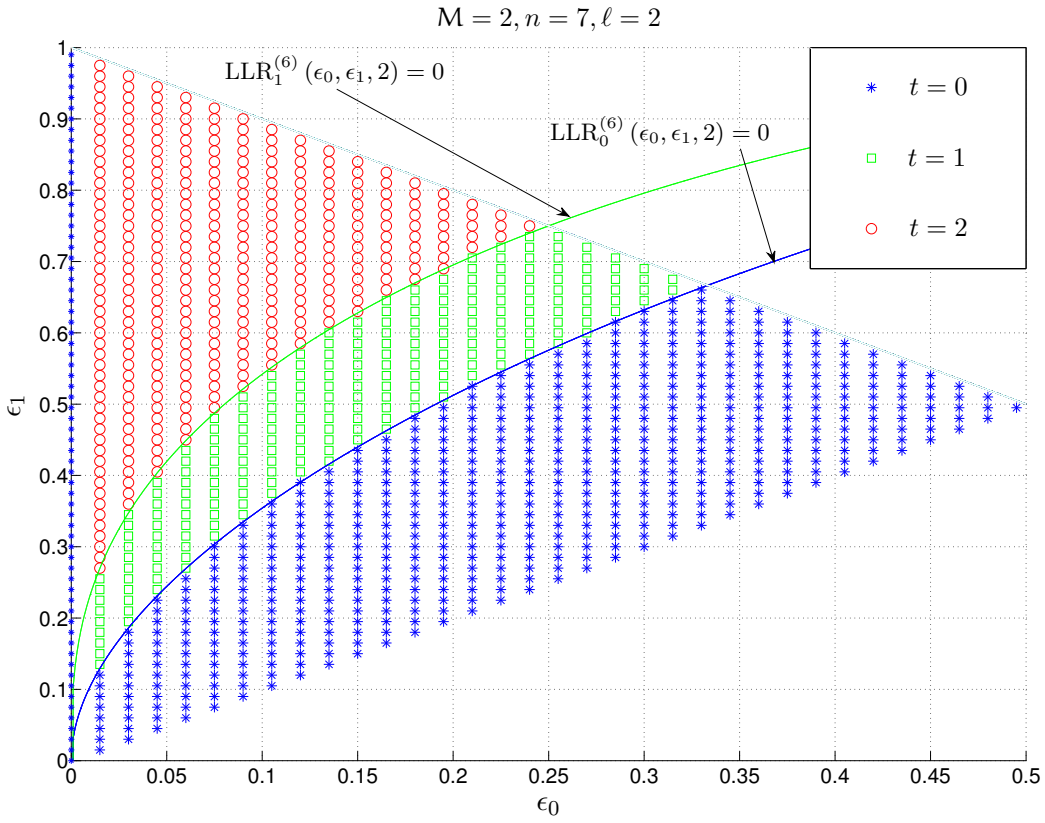


Figure 8.9: Best codebooks on a BAC for a fixed decision rule: for all possible (ϵ_0, ϵ_1) this plot shows the best choice of the code parameter t. The blocklength is $n = 7$ and the decision rule is $\ell = 2$.

Even more importantly, Theorem 8.3 also allows us to locate the exact location of some of the boundaries between the different areas of *globally optimal* codes (Fig. 8.6).

Corollary 8.6 *Consider the boundary between two areas of globally optimal codes (as, e.g., shown in Fig. 8.6). If the optimal decision rule on both sides of the boundary takes the same value ℓ and if the optimal code on the left is $t+1$, while the optimal code on the right is t, then this boundary is identical to a corresponding boundary in the situation with a fixed decision rule ℓ. In particular, this boundary is given by the roots of $\mathrm{LLR}_t^{(n-1)}(\epsilon_0, \epsilon_1, \ell)$.*

We again show the example of $n = 7$ from Fig. 8.6: in Fig. 8.10 the same plot is shown including a boundary that is identical to a boundary given in Fig. 8.9.

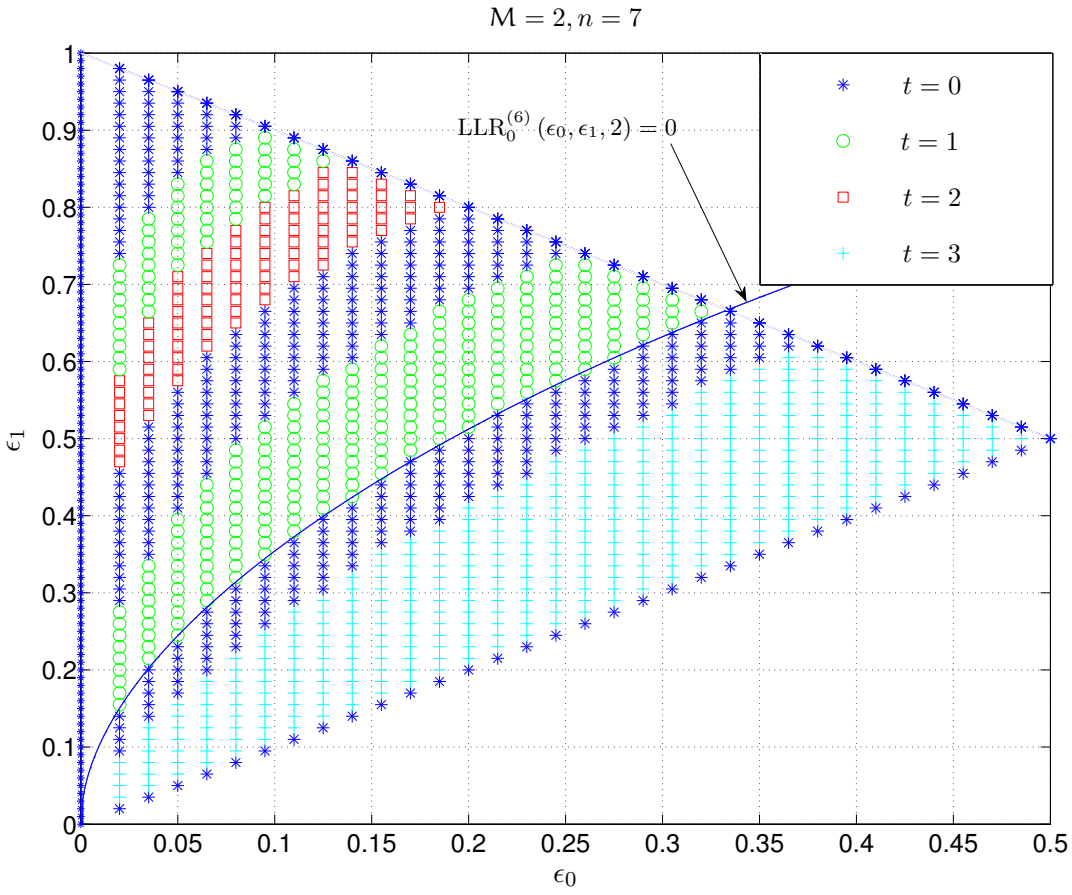


Figure 8.10: Globally optimal codebooks on a BAC for a blocklength $n = 7$ (identical to Fig. 8.6). The shown boundary between $t = 1$ and $t = 0$ is identical to the corresponding boundary given in Fig. 8.9, where a fixed decision rule $\ell = 2$ has been assumed.

We also would like to point out that the results for a given fixed decision rule simplify the search for a globally optimal code considerably. It can be summarized by the following algorithm.

Step 0: Fix a channel (ϵ_0, ϵ_1) and find the best t under the fixed decision rule $\ell = 0$ and its corresponding error probability $p \triangleq P_{\mathrm{e}}^{(0)}\big(\mathscr{C}_t^{(2,n)}\big)$. Then set $\ell \triangleq 1$.

Step 1: Find the best t_{temp} under a fixed decision rule ℓ and the corresponding error probability $P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_{t_{\mathrm{temp}}}^{(2,n)}\big)$.

Step 2: Check whether $P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_{t_{\mathrm{temp}}}^{(2,n)}\big) < p$. If yes, set $t \triangleq t_{\mathrm{temp}}$ and $p \triangleq P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_{t_{\mathrm{temp}}}^{(2,n)}\big)$.

Step 3: If $\ell < \left\lfloor \frac{n-1}{2} \right\rfloor$, $\ell \to \ell + 1$ and return to Step 1. Otherwise put out t (describing the optimal code) and p (giving the minimum error probability).

Chapter 9

Analysis of the ZC

9.1 Optimal Codes with Two Codewords ($\mathsf{M} = 2$)

Theorem 9.1 *For a ZC and for any $n \geq 1$, an optimal codebook with two codewords $\mathsf{M} = 2$ is the flip code of type 0, $\mathscr{C}_0^{(2,n)}$. It has an error probability*

$$P_{\mathrm{e}}\big(\mathscr{C}_0^{(2,n)}\big) = \frac{1}{2}\epsilon_1^n. \tag{9.1}$$

Proof: Due to Theorem 8.1, we can restrict our search to flip codes of some type t, $\mathscr{C}_t^{(2,n)}$, i.e., $\mathbf{x}_2 = \bar{\mathbf{x}}$ is the flipped version of $\mathbf{x}_1 = \mathbf{x}$.

For such a flip code, we observe that due to the peculiarity of the ZC that will never flip a zero to a one, an error can only occur when the received vector is the all-zero vector $\mathbf{y} = \mathbf{0}$:

$$\begin{aligned}&\min\left\{P_{Y|X}^n(\mathbf{y}|\mathbf{x}_1), P_{Y|X}^n(\mathbf{y}|\mathbf{x}_2)\right\}\\ &\quad= \begin{cases} 0 & \text{if } \mathbf{y} \neq \mathbf{0}\\ \epsilon_1^{\max\{w_{\mathrm{H}}(\mathbf{x}_1)w_{\mathrm{H}}(\mathbf{x}_2)\}} & \text{if } \mathbf{y} = \mathbf{0}. \end{cases}\end{aligned} \tag{9.2}$$

This error probability is minimized if one of the codewords is the all-one codeword; hence, $\mathscr{C}_0^{(2,n)}$ is optimal. □

Note the optimal code is linear. Moreover, from the proof it also follows that $\mathscr{C}_0^{(2,n)}$ is the unique optimal code.

9.2 Optimal Codes with Three or Four Codewords ($\mathsf{M} = 3, 4$)

Before we describe how we addressing the optimal codes for a ZC, we can firstly show that actually an optimal code must contain the all-zero vector $\mathbf{0}$ to be a codeword, this general property for any number of codewords M is proved in Lemma B.1 in Appendix B.1. Next we have shown that the optimal codes with theree or four codewords for a ZC can be constructed by the weak flip codes of type (t_2, t_3).

Theorem 9.2 *For a ZC and for any $n \geq 2$, an optimal codebook with three codewords $\mathsf{M}=3$ or four codewords $\mathsf{M}=4$ is the weak flip code of type $(t^*,0)$, $\mathscr{C}_{t^*,0}^{(\mathsf{M},n)}$, with*

$$t^* \triangleq \left\lfloor \frac{n}{2} \right\rfloor. \tag{9.3}$$

Moreover, the optimal code achieves the average error probability

$$P_{\mathrm{e}}\big(\mathscr{C}_{t^*,0}^{(\mathsf{M},n)}\big) = \begin{cases} \frac{1}{3}\big(\epsilon_1^{t^*} + \epsilon_1^{n-t^*}\big) & \text{if } \mathsf{M}=3 \\ \frac{1}{4}\big(2\epsilon_1^{t^*} + 2\epsilon_1^{n-t^*} - \epsilon_1^{n}\big) & \text{if } \mathsf{M}=4. \end{cases} \tag{9.4}$$

Proof: See Appendix B.1. □

Similar to the case of $\mathsf{M}=2$, we see that for $\mathsf{M}=4$ the optimal code given in Theorem 9.2 is linear. Also note that from the discussion in Appendix B.1 it follows that for even n, these linear codes are the unique optimal codes, while for odd n there are other (also nonlinear) designs that achieve the same optimal performance.

It is remarkable that these optimal codes perform quite well even for a very short blocklength. As an example, consider four codewords $\mathsf{M}=4$ of blocklength $n=10$ that are used over a ZC with $\epsilon_1 = 0.3$. The optimal average error probability is $P_{\mathrm{e}}\big(\mathscr{C}_{5,0}^{(4,10)}\big) \approx 2.43 \cdot 10^{-3}$. If we increase the blocklength to $n=20$, we already achieve an average error probability $P_{\mathrm{e}}\big(\mathscr{C}_{10,0}^{(4,20)}\big) \approx 5.90 \cdot 10^{-6}$. The asymptotic behavior of the optimal error probability for n going to infinity will be discussed in next section.

Next we will investigate the optimal code design from a new perspective: based on the fact that we consider a DMC, i.e., a channel that is memoryless and stationary, we would like to construct the codes *recursively* in the blocklength n.

We start with the following lemma.

Lemma 9.3 *Fix some arbitrary integers $\mathsf{M} \geq 2$, $n \geq 1$, and $\gamma \geq 1$. Consider a DMC and a code $\mathscr{C}^{(\mathsf{M},n)}$ for this DMC with M codewords and blocklength n, and create a new code $\mathscr{C}^{(\mathsf{M},n+\gamma)}$ by appending γ arbitrary column vectors to the codebook matrix of $\mathscr{C}^{(\mathsf{M},n)}$. Then the average success probability of this new code cannot be smaller than the success probability of the original code:*

$$P_{\mathrm{c}}\big(\mathscr{C}^{(\mathsf{M},n+\gamma)}\big) \geq P_{\mathrm{c}}\big(\mathscr{C}^{(\mathsf{M},n)}\big). \tag{9.5}$$

Proof: For a given code $\mathscr{C}^{(\mathsf{M},n)}$, the average success probability is given by

$$P_{\mathrm{c}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) = \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \sum_{\mathbf{y}^{(n)} \in \mathcal{D}_m^{(n)}} P_{\mathbf{Y}|\mathbf{X}}\big(\mathbf{y}^{(n)} \big| \mathbf{x}_m^{(n)}\big). \tag{9.6}$$

Now we consider the new codebook $\mathscr{C}^{(\mathsf{M},n+\gamma)}$, which is formed by appending γ columns to the original codebook matrix of $\mathscr{C}^{(\mathsf{M},n)}$. For convenience, we express the new codewords by

$$\mathbf{x}_m^{(n+\gamma)} = \big[\mathbf{x}_m^{(n)}\ \mathbf{x}_m^{(\gamma)}\big] \tag{9.7}$$

$$\triangleq \big(x_{m,1}\ \ x_{m,2}\ \cdots\ x_{m,n}\ \ x_{m,n+1}\ \cdots\ x_{m,n+\gamma}\big) \tag{9.8}$$

and likewise the extended received vector by

$$\mathbf{y}^{(n+\gamma)} = \left[\mathbf{y}^{(n)}\ \mathbf{y}^{(\gamma)}\right] \triangleq \left(y_1\ \ y_2\ \cdots\ y_{n+\gamma}\right). \tag{9.9}$$

Assume that a length-n received vector $\mathbf{y}^{(n)}$ is in the mth decoding region, $\mathbf{y}^{(n)} \in \mathcal{D}_m^{(n)}$. According to the ML decoding rule, a corresponding new received vector $\mathbf{y}^{(n+\gamma)}$ will change to another decoding region $\mathcal{D}_{m'}^{(n+\gamma)}$ if

$$\frac{P_{\mathbf{Y}|\mathbf{X}}\left(\left[\mathbf{y}^{(n)}\ \mathbf{y}^{(\gamma)}\right]\middle|\left[\mathbf{x}_{m'}^{(n)}\ \mathbf{x}_{m'}^{(\gamma)}\right]\right)}{P_{\mathbf{Y}|\mathbf{X}}\left(\left[\mathbf{y}^{(n)}\ \mathbf{y}^{(\gamma)}\right]\middle|\left[\mathbf{x}_{m}^{(n)}\ \mathbf{x}_{m}^{(\gamma)}\right]\right)} \geq 1. \tag{9.10}$$

Obviously, if no extended received vectors change its original decoding region from its length-n counterpart, then

$$P_{\mathrm{c}}\left(\mathscr{C}^{(\mathsf{M},n+\gamma)}\right) = \frac{1}{\mathsf{M}}\sum_{m=1}^{\mathsf{M}}\left(\sum_{\mathbf{y}^{(n)}\in\mathcal{D}_m^{(n)}} P_{\mathbf{Y}|\mathbf{X}}\left(\mathbf{y}^{(n)}\middle|\mathbf{x}_m^{(n)}\right) \cdot \underbrace{\sum_{\mathbf{y}^{(\gamma)}\in\mathcal{Y}^{\gamma}} P_{\mathbf{Y}|\mathbf{X}}\left(\mathbf{y}^{(\gamma)}\middle|\mathbf{x}_m^{(\gamma)}\right)}_{=1}\right) \tag{9.11}$$

$$= P_{\mathrm{c}}\left(\mathscr{C}^{(\mathsf{M},n)}\right) \tag{9.12}$$

where $\mathcal{Y}$ denotes the output alphabet. However, if some $\mathbf{y}^{(n+\gamma)}$ change its original decoding region of blocklength n, the new success probability will be

$$\begin{aligned} &P_{\mathrm{c}}\left(\mathscr{C}^{(\mathsf{M},n+\gamma)}\right) \\ &= P_{\mathrm{c}}\left(\mathscr{C}^{(\mathsf{M},n)}\right) + \frac{1}{\mathsf{M}}\sum_{m=1}^{\mathsf{M}}\sum_{\substack{\mathbf{y}^{(n+\gamma)} \\ \text{s.t. } \mathbf{y}^{(n)}\in\mathcal{D}_m^{(n)} \\ \text{but } \mathbf{y}^{(n+\gamma)}\in\mathcal{D}_{m'}^{(n+\gamma)}}}\left(P_{\mathbf{Y}|\mathbf{X}}\left(\mathbf{y}^{(n+\gamma)}\middle|\mathbf{x}_{m'}^{(n+\gamma)}\right)\right. \\ &\qquad\qquad \left. - P_{\mathbf{Y}|\mathbf{X}}\left(\mathbf{y}^{(n+\gamma)}\middle|\mathbf{x}_{m}^{(n+\gamma)}\right)\right) \qquad (9.13) \\ &\triangleq P_{\mathrm{c}}\left(\mathscr{C}^{(\mathsf{M},n)}\right) + \Delta\Psi\left(\mathscr{C}^{(\mathsf{M},n+\gamma)}\right). \qquad (9.14) \end{aligned}$$

The proof of Lemma 9.3 is completed by noting from (9.10) that $\Delta\Psi\left(\mathscr{C}^{(\mathsf{M},n+\gamma)}\right)$ is always nonnegative. □

Definition 9.4 *The term* $\Delta\Psi\left(\mathscr{C}^{(\mathsf{M},n+\gamma)}\right)$ *in* (9.14) *is called* total probability increase for a step-size γ *and describes the amount by which the average success probability of the code* $\mathscr{C}^{(\mathsf{M},n)}$ *grows when* γ *column vectors are appended to its codebook matrix.*

Lemma 9.5 *For a ZC, for any* $n \geq 2$, *and for* $1 \leq t \leq \left\lfloor \frac{n}{2} \right\rfloor$, *consider the weak flip code of type* $(t,0)$ *with four codewords* $\mathsf{M} = 4$, $\mathscr{C}_{t,0}^{(4,n)}$, *and append a column to the codebook matrix*

to create a new code of length $n+1$. Then the total probability increase is maximized if, among all possible $2^4 = 16$ columns, we choose $\mathbf{c}_2^{(4)}$. If $t < \lfloor \frac{n}{2} \rfloor$, or if n is odd and $t = \lfloor \frac{n}{2} \rfloor$, then this choice is unique.

For $\mathsf{M} = 3$, appending $\mathbf{c}_2^{(3)}$ or $\mathbf{c}_3^{(3)}$ to $\mathscr{C}_{t,0}^{(3,n)}$ is equally optimal.

Proof: See Appendix B.2. □

We would like to point out that the codes $\mathscr{C}_{t,0}^{(4,n)}$ can be seen as *double-flip codes* consisting of the combination of the (two-codeword) flip code of type 0 with the (two-codeword) flip code of type $t > 0$:

$$\mathscr{C}_{t,0}^{(4,n)} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{x} \\ \bar{\mathbf{x}} \\ \mathbf{1} \end{pmatrix} \tag{9.15}$$

with $\mathbf{x}$ and $\bar{\mathbf{x}}$ defined in (6.1).

From the recursive technique that we have used in the derivation of Lemma 9.5 and that is based on the addition of columns to the codebook matrix, it immediately follows that our optimal codes can be constructed recursively in n. Concretely, we have the following corollary.

Corollary 9.6 *The optimal codebooks defined in Theorem 9.2 for $\mathsf{M} = 3$ and $\mathsf{M} = 4$ can be constructed recursively in the blocklength n by adding a column that yields the maximum total probability increase. We start with an optimal codebook for $n = 2$:*

$$\mathscr{C}_{\mathrm{ZC}}^{(\mathsf{M},2)*} = \left(\mathbf{c}_1^{(\mathsf{M})} \quad \mathbf{c}_2^{(\mathsf{M})} \right). \tag{9.16}$$

Then, we recursively construct the optimal codebook for $n \geq 3$ by using $\mathscr{C}_{\mathrm{ZC}}^{(\mathsf{M},n-1)*}$ and appending

$$\begin{cases} \mathbf{c}_1^{(\mathsf{M})} & \text{if } n \bmod 2 = 1 \\ \mathbf{c}_2^{(\mathsf{M})} & \text{if } n \bmod 2 = 0. \end{cases} \tag{9.17}$$

Proof: We only need to show that the constructed codes from (9.17) are equivalent to the optimal codes given in Theorem 9.2. The optimal code for $\mathsf{M} = 4$ and $n = 2$ is trivial and given by (9.16). Next assume that for blocklength n, $\mathscr{C}_{\lfloor \frac{n}{2} \rfloor,0}^{(4,n)}$ is optimal. From Lemma 9.5 we know that the largest total probability increase is achieved when adding column $\mathbf{c}_2^{(4)}$. Now note that for n even with $t = \frac{n}{2}$, adding the column $\mathbf{c}_2^{(4)}$ to the code $\mathscr{C}_{\frac{n}{2},0}^{(4,n)}$ will result in a code that is equivalent to $\mathscr{C}_{\lfloor \frac{n+1}{2} \rfloor,0}^{(4,n+1)}$: we only need to exchange the roles of the second and third codeword and then re-order the columns. For n odd with $t = \lfloor \frac{n}{2} \rfloor$, adding the column $\mathbf{c}_2^{(4)}$ to the code $\mathscr{C}_{\lfloor \frac{n}{2} \rfloor,0}^{(4,n)}$ results in $\mathscr{C}_{\frac{n+1}{2},0}^{(4,n+1)}$.

Hence, we see that $\mathscr{C}_{\lfloor \frac{n+1}{2} \rfloor,0}^{(4,n+1)}$ is still optimal. The claim now follows by induction in n. The case with three codewords $\mathsf{M} = 3$ can be proved in a similar manner.

Note that we have actually proven that any codebook consisting of $n - t^*$ columns $\mathbf{c}_1^{(3)}$ and t^* columns arbitrarily chosen from $\mathbf{c}_2^{(3)}$ or $\mathbf{c}_3^{(3)}$ is optimal on a ZC (see the main discussion in Appendix B.1). □

We conclude this section by a remark. While it is very intuitive to construct the codes recursively, i.e., to start from an optimal code for n and then to add one column that maximizes the total probability increase, unfortunately, from a proof perspective, such a recursive construction only guarantees local optimality: one still needs a proof (Thm. 9.2) that the achieved code of blocklength $n+1$ is globally optimum.

9.3 Error Exponents

Since the ZC is not pairwise reversible, the error exponents for $\mathsf{M} = 3$ or $\mathsf{M} = 4$ codewords were previously unknown. Using that for the optimal code $\mathscr{C}_{t^*,0}^{(\mathsf{M},n)}$ we have

$$\mathsf{D}_{\min}^{(\mathrm{ZC})}\left(\mathscr{C}_{t^*,0}^{(\mathsf{M},n)}\right) = \begin{cases} -\frac{1}{2}\log\epsilon_1 & \text{if } n \bmod 2 = 0 \\ -\frac{\lfloor \frac{n}{2} \rfloor}{n}\log\epsilon_1 & \text{if } n \bmod 2 = 1 \end{cases} \tag{9.18}$$

and

$$\mathsf{D}_{\min}^{(\mathrm{ZC})}\left(\mathscr{C}_{\lfloor \frac{n+1}{3} \rfloor, \lfloor \frac{n}{3} \rfloor}^{(\mathsf{M},n)}\right) = \begin{cases} -\frac{1}{3}\log\epsilon_1 & \text{if } n \bmod 3 = 0 \\ -\frac{\lfloor \frac{n}{3} \rfloor + 1}{n}\log\epsilon_1 & \text{if } n \bmod 3 = 1 \\ -\frac{\lfloor \frac{n}{3} \rfloor + 1}{n}\log\epsilon_1 & \text{if } n \bmod 3 = 2 \end{cases} \tag{9.19}$$

we can now compute them:

$$\mathsf{E}_3 = \mathsf{E}_4 = -\frac{1}{2}\log\epsilon_1. \tag{9.20}$$

9.4 Application to Known Bounds on the Error Probability for a Finite Blocklength

Since we now know the optimal code structure and its performance, it is interesting to compare it to the known bounds introduced in Section 7.1. Fig. 9.11 and Fig. 9.12 compare some SGB bounds and the Gallager bound with the exact performance of the optimal code (for $\mathsf{M} = 3$ and $\mathsf{M} = 4$ codewords, respectively). Besides the Gallager bound, we plot the SGB lower bound based on the optimal code structure (thereby making sure that this lower bound is valid generally), and we show two SGB upper bounds: one that is based on the optimal code design and one that is based on the fair weak flip code used by Shannon *et al.*

We see that the SGB upper bound that is based on the optimal code is quite close to the exact performance, in particular, it exhibits the correct error exponent. The SGB upper bound that is based on the fair weak flip code, on the other hand, does not achieve the error exponent (which can be expected because the ZC is not pairwise reversible). Also the Gallager bound does not achieve the correct exponential behavior. The SGB lower bound is quite loose.

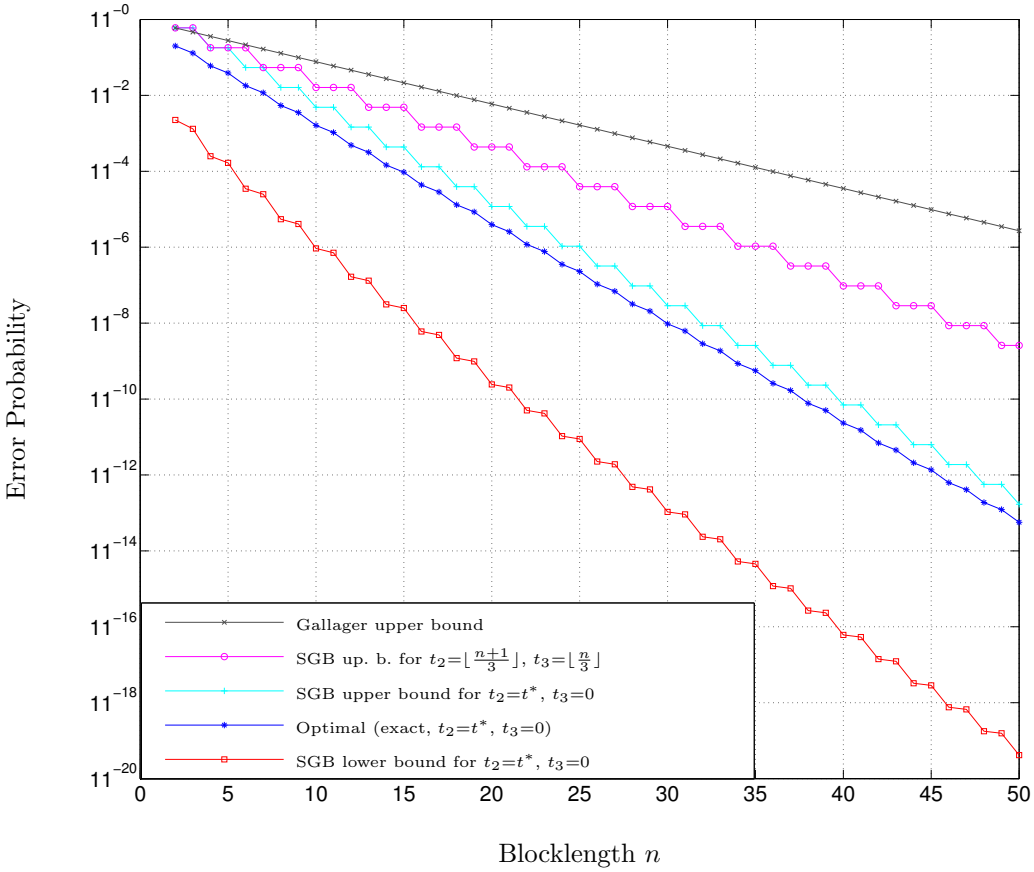


Figure 9.11: Exact value of, and bounds on, the performance of an optimal code with $\mathrm{M} = 3$ codewords on the ZC with $\epsilon_1 = 0.3$ as a function of the blocklength n.

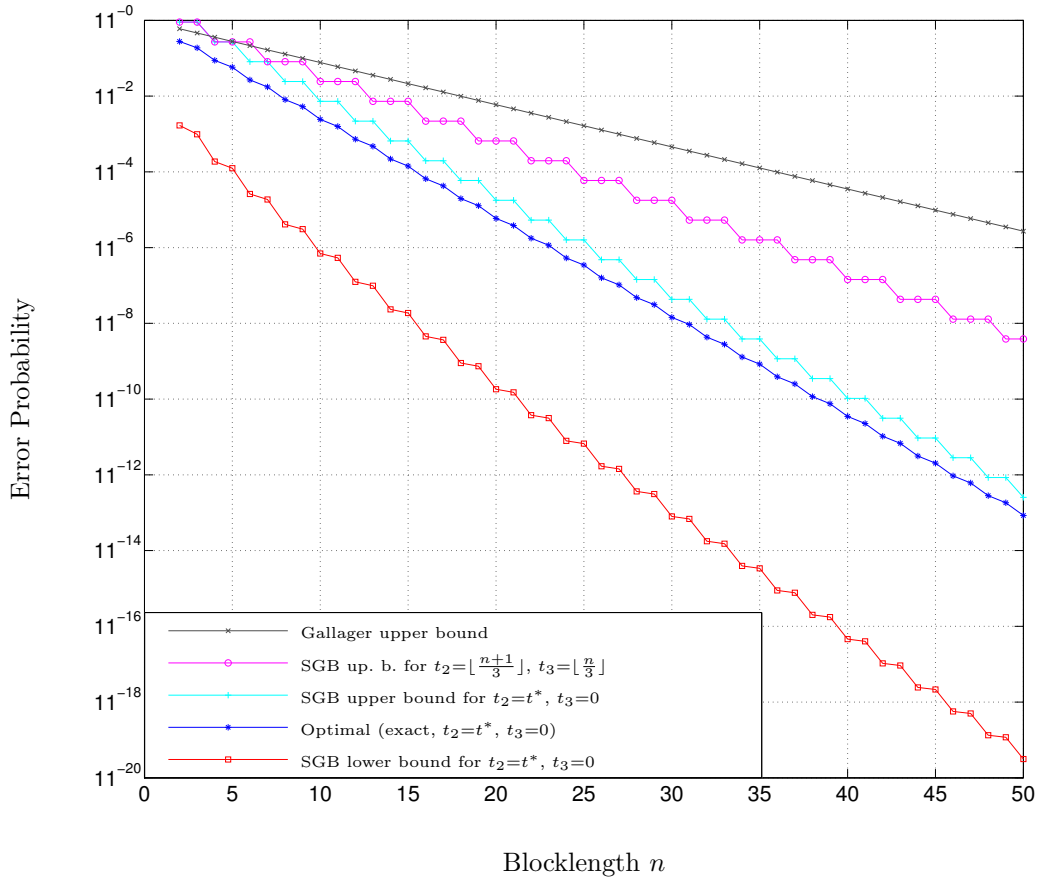


Figure 9.12: Exact value of, and bounds on, the performance of an optimal code with $\mathsf{M} = 4$ codewords on the ZC with $\epsilon_1 = 0.3$ as a function of the blocklength n.

9.5 Conjectured Optimal Codes with Five Codewords ($\mathsf{M} = 5$)

The idea of designing an optimal code recursively promises to be a very powerful approach. Unfortunately, for larger values of M, we might need a recursion from n to $n+\gamma$ with a step-size $\gamma > 1$. In the following we conjecture an optimal code construction for a ZC in the case of five codewords $\mathsf{M} = 5$ with a different recursive design for n odd and n even (i.e., with a step-size $\gamma = 2$).

We define the following five weak flip column vectors:

$$\left\{ \mathbf{c}_1^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \ \mathbf{c}_2^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \ \mathbf{c}_3^{(5)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \right.$$
$$\left. \mathbf{c}_4^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \ \mathbf{c}_5^{(5)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \right\}. \tag{9.21}$$

An optimal code can be constructed recursively for even n in the following way. We start with an optimal codebook for $n = 8$:

$$\mathscr{C}_{\mathrm{ZC}}^{(5,8)*} = \left(\mathbf{c}_1^{(5)} \quad \mathbf{c}_2^{(5)} \quad \mathbf{c}_3^{(5)} \quad \mathbf{c}_1^{(5)} \quad \mathbf{c}_2^{(5)} \quad \mathbf{c}_3^{(5)} \quad \mathbf{c}_4^{(5)} \quad \mathbf{c}_5^{(5)} \right). \tag{9.22}$$

Then we recursively construct the optimal codebook for $n \geq 10$, n even, by using $\mathscr{C}_{\mathrm{ZC}}^{(5,n-2)*}$ and appending

$$\begin{cases} \begin{pmatrix} \mathbf{c}_4^{(5)} & \mathbf{c}_5^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 0 \\ \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_2^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 2 \\ \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_3^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 4 \\ \begin{pmatrix} \mathbf{c}_3^{(5)} & \mathbf{c}_4^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 6 \\ \begin{pmatrix} \mathbf{c}_2^{(5)} & \mathbf{c}_5^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 8. \end{cases} \tag{9.23}$$

For n odd, we start with the length-9 code

$$\mathscr{C}_{\mathrm{ZC}}^{(5,9)*} = \Big(\mathbf{c}_1^{(5)} \quad \mathbf{c}_2^{(5)} \quad \mathbf{c}_3^{(5)} \quad \mathbf{c}_4^{(5)} \quad \mathbf{c}_5^{(5)} \quad \mathbf{c}_1^{(5)} \quad \mathbf{c}_2^{(5)} \quad \mathbf{c}_1^{(5)} \quad \mathbf{c}_3^{(5)} \Big) \tag{9.24}$$

and recursively construct the optimal codebook for $n \geq 11$, n odd, by using $\mathscr{C}_{\mathrm{ZC}}^{(5,n-2)*}$ and

appending

$$\begin{cases} \begin{pmatrix} \mathbf{c}_3^{(5)} & \mathbf{c}_4^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 1 \\ \begin{pmatrix} \mathbf{c}_2^{(5)} & \mathbf{c}_5^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 3 \\ \begin{pmatrix} \mathbf{c}_4^{(5)} & \mathbf{c}_5^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 5 \\ \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_2^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 7 \\ \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_3^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 9. \end{cases} \tag{9.25}$$

Note that the recursive structure in (9.23) and (9.25) is actually identical apart from the ordering. Also note that when increasing the blocklength by 10, we add each of the five column vectors in (A.5) exactly twice. For $n < 10$ the optimal code structure goes through some transient states.

Chapter 10

Analysis of the BSC

10.1 Optimal Codes with Two Codewords ($\mathsf{M} = 2$)

Theorem 10.1 *For a BSC and for any $n \geq 1$, an optimal codebook with two codewords $\mathsf{M} = 2$ is the flip code of type t for any $t \in \{0, 1, \ldots, \lfloor \frac{n}{2} \rfloor\}$.*

Proof: From Theorem 8.1 we already know that there must exist a flip code that is optimal. Moreover, Theorem 8.1 also shows that the all-zero and the all-one column in a codebook matrix is strictly suboptimal. So, we only have two possible choices of candidate columns: $(0\ 1)^{\mathsf{T}}$ and $(1\ 0)^{\mathsf{T}}$. However, by the symmetry of a BSC, both columns will result in an identical performance. Therefore every flip code has the same performance, i.e., all of them must be optimal. □

10.2 Optimal Codes with Three or Four Codewords ($\mathsf{M} = 3, 4$)

Unlike in the case of a ZC, for a BSC it is not easy to derive the exact average error probability expressed only by the candidate column parameters t_i. So instead we use the idea of a recursive code construction that guarantees largest total probability increase.

Theorem 10.2 *For a BSC with arbitrary crossover probability $0 \leq \epsilon < \frac{1}{2}$, the optimal code for $n = 2$ is*

$$\mathscr{C}_{\mathrm{BSC}}^{(\mathsf{M},2)*} = \begin{pmatrix} \mathbf{c}_1^{(\mathsf{M})} & \mathbf{c}_2^{(\mathsf{M})} \end{pmatrix}. \tag{10.1}$$

If we recursively construct a locally optimal codebook for $n \geq 3$ by using $\mathscr{C}_{\mathrm{BSC}}^{(\mathsf{M},n-1)*}$ and appending a new column, the total probability increase is maximized by the following choice of appended columns:

$$\begin{cases} \mathbf{c}_1^{(\mathsf{M})} & \text{if } n \bmod 3 = 0 \\ \mathbf{c}_3^{(\mathsf{M})} & \text{if } n \bmod 3 = 1 \\ \mathbf{c}_2^{(\mathsf{M})} & \text{if } n \bmod 3 = 2. \end{cases} \tag{10.2}$$

Proof: See Appendix C.1. □

While Theorem 10.2 only guarantees optimality under the condition that the optimal code can be constructed recursively, much points to that the given construction indeed is optimum.

Theorem 10.3 *For a BSC and for any $n \geq 2$, an optimal codebook with three codewords $\mathsf{M} = 3$ or a linear optimal codebook with four codewords $\mathsf{M} = 4$ is the weak flip code of type (t_2^*, t_3^*), $\mathscr{C}_{t_2^*,t_3^*}^{(\mathsf{M},n)}$, where*

$$t_2^* \triangleq \left\lfloor \frac{n+1}{3} \right\rfloor, \quad t_3^* \triangleq \left\lfloor \frac{n-1}{3} \right\rfloor. \tag{10.3}$$

Using the shorthands

$$k \triangleq \left\lfloor \frac{n}{3} \right\rfloor, \quad p \triangleq \left(\frac{\epsilon}{1-\epsilon} \right) < 1 \tag{10.4}$$

the code parameters of these optimal codes can be written as

$$[t_1^*, t_2^*, t_3^*] = \begin{cases} [k+1, k, k-1] & \text{if } n \bmod 3 = 0 \\ [k+1, k, k] & \text{if } n \bmod 3 = 1 \\ [k+1, k+1, k] & \text{if } n \bmod 3 = 2. \end{cases} \tag{10.5}$$

Furthermore, the exact average success probability can be derived recursively in blocklength n, starting with

$$3P_{\mathrm{c}}(\mathscr{C}_{\mathrm{BSC}}^{(3,2)*}) = (1-\epsilon)^2 \cdot (3+p). \tag{10.6}$$

Then

$$\begin{aligned} 3P_{\mathrm{c}}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n)}\big) &= 3P_{\mathrm{c}}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) \\ &\quad + \sum_{a_3=0}^{k-1} \sum_{a_2=0}^{a_3} \binom{k}{a_1 = a_2} \binom{k}{a_2} \binom{k-1}{a_3} \\ &\quad \cdot (1-\epsilon)^n \left(p^{2k-1-a_3} - p^{2k-a_3} \right) \\ &\quad \text{if } n = 3k \end{aligned} \tag{10.7}$$

$$\begin{aligned} 3P_{\mathrm{c}}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n)}\big) &= 3P_{\mathrm{c}}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) \\ &\quad + \sum_{a_2=1}^{k} \sum_{a_1=1}^{a_2} \binom{k+1}{a_1} \binom{k}{a_2} \binom{k-1}{a_3 = a_1 - 1} \\ &\quad \cdot (1-\epsilon)^n \left(p^{2k-a_2} - p^{2k+1-a_2} \right) \\ &\quad \text{if } n = 3k+1 \end{aligned} \tag{10.8}$$

$$\begin{aligned} 3P_{\mathrm{c}}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n)}\big) &= 3P_{\mathrm{c}}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) \\ &\quad + \sum_{a_1=1}^{k+1} \sum_{a_3=0}^{a_1-1} \binom{k+1}{a_1} \binom{k}{a_2 = a_3} \binom{k}{a_3} \\ &\quad \cdot (1-\epsilon)^n \left(p^{2k+1-a_1} - p^{2k+2-a_1} \right) \\ &\quad \text{if } n = 3k+2. \end{aligned} \tag{10.9}$$

While for $\mathsf{M} = 4$ *the exact average success probability is very similar to the case of* $\mathsf{M} = 3$. *In the above equations,* a_m *actually denotes the number of positions where the receive vector differrs from those* t_m^* *columns. These convenient notations will be useful in proving the results (See* (C.31)*).*"

Note that for $\mathsf{M} = 2$, the optimal codes given in Theorem 10.1 can be linear or nonlinear. For $\mathsf{M} = 4$, only the linear globally optimal codes in Theorem 10.3 is investigated. However, due to the strong symmetry of the BSC, there also exist nonlinear codes with the same optimal performance.

We also would like to point out the regularity of the code design in Theorem 10.2 that repeats in n with a period of 3. For $\mathsf{M} = 5$, we expect a similar behavior, but with a period that is larger than 3.

Moreover, a closer inspection of the proof of Theorem 10.3 shows that when $\mathsf{M} = 3$, the received vector $\mathbf{y}$ farthest from the three codewords is

$$\mathbf{y} = (\underbrace{1 \cdots 1}_{t_1^*} \underbrace{1 \cdots 1}_{t_2^*} \underbrace{0 \cdots 0}_{t_3^*}), \tag{10.10}$$

which corresponds to the optimal choice of the fourth linear codeword $\mathbf{x}_4$ when $\mathsf{M} = 4$.

10.3 Pairwise Hamming Distance Structure

As already mentioned in Section 4.5, it is quite common in conventional coding theory to use the *minimum Hamming distance* or the *weight enumerating function (WEF)* of a code as a design and quality criterion [12]. This is motivated by the equivalence of Hamming weight and Hamming distance for linear codes, and by the union bound that converts the search for the global error probability into pairwise error probabilities. Since we are interested in the globally optimal code design and the best performance achieved by an ML decoder, we can neither use the union bound, nor can we *a priori* restrict our search to linear codes. Note that for most values of M, linear codes do not even exist![7]

We would like to come back to the example shown in Chapter 5 and further deepen our analysis of the minimum Hamming distance of our optimal codes on the very symmetric BSC. Although, as (4.16) shows, the error probability performance of a BSC is completely specified by the Hamming distance between codewords and received vectors, we will now demonstrate that a design based on the minimum Hamming distance can fail, even for the very symmetric BSC and even for linear codes. In the case of a more general (and not symmetric) BAC, this will be even more pronounced.

We compare the optimal codes given in Theorem 10.2 with the following different weak

[7] Interestingly, a subfamily of the weak flip codes can be shown to have many linear-like properties. For more details see [14].

flip code $\mathscr{C}_{\text{subopt}}^{(\mathsf{M},n)}$ with code parameters

$$[t_1,t_2,t_3]=\begin{cases}[k,k,k] & \text{if } n \bmod 3=0\\ [k+1,k+1,k-1] & \text{if } n \bmod 3=1\\ [k+2,k,k] & \text{if } n \bmod 3=2.\end{cases} \tag{10.11}$$

This code can be constructed from the optimal code $\mathscr{C}_{\text{BSC}}^{(\mathsf{M},n-1)*}$ by appending a suboptimal column[8] and—based on a closer inspection of the proof of Theorem 10.2—can be shown to be strictly suboptimal.

Recalling Lemma 6.7, we compute the pairwise Hamming distance vector of the optimal code for $\mathsf{M}=3$:

$$\mathbf{d}\big(\mathscr{C}_{\text{BSC}}^{(3,n)*}\big)=\begin{cases}(2k-1,2k,2k+1) & \text{if } n \bmod 3=0\\ (2k,2k,2k+1) & \text{if } n \bmod 3=1\\ (2k+1,2k+1,2k+2) & \text{if } n \bmod 3=2\end{cases} \tag{10.12}$$

i.e.,

$$d_{\min}\big(\mathscr{C}_{\text{BSC}}^{(3,n)*}\big)=\begin{cases}2k-1 & \text{if } n \bmod 3=0\\ 2k & \text{if } n \bmod 3=1\\ 2k+1 & \text{if } n \bmod 3=2.\end{cases} \tag{10.13}$$

For $\mathsf{M}=4$ we get accordingly:

$$\begin{aligned}&\mathbf{d}\big(\mathscr{C}_{\text{BSC}}^{(4,n)*}\big)\\ &=\begin{cases}(2k-1,2k,2k+1,2k+1,2k,2k-1) & \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } n \bmod 3=0\\ (2k,2k,2k+1,2k+1,2k,2k) & \text{if } n \bmod 3=1\\ (2k+1,2k+1,2k+2,2k+2,2k+1,2k+1) & \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } n \bmod 3=2\end{cases}\end{aligned} \tag{10.14}$$

with the same values for the minimum Hamming distance as for the $\mathsf{M}=3$.

Comparing this with the suboptimal code (10.11) now yields for $\mathsf{M}=3$:

$$\mathbf{d}\big(\mathscr{C}_{\text{subopt}}^{(3,n)}\big)=\begin{cases}(2k,2k,2k) & \text{if } n \bmod 3=0\\ (2k,2k,2k+2) & \text{if } n \bmod 3=1\\ (2k,2k+2,2k+2) & \text{if } n \bmod 3=2\end{cases} \tag{10.15}$$

[8] The choice of column depends on n.

i.e., $d_{\min}\big(\mathscr{C}_{\text{subopt}}^{(3,n)}\big) = 2k$ in all cases. For $\mathsf{M} = 4$ we have

$$\mathbf{d}\big(\mathscr{C}_{\text{subopt}}^{(4,n)}\big) = \begin{cases} (2k, 2k, 2k, 2k, 2k, 2k) & \text{if } n \bmod 3 = 0 \\ (2k, 2k, 2k+2, 2k+2, 2k, 2k) & \text{if } n \bmod 3 = 1 \\ (2k, 2k+2, 2k+2, 2k+2, 2k+2, 2k) & \\ & \text{if } n \bmod 3 = 2 \end{cases} \tag{10.16}$$

with also $d_{\min}\big(\mathscr{C}_{\text{subopt}}^{(4,n)}\big) = 2k$ in all cases.

Hence, we see that for $n \bmod 3 = 0$ the minimum Hamming distance of the optimal code is $2k-1$ and therefore strictly smaller than the corresponding minimum Hamming distance $2k$ of the suboptimal code.

By adapting the construction of the strictly suboptimal code $\mathscr{C}_{\text{subopt}}^{(\mathsf{M},n)}$, a similar statement can be made for the case when $n \bmod 3 = 1$.

We have shown the following proposition.

Proposition 10.4 *On a BSC for $\mathsf{M} = 3$ or $\mathsf{M} = 4$ and for all n with $n \bmod 3 = 0$ or $n \bmod 3 = 1$, codes that maximize the minimum Hamming distance $d_{\min}\big(\mathscr{C}^{(\mathsf{M},n)}\big)$ can be strictly suboptimal. This is not true in the case of $n \bmod 3 = 2$.*

As a matter of fact, using a result from [14], one can show that on a BSC for $\mathsf{M} = 3$ or $\mathsf{M} = 4$ and in the case of $n \bmod 3 = 0$, all codes that maximize the minimum Hamming distance are strictly suboptimal.

10.4 Application to Known Bounds on the Error Probability for a Finite Blocklength

We again provide a comparison between the performance of the optimal code to the known bounds of Chapter 7.

Note that the error exponents for $\mathsf{M} = 3, 4$ codewords are

$$\mathsf{E}_3 = \mathsf{E}_4 = -\frac{2}{3}\log\sqrt{4\epsilon(1-\epsilon)}. \tag{10.17}$$

Moreover, for $\mathsf{M} = 3, 4$,

$$\mathsf{D}_{\min}^{(\text{BSC})}\Big(\mathscr{C}_{\lfloor\frac{n+1}{3}\rfloor,\lfloor\frac{n}{3}\rfloor}^{(\mathsf{M},n)}\Big) = \begin{cases} -\frac{2}{3}\log\sqrt{4\epsilon(1-\epsilon)} & \text{if } n \bmod 3 = 0 \\ -\frac{\lfloor\frac{n}{3}\rfloor+\lfloor\frac{n+1}{3}\rfloor}{n}\log\sqrt{4\epsilon(1-\epsilon)} & \text{if } n \bmod 3 = 1 \\ -\frac{\lfloor\frac{n}{3}\rfloor+\lfloor\frac{n+1}{3}\rfloor}{n}\log\sqrt{4\epsilon(1-\epsilon)} & \text{if } n \bmod 3 = 2. \end{cases} \tag{10.18}$$

Figs. 11.15 and 11.16 compare the exact optimal performance for $\mathsf{M} = 3$ and $\mathsf{M} = 4$, respectively, with some bounds: the SGB upper bound based on the weak flip code used by

Shannon *et al.*,[9], the SGB lower bound based on the weak flip code (which is suboptimal, but achieves the optimal $\mathsf{D}_{\min}^{(\mathrm{DMC})}$ and is therefore a generally valid lower bound), the Gallager upper bound, and also the PPV upper and lower bounds.

We can see that the PPV upper bound is tighter to the exact optimal performance than the SGB upper bound. Note, however, that neither exhibits the correct error exponent. It is shown in [20] that, for n going to infinity, the random coding (PPV) upper bound tends to the Gallager exponent for $R = 0$ [6], which is of course not necessarily equal to E_{M} for finite M.

Concerning the lower bounds, we see that the PPV lower bound (meta-converse) is much better for finite n than the SGB bound. However, for n large enough, its exponential growth will approach that of the sphere-packing bound [17], which does not equal to E_{M} either.

Once more we would like to point out that even though the fair weak flip codes achieve the error exponent, they are strictly suboptimal for every $n \bmod 3 = 0$.

[9]The SGB upper bound based on the optimal code performs almost identically (because the BSC is pairwise reversible) and is therefore omitted.

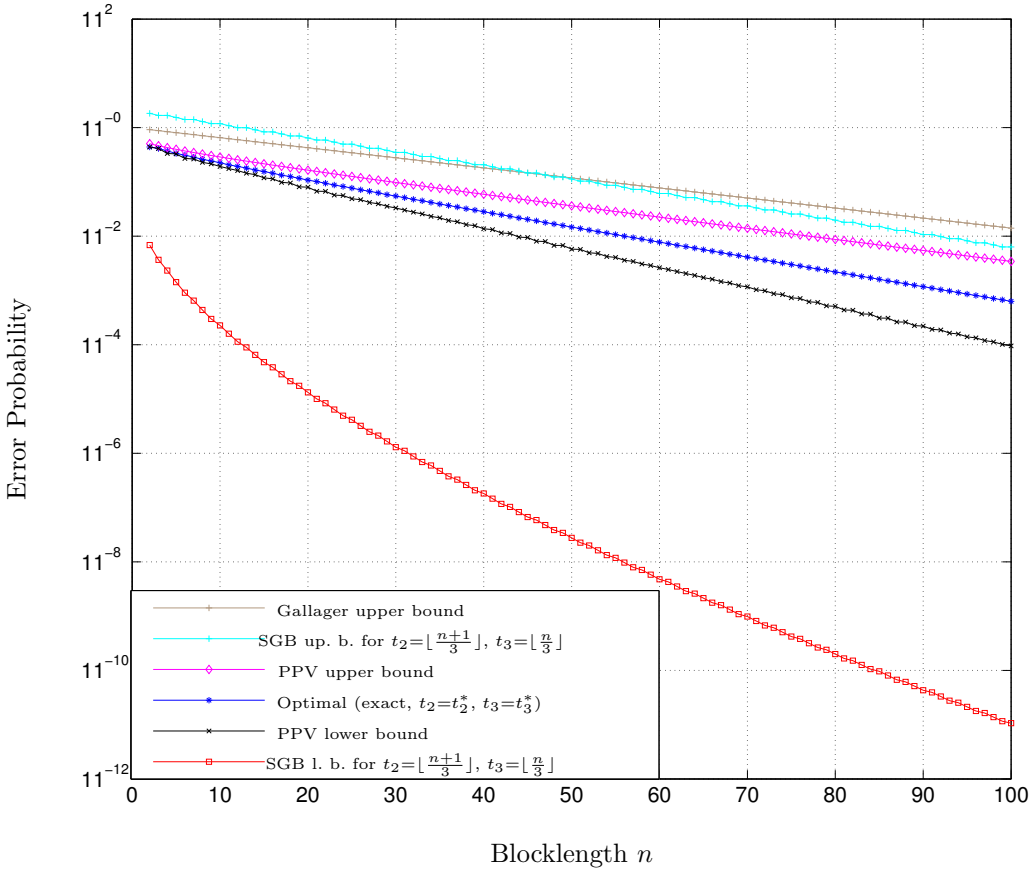


Figure 10.13: Exact value of, and bounds on, the performance of an optimal code with $\mathrm{M} = 3$ codewords on the BSC with $\epsilon = 0.3$ as a function of the blocklength n.

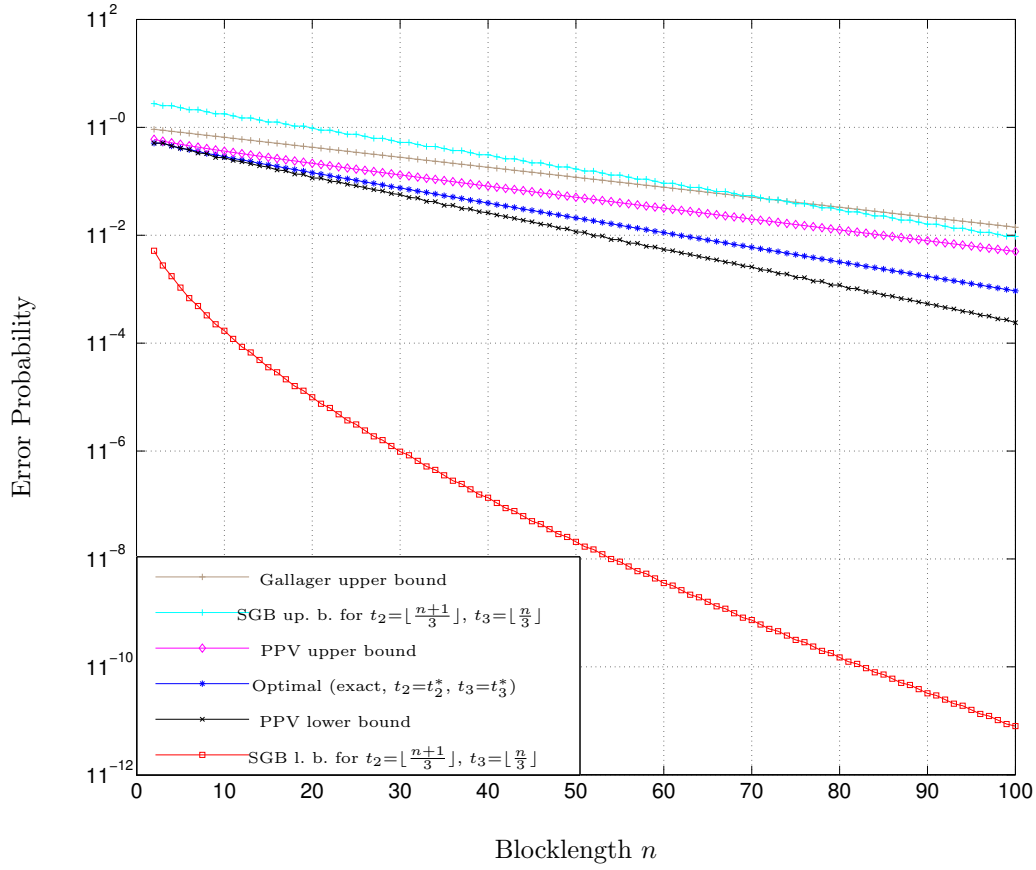


Figure 10.14: Exact value of, and bounds on, the performance of an optimal code with $\mathrm{M} = 4$ codewords on the BSC with $\epsilon = 0.3$ as a function of the blocklength n.

Chapter 11

Analysis of the BEC

The definition of the flip, the weak flip, and the fair weak flip codes is interesting not only due to their generalization of the concept of linear codes, but also because we can show that they are optimal for the BEC for many values of the blocklength n.

11.1 Optimal Codes with Two Codewords ($\mathsf{M} = 2$)

Theorem 11.1 *For a BEC and for any $n \geq 1$, an optimal codebook with $\mathsf{M} = 2$ codewords is the flip code of type t for any $t \in \{0, 1, \ldots, \lfloor \frac{n}{2} \rfloor\}$.*

Proof: Omitted. Similar argument as Theorem 10.1. □

11.2 Optimal Codes with Three or Four Codewords ($\mathsf{M} = 3, 4$)

Theorem 11.2 *For a BEC with arbitrary crossover probability $0 \leq \delta < 1$, the optimal code for $n = 2$ is*

$$\mathscr{C}_{\mathrm{BEC}}^{(\mathsf{M},2)*} = \begin{pmatrix} \mathbf{c}_1^{(\mathsf{M})} & \mathbf{c}_2^{(\mathsf{M})} \end{pmatrix}. \tag{11.1}$$

If we recursively construct a locally optimal codebook for $n \geq 3$ by using $\mathscr{C}_{\mathrm{BEC}}^{(\mathsf{M},n-1)*}$ and appending a new column, the total probability increase is maximized by the following choice of appended columns:

$$\begin{cases} \mathbf{c}_3^{(\mathsf{M})} & \text{if } n \bmod 3 = 0 \\ \mathbf{c}_1^{(\mathsf{M})} & \text{if } n \bmod 3 = 1 \\ \mathbf{c}_2^{(\mathsf{M})} & \text{if } n \bmod 3 = 2. \end{cases} \tag{11.2}$$

Proof: See Appendix D.1. □

While Theorem 11.2 only guarantees optimality under the condition that the optimal code are happened to be the code maximizing the total probability increase that constructed recursively.

Theorem 11.3 *For a BEC and for any $n \geq 2$, an optimal codebook with $\mathsf{M} = 3$ or a linear optimal codebook with $\mathsf{M} = 4$ codewords is the weak flip code of type (t_2^*, t_3^*), where*

$$t_2^* \triangleq \left\lfloor \frac{n+1}{3} \right\rfloor, \quad t_3^* \triangleq \left\lfloor \frac{n}{3} \right\rfloor. \tag{11.3}$$

Using the shorthands

$$k \triangleq \left\lfloor \frac{n}{3} \right\rfloor \tag{11.4}$$

the code parameters of these optimal codes can be written as

$$[t_1^*, t_2^*, t_3^*] = \begin{cases} [k, k, k] & \text{if } n \bmod 3 = 0 \\ [k+1, k, k] & \text{if } n \bmod 3 = 1 \\ [k+1, k+1, k] & \text{if } n \bmod 3 = 2. \end{cases} \tag{11.5}$$

Furthermore, the exact average success probability can be derived recursively in blocklength n, starting with

$$3P_{\mathrm{c}}(\mathscr{C}_{\mathrm{BEC}}^{(3,2)*}) = 3(1-\delta)^2 + 4\delta(1-\delta) + \delta^2. \tag{11.6}$$

Then

$$3P_{\mathrm{c}}(\mathscr{C}_{t_2^*,t_3^*}^{(3,n)}) = 3P_{\mathrm{c}}(\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}) + \left(\delta^{2k-1} + \delta^{2k-1} - \delta^{n-1}\right); \text{(if } n = 3k\text{)} \tag{11.7}$$

$$3P_{\mathrm{c}}(\mathscr{C}_{t_2^*,t_3^*}^{(3,n)}) = 3P_{\mathrm{c}}(\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}) + \left(\delta^{2k} + \delta^{2k} - \delta^{n-1}\right); \quad \text{(if } n = 3k+1\text{)} \tag{11.8}$$

$$3P_{\mathrm{c}}(\mathscr{C}_{t_2^*,t_3^*}^{(3,n)}) = 3P_{\mathrm{c}}(\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}) + \left(\delta^{2k} + \delta^{2k+1} - \delta^{n-1}\right). \quad \text{(if } n = 3k+2\text{)} \tag{11.9}$$

While for $\mathsf{M} = 4$ the exact average success probability is very similar to the case of $\mathsf{M} = 3$.

Proof: Similar to the proof of Theorem 10.3, combing with the proof of Theorem 11.2. □

Note that the idea of designing an optimal code recursively promises to be a very powerful approach. Unfortunately, for larger values of M, we might need a recursion from n to $n+\gamma$ with a step-size $\gamma > 1$, and this step-size γ might be a function of blocklength n. However, based on our definition of fair weak flip codes and on Conjecture 11.5 below, we conjecture that the necessary step-size satisfies $\gamma \leq \binom{2\ell-1}{\ell}$.

We have conjectured that this recursive approach also to the cases of $\mathsf{M} = 5$ and $\mathsf{M} = 6$.

Conjecture 11.4 *For a BEC and for any $n \geq 3$, if the optimal codebook can be recursively constructed in blocklength n, an optimal codebook with $\mathsf{M} = 5$ codewords can be constructed recursively in the blocklength n. We start with an optimal codebook for $n = 3$:*

$$\mathscr{C}_{\mathrm{BEC}}^{(\mathsf{M},3)*} = \left(\mathbf{c}_1^{(\mathsf{M})}, \mathbf{c}_2^{(\mathsf{M})}, \mathbf{c}_5^{(\mathsf{M})}\right) \tag{11.10}$$

and recursively construct the optimal codebook for $n \geq 5$ by using $\mathscr{C}_{\mathrm{BEC}}^{(\mathsf{M},n-\gamma)*}$, $\gamma \in \{1,2,3\}$, and appending

$$\begin{cases} \left(\mathbf{c}_1^{(\mathsf{M})}, \mathbf{c}_2^{(\mathsf{M})}, \mathbf{c}_5^{(\mathsf{M})}\right) & \text{if } n \bmod 10 = 3, \\ \left(\mathbf{c}_3^{(\mathsf{M})}, \mathbf{c}_6^{(\mathsf{M})}\right) & \text{if } n \bmod 10 = 5, \\ \left(\mathbf{c}_9^{(\mathsf{M})}, \mathbf{c}_{10}^{(\mathsf{M})}\right) & \text{if } n \bmod 10 = 7, \\ \left(\mathbf{c}_4^{(\mathsf{M})}, \mathbf{c}_7^{(\mathsf{M})}\right) & \text{if } n \bmod 10 = 9, \\ \mathbf{c}_8^{(\mathsf{M})} & \text{if } n \bmod 10 = 0. \end{cases} \tag{11.11}$$

For $\mathsf{M} = 6$ codewords, an optimal codebook can be constructed recursively in the blocklength n by starting with an optimal codebook for $n = 4$:

$$\mathscr{C}_{\mathrm{BEC}}^{(\mathsf{M},3)*} = \left(\mathbf{c}_1^{(\mathsf{M})}, \mathbf{c}_2^{(\mathsf{M})}, \mathbf{c}_6^{(\mathsf{M})}, \mathbf{c}_8^{(\mathsf{M})}\right). \tag{11.12}$$

Then we recursively construct the optimal codebook for $n \geq 6$ by using $\mathscr{C}_{\mathrm{BEC}}^{(\mathsf{M},n-2)*}$ and appending

$$\begin{cases} \left(\mathbf{c}_1^{(\mathsf{M})}, \mathbf{c}_2^{(\mathsf{M})}\right) & \text{if } n \bmod 10 = 2, \\ \left(\mathbf{c}_6^{(\mathsf{M})}, \mathbf{c}_8^{(\mathsf{M})}\right) & \text{if } n \bmod 10 = 4, \\ \left(\mathbf{c}_3^{(\mathsf{M})}, \mathbf{c}_5^{(\mathsf{M})}\right) & \text{if } n \bmod 10 = 6, \\ \left(\mathbf{c}_4^{(\mathsf{M})}, \mathbf{c}_7^{(\mathsf{M})}\right) & \text{if } n \bmod 10 = 8, \\ \left(\mathbf{c}_9^{(\mathsf{M})}, \mathbf{c}_{10}^{(\mathsf{M})}\right) & \text{if } n \bmod 10 = 0. \end{cases} \tag{11.13}$$

For space reasons we omit the proof and only remark once again that the ideas of the derivation follow the same ideas as shown above in Lemma 9.3 and Claim D.1.

An interesting special case of Conjecture 11.4 is as follows.

Conjecture 11.5 *For a BEC and for any n being a multiple of 10, an optimal codebook with $\mathsf{M} = 5$ or $\mathsf{M} = 6$ codewords is the corresponding fair weak flip code.*

Note that the restriction on n comes from the restriction that fair weak flip codes are only defined for n with $n \bmod \binom{2\ell-1}{\ell} = n \bmod 10 = 0$. Even though Conjecture 11.5 actually follows as special case from Conjecture 11.4, it can be proven directly and more elegantly using the properties of fair weak flip codes derived in Section 6.1.

How about the optimal codes on BEC for higher number of codewords M? We strongly believe that Conjecture 11.5 can be generalized to arbitrary M.

Conjecture 11.6 *For a BEC and for an arbitrary M, the optimal code for a blocklength n that satisfies $n \bmod \binom{2\ell-1}{\ell} = 0$ is the corresponding fair weak flip code.*

11.3 Quick Comparison between BSC and BEC

It has been shown that optimal codes for $\mathsf{M} = 3$ or linear optimal codes for $\mathsf{M} = 4$ are weak flip codes with code parameters:

$$[t_1^*, t_2^*, t_3^*] = \begin{cases} [k+1, k, k-1] & \text{if } n \bmod 3 = 0, \\ [k+1, k, k] & \text{if } n \bmod 3 = 1, \\ [k+1, k+1, k] & \text{if } n \bmod 3 = 2, \end{cases} \tag{11.14}$$

where we use

$$k \triangleq \left\lfloor \frac{n}{3} \right\rfloor. \tag{11.15}$$

The corresponding pairwise Hamming distance vectors (see Lemma 6.7) are

$$\begin{cases} (2k-1, 2k, 2k+1) & \text{if } n \bmod 3 = 0, \\ (2k, 2k+1, 2k+1) & \text{if } n \bmod 3 = 1, \\ (2k+1, 2k+1, 2k+2) & \text{if } n \bmod 3 = 2. \end{cases} \tag{11.16}$$

If we compare this to Theorem 11.2:

$$[t_1^*, t_2^*, t_3^*] = \begin{cases} [k, k, k] & \text{if } n \bmod 3 = 0, \\ [k+1, k, k] & \text{if } n \bmod 3 = 1, \\ [k+1, k+1, k] & \text{if } n \bmod 3 = 2 \end{cases} \tag{11.17}$$

with corresponding pairwise Hamming distance vectors

$$\begin{cases} (2k, 2k, 2k) & \text{if } n \bmod 3 = 0, \\ (2k, 2k+1, 2k+1) & \text{if } n \bmod 3 = 1, \\ (2k+1, 2k+1, 2k+2) & \text{if } n \bmod 3 = 2, \end{cases} \tag{11.18}$$

we can conclude the following.

Corollary 11.7 *Apart from $n \bmod 3 = 0$, the optimal codes for a BSC are identical to the optimal codes for a BEC for $\mathsf{M} = 3$ or linear optimal codes for $\mathsf{M} = 4$ codewords.*

It is interesting to note that for $n \bmod 3 = 0$ the optimal codes for the BEC are fair and therefore maximize the minimum Hamming distance, while this is not the case for the (very symmetric!) BSC. However, note that the converse is *not* true: if a code maximizes the minimum Hamming distance, then it is not necessarily an optimal code for the BEC! So, in particular, it is not clear if binary nonlinear Hadamard codes are optimal.

11.4 Application to Known Bounds on the Error Probability for a Finite Blocklength

We again provide a comparison between the performance of the optimal code to the known bounds of Chapter 7.

Note that the error exponents for $\mathsf{M} = 3, 4$ codewords are

$$\mathsf{E}_3 = \mathsf{E}_4 = -\frac{2}{3}\log\delta. \tag{11.19}$$

Moreover, for $\mathsf{M} = 3, 4$,

$$\begin{aligned}&\mathsf{D}_{\min}^{(\mathrm{BEC})}\left(\mathscr{C}^{(\mathsf{M},n)}_{\lfloor\frac{n+1}{3}\rfloor,\lfloor\frac{n}{3}\rfloor}\right)\\ &= \begin{cases} -\frac{2}{3}\log\delta & \text{if } n \bmod 3 = 0\\ -\frac{\lfloor\frac{n}{3}\rfloor+\lfloor\frac{n+1}{3}\rfloor}{n}\log\delta & \text{if } n \bmod 3 = 1\\ -\frac{\lfloor\frac{n}{3}\rfloor+\lfloor\frac{n+1}{3}\rfloor}{n}\log\delta & \text{if } n \bmod 3 = 2. \end{cases}\end{aligned} \tag{11.20}$$

Figs. 11.15 and 11.16 compare the exact optimal performance for $\mathsf{M} = 3$ and $\mathsf{M} = 4$, respectively, with some bounds: the SGB upper bound based on the weak flip code used by Shannon *et al.*,[10], the SGB lower bound based on the weak flip code (which is suboptimal, but achieves the optimal $\mathsf{D}_{\min}^{(\mathrm{DMC})}$ and is therefore a generally valid lower bound), the Gallager upper bound, and also the PPV upper and lower bounds.

We can see that the SGB upper bound is tighter to the exact optimal performance than the PPV upper bound. Note, however, the PPV upper bound does not exhibit the correct error exponent. It is shown in [20] that, for n going to infinity, the random coding (PPV) upper bound tends to the Gallager exponent for $R = 0$ [6], which is of course not necessarily equal to E_{M} for finite M.

Concerning the lower bounds, we see that the PPV lower bound (converse) is much better for finite n than the SGB bound. However, for n large enough, its exponential growth will approach that of the sphere-packing bound [17], which does not equal to E_{M} either.

Once more we would like to point out that even though the fair weak flip codes achieve the error exponent, they are optimal codes in the BEC, however, they are strictly suboptimal for every $n \bmod 3 = 0$ in the BSC.

[10]The SGB upper bound based on the optimal code performs almost identically (because the BSC is pairwise reversible) and is therefore omitted.

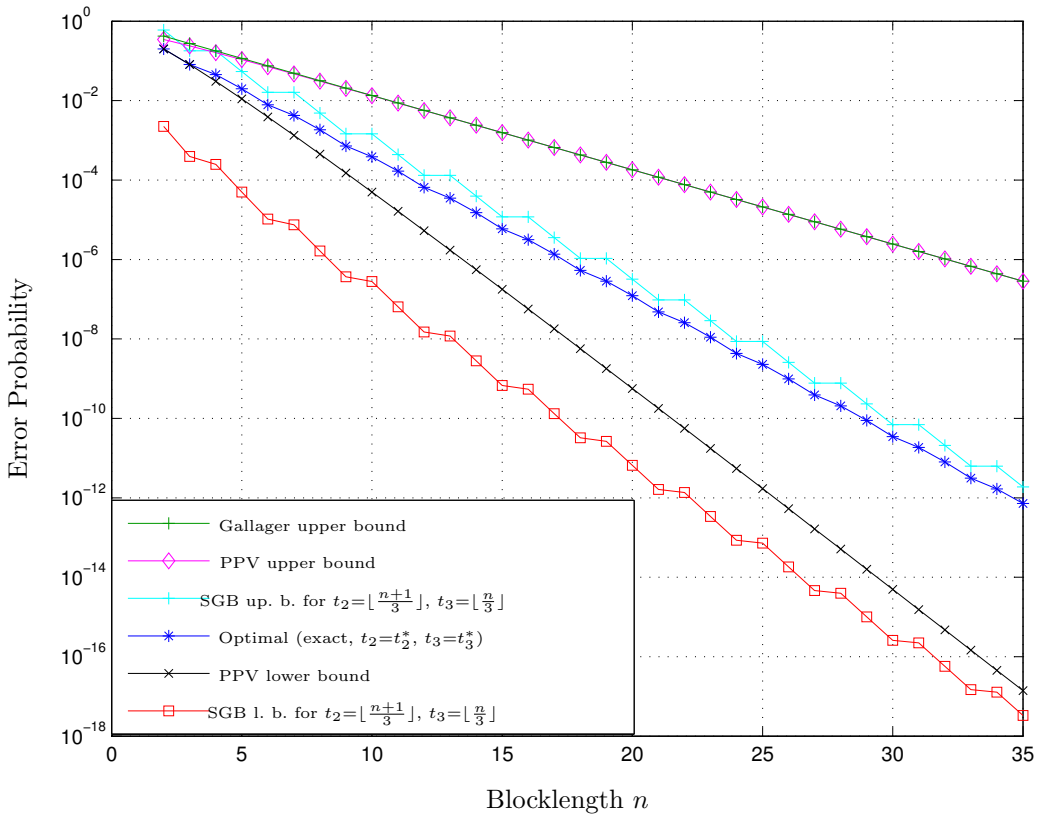


Figure 11.15: Exact value of, and bounds on, the performance of an optimal code with $\mathrm{M} = 3$ codewords on the BEC with $\delta = 0.3$ as a function of the blocklength n.

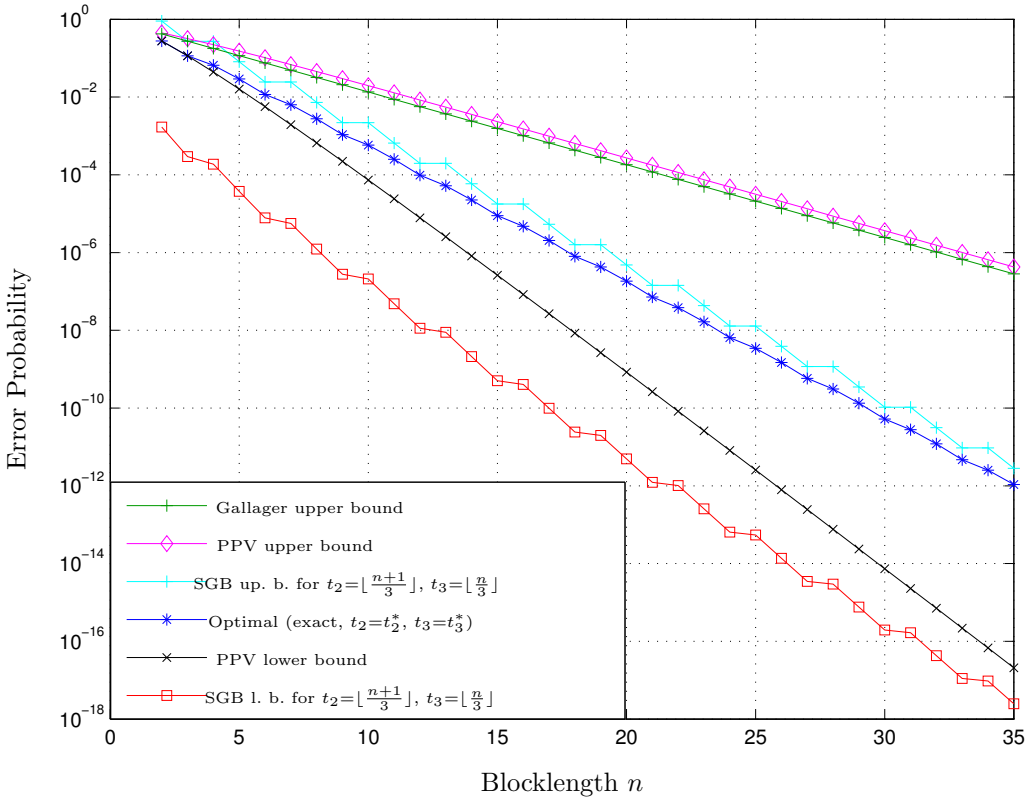


Figure 11.16: Exact value of, and bounds on, the performance of an optimal code with $\mathrm{M} = 4$ codewords on the BEC with $\delta = 0.3$ as a function of the blocklength n.

Chapter 12

Conclusion

For an arbitrary finite blocklength n, we have studied the optimal code design of ultra-small block-codes for the most general binary input and the binary output discrete memoryless channel, the so-called *binary asymmetric channel (BAC)*, and the binary input and ternary output symmetric discrete memoryless channel: the *binary erasure channel (BEC)*.

We then have put special emphasis on the two most important special cases of binary channels, the *Z-channel (ZC)* and the *binary symmetric channel (BSC)*. There, again for an arbitrary finite blocklength n, we have derived an optimal code design with four or less messages. In the case of the ZC, we have also conjectured an optimal code design with five messages. Note that since the optimal codes we proposed do not depend on the crossover probability of the channel the optimal codes remain the same even if the channel is nonergodic or nonstationary. Also note that the optimal weak flip codes are by definition coset codes: the $\mathsf{M} = 3$ nonlinear code is always a coset of the $\mathsf{M} = 4$ linear code. However, they are not fixed composition codes.

We have introduced a new way of generating these codes recursively by using a column-wise build-up of the codebook matrix. This column view of the codebook turns out to be far more powerful for analysis than the standard row-wise view (i.e., the analysis based on the codewords). We believe that the recursive construction of codes may be extended to a higher number of codewords and also to more complex channel models. Indeed, we have achieved some first promising results for the binary erasure channel (BEC). Note, however, that in these more complex situations we might need a recursion from n to $n+\gamma$ with a step-size $\gamma > 1$.

We have also investigated the well-known and commonly used code parameter *minimum Hamming distance*. We show that it may not be suitable as a design criterion for optimal codes, even for very symmetric channels like the BSC.

Finally, we would like to point out that the family of weak flip codes defined in Chapter 6 (and in particular a subfamily called *fair weak flip codes*) turn out to have many interesting properties. They can be seen as a subset of a well-known codes: the Hadamard codes. A first closer investigation of some of these properties and these codes' relation to linear codes have also been presented.

Appendix A

Derivations concerning the BAC

A.1 Proof of Proposition 4.1

Let $P_X(0) = p$, then $P_X(1) = 1 - p$, we have

$$P_Y(0) = P_X(0)P_{Y|X}(0|0) + P_X(1)P_{Y|X}(0|1) = p(1-\epsilon_0) + (1-p)\epsilon_1, \tag{A.1}$$
$$P_Y(1) = P_X(0)P_{Y|X}(1|0) + P_X(1)P_{Y|X}(1|1) = p\epsilon_0 + (1-p)(1-\epsilon_1). \tag{A.2}$$

Then the mutual information $I(X;Y)$ of BAC

$$I(X;Y) = H(Y) - H(Y|X) \tag{A.3}$$
$$= -P_Y(0)\log P_Y(0) - P_Y(1)\log P_Y(1) - \sum_{x=0,1} P_X(x)H(Y|X=x) \tag{A.4}$$
$$\begin{aligned} &= -\big[p(1-\epsilon_0) + (1-p)\epsilon_1\big]\log\big[p(1-\epsilon_0) + (1-p)\epsilon_1\big] \\ &\quad -\big[p\epsilon_0 + (1-p)(1-\epsilon_1)\big]\log\big[p\epsilon_0 + (1-p)(1-\epsilon_1)\big] \\ &\quad -pH_{\mathrm{b}}(\epsilon_0)\log 2 - (1-p)H_{\mathrm{b}}(\epsilon_1)\log 2 \quad \text{(nats)}. \end{aligned} \tag{A.5}$$

Therefore, take the derivative of the mutual information, we get

$$\begin{aligned} &-\big[1-\epsilon_0-\epsilon_1\big]\log\big[p(1-\epsilon_0) + (1-p)\epsilon_1\big] - 1\cdot\big(1-\epsilon_0-\epsilon_1\big) \\ &-\big[\epsilon_0 - (1-\epsilon_1)\big]\log\big[p\epsilon_0 + (1-p)(1-\epsilon_1)\big] - 1\cdot\big(\epsilon_0 - (1-\epsilon_1)\big) \\ &-H_{\mathrm{b}}(\epsilon_0)\log 2 + H_{\mathrm{b}}(\epsilon_1)\log 2 \stackrel{!}{=} 0 \end{aligned} \tag{A.6}$$
$$\Longrightarrow (1-\epsilon_0-\epsilon_1)\left(\log_2\frac{p(1-\epsilon_0)+(1-p)\epsilon_1}{p\epsilon_0+(1-p)(1-\epsilon_1)}\right) = H_{\mathrm{b}}(\epsilon_1) - H_{\mathrm{b}}(\epsilon_0) \tag{A.7}$$
$$\Longrightarrow \frac{p(1-\epsilon_0)+(1-p)\epsilon_1}{p\epsilon_0+(1-p)(1-\epsilon_1)} = 2^{\frac{H_{\mathrm{b}}(\epsilon_1)-H_{\mathrm{b}}(\epsilon_0)}{1-\epsilon_0-\epsilon_1}} \tag{A.8}$$
$$\Longrightarrow \frac{(1-\epsilon_0)+(\frac{1-p}{p})\epsilon_1}{\epsilon_0+(\frac{1-p}{p})(1-\epsilon_1)} = z \Longrightarrow (\frac{1-p}{p})(\epsilon_1 - z(1-\epsilon_1)) = z\epsilon_0 - (1-\epsilon_0) \tag{A.9}$$
$$\Longrightarrow \frac{1}{p} - 1 = \frac{1-\epsilon_0-z\epsilon_0}{z(1-\epsilon_1)-\epsilon_1} \tag{A.10}$$

$$\Longrightarrow \frac{1}{p} = \frac{z(1-\epsilon_1) - \epsilon_1 + 1 - \epsilon_0 - z\epsilon_0}{z(1-\epsilon_1) - \epsilon_1} \tag{A.11}$$

$$\Longrightarrow p = \frac{z - \epsilon_1(1+z)}{(1+z)(1-\epsilon_1-\epsilon_0)}. \tag{A.12}$$

Hecne, the capacity-input achieving distributions are

$$P_X^*(0) = \frac{z - \epsilon_1(1+z)}{(1+z)(1-\epsilon_0-\epsilon_1)}, \qquad P_X^*(1) = \frac{1 - \epsilon_0(1+z)}{(1+z)(1-\epsilon_0-\epsilon_1)} \tag{A.13}$$

Similarly, the capacity-output achieving distributions are

$$P_Y^*(0) = \frac{(1+z)(1-\epsilon_0-\epsilon_1) - (1-\epsilon_0-\epsilon_1)}{(1+z)(1-\epsilon_0-\epsilon_1)} = \frac{z}{1+z}$$

$$P_Y^*(1) = \frac{1}{1+z}. \tag{A.14}$$

Next we substitute this p into $I(X;Y)$ (A.5), we have

$$\mathrm{C_{BAC}} = -\frac{z}{1+z}\log_2\left(\frac{z}{1+z}\right) - \frac{1}{1+z}\log_2\frac{1}{1+z} - P_X^*(0)H_\mathrm{b}(\epsilon_0) - P_X^*(1)H_\mathrm{b}(\epsilon_1) \tag{A.15}$$

$$= -\frac{z}{1+z}\log_2 2^{\frac{H_\mathrm{b}(\epsilon_1)-H_\mathrm{b}(\epsilon_0)}{1-\epsilon_0-\epsilon_1}} + \log_2(1+z) - \frac{z-\epsilon_1(1+z)}{(1+z)(1-\epsilon_0-\epsilon_1)}H_\mathrm{b}(\epsilon_0) - \frac{1-\epsilon_0(1+z)}{(1+z)(1-\epsilon_0-\epsilon_1)}H_\mathrm{b}(\epsilon_1) \tag{A.16}$$

$$= \left[\frac{z}{(1+z)(1-\epsilon_0-\epsilon_1)} - \frac{z-\epsilon_1(1+z)}{(1+z)(1-\epsilon_0-\epsilon_1)}\right]H_\mathrm{b}(\epsilon_0) + \left[-\frac{z}{(1+z)(1-\epsilon_0-\epsilon_1)} - \frac{1-\epsilon_0(1+z)}{(1+z)(1-\epsilon_0-\epsilon_1)}\right]H_\mathrm{b}(\epsilon_1) + \log_2(1+z) \tag{A.17}$$

$$= \frac{\epsilon_1}{1-\epsilon_0-\epsilon_1}H_\mathrm{b}(\epsilon_0) - \frac{1-\epsilon_0}{1-\epsilon_0-\epsilon_1}H_\mathrm{b}(\epsilon_1) + \log_2(1+z). \tag{A.18}$$

A.2 The LLR Function

Proposition A.1 (Properties of $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$)

1. *If $\epsilon_0+\epsilon_1=1$, then $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)=0$ irrespective of d, t, or n.*
2. *$\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$ is a nonincreasing function in d for every n, t:*

$$\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d) \le \mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d-1), \quad 1 \le d \le n. \tag{A.19}$$

3. *For certain values of d, the value of $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$ is always nonnegative (or always nonpositive) for all ϵ_0 and ϵ_1:*

$$\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d) \begin{cases} \ge 0 & \text{if } 0 \le d \le t \\ \lessgtr 0 & \text{if } t < d \le \left\lfloor \frac{n}{2} \right\rfloor \text{ (depending on } \epsilon_0,\ \epsilon_1) \\ \le 0 & \text{if } \left\lfloor \frac{n}{2} \right\rfloor < d \le n. \end{cases} \tag{A.20}$$

4. $\mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, d)$ *is a nondecreasing function in* n *for fixed* t, d, *and* (ϵ_0, ϵ_1).

5. $\mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, d)$ *is a nondecreasing function in* t *for fixed* n, d, *and* (ϵ_0, ϵ_1).

6. *For* $0 \le d \le n$,

$$\mathrm{LLR}_t^{(n+1)}(\epsilon_0, \epsilon_1, d+1) < \mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, d). \tag{A.21}$$

Proof: These properties follow quite easily from the definition of $\mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, d)$ and the relations (3.1)–(3.3). We only show a proof of the second property:

$$\begin{aligned} &\mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, d-1) - \mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, d) \\ &= \log\left(\frac{1-\epsilon_1}{\epsilon_0}\right) + \log\left(\frac{1-\epsilon_0}{\epsilon_1}\right) \ge 0. \end{aligned} \tag{A.22}$$

□

A.3 Alternative Proof of Theorem 8.1

Assume that the optimal code for blocklength n is not a flip code. Then the code has a number j of positions where both codewords have the same symbol. The optimal decoder will ignore these j positions completely. Hence, the performance of this code will be identical to a flip code of length $n-j$.

We therefore only need to show that increasing n will always allow us to find a new flip code with a better performance. In other words, Theorem 8.1 is proven once we have shown that

$$P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n-1)}\big) \ge \max\left\{P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n)}\big), P_{\mathrm{e}}\big(\mathscr{C}_{t+1}^{(2,n)}\big)\right\}. \tag{A.23}$$

Note that for the length-$(n-1)$ flip code of type t

$$\mathscr{C}_t^{(2,n-1)} = \begin{pmatrix} \mathbf{x}_1^{(n-1)} \\ \mathbf{x}_2^{(n-1)} \end{pmatrix} \tag{A.24}$$

we can derive two nontrivial length-n codes:

$$\mathscr{C}_t^{(2,n)} = \begin{pmatrix} [\mathbf{x}_1^{(n-1)} \ 0] \\ [\mathbf{x}_2^{(n-1)} \ 1] \end{pmatrix}, \quad \mathscr{C}_{t+1}^{(2,n)} = \begin{pmatrix} [\mathbf{x}_1^{(n-1)} \ 1] \\ [\mathbf{x}_2^{(n-1)} \ 0] \end{pmatrix}. \tag{A.25}$$

Both of these codes happen to be (or at least be equivalent to) flip codes. We would like to remind the reader that $\mathbf{x}_2^{(n-1)}$ is a flipped version of $\mathbf{x}_1^{(n-1)}$.

Since in the following we are going to compare different flip codes of either length $n-1$ or n, we need to clarify our notation. So for the received vectors $\mathbf{y}^{(n)}$ we use a superscript (n) to denote their length, and for the codewords $\mathbf{x}_m^{(n)}$, optimal decoding threshold $\ell^{(n)}$, and the Hamming distance $d^{(n)}$ between a received sequence and the first codeword we

use the superscript (n) to denote their affiliation with the corresponding code of length n. Hence, as shown in Cor. 8.2, the optimal ML decision rule for $\mathscr{C}_t^{(2,n)}$ can be expressed as

$$g(\mathbf{y}) = \begin{cases} 1 & \text{if } 0 \le d^{(n)} \le \ell^{(n)} \\ 2 & \text{if } \ell^{(n)} + 1 \le d^{(n)} \le n. \end{cases} \tag{A.26}$$

The threshold satisfies $0 \le \ell^{(n)} \le \lfloor \frac{n-1}{2} \rfloor$. Note that when $\ell^{(n)} = \lfloor \frac{n-1}{2} \rfloor$, the decision rule is equivalent to a majority rule. Also note that when n is even and $d^{(n)} = \frac{n}{2}$, the decisions for $\mathbf{x}_1^{(n)}$ and $\mathbf{x}_2^{(n)}$ are equally likely, i.e., without loss of generality we then always decode to $\mathbf{x}_2^{(n)}$.

So let the threshold for $\mathscr{C}_t^{(2,n-1)}$ be $\ell^{(n-1)}$. We will now argue that the threshold for $\mathscr{C}_t^{(2,n)}$ and $\mathscr{C}_{t+1}^{(2,n)}$ (compare with (A.25)) must satisfy

$$\ell^{(n)} = \ell^{(n-1)} \quad \text{or} \quad \ell^{(n)} = \ell^{(n-1)} + 1. \tag{A.27}$$

Consider firstly the code $\mathscr{C}_t^{(2,n)}$ and assume by contradiction for the moment that $\ell^{(n)} < \ell^{(n-1)}$. Then pick a received $\mathbf{y}^{(n-1)}$ with $d^{(n-1)} = \ell^{(n-1)}$ that (for the code $\mathscr{C}_t^{(n-1)}$) is decoded to $\mathbf{x}_1^{(n-1)}$. The received length-n vector $\mathbf{y}^{(n)} = \big[\mathbf{y}^{(n-1)}\ 0\big]$ has $d^{(n)} = \ell^{(n-1)} > \ell^{(n)}$, i.e., it will be now decoded to $\mathbf{x}_2^{(n)}$. This, however, is a contradiction to the assumption that the ML decision for the code $\mathscr{C}_t^{(2,n-1)}$ was $\mathbf{x}_1^{(n-1)}$.

Secondly, again considering code $\mathscr{C}_t^{(2,n)}$, assume by contradiction that $\ell^{(n)} > \ell^{(n-1)} + 1$. Pick a received $\mathbf{y}^{(n-1)}$ with $d^{(n-1)} = \ell^{(n-1)} + 1$ that (for the code $\mathscr{C}_t^{(2,n-1)}$) is decoded to $\mathbf{x}_2^{(n-1)}$. The received length-n vector $\mathbf{y}^{(n)} = \big[\mathbf{y}^{(n-1)}\ 1\big]$ has $d^{(n)} = \ell^{(n-1)} + 2 < \ell^{(n)} + 1$, i.e., it will be now decoded to $\mathbf{x}_1^{(n)}$. This, however, is a contradiction to the assumption that the ML decision for the code $\mathscr{C}_t^{(2,n-1)}$ was $\mathbf{x}_1^{(n-1)}$.

The same arguments also hold for the other code $\mathscr{C}_{t+1}^{(2,n)}$. Hence, we see that there are only two possible changes with respect to the decoding rule to be considered. We will next use this fact to prove that $P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n-1)}\big) \ge P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n)}\big)$.

The error probability of a length-n code with two codewords $\mathbf{x}_1$ and $\mathbf{x}_2$ is given by

$$P_{\mathrm{e}} = \frac{1}{2} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})=2}} P_{Y|X}^n(\mathbf{y}|\mathbf{x}_1) + \frac{1}{2} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})=1}} P_{Y|X}^n(\mathbf{y}|\mathbf{x}_2). \tag{A.28}$$

For $\mathscr{C}_t^{(2,n-1)}$, (A.28) can be written as follows:

$$\begin{aligned} &2P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n-1)}\big) \\ &\quad = \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+1 \le d^{(n-1)} \le n-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big) \\ &\qquad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \le d^{(n-1)} \le \ell^{(n-1)}}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big) \\ &\quad = \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+1 \le d^{(n-1)} \le n-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big) P_{Y|X}(1|0) \end{aligned} \tag{A.29}$$

$$
\begin{aligned}
&+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+1 \le d^{(n-1)} \le n-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big) P_{Y|X}(0|0) \\
&+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \le d^{(n-1)} \le \ell^{(n-1)}}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big) P_{Y|X}(1|1) \\
&+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \le d^{(n-1)} \le \ell^{(n-1)}}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big) P_{Y|X}(0|1) \qquad \text{(A.30)} \\
&= \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+2 \le d^{(n)} \le n}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_1^{(n)}\Big) \\
&+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+1 \le d^{(n)} \le n-1}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big) \\
&+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 1 \le d^{(n)} \le \ell^{(n-1)}+1}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big) \\
&+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \le d^{(n)} \le \ell^{(n-1)}}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_2^{(n)}\Big). \qquad \text{(A.31)}
\end{aligned}
$$

Here, in (A.30) we use the fact that $P_{Y|X}(1|0)+P_{Y|X}(0|0) = 1$ and $P_{Y|X}(1|1)+P_{Y|X}(0|1) = 1$; and in (A.31) we combine the terms together using the definition of $\mathscr{C}_t^{(2,n)}$ according to (6.1) (and (A.25)).

We can now distinguish the two cases (A.27):

(i) If the decision rule for $\mathscr{C}_t^{(2,n)}$ is unchanged, i.e., $\ell^{(n)} = \ell^{(n-1)}$, we only need to take care of the third summation in (A.31) that contains some terms that will now be decoded differently. We split this sum up into two parts:

$$
\begin{aligned}
&\sum_{\substack{\mathbf{y}^{(n-1)} \\ 1 \le d^{(n)} \le \ell^{(n-1)}+1}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big) \\
&\quad= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n)} = \ell^{(n-1)}+1}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big) \\
&\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 1 \le d^{(n)} \le \ell^{(n-1)}}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big). \qquad \text{(A.32)}
\end{aligned}
$$

Since we have assumed that $\ell^{(n)} = \ell^{(n-1)}$, we know that for all $\mathbf{y}^{(n-1)}$ with $d^{(n-1)} = \ell^{(n-1)}$ the length-n received vector $\big[\mathbf{y}^{(n-1)}\ 1\big]$ has $d^{(n)} = \ell^{(n-1)} + 1 = \ell^{(n)} + 1$ and

will be decoded to $\mathbf{x}_2^{(n)}$. Hence we must have

$$\frac{P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_1^{(n)}\Big)}{P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big)} \le 1. \tag{A.33}$$

Hence, we have

$$\begin{aligned}
&2P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n-1)}\big) \\
&\quad\ge \sum_{\substack{\mathbf{y}^{(n-1)}\\ \ell^{(n-1)}+2\le d^{(n)}\le n}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_1^{(n)}\Big) \\
&\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)}\\ \ell^{(n-1)}+1\le d^{(n)}\le n-1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big) \\
&\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)}\\ d^{(n)}=\ell^{(n-1)}+1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_1^{(n)}\Big) \\
&\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)}\\ 1\le d^{(n)}\le \ell^{(n-1)}}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big) \\
&\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)}\\ 0\le d^{(n)}\le \ell^{(n-1)}}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_2^{(n)}\Big) \qquad\text{(A.34)} \\
&\quad= \sum_{\substack{\mathbf{y}^{(n)}\\ \ell^{(n-1)}+1\le d^{(n)}\le n}} P_{Y|X}^n\big(\mathbf{y}^{(n)}\big|\mathbf{x}_1^{(n)}\big) \\
&\qquad+ \sum_{\substack{\mathbf{y}^{(n)}\\ 0\le d^{(n)}\le \ell^{(n-1)}}} P_{Y|X}^n\big(\mathbf{y}^{(n)}\big|\mathbf{x}_2^{(n)}\big) \qquad\text{(A.35)} \\
&\quad= 2P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n)}\big). \qquad\text{(A.36)}
\end{aligned}$$

(ii) If the decision rule is changed such that $\ell^{(n)} = \ell^{(n-1)} + 1$, we need to take care of the second summation in (A.31) that contains some terms that will now be decoded differently. Again, we split this sum into two parts:

$$\begin{aligned}
&\sum_{\substack{\mathbf{y}^{(n-1)}\\ \ell^{(n-1)}+1\le d^{(n)}\le n-1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big) \\
&\quad= \sum_{\substack{\mathbf{y}^{(n-1)}\\ d^{(n)}=\ell^{(n-1)}+1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big) \\
&\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)}\\ \ell^{(n-1)}+2\le d^{(n)}\le n-1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big). \qquad\text{(A.37)}
\end{aligned}$$

Since we have assumed that $\ell^{(n)} = \ell^{(n-1)} + 1$, we know that for all $\mathbf{y}^{(n-1)}$ with $d^{(n-1)} = \ell^{(n-1)}+1$ the length-n received vector $\left[\mathbf{y}^{(n-1)}\ 0\right]$ has $d^{(n)} = \ell^{(n-1)}+1 = \ell^{(n)}$ and will be decoded to $\mathbf{x}_1^{(n)}$. Hence we must have

$$\frac{P_{Y|X}^n\left(\left[\mathbf{y}^{(n-1)}\ 0\right]\middle|\mathbf{x}_1^{(n)}\right)}{P_{Y|X}^n\left(\left[\mathbf{y}^{(n-1)}\ 0\right]\middle|\mathbf{x}_2^{(n)}\right)} \geq 1. \tag{A.38}$$

The rest of the argument now is analogous to Case (i).

This proves that $P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n-1)}\big) \geq P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n)}\big)$. The remaining proof of $P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n-1)}\big) \geq P_{\mathrm{e}}\big(\mathscr{C}_{t+1}^{(2,n)}\big)$ is similar and omitted.

We remark that while in general $P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n-1)}\big) \geq P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n)}\big)$, we only achieve equality if n is even and $\ell^{(n-1)} = \left\lfloor \frac{n-1}{2} \right\rfloor$. This long proof suggests that how to deal with the total probability increase if we know the ML decoder threshold. It is very helpful for the proof of Theorem 8.3. However, we don't have the closed form for the total probability increase when the threshold strongly depends on the blocklength n on BAC.

A.4 Proof of Theorem 8.3

In order to derive the error probability expressions for $\mathscr{C}_t^{(2,n)}$ and $\mathscr{C}_{t+1}^{(2,n)}$, we introduce the flip code $\mathscr{C}_t^{(2,n-1)}$ and add either a column $(0\ 1)^{\mathsf{T}}$ or $(1\ 0)^{\mathsf{T}}$, respectively. Moreover, we assume that $\mathscr{C}_t^{(2,n-1)}$ also is decoded using the same fixed threshold ℓ.

Note that since we are using a similar approach as in Appendix A.3, we also apply the notation introduced there, i.e., we use a superscript (n) to denote length and affiliation.

We now write the error probability of $\mathscr{C}_t^{(2,n)}$ for the given decoding rule ℓ as follows:

$$\begin{aligned}
&2P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_t^{(2,n)}\big) \\
&\quad= \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell+1\leq d^{(n-1)}\leq n-1}} P_{Y|X}^n\left(\left[\mathbf{y}^{(n-1)}\ 0\right]\middle|\left[\mathbf{x}_1^{(n-1)}\ 0\right]\right) \\
&\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell\leq d^{(n-1)}\leq n-1}} P_{Y|X}^n\left(\left[\mathbf{y}^{(n-1)}\ 1\right]\middle|\left[\mathbf{x}_1^{(n-1)}\ 0\right]\right) \\
&\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0\leq d^{(n-1)}\leq \ell}} P_{Y|X}^n\left(\left[\mathbf{y}^{(n-1)}\ 0\right]\middle|\left[\mathbf{x}_2^{(n-1)}\ 1\right]\right) \\
&\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0\leq d^{(n-1)}\leq \ell-1}} P_{Y|X}^n\left(\left[\mathbf{y}^{(n-1)}\ 1\right]\middle|\left[\mathbf{x}_2^{(n-1)}\ 1\right]\right) \\
&\quad= \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell+1\leq d^{(n-1)}\leq n-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)(1-\epsilon_0+\epsilon_0)
\end{aligned} \tag{A.39}$$

$$
\begin{aligned}
&+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)}=\ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\epsilon_0 \\
&+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0\le d^{(n-1)}\le \ell-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)(\epsilon_1+1-\epsilon_1) \\
&+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)}=\ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)\epsilon_1. \qquad \text{(A.40)}
\end{aligned}
$$

Similarly, we can express the error probability of $\mathscr{C}_{t+1}^{(2,n)}$:

$$
\begin{aligned}
&2P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_{t+1}^{(2,n)}\big) \\
&= \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell+1\le d^{(n-1)}\le n-1}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\big[\mathbf{x}_1^{(n-1)}\ 1\big]\Big) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell\le d^{(n-1)}\le n-1}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\big[\mathbf{x}_1^{(n-1)}\ 1\big]\Big) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0\le d^{(n-1)}\le \ell}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\big[\mathbf{x}_2^{(n-1)}\ 0\big]\Big) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0\le d^{(n-1)}\le \ell-1}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\big[\mathbf{x}_2^{(n-1)}\ 0\big]\Big) \qquad \text{(A.41)} \\
&= \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell+1\le d^{(n-1)}\le n-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)(1-\epsilon_1+\epsilon_1) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)}=\ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\epsilon_1 \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0\le d^{(n-1)}\le \ell-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)(\epsilon_0+1-\epsilon_0) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)}=\ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)\epsilon_0. \qquad \text{(A.42)}
\end{aligned}
$$

Subtracting (A.42) from (A.40) then yields

$$
\begin{aligned}
&2P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_{t}^{(2,n)}\big) - 2P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_{t+1}^{(2,n)}\big) \\
&= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)}=\ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\epsilon_0 \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)}=\ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)\epsilon_1
\end{aligned}
$$

$$
\begin{aligned}
&- \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)}=\ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\epsilon_1 \\
&- \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)}=\ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)\epsilon_0 \qquad \text{(A.43)} \\
&= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)}=\ell}} \Big(P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big) \\
&\qquad - P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\Big)(\epsilon_1-\epsilon_0) \qquad \text{(A.44)} \\
&= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)}=\ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big) \\
&\qquad \cdot \left(1 - \frac{P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)}{P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)}\right)(\epsilon_1-\epsilon_0) \qquad \text{(A.45)} \\
&= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)}=\ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big) \\
&\qquad \cdot \Big(1 - e^{\mathrm{LLR}_t^{(n-1)}(\epsilon_0,\epsilon_1,\ell)}\Big)(\epsilon_1-\epsilon_0) \qquad \text{(A.46)} \\
&= \Big(1 - e^{\mathrm{LLR}_t^{(n-1)}(\epsilon_0,\epsilon_1,\ell)}\Big) \\
&\cdot (\epsilon_1-\epsilon_0) \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)}=\ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big) \qquad \text{(A.47)}
\end{aligned}
$$

where in (A.46) we make use of our assumption that $\mathscr{C}_t^{(2,n-1)}$ is decoded also using the same threshold ℓ.

Hence, we see that unless $\epsilon_0 = \epsilon_1$, in which case the difference is always zero, $2P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_t^{(2,n)}\big) - 2P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_{t+1}^{(2,n)}\big)$ can only be zero if

$$\mathrm{LLR}_t^{(n-1)}(\epsilon_0,\epsilon_1,\ell) = 0. \qquad \text{(A.48)}$$

From the definition of the log-likelihood ratio, we see that if we fix ϵ_0, then there exists at most one ϵ_1 such that (A.48) is satisfied. The same is true if we fix ϵ_1 and search for an ϵ_0.

Appendix B

Derivations concerning the ZC

B.1 Proof of Theorem 9.2

We first start with a general lemma that helps to reduce the size of the set of candidate columns that needs to be searched for the optimal codebooks of the ZC.

Lemma B.1 (Sufficient Set of Candidate Columns for the ZC) *For a ZC, for any blocklength n, and for an arbitrary number M of codewords, an optimal codebook must contain the all-zero codeword $\mathbf{0}$.*

Proof: Consider a general codebooks matrix $\mathscr{C}^{(\mathsf{M},n)}$ with codewords $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{\mathsf{M}}$. Considering Rem. 2.11, we can assume without loss of generality that

$$w \triangleq w_{\mathrm{H}}(\mathbf{x}_1) \leq w_{\mathrm{H}}(\mathbf{x}_2) \leq \cdots \leq w_{\mathrm{H}}(\mathbf{x}_{\mathsf{M}}) \tag{B.1}$$

and that all ones of the first codeword are in the last w positions, i.e.,

$$\mathbf{x}_1 = (0\,0\,\cdots\,0\,\underbrace{1\,\cdots\,1}_{w \text{ pos.}}). \tag{B.2}$$

We are going to show that an optimal codebook must satisfy $w = 0$.

We note that for each $\mathbf{y}$ and for every codeword $\mathbf{x}_m$, $1 \leq m \leq \mathsf{M}$, the conditional channel law is

$$\begin{aligned} P_{X|Y}(\mathbf{y}|\mathbf{x}_m) = \mathrm{I}\left\{d_{0,1}(\mathbf{x}_m, \mathbf{y}) = 0\right\} \epsilon_1^{w_{\mathrm{H}}(\mathbf{x}_m) - d_{1,1}(\mathbf{x}_1, \mathbf{y})} \\ \cdot (1-\epsilon_1)^{d_{1,1}(\mathbf{x}_1, \mathbf{y})} \end{aligned} \tag{B.3}$$

where $\mathrm{I}\{\cdot\}$ denotes again the indicator function. Hence, for any $\mathbf{y} = (0\ \cdots\ 0\ y_{n-w+1}\ \cdots\ y_n)$ with $0 \leq w_{\mathrm{H}}(\mathbf{y}) \leq w$,

$$\begin{aligned} &\max\left\{P_{Y|X}^n(\mathbf{y}|\mathbf{x}_1), \ldots, P_{Y|X}^n(\mathbf{y}|\mathbf{x}_m), \ldots, P_{Y|X}^n(\mathbf{y}|\mathbf{x}_{\mathsf{M}})\right\} \\ &= \max\Big\{\epsilon_1^{w - d_{1,1}(\mathbf{x}_1, \mathbf{y})}(1-\epsilon_1)^{d_{1,1}(\mathbf{x}_1, \mathbf{y})}, \\ &\quad \mathrm{I}\left\{d_{0,1}(\mathbf{x}_2, \mathbf{y}) = 0\right\} \epsilon_1^{w_{\mathrm{H}}(\mathbf{x}_2) - d_{1,1}(\mathbf{x}_1, \mathbf{y})}(1-\epsilon_1)^{d_{1,1}(\mathbf{x}_1, \mathbf{y})}, \end{aligned}$$

$$\cdots,$$
$$\mathrm{I}\{d_{0,1}(\mathbf{x}_{\mathrm{M}},\mathbf{y})=0\}\,\epsilon_1^{w_{\mathrm{H}}(\mathbf{x}_{\mathrm{M}})-d_{1,1}(\mathbf{x}_1,\mathbf{y})}(1-\epsilon_1)^{d_{1,1}(\mathbf{x}_1,\mathbf{y})}\Big\} \tag{B.4}$$

$$=\epsilon_1^{w-d_{1,1}(\mathbf{x}_1,\mathbf{y})}(1-\epsilon_1)^{d_{1,1}(\mathbf{x}_1,\mathbf{y})}. \tag{B.5}$$

Since when transmitting $\mathbf{x}_1$, the received sequence cannot have any ones in the first $n-w$ positions, this now shows that the optimal decoding region for the first codeword is

$$\mathcal{D}_1^{(\mathsf{M},n)}=\Big\{\mathbf{y}\colon \mathbf{y}=\big(\underbrace{0\ \cdots\ 0}_{n-w\text{ pos.}}\ \underbrace{y_{n-w+1}\ \cdots\ y_n}_{w\text{ pos.}}\big)\Big\} \tag{B.6}$$

which yields the conditional success probability

$$\psi_1=\sum_{d=0}^{w}\binom{w}{d}\epsilon_1^d\cdot(1-\epsilon_1)^{w-d}=1. \tag{B.7}$$

Hence, we see that $\psi_1=1$ independent of the choice of w. If we choose $w=0$, though, then the size of $\mathcal{D}_1^{(\mathsf{M},n)}$ is minimized, i.e., many vectors $\mathbf{y}$ that belong to $\mathcal{D}_1^{(\mathsf{M},n)}$ for $w>0$ will be moved to some other decoding region $\mathcal{D}_m^{(\mathsf{M},n)}$, $m>1$. This move will increase the success probabilities ψ_m of the corresponding other codeword (because the success probability will contain more terms in their corresponding sum over all $\mathbf{y}\in\mathcal{D}_m^{(\mathsf{M},n)}$). Hence, as ψ_1 remains constant, the total success probability is increased.

Note that this increase is strictly larger than zero if there exist some other codewords that have one or more ones in the last w positions. □

Now we are ready to prove Theorem 9.2. Our proof is based on an exact expression of the average success probability as a function of the numbers of candidate columns t_i. The problem is then transformed into an optimization problem.

We firstly consider the easier case of $\mathsf{M}=3$. By Lemma B.1 and because the all-zero column can be ignored (based on the argument used in the proof of Theorem 8.1), we can restrict our search to the candidate columns given in (6.5). Hence, for any blocklength n, with $t_1+t_2+t_3=n$, denote an arbitrary codebook $\mathscr{C}_{t_2,t_3}^{(3,n)}$. Again, without loss of generality, assume that

$$w_{\mathrm{H}}(\mathbf{x}_1)\le w_{\mathrm{H}}(\mathbf{x}_2)\le w_{\mathrm{H}}(\mathbf{x}_3) \tag{B.8}$$

and note that

$$w_{\mathrm{H}}(\mathbf{x}_1)=0,\ w_{\mathrm{H}}(\mathbf{x}_2)=t_2+t_3,\ w_{\mathrm{H}}(\mathbf{x}_3)=t_1+t_3 \tag{B.9}$$

and (because $w_{\mathrm{H}}(\mathbf{x}_2)\le w_{\mathrm{H}}(\mathbf{x}_3)$) that $t_2\le t_1$.

The decoding region of the first codeword is then just the all-zero vector $\mathbf{0}$ with $\psi_1=1$.

Defining $t\triangleq t_2+t_3$ and using a derivation similar to (B.4)–(B.6), we further realize that

$$\mathcal{D}_{t_2,t_3;2}^{(3,n)}=\Big\{\mathbf{y}\colon \mathbf{y}=\big(\underbrace{0\ \cdots\ 0}_{n-t\text{ pos.}}\ \underbrace{y_{n-t+1}\ \cdots\ y_n}_{t\text{ pos.}}\big)\ \text{with } 1\le w_{\mathrm{H}}(\mathbf{y})\le t\Big\} \tag{B.10}$$

and

$$\psi_2 = 1 - \epsilon_1^t. \tag{B.11}$$

Finally, the remaining $\mathbf{y}$ belong to $\mathcal{D}_{t_2,t_3;3}^{(3,n)}$:

$$\mathcal{D}_{t_2,t_3;3}^{(3,n)} = \{0,1\}^n \setminus \left(\mathcal{D}_{t_2,t_3;2}^{(3,n)} \cup \{\mathbf{0}\}\right) \tag{B.12}$$

$$= \left\{\mathbf{y}\colon [\mathbf{y}^{(t_1)}\ \mathbf{0}^{(t_2)}\ \mathbf{y}^{(t_3)}] \text{ with } 1 \le w_{\mathrm{H}}(\mathbf{y}^{(t_1)}) \le t_1,\ 0 \le w_{\mathrm{H}}(\mathbf{y}^{(t_3)}) \le t_3\right\} \tag{B.13}$$

with

$$\psi_3 = \left(\sum_{d=0}^{t_1-1} \epsilon_1^d (1-\epsilon_1)^{t_1-d}\right) \cdot \left(\sum_{d=0}^{t_3} \epsilon_1^d (1-\epsilon_1)^{t_3-d}\right) \tag{B.14}$$

$$= \left(1 - \epsilon_1^{t_1}\right) \cdot 1. \tag{B.15}$$

Hence, the average success probability for a codebook $\mathscr{C}_{t_2,t_3}^{(3,n)}$ with $t = t_2 + t_3$ and $t_1 \ge t_2$ is

$$3P_{\mathrm{c}}\left(\mathscr{C}_{t_2,t_3}^{(3,n)}\right) = 1 + (1 - \epsilon_1^t) + (1 - \epsilon_1^{n-t}). \tag{B.16}$$

The proof for the case $\mathsf{M} = 3$ is now completed by showing that the average success probability (B.16) is maximized by the choice $t^* = \lfloor n/2 \rfloor$.

In the case of $\mathsf{M} = 4$, we cannot only rely on the candidate columns in (6.6), but unfortunately need to consider totally seven candidate columns:

$$\left\{\mathbf{c}_1^{(4)} \triangleq \begin{pmatrix}0\\0\\0\\1\end{pmatrix}\ \mathbf{c}_2^{(4)} \triangleq \begin{pmatrix}0\\0\\1\\0\end{pmatrix},\ \mathbf{c}_3^{(4)} \triangleq \begin{pmatrix}0\\0\\1\\1\end{pmatrix},\ \mathbf{c}_4^{(4)} \triangleq \begin{pmatrix}0\\1\\0\\0\end{pmatrix},\right.$$
$$\left.\mathbf{c}_5^{(4)} \triangleq \begin{pmatrix}0\\1\\0\\1\end{pmatrix},\ \mathbf{c}_6^{(4)} \triangleq \begin{pmatrix}0\\1\\1\\0\end{pmatrix},\ \mathbf{c}_7^{(4)} \triangleq \begin{pmatrix}0\\1\\1\\1\end{pmatrix}\right\}. \tag{B.17}$$

We use $\mathbf{t} = [t_1, t_2, t_3, t_4, t_5, t_6, t_7]$ to describe an arbitrary code, and again, without loss of generality, assume that

$$w_{\mathrm{H}}(\mathbf{x}_1) \le w_{\mathrm{H}}(\mathbf{x}_2) \le w_{\mathrm{H}}(\mathbf{x}_3) \le w_{\mathrm{H}}(\mathbf{x}_4). \tag{B.18}$$

Also note that

$$w_{\mathrm{H}}(\mathbf{x}_1) = 0 \tag{B.19}$$

$$w_{\mathrm{H}}(\mathbf{x}_2) = t_4 + t_5 + t_6 + t_7 \tag{B.20}$$

$$w_{\mathrm{H}}(\mathbf{x}_3) = t_2 + t_3 + t_6 + t_7 \tag{B.21}$$

$$w_{\mathrm{H}}(\mathbf{x}_3) = t_1 + t_3 + t_5 + t_7 \tag{B.22}$$

and, as a result, $t_4+t_5 \le t_2+t_3$ and $t_2+t_6 \le t_1+t_5$. Again, we investigate the decoding region with corresponding success probability.

The first two decoding regions are very similar to case of $\mathsf{M}=3$ and yield

$$\psi_1 = 1, \quad \psi_2 = 1-\epsilon_1^{t_4+t_5+t_6+t_7}. \tag{B.23}$$

Then, we have

$$\begin{aligned}\mathcal{D}_3^{(4,n)} = \Big\{\mathbf{y}^{(n)} \colon \mathbf{y}^{(n)} = \big[\mathbf{0}^{(t_1)}\ \mathbf{y}^{(t_2+t_3)}\ \mathbf{0}^{(t_4+t_5)}\ \mathbf{y}^{(t_6+t_7)}\big] \\ \text{with } 1 \le w_{\mathrm{H}}\big(\mathbf{y}^{(t_2+t_3)}\big) \le t_2+t_3, \\ \text{and } 0 \le w_{\mathrm{H}}(\mathbf{y}^{(t_6+t_7)}) \le t_6+t_7\Big\}\end{aligned} \tag{B.24}$$

with

$$\psi_3 = 1-\epsilon_1^{t_2+t_3}. \tag{B.25}$$

The fourth decoding region is more complicated. It can be written as

$$\mathcal{D}_4^{(4,n)} = \mathcal{P} \setminus \Big(\mathcal{D}_2^{(4,n)} \cup \mathcal{D}_3^{(4,n)}\Big) \tag{B.26}$$

where

$$\begin{aligned}\mathcal{P} \triangleq \Big\{\mathbf{y}^{(n)} \colon \mathbf{y}^{(n)} = \big[\mathbf{y}^{(t_1)}\ \mathbf{0}^{(t_2)}\ \mathbf{y}^{(t_3)}\ \mathbf{0}^{(t_4)}\ \mathbf{y}^{(t_5)}\ \mathbf{0}^{(t_6)}\ \mathbf{y}^{t_7)}\big] \\ \text{with } 1 \le w_{\mathrm{H}}(\mathbf{y}^{(n)}) \le t_1+t_3+t_5+t_7\Big\}.\end{aligned} \tag{B.27}$$

Hence

$$\begin{aligned}\psi_4 &= \sum_{\mathbf{y}\in\mathcal{P}} P_{Y|X}(\mathbf{y}|\mathbf{x}_4) - \sum_{\mathbf{y}\in\mathcal{P}\cap\mathcal{D}_2^{(4,n)}} P_{Y|X}(\mathbf{y}|\mathbf{x}_4) \\ &\quad - \sum_{\mathbf{y}\in(\mathcal{P}\setminus\mathcal{D}_2^{(4,n)})\cap\mathcal{D}_3^{(4,n)}} P_{Y|X}(\mathbf{y}|\mathbf{x}_4) \qquad \text{(B.28)} \\ &= \left(\sum_{d=0}^{t_1+t_3+t_5+t_7-1} \epsilon_1^d(1-\epsilon_1)^{t_1+t_3+t_5+t_7-d}\right) \\ &\quad - \epsilon_1^{t_1+t_3}\left(\sum_{d=0}^{t_5+t_7-1} \epsilon_1^d(1-\epsilon_1)^{t_5+t_7-d}\right) \\ &\quad - \epsilon_1^{t_1+t_5}\left(\sum_{d=1}^{t_3-1} \epsilon_1^d(1-\epsilon_1)^{t_3-d}\right)\left(\sum_{d=1}^{t_7} \epsilon_1^d(1-\epsilon_1)^{t_7-d}\right) \qquad \text{(B.29)} \\ &= \left(1-\epsilon_1^{t_1+t_3+t_5+t_7}\right) - \epsilon_1^{t_1+t_3}\left(1-\epsilon_1^{t_5+t_7}\right) - \epsilon_1^{t_1+t_5}\left(1-\epsilon_1^{t_3}\right) \qquad \text{(B.30)} \\ &= 1-\epsilon_1^{t_1+t_3} - \epsilon_1^{t_1+t_5}(1-\epsilon_1^{t_3}) \qquad \text{(B.31)}\end{aligned}$$

where

$$\begin{aligned}\mathcal{P} \cap \mathcal{D}_2^{(4,n)} &= \Big\{\mathbf{y}^{(n)} \colon \mathbf{y}^{(n)} = \big[\mathbf{0}^{(t_1)}\ \mathbf{0}^{(t_2)}\ \mathbf{0}^{(t_3)}\ \mathbf{0}^{(t_4)}\ \mathbf{y}^{(t_5)}\ \mathbf{0}^{(t_6)}\ \mathbf{y}^{t_7)}\big] \\ &\qquad \text{with } 1 \le w_{\mathrm{H}}(\mathbf{y}) \le t_5 + t_7\Big\} \end{aligned} \tag{B.32}$$

and

$$\begin{aligned}\big(\mathcal{P} \setminus \mathcal{D}_2^{(4,n)}\big) \cap \mathcal{D}_3^{(4,n)} &= \Big\{\mathbf{y}^{(n)} \colon \mathbf{y}^{(n)} = \big[\mathbf{0}^{(t_1)}\ \mathbf{0}^{(t_2)}\ \mathbf{y}^{(t_3)}\ \mathbf{0}^{(t_4)}\ \mathbf{0}^{(t_5)}\ \mathbf{0}^{(t_6)}\ \mathbf{y}^{(t_7)}\big] \\ &\qquad \text{with } 1 \le w_{\mathrm{H}}(\mathbf{y}^{(t_3)}) \le t_3, 0 \le w_{\mathrm{H}}(\mathbf{y}^{(t_7)}) \le t_7\Big\}. \end{aligned} \tag{B.33}$$

Hence, the average success probability for a codebook $\mathscr{C}^{(4,n)}$ with $t_4 + t_5 \le t_2 + t_3$ and $t_2 + t_6 \le t_1 + t_5$ is

$$\begin{aligned}4P_{\mathrm{c}}\big(\mathscr{C}^{(4,n)}\big) = 1 &+ \big(1 - \epsilon_1^{n-(t_1+t_2+t_3)}\big) + \big(1 - \epsilon_1^{t_2+t_3}\big) \\ &+ \big(1 - \epsilon_1^{t_1+t_3} - \epsilon_1^{t_1+t_5}(1 - \epsilon_1^{t_3})\big) \end{aligned} \tag{B.34}$$

and is maximized for

$$\mathbf{t}^* = \left[0, 0, \left\lfloor \frac{n}{2} \right\rfloor, 0, \left\lceil \frac{n}{2} \right\rceil, 0, 0\right]. \tag{B.35}$$

Furthermore, it can be shown that the optimum $\mathbf{t}^*$ is unique for even n, while there are also other solutions for odd n.

B.2 Proof of Lemma 9.5

We apply (B.16) and (B.34) to the weak flip code of type $(t, 0)$.

Corollary B.2 *On a ZC, for* $\mathsf{M} = 3$ *or* $\mathsf{M} = 4$*, and for any* $n \ge 2$*, the optimal decoding regions* $\mathcal{D}_{t,0;m}^{(\mathsf{M},n)}$ *for the weak flip code of type* $(t, 0)$*,* $\mathscr{C}_{t,0}^{(\mathsf{M},n)}$*, for* $1 \le t \le \lfloor \frac{n}{2} \rfloor$*, are*

$$\mathcal{D}_{t,0;1}^{(\mathsf{M},n)} = \{\mathbf{0}\} \tag{B.36}$$

$$\begin{aligned}\mathcal{D}_{t,0;2}^{(\mathsf{M},n)} = \Big\{\mathbf{y} \colon \mathbf{y} = \big(&\underbrace{0\ \cdots\ 0}_{\substack{n-t \\ \text{pos.}}}\ \underbrace{y_{n-t+1}\ \cdots\ y_n}_{t\text{ pos.}}\big) \\ &\text{with } 1 \le w_{\mathrm{H}}(\mathbf{y}) \le t\Big\} \end{aligned} \tag{B.37}$$

$$\begin{aligned}\mathcal{D}_{t,0;3}^{(\mathsf{M},n)} = \Big\{\mathbf{y} \colon \mathbf{y} = \big(&\underbrace{y_1\ \cdots\ y_{n-t}}_{n-t\text{ pos.}}\ \underbrace{0\ \cdots\ 0}_{t\text{ pos.}}\big) \\ &\text{with } 1 \le w_{\mathrm{H}}(\mathbf{y}) \le n - t\Big\} \end{aligned} \tag{B.38}$$

$$\mathcal{D}_{t,0;4}^{(4,n)} = \{0,1\}^n \setminus \bigcup_{m=1}^{3} \mathcal{D}_{t,0;m}^{(4,n)}. \tag{B.39}$$

The corresponding average success probabilities are

$$3P_{\mathrm{c}}\big(\mathscr{C}_{t,0}^{(3,n)}\big) = 1 + \big(1-\epsilon_1^t\big) + \big(1-\epsilon_1^{n-t}\big) \tag{B.40}$$

$$\begin{aligned} 4P_{\mathrm{c}}\big(\mathscr{C}_{t,0}^{(4,n)}\big) &= 1 + \big(1-\epsilon_1^t\big) + \big(1-\epsilon_1^{n-t}\big) \\ &\quad + \big(1-\epsilon_1^{n-t}\big) - \epsilon_1^t\big(1-\epsilon_1^{n-t}\big). \end{aligned} \tag{B.41}$$

Note that all received sequences in $\mathcal{D}_{t,0;4}^{(4,n)}$ have zero probability of occurring in the situation of $\mathsf{M}=3$ because the code $\mathscr{C}_{t,0}^{(3,n)}$ does not contain the all-one codeword. Therefore, we do not need to include them into any decoding region for $\mathsf{M}=3$.

We start with $\mathsf{M}=4$ and recall again that there are the seven possible columns given in (B.17) that we can choose from (the all-zero column can be ignored based on the argument used in the proof of Theorem 8.1). To prove Lemma 9.5 we append an additional bit to all four codewords of $\mathscr{C}_{t,0}^{(4,n)}$ as follows:

$$\begin{pmatrix} [\mathbf{0} & x_{1,n+1}] \\ [\mathbf{x} & x_{2,n+1}] \\ [\bar{\mathbf{x}} & x_{3,n+1}] \\ [\mathbf{1} & x_{4,n+1}] \end{pmatrix} \tag{B.42}$$

where $x_{m,n+1} \in \{0,1\}$ and where $\mathbf{x}$ and $\bar{\mathbf{x}}$ are given in (6.1) with $t \in \big\{1,2,\ldots,\lfloor \frac{n}{2} \rfloor\big\}$. We denote[11] this new code by $\mathscr{C}^{(4,n+1)}$. We now need to establish the decoding regions for the new code $\mathscr{C}^{(4,n+1)}$. If we simply extend the decoding regions given in (B.36)–(B.39) by one bit, $\big[\mathcal{D}_{t,0;m}^{(4,n)}\ 0\big] \cup \big[\mathcal{D}_{t,0;m}^{(4,n)}\ 1\big]$, for $m=1,2,3,4$, then we retain the same success probability because

$$\begin{aligned} &\psi_m\big(\mathscr{C}^{(4,n+1)}\big) \\ &= \psi_m\big(\mathscr{C}_{t,0}^{(4,n)}\big) \cdot P_{Y|X}(0|x_{m,n+1}) \\ &\quad + \psi_m\big(\mathscr{C}_{t,0}^{(4,n)}\big) \cdot P_{Y|X}(1|x_{m,n+1}) \qquad \text{(B.43)} \\ &= \psi_m\big(\mathscr{C}_{t,0}^{(4,n)}\big) \cdot \big(P_{Y|X}(0|x_{m,n+1}) + P_{Y|X}(1|x_{m,n+1})\big) \\ &\qquad \text{(B.44)} \\ &= \psi_m\big(\mathscr{C}_{t,0}^{(4,n)}\big). \qquad \text{(B.45)} \end{aligned}$$

However, it is quite clear that these regions are in general no longer the optimal decision regions for $\mathscr{C}^{(4,n+1)}$. So the question is how to fix them to make them optimal again (and thereby also finding how to optimally choose $x_{m,n+1}$).

Firstly note that if $x_{m,n+1}=0$, adding a 0 to the received vector $\mathbf{y}^{(n)}$ will not change the decision m because 0 is the success outcome anyway. Similarly, if $x_{m,n+1}=1$, adding a 1 to the vector $\mathbf{y}^{(n)}$ will not change the decision m.

Secondly, we claim that even if $x_{m,n+1}=1$, all received vectors $\mathbf{y}^{(n+1)} \in \big[\mathcal{D}_{t,0;m}^{(4,n)}\ 0\big]$ still will optimally be decoded to m. To see this, we have a look at the four cases separately:

[11] Note that again we use a proof technique that uses a given code to create a new code by adding a column to the codebook matrix. We therefore again use the notation introduced in Appendix A.3, i.e., we use superscripts (n) to clarify length and affiliation.

- $\big[\mathcal{D}_{t,0;1}^{(4,n)}\ 0\big]$: The decoding region $\big[\mathcal{D}_{t,0;1}^{(4,n)}\ 0\big]$ only contains one vector: the all-zero vector. We have

$$\begin{aligned} &P_{Y|X}^{n+1}\big(\mathbf{0}^{(n+1)}\big|\mathbf{x}_1^{(n+1)} = \big[\mathbf{0}^{(n)}\ 1\big]\big) \\ &\quad = \epsilon_1 \geq P_{Y|X}^{n+1}\big(\mathbf{0}^{(n+1)}\big|\mathbf{x}_m^{(n+1)}\big), \quad \forall\, m = 2,3,4 \end{aligned} \tag{B.46}$$

  independently of the choices for $x_{m,n+1}$, $m = 2,3,4$. Hence, we decide for $m = 1$.

- $\big[\mathcal{D}_{t,0;2}^{(4,n)}\ 0\big]$: All vectors in $\big[\mathcal{D}_{t,0;2}^{(4,n)}\ 0\big]$ contain ones in positions that make it impossible to decode it as $m = 1$ or $m = 3$. On the other hand, $m = 4$ obviously is less likely than $m = 2$, i.e., we decide $m = 2$.

- $\big[\mathcal{D}_{t,0;3}^{(4,n)}\ 0\big]$: All vectors in $\big[\mathcal{D}_{t,0;3}^{(4,n)}\ 0\big]$ contain ones in positions that make it impossible to decode it as $m = 1$ or $m = 2$. On the other hand, $m = 4$ obviously is less likely than $m = 3$, i.e., we decide $m = 3$.

- $\big[\mathcal{D}_{t,0;4}^{(4,n)}\ 0\big]$: All vectors in $\big[\mathcal{D}_{t,0;4}^{(4,n)}\ 0\big]$ contain ones in positions that make it impossible to decode it as $m = 1$, $m = 2$, or $m = 3$. It only remains to decide $m = 4$.

So, it only remains to investigate the decisions made about the vectors in $\big[\mathcal{D}_{t,0;m}^{(4,n)}\ 1\big]$ if $x_{m,n+1} = 0$. Note that we do not need to bother about $\big[\mathcal{D}_{t,0;4}^{(4,n)}\ 1\big]$ as it is impossible to receive such a vector. For $m = 1$, 2, or 3, if $x_{m,n+1} = 0$, the received vectors in $\big[\mathcal{D}_{t,0;m}^{(4,n)}\ 1\big]$ will change to another decoding region not equal to m because $P_{Y|X}(1|0) = 0$.

- $\big[\mathcal{D}_{t,0;1}^{(4,n)}\ 1\big]$: If we assign these vectors (actually, it has only one) to the new decoding region $\mathcal{D}_{t,0;2}^{(4,n+1)}$, the conditional success probability for $m = 2$ is increased by

$$\begin{aligned} \Delta\psi_2 &\triangleq \psi_2\big(\mathscr{C}^{(4,n+1)}\big) - \psi_2\big(\mathscr{C}_{t,0}^{(4,n)}\big) && \text{(B.47)} \\ &= \sum_{\mathbf{y}^{(n)} \in \mathcal{D}_{t,0;1}^{(4,n)}} P_{Y|X}^{n+1}\Big(\big[\mathbf{y}^{(n)}\ 1\big]\Big|\big[\mathbf{0}^{(n-t)}\ \mathbf{1}^t\ 1\big]\Big) \\ &\qquad \cdot (x_{2,n+1} - x_{1,n+1})^+ && \text{(B.48)} \\ &= \epsilon_1^t(1-\epsilon_1)(x_{2,n+1} - x_{1,n+1})^+ && \text{(B.49)} \end{aligned}$$

  where

$$(x)^+ = x \cdot \mathrm{I}\{x \geq 0\} = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases} \tag{B.50}$$

  Note that we only have a positive increase in the success probability if $x_{2,n+1} = 1$. Similarly, we compute

$$\Delta\psi_3 = \epsilon_1^{n-t}(1-\epsilon_1)(x_{3,n+1} - x_{1,n+1})^+ \tag{B.51}$$

$$\Delta\psi_4 = \epsilon_1^{n}(1-\epsilon_1)(x_{4,n+1} - x_{1,n+1})^+. \tag{B.52}$$

From $\epsilon_1^t \geq \epsilon_1^{n-t} > \epsilon_1^n$, we see that $\Delta\psi_2$ gives the highest increase, followed by $\Delta\psi_3$ and then $\Delta\psi_4$. Hence, in order to represent this choice of ordering, we rewrite (B.49), (B.51), and (B.52) as follows:

$$\Delta\psi_2 = \epsilon_1^t(1-\epsilon_1)(x_{2,n+1} - x_{1,n+1})^+ \tag{B.53}$$
$$\Delta\psi_3 = \epsilon_1^{n-t}(1-\epsilon_1)(x_{3,n+1} - x_{2,n+1} - x_{1,n+1})^+ \tag{B.54}$$
$$\Delta\psi_4 = \epsilon_1^n(1-\epsilon_1)\cdot(x_{4,n+1} - x_{3,n+1} - x_{2,n+1} - x_{1,n+1})^+. \tag{B.55}$$

- $\left[\mathcal{D}_{t,0;2}^{(4,n)}\ 1\right]$: In this case, only $\mathcal{D}_{t,0;4}^{(4,n+1)}$ yields a nonzero additional conditional success probability:

$$\Delta\psi_4 = \sum_{\mathbf{y}^{(n)}\in\mathcal{D}_{t,0;2}^{(4,n)}} P_{Y|X}^{n+1}\left(\left[\mathbf{y}^{(n)}\ 1\right]\middle|\left[\mathbf{1}^{(n)}\ 1\right]\right)\cdot(x_{4,n+1} - x_{2,n+1})^+ \tag{B.56}$$
$$= \sum_{d=0}^{t-1}\binom{t}{d}(1-\epsilon_1)^{t-d}\epsilon_1^{n-t+d}(1-\epsilon_1)\cdot(x_{4,n+1} - x_{2,n+1})^+ \tag{B.57}$$
$$= \left(\epsilon_1^{n-t} - \epsilon_1^n\right)(1-\epsilon_1)(x_{4,n+1} - x_{2,n+1})^+. \tag{B.58}$$

- $\left[\mathcal{D}_{t,0;3}^{(4,n)}\ 1\right]$: Again, only $\mathcal{D}_{t,0;4}^{(4,n+1)}$ yields a nonzero additional conditional success probability:

$$\Delta\psi_4 = \sum_{\mathbf{y}^{(n)}\in\mathcal{D}_{t,0;3}^{(4,n)}} P_{Y|X}^{n+1}\left(\left[\mathbf{y}^{(n)}\ 1\right]\middle|\left[\mathbf{1}^{(n)}\ 1\right]\right)\cdot(x_{4,n+1} - x_{3,n+1})^+ \tag{B.59}$$
$$= \left(\epsilon_1^t - \epsilon_1^n\right)(1-\epsilon_1)(x_{4,n+1} - x_{3,n+1})^+. \tag{B.60}$$

For $\epsilon_1^t > \epsilon_1^{n-t} > \epsilon_1^n$, we can now conclude that the unique best solution for the choice of $x_{m,n+1}$, yielding the largest increase in success probability in (B.53), (B.54), (B.55), (B.58), and (B.60), is as follows:

$$\begin{cases} x_{2,n+1} - x_{1,n+1} = 1 \\ x_{4,n+1} - x_{2,n+1} = 0 \\ x_{4,n+1} - x_{3,n+1} = 1 \end{cases} \implies \begin{cases} x_{1,n+1} = 0 \\ x_{2,n+1} = 1 \\ x_{3,n+1} = 0 \\ x_{4,n+1} = 1 \end{cases} \tag{B.61}$$

which corresponds to $\mathbf{c}_2^{(4)}$. This choice will lead to a total success probability increase of

$$\Delta\Psi\left(\mathscr{C}_{t+1,0}^{(4,n+1)}\right) = \frac{1}{4}\epsilon_1^t(1-\epsilon_1) + \frac{1}{4}\left(\epsilon_1^t - \epsilon_1^n\right)(1-\epsilon_1) \tag{B.62}$$
$$= \frac{1}{4}\left(2\epsilon_1^t - \epsilon_1^n\right)(1-\epsilon_1). \tag{B.63}$$

If n is even and $t = \frac{n}{2}$, then $\epsilon_1^t = \epsilon_1^{n-t}$. In this case $\mathbf{c}_2^{(4)}$ still yields the largest increase in success probability, but it is not anymore the unique choice to do so.

The proof for $\mathsf{M} = 3$ is similar and omitted.

Appendix C

Derivations concerning the BSC

C.1 Proof of Theorem 10.2

We firstly consider the case $\mathsf{M} = 3$.

Our proof is based on induction in n. We start with a locally optimal code of length $n-1$ and then prove that appending a column according to the choice given in Theorem 10.2 will result in a new locally optimal code that maximizes the total probability increase. We rely on a couple of observations that for clarity are summarized here once more:

- The proof that the $n = 2$ binary code given in (11.1) is optimal is straightforward and omitted.
- We do not need to worry about any other codebook columns than those given in (6.5) because firstly the all-zero and the all-one column can be neglected by the same argument as used in the proof of Theorem 8.1, and because secondly the flipped version of the columns $\mathbf{c}_1^{(3)}$, $\mathbf{c}_2^{(3)}$, and $\mathbf{c}_3^{(3)}$ will result in the same performance because the BSC is strongly symmetric.
- Due to Lemma 6.7 and the average success probability only depends on $\mathbf{d}\big(\mathscr{C}_{t_2,t_3}^{(3,n)}\big)$, no matter how the components order is. W.L.O.G., we can assume that the general code parameters $[t_1, t_2, t_3]$ with $t_1 \geq t_2 \geq t_3$.
- We need to distinguish three cases in the induction from $n-1$ to n, depending on whether $n \bmod 3 = 0$, 1, or 2.

Note that once again we use the notation introduced in App. A.3, i.e., we use a superscript (n) to denote length and affiliation. Moreover, we introduce the following shorthands:

$$d_m^{(n)}(\mathbf{y}) \triangleq d_{\mathrm{H}}(\mathbf{x}_m, \mathbf{y}), \quad m = 1, \ldots, \mathsf{M} \tag{C.1}$$

and

$$\mathbf{d}^{(n)}(\mathbf{y}) \triangleq \big(d_1^{(n)}(\mathbf{y}), d_2^{(n)}(\mathbf{y}), \ldots, d_{\mathsf{M}}^{(n)}(\mathbf{y})\big). \tag{C.2}$$

Be aware not to confuse $\mathbf{d}^{(n)}(\mathbf{y})$, which is a vector that compares all length-n codewords with a given received vector $\mathbf{y}$, with the pairwise Hamming distance vector $\mathbf{d}\big(\mathscr{C}^{(\mathrm{M},n)}\big)$, which compares all possible pairing combinations of the codewords of a codebook $\mathscr{C}^{(\mathrm{M},n)}$.

We also remind the reader that $k \triangleq \lfloor \frac{n}{3} \rfloor$ and $p \triangleq \left(\frac{\epsilon}{1-\epsilon}\right)$.

Using these shorthands, we can describe the ML decoding rule for a BSC quite simply as

$$g(\mathbf{y}) = \underset{1\le m\le \mathrm{M}}{\operatorname{argmin}} \big\{d_m^{(n)}(\mathbf{y})\big\}. \tag{C.3}$$

We start with an observation about a basic property of the weak flip code given in (10.3).

Claim C.1 *For the weak flip code of* (10.3), $\mathscr{C}_{t_2^*,t_3^*}^{(3,n)}$, *the largest received Hamming distance between any* $\mathbf{y}$ *and the nearest codeword is given by the minimum Hamming distance of the codebook:*

$$\max_{\mathbf{y}} \min_{j\in\{1,2,3\}} d_j^{(n)}(\mathbf{y}) = d_{\min}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n)}\big). \tag{C.4}$$

Proof: It is not too difficult to see that a $\mathbf{y}$ that achieves the maximum in (C.4) should have t_1^* ones, t_2^* ones, and t_3^* zeros in the positions where the optimal codebook consists of $\mathbf{c}_1^{(3)}$, $\mathbf{c}_2^{(3)}$, and $\mathbf{c}_3^{(3)}$, respectively:

$$\mathbf{y}_{\max} \triangleq (\underbrace{1\ \cdots\ 1}_{t_1^*}\ \underbrace{1\ \cdots\ 1}_{t_2^*}\ \underbrace{0\ \cdots\ 0}_{t_3^*}). \tag{C.5}$$

Then,

$$\begin{aligned}
&\max_{\mathbf{y}} \min_{j\in\{1,2,3\}} d_j^{(n)}(\mathbf{y}) \\
&= \max_{\mathbf{y}} \min \big\{d_1^{(n)}(\mathbf{y}), d_2^{(n)}(\mathbf{y}), d_3^{(n)}(\mathbf{y})\big\} && \text{(C.6)}\\
&= \min \big\{d_1^{(n)}(\mathbf{y}_{\max}), d_2^{(n)}(\mathbf{y}_{\max}), d_3^{(n)}(\mathbf{y}_{\max})\big\} && \text{(C.7)}\\
&= \min\{t_1^*+t_2^*, t_1^*+t_3^*, t_2^*+t_3^*\} && \text{(C.8)}\\
&= \min\Big\{d_{\mathrm{H}}\big(\mathbf{x}_2^{(n)},\mathbf{x}_3^{(n)}\big), d_{\mathrm{H}}\big(\mathbf{x}_1^{(n)},\mathbf{x}_3^{(n)}\big), d_{\mathrm{H}}\big(\mathbf{x}_1^{(n)},\mathbf{x}_2^{(n)}\big)\Big\} && \text{(C.9)}\\
&= d_{\min}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n)}\big). && \text{(C.10)}
\end{aligned}$$

Note that for other code structures, this claim is in general not true. □

Also note that the (length-3) pairwise Hamming distance vector of any code $\mathscr{C}^{(3,n-1)}$ will have exactly 2 components increased by 1 when appending either $\mathbf{c}_1^{(3)}$, $\mathbf{c}_2^{(3)}$, or $\mathbf{c}_3^{(3)}$ to the codebook matrix to form a new code $\mathscr{C}^{(3,n)}$. For example, if we add $\mathbf{c}_1^{(3)}$, then

$$\begin{aligned}
\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = \Big(&d_{\mathrm{H}}\big(\mathbf{x}_1^{(n-1)},\mathbf{x}_2^{(n-1)}\big), d_{\mathrm{H}}\big(\mathbf{x}_1^{(n-1)},\mathbf{x}_3^{(n-1)}\big)+1,\\
&d_{\mathrm{H}}\big(\mathbf{x}_2^{(n-1)},\mathbf{x}_3^{(n-1)}\big)+1\Big).
\end{aligned} \tag{C.11}$$

We are now ready for our induction proof.

C.1.1 Case i: Step from $n-1=3k-1$ to $n=3k$

We start with the code $\mathscr{C}^{(3,n-1)}_{t_2^*,t_3^*}$, whose code parameters, pairwise Hamming distance vector, and minimum Hamming distance are as follows:

code parameters:

$$[t_1^*, t_2^*, t_3^*] = [k, k, k-1] \tag{C.12}$$

pairwise Hamming distance vector:

$$\mathbf{d}\big(\mathscr{C}^{(3,n-1)}_{t_2^*,t_3^*}\big) = (2k-1, 2k-1, 2k) \tag{C.13}$$

minimum Hamming distance:

$$d_{\min}\big(\mathscr{C}^{(3,n-1)}_{t_2^*,t_3^*}\big) = 2k-1. \tag{C.14}$$

The corresponding success probability formula looks as follows:

$$\begin{aligned}
&3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n-1)}_{t_2^*,t_3^*}\big) \\
&= \sum_{m=1}^{3} \sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k,k-1;m}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)} \big| \mathbf{x}_m^{(n-1)}\big) && \text{(C.15)} \\
&= (1-\epsilon)^{n-1} \sum_{m=1}^{3} \sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k,k-1;m}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_{\mathrm{H}}(\mathbf{x}_m^{(n-1)}, \mathbf{y}^{(n-1)})} && \text{(C.16)} \\
&= (1-\epsilon)^{n-1} \Bigg(\sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k,k-1;1}} p^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})} \\
&\qquad + \sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k,k-1;2}} p^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})} \\
&\qquad + \sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k,k-1;3}} p^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})} \Bigg) && \text{(C.17)} \\
&= (1-\epsilon)^{n} \Bigg(\sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k,k-1;1}} p^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})} \\
&\qquad + \sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k,k-1;1}} p^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})+1} \\
&\qquad + \sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k,k-1;2}} p^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})} \\
&\qquad + \sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k,k-1;2}} p^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})+1}
\end{aligned}$$

$$+ \sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k,k-1;3}} p^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})} + \sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k,k-1;3}} p^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})+1} \Bigg) \tag{C.18}$$

where in the last equality we used the trick to write

$$1 = (1-\epsilon)\left(1 + \frac{\epsilon}{1-\epsilon}\right) = (1-\epsilon)(1+p). \tag{C.19}$$

Appending $\mathbf{c}_3^{(3)}$: We now build a new length-n (weak flip) code $\mathscr{C}^{(3,n)}$ from the given code $\mathscr{C}^{(3,n-1)}_{t_2^*,t_3^*}$ by appending $\mathbf{c}_2^{(3)} = (0\ 1\ 1)^\mathsf{T}$. The cases when we append $\mathbf{c}_1^{(3)}$ or $\mathbf{c}_2^{(3)}$ will be discussed later. The new code has the following parameters:

$$[t_1, t_2, t_3] = [k, k, k] \tag{C.20}$$

$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k, 2k, 2k) \tag{C.21}$$

$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k. \tag{C.22}$$

Note that we can rewrite (C.18) in the following way:

$$\begin{aligned} &3P_{\mathrm{c}}\Big(\mathscr{C}^{(3,n-1)}_{t_2^*,t_3^*}\Big) \\ &= (1-\epsilon)^n \Bigg(\sum_{\mathbf{y}^{(n)} \in \left[\mathcal{D}^{(3,n-1)}_{k,k-1;1}\ 0\right]} p^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})} \\ &\quad + \sum_{\mathbf{y}^{(n)} \in \left[\mathcal{D}^{(3,n-1)}_{k,k-1;1}\ 1\right]} p^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})+1} \\ &\quad + \sum_{\mathbf{y}^{(n)} \in \left[\mathcal{D}^{(3,n-1)}_{k,k-1;2}\ 1\right]} p^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})} \\ &\quad + \sum_{\mathbf{y}^{(n)} \in \left[\mathcal{D}^{(3,n-1)}_{k,k-1;2}\ 0\right]} p^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})+1} \\ &\quad + \sum_{\mathbf{y}^{(n)} \in \left[\mathcal{D}^{(3,n-1)}_{k,k-1;3}\ 1\right]} p^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})} \\ &\quad + \sum_{\mathbf{y}^{(n)} \in \left[\mathcal{D}^{(3,n-1)}_{k,k-1;3}\ 0\right]} p^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})+1} \Bigg). \end{aligned} \tag{C.23}$$

We compare this with the success probability of the new code:

$$3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}\big)$$

$$= (1-\epsilon)^n \left(\sum_{\mathbf{y}^{(n)} \in \mathcal{D}_1^{(3,n)}} p^{d_1^{(n)}(\mathbf{y}^{(n)})} + \sum_{\mathbf{y}^{(n)} \in \mathcal{D}_2^{(3,n)}} p^{d_2^{(n)}(\mathbf{y}^{(n)})} + \sum_{\mathbf{y}^{(n)} \in \mathcal{D}_3^{(3,n)}} p^{d_3^{(n)}(\mathbf{y}^{(n)})} \right) \tag{C.24}$$

where we use $\mathcal{D}_m^{(3,n)}$ to denote the decoding region of the new code $\mathscr{C}^{(3,n)}$. In order to be able to compare (C.23) with (C.24), we need to be able to compare $\mathcal{D}_{k,k-1;m}^{(3,n-1)}$ with $\mathcal{D}_m^{(3,n)}$ and $d_m^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ with $d_m^{(n)}\big(\mathbf{y}^{(n)}\big)$. Note that every $\mathbf{y}^{(n)}$ can be uniquely written as some $\mathbf{y}^{(n-1)}$ plus an appended 0 or 1.

Since we have appended $\mathbf{c}_3^{(3)} = (0\ 1\ 1)^\intercal$ to the code of length $n-1$, it is obvious that

$$\begin{aligned}
&\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;1}^{(3,n-1)} \implies \\
&\quad \big[\mathbf{y}^{(n-1)}\ 0\big] \in \mathcal{D}_1^{(3,n)};\ d_1^{(n)}\big(\mathbf{y}^{(n)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) && \text{(C.25)} \\
&\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;2}^{(3,n-1)} \implies \\
&\quad \big[\mathbf{y}^{(n-1)}\ 1\big] \in \mathcal{D}_2^{(3,n)};\ d_2^{(n)}\big(\mathbf{y}^{(n)}\big) = d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) && \text{(C.26)} \\
&\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;3}^{(3,n-1)} \implies \\
&\quad \big[\mathbf{y}^{(n-1)}\ 1\big] \in \mathcal{D}_3^{(3,n)};\ d_3^{(n)}\big(\mathbf{y}^{(n)}\big) = d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big). && \text{(C.27)}
\end{aligned}$$

The problems are the other three cases. For example,

$$\begin{aligned}
&\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;1}^{(3,n-1)} \implies \\
&\quad \big[\mathbf{y}^{(n-1)}\ 1\big] \in \mathcal{D}_1^{(3,n)} \text{ or } \mathcal{D}_2^{(3,n)} \text{ or } \mathcal{D}_3^{(3,n)} && \text{(C.28)}
\end{aligned}$$

depending on the exact value of $d_m^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$. To be able to investigate the different possible cases depending on $d_m^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$, we introduce a shorthand

$$d \triangleq \min_{m \in \{1,2,3\}} d_m^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \tag{C.29}$$

to denote the distance to the closest codeword (which is the first codeword in this case) and another shorthand d^+ to denote any value strictly larger than d. The received Hamming distance vector can take on one out of four possible values:

$$\begin{aligned}
\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d,d,d) &\text{ or } (d,d,d^+) \text{ or } (d,d^+,d) \\
&\text{or } (d,d^+,d^+). && \text{(C.30)}
\end{aligned}$$

If we append a 1 to $\mathbf{y}^{(n-1)}$, then only the first component of $\mathbf{d}^{(n)}\big(\mathbf{y}^{(n)}\big)$ will be increased by 1 in comparison to $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$, while the second and third components remains unchanged. This means that in the fourth case in (C.30), the new vector $\big[\mathbf{y}^{(n-1)}\ 1\big]$ will belong to $\mathcal{D}_1^{(3,n)}$, while in the other cases it will belong to $\mathcal{D}_2^{(3,n)}$ or $\mathcal{D}_3^{(3,n)}$. However, we will show next that the first and the second case can never occur!

To show this, first of all note that $d \geq k$ because the codebook's minimum Hamming distance between codewords is $2k-1$ and therefore it is not possible that a vector $\mathbf{y}^{(n-1)}$ has a distance to two (or more) codewords that is smaller than k. Also, from Claim C.1 it follows that $d \leq 2k-1$.

Now let's describe $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$, using $\mathbf{y}_{\max}^{(n-1)}$ defined analogously to (C.5). To that goal we define a_m to be the number of positions where $\mathbf{y}^{(n-1)}$ differs from $\mathbf{y}_{\max}^{(n-1)}$ when we only consider the t_m^* positions corresponding to $\mathbf{c}_m^{(3)}$, i.e., $0 \leq a_m \leq t_m^*$, $m = 1, 2, 3$. For example, the all-zero vector $\mathbf{y} = \mathbf{0}$ has $a_1 = t_1^*$, $a_2 = t_2^*$, and $a_3 = 0$.

Then we define a matrix

$$\begin{aligned}
&\begin{pmatrix} t_1^* - a_1 & t_2^* - a_2 & a_3 \\ t_1^* - a_1 & a_2 & t_3^* - a_3 \\ a_1 & t_2^* - a_2 & t_3^* - a_3 \end{pmatrix} \\
&= \begin{pmatrix} k - a_1 & k - a_2 & a_3 \\ k - a_1 & a_2 & k - 1 - a_3 \\ a_1 & k - a_2 & k - 1 - a_3 \end{pmatrix}
\end{aligned} \tag{C.31}$$

from which the received Hamming distance vector can be computed as follows:

$$\begin{aligned}
&\begin{pmatrix} d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \end{pmatrix} \\
&= \begin{pmatrix} k - a_1 & k - a_2 & a_3 \\ k - a_1 & a_2 & k - 1 - a_3 \\ a_1 & k - a_2 & k - 1 - a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.
\end{aligned} \tag{C.32}$$

It is straightforward to prove the following claim.

Claim C.2 *There exists no integer solution (a_1, a_2, a_3), $0 \leq a_1 \leq k$, $0 \leq a_2 \leq k$, $0 \leq a_3 \leq k-1$, that satisfies*

$$\begin{aligned}
&\begin{pmatrix} k - a_1 & k - a_2 & a_3 \\ k - a_1 & a_2 & k - 1 - a_3 \\ a_1 & k - a_2 & k - 1 - a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \\
&= \begin{pmatrix} d \\ d \\ d \end{pmatrix} \quad or \quad \begin{pmatrix} d \\ d \\ d^+ \end{pmatrix} \quad or \quad \begin{pmatrix} d \\ d^+ \\ d \end{pmatrix}
\end{aligned} \tag{C.33}$$

for $k \leq d \leq 2k-1$ and $d^+ > d$. But there do exist integer solutions that satisfy

$$\begin{pmatrix} k - a_1 & k - a_2 & a_3 \\ k - a_1 & a_2 & k - 1 - a_3 \\ a_1 & k - a_2 & k - 1 - a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d^+ \\ d \\ d \end{pmatrix}. \tag{C.34}$$

Hence, we have shown that

$$\begin{aligned}&\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;1}^{(3,n-1)} \implies \\ &\quad \left[\mathbf{y}^{(n-1)}\ 1\right] \in \mathcal{D}_1^{(3,n)};\ d_1^{(n)}\left(\mathbf{y}^{(n)}\right) = d_1^{(n-1)}\left(\mathbf{y}^{(n-1)}\right) + 1.\end{aligned} \tag{C.35}$$

Similarly,

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;2}^{(3,n-1)} \implies \left[\mathbf{y}^{(n-1)}\ 0\right] \in \mathcal{D}_1^{(3,n)} \text{ or } \mathcal{D}_2^{(3,n)} \tag{C.36}$$

depending on the exact value of $\mathbf{d}^{(n-1)}\left(\mathbf{y}^{(n-1)}\right)$. Note that $\left[\mathbf{y}^{(n-1)}\ 0\right] \notin \mathcal{D}_3^{(3,n)}$ because we have added a 1 to the third codeword. If we append a 0 to $\mathbf{y}^{(n-1)}$, then the second and the third component of $\mathbf{d}^{(n)}\left(\mathbf{y}^{(n)}\right)$ will be increased by 1 in comparison to $\mathbf{d}^{(n-1)}\left(\mathbf{y}^{(n-1)}\right)$, while the first component remains unchanged. Again, the received Hamming distance vector can take on one out of four possible values:

$$\begin{aligned}\mathbf{d}^{(n-1)}\left(\mathbf{y}^{(n-1)}\right) = (d,d,d) &\text{ or } (d,d,d^+) \text{ or } (d^+,d,d) \\ &\text{or } (d^+,d,d^+).\end{aligned} \tag{C.37}$$

In the first and two case $\left[\mathbf{y}^{(n-1)}\ 0\right]$ will change to $\mathcal{D}_1^{(3,n)}$, in the other two cases it will remain in $\mathcal{D}_2^{(3,n)}$. However, both the first and the two case are not possible according to (C.33).

Finally,

$$\begin{aligned}&\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;3}^{(3,n-1)} \implies \\ &\quad \left[\mathbf{y}^{(n-1)}\ 0\right] \in \mathcal{D}_1^{(3,n)} \text{ or } \mathcal{D}_3^{(3,n)}\end{aligned} \tag{C.38}$$

depending on the exact value of $\mathbf{d}^{(n-1)}\left(\mathbf{y}^{(n-1)}\right)$:

$$\begin{aligned}\mathbf{d}^{(n-1)}\left(\mathbf{y}^{(n-1)}\right) = (d,d,d) &\text{ or } (d^+,d,d) \text{ or } (d,d^+,d) \\ &\text{or } (d^+,d^+,d).\end{aligned} \tag{C.39}$$

In the first and third case $\left[\mathbf{y}^{(n-1)}\ 0\right]$ will change to $\mathcal{D}_1^{(3,n)}$, while in the other two cases it will remain in $\mathcal{D}_3^{(3,n)}$. Again, the first and the third case are not possible according to (C.33).

Hence, we have shown that

$$\begin{aligned}&\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;1}^{(3,n-1)} \implies \\ &\quad \left[\mathbf{y}^{(n-1)}\ 1\right] \in \mathcal{D}_1^{(3,n)};\ d_1^{(n)}\left(\mathbf{y}^{(n)}\right) = d_1^{(n-1)}\left(\mathbf{y}^{(n-1)}\right) + 1\end{aligned} \tag{C.40}$$

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;2}^{(3,n-1)} \implies$$

$$\left[\mathbf{y}^{(n-1)}\ 0\right] \in \mathcal{D}_2^{(3,n)};\ d_2^{(n)}\left(\mathbf{y}^{(n)}\right) = d_2^{(n-1)}\left(\mathbf{y}^{(n-1)}\right) + 1 \tag{C.41}$$

$$\begin{aligned}&\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;3}^{(3,n-1)} \implies \\ &\quad \left[\mathbf{y}^{(n-1)}\ 0\right] \in \mathcal{D}_3^{(3,n)};\ d_3^{(n)}\left(\mathbf{y}^{(n)}\right) = d_3^{(n-1)}\left(\mathbf{y}^{(n-1)}\right) + 1.\end{aligned} \tag{C.42}$$

But this proves that the success probability of (C.24) is identical to the success probability of (C.23)! So in spite of increasing the length $n-1$ by 1, we have not improved our performance.

Appending $\mathbf{c}_1^{(3)}$: Next, we investigate what happens if we append $\mathbf{c}_1^{(3)} = (0\ 0\ 1)^\mathsf{T}$. The new code has the following parameters:

$$[t_1, t_2, t_3] = [k+1, k, k-1] \tag{C.43}$$

$$\mathbf{d}\left(\mathscr{C}^{(3,n)}\right) = (2k-1, 2k, 2k+1) \tag{C.44}$$

$$d_{\min}\left(\mathscr{C}^{(3,n)}\right) = 2k-1. \tag{C.45}$$

One of the three problematic cases now is

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;2}^{(3,n-1)} \implies \left[\mathbf{y}^{(n-1)}\ 1\right] \in \mathcal{D}_2^{(3,n)} \text{ or } \mathcal{D}_3^{(3,n)} \tag{C.46}$$

depending on the exact value of $\mathbf{d}^{(n-1)}\left(\mathbf{y}^{(n-1)}\right)$ given in (C.37). If we append a 1 to $\mathbf{y}^{(n-1)}$, the first and the second component of $\mathbf{d}^{(n)}\left(\mathbf{y}^{(n)}\right)$ will be increased by 1 in comparison to $\mathbf{d}^{(n-1)}\left(\mathbf{y}^{(n-1)}\right)$, while the third component remains unchanged. This means that in the first and third case the new vector $\left[\mathbf{y}^{(n-1)}\ 1\right]$ will belong to $\mathcal{D}_3^{(3,n)}$, while in the second and the fourth case it will belong to $\mathcal{D}_2^{(3,n)}$. According to Claim C.2, the third case is possible and does happen. If $\left[\mathbf{y}^{(n-1)}\ 1\right] \in \mathcal{D}_3^{(3,n)}$, then we have that

$$d_3^{(n)}\left(\mathbf{y}^{(n)}\right) = d_2^{(n-1)}\left(\mathbf{y}^{(n-1)}\right) \tag{C.47}$$

without the additional increase by 1. This then means that the success probability of (C.24) is strictly larger than the success probability of $\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}$ because

$$p^{d_3^{(n)}(\mathbf{y}^{(n)})} = p^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})} \tag{C.48}$$

$$> p^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})+1} \tag{C.49}$$

and the choice of $\mathbf{c}_1^{(3)}$ is effective.

The investigation of the other two problematic cases is similar and omitted.

Appending $\mathbf{c}_2^{(3)}$: Finally, we look at the case when we append $\mathbf{c}_2^{(3)} = (0\ 1\ 0)^\mathsf{T}$. The new code has the following parameters:

$$[t_1, t_2, t_3] = [k, k+1, k-1] \tag{C.50}$$

$$\mathbf{d}\left(\mathscr{C}^{(3,n)}\right) = (2k, 2k-1, 2k+1) \tag{C.51}$$

$$d_{\min}\left(\mathscr{C}^{(3,n)}\right) = 2k-1. \tag{C.52}$$

We realize that these code parameters are simply a permutation of the parameters of the case when we append $\mathbf{c}_1^{(3)}$. Hence, the investigation will not fundamentally change and result in an identical performance. So, both choices of vectors $\mathbf{c}_1^{(3)}$ and $\mathbf{c}_2^{(3)}$ are optimal. We decide to choose $\mathbf{c}_1^{(3)}$ for keeping the ordering $t_1 \geq t_2 \geq t_3$.

C.1.2 Case ii: Step from $n-1=3k$ to $n=3k+1$

In this case, we start with the code $\mathscr{C}^{(3,n-1)}_{t_2^*,t_3^*}$ with code parameters, pairwise Hamming distance vector, and minimum Hamming distance as follows:

code parameters:

$$[t_1^*, t_2^*, t_3^*] = [k+1, k, k-1] \tag{C.53}$$

pairwise Hamming distance vector:

$$\mathbf{d}\big(\mathscr{C}^{(3,n-1)}_{t_2^*,t_3^*}\big) = (2k-1, 2k, 2k+1) \tag{C.54}$$

minimum Hamming distance:

$$d_{\min}\big(\mathscr{C}^{(3,n-1)}_{t_2^*,t_3^*}\big) = 2k-1. \tag{C.55}$$

If we append $\mathbf{c}_1^{(3)} = (0\ 0\ 1)^\mathsf{T}$, we get a new code with the following parameters:

$$[t_1, t_2, t_3] = [k+2, k, k-1] \tag{C.56}$$

$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k-1, 2k+1, 2k+2) \tag{C.57}$$

$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k-1. \tag{C.58}$$

If we append $\mathbf{c}_2^{(3)} = (0\ 1\ 0)^\mathsf{T}$, we get a new code with the following parameters:

$$[t_1, t_2, t_3] = [k+1, k+1, k-1] \tag{C.59}$$

$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k, 2k, 2k+2) \tag{C.60}$$

$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k. \tag{C.61}$$

And if we append $\mathbf{c}_3^{(3)} = (0\ 1\ 1)^\mathsf{T}$, we get a new code with the following parameters:

$$[t_1, t_2, t_3] = [k+1, k, k] \tag{C.62}$$

$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k, 2k+1, 2k+1) \tag{C.63}$$

$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k. \tag{C.64}$$

The corresponding investigation of possible situations now reads as follows.

Claim C.3 *There exists no integer solution* (a_1, a_2, a_3), $0 \leq a_1 \leq k+1$, $0 \leq a_2 \leq k$, $0 \leq a_3 \leq k-1$, *that satisfies*

$$\begin{pmatrix} k+1-a_1 & k-a_2 & a_3 \\ k+1-a_1 & a_2 & k-1-a_3 \\ a_1 & k-a_2 & k-1-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d \\ d \end{pmatrix} \text{ or } \begin{pmatrix} d \\ d \\ d^+ \end{pmatrix} \text{ or } \begin{pmatrix} d^+ \\ d \\ d \end{pmatrix} \tag{C.65}$$

for $k \le d \le 2k-1$ and $d^+ > d$. But there do exist integer solutions that satisfy

$$\begin{pmatrix} k+1-a_1 & k-a_2 & a_3 \\ k+1-a_1 & a_2 & k-1-a_3 \\ a_1 & k-a_2 & k-1-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d^+ \\ d \end{pmatrix}. \tag{C.66}$$

The investigation is similar and shows that appending $\mathbf{c}_2^{(3)}$ is strictly suboptimal, while appending $\mathbf{c}_1^{(3)}$ and $\mathbf{c}_3^{(3)}$ are equivalent and optimal. The detailed examination will be discussed in Appendix C.2.

C.1.3 Case iii: Step from $n-1 = 3k+1$ to $n = 3k+2$

In this case, we start with the code $\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}$ with code parameters, pairwise Hamming distance vector, and minimum Hamming distance as follows:

code parameters:

$$[t_1^*, t_2^*, t_3^*] = [k+1, k, k] \tag{C.67}$$

pairwise Hamming distance vector:

$$\mathbf{d}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) = (2k, 2k+1, 2k+1) \tag{C.68}$$

minimum Hamming distance:

$$d_{\min}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) = 2k. \tag{C.69}$$

If we append $\mathbf{c}_1^{(3)} = (0\ 0\ 1)^\mathsf{T}$, we get a new code with the following parameters:

$$[t_1, t_2, t_3] = [k+2, k, k] \tag{C.70}$$
$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k, 2k+2, 2k+2) \tag{C.71}$$
$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k. \tag{C.72}$$

If we append $\mathbf{c}_2^{(3)} = (0\ 1\ 0)^\mathsf{T}$, we get a new code with the following parameters:

$$[t_1, t_2, t_3] = [k+1, k+1, k] \tag{C.73}$$
$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k+1, 2k+1, 2k+2) \tag{C.74}$$
$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k+1. \tag{C.75}$$

And if we append $\mathbf{c}_3^{(3)} = (0\ 1\ 1)^\mathsf{T}$, we get a new code with the following parameters:

$$[t_1, t_2, t_3] = [k+1, k, k+1] \tag{C.76}$$
$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k+1, 2k+2, 2k+1) \tag{C.77}$$
$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k+1. \tag{C.78}$$

The corresponding investigation of possible situations now reads as follows.

Claim C.4 *There exists no integer solution* (a_1, a_2, a_3), $0 \leq a_1 \leq k+1$, $0 \leq a_2 \leq k$, $0 \leq a_3 \leq k$, *that satisfies*

$$\begin{pmatrix} k+1-a_1 & k-a_2 & a_3 \\ k+1-a_1 & a_2 & k-a_3 \\ a_1 & k-a_2 & k-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d \\ d \end{pmatrix} \text{ or } \begin{pmatrix} d^+ \\ d \\ d \end{pmatrix} \text{ or } \begin{pmatrix} d \\ d^+ \\ d \end{pmatrix} \tag{C.79}$$

for $k \leq d \leq 2k$ *and* $d^+ > d$. *But there do exist integer solutions that satisfy*

$$\begin{pmatrix} k+1-a_1 & k-a_2 & a_3 \\ k+1-a_1 & a_2 & k-a_3 \\ a_1 & k-a_2 & k-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d \\ d^+ \end{pmatrix}. \tag{C.80}$$

The investigation is similar and shows that appending $\mathbf{c}_1^{(3)}$ is strictly suboptimal, while appending $\mathbf{c}_2^{(3)}$ and $\mathbf{c}_3^{(3)}$ are equivalent and optimal.

This completes the proof for $\mathsf{M} = 3$.

Finally, we turn to the case $\mathsf{M} = 4$. We note that the fourth codeword for $\mathsf{M} = 4$ is exactly the furthest received vector for $\mathsf{M} = 3$. We can therefore adapt the computation of the received Hamming distance vector as follows:

$$\begin{pmatrix} d_1^{(n-1)}(\mathbf{y}^{(n-1)}) \\ d_2^{(n-1)}(\mathbf{y}^{(n-1)}) \\ d_3^{(n-1)}(\mathbf{y}^{(n-1)}) \\ d_4^{(n-1)}(\mathbf{y}^{(n-1)}) \end{pmatrix} = \begin{pmatrix} t_1-a_1 & t_2-a_2 & a_3 \\ t_1-a_1 & a_2 & t_3-a_3 \\ a_1 & t_2-a_2 & t_3-a_3 \\ a_1 & a_2 & a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \tag{C.81}$$

The derivation follows then exactly the same lines as for $\mathsf{M} = 3$. The only main difference is that we need to investigate more different columns. Actually, we need to investigate also some columns that have not been named in Definition 6.5. like, e.g., $\mathbf{c} = (0\ 0\ 0\ 1)^\mathsf{T}$ and prove that they are strictly suboptimal. The details are omitted.

C.2 Proof of Theorem 10.3

This proof will use the same approach as App. C.1 but is much more elaborate. Unlike the case of the ZC, we don't have the closed form for the exact average success probability for given a general codebook $\mathscr{C}^{(\mathsf{M},n)}$. Hence, to solve this global optimization problem for discrete variables, we still use the method based on induction in n. In addition, to be able to compare the total probability increase for all possible codebooks, we use the recursive construction in blocklength n for not only the locally optimal codebooks $\mathscr{C}_{\mathrm{BSC}}^{(\mathsf{M},n)*}$ given in Theorem 10.2, but also other locally optimal codebooks.

We again firstly consider the case $\mathsf{M} = 3$ and because of the similar argument in App. C.1, here in this proof we only think about the case of $n = 3k$. We summarized some important observations for our long proof.

- A principal lemma shows that how to simplify the recursive construction in block-length n by fixing one of the code parameters.
- By fixing the code parameter t_3, the free discrete variable left only is t_2 because $t_1 = n - t_2 - t_3$; we then try to find the best code parameters $[t_1^\star, t_2^\star, t_3]$ by examing all possible code parameters for the given t_3.
- We will list all the possible comparable best code parameters when we fixed the code parameter t_3.
- After obtaining the best code parameters $[t_1^\star, t_2^\star, t_3]$, we allow the t_3 to be a free discrete variable again; we then prove that the optimal code parameter is equal to $[t_1^*, t_2^*, t_3^*] = [k+1, k, k-1]$.

The following lemma shows that a better strategy for appending the new n-th column to a given $\mathscr{C}_{t_2,t_3}^{(3,n-1)}$ as we fix one of the code parameters.

Lemma C.5 *Consider the general code parameters $[t_1, t_2, t_3]$ with $t_1 \geq t_2 \geq t_3$, $t_1 + t_2 + t_3 = (n-1)$ for a BSC. Fixing one of the code parameters, if we recursively construct a locally optimal codebook for $n \geq 3$ by appending a new n-th column from one of the two other columns. The better choice to have a larger total probability increase is by appending the following choice of appended columns:*

1. *If t_3 is fixed, appending*

$$\begin{cases} \mathbf{c}_1^{(3)} & \text{if } (t_1 - t_3) \text{ is even but } (t_2 - t_3) \text{ is odd;} \\ \mathbf{c}_2^{(3)} & \text{if } (t_1 - t_3) \text{ is even and } (t_2 - t_3) \text{ is even;} \\ \mathbf{c}_2^{(3)} & \text{if } (t_1 - t_3) \text{ is odd but } (t_2 - t_3) \text{ is even;} \\ \mathbf{c}_1^{(3)} \equiv \mathbf{c}_2^{(3)} & \text{if } (t_1 - t_3) \text{ is odd and } (t_2 - t_3) \text{ is odd;} \end{cases} \tag{C.82}$$

2. *If t_2 is fixed, appending*

$$\begin{cases} \mathbf{c}_1^{(3)} & \text{if } (t_1 - t_2) \text{ is even but } (t_2 - t_3) \text{ is odd;} \\ \mathbf{c}_3^{(3)} & \text{if } (t_1 - t_2) \text{ is even and } (t_2 - t_3) \text{ is even;} \\ \mathbf{c}_3^{(3)} & \text{if } (t_1 - t_2) \text{ is odd but } (t_2 - t_3) \text{ is even;} \\ \mathbf{c}_1^{(3)} \equiv \mathbf{c}_3^{(3)} & \text{if } (t_1 - t_2) \text{ is odd and } (t_2 - t_3) \text{ is odd;} \end{cases} \tag{C.83}$$

3. If t_1 is fixed, appending

$$\begin{cases} \mathbf{c}_2^{(3)} & \text{if } (t_1 - t_2) \text{ is even but } (t_1 - t_3) \text{ is odd;} \\ \mathbf{c}_3^{(3)} & \text{if } (t_1 - t_2) \text{ is even and } (t_1 - t_3) \text{ is even;} \\ \mathbf{c}_3^{(3)} & \text{if } (t_1 - t_2) \text{ is odd but } (t_1 - t_3) \text{ is even;} \\ \mathbf{c}_2^{(3)} \equiv \mathbf{c}_3^{(3)} & \text{if } (t_1 - t_2) \text{ is odd and } (t_1 - t_3) \text{ is odd;} \end{cases} \tag{C.84}$$

Proof: Analogous to (C.31), the general code parameters $[t_1, t_2, t_3]$ with received Hamming distances $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ can be computed as follows

$$\begin{aligned} &\begin{pmatrix} d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \end{pmatrix} \\ &= \begin{pmatrix} t_1 - a_1 & t_2 - a_2 & a_3 \\ t_1 - a_1 & a_2 & t_3 - a_3 \\ a_1 & t_2 - a_2 & t_3 - a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \end{aligned} \tag{C.85}$$

Next we start clarifying the decoding regions $\mathcal{D}_m^{(3,n)}$ change depending on the n-th column we appended.

Appending $\mathbf{c}_1^{(3)}$: Following the previous discussion in App. C.1, we know that one of the problematic case is

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{t_2,t_3;1}^{(3,n-1)} \implies \big[\mathbf{y}^{(n-1)}\ 1\big] \in \mathcal{D}_1^{(3,n)} \text{ or } \mathcal{D}_3^{(3,n)} \tag{C.86}$$

depending on the exact value of $d_m^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ in (C.30), the first and third case in (C.30) will make $\big[\mathbf{y}^{(n-1)}\ 1\big]$ change to $\mathcal{D}_3^{(3,n)}$. Note that because of the first and third received Hamming distances are both equal to d, we can decode these $\mathbf{y}^{(n-1)}$ to $\mathcal{D}_{t_2,t_3;1}^{(3,n-1)}$, while in the case of $\mathbf{y}^{(n-1)} \in \mathcal{D}_{t_2,t_3;3}^{(3,n-1)}$, we will not count these $\mathbf{y}^{(n-1)}$ repeatedly.

The other two cases are

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{t_2,t_3;2}^{(3,n-1)} \implies \big[\mathbf{y}^{(n-1)}\ 1\big] \in \mathcal{D}_2^{(3,n)} \text{ or } \mathcal{D}_3^{(3,n)} \tag{C.87}$$

$$\begin{aligned} &\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{t_2,t_3;3}^{(3,n-1)} \implies \\ &\qquad \big[\mathbf{y}^{(n-1)}\ 0\big] \in \mathcal{D}_1^{(3,n)} \text{ or } \mathcal{D}_2^{(3,n)} \text{ or } \mathcal{D}_3^{(3,n)}. \end{aligned} \tag{C.88}$$

Again, in (C.37) the first and third case will make $\big[\mathbf{y}^{(n-1)}\ 1\big]$ change to $\mathcal{D}_3^{(3,n)}$. In (C.39) except the fourth case, other cases will make $\big[\mathbf{y}^{(n-1)}\ 0\big]$ change to $\mathcal{D}_1^{(3,n)}$ or $\mathcal{D}_2^{(3,n)}$. However, as mentioned before, it is not necessary to count these received Hamming distances again.

Finally, to figure out what the total probability increase are when appending $\mathbf{c}_1^{(3)}$, the whole cases that we have to take into account are

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d,d,d) \text{ or } (d,d^+,d) \text{ or } (d^+,d,d). \tag{C.89}$$

Now we study the conditions that result in the integer solutions of (a_1,a_2,a_3). *I.e.*, corresponding a positive probability increase $\Delta\psi_m$ (See the explanations from (C.47)–(C.49)).

The first case of (C.89) is a special case if both the other two cases holds. We first investigate the second case

$$d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = d = d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \tag{C.90}$$
$$\Longrightarrow (t_1 - a_1) + a_3 = a_1 + (t_3 - a_3) \tag{C.91}$$
$$\Longrightarrow (t_1 - t_3) = 2(a_1 - a_3) \tag{C.92}$$
$$\Longrightarrow a_1 - a_3 = \frac{t_1 - t_3}{2} \tag{C.93}$$

Since we only have the integer solutions a_i by assumption, hence there do exist integer solutions a_i if $t_1 - t_3$ is even. On the other hand, there exists no integer solutions a_i if $t_1 - t_3$ is odd. Similarly, in the third case of (C.89), there do exist integer solutions a_i if $t_1 - t_2$ is even. Hence, we have shown that there do exist integer solutions such that

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = \begin{cases} (d,d^+,d) \\ (d^+,d,d) \\ (d,d,d) \end{cases} \text{ if } \begin{cases} (t_1-t_3) \text{ is even} \\ (t_1-t_2) \text{ is even} \\ (t_1-t_2),\ (t_2-t_3),\ (t_1-t_3) \text{ are all even.} \end{cases} \tag{C.94}$$

Appending $\mathbf{c}_2^{(3)}$: Using the same argument, we can also show that

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = \begin{cases} (d,d,d^+) \\ (d^+,d,d) \\ (d,d,d) \end{cases} \text{ if } \begin{cases} (t_2-t_3) \text{ is even} \\ (t_1-t_2) \text{ is even} \\ (t_1-t_2),\ (t_2-t_3),\ (t_1-t_3) \text{ are all even.} \end{cases} \tag{C.95}$$

Appending $\mathbf{c}_3^{(3)}$: In this case, we have shown that

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = \begin{cases} (d, d, d^+) \\ (d, d^+, d) \\ (d, d, d) \end{cases} \text{ if } \begin{cases} (t_2 - t_3) \text{ is even} \\ (t_1 - t_3) \text{ is even} \\ (t_1 - t_2),\ (t_2 - t_3),\ (t_1 - t_3) \text{ are all even.} \end{cases} \tag{C.96}$$

Fixing t_3

To prove the first statement, when we fixed t_3, let us only allow the code parameters t_1 and t_2 to increase step 1 for building up a new $\mathscr{C}^{(3,n)}$ code. Compare (C.94) and (C.95), we can see that the condition of $(t_1 - t_2)$ is even both shows up for appending $\mathbf{c}_1^{(3)}$ or $\mathbf{c}_2^{(3)}$. Hence, for instance, if we have $(t_1 - t_3)$ is even, but $(t_2 - t_3)$ is odd. Since the case of (d, d, d^+) does not happen if $(t_1 - t_3)$ is odd and no matter $(t_1 - t_2)$ is odd or even, it is obvious that the total probability increase of appending $\mathbf{c}_1^{(3)}$ will strictly lager than the total probability increase of appending $\mathbf{c}_2^{(3)}$ (Actually, in this situation, appending $\mathbf{c}_2^{(3)}$ result in zero total probability increase, since $(t_1 - t_2)$ can not be even). This argument can be used to those cases that at least one of $(t_1 - t_3)$ or $(t_2 - t_3)$ is odd.

The most problematic case is that both $(t_1 - t_3)$ and $(t_2 - t_3)$ are even, note that there are only two possible values of the code parameters $[t_1, t_2, t_3]$ in this case, *i.e.* all t_i are odd or all t_i are even. We are going to show that appending $\mathbf{c}_2$ will result in a larger total probability increase. First introduce the shorthands

$$u \triangleq \frac{t_1 - t_3}{2}, \quad v \triangleq \frac{t_2 - t_3}{2}, \quad \bar{d} \triangleq \frac{t_1 + t_2}{2} + t_3. \tag{C.97}$$

Note that as the special case of $t_1 = t_2$, the code parameters is $[t_1, t_1, t_3]$, then appending $\mathbf{c}_1^{(3)}$ or $\mathbf{c}_2^{(3)}$ are equivalent since $[t_1 + 1, t_1, t_3] \equiv [t_1, t_1 + 1, t_3]$. W.L.O.G., we can assume that $t_1 > t_2$, then we have $u > v$.

By (C.93), we then compute the received Hamming distances

$$\begin{aligned} d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) &= d = (t_2 - a_2) + \frac{t_1 + t_3}{2}; \\ &= d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big); \\ d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) &= d^+ = a_2 + (t_1 + t_3) - (a_1 + a_3); \end{aligned} \tag{C.98}$$

with the solutions of (a_1, a_2, a_3):

$$a_1 = u + a_3, \quad a_2 \geq v + a_3. \tag{C.99}$$

Set $a_2 \triangleq v + r$ with $r \geq a_3$, then

$$d = (t_2 - v - r) + \frac{t_1 + t_3}{2} = \frac{t_1 + t_2}{2} + t_3 - r; \tag{C.100}$$

$$\begin{aligned} d^+ &= v + r + (t_1 + t_3) - (u + 2a_3) \\ &= \frac{t_1 + t_2}{2} + t_3 + (r - 2a_3). \end{aligned} \tag{C.101}$$

Reminding that the range of integer solutions a_i is $0 \leq a_i \leq t_i$, the corresponding $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ is

$$\begin{pmatrix} d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \end{pmatrix} = \begin{pmatrix} d \\ d^+ \\ d \end{pmatrix} = \begin{pmatrix} \bar{d} - r \\ \bar{d} + (r - 2a_3) \\ \bar{d} - r \end{pmatrix}, \tag{C.102}$$

for $0 \leq a_3 \leq t_3$, $a_3 \leq r \leq \frac{t_2+t_3}{2}$.

In such situation, say

$$\mathbf{y}^{(n-1)} \in \mathcal{D}_{t_2,t_3;1}^{(3,n-1)} \Longrightarrow [\mathbf{y}^{(n-1)}\ 1] \in \mathcal{D}_3^{(3,n)} \tag{C.103}$$

$$d_3^{(n)}\big(\mathbf{y}^{(n-1)}\ 1\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = d \tag{C.104}$$

depending on the solutions of $(a_1, a_2, a_3) = (u + a_3, v + r, a_3)$. And the probability increase for each $\mathbf{y}^{(n-1)}$ is

$$p^{d_3^{(n)}\left(\left[\mathbf{y}^{(n-1)}\ 1\right]\right)} - p^{d_1^{(n-1)}\left(\mathbf{y}^{(n-1)}\right)+1} = p^d - p^{d+1}. \tag{C.105}$$

Moreover, for these type of $\mathbf{y}^{(n-1)}$ with $d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = d$, there are

$$\binom{t_1}{u + a_3}\binom{t_2}{v + r}\binom{t_3}{a_3} \tag{C.106}$$

numbers of such $\mathbf{y}^{(n-1)}$.

The computation for the success probability increase of the new decoding region $\mathcal{D}_3^{(3,n)}$ can be derived as

$$\begin{aligned} \Delta\psi_3\big(\mathscr{C}_{t_2,t_3}^{(3,n)}\big) = \sum_{a_3=0}^{t_3} \sum_{r \geq a_3}^{\frac{t_2+t_3}{2}} &\binom{t_1}{u + a_3}\binom{t_2}{v + r}\binom{t_3}{a_3} \\ &\cdot (1 - \epsilon)^n \big(p^{\bar{d}-r} - p^{\bar{d}-r+1}\big). \end{aligned} \tag{C.107}$$

If we interchange the roles of r and a_3, rewrite (C.107), then

$$\begin{aligned} \Delta\psi_3\big(\mathscr{C}_{t_2,t_3}^{(3,n)}\big) = \sum_{r=0}^{\frac{t_2+t_3}{2}} \sum_{a_3=0}^{\min\{r,t_3\}} &\binom{t_1}{u + a_3}\binom{t_2}{v + r}\binom{t_3}{a_3} \\ &\cdot (1 - \epsilon)^n \big(p^{\bar{d}-r} - p^{\bar{d}-r+1}\big). \end{aligned} \tag{C.108}$$

Similarly, in the case of $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d, d, d^+)$, we have

$$\begin{aligned} d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) &= d = (t_1 - a_1) + \frac{t_2 + t_3}{2} \\ &= d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big); \\ d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) &= d^+ = a_1 + (t_2 + t_3) - (a_2 + a_3). \end{aligned} \tag{C.109}$$

with the solutions of (a_1, a_2, a_3):

$$a_1 \triangleq u + r \geq u + a_3, \quad a_2 = v + a_3. \tag{C.110}$$

As a consequence, the corresponding $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ is:

$$\begin{pmatrix} d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \end{pmatrix} = \begin{pmatrix} \bar{d} - r \\ \bar{d} - r \\ \bar{d} + (r - 2a_3) \end{pmatrix}, \tag{C.111}$$

for $0 \leq a_3 \leq t_3$, $a_3 \leq r \leq \frac{t_1+t_3}{2}$.

Likewise, the success probability increase of the new decoding region $\mathcal{D}_2^{(3,n)}$ can be derived as

$$\begin{aligned} \Delta\psi_2\big(\mathscr{C}_{t_2+1,t_3}^{(3,n)}\big) &= \sum_{a_3=0}^{t_3} \sum_{r \geq a_3}^{\frac{t_1+t_3}{2}} \binom{t_1}{u+r} \binom{t_2}{v+a_3} \binom{t_3}{a_3} \\ &\quad \cdot (1-\epsilon)^n \big(p^{\bar{d}-r} - p^{\bar{d}-r+1}\big) \end{aligned} \tag{C.112}$$

$$\begin{aligned} &= \sum_{r=0}^{\frac{t_1+t_3}{2}} \sum_{a_3=0}^{\min\{r,t_3\}} \binom{t_1}{u+r} \binom{t_2}{v+a_3} \binom{t_3}{a_3} \\ &\quad \cdot (1-\epsilon)^n \big(p^{\bar{d}-r} - p^{\bar{d}-r+1}\big). \end{aligned} \tag{C.113}$$

Therefore, compare the coefficients of (C.108) to (C.113), we are going to show that (C.113) > (C.108) with the assumption $t_1 > t_2$. This will complete the proof of statement 1).

Claim C.6 *For the nonnegative integers t_1, t_2 both are even or odd, let $t_1 > t_2$ and ν_1, ν_2 be two nonnegative integers with $\frac{t_2}{2} \geq \nu_1 > \nu_2 \geq 0$, we have*

$$\begin{aligned} &\binom{t_1}{\lceil \frac{t_1}{2} \rceil + \nu_1} \binom{t_2}{\lceil \frac{t_2}{2} \rceil + \nu_2} - \binom{t_1}{\lceil \frac{t_1}{2} \rceil + \nu_2} \binom{t_2}{\lceil \frac{t_2}{2} \rceil + \nu_1} \\ &= \binom{t_1}{\lfloor \frac{t_1}{2} \rfloor - \nu_1} \binom{t_2}{\lceil \frac{t_2}{2} \rceil + \nu_2} - \binom{t_1}{\lceil \frac{t_1}{2} \rceil + \nu_2} \binom{t_2}{\lfloor \frac{t_2}{2} \rfloor - \nu_1} \\ &= \binom{t_1}{\lfloor \frac{t_1}{2} \rfloor - \nu_1} \binom{t_2}{\lfloor \frac{t_2}{2} \rfloor - \nu_2} - \binom{t_1}{\lfloor \frac{t_1}{2} \rfloor - \nu_2} \binom{t_2}{\lfloor \frac{t_2}{2} \rfloor - \nu_1} \\ &= \binom{t_1}{\lceil \frac{t_1}{2} \rceil + \nu_1} \binom{t_2}{\lfloor \frac{t_2}{2} \rfloor - \nu_2} - \binom{t_1}{\lfloor \frac{t_1}{2} \rfloor - \nu_2} \binom{t_2}{\lceil \frac{t_2}{2} \rceil + \nu_1} > 0 \end{aligned} \tag{C.114}$$

The proof of Claim C.6 can be shown as following, note that $\binom{t}{\lfloor \frac{t}{2} \rfloor - \nu}$ is equal to $\binom{t}{\lceil \frac{t}{2} \rceil + \nu}$ is clear from the definition of binomial coefficients. We then only prove the first equation of (C.114) for the case that t_1, t_2 are both even. Write

$$\begin{aligned}\binom{t_1}{\frac{t_1}{2}+\nu_1}\binom{t_2}{\frac{t_2}{2}+\nu_2} = &\frac{\left(\frac{t_1}{2}-\nu_1+1\right)\cdots\left(\frac{t_1}{2}\right)}{\left(\frac{t_1}{2}+\nu_1\right)\cdots\left(\frac{t_1}{2}+1\right)} \times \frac{t_1!}{\frac{t_1}{2}!\frac{t_1}{2}!} \\ &\times \frac{\left(\frac{t_2}{2}-\nu_2+1\right)\cdots\left(\frac{t_2}{2}\right)}{\left(\frac{t_2}{2}+\nu_2\right)\cdots\left(\frac{t_2}{2}+1\right)} \times \frac{t_2!}{\frac{t_2}{2}!\frac{t_2}{2}!}\end{aligned} \tag{C.115}$$

$$\begin{aligned}\binom{t_1}{\frac{t_1}{2}+\nu_2}\binom{t_2}{\frac{t_2}{2}+\nu_1} = &\frac{\left(\frac{t_1}{2}-\nu_2+1\right)\cdots\left(\frac{t_1}{2}\right)}{\left(\frac{t_1}{2}+\nu_2\right)\cdots\left(\frac{t_1}{2}+1\right)} \times \frac{t_1!}{\frac{t_1}{2}!\frac{t_1}{2}!} \\ &\times \frac{\left(\frac{t_2}{2}-\nu_1+1\right)\cdots\left(\frac{t_2}{2}\right)}{\left(\frac{t_2}{2}+\nu_1\right)\cdots\left(\frac{t_2}{2}+1\right)} \times \frac{t_2!}{\frac{t_2}{2}!\frac{t_2}{2}!}\end{aligned} \tag{C.116}$$

Divide (C.115) by (C.116), since $t_1 > t_2$ and $\nu_1 > \nu_2$, we get

$$\frac{\binom{t_1}{\frac{t_1}{2}+\nu_1}\binom{t_2}{\frac{t_2}{2}+\nu_2}}{\binom{t_1}{\frac{t_1}{2}+\nu_2}\binom{t_2}{\frac{t_2}{2}+\nu_1}} = \frac{\left(\frac{\frac{t_1}{2}-\nu_1+1}{\frac{t_1}{2}+\nu_1}\right)\cdots\left(\frac{\frac{t_1}{2}-\nu_2}{\frac{t_1}{2}+\nu_2+1}\right)}{\left(\frac{\frac{t_2}{2}-\nu_1+1}{\frac{t_2}{2}+\nu_1}\right)\cdots\left(\frac{\frac{t_2}{2}-\nu_2}{\frac{t_2}{2}+\nu_2+1}\right)} > 1. \tag{C.117}$$

where the inequality of (C.117) is because the fact that with $b, c, e \in \mathbb{N}$,

$$\frac{b}{c} > \frac{b-e}{c-e} \quad \text{provided that } e < b < c. \tag{C.118}$$

This complete the proof of Claim C.6.

In the rest of proof, we also only deal with the case that t_i are all even. Now we substracting (C.108) from (C.113), we have

$$\begin{aligned}&\Delta\psi_2\left(\mathscr{C}^{(3,n)}_{t_2+1,t_3}\right) - \Delta\psi_3\left(\mathscr{C}^{(3,n)}_{t_2,t_3}\right) \\ &= \sum_{r=0}^{\frac{t_2+t_3}{2}} \sum_{a_3=0}^{\min\{r,t_3\}} \left[\binom{t_1}{u+r}\binom{t_2}{v+a_3} - \binom{t_1}{u+a_3}\binom{t_2}{v+r}\right] \\ &\quad \cdot \binom{t_3}{a_3}(1-\epsilon)^n\left(p^{\bar{d}-r} - p^{\bar{d}-r+1}\right) \\ &\quad + \sum_{r=\frac{t_2+t_3}{2}+1}^{\frac{t_1+t_3}{2}} \sum_{a_3=0}^{\min\{r,t_3\}} \binom{t_1}{u+r}\binom{t_2}{v+a_3}\binom{t_3}{a_3} \\ &\quad \cdot (1-\epsilon)^n\left(p^{\bar{d}-r} - p^{\bar{d}-r+1}\right).\end{aligned} \tag{C.119}$$

We observed that the second term of (C.119) is strictly larger than zero. The first term

of (C.119) can be rewritten as

$$\begin{aligned}
&\sum_{r=0}^{t_3}\sum_{a_3=0}^{r}\left[\binom{t_1}{u+r}\binom{t_2}{v+a_3}-\binom{t_1}{u+a_3}\binom{t_2}{v+r}\right]\\
&\quad\cdot\binom{t_3}{a_3}(1-\epsilon)^n\left(p^{\bar{d}-r}-p^{\bar{d}-r+1}\right)\\
&+\sum_{r=t_3+1}^{\frac{t_2+t_3}{2}}\sum_{a_3=0}^{t_3}\left[\binom{t_1}{u+r}\binom{t_2}{v+a_3}-\binom{t_1}{u+a_3}\binom{t_2}{v+r}\right]\\
&\quad\cdot\binom{t_3}{a_3}(1-\epsilon)^n\left(p^{\bar{d}-r}-p^{\bar{d}-r+1}\right)
\end{aligned}\tag{C.120}$$

Then we consider the first term binomial coefficients of (C.120), they are equal to

$$\begin{aligned}
&\left[\binom{t_1}{\frac{t_1}{2}+r-\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+a_3-\frac{t_3}{2}}\right.\\
&\quad\left.-\binom{t_1}{\frac{t_1}{2}+a_3-\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+r-\frac{t_3}{2}}\right]\binom{t_3}{a_3}.
\end{aligned}\tag{C.121}$$

Because of $t_3 \geq r \geq a_3 \geq 0$, observed that if $r = a_3$, we have

$$\begin{aligned}
&\binom{t_1}{\frac{t_1}{2}+r-\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+a_3-\frac{t_3}{2}}\\
&\quad-\binom{t_1}{\frac{t_1}{2}+a_3-\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+r-\frac{t_3}{2}}=0
\end{aligned}\tag{C.122}$$

If $r + a_3 = t_3$, then we also get

$$\begin{aligned}
&\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-r}\binom{t_2}{\frac{t_2}{2}+\frac{t_3}{2}-r}\\
&\quad-\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-r}\binom{t_2}{\frac{t_2}{2}+r-\frac{t_3}{2}}=0
\end{aligned}\tag{C.123}$$

If neither $r - a_3 = 0$ nor $r + a_3 = t_3$, in general, we have

$$r - a_3 = 1, \cdots, t_3 - 1. \tag{C.124}$$

We will next illustate the case that $r - a_3 = 1 \Rightarrow r = a_3 + 1$, then (C.121) becomes

$$\begin{aligned}
&\left[\binom{t_1}{\frac{t_1}{2}+a_3+1-\frac{t_3}{2}}\binom{t_2}{\frac{t_1}{2}+a_3-\frac{t_3}{2}}\right.\\
&\left.-\binom{t_1}{\frac{t_1}{2}+a_3-\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+a_3+1-\frac{t_3}{2}}\right]\binom{t_3}{a_3}
\end{aligned}\tag{C.125}$$

If $a_3 \leq \frac{t_3}{2} - 1$, then (C.125) equal to

$$\begin{aligned}
&\left[\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-a_3-1}\binom{t_2}{\frac{t_1}{2}+\frac{t_3}{2}-a_3}\right.\\
&\left.-\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-a_3}\binom{t_2}{\frac{t_2}{2}+\frac{t_3}{2}-a_3-1}\right]\binom{t_3}{a_3},
\end{aligned}\tag{C.126}$$

which is smaller than zero due to Claim C.6. However, if $a_3 > \frac{t_3}{2} - 1$, then (C.125) becomes

$$\left[\binom{t_1}{\frac{t_1}{2}+a_3-\frac{t_3}{2}+1}\binom{t_2}{\frac{t_1}{2}+a_3-\frac{t_3}{2}} - \binom{t_1}{\frac{t_1}{2}+a_3-\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+a_3-\frac{t_3}{2}+1}\right]\binom{t_3}{a_3}, \tag{C.127}$$

which is larger than zero. Reminding that the range of a_3 is from 0 to t_3-1, taking $a_3 = 0$ ($r = 1$) for (C.126) and $a_3' = t_3 - 1$ ($r = t_3$) for (C.127), sum up them together

$$\begin{aligned}
&\left[\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-1}\binom{t_2}{\frac{t_1}{2}+\frac{t_3}{2}} - \binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+\frac{t_3}{2}-1}\right]\binom{t_3}{0} \\
&+\left[\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}}\binom{t_2}{\frac{t_1}{2}+\frac{t_3}{2}-1} - \binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-1}\binom{t_2}{\frac{t_2}{2}+\frac{t_3}{2}}\right]\binom{t_3}{t_3-1} \qquad \text{(C.128)} \\
&= \left[\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}}\binom{t_2}{\frac{t_1}{2}+\frac{t_3}{2}-1} - \binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-1}\binom{t_2}{\frac{t_2}{2}+\frac{t_3}{2}}\right] \\
&\cdot\left[\binom{t_3}{t_3-1} - \binom{t_3}{0}\right] > 0, \qquad \text{(C.129)}
\end{aligned}$$

where (C.129) is due to that $\binom{t_3}{t_3-1} > \binom{t_3}{0}$. Furthermore, since $\left(p^{\bar{d}-r} - p^{\bar{d}-r+1}\right)$ is strictly increasing in r, hence

$$\begin{aligned}
&\left[\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}}\binom{t_2}{\frac{t_1}{2}+\frac{t_3}{2}-1} - \binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-1}\binom{t_2}{\frac{t_2}{2}+\frac{t_3}{2}}\right] \\
&\cdot\left[\binom{t_3}{t_3-1}\left(p^{\bar{d}-t_3} - p^{\bar{d}-t_3+1}\right) - \binom{t_3}{0}\left(p^{\bar{d}} - p^{\bar{d}+1}\right)\right] \\
&> 0 \qquad \text{(C.130)}
\end{aligned}$$

Similarly, for the other cases that $a_3 + a_3' = t_3 - 1$, the binomial coefficients terms of $a_3' > \frac{t_3}{2} - 1$ will always compensate for the binomial coefficients terms of $a_3 \le \frac{t_3}{2} - 1$. These would make the whole summations are still larger than zero for the case of $r - a_3 = 1$. For the rest cases that $r - a_3 = \kappa$, $\kappa = 2, \ldots, t_3 - 1$, we can apply the similar argument to show that the first summation term of (C.120) is always larger than zero. The final step is showing that the second term of (C.120) is always strictly larger than zero, too.

Again, consider the ranges of (r, a_3): $t_3 \le r \le \frac{t_2+t_3}{2}$, $0 \le a_3 \le t_3$, for instance, substitute $r = t_3 + 1$ in (C.121), we have

$$\left[\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}+1}\binom{t_2}{\frac{t_2}{2}+a_3-\frac{t_3}{2}} - \binom{t_1}{\frac{t_1}{2}+a_3-\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+\frac{t_3}{2}+1}\right]\binom{t_3}{a_3}. \tag{C.131}$$

Since $0 \le a_3 \le t_3$, then $\left|a_3 - \frac{t_3}{2}\right| \le \frac{t_3}{2}$, therefore, apply Claim C.6 again, (C.131) is always lager than zero. This complete the proof that (C.113)> (C.108).

The second and third statement of Lemma C.5 can be proved in a similar way. □

According to the proof in Lemma C.5, we can also recursively compute the total probability increase in blocklength n for applying each case in Theorem 10.2.

Corollary C.7 *For a BSC and for any $n \ge 2$, the optimal codes exact average success probability with three codewords $\mathsf{M} = 3$ can be derived recursively in blocklength n. Starting as (10.6), and then apply (10.7)–(10.9).*

Proof: It is quite simple to get the starting expression (10.6) for $n = 2$ from (11.1).

We only illustrate the calculation for the case as $n - 1 = 3k - 1$ to $n = 3k$. The optimal code parameters for $n - 1 = 3k - 1$ is $[k, k, k-1]$. Since we are going to append $\mathbf{c}_1^{(3)}$ for $n = 3k$, the solutions of (a_1, a_2, a_3) for $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d^+, d, d)$ are

$$a_1 = a_2, \quad a_2 \le a_3. \tag{C.132}$$

The corresponding $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ is

$$\begin{pmatrix} d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \end{pmatrix} = \begin{pmatrix} 2k + (a_3 - 2a_2) \\ (2k-1) - a_3 \\ (2k-1) - a_3 \end{pmatrix}, \tag{C.133}$$

for $0 \le a_2 \le t_2$, $a_2 \le a_3 \le t_3$.

Therefore, the total probability increase from the third decoding region $\mathcal{D}_3^{(3,n)}$ is

$$\Delta\Psi\big(\mathscr{C}_{k,k-1}^{(3,n)}\big) = \sum_{a_3=0}^{k-1} \sum_{a_2=0}^{a_3} \binom{k}{a_2}\binom{k}{a_2}\binom{k-1}{a_3} \cdot (1-\epsilon)^n \big(p^{2k-1-a_3} - p^{2k-a_3}\big). \tag{C.134}$$

The other two cases are similar to (C.108) and (C.113). □

Now we back into the proof of Theorem 10.3, we have a important strictly increasing property in t_2.

Corollary C.8 *For a BSC and for any $n \ge 2$ with any code parameters $[t_1, t_2, t_3]$ that satisfy $t_1 \ge t_2 + 2 \ge t_3$, we have*

$$3P_{\mathrm{c}}\big(\mathscr{C}_{t_2,t_3}^{(3,n)}\big) < 3P_{\mathrm{c}}\big(\mathscr{C}_{t_2+2,t_3}^{(3,n)}\big). \tag{C.135}$$

Proof: Consider the codebook $\mathscr{C}_{t_2,t_3}^{(3,n-2)}$ with code parameters $[t_1, t_2, t_3]$. From blocklength $n-2$ to blocklength n by fixing the number of t_3, there are three possible code parameters extensions that will happen in blocklength n. The following picture C.17 shows that how to recursively construct the code parameters corresponding to their average success probabilities.

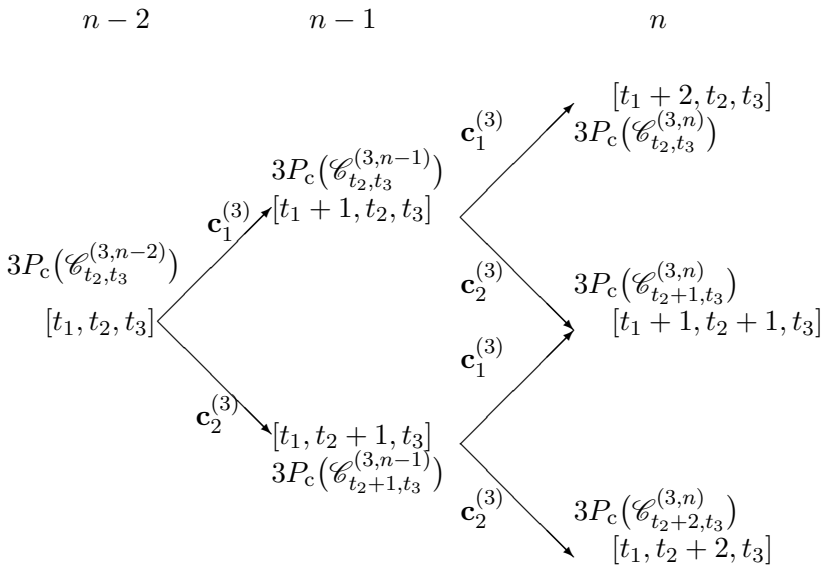


Figure C.17: The code parameters construction from blocklength $n-2$ to n

The condition of $t_1 \geq t_2 + 2$ is needed to make sure that at blocklength n, the code parameters ordering is still nonincreasing.

Using the same approach as Lemma C.5, there are four cases have to be concerned in the staring case at blocklength $n-2$. *I.e.* $(t_1 - t_3)$ is even and $(t_2 - t_3)$ is even, $(t_1 - t_3)$ is even but $(t_2 - t_3)$ is odd, $(t_1 - t_3)$ is odd and $(t_2 - t_3)$ is even, or $(t_1 - t_3)$ is odd and $(t_2 - t_3)$ is odd.

The two cases that only one of the two differences $(t_1 - t_3)$ or $(t_2 - t_3)$ is odd are simple to argue. For instance, if $(t_1 - t_3)$ is even but $(t_2 - t_3)$ is odd, then by Lemma C.5, we have

$$3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n-1)}_{t_2,t_3}\big) > 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n-1)}_{t_2+1,t_3}\big). \tag{C.136}$$

However, both $(t_1 + 1 - t_3)$ and $(t_2 - t_3)$ are odd, therefore

$$3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{t_2,t_3}\big) = 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{t_2+1,t_3}\big). \tag{C.137}$$

Furthermore, from $[t_1, t_2 + 1, t_3]$, due to that both $(t_1 - t_3)$ and $(t_2 + 1 - t_3)$ are even, by Lemma C.5 again, we have

$$3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{t_2+2,t_3}\big) > 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{t_2+1,t_3}\big) \tag{C.138}$$

$$= 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{t_2,t_3}\big). \tag{C.139}$$

The case that both $(t_1 - t_3)$ and $(t_2 - t_3)$ are even can be shown as follows, according to Lemma C.5, then

$$3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n-1)}_{t_2+1,t_3}\big) > 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n-1)}_{t_2,t_3}\big). \tag{C.140}$$

Since (t_1+1-t_3) is even but (t_2-t_3) is odd, and (t_1-t_3) is even but (t_2+1-t_3) is odd, we have

$$3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{t_2,t_3}\big) = 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n-1)}_{t_2,t_3}\big) \tag{C.141}$$

$$< 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n-1)}_{t_2+1,t_3}\big) = 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{t_2+2,t_3}\big). \tag{C.142}$$

Finally, for the case of both (t_1-t_3) and (t_2-t_3) are odd. Because of

$$3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n-1)}_{t_2,t_3}\big) = 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n-1)}_{t_2+1,t_3}\big). \tag{C.143}$$

Now since (t_1+1-t_3) and (t_2+1-t_3) are even, and by assumption $(t_1+1-t_3) \geq (t_2+1-t_3)$. Using the similar discussion in the proof of C.5, we can also show that

$$3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{t_2,t_3}\big) = 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n-1)}_{t_2,t_3}\big) + \Delta\Psi\big(\mathscr{C}^{(3,n)}_{t_2,t_3}\big) \tag{C.144}$$

$$< 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n-1)}_{t_2+1,t_3}\big) + \Delta\Psi\big(\mathscr{C}^{(3,n)}_{t_2+2,t_3}\big) \tag{C.145}$$

$$= 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{t_2+2,t_3}\big). \tag{C.146}$$

□

Corollary C.8 will lead us to get the optimized code parameters $[t_1^\star, t_2^\star, t_3]$ while fixing t_3.

Corollary C.9 *For a BSC and for any $n \geq 2$, set $n = 3k$ with any code parameters $[t_1, t_2, t_3]$ satisfy $t_1 \geq t_2 \geq t_3$. Fixing $t_3 = k-\kappa$, $0 \leq \kappa \leq k$. Then the best code parameters among all possible t_2 for every given t_3 is*

$$[t_1^\star, t_2^\star, k-\kappa] = \begin{cases} [k,k,k] \\ \left[k+\left\lceil \frac{\kappa}{2} \right\rceil, k+\left\lfloor \frac{\kappa}{2} \right\rfloor, k-\kappa\right] \\ \left[k+\left\lceil \frac{\kappa}{2} \right\rceil+1, k+\left\lfloor \frac{\kappa}{2} \right\rfloor-1, k-\kappa\right] \end{cases}$$

$$\text{if} \quad \begin{cases} \kappa = 0 \\ \kappa \bmod 4 = 1,2,3 \\ \kappa \bmod 4 = 0,\ \kappa \neq 0. \end{cases} \tag{C.147}$$

In the proofs later on, we again introduce the shorthand:

$$\eta \triangleq \left\lfloor \frac{\kappa}{4} \right\rfloor. \tag{C.148}$$

Proof: Due to $n = 3k$, then $t_1 + t_2 = 2k + \kappa$. Hence, by Cor. C.8, we know that while fixing t_3, the average success probability have a strictly increasing property in t_2 to $t_2 + 2$, also since $t_1 \geq t_2 \geq t_3 = k - \kappa$, the two possible best code parameters can only be either

$$\left[k+\left\lceil \frac{\kappa}{2} \right\rceil, k+\left\lfloor \frac{\kappa}{2} \right\rfloor, k-\kappa\right] \text{ or}$$

$$\left[k+\left\lceil \frac{\kappa}{2} \right\rceil+1, k+\left\lfloor \frac{\kappa}{2} \right\rfloor-1, k-\kappa\right] \tag{C.149}$$

In the case of $\kappa = 0$, the only possible code parameters is the first one of (C.149): $[k, k, k]$ because of $k - 1 < k$.

We now illustrate the case of $\kappa \bmod 4 = 1$, let $\kappa = 4\eta + 1$. Then the two possible best code parameters become either

$$\begin{aligned} &[k + 2\eta + 1, k + 2\eta, k - (4\eta + 1)] \text{ or} \\ &[k + 2\eta + 2, k + 2\eta - 1, k - (4\eta + 1)]\,. \end{aligned} \tag{C.150}$$

The two possible best code parameters are both from the $n - 1$ code parameters:

$$[k + 2\eta + 1, k + 2\eta - 1, k - (4\eta + 1)] \tag{C.151}$$

Since $k + 2\eta + 1 - (k - (4\eta + 1)) = 6\eta + 2$ and $k + 2\eta - 1 - (k - (4\eta + 1)) = 6\eta$ both are even, by Lemma C.5, we have

$$3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{k+2\eta,k-(4\eta+1)}\big) > 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{k+2\eta-1,k-(4\eta+1)}\big). \tag{C.152}$$

The remaining proofs of other cases are similar and omitted. □

Finally, to finish the proof that verifying the code parameters $[k + 1, k, k - 1]$ is the global optimal code. We are using Cor. C.9 and Lemma C.5 to complete the proof. Note that in the proof of Theorem 10.2, we have shown that

$$3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{k,k}\big) < 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{k,k-1}\big). \tag{C.153}$$

Claim C.10 *For $\kappa \geq 1$, the average success probability is decreasing in κ among all $[t_1^\star, t_2^\star, k - \kappa]$ given in (C.147). In particular, we have*

$$3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{t_2^\star,k-\kappa}\big) \geq 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{t_2^\star,k-(\kappa+1)}\big). \tag{C.154}$$

Proof: In the case of $\kappa = 4\eta+1$, $\kappa+1 = 4\eta+2$. The best code parameters $[t_1^\star, t_2^\star, t_3]$ are

$$[k + 2\eta + 1, k + 2\eta, k - (4\eta + 1)]\,, \tag{C.155}$$
$$[k + 2\eta + 1, k + 2\eta + 1, k - (4\eta + 2)]\,, \tag{C.156}$$

respectively. These two best code parameters are from the $n - 1$ blocklength code parameters

$$[k + 2\eta + 1, k + 2\eta, k - (4\eta + 2)]. \tag{C.157}$$

Again, use Lemma C.5, due to that both $\big(k+2\eta+1-(k+2\eta)\big) = 1$ and $(k + 2\eta + 1 - (k - (4\eta + 2))) = 6\eta + 3$ are odd, thus

$$3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{k+2\eta,k-(4\eta+1)}\big) = 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{k+2\eta+1,k-(4\eta+2)}\big). \tag{C.158}$$

In the case of $\kappa = 4\eta + 2$. The best code parameters $[t_1^\star, t_2^\star, t_3]$ are

$$[k + 2\eta + 1, k + 2\eta + 1, k - (4\eta + 2)], \tag{C.159}$$
$$[k + 2\eta + 2, k + 2\eta + 1, k - (4\eta + 3)], \tag{C.160}$$

respectively. Both of those are from

$$[k + 2\eta + 1, k + 2\eta + 1, k - (4\eta + 3)] \tag{C.161}$$

Since $\big(k + 2\eta + 1 - (k + 2\eta + 1)\big) = 0$ and $(k + 2\eta + 1 - (k - (4\eta + 3))) = 6\eta + 4$ are even, thus

$$3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{k+2\eta,k-(4\eta+2)}\big) > 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{k+2\eta+1,k-(4\eta+3)}\big) \tag{C.162}$$

The remaining cases are similar and omitted. As a consequence, we have

$$3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{k+2\eta+1,k-(4\eta+3)}\big) = 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{k+2\eta+1,k-(4\eta+4)}\big), \tag{C.163}$$
$$3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{k+2\eta+1,k-(4\eta+4)}\big) > 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{k+2\eta+2,k-(4\eta+5)}\big). \tag{C.164}$$

This complete the proof. □

Because we have shown that $3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{t_2^\star,(k-\kappa)}\big)$ is decreasing in $\kappa \geq 1$ and $3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{k,k-1}\big) > 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{k,k}\big)$, we then prove that $3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}_{k,k-1}\big)$ is the largest average success probability we can get among all possible code parameters $[t_1, t_2, t_3]$ satisfy that $t_1 \geq t_2 \geq t_3$. Note that in the proof of Claim C.10. According to (C.158), we know that for $n = 3k$, there are two global optimal code parameters $[t_1^*, t_2^*, t_3^*] = [k + 1, k, k - 1]$ or $[k + 1, k + 1, k - 2]$.

In the cases of $n = 3k + 1$, $3k + 2$ are similar to those argument above.

In the end, we turn to the case of $\mathsf{M} = 4$, we firstly have to remark that the linear optimal codes of $\mathsf{M} = 4$ can be constructed from the the weak flip codes of type (t_2, t_3). Using the same derivations as we early discussed that $(t_1 - t_2)$, $(t_2 - t_3)$ or $(t_1 - t_3)$ is even or odd. This can be shown by using the adapt computation of the received Hamming distance vector as (C.81). The other derivations will follow the same line as the proofs of $\mathsf{M} = 3$. The details are omitted.

Appendix D

Derivations concerning the BEC

D.1 Proof of Theorem 11.2

In the proof of Theorem 11.2, our goal is to maximize $\Delta\Psi\big(\mathscr{C}^{(\mathsf{M},\gamma)}\big)$ among all possible $\mathscr{C}^{(\mathsf{M},\gamma)}$; hence, for every blocklength n, we can maximize the improvement of performance. Note that our optimal codes based on an important assumption: if the optimal codes can be constructed recursively in maximizing the improvement of performance for every blocklength n. This induction proof for a BEC follows the lines of the proof for the BSC shown in [16, App. C] with some modifications that take into account the details of the decoding rules for the BEC. Similarly to [16, App. C], we need a case distinction depending on n mod 3. For space reason, we only outline the case from $n = 3k-1$ to $n = 3k$.

For $\mathsf{M} = 3$, we note that similarly to the proof for the BSC and due to the symmetry of the BEC (see Lemma 3.1), we can reduce the number of candidate columns to $\mathbf{c}_1^{(3)}, \mathbf{c}_2^{(3)}, \mathbf{c}_3^{(3)}$. We start with the code $\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}$, whose code parameters, pairwise Hamming distance vector, and minimum Hamming distance are as follows:

$$[t_1^*, t_2^*, t_3^*] = [k, k-1, k]; \tag{D.1}$$

$$\mathbf{d}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) = (2k-1, 2k, 2k-1); \tag{D.2}$$

$$d_{\min}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) = 2k-1. \tag{D.3}$$

We require to show that appending $\mathbf{c}_2^{(3)}$ yields a larger success probability than appending $\mathbf{c}_1^{(3)}$ or $\mathbf{c}_3^{(3)}$. Note that appending $\mathbf{c}_1^{(3)}$ will result in the same success probability as appending $\mathbf{c}_3^{(3)}$.

Consider the three possible extended decoding regions of blocklength n, i.e., $\big[\mathcal{D}_m^{(n-1)}\ 0\big]$, $\big[\mathcal{D}_m^{(n-1)}\ 1\big]$, and $\big[\mathcal{D}_m^{(n-1)}\ 2\big]$. Owing to $P_{Y|X}(0|1) = P_{Y|X}(1|0) = 0$, we know for the mth new codeword of blocklength n with $x_{m,n} = b$, where $b \in \{0,1\}$, its extended decoding region $\mathcal{D}_m^{(n)}$ should include both $\big[\mathcal{D}_m^{(n-1)}\ b\big]$ and $\big[\mathcal{D}_m^{(n-1)}\ 2\big]$, and all the received vectors in $\big[\mathcal{D}_m^{(n-1)}\ \bar{b}\big]$ will be decoded to one of the other two codewords. Since $\psi_m^{(n-1)}$ is equal to the occurrence probabilities of those received vectors in the union of $\big[\mathcal{D}_m^{(n-1)}\ b\big]$ and $\big[\mathcal{D}_m^{(n-1)}\ 2\big]$, $\psi_m^{(n)}$ is no less than $\psi_m^{(n-1)}$. As a result, the increment of success probability for

each codeword will be determined by how the received vectors in $\left[\mathcal{D}_m^{(n-1)}\ \bar{b}\right]$ are decoded to the other two codewords.

The following claim is going to help answering this question.

Claim D.1 *Let m, m' and m'' be distinct numbers in $\{1,2,3\}$. If $d_{\mathrm{H}}\big(\mathbf{x}_m^{(n-1)}, \mathbf{x}_{m'}^{(n-1)}\big) \geq d_{\mathrm{H}}\big(\mathbf{x}_m^{(n-1)}, \mathbf{x}_{m''}^{(n-1)}\big)$ and if $x_{m,n} = b$ is different from $x_{m',n} = x_{m'',n} = \bar{b}$, then the received vectors in $\left[\mathcal{D}_m^{(n-1)}\ \bar{b}\right]$ should be assigned to $\mathcal{D}_{m''}^{(n)}$ rather than to $\mathcal{D}_{m'}^{(n)}$, as this will result in a higher success probability.*

Proof of Claim D.1: To facilitate the explanation of our idea behind the proof of Claim D.1, we assume without loss of generality that $m = 1$, $m' = 2$ and $m'' = 3$, and consider $\mathbf{y}^{(n-1)} \in \mathcal{D}_1^{(n-1)}$, whose components must be either an erasure 2 or equal to the corresponding component of the first codeword: $y_j \in \{x_{1,j}, 2\}$ (where actually $x_{1,j} = 0$; also note that since $m = 1$, we have $b = 0$). Now we investigate all those length-n received vectors $\mathbf{y}^{(n)}$ in $\left[\mathcal{D}_1^{(n-1)}\ \bar{b}\right]$ with positive probability. Note that because of the last digit $y_n = \bar{b} = 1$ these vectors cannot be assigned to $\mathcal{D}_1^{(n)}$.

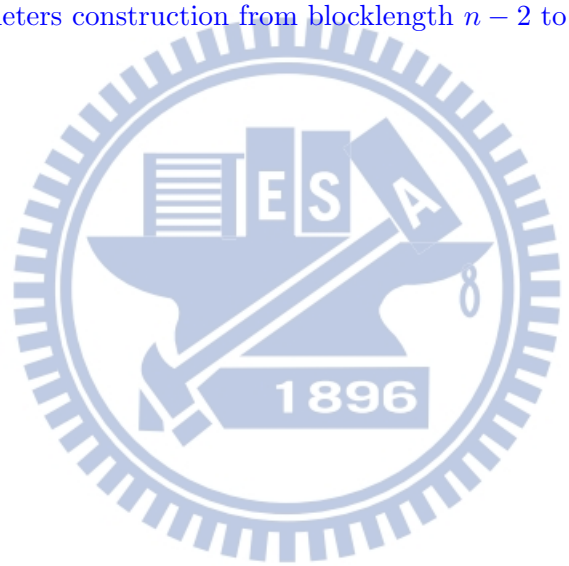
If there exists a position y_j of $\mathbf{y}^{(n)}$ that corresponds to a code matrix column $\mathbf{c}_1^{(3)}$ and that takes value $y_j = x_{1,j}$ $(= 0)$, then this received vector must be assigned to $\mathcal{D}_2^{(n)}$, where we can infer from the assumption of $\mathbf{y}^{(n)}$ having positive probability that all positions in $\mathbf{y}^{(n)}$ corresponding to code matrix columns $\mathbf{c}_2^{(3)}$ or $\mathbf{c}_3^{(3)}$ must be erased to 2. Likewise, if there exists a position y_j that corresponds to a code matrix column $\mathbf{c}_2^{(3)}$ and $y_j = x_{1,j}$ $(= 0)$, then such received vectors will be classified to $\mathcal{D}_3^{(n)}$, where we can infer that all positions of $\mathbf{y}^{(n)}$ corresponding to code matrix columns $\mathbf{c}_1^{(3)}$ or $\mathbf{c}_3^{(3)}$ must be 2.

Since by assumption $d_{\mathrm{H}}\big(\mathbf{x}_1^{(n-1)}, \mathbf{x}_2^{(n-1)}\big)$ is larger than $d_{\mathrm{H}}\big(\mathbf{x}_1^{(n-1)}, \mathbf{x}_3^{(n-1)}\big)$, in the code matrix of length $n-1$, $\mathbf{c}_2^{(3)}$ will occur more often than $\mathbf{c}_1^{(3)}$. We will therefore gain a higher increase in the success probability if the vectors in $\left[\mathcal{D}_1^{(n-1)}\ \bar{b}\right]$ are assigned to $\mathcal{D}_3^{(n)}$. □

Using a similar approach as shown in the proof of Claim D.1 together with $d_{12}^{(n-1)} = 2k-1 < d_{13}^{(n-1)} = 2k$, we can proceed to show that we gain a larger increment of success probability if we append $\mathbf{c}_2^{(3)}$ as the nth code matrix column rather than appending $\mathbf{c}_1^{(3)}$. This then completes the proof of the exemplified special case in Theorem 11.2.

Similar arguments can be applied to $\mathsf{M} = 4$.

List of Figures



Bibliography

[1] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, July and October 1948.

[2] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[3] M. C. Gursoy, "Throughput analysis of buffer-constrained wireless systems in the finite blocklength regime," in *Proceedings IEEE International Conference on Communications (ICC)*, Kyoto, Japan, June 5–9, 2011.

[4] T. J. Riedl, T. P. Coleman, and A. C. Singer, "Finite block-length achievable rates for queuing timing channels," in *Proceedings IEEE Information Theory Workshop (ITW)*, Paraty, Brazil, October 16–20, 2011, pp. 200–204.

[5] A. Martinez and A. Guillén i Fàbregas, "Saddlepoint approximation of random coding bounds," in *Proceedings Information Theory and Applications Workshop (ITA), University of California*, San Diego, USA, February 6–11, 2011.

[6] R. G. Gallager, *Information Theory and Reliable Communication*. New York: John Wiley & Sons, 1968.

[7] S. Shamai (Shitz) and S. Verdú, "The empirical distribution of good codes," *IEEE Transactions on Information Theory*, vol. 43, no. 3, pp. 836–846, May 1997.

[8] C.-L. Wu, P.-N. Chen, Y. S. Han, and Y.-X. Zheng, "On the coding scheme for joint channel estimation and error correction over block fading channels," in *Proceedings IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Tokyo, Japan, September 13–16, 2009, pp. 1272–1276.

[9] M. Dohler, R. W. Heath Jr., A. Lozano, C. B. Papadias, and R. A. Valenzuela, "Is the PHY layer dead?" *IEEE Communications Magazine*, vol. 49, no. 4, pp. 159–165, April 2011.

[10] J. N. Laneman, "On the distribution of mutual information," in *Proceedings Information Theory and Applications Workshop (ITA), University of California*, San Diego, USA, February 6–10, 2006.

[11] D. Buckingham and M. C. Valenti, "The information-outage probability of finite-length codes over AWGN channels," in *Proceedings Annual Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, USA, March 19–21, 2008, pp. 390–395.

[12] S. Lin and D. J. Costello, Jr., *Error Control Coding*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2004.

[13] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam: North-Holland, 1977.

[14] P.-N. Chen, H.-Y. Lin, and S. M. Moser, "Weak flip codes and applications to optimal code design on the binary erasure channel," in *Proceedings Fiftieth Allerton Conference on Communication, Control and Computing*, Allerton House, Monticello, IL, USA, October 1–5, 2012.

[15] S. M. Moser, *Information Theory (Lecture Notes)*, version 1, fall semester 2011/2012, Information Theory Lab, Department of Electrical Engineering, National Chiao Tung University (NCTU), September 2011. [Online]. Available: http://moser.cm.nctu.edu.tw/scripts.html

[16] P.-N. Chen, H.-Y. Lin, and S. M. Moser, "Optimal ultra-small block-codes for binary discrete memoryless channels," 2013, to appear in *IEEE Transactions on Information Theory*. [Online]. Available: http://moser.cm.nctu.edu.tw/publications.html

[17] C. E. Shannon, R. G. Gallager, and E. R. Berlekamp, "Lower bounds to error probability for coding on discrete memoryless channels," *Information and Control*, pp. 522–552, May 1967, part II.

[18] P.-N. Chen, H.-Y. Lin, and S. M. Moser, "Equidistant codes meeting the Plotkin bound are not optimal on the binary symmetric channel," to be presented at *IEEE International Symposium on Information Theory (ISIT)*, Istanbul, Turkey, July 7–13, 2013. [Online]. Available: http://moser.cm.nctu.edu.tw/publications.html

[19] S. J. MacMullan and O. M. Collins, "A comparison of known codes, random codes, and the best codes," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 3009–3022, October 1998.

[20] Y. Polyanskiy, "Saddle point in the minimax converse for channel coding," 2013, to appear in *IEEE Transactions on Information Theory*.