Error Probability Analysis of Binary Asymmetric Channels

Final Report of NSC Project

"Finite Blocklength Capacity"

Date:	31 January 2012
Project-Number:	NSC 97–2221–E–009–003–MY3
Project Duration:	1 August 2008 – 31 October 2011
Funded by:	National Science Council, Taiwan

Author:	Stefan M. Moser
Co-Authors:	Po-Ning Chen, Hsuan-Yin Lin
Organization:	Information Theory Laboratory Department of Electrical and Computer Engineering National Chiao Tung University
Address:	Engineering Building IV, Office 727 1001 Daxue Rd. Hsinchu 30010, Taiwan
E-mail:	stefan.moser@ieee.org

Abstract

In his world-famous paper of 1948, Shannon defined *channel capacity* as the ultimate rate at which information can be transmitted over a communication channel with an error probability that will vanish if we allow the blocklength to get infinitely large. While this result is of tremendous theoretical importance, the reality of practical systems looks quite different: no communication system will tolerate an infinite delay caused by an extremely large blocklength, nor can it deal with the computational complexity of decoding such huge codewords. On the other hand, it is not necessary to have an error probability that is exactly zero either, a small, but finite value will suffice.

Therefore, the question arises what can be done in a practical scheme. In particular, what is the maximal rate at which information can be transmitted over a communication channel for a given fixed maximum blocklength (i.e., a fixed maximum delay) if we allow a certain maximal probability of error? In this project, we have started to study these questions.

Block-codes with very short blocklength over the most general binary channel, the *binary asymmetric channel (BAC)*, are investigated. It is shown that for only two possible messages, flip-flop codes are optimal, however, depending on the blocklength and the channel parameters, not necessarily the linear flip-flop code. Further it is shown that the optimal decoding rule is a threshold rule. Some fundamental dependencies of the best code on the channel are given.

Block-codes with a very small number of codewords are investigated for the two special binary memoryless channels, the *binary symmetric channel (BSC)* and the *Z-channel (ZC)*. The optimal (in the sense of minimum average error probability, using maximum likelihood decoding) code structure is derived for the cases of two, three, and four codewords and an arbitrary blocklength. It is shown that for two possible messages, on a BSC, the so-called *flip codes of type* t are optimal for any t, while on a ZC, the flip code of type 0 is optimal. For codes with three or four messages it is shown that the so-called *weak flip codes* of some given type are optimal where the type depends on the blocklength. For all cases an algorithm is presented that constructs an optimal code for blocklength n recursively from an optimal code of length $n-1$. In the situation of two and four messages, the optimal code is shown to be linear. For the ZC a recursive optimal code design is conjectured in the case of five possible messages.

The derivation of these optimal codes relies heavily on a new approach of constructing and analyzing the code-matrix not row-wise (codewords), but *column-wise*. Moreover, these results also prove that the minimum Hamming distance might be the wrong design criterion for optimal codes even for very symmetric channels like the BSC.

Keywords: Channel capacity, binary asymmetric channel (BAC), error probability, finite blocklengths, ML, optimal codes, Z-channel.

Contents

1 Introduction

The analytical study of optimal communication over a channel is very difficult even if we restrict ourselves to discrete memoryless channels (DMCs). Most known results are derived using the mathematical trick of considering some limits, in particular, usually it is assumed that the blocklength tends to infinity. The insights that have been achieved in this way are considerable, but there still remains the open question how far these asymptotic results can be applied to the practical scenario where the blocklength is strongly restricted.

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 need to have large blocklength.

For many practical applications exactly this latter constraint is rather unfortunate as often we cannot tolerate too much delay (e.g., inter-human communication, time-critical control and communication, etc.). Moreover, the system complexity usually will grow exponentially in the blocklength. So we see that having large blocklength might not be an option and we have to restrict the blocklength 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 increased interests in the theoretical understanding of finite-length coding, see for example [2], [3]. There are several possible approaches on how one can approach the problem of finite-length codes. In [3] the authors fix an acceptable error probability and a finite blocklength and then try to find bounds on the possible transmission rates. In another approach, one fixes the transmission rate and studies how the error probability depends on the blocklength (i.e., one basically studies error exponents, but for relatively small n) [2]. Both approaches are related to Shannon's ideas in the sense that they try to make fundamental statements of what is possible and what not. The exact manner in which these systems have to be built is ignored on purpose.

Our approach in this work 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 an only very small fixed number of codewords: so called *ultra-small block-codes*. For such codes 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 design an optimal system.

For these reasons we have started to investigate the fundamental behavior of communication in the extreme case of an ultra-short blocklength. We would like to ask the following questions: What performance can we expect from codes of fixed, very short blocklength? What can we say about good design for such codes?

There are interesting applications for such codes. For example, 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 much limited to usually only a couple of bits. However, 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 [4]. 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 true error probability rather than an upper bound on the error probability derived from the union bound. To simplify our analysis, we have restricted ourselves for the moment to binary discrete memoryless channels.

For simplification of the exposition, in this paper we will exclusively focus on two special cases: the *binary symmetric channel (BSC)* and the *Z-channel (ZC)*. For results on general binary channels we refer to [5]. Note that while particularly for the BSC much is known about linear code design [6], there is basically no literature about *optimal*, possibly *nonlinear* codes.

The remainder of this report is structured as follows: after some comments about our notation we will introduce the channel models in Section 3. In Section 5 we will give some code definitions that will be used for the main results that are summarized in Section 6. Some of the proofs are omitted for space reasons. We refer to [5] for more details. Finally, Section 7 contains a discussion about the optimal code structure for the BSC.

As it 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 huge 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 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.

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

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

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 1. We say a DMC is used *without feedback*, if

$$P(x_k|x_1,\ldots,x_{k-1},y_1,\ldots,y_{k-1}) = P(x_k|x_1,\ldots,x_{k-1}) \quad \forall\, k, \qquad (2)$$

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. *If a DMC is used without feedback, then*

$$P(y_1,\ldots,y_n|x_1,\ldots,x_n) = \prod_{k=1}^{n} P_{Y|X}(y_k|x_k) \quad \forall\, n \geq 1. \qquad (3)$$

Proof. See, e.g., [7]. □

2.2 Coding for DMC

Definition 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 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 5. The *rate*[1] of a (M, n) code is defined as

$$\mathsf{R} \triangleq \frac{\log_2 \mathsf{M}}{n} \text{ bits/transmission.} \tag{4}$$

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!

Definition 6. An (M, n) coding scheme for a BAC consists of a codebook $\mathscr{C}^{(\mathsf{M},n)}$ with M binary codewords $\mathbf{x}_m$ of length n, 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.

[1] 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 7. Given that message m has been sent, let $\lambda_m^{(n)}$ be the *probability of a decoding error* of an (M, n) coding scheme with blocklength n:

$$\lambda_m^{(n)} \triangleq \Pr[g(\mathbf{Y}) \neq m \mid \mathbf{X} = \mathbf{x}_m] \tag{5}$$

$$= \sum_{\mathbf{y}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m)\ \mathrm{I}\{g(\mathbf{y}) \neq m\}, \tag{6}$$

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

$$\mathrm{I}\{\text{statement}\} \triangleq \begin{cases} 1 & \text{if statement is true,} \\ 0 & \text{if statement is wrong.} \end{cases} \tag{7}$$

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

$$\lambda^{(n)} \triangleq \max_{m \in \mathcal{M}} \lambda_m. \tag{8}$$

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

$$P_{\mathrm{e}}^{(n)} \triangleq \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \lambda_m^{(n)}. \tag{9}$$

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

$$\psi_m^{(n)} \triangleq \Pr[g(\mathbf{Y}) = m \mid \mathbf{X} = \mathbf{x}_m] \tag{10}$$

$$= \sum_{\mathbf{y}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m) \mathrm{I}\{g(\mathbf{y}) = m\}. \tag{11}$$

The definition of maximum success probability $\psi^{(n)}$ and the average success probability[2] $P_{\mathrm{c}}^{(n)}$ are accordingly.

Definition 8. For a given codebook $\mathscr{C}$, we define the *decoding region* $\mathcal{D}_m$ corresponding to the m-th codeword $\mathbf{x}_m$ as follows:

$$\mathcal{D}_m \triangleq \{\mathbf{y} \colon g(\mathbf{y}) = m\}. \tag{12}$$

Note that we will always assume that the decoder g is a *maximum likelihood (ML) decoder*:

$$g(\mathbf{y}) \triangleq \arg \max_{1 \leq m \leq \mathsf{M}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m) \tag{13}$$

that minimizes the average error probability $P_{\mathrm{e}}^{(n)}$.

Note that we write the codebook $\mathscr{C}^{(\mathsf{M},n)}$ as an $\mathsf{M} \times n$ matrix with the M rows corresponding to the M codewords:

$$\mathscr{C}^{(\mathsf{M},n)} = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{\mathsf{M}} \end{pmatrix}. \tag{14}$$

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 completely equivalent to the first in

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

the sense that it has the exact same error probability. Similarly, 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. Therefore, in the remainder of this paper, we will always consider such equivalent codes as being the same. In particular, when we speak of *unique design* we do not exclude the always possible permutations of columns and rows.

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

Theorem 9 (The Channel Coding Theorem for a DMC). *Define*

$$\mathsf{C} \triangleq \max_{P_X(\cdot)} I(X;Y) \tag{15}$$

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 $\lambda^{(n)} \to 0$ as the blocklength n gets very large.

Conversely, any sequence of $(2^{n\mathsf{R}}, n)$ coding schemes with maximum error probability $\lambda^{(n)} \to 0$ 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?

3 Channel Models

In the following we will concentrate on the special cases of *binary* DMCs, i.e., we restrict our channel alphabets to be binary.

The most general binary discrete memoryless channel is the so-called *binary asymmetric channel (BAC)*. It has a probability ϵ_0 that an input 0 will be flipped into a 1 and a (possible different) probability ϵ_1 for a flip from 1 to 0. See Figure 1.

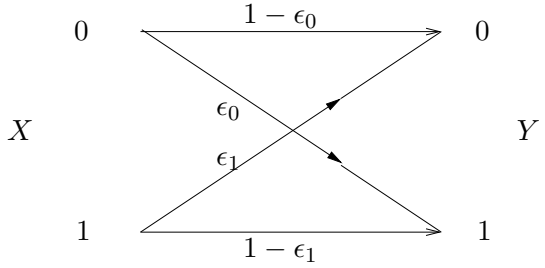

Figure 1: Binary asymmetric channel (BAC).

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

$$0 \leq \epsilon_0 \leq \epsilon_1 \leq 1, \tag{16}$$

$$\epsilon_0 \leq 1 - \epsilon_0, \tag{17}$$

$$\epsilon_0 \leq 1 - \epsilon_1. \tag{18}$$

Note that in the case where $\epsilon_0 > \epsilon_1$ we simply can flip all zeros to ones and vice-versa to get an equivalent channel with $\epsilon_0 \leq \epsilon_1$. For the case where $\epsilon_0 > 1 - \epsilon_0$, we can 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 (17) can be simplified to $\epsilon_0 \leq \frac{1}{2}$. And for the case where $\epsilon_0 > 1 - \epsilon_1$, we can 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 Figure 2. The region of interesting choices given by (16) and (17) is denoted by Ω.

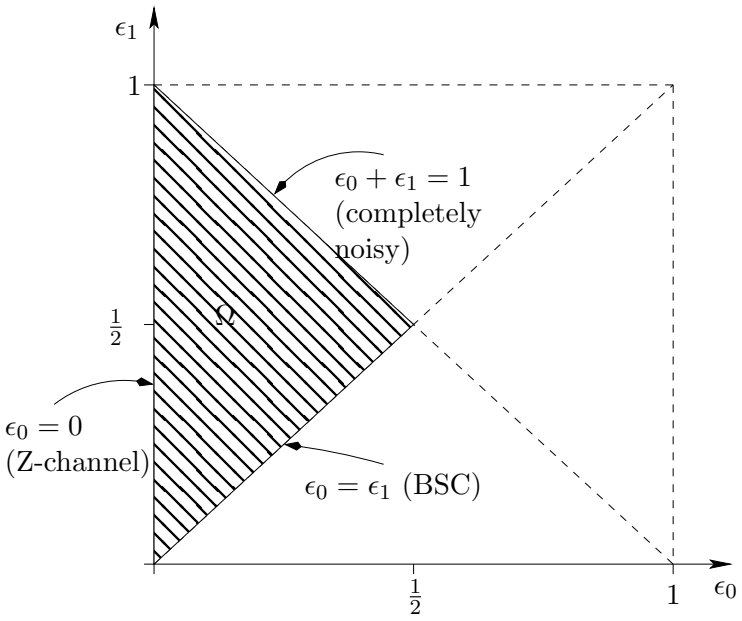

Figure 2: Region of possible choices of the channel parameters ϵ_0 and ϵ_1 of a BAC. The shaded area corresponds to the interesting area according to (16), (17) and (18).

Note that the BAC includes all well-known binary channel models: if $\epsilon_0 = \epsilon_1$, we have a BSC; and if $\epsilon_0 = 0$, we have a Z-channel. In the case when $\epsilon_0 = 1 - \epsilon_1$ we end up with a completely noisy channel of zero capacity: given $Y = y$, $X = 0$ and $X = 1$ are equally likely, i.e., $X \perp\!\!\!\perp Y$.

In this report we will put special emphasis on the former two special cases of the BAC. The *binary symmetric channel (BSC)* has equal cross-over probability $\epsilon_0 = \epsilon_1 = \epsilon$, see Fig. 3. For symmetry reasons and without loss of generality, we assume that $\epsilon < \frac{1}{2}$.

The *Z-channel (ZC)* will never distort an input 0, i.e., $\epsilon_0 = 0$. But the input 1 is flipped to 0 with probability $\epsilon_1 < 1$, see Fig. 4.

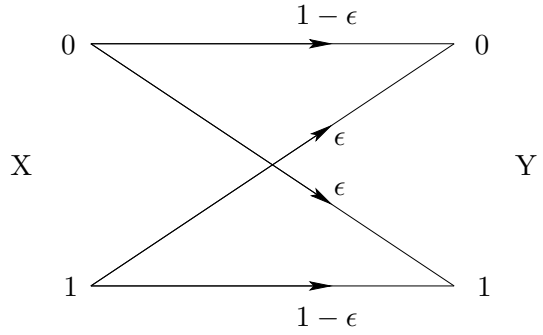

Figure 3: The binary symmetric channel (BSC).

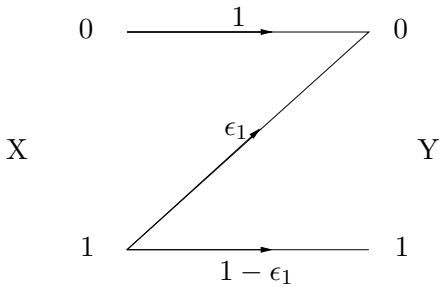

Figure 4: The Z-channel (ZC).

4 Preliminaries

4.1 Capacity of the BAC

As mentioned above, without loss of generality, we only consider BACs with $0 \leq \epsilon_0 \leq \epsilon_1 \leq 1$. The capacity of a BAC is given by

$$\mathrm{C_{BAC}} = \frac{\epsilon_0}{1-\epsilon_0-\epsilon_1} \cdot H_{\mathrm{b}}(\epsilon_1) - \frac{1-\epsilon_1}{1-\epsilon_0-\epsilon_1} \cdot H_{\mathrm{b}}(\epsilon_0) + \log_2\left(1 + 2^{\frac{H_{\mathrm{b}}(\epsilon_0)-H_{\mathrm{b}}(\epsilon_1)}{1-\epsilon_0-\epsilon_1}}\right) \quad (19)$$

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

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

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

$$P_X^*(0) = 1 - P_X^*(1) = \frac{1-\epsilon_1(1+z)}{(1-\epsilon_0-\epsilon_1)(1+z)} \quad (20)$$

with

$$z \triangleq 2^{\frac{H_{\mathrm{b}}(\epsilon_0)-H_{\mathrm{b}}(\epsilon_1)}{1-\epsilon_0-\epsilon_1}}. \quad (21)$$

4.2 Error Probability of the BAC

To simplify our notation we introduce $d_{\alpha\beta}(\mathbf{x}_m, \mathbf{y})$ to be the number of positions j where $x_m^{(j)} = \alpha$ and $y^{(j)} = \beta$, where as usual $\mathbf{x}_m$, $i \in \{1, 2, \ldots, \mathsf{M}\}$, is the sent codeword and $\mathbf{y}$ is the received sequence.

The conditional probability of the received vector given the sent codeword can now be written as

$$P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m) = (1-\epsilon_0)^{d_{00}(\mathbf{x}_m,\mathbf{y})} \cdot \epsilon_0^{d_{01}(\mathbf{x}_m,\mathbf{y})} \cdot \epsilon_1^{d_{10}(\mathbf{x}_m,\mathbf{y})} \cdot (1-\epsilon_1)^{d_{11}(\mathbf{x}_m,\mathbf{y})}. \quad (22)$$

Note that we can express these different $d_{\alpha\beta}$'s as follows:

$$d_{11}(\mathbf{x}_m, \mathbf{y}) = \frac{1}{2} w_{\mathrm{H}}(\mathbf{x}_m + \mathbf{y} - |\mathbf{x}_m - \mathbf{y}|), \tag{23}$$

$$d_{10}(\mathbf{x}_m, \mathbf{y}) = w_{\mathrm{H}}(\mathrm{I}\{\mathbf{x}_m - \mathbf{y} > 0\}), \tag{24}$$

$$d_{01}(\mathbf{x}_m, \mathbf{y}) = w_{\mathrm{H}}(\mathrm{I}\{\mathbf{y} - \mathbf{x}_m > 0\}), \tag{25}$$

$$d_{00}(\mathbf{x}_m, \mathbf{y}) = n - d_{11}(\mathbf{x}_m, \mathbf{y}) - d_{10}(\mathbf{x}_m, \mathbf{y}) - d_{01}(\mathbf{x}_m, \mathbf{y}), \tag{26}$$

where $w_{\mathrm{H}}(\mathbf{x})$ is the Hamming weight of $\mathbf{x}$.

The average error probability of a code $\mathscr{C}$ over a BAC (assuming equally likely messages) can be expressed as

$$P_{\mathrm{e}}^{(n)}(\mathscr{C}) = \frac{(1-\epsilon_0)^n}{\mathsf{M}} \sum_{\mathbf{y}} \sum_{\substack{i=1 \\ i \neq g(\mathbf{y})}}^{\mathsf{M}} \left(\frac{\epsilon_0}{1-\epsilon_0}\right)^{d_{01}(\mathbf{x}_m, \mathbf{y})} \left(\frac{\epsilon_1}{1-\epsilon_0}\right)^{d_{10}(\mathbf{x}_m, \mathbf{y})} \left(\frac{1-\epsilon_1}{1-\epsilon_0}\right)^{d_{11}(\mathbf{x}_m, \mathbf{y})} \tag{27}$$

$$= \frac{1}{\mathsf{M}} \sum_{i=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y}) \neq i}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m), \tag{28}$$

where $g(\mathbf{y})$ is the ML decision (13) for the observation $\mathbf{y}$.

Note that a closer investigation shows that some of these optimal codes are linear, but some are not.

4.3 Error Probability of the BSC

Consider the situation of a BSC and assume that we transmit the m-th codeword $\mathbf{x}_m$, $1 \leq m \leq \mathsf{M}$, and that we receive $\mathbf{y}$. The *maximum likelihood (ML)* decision is then

$$g(\mathbf{y}) \triangleq \arg \max_{1 \leq i \leq \mathsf{M}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m). \tag{29}$$

The average probability of error can be computed as

$$P_{\mathrm{e}}^{(n)} = \frac{1}{\mathsf{M}} (1-\epsilon)^n \sum_{\mathbf{y}} \sum_{\substack{i=1 \\ i \neq g(\mathbf{y})}}^{\mathsf{M}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_{\mathrm{H}}(\mathbf{x}_m, \mathbf{y})} \tag{30}$$

where $d_{\mathrm{H}}(\cdot, \cdot)$ is the Hamming distance.

Note that if we want to find the *best* average error probability, we need to check through **all possible** codes (including both linear and nonlinear codes). The complexity of such a search grows exponentially fast in n: for $\mathsf{M} = 4$ and

- for $n = 3$ there are $\binom{8}{4} = 70$ different codes;
- for $n = 4$ there are $\binom{16}{4} = 1820$ different codes;
- for $n = 5$ there are $\binom{32}{4} = 35960$ different codes, etc.

It turns out that for a given BSC, blocklength n, and number of message M, there is a vast amount of different codes (linear and nonlinear) that are all optimal. This is not really surprising because the BSC is *strongly symmetric*.

4.4 Error (and Success) Probability of the Z-Channel

A special case of the BAC is the Z-channel where we have $\epsilon_0 = 0$. By symmetry, assume that $\epsilon_1 \leq \frac{1}{2}$. Note that it is often easier to maximize the success probability instead of minimizing the error probability. For the convenience of later derivations, we now are going to derive its error and success probabilities:

$$P_{\mathrm{c}}^{(n)}(\mathscr{C}) = \frac{1}{\mathsf{M}} \sum_{i=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})=i}} (1-\epsilon_1)^{w_{\mathrm{H}}(\mathbf{x}_m)} \left(\frac{\epsilon_1}{1-\epsilon_1}\right)^{d_{10}(\mathbf{x}_m,\mathbf{y})} \cdot \mathrm{I}\left\{\text{if } x_m^{(j)} = 0 \implies y^{(j)} = 0, \ \forall\, j\right\}. \tag{31}$$

The error probability formula is accordingly

$$P_{\mathrm{e}}^{(n)}(\mathscr{C}) = \frac{1}{\mathsf{M}} \sum_{i=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})\neq i}} (1-\epsilon_1)^{w_{\mathrm{H}}(\mathbf{x}_m)} \left(\frac{\epsilon_1}{1-\epsilon_1}\right)^{d_{10}(\mathbf{x}_m,\mathbf{y})} \cdot \mathrm{I}\left\{\text{if } x_m^{(j)} = 0 \implies y^{(j)} = 0, \ \forall\, j\right\}. \tag{32}$$

Note that the capacity-achieving distribution for $\epsilon_1 = \frac{1}{2}$ is

$$\Pr[X = 1] = \frac{2}{5}. \tag{33}$$

4.5 Pairwise Hamming Distance

The minimum Hamming distance is a well-known and often used quality criterion of a codebook. However, for the description of an optimal code design for a fixed blocklength n, it turns out to be too crude. We define a slightly more general and more concise description of a codebook: the *pairwise Hamming distance vector*.

Definition 10. Given a codebook $\mathscr{C}^{(\mathsf{M},n)}$ with codewords $\mathbf{x}_m$ we define the *pairwise Hamming distance vector* $\mathbf{d}^{(\mathsf{M},n)}$ of length $\frac{(\mathsf{M}-1)\mathsf{M}}{2}$ as

$$\mathbf{d}^{(\mathsf{M},n)} \triangleq \Big(d_{12}^{(n)},\ d_{13}^{(n)}, d_{23}^{(n)},\ d_{14}^{(n)}, d_{24}^{(n)}, d_{34}^{(n)}, \ldots, d_{1\mathsf{M}}^{(n)}, d_{2\mathsf{M}}^{(n)}, \ldots, d_{(\mathsf{M}-1)\mathsf{M}}^{(n)}\Big) \tag{34}$$

with $d_{ij}^{(n)} \triangleq d_{\mathrm{H}}(\mathbf{x}_i, \mathbf{x}_j)$, $1 \leq i < j \leq \mathsf{M}$. The *minimum Hamming distance* $d_{\min}^{(\mathsf{M},n)}$ is defined as the minimum component of the vector $\mathbf{d}^{(\mathsf{M},n)}$.

Note that we have seen in Section 4.3 that the error probability of a binary code that is used over a BSC can be described using the Hamming distance between the received vectors $\mathbf{y}$ and the different codewords $\mathbf{x}_m$. We introduce the following notation for this type of Hamming distance.

Definition 11. For a given codebook $\mathscr{C}^{(\mathsf{M},n)}$ and for some received n-vector $\mathbf{y}$, the *received Hamming distance vector* $\mathbf{d}^{(n)}(\mathbf{y})$ is defined as

$$\mathbf{d}^{(n)}(\mathbf{y}) = \big(d_1^{(n)}(\mathbf{y}), \ldots, d_{\mathsf{M}}^{(n)}(\mathbf{y})\big) \triangleq \big(d_{\mathrm{H}}(\mathbf{y}, \mathbf{x}_1), \ldots, d_{\mathrm{H}}(\mathbf{y}, \mathbf{x}_{\mathsf{M}})\big), \tag{35}$$

where $d_m^{(n)}(\mathbf{y}) \triangleq d_{\mathrm{H}}(\mathbf{y}, \mathbf{x}_m)$ denotes its mth component and is called *received Hamming distance*.

Note that an ML decoder will always decode a received vector $\mathbf{y}$ to that message that results into a minimum value of the received Hamming distance:

$$d_{\min}^{(n)}(\mathbf{y}) \triangleq \min_{m=1,\ldots,\mathsf{M}} d_m^{(n)}(\mathbf{y}). \tag{36}$$

5 Flip Codes and Weak Flip Codes

Next, we will introduce some special codebooks that will be used later on.

Definition 12. 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 $\mathscr{C}_t^{(2,n)}$:

$$\mathscr{C}_t^{(2,n)} \triangleq \begin{pmatrix} \mathbf{x} \\ \bar{\mathbf{x}} \end{pmatrix} = \begin{pmatrix} 0 & \cdots & 0 & \overbrace{1 & \cdots & 1}^{t \text{ columns}} \\ 1 & \cdots & 1 & 0 & \cdots & 0 \end{pmatrix}. \tag{37}$$

Defining the column vectors

$$\left\{ \mathbf{c}_1^{(2)} \triangleq \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \ \mathbf{c}_2^{(2)} \triangleq \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}, \tag{38}$$

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 BSC and the ZC, the order of the columns of any codebook matrix is irrelevant. 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.

Definition 13. A *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

$$\left\{ \mathbf{c}_1^{(3)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \ \mathbf{c}_2^{(3)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \ \mathbf{c}_3^{(3)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\} \tag{39}$$

or

$$\left\{ \mathbf{c}_1^{(4)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \ \mathbf{c}_2^{(4)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \ \mathbf{c}_3^{(4)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \right\}, \tag{40}$$

respectively.[3] We often describe a weak flip code of type (t_2, t_3) by the *code parameters*

$$[t_1, t_2, t_3] \tag{41}$$

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

Lemma 14. *The pairwise Hamming distance vector of a weak flip code can be computed as follows:*

$$\mathbf{d}^{(3,n)} = (t_2 + t_3, t_1 + t_3, t_1 + t_2), \tag{42}$$

$$\mathbf{d}^{(4,n)} = (t_2 + t_3, t_1 + t_3, t_1 + t_2, t_1 + t_2, t_1 + t_3, t_2 + t_3). \tag{43}$$

[3] The name *weak flip code* is motivated by the fact that the weak flip code is a generalization of the flip code: while for $\mathsf{M} = 3$ it is not possible to have all codewords to be flipped versions of other codewords and for $\mathsf{M} = 4$ such a definition would be too restrictive, it is still true that the distribution of zeros and ones in the candidate columns $\mathbf{c}_1$, $\mathbf{c}_2$, and $\mathbf{c}_3$ is very balanced.

6 Main Results

6.1 An Example

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 small example before we summarize our main results.

Example 15. Assume a BSC with cross probability $\epsilon = 0.4$, $\mathsf{M} = 4$, and a blocklength $n = 4$. Then consider the following two weak flip codes:

$$\mathcal{C}_{1,0}^{(4,4)} \triangleq \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \quad \mathcal{C}_{2,0}^{(4,4)} \triangleq \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}. \tag{44}$$

We observe that while both codes are linear, the first code has a minimum Hamming distance 1, and the second has 2. Assuming an ML decoder, the average error probability can be expressed using the Hamming distance between the received sequence and the codewords:

$$P_{\mathrm{e}}^{(n)}(\mathcal{C}) = \frac{1}{\mathsf{M}} \sum_{m=1}^{4} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y}) \neq m}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m) \tag{45}$$

$$= \frac{(1-\epsilon)^4}{4} \sum_{m=1}^{4} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y}) \neq m}} \left(\frac{\epsilon}{1-\epsilon} \right)^{d_{\mathrm{H}}(\mathbf{x}_m, \mathbf{y})}, \tag{46}$$

where $d_H(\mathbf{x}_m, \mathbf{y})$ is the Hamming distance between a codeword $\mathbf{x}_m$ and a received vector $\mathbf{y}$.

If evaluated, we get an error probability $P_{\mathrm{e}}^{(n)} = 0.6112$ for $\mathcal{C}_{1,0}^{(4,4)}$ and 0.64 for $\mathcal{C}_{2,0}^{(4,4)}$. 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

$$P_{\mathrm{e}}^{(n)}\big(\mathscr{C}^{(\mathsf{M},n)*}\big) \leq P_{\mathrm{e}}^{(n)}\big(\mathscr{C}^{(\mathsf{M},n)}\big), \tag{47}$$

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

6.2 Optimal Codes on BAC for $\mathsf{M} = 2$

Due to the memorylessness of the BAC, the order of the columns of any code is irrelevant. We therefore can restrict ourselves without loss of generality to flip-flop codes of type t to describe all possible flip-flop codes. Also note that the only possible linear flip-flop code is $\mathscr{C}_0^{(2,n)}$. All other flip-flop codes are nonlinear.

We are now ready for the following result.

Proposition 16. *Consider the case $\mathsf{M} = 2$, and fix the blocklength n. Then, irrespective of the channel parameters ϵ_0 and ϵ_1, on a BAC there always exists a t, $0 \leq t \leq \lfloor \frac{n}{2} \rfloor$, such that the flip-flop code of type t, $\mathscr{C}_t^{(2,n)}$, is an optimal code in the sense that it minimizes the error probability.*

Proof. See Appendix A. □

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. However, the proposition does not exclude the possibility that such a code might exist that also is optimal.

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 Figure 5 for $n = 5$. We see that depending on the channel parameters, the optimal

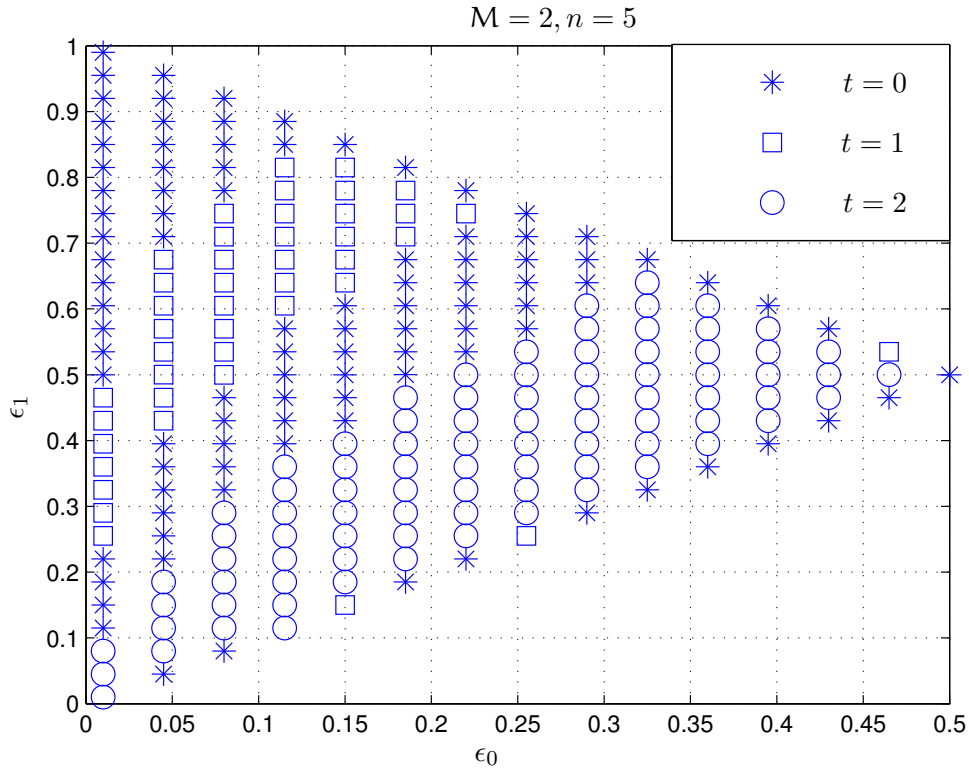

Figure 5: Optimal codebooks on a BAC: the optimal choice of the parameter t for different values of ϵ_0 and ϵ_1 for a fixed blocklength $n = 5$.

value of t changes. Note that on the boundaries the optimal choice is not unique: for a completely noisy BAC ($\epsilon_1 = 1 - \epsilon_0$), the choice of the codebook is irrelevant since the probability of error is $\frac{1}{2}$ in any case. For a BSC, $t = 0$, $t = 1$, or $t = 2$ are equivalent. And for a Z-channel we can prove that a linear code is always optimal.[4]

6.3 Optimal Decision Rule on BAC for M = 2

In any system with only two possible messages the optimal ML receiver can easily be described by the *log likelihood ratio (LLR)*:

$$\mathrm{LLR}(\mathbf{y}) \triangleq \log\left(\frac{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_1)}{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_2)}\right). \tag{48}$$

If $\mathrm{LLR}(\mathbf{y}) > 0$, then the receiver decides for 1, while if $\mathrm{LLR}(\mathbf{y}) < 0$, it decides for 2. In the situation of $\mathrm{LLR}(\mathbf{y}) = 0$, both decisions are equally good.

[4] As seen next in Section 6.4, a linear code is also optimal for the Z-channel for the case M = 4.

In the situation of a flip-flop code of type t, $\mathscr{C}_t^{(2,n)}$, the LLR is given as

$$\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d) \triangleq (t-d)\log\left(\frac{1-\epsilon_1}{\epsilon_0}\right)+(n-t-d)\log\left(\frac{1-\epsilon_0}{\epsilon_1}\right), \tag{49}$$

where d is defined to be the Hamming distance between the received vector and the ***first*** codeword:

$$d \triangleq d_{\mathrm{H}}(\mathbf{x}_1,\mathbf{y}). \tag{50}$$

Note that $0 \le d \le n$ depends on the received vector, while t and n are code parameters, and ϵ_0 and ϵ_1 are channel parameters.

Hence, the optimal decision rule can be expressed in terms of d.

Proposition 17. *We list some properties of* $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$*:*

1. *If* $\epsilon_0+\epsilon_1=1$*, then* $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)=0$ *for all* d*.*
2. $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$ *is a decreasing function in* d*:*
$$\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d) \ge \mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d+1), \quad \forall\, 0\le d\le n-1. \tag{51}$$
3. *For* $d\le t$ *and* $d>\left\lfloor\frac{n}{2}\right\rfloor$ *the* $\mathrm{LLR}_t^{(n)}$ *is always larger or smaller than zero, respectively:*
$$\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)\begin{cases}\ge 0 & \text{for } 0\le d\le t,\\ \lesseqgtr 0 & \text{for } t<d\le\left\lfloor\frac{n}{2}\right\rfloor, \text{ depending on } \epsilon_0,\ \epsilon_1,\\ \le 0 & \text{for } \left\lfloor\frac{n}{2}\right\rfloor<d\le n.\end{cases} \tag{52}$$
4. $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$ *is an increasing function in* n*, when we fix* d*,* ϵ_0*, and* ϵ_1*.*
5. $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$ *is an increasing function in* t *when we fix* n*,* d*,* ϵ_0*, and* ϵ_1*.*
6. *For* $0\le d\le n-1$*,*
$$\mathrm{LLR}_t^{(n+1)}(\epsilon_0,\epsilon_1,d+1) < \mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d). \tag{53}$$

Proof. Omitted. □

From these properties we immediately obtain an interesting result about the optimal decision rule.

Proposition 18 (Optimal Decision Rule has a Threshold). *For a fixed flip-flop code* $\mathscr{C}_t^{(2,n)}$ *and a fixed BAC* $(\epsilon_0,\epsilon_1)\in\Omega$*, there exists a* ***threshold*** ℓ*,* $t\le\ell\le\left\lfloor\frac{n-1}{2}\right\rfloor$*, such that the optimal ML decision rule can be stated as*

$$g(\mathbf{y})=\begin{cases}1 & \text{if } 0\le d\le\ell,\\ 2 & \text{if } \ell+1\le d\le n.\end{cases} \tag{54}$$

The threshold ℓ *depends on* (ϵ_0,ϵ_1)*, but similar channels will usually have the same threshold. We define the region of channel parameters with identical threshold as follows:*

$$\Omega_{\ell,t}^{(n)} \triangleq \left\{(\epsilon_0,\epsilon_1)\,\middle|\,\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,\ell)\ge 0\right\}\bigcap\left\{(\epsilon_0,\epsilon_1)\,\middle|\,\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,\ell+1)\le 0\right\}. \tag{55}$$

In Figure 6 an example of this threshold behavior is shown. For $\epsilon_0\in[0.136,0.270]$ we see that $\mathrm{LLR}_1^{(7)}(\epsilon_0,1-2\epsilon_0,d)\ge 0$ for $d=0$, $d=1$, and $d=2$, while $\mathrm{LLR}_1^{(7)}(\epsilon_0,1-2\epsilon_0,d)<0$ for $d\ge 3$. Hence, $\ell=2$.

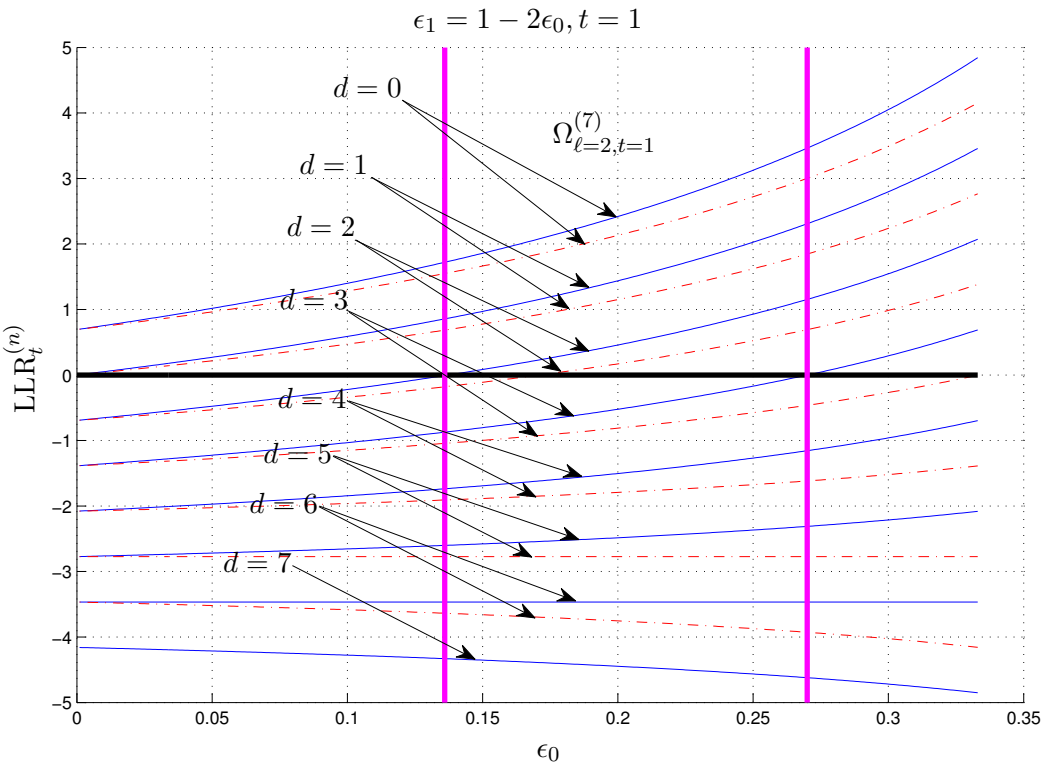

Figure 6: The log likelihood function depicted as a function of (ϵ_0, ϵ_1) for different values of d. To simplify the plot, only ϵ_0 is depicted with ϵ_1 being a fixed function of ϵ_0. The solid blue lines depict the case $n = 7$, the dashed red lines $n = 6$. The code is fixed to be $t = 1$. We see that for $n = 7$ and $\epsilon_0 \in [0.136, 0.270]$ the threshold is $\ell = 2$.

6.4 Optimal Codes on ZC

Theorem 19. *For a ZC and for any $n \geq 1$, an optimal codebook with two codewords $\mathsf{M} = 2$ is the flip codebook of type 0, $\mathscr{C}_0^{(2,n)}$. It has an error probability*

$$P_{\mathrm{e}}^{(n)}\big(\mathscr{C}_0^{(2,n)}\big) = \frac{1}{2}\epsilon_1^n. \tag{56}$$

Proof. Let the m-th codeword be $\mathbf{x}_m = (x_{m,1}, x_{m,2}, \ldots, x_{m,n})$, $m = 1, 2$, and let the received vector be $\mathbf{y} = (y_1, y_2, \ldots, y_n)$. Then the average error probability of a ZC can be expressed as

$$\begin{aligned} P_{\mathrm{e}}^{(n)}(\mathscr{C}) &= \frac{1}{2}\sum_{\mathbf{y}} \sum_{\substack{m=1 \\ m\neq g(\mathbf{y})}}^{2} (1-\epsilon_1)^{w_{\mathrm{H}}(\mathbf{x}_m)} \left(\frac{\epsilon_1}{1-\epsilon_1}\right)^{d_{10}(\mathbf{x}_m,\mathbf{y})} \\ &\qquad \cdot \mathrm{I}\left\{x_{m,j} = 0 \Longrightarrow y_j = 0,\ \forall\, j\right\} \qquad (57) \\ &= \frac{1}{2}\sum_{\mathbf{y}} \min\left\{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_1), P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_2)\right\}, \qquad (58) \end{aligned}$$

where $w_{\mathrm{H}}(\mathbf{x}_m)$ is the Hamming weight of the codeword $\mathbf{x}_m$ and $d_{10}(\mathbf{x}_m, \mathbf{y})$ denotes the number of positions j where $x_{m,j} = 1$ and $y_j = 0$.

Now note that Proposition 16 shows that an optimal code should have two codewords that are flipped to each other. The intuition behind this is that an optimal decoder will simply ignore all those bit positions where both codewords are identical, leading to the same performance that can be achieved for a code of shorter length. We therefore now assume that $\mathbf{x}_2 = \bar{\mathbf{x}}$ is the flipped version of $\mathbf{x}_1 = \mathbf{x}$, with $\mathbf{x}$ defined in (37).

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

$$\min\left\{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}), P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\bar{\mathbf{x}})\right\} = \begin{cases} 0 & \text{if } \mathbf{y} \neq \mathbf{0}, \\ \epsilon_1^{\max\{w_{\mathrm{H}}(\mathbf{x}), w_{\mathrm{H}}(\bar{\mathbf{x}})\}} & \text{if } \mathbf{y} = \mathbf{0}. \end{cases} \tag{59}$$

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

Lemma 20. *For a ZC and for any $n \geq 2$, the average success probabilities of the weak flip code of type $(t, 0)$, $1 \leq t \leq \lfloor \frac{n}{2} \rfloor$, with three codewords $\mathsf{M} = 3$ or four codewords $\mathsf{M} = 4$ are*

$$3P_{\mathrm{c}}^{(n)}\big(\mathscr{C}_{t,0}^{(3,n)}\big) = 1 + \sum_{d=0}^{t-1} \binom{t}{d} (1-\epsilon_1)^{t-d}\, \epsilon_1^d + \sum_{d=0}^{(n-t)-1} \binom{n-t}{d} (1-\epsilon_1)^{(n-t)-d}\, \epsilon_1^d; \tag{60}$$

$$\begin{aligned} 4P_{\mathrm{c}}^{(n)}\big(\mathcal{C}_{t,0}^{(4,n)}\big) &= 1 + \sum_{d=0}^{t-1} \binom{t}{d} (1-\epsilon_1)^{t-d}\, \epsilon_1^d + \sum_{d=0}^{n-t-1} \binom{n-t}{d} (1-\epsilon_1)^{(n-t)-d}\, \epsilon_1^d \\ &\quad + \sum_{d=0}^{n-1} \left[\binom{n}{d} - \binom{n-t}{d-t} - \binom{t}{d-(n-t)}\right] (1-\epsilon_1)^{n-d}\, \epsilon_1^d. \qquad (61) \end{aligned}$$

Moreover, these average success probabilities are increasing with t.

Proof. Note that the average success probability of a code with M messages used over a ZC can be written as follows:

$$P_{\mathrm{c}}^{(n)}(\mathscr{C}) = \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})=m}} (1-\epsilon_1)^{w_{\mathrm{H}}(\mathbf{x}_m)} \left(\frac{\epsilon_1}{1-\epsilon_1}\right)^{d_{10}(\mathbf{x}_m,\mathbf{y})} \cdot \mathrm{I}\left\{x_{m,j} = 0 \Longrightarrow y_j = 0,\ \forall\, j\right\}. \tag{62}$$

We will use this together with the peculiar behavior of the ZC, which ensures that $P_{Y|X}(1|0) = 0$, to derive (60) and (61).

We start with $\mathsf{M} = 4$. Consider the weak flip code of type $(t, 0)$, $\mathscr{C}_{t,0}^{(4,n)}$, where $1 \le t \le \lfloor \frac{n}{2} \rfloor$. Denote the success probability of the m-th codeword by $\psi_{t,m}^{(4,n)}$ and the corresponding decoding region by $\mathcal{D}_{t,m}^{(4,n)}$. Also, in the following we will use a superscript to emphasize the length of a vector, e.g., $\mathbf{y}^{(t)}$ is a vector of length t.

Recall that the first codeword of $\mathscr{C}_{t,0}^{(4,n)}$ is the all-zero codeword, the second codeword has Hamming weight t, and the remaining two are flipped version of the first two. From $P_{Y|X}(0|0) = 1$, the first codeword will always be transmitted to $\mathbf{0}^{(n)}$, i.e., $\mathcal{D}_{t,1}^{(4,n)}$ only consists of the all-zero vector.

Next note that for any $\mathbf{y}$ with $\mathbf{y}^{(n)} = [\mathbf{0}^{(n-t)}\mathbf{y}^{(t)}]$ and where $w_{\mathrm{H}}(\mathbf{y}^{(t)}) \ge 1$ we have

$$\begin{aligned}
&\max\left\{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_1), P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_2), P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_3), P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_4)\right\} \\
&\quad = \max\left\{0, P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_2), 0, P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_4)\right\} && (63)\\
&\quad = P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_2) && (64)\\
&\quad = (1-\epsilon_1)^{t-d} \cdot \epsilon_1^d, && (65)
\end{aligned}$$

where $d \le t-1$ denotes the Hamming distance between $\mathbf{y}$ and $\mathbf{x}_2$. The last step follow because we assume that $0 < \epsilon_1 \le \frac{1}{2}$ and because $w_{\mathrm{H}}(\mathbf{x}_4) = n > t = w_{\mathrm{H}}(\mathbf{x}_2)$.

The same argument can be applied to the decoding region $\mathcal{D}_{t,3}^{(4,n)}$, and $\mathcal{D}_{t,4}^{(4,n)}$ must be $\{0,1\}^n \setminus \bigcup_{i=1}^{3} \mathcal{D}_{t,i}^{(4,n)}$. Hence we get the following list:

$$\begin{aligned}
\mathcal{D}_{t,1}^{(4,n)} &= \left\{\mathbf{0}^{(n)}\right\}, && (66)\\
\mathcal{D}_{t,2}^{(4,n)} &= \left\{\mathbf{y}^{(n)} \colon \mathbf{y}^{(n)} = [\mathbf{0}^{(n-t)}\mathbf{y}^{(t)}] \right. \\
&\qquad \left. \text{with } 1 \le w_{\mathrm{H}}(\mathbf{y}^{(t)}) \le t\right\}, && (67)\\
\mathcal{D}_{t,3}^{(4,n)} &= \left\{\mathbf{y}^{(n)} \colon \mathbf{y}^{(n)} = [\mathbf{y}^{(n-t)}\mathbf{0}^{(t)}] \right. \\
&\qquad \left. \text{with } 1 \le w_{\mathrm{H}}(\mathbf{y}^{(n-t)}) \le n-t\right\}, && (68)\\
\mathcal{D}_{t,4}^{(4,n)} &= \{0,1\}^n \setminus \bigcup_{m=1}^{3} \mathcal{D}_{t,m}^{(4,n)}. && (69)
\end{aligned}$$

Using this in (62) will then lead to (61). Similarly, we also can show that the success probability of $\mathscr{C}_{t,0}^{(3,n)}$ is as given in (60). □

Theorem 21. *For a ZC and for any $n \ge 2$, an optimal codebook with three codewords $\mathsf{M} = 3$ or four codewords $\mathsf{M} = 4$ is the weak flip code of type $(t^*, 0)$ with $t^* \triangleq \lfloor \frac{n}{2} \rfloor$:*

$$\mathscr{C}_{\mathrm{ZC}}^{(\mathsf{M},n)*} = \mathscr{C}_{t^*,0}^{(\mathsf{M},n)}. \tag{70}$$

Proof. Our proof is based on induction on n. The optimal code for $\mathsf{M} = 4$ and $n = 2$ is trivial since there are only four possible different codewords. The optimal code is

$$\mathscr{C}_{1,0}^{(4,2)} = \left(\mathbf{c}_1^{(4)}, \mathbf{c}_2^{(4)}\right). \tag{71}$$

Next assume that for blocklength n, $\mathscr{C}_{\lfloor \frac{n}{2} \rfloor,0}^{(4,n)}$ is optimal. Then, from Claim 22 below, this assumption still holds for blocklength $(n+1)$. This will prove the theorem.

Claim 22. *Let's append one new column to the weak flip code of type* $(\lfloor \frac{n}{2} \rfloor, 0)$, $\mathscr{C}_{\lfloor \frac{n}{2} \rfloor,0}^{(4,n)}$, *to generate a new code of length* $(n+1)$. *The optimal (in the sense of resulting in the biggest success probability) choice among all possible* $2^{\mathsf{M}} = 16$ *columns is* $\mathbf{c}_2^{(4)}$.

In fact, this claim holds not only for $t = \lfloor \frac{n}{2} \rfloor$ but for any $t \in \{1, 2 \ldots, \lfloor \frac{n}{2} \rfloor\}$. Note that there are actually only 14 possible columns that we could choose as the $(n+1)$-th column because the all-zero and the all-one columns clearly are suboptimal as in this case an optimal decoder will simply ignore the $(n+1)$-th received digit.

To prove Claim 22 we append an additional bit to all four codewords as follows:

$$\begin{pmatrix} [\mathbf{0}\ x_{1,n+1}] \\ [\mathbf{x}\ x_{2,n+1}] \\ [\bar{\mathbf{x}}\ x_{3,n+1}] \\ [\mathbf{1}\ x_{4,n+1}] \end{pmatrix} \tag{72}$$

where $x_{m,n+1} \in \{0, 1\}$ and the $\mathbf{x}$ is given in (37) with $t \triangleq \lfloor \frac{n}{2} \rfloor$. Note that in the remainder of this proof t can be read as shorthand for $\lfloor \frac{n}{2} \rfloor$.

We now extend the decoding regions given in (66)–(69) by one bit for $m = 1, 2, 3, 4$: $[\mathcal{D}_{t,m}^{(4,n)}\ 0] \cup [\mathcal{D}_{t,m}^{(4,n)}\ 1]$. Observe that these new decoding regions retain the same success probability $\psi_{t,m}^{(4,n+1)} = \psi_{t,m}^{(4,n)} \cdot 1$, because

$$P_{Y|X}(0|x_{m,n+1}) + P_{Y|X}(1|x_{m,n+1}) = 1. \tag{73}$$

However, it is quite clear that these new regions are in general not the optimal decision regions anymore for the new code. So the question is how to fix them to make them optimal again (and thereby also finding out 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 $\mathbf{y}^{(n+1)} \in [\mathcal{D}_{t,m}^{(4,n)}\ 0]$ still will optimally be decoded to m. To see this, let's have a look at the four cases separately:

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

$$\begin{aligned} P_{\mathbf{Y}|\mathbf{X}}\big(\mathbf{0}^{(n+1)} \,\big|\, \mathbf{x}_1^{(n+1)} = [\mathbf{0}^{(n)} 1]\big) &= \epsilon_1 \\ &\geq P_{\mathbf{Y}|\mathbf{X}}\big(\mathbf{0}^{(n+1)} \,\big|\, \mathbf{x}_j^{(n+1)}\big), \quad \forall\, j = 2, 3, 4, \end{aligned} \tag{74}$$

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

- $[\mathcal{D}_{t,2}^{(4,n)}\ 0]$: All vectors in $[\mathcal{D}_{t,2}^{(4,n)}\ 0]$ 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 $m = 2$.

- $[\mathcal{D}_{t,3}^{(4,n)}\ 0]$: All vectors in $[\mathcal{D}_{t,3}^{(4,n)}\ 0]$ 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 $m = 3$.

- $[\mathcal{D}_{t,4}^{(4,n)}\ 0]$: All vectors in $[\mathcal{D}_{t,4}^{(4,n)}\ 0]$ 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 $[\mathcal{D}_{t,m}^{(4,n)}\ 1]$ if $x_{m,n+1} = 0$. Note that we do not need to bother about $[\mathcal{D}_{t,4}^{(4,n)}\ 1]$ as it is impossible to receive such a vector because for all $\mathbf{y} \in \mathcal{D}_{t,4}^{(4,n)}$,

$$P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}^{(n)}|\mathbf{0}^{(n)}) = P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}^{(n)}|\mathbf{0}^{(n-t)}\mathbf{1}^{(t)}) = P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}^{(n)}|\mathbf{1}^{(n-t)}\mathbf{0}^{(t)}) = 0. \tag{75}$$

For $m = 1$, 2, or 3, if $x_{m,n+1} = 0$, the received vectors in $[\mathcal{D}_{t,m}^{(4,n)}\ 1]$ will change to another decoding region not equal to m because $P_{Y|X}(1|0) = 0$.

- $[\mathcal{D}_{t,1}^{(4,n)}\ 1]$: If we assign these vectors (actually, it's only one) to the new decoding region $\mathcal{D}_{t,2}^{(4,n+1)}$, the amount of newly added conditional success probability for $m = 2$ is given by

$$\begin{aligned}
\Delta\psi_2 &\triangleq \psi_{t,2}^{(4,n+1)} - \psi_{t,2}^{(4,n)} && (76)\\
&= \sum_{\mathbf{y}^{(n)} \in \mathcal{D}_{t,1}^{(4,n)}} P_{\mathbf{Y}|\mathbf{X}}\big([\mathbf{y}^{(n)}\ 1]\,\big|\,[\mathbf{0}^{(n-t)}\mathbf{1}^t\ 1]\big) \\
&\qquad \cdot (x_{2,n+1} - x_{1,n+1})^+ && (77)\\
&= \epsilon_1^t \cdot (1-\epsilon_1) \cdot (x_{2,n+1} - x_{1,n+1})^+, && (78)
\end{aligned}$$

where

$$(x)^+ = \begin{cases} x & \text{if } x \geq 0, \\ 0 & \text{if } x < 0. \end{cases} \tag{79}$$

Note that $x_{2,n+1}$ must be 1 if it shall be possible for this event to occur!

Similarly, we compute

$$\begin{aligned}
\Delta\psi_3 &= \epsilon_1^{n-t} \cdot (1-\epsilon_1) \cdot (x_{3,n+1} - x_{1,n+1})^+; && (80)\\
\Delta\psi_4 &= \epsilon_1^{n} \cdot (1-\epsilon_1) \cdot (x_{4,n+1} - x_{1,n+1})^+. && (81)
\end{aligned}$$

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, we should write them as follows:

$$\begin{aligned}
\Delta\psi_2 &= \epsilon_1^t \cdot (1-\epsilon_1) \cdot (x_{2,n+1} - x_{1,n+1})^+, && (82)\\
\Delta\psi_3 &= \epsilon_1^{n-t} \cdot (1-\epsilon_1) \\
&\qquad \cdot (x_{3,n+1} - x_{2,n+1} - x_{1,n+1})^+, && (83)\\
\Delta\psi_4 &= \epsilon_1^{n} \cdot (1-\epsilon_1) \\
&\qquad \cdot (x_{4,n+1} - x_{3,n+1} - x_{2,n+1} - x_{1,n+1})^+. && (84)
\end{aligned}$$

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

$$\begin{aligned}\Delta\psi_4 &= \sum_{\mathbf{y}^{(n)}\in\mathcal{D}_{t,2}^{(4,n)}} P_{\mathbf{Y}|\mathbf{X}}\big([\mathbf{y}^{(n)}\ 1]\,\big|\,[\mathbf{1}^{(n)}\ 1]\big) \\ &\qquad \cdot (x_{4,n+1}-x_{2,n+1})^+ \qquad (85)\\ &= \sum_{d=0}^{t-1}\binom{t}{d}(1-\epsilon_1)^{t-d}\cdot\epsilon_1^{n-t+d}\cdot(1-\epsilon_1) \\ &\qquad \cdot (x_{4,n+1}-x_{2,n+1})^+ \qquad (86)\\ &= \big(\epsilon_1^{n-t}-\epsilon_1^n\big)\cdot(1-\epsilon_1)\cdot(x_{4,n+1}-x_{2,n+1})^+. \qquad (87)\end{aligned}$$

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

$$\begin{aligned}\Delta\psi_4 &= \sum_{\mathbf{y}^{(n)}\in\mathcal{D}_{t,3}^{(4,n)}} P_{\mathbf{Y}|\mathbf{X}}\big([\mathbf{y}^{(n)}\ 1]\,\big|\,[\mathbf{1}^{(n)}\ 1]\big) \\ &\qquad \cdot (x_{4,n+1}-x_{3,n+1})^+ \qquad (88)\\ &= \big(\epsilon_1^{t}-\epsilon_1^n\big)\cdot(1-\epsilon_1)\cdot(x_{4,n+1}-x_{3,n+1})^+. \qquad (89)\end{aligned}$$

From $\epsilon_1^t \geq \epsilon_1^{n-t} > \epsilon_1^n$, we can therefore now conclude that the best solution for the choice of $x_{m,n+1}$ yielding the largest increase in success probability in (82), (83), (84), (87), and (89) is as follows:

$$\begin{cases} x_{2,n+1}-x_{1,n+1}=1,\\ x_{4,n+1}-x_{2,n+1}=0,\\ x_{4,n+1}-x_{3,n+1}=1\end{cases} \quad\Longrightarrow\quad \begin{cases} x_{1,n+1}=0,\\ x_{2,n+1}=1,\\ x_{3,n+1}=0,\\ x_{4,n+1}=1.\end{cases} \qquad (90)$$

This will lead to a total increase of success probability of

$$4\Delta P_{\mathrm{c}} = \epsilon_1^t(1-\epsilon_1)+(\epsilon_1^t-\epsilon_1^n)(1-\epsilon_1). \qquad (91)$$

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 $\mathscr{C}_{\lfloor\frac{n+1}{2}\rfloor,0}^{(4,n+1)}$ by just exchanging the roles of the second and third codeword and 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 $\mathscr{C}_{\frac{n+1}{2},0}^{(4,n+1)}$. In particular, since $t=\lfloor\frac{n}{2}\rfloor < n-t$, this also proves that for even blocklength these optimal linear codes are unique.

Finally, the case with three codewords $\mathsf{M}=3$ can be proved in a similar manner. We observe that

$$\Big(\mathbf{c}_1^{(3)},\mathbf{c}_3^{(3)}\Big) \equiv \Big(\mathbf{c}_1^{(3)},\mathbf{c}_2^{(3)}\Big) \qquad (92)$$

are optimal codebooks for $n=2$. An optimal way of extending these codes is then to add columns $\mathbf{c}_2^{(3)}$ or $\mathbf{c}_3^{(3)}$.

Similarly, we also can prove that the 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. □

Note that for $\mathsf{M} = 2$ and $\mathsf{M} = 4$, the optimal codes given in Theorem 19 and Theorem 21 are linear. The proof of Theorem 21 shows that for even n, these linear codes are the unique optimal codes. 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 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 $P_{\mathrm{e}}^{(n)}\big(\mathscr{C}_{5,0}^{(4,10)}\big) \approx 2.43 \cdot 10^{-3}$. If we increase the blocklength to $n = 20$, we already achieve an average error probability $P_{\mathrm{e}}^{(n)}\big(\mathscr{C}_{10,0}^{(4,20)}\big) \approx 5.90 \cdot 10^{-6}$.

Moreover, also note that the optimal code $\mathcal{C}_{t,0}^{(4,n)}$ can be seen as a *double-flip code* consisting of the combination of the flip-code of type 0 with the flip-code of type $t > 0$:

$$\mathcal{C}_{t,0}^{(4,n)} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{x} \\ \bar{\mathbf{x}} \\ \mathbf{1} \end{pmatrix} \tag{93}$$

with $\mathbf{x}$ defined in (37).

Since we know that the success probability increases with n on a binary DMC, it is quite natural to try to construct the optimal codes recursively in n.

Corollary 23. *The optimal codebooks defined in Theorem 21 for $\mathsf{M} = 3$ and 4 can be constructed recursively in the blocklength n. We start with an optimal codebook for $n = 2$:*

$$\mathscr{C}_{\mathrm{ZC}}^{(\mathsf{M},2)*} = \left(\mathbf{c}_1^{(\mathsf{M})}, \mathbf{c}_2^{(\mathsf{M})}\right). \tag{94}$$

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

$$\begin{cases} \mathbf{c}_1^{(\mathsf{M})} & \text{if } n \bmod 2 = 1, \\ \mathbf{c}_2^{(\mathsf{M})} & \text{if } n \bmod 2 = 0. \end{cases} \tag{95}$$

6.5 Conjectured Optimal Codes on ZC for $\mathsf{M} = 5$

The idea of designing an optimal code recursively promises to be a very powerful approach. However, note that for larger values of M, the recursion might need a step-size larger than 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.

We define the following five column vectors:

$$\left\{ \mathbf{c}_1^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \mathbf{c}_2^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \mathbf{c}_3^{(5)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \right.$$
$$\left. \mathbf{c}_4^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \mathbf{c}_5^{(5)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \right\}. \tag{96}$$

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

$$\mathscr{C}_{\mathrm{ZC}}^{(5,8)*} = \left(\mathbf{c}_1^{(5)}, \mathbf{c}_2^{(5)}, \mathbf{c}_3^{(5)}, \mathbf{c}_1^{(5)}, \mathbf{c}_2^{(5)}, \mathbf{c}_3^{(5)}, \mathbf{c}_4^{(5)}, \mathbf{c}_5^{(5)}\right). \tag{97}$$

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

$$\begin{cases} \left(\mathbf{c}_4^{(5)}, \mathbf{c}_5^{(5)}\right) & \text{if } n \bmod 10 = 0, \\ \left(\mathbf{c}_1^{(5)}, \mathbf{c}_2^{(5)}\right) & \text{if } n \bmod 10 = 2, \\ \left(\mathbf{c}_1^{(5)}, \mathbf{c}_3^{(5)}\right) & \text{if } n \bmod 10 = 4, \\ \left(\mathbf{c}_3^{(5)}, \mathbf{c}_4^{(5)}\right) & \text{if } n \bmod 10 = 6, \\ \left(\mathbf{c}_2^{(5)}, \mathbf{c}_5^{(5)}\right) & \text{if } n \bmod 10 = 8. \end{cases} \tag{98}$$

For n odd we have

$$\mathscr{C}_{\mathrm{ZC}}^{(5,9)*} = \left(\mathbf{c}_1^{(5)}, \mathbf{c}_2^{(5)}, \mathbf{c}_3^{(5)}, \mathbf{c}_4^{(5)}, \mathbf{c}_5^{(5)}, \mathbf{c}_1^{(5)}, \mathbf{c}_2^{(5)}, \mathbf{c}_1^{(5)}, \mathbf{c}_3^{(5)}\right). \tag{99}$$

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

$$\begin{cases} \left(\mathbf{c}_3^{(5)}, \mathbf{c}_4^{(5)}\right) & \text{if } n \bmod 10 = 1, \\ \left(\mathbf{c}_2^{(5)}, \mathbf{c}_5^{(5)}\right) & \text{if } n \bmod 10 = 3, \\ \left(\mathbf{c}_4^{(5)}, \mathbf{c}_5^{(5)}\right) & \text{if } n \bmod 10 = 5, \\ \left(\mathbf{c}_1^{(5)}, \mathbf{c}_2^{(5)}\right) & \text{if } n \bmod 10 = 7, \\ \left(\mathbf{c}_1^{(5)}, \mathbf{c}_3^{(5)}\right) & \text{if } n \bmod 10 = 9. \end{cases} \tag{100}$$

Note that the recursive structure in (98) and (100) 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 (96) exactly twice.

For $n < 10$ the optimal code structure goes through some transient states.

6.6 Optimal Codes on BSC

Theorem 24. *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 $t \in \left\{0, 1, \ldots, \left\lfloor \frac{n}{2} \right\rfloor\right\}$.*

Proof. This proof is basically a corollary of Proposition 16. The details are omitted. □

Theorem 25. *For a BSC 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_2^*, t_3^*):*

$$\mathscr{C}_{\mathrm{BSC}}^{(\mathsf{M},n)*} = \mathscr{C}_{t_2^*, t_3^*}^{(\mathsf{M},n)}, \tag{101}$$

where we define

$$t_2^* \triangleq \left\lfloor \frac{n-1}{3} \right\rfloor, \quad t_3^* \triangleq \left\lfloor \frac{n+1}{3} \right\rfloor. \tag{102}$$

Proof. See Appendix B. □

Note that for $\mathsf{M} = 2$, the optimal codes given in Theorem 24 can be linear or nonlinear. For $\mathsf{M} = 4$, by the definition of weak flip code of type (t_2, t_3), the optimal codes in Theorem 25 are linear. However, due to the strong symmetry of the BSC, there also exist nonlinear codes with the same optimal performance.

Moreover, note that one can learn from the proof of Theorem 25 that the received vector $\mathbf{y}$ that is farthest from the three codewords when $\mathsf{M} = 3$ is

$$\mathbf{y} = (\underbrace{1, 1, \cdots, 1}_{t_1^*}, \underbrace{1, 1, \cdots, 1}_{t_2^*}, \underbrace{0, 0, \ldots, 0}_{t_3^*}). \tag{103}$$

This is identical to the optimal choice of a fourth codeword $\mathbf{x}_4$ when $\mathsf{M} = 4$.

Corollary 26. *The optimal codebooks defined in Theorem 21 for* $\mathsf{M} = 3$ *and* $\mathsf{M} = 4$ *can be constructed recursively in the blocklength* n*. We start with an optimal codebook for* $n = 2$*:*

$$\mathscr{C}_{\mathrm{BSC}}^{(\mathsf{M},2)*} = \left(\mathbf{c}_1^{(\mathsf{M})}, \mathbf{c}_3^{(\mathsf{M})}\right). \tag{104}$$

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

$$\begin{cases} \mathbf{c}_1^{(\mathsf{M})} & \text{if } n \bmod 3 = 0, \\ \mathbf{c}_2^{(\mathsf{M})} & \text{if } n \bmod 3 = 1, \\ \mathbf{c}_3^{(\mathsf{M})} & \text{if } n \bmod 3 = 2. \end{cases} \tag{105}$$

7 Pairwise Hamming Distance Structure

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 [6]. 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!

In order to demonstrate that these commonly used design criteria do not work when searching for an *optimal* code, we will now investigate the minimum Hamming distance of an optimal code. Although, as (46) shows, the error probability performance of a BSC is completely specified by the Hamming distance between codewords and received vectors, it turns out that a design based on the minimum Hamming distance can fail, even for the very symmetric BSC and even for linear codes. Recall that we have seen a first glimpse of this behavior in Example 15. In the case of a more general (and not symmetric) BAC, this is even more pronounced [5].

For the symmetric case of a BSC, one can rely on the *pairwise Hamming distance vector* as defined in Section 4.5.

For $\mathsf{M} = 3$ or $\mathsf{M} = 4$, we know from Theorem 25 that the optimal code $\mathscr{C}_{\mathrm{BSC}}^{(\mathsf{M},n)*}$ for a BSC consists of t_2^* columns $\mathbf{c}_2^{(\mathsf{M})}$, t_3^* columns $\mathbf{c}_3^{(\mathsf{M})}$, and $t_1^* \triangleq n - t_2^* - t_3^*$ columns $\mathbf{c}_1^{(\mathsf{M})}$, where the parameters t_2^* and $_3^*$ are defined in (102).

Using the shorthand $k \triangleq \lfloor \frac{n}{3} \rfloor$, we can write the optimal code parameters of (102) as

$$[t_1^*, t_2^*, t_3^*] = \begin{cases} [k+1, k-1, k] & \text{if } n \bmod 3 = 0, \\ [k+1, k, k] & \text{if } n \bmod 3 = 1, \\ [k+1, k, k+1] & \text{if } n \bmod 3 = 2. \end{cases} \tag{106}$$

Using (42) we can compute the pairwise Hamming distance vector of this code for $\mathsf{M} = 3$ as follows:

$$\mathbf{d}^{(3,n)} = \begin{cases} (2k-1, 2k+1, 2k) & \text{if } n \bmod 3 = 0, \\ (2k, 2k+1, 2k) & \text{if } n \bmod 3 = 1, \\ (2k+1, 2k+2, 2k+1) & \text{if } n \bmod 3 = 2, \end{cases} \tag{107}$$

i.e.,

$$d_{\min}^{(3,n)} = \begin{cases} 2k-1 & \text{if } n \bmod 3 = 0, \\ 2k & \text{if } n \bmod 3 = 1, \\ 2k+1 & \text{if } n \bmod 3 = 2. \end{cases} \tag{108}$$

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

$$\mathbf{d}^{(4,n)} = \begin{cases} (2k-1, 2k+1, 2k, 2k, 2k+1, 2k-1) & \text{if } n \bmod 3 = 0, \\ (2k, 2k+1, 2k, 2k, 2k+1, 2k) & \text{if } n \bmod 3 = 1, \\ (2k+1, 2k+2, 2k+1, 2k+1, 2k+2, 2k+1) & \text{if } n \bmod 3 = 2, \end{cases} \tag{109}$$

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

We will compare this optimal code with the following different weak flip code $\mathscr{C}_{\text{subopt}}^{(\mathsf{M},n)}$:

$$[t_1, t_2, t_3] = \begin{cases} [k, k, k] & \text{if } n \bmod 3 = 0, \\ [k+1, k-1, k+1] & \text{if } n \bmod 3 = 1, \\ [k+2, k, k] & \text{if } n \bmod 3 = 2. \end{cases} \tag{110}$$

This code can actually be constructed from the optimal code $\mathscr{C}_{\text{BSC}}^{(\mathsf{M},n-1)*}$ by appending a corresponding column (depending on n). In fact, by adapting the proof of Corollary 26, we can show that this second weak flip code is strictly suboptimal.

The pairwise Hamming distance vectors of this suboptimal code is given as follows. For $\mathsf{M} = 3$:

$$\mathbf{d}^{(3,n)} = \begin{cases} (2k, 2k, 2k) & \text{if } n \bmod 3 = 0, \\ (2k, 2k+2, 2k) & \text{if } n \bmod 3 = 1, \\ (2k, 2k+2, 2k+2) & \text{if } n \bmod 3 = 2, \end{cases} \tag{111}$$

i.e., $d_{\min}^{(3,n)} = 2k$ in all cases. For $\mathsf{M} = 4$ the situation is accordingly with also $d_{\min}^{(4,n)} = 2k$ in all cases.

Hence, we see that the minimum Hamming distance of the optimal code is $2k-1$ and therefore strictly smaller than the 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 27. *On a BSC for $\mathsf{M} = 3$ or $\mathsf{M} = 4$ and for all n with $n \bmod 3 = 0$ or $n \bmod 3 = 1$, the codes that maximize the minimum Hamming distance $d_{\min}^{(n)}$ can be strictly suboptimal. This is not true in the case of $n \bmod 3 = 2$.*

8 Conclusion

We have studied ultra-small block-codes to be used on the most general binary channel, the *binary asymmetric channel* (BAC) and its two special cases, the Z-channel and the binary symmetric channel. We have shown that in contrast to capacity that always can be achieved with linear codes, the best codes in the sense that they achieve the smallest average probability of error for a fixed blocklength, often are not linear. For an arbitrary blocklength, we have given the optimal construction for the cases of four or less messages. In the case of the Z-channel, we have also conjectured an optimal construction for the case of $\mathsf{M} = 5$ messages.

We have introduced a new powerful way of generating these codes recursively by using a column-wise build-up of the codebook matrix. This recursive construction might be extended to a higher number of codewords $\mathsf{M} \geq 5$, however, we might then require a larger step-size, i.e., the optimal code of blocklength n can be constructed from the optimal code of blocklength $n-k$, where the step-size k might be larger than 1. We have conjectured a $k = 2$ for the case of $\mathsf{M} = 5$ for the Z-channel.

In any case, the idea of the column-wise approach is a very powerful tool both for construction and analysis that will also help in the quest of finding the optimal construction for codes with more than four codewords.

Finally, we have shown that the well-known and commonly used code parameter *minimum Hamming distance* might not be suitable as an optimum design criterion for codes even for strongly symmetric channels like the BSC.

References

[1] Claude E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, July and October 1948.

[2] Robert G. Gallager, *Information Theory and Reliable Communication*. John Wiley & Sons, 1968.

[3] Yury Polyanskiy, H. Vincent Poor, and Sergio Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[4] Chia-Lung Wu, Po-Ning Chen, Yunghsiang S. Han, and Yan-Xiu 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.

[5] Po-Ning Chen, Hsuan-Yin Lin, and Stefan M. Moser, "On ultra-small block-codes for binary discrete memoryless channels," August 2011, in preparation.

[6] Shu Lin and Daniel J. Costello, Jr., *Error Control Coding*, 2nd ed. Prentice Hall, 2004.

[7] Stefan 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

A Appendix: Derivation of Proposition 16

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

We therefore only need to show that increasing n will always allow us to find a new flip-flop code with a better performance. In other words, Proposition 16 is proven once we have shown that

$$P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n-1)}\big) \geq \max\Big\{P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n)}\big), P_{\mathrm{e}}\big(\mathscr{C}_{t+1}^{(2,n)}\big)\Big\}. \tag{112}$$

Here we have used the following notation:

$$\mathscr{C}_t^{(2,n-1)} = \begin{pmatrix}\mathbf{x}^{(n-1)}\\ \bar{\mathbf{x}}^{(n-1)}\end{pmatrix} \tag{113}$$

is a length-$(n-1)$ flip-flop code of some type t, and

$$\mathscr{C}_t^{(2,n)} = \begin{pmatrix}[\mathbf{x}^{(n-1)}\,0]\\ [\bar{\mathbf{x}}^{(n-1)}\,1]\end{pmatrix}, \quad \mathscr{C}_{t+1}^{(2,n)} = \begin{pmatrix}[\mathbf{x}^{(n-1)}\,1]\\ [\bar{\mathbf{x}}^{(n-1)}\,0]\end{pmatrix} \tag{114}$$

are the two length-n flip-flop codes that can be derived from $\mathscr{C}_t^{(2,n-1)}$.

As shown in Proposition 18, the optimal decision rule for any flip-flop code is a threshold rule with some threshold ℓ: the decision rule for received $\mathbf{y}$ only depends on d such that

$$g(\mathbf{y}) = \begin{cases}\mathbf{x} & \text{if } 0 \leq d \leq \ell,\\ \bar{\mathbf{x}} & \text{if } \ell+1 \leq d \leq n,\end{cases} \tag{115}$$

where we use $g(\cdot)$ to denote the ML decoding rule.

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

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 $\mathscr{C}_{t+1}^{(2,n)}$ according to (114) must satisfy

$$\ell^{(n-1)} \leq \ell^{(n)} \leq \ell^{(n-1)} + 1. \tag{116}$$

Consider the code $\mathscr{C}_t^{(2,n)}$. Note that since t is unchanged ($\mathscr{C}_t^{(2,n-1)}$ is changed to $\mathscr{C}_t^{(2,n)}$), the first codeword was appended a 0, while the second codeword was appended a 1, i.e., $\mathbf{x}^{(n)} = [\mathbf{x}^{(n-1)}\,0]$ and $\bar{\mathbf{x}}^{(n)} = [\bar{\mathbf{x}}^{(n-1)}\,1]$, see (114).

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

Then secondly 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 $\bar{\mathbf{x}}^{(n-1)}$. The received length-n vector $\mathbf{y}^{(n)} = [\mathbf{y}^{(n-1)}\,1]$ has $d^{(n)} = \ell^{(n-1)} + 2 < \ell^{(n)} + 1$, i.e., it will be now decoded to $\mathbf{x}^{(n)}$. This, however, is a contradiction to the assumption that the ML decision for the code $\mathscr{C}_t^{(2,n-1)}$ was $\bar{\mathbf{x}}^{(n-1)}$.

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 $P_{\mathrm{e}}^{(n-1)}\big(\mathscr{C}_t^{(2,n-1)}\big) \geq P_{\mathrm{e}}^{(n)}\big(\mathscr{C}_t^{(2,n)}\big)$.

The error probability is given by

$$P_{\mathrm{e}} = \frac{1}{2} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})=\bar{\mathbf{x}}}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}) + \frac{1}{2} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})=\mathbf{x}}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\bar{\mathbf{x}}). \tag{117}$$

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

$$\begin{aligned}
2P_{\mathrm{e}}^{(n-1)}\big(\mathscr{C}_t^{(2,n-1)}\big) &= \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+1 \leq d^{(n-1)} \leq n-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)} \,\big|\, \mathbf{x}^{(n-1)}\big) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \leq d^{(n-1)} \leq \ell^{(n-1)}}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)} \,\big|\, \bar{\mathbf{x}}^{(n-1)}\big) \qquad (118) \\
&= \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+1 \leq d^{(n-1)} \leq n-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)} \,\big|\, \mathbf{x}^{(n-1)}\big) P_{Y|X}(1|0) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+1 \leq d^{(n-1)} \leq n-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)} \,\big|\, \mathbf{x}^{(n-1)}\big) P_{Y|X}(0|0) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \leq d^{(n-1)} \leq \ell^{(n-1)}}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)} \,\big|\, \bar{\mathbf{x}}^{(n-1)}\big) P_{Y|X}(1|1) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \leq d^{(n-1)} \leq \ell^{(n-1)}}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)} \,\big|\, \bar{\mathbf{x}}^{(n-1)}\big) P_{Y|X}(0|1) \qquad (119) \\
&= \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+2 \leq d^{(n)} \leq n}} P_{Y|X}^{n}\big([\mathbf{y}^{(n-1)}\, 1] \,\big|\, \mathbf{x}^{(n)}\big) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+1 \leq d^{(n)} \leq n-1}} P_{Y|X}^{n}\big([\mathbf{y}^{(n-1)}\, 0] \,\big|\, \mathbf{x}^{(n)}\big) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 1 \leq d^{(n)} \leq \ell^{(n-1)}+1}} P_{Y|X}^{n}\big([\mathbf{y}^{(n-1)}\, 1] \,\big|\, \bar{\mathbf{x}}^{(n)}\big) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \leq d^{(n)} \leq \ell^{(n-1)}}} P_{Y|X}^{n}\big([\mathbf{y}^{(n-1)}\, 0] \,\big|\, \bar{\mathbf{x}}^{(n)}\big). \qquad (120)
\end{aligned}$$

Here, in (119) 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 (120) we combine the terms together using the definition of $\mathscr{C}_t^{(2,n)}$ according to (114).

We can now distinguish two cases in (116):

(i) If the decision rule is unchanged, i.e., $\ell^{(n)} = \ell^{(n-1)}$, we only need to take care of the third summation in (120) that contains some terms that will now be decoded differently. 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 $[\mathbf{y}^{(n-1)}\,1]$ has $d^{(n)} = \ell^{(n-1)} + 1 = \ell^{(n)} + 1$ and will be decoded to $\bar{\mathbf{x}}^{(n)}$. Hence we must have

$$\frac{P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,1]\,\big|\,\mathbf{x}^{(n)}\big)}{P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,1]\,\big|\,\bar{\mathbf{x}}^{(n)}\big)} \le 1, \tag{121}$$

and therefore

$$\begin{aligned}
&\sum_{\substack{\mathbf{y}^{(n-1)}\\ 1\le d^{(n)}\le \ell^{(n-1)}+1}} P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,1]\,\big|\,\bar{\mathbf{x}}^{(n)}\big) \\
&= \sum_{\substack{\mathbf{y}^{(n-1)}\\ d^{(n)}=\ell^{(n-1)}+1}} \underbrace{P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,1]\,\big|\,\bar{\mathbf{x}}^{(n)}\big)}_{\ge P_{Y|X}^n\left([\mathbf{y}^{(n-1)}\,1]\,\middle|\,\mathbf{x}^{(n)}\right)} + \sum_{\substack{\mathbf{y}^{(n-1)}\\ 1\le d^{(n)}\le \ell^{(n-1)}}} P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,1]\,\big|\,\bar{\mathbf{x}}^{(n)}\big) \qquad (122)\\
&\ge \sum_{\substack{\mathbf{y}^{(n-1)}\\ d^{(n)}=\ell^{(n-1)}+1}} P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,1]\,\big|\,\mathbf{x}^{(n)}\big) + \sum_{\substack{\mathbf{y}^{(n-1)}\\ 1\le d^{(n)}\le \ell^{(n-1)}}} P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,1]\,\big|\,\bar{\mathbf{x}}^{(n)}\big). \qquad (123)
\end{aligned}$$

Hence, we get from (120):

$$\begin{aligned}
2P_{\mathrm{e}}^{(n-1)}\big(\mathscr{C}_t^{(2,n-1)}\big) \ge &\sum_{\substack{\mathbf{y}^{(n-1)}\\ \ell^{(n-1)}+2\le d^{(n)}\le n}} P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,1]\,\big|\,\mathbf{x}^{(n)}\big)\\
&+ \sum_{\substack{\mathbf{y}^{(n-1)}\\ \ell^{(n-1)}+1\le d^{(n)}\le n-1}} P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,0]\,\big|\,\mathbf{x}^{(n)}\big)\\
&+ \sum_{\substack{\mathbf{y}^{(n-1)}\\ d^{(n)}=\ell^{(n-1)}+1}} P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,1]\,\big|\,\mathbf{x}^{(n)}\big)\\
&+ \sum_{\substack{\mathbf{y}^{(n-1)}\\ 1\le d^{(n)}\le \ell^{(n-1)}}} P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,1]\,\big|\,\bar{\mathbf{x}}^{(n)}\big)\\
&+ \sum_{\substack{\mathbf{y}^{(n-1)}\\ 0\le d^{(n)}\le \ell^{(n-1)}}} P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,0]\,\big|\,\bar{\mathbf{x}}^{(n)}\big) \qquad (124)\\
= &\sum_{\substack{\mathbf{y}^{(n)}\\ \ell^{(n-1)}+1\le d^{(n)}\le n}} P_{Y|X}^n\big(\mathbf{y}^{(n)}\,\big|\,\mathbf{x}^{(n)}\big)\\
&+ \sum_{\substack{\mathbf{y}^{(n)}\\ 0\le d^{(n)}\le \ell^{(n-1)}}} P_{Y|X}^n\big(\mathbf{y}^{(n)}\,\big|\,\bar{\mathbf{x}}^{(n)}\big) \qquad (125)\\
= 2P_{\mathrm{e}}^{(n)}\big(\mathscr{C}_t^{(2,n)}\big). &\qquad (126)
\end{aligned}$$

(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 (120) that contains some terms that will now be decoded differently. 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

$[\mathbf{y}^{(n-1)}\,0]$ has $d^{(n)} = \ell^{(n-1)} + 1 = \ell^{(n)}$ and will be decoded to $\mathbf{x}^{(n)}$. Hence we must have

$$\frac{P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,0]\,\big|\,\mathbf{x}^{(n)}\big)}{P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,0]\,\big|\,\bar{\mathbf{x}}^{(n)}\big)} \geq 1, \tag{127}$$

and therefore

$$\begin{aligned}
&\sum_{\substack{\mathbf{y}^{(n-1)}\\ \ell^{(n-1)}+1\leq d^{(n)}\leq n-1}} P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,0]\,\big|\,\mathbf{x}^{(n)}\big) \\
&\quad= \sum_{\substack{\mathbf{y}^{(n-1)}\\ d^{(n)}=\ell^{(n-1)}+1}} P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,0]\,\big|\,\mathbf{x}^{(n)}\big) \\
&\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)}\\ \ell^{(n-1)}+2\leq d^{(n)}\leq n-1}} P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,0]\,\big|\,\mathbf{x}^{(n)}\big) && (128)\\
&\quad\geq \sum_{\substack{\mathbf{y}^{(n-1)}\\ d^{(n)}=\ell^{(n-1)}+1}} P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,0]\,\big|\,\bar{\mathbf{x}}^{(n)}\big) \\
&\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)}\\ \ell^{(n-1)}+2\leq d^{(n)}\leq n-1}} P_{Y|X}^n\big([\mathbf{y}^{(n-1)}\,0]\,\big|\,\mathbf{x}^{(n)}\big). && (129)\\
&&& (130)
\end{aligned}$$

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

This proves that $P_{\mathrm{e}}^{(n-1)}\big(\mathscr{C}_t^{(2,n-1)}\big) \geq P_{\mathrm{e}}^{(n)}\big(\mathscr{C}_t^{(2,n)}\big)$. It only remains to show that $P_{\mathrm{e}}^{(n-1)}\big(\mathscr{C}_t^{(2,n-1)}\big) \geq P_{\mathrm{e}}^{(n)}\big(\mathscr{C}_{t+1}^{(2,n)}\big)$. This derivation is similar and therefore omitted.

B Appendix: Derivation of Theorem 25

B.1 $\mathsf{M} = 3$

Our proof is based on induction in n. We start with an optimal code of length $n-1$ and then prove that appending a column according to the choice given in Corollary 26 will result in a new optimal code. 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 (104) is optimal is straightforward and omitted.
- We do not need to worry about any other codebook columns than those given in (39) because firstly the all-zero and the all-one column can be neglected by the same argument as used in the proof of Proposition 16, 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.
- We need to distinguish three cases in the induction from $n-1$ to n, depending on whether $n \bmod 3 = 0$, 1, or 2.

Considering the possible choices $\mathbf{c}_1^{(3)}$, $\mathbf{c}_2^{(3)}$, and $\mathbf{c}_3^{(3)}$ of (39), we see that exactly 2 components in $\mathbf{d}^{(3,n-1)}$ will be increased by 1 to form the new $\mathbf{d}^{(3,n)}$. For example, if the newly added column is $\mathbf{c}_1^{(3)}$, then $(d_{12}^{(n)}, d_{13}^{(n)}, d_{23}^{(n)}) = (d_{12}^{(n-1)}, d_{13}^{(n-1)}+1, d_{23}^{(n-1)}+1)$.

Before we start our induction proof, we make an observation about a basic property of the weak flip code given in (102).

Claim 28. *For the weak flip code of (102), the largest received Hamming distance between any* $\mathbf{y}$ *and the nearest codeword is given by the minimum Hamming distance of the codebook:*

$$\max_{\mathbf{y}} d_{\min}^{(n)}(\mathbf{y}) = d_{\min}^{(3,n)}\big(\mathscr{C}_{\mathrm{BSC}}^{(3,n)*}\big). \tag{131}$$

Proof. It is not too difficult to see that a $\mathbf{y}$ that achieves the maximum in (131) should have t_1^* ones, t_2^* ones, and t_3^* zeros in the positions where the optimal codebook consists of $\mathbf{c}_1$, $\mathbf{c}_2$, and $\mathbf{c}_3$, respectively. I.e.,

$$\mathbf{y}_{\max}^{(n)} \triangleq (\underbrace{1,1,\ldots,1}_{t_1^*},\underbrace{1,1,\ldots,1}_{t_2^*},\underbrace{0,0,\ldots,0}_{t_3^*}). \tag{132}$$

Then,

$$\begin{aligned}
\max_{\mathbf{y}} d_{\min}^{(n)}(\mathbf{y}) &= \max_{\mathbf{y}} \min\big\{d_1^{(n)}(\mathbf{y}), d_2^{(n)}(\mathbf{y}), d_3^{(n)}(\mathbf{y})\big\} && (133)\\
&= \min\big\{d_1^{(n)}(\mathbf{y}_{\max}), d_2^{(n)}(\mathbf{y}_{\max}), d_3^{(n)}(\mathbf{y}_{\max})\big\} && (134)\\
&= \min\{t_1^*+t_2^*, t_1^*+t_3^*, t_2^*+t_3^*\} && (135)\\
&= \min\big\{d_{23}^{(n)*}, d_{13}^{(n)*}, d_{12}^{(n)*}\big\} && (136)\\
&= d_{\min}\big(\mathscr{C}_{\mathrm{BSC}}^{(3,n)*}\big). && (137)
\end{aligned}$$

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

Case i: Step from $n-1=3k-1$ to $n=3k$: We start with the code $\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}$ with code parameters, pairwise Hamming distance vector, and minimum Hamming distance as follows:

$$\begin{aligned}
&\text{code parameters:} && [k, k-1, k], && (138)\\
&\text{pairw. distance vector:} && \mathbf{d}^{(3,n-1)} = (2k-1, 2k, 2k-1), && (139)\\
&\text{min. Hamming distance:} && d_{\min}^{(3,n-1)} = 2k-1. && (140)
\end{aligned}$$

The corresponding success probability formula looks as follows:

$$\begin{aligned}
3P_{\mathrm{c}}^{(n-1)}\Big(\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}\Big) &= \sum_{m=1}^{3} \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}_m^{(n-1)}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_m^{(n-1)}\big) && (141)\\
&= (1-\epsilon)^{n-1} \sum_{m=1}^{3} \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}_m^{(n-1)}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_{\mathrm{H}}(\mathbf{x}_m^{(n-1)},\mathbf{y}^{(n-1)})} && (142)\\
&= (1-\epsilon)^{n-1} \Bigg(\sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}_1^{(n-1)}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})}\\
&\qquad + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}_2^{(n-1)}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})}\\
&\qquad + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}_3^{(n-1)}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})} \Bigg) && (143)
\end{aligned}$$

$$= (1-\epsilon)^n \Bigg(\sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}_1^{(n-1)}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})} + \sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}_1^{(n-1)}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})+1} + \sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}_2^{(n-1)}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})} + \sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}_2^{(n-1)}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})+1} + \sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}_3^{(n-1)}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})} + \sum_{\mathbf{y}^{(n-1)} \in \mathcal{D}_3^{(n-1)}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})+1} \Bigg), \quad (144)$$

where in the last equality we used the trick to write

$$1 = (1-\epsilon)\left(1 + \frac{\epsilon}{1-\epsilon}\right). \quad (145)$$

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

$$[k,k,k]; \qquad \mathbf{d}^{(3,n)} = (2k,2k,2k); \qquad d_{\min}^{(3,n)} = 2k. \quad (146)$$

Now note that we can rewrite (144) in the following way:

$$3P_{\mathrm{c}}^{(n-1)}\left(\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}\right) = (1-\epsilon)^n \Bigg(\sum_{\mathbf{y}^{(n)} \in [\mathcal{D}_1^{(n-1)}\ 0]} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})} + \sum_{\mathbf{y}^{(n)} \in [\mathcal{D}_1^{(n-1)}\ 1]} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})+1} + \sum_{\mathbf{y}^{(n)} \in [\mathcal{D}_2^{(n-1)}\ 1]} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})} + \sum_{\mathbf{y}^{(n)} \in [\mathcal{D}_2^{(n-1)}\ 0]} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})+1}$$

$$+ \sum_{\mathbf{y}^{(n)} \in [\mathcal{D}_3^{(n-1)}\ 0]} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})} + \sum_{\mathbf{y}^{(n)} \in [\mathcal{D}_3^{(n-1)}\ 1]} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})+1} \Bigg). \tag{147}$$

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

$$\begin{aligned} &3P_{\mathrm{c}}^{(n)}\big(\mathcal{C}^{(3,n)}\big) \\ &\quad = (1-\epsilon)^n \Bigg(\sum_{\mathbf{y}^{(n)} \in \mathcal{D}_1^{(n)}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_1^{(n)}(\mathbf{y}^{(n)})} + \sum_{\mathbf{y}^{(n)} \in \mathcal{D}_2^{(n)}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_2^{(n)}(\mathbf{y}^{(n)})} \\ &\qquad\qquad + \sum_{\mathbf{y}^{(n)} \in \mathcal{D}_3^{(n)}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_3^{(n)}(\mathbf{y}^{(n)})} \Bigg). \end{aligned} \tag{148}$$

In order to be able to compare (147) with (148), we need to be able to compare $\mathcal{D}_i^{(n-1)}$ with $\mathcal{D}_i^{(n)}$ and $d_i^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ with $d_i^{(n)}\big(\mathbf{y}^{(n)}\big)$. 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 $\mathbf{c}_2^{(3)} = (0,1,0)^{\mathsf{T}}$ to the code of length $n-1$, it is obvious that

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_1^{(n-1)} \quad \Longrightarrow \quad [\mathbf{y}^{(n-1)}\ 0] \in \mathcal{D}_1^{(n)}; \quad d_1^{(n)}\big(\mathbf{y}^{(n)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big); \tag{149}$$

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_2^{(n-1)} \quad \Longrightarrow \quad [\mathbf{y}^{(n-1)}\ 1] \in \mathcal{D}_2^{(n)}; \quad d_2^{(n)}\big(\mathbf{y}^{(n)}\big) = d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big); \tag{150}$$

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_3^{(n-1)} \quad \Longrightarrow \quad [\mathbf{y}^{(n-1)}\ 0] \in \mathcal{D}_3^{(n)}; \quad d_3^{(n)}\big(\mathbf{y}^{(n)}\big) = d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big). \tag{151}$$

The problems are the other three cases. For example,

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_1^{(n-1)} \quad \Longrightarrow \quad [\mathbf{y}^{(n-1)}\ 1] \in \mathcal{D}_1^{(n)} \text{ or } \mathcal{D}_2^{(n)}, \tag{152}$$

depending on the exact value of $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$. Note that $[\mathbf{y}^{(n-1)}\ 1] \notin \mathcal{D}_3^{(n)}$ because we have added a 0 to the third codeword. To be able to investigate the different possible cases depending on $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$, we introduce a shorthand

$$d \triangleq d_{\min}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \tag{153}$$

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 situations:

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d,d,d) \text{ or } (d,d,d^+) \text{ or } (d,d^+,d) \text{ or } (d,d^+,d^+). \tag{154}$$

If we prolong $\mathbf{y}^{(n-1)}$ by appending a 1, then the first and the third component will be increased by 1, while the second component remains unchanged. This

means that in the third and fourth case in (154), the new vector $[\mathbf{y}^{(n-1)}\ 1]$ will belong to $\mathcal{D}_1^{(n)}$, while in the first and second case it will belong to $\mathcal{D}_2^{(n)}$. 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 codebooks minimum Hamming distance between codewords is $2k-1$ and therefore it is not possible that a vector $\mathbf{y}^{(n-1)}$ has a distance two two or more codewords smaller than k! Also, from Claim 28 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 in (132). To that goal we define a_i to be the number of positions where $\mathbf{y}^{(n-1)}$ differs from $\mathbf{y}_{\max}^{(n-1)}$, but only considering the corresponding t_i^* positions, i.e., $0 \leq a_i \leq t_i^*$, $i = 1, 2, 3$. 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 an associative matrix

$$\begin{pmatrix} t_1^* - a_1 & t_2^* - a_2 & a_3 \\ t_1^* - a_1 & a_2 & t_3^* - a_3 \\ a_1 & t_2^* - a_2 & t_3^* - a_3 \end{pmatrix} = \begin{pmatrix} k - a_1 & (k-1) - a_2 & a_3 \\ k - a_1 & a_2 & k - a_3 \\ a_1 & (k-1) - a_2 & k - a_3 \end{pmatrix} \tag{155}$$

from which the received Hamming distance vector can be computed as follows:

$$\begin{pmatrix} d_1^{(n-1)} \\ d_2^{(n-1)} \\ d_3^{(n-1)} \end{pmatrix} = \begin{pmatrix} k - a_1 & (k-1) - a_2 & a_3 \\ k - a_1 & a_2 & k - a_3 \\ a_1 & (k-1) - a_2 & k - a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \tag{156}$$

It is straightforward to prove the following claim.

Claim 29. *There exists no integer solution of a_i's, $0 \leq a_1 \leq k$, $0 \leq a_2 \leq k-1$, $0 \leq a_3 \leq k$, that satisfies*

$$\begin{pmatrix} k - a_1 & (k-1) - a_2 & a_3 \\ k - a_1 & a_2 & k - a_3 \\ a_1 & (k-1) - a_2 & k - a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d \\ d \end{pmatrix} \text{ or } \begin{pmatrix} d \\ d \\ d^+ \end{pmatrix} \text{ or } \begin{pmatrix} d^+ \\ d \\ d \end{pmatrix} \tag{157}$$

for $k \leq d \leq 2k-1$ and $d^+ > d$. But there do exist integer solutions that satisfy

$$\begin{pmatrix} k - a_1 & (k-1) - a_2 & a_3 \\ k - a_1 & a_2 & k - a_3 \\ a_1 & (k-1) - a_2 & k - a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d^+ \\ d \end{pmatrix}. \tag{158}$$

Hence, we have shown that

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_1^{(n-1)} \implies [\mathbf{y}^{(n-1)}\ 1] \in \mathcal{D}_1^{(n)}; \quad d_1^{(n)}\big(\mathbf{y}^{(n)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) + 1. \tag{159}$$

Similarly,

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_2^{(n-1)} \quad \implies \quad [\mathbf{y}^{(n-1)}\ 0] \in \mathcal{D}_1^{(n)} \text{ or } \mathcal{D}_2^{(n)} \text{ or } \mathcal{D}_3^{(n)}, \tag{160}$$

depending on the exact value of $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$:

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d, d, d) \text{ or } (d, d, d^+) \text{ or } (d^+, d, d) \text{ or } (d^+, d, d^+). \tag{161}$$

In the fourth case $[\mathbf{y}^{(n-1)}\ 0]$ will remain in $\mathcal{D}_2^{(n)}$, in all other three cases it will change to another decision region. However all these three cases are not possible according to (157).

Finally,

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_3^{(n-1)} \quad \Longrightarrow \quad [\mathbf{y}^{(n-1)}\ 1] \in \mathcal{D}_2^{(n)} \text{ or } \mathcal{D}_3^{(n)}, \tag{162}$$

depending on the exact value of $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$:

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d, d, d) \text{ or } (d^+, d, d) \text{ or } (d, d^+, d) \text{ or } (d^+, d^+, d). \tag{163}$$

In the first and second case $[\mathbf{y}^{(n-1)}\,1]$ will change to $\mathcal{D}_2^{(n)}$, in the other two cases it will remain in $\mathcal{D}_3^{(n)}$. Again, the first and the second case are not possible according to (157).

Hence, we have shown that

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_1^{(n-1)} \Longrightarrow [\mathbf{y}^{(n-1)}\ 1] \in \mathcal{D}_1^{(n)}; \quad d_1^{(n)}\big(\mathbf{y}^{(n)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) + 1; \tag{164}$$

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_2^{(n-1)} \Longrightarrow [\mathbf{y}^{(n-1)}\ 0] \in \mathcal{D}_2^{(n)}; \quad d_2^{(n)}\big(\mathbf{y}^{(n)}\big) = d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) + 1; \tag{165}$$

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_3^{(n-1)} \Longrightarrow [\mathbf{y}^{(n-1)}\ 1] \in \mathcal{D}_3^{(n)}; \quad d_3^{(n)}\big(\mathbf{y}^{(n)}\big) = d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) + 1. \tag{166}$$

But this proves that the success probability of (148) is identical to the success probability of (147)! So in spite of increasing the length $n-1$ by 1, we have not improved our performance.

2. **Appending $\mathbf{c}_1^{(3)}$:** Next, we investigate what happens if we append $\mathbf{c}_1^{(3)} = (0,0,1)^\intercal$. The new code has the following parameters:

$$[k+1, k-1, k]; \qquad \mathbf{d}^{(3,n)} = (2k-1, 2k+1, 2k); \qquad d_{\min}^{(3,n)} = 2k-1. \tag{167}$$

One of the three problematic cases now is

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_1^{(n-1)} \quad \Longrightarrow \quad [\mathbf{y}^{(n-1)}\ 1] \in \mathcal{D}_1^{(n)} \text{ or } [\mathbf{y}^{(n-1)}\ 1] \in \mathcal{D}_3^{(n)}, \tag{168}$$

depending on the exact value of $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ given in (154). If we prolong $\mathbf{y}^{(n-1)}$ by appending a 1, now the first and the second component will be increased by 1, while the third component remains unchanged. This means that in the first and third case the new vector $[\mathbf{y}^{(n-1)}\ 1]$ will belong to $\mathcal{D}_3^{(n)}$, while in the second and fourth cases it will belong to $\mathcal{D}_1^{(n)}$. According to Claim 29 both the third and and the fourth cases are possible and do happen. In the cases where $[\mathbf{y}^{(n-1)}\ 1] \in \mathcal{D}_3^{(n)}$ we then have that

$$d_1^{(n)}\big(\mathbf{y}^{(n)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \tag{169}$$

without the additional increase by 1. This then means that the success probability of (148) is strictly larger than the success probability of $\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}$ and the choice of $\mathbf{c}_1^{(3)}$ is effective.

The investigation of the other two problematic cases is similar and omitted.

3. **Appending $\mathbf{c}_3^{(3)}$:** Finally, we look at the case when we append $\mathbf{c}_3^{(3)} = (0,1,1)^\mathsf{T}$. The new code has the following parameters:

$$[k, k-1, k+1]; \qquad \mathbf{d}^{(3,n)} = (2k, 2k+1, 2k-1); \qquad d_{\min}^{(3,n)} = 2k-1. \quad (170)$$

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 $\mathbf{c}_3^{(3)}$ are optimal. We decide to choose $\mathbf{c}_1^{(3)}$.

Case ii: Step from $n-1 = 3k$ to $n = 3k+1$: In this case, we start with the code $\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}$ with code parameters, pairwise Hamming distance vector, and minimum Hamming distance as follows:

$$\begin{aligned} &\text{code parameters:} && [k+1, k-1, k], && (171)\\ &\text{pairw. distance vector:} && \mathbf{d}^{(3,n-1)} = (2k-1, 2k+1, 2k), && (172)\\ &\text{min. Hamming distance:} && d_{\min}^{(3,n-1)} = 2k-1. && (173) \end{aligned}$$

If we append $\mathbf{c}_1^{(3)} = (0,0,1)^\mathsf{T}$, we get a new code with the following parameters:

$$[k+2, k-1, k]; \qquad \mathbf{d}^{(3,n)} = (2k-1, 2k+2, 2k+1); \qquad d_{\min}^{(3,n)} = 2k-1. \quad (174)$$

If we append $\mathbf{c}_2^{(3)} = (0,1,0)^\mathsf{T}$, we get a new code with the following parameters:

$$[k+1, k, k]; \qquad \mathbf{d}^{(3,n)} = (2k, 2k+1, 2k+1); \qquad d_{\min}^{(3,n)} = 2k. \quad (175)$$

If we append $\mathbf{c}_3^{(3)} = (0,1,1)^\mathsf{T}$, we get a new code with the following parameters:

$$[k+1, k-1, k+1]; \qquad \mathbf{d}^{(3,n)} = (2k, 2k+2, 2k); \qquad d_{\min}^{(3,n)} = 2k. \quad (176)$$

The corresponding investigation of possible situations now reads as follows.

Claim 30. *There exists no solution of a_i's, $0 \le a_1 \le k$, $0 \le a_2 \le k-1$, $0 \le a_3 \le k$, that satisfies*

$$\begin{pmatrix} (k+1)-a_1 & (k-1)-a_2 & a_3 \\ (k+1)-a_1 & a_2 & k-a_3 \\ a_1 & (k-1)-a_2 & k-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d \\ d \end{pmatrix} \text{ or } \begin{pmatrix} d \\ d^+ \\ d \end{pmatrix} \text{ or } \begin{pmatrix} d \\ d \\ d^+ \end{pmatrix} \quad (177)$$

for $k \le d \le 2k-1$ and $d^+ > d$. But there do exist solutions that satisfy

$$\begin{pmatrix} (k+1)-a_1 & (k-1)-a_2 & a_3 \\ (k+1)-a_1 & a_2 & k-a_3 \\ a_1 & (k-1)-a_2 & k-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d^+ \\ d \\ d \end{pmatrix}. \quad (178)$$

The investigation is similar and shows that appending $\mathbf{c}_3^{(3)}$ is strictly suboptimal, while appending $\mathbf{c}_1^{(3)}$ and $\mathbf{c}_2^{(3)}$ are equivalent and optimal.

Case iii: Step from $n-1=3k+1$ to $n=3k+2$: In this case, we start with the code $\mathcal{C}^{(3,n-1)}_{t_2^*,t_3^*}$ with code parameters, pairwise Hamming distance vector, and minimum Hamming distance as follows:

$$\begin{aligned} &\text{code parameters:} && [k+1,k,k], && (179)\\ &\text{pairw. distance vector:} && \mathbf{d}^{(3,n-1)}=(2k,2k+1,2k+1), && (180)\\ &\text{min. Hamming distance:} && d^{(3,n-1)}_{\min}=2k. && (181)\end{aligned}$$

If we append $\mathbf{c}_1^{(3)}=(0,0,1)^{\mathsf{T}}$, we get a new code with the following parameters:

$$[k+2,k,k]; \qquad \mathbf{d}^{(3,n)}=(2k,2k+2,2k+2); \qquad d^{(3,n)}_{\min}=2k. \tag{182}$$

If we append $\mathbf{c}_2^{(3)}=(0,1,0)^{\mathsf{T}}$, we get a new code with the following parameters:

$$[k+1,k+1,k]; \qquad \mathbf{d}^{(3,n)}=(2k+1,2k+1,2k+2); \qquad d^{(3,n)}_{\min}=2k+1. \tag{183}$$

If we append $\mathbf{c}_3^{(3)}=(0,1,1)^{\mathsf{T}}$, we get a new code with the following parameters:

$$[k+1,k,k+1]; \qquad \mathbf{d}^{(3,n)}=(2k+1,2k+2,2k+1); \qquad d^{(3,n)}_{\min}=2k+1. \tag{184}$$

The corresponding investigation of possible situations now reads as follows.

Claim 31. *There exists no solution of a_i's, $0\le a_1\le k$, $0\le a_2\le k-1$, $0\le a_3\le k$, that satisfies*

$$\begin{pmatrix}(k+1)-a_1 & k-a_2 & a_3\\ (k+1)-a_1 & a_2 & k-a_3\\ a_1 & k-a_2 & k-a_3\end{pmatrix}\begin{pmatrix}1\\1\\1\end{pmatrix}=\begin{pmatrix}d\\d\\d\end{pmatrix}\ \text{or}\ \begin{pmatrix}d^+\\d\\d\end{pmatrix}\ \text{or}\ \begin{pmatrix}d\\d^+\\d\end{pmatrix} \tag{185}$$

for $k\le d\le 2k-1$ and $d^+>d$. But there do exist solutions that satisfy

$$\begin{pmatrix}(k+1)-a_1 & k-a_2 & a_3\\ (k+1)-a_1 & a_2 & k-a_3\\ a_1 & k-a_2 & k-a_3\end{pmatrix}\begin{pmatrix}1\\1\\1\end{pmatrix}=\begin{pmatrix}d\\d\\d^+\end{pmatrix}. \tag{186}$$

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.

B.2 $\mathsf{M}=4$

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:

$$\begin{pmatrix}d_1^{(n)}\\d_2^{(n)}\\d_3^{(n)}\\d_4^{(n)}\end{pmatrix}=\begin{pmatrix}t_1-a_1 & t_2-a_2 & a_3\\ t_1-a_1 & a_2 & t_3-a_3\\ a_1 & t_2-a_2 & t_3-a_3\\ a_1 & a_2 & a_3\end{pmatrix}\begin{pmatrix}1\\1\\1\end{pmatrix}. \tag{187}$$

The derivation follows then exactly the same lines as in Appendix B.1. 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 13, like, e.g., $\mathbf{c}=(0,0,0,1)^{\mathsf{T}}$ and prove that they are strictly suboptimal. The details are omitted.