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

Intermediate Report of NSC Project

"Ultra-Short Blocklength Communication"

Date:	31 May 2012
Project-Number:	NSC 100-2221-E-009-068-MY3
Project Duration:	1 August 2011 – 31 July 2014
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

Optimal block-codes with a small number of codewords are investigated for the binary asymmetric channel (BAC), including the two special cases of 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 ZC and for the BSC in the cases of two, three, and four codewords and an arbitrary finite blocklength. For a general BAC, it is shown that so-called *flip codes* are optimal codes with two codewords. For the BSC and the ZC, it is shown that so-called *weak flip codes* are optimal codes with three or four codewords.

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

Keywords: Binary asymmetric channel (BAC), binary symmetric channel (BSC), channel capacity, finite blocklength, flip codes, maximum likelihood (ML) decoder, minimum average error probability, optimal codes, weak flip codes, Z-channel.

Contents

1 Introduction

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

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

During the last years, there has been an increased interest in the theoretical understanding of finite-length coding [2]–[11]. There are several possible ways on how one can approach the problem of finite-length codes. In [2], the authors fix an acceptable error probability and a finite blocklength and then find bounds on the maximal achievable transmission rate. This parallels the method of Shannon who set the acceptable error probability to zero, but allowed infinite blocklength, and then found the maximum achievable transmission rate (the capacity). A typical example in [2] shows that for a blocklength of 1800 channel uses and for an error probability of 10^{-6}, one can achieve a rate of approximately 80 percent of the capacity of a certain binary symmetric channel.

The publication of [2] spawned many successive investigations. In [3] and [4] the techniques of [2] are applied in the situation of multiple-access and fading channels, respectively. In [5] and [6] the authors present analyses about finite-blocklength systems with additional queuing constraints. In [7] practical finite-length restrictions on the codewords are discussed for an exponential server timing channel. The ideas of [2] have also been applied in the context of source coding [8].

In another approach, one fixes the transmission rate and studies how the error probability depends on the blocklength n (i.e., one basically studies error exponents, but for relatively small n [12]). For example, [9] and [10] introduce new random coding bounds that enable simple numerical evaluation of the error probability for finite blocklengths and for various types of practical decoders. In [11] it is shown that error exponents and some of the important parameters introduced in [2] are closely related.

All these results have in common that they are related to Shannon's ideas in the sense that they try to make fundamental statements about what is possible and what not. The exact manner how these systems have to be built is ignored on purpose.

Our approach in this paper is different. Based on the insight that for very short blocklength, one has no big hope of transmitting much information with acceptable error probability, we concentrate on codes with a small *fixed* number of codewords: so called *ultra-small block-codes*. By this reduction of the transmission rates, our results are directly applicable even for very short blocklengths. In contrast to [2]–[11] that provide *bounds* on the best possible *theoretical performance*, we try to find a *best* possible *design* that minimizes the average error probability. Hence, we put a big emphasis on finding insights in how to actually build an optimal system. In this respect, this paper could rather be compared to [13]. There the authors try to

describe the empirical distribution of good codes (i.e., of codes that approach capacity with vanishing error probability) and show that for a large enough blocklength, the empirical distribution of certain good codes converges in the sense of divergence to a set of input distributions that maximize the input-output mutual information. Note, however, that [13] again focuses on the asymptotic regime, while our focus lies on finite blocklength.

There are interesting applications for ultra-small block-codes, e.g., in the situation of establishing an initial connection in a wireless link: the amount of information that needs to be transmitted during the setup of the link is very limited, usually only a couple of bits, but these bits need to be transmitted in very short time (e.g., blocklength in the range of $n = 20$ to $n = 30$) with the highest possible reliability [14]. Another important application for ultra-small block-codes is in the area of *quality of service (QoS)*. In many delay-sensitive wireless systems like, e.g., voice over IP (VoIP) and wireless interactive and streaming video applications, it is essential to comply with certain limitations on queuing delays or buffer violation probabilities [5]–[7]. Hence, it is of significant interest to conduct an analysis of (and to provide predictions for) the performance levels of practical finite-blocklength systems. Note that while the motivation of this work focuses on rather smaller values of n, our results nevertheless hold for arbitrary finite n.

The study of ultra-small block-codes is interesting not only because of the above mentioned direct applications, but because their analytic description is a first step to a better fundamental understanding of optimal *nonlinear* coding schemes (with ML decoding) and of their performance based on the *exact* error probability rather than on an upper bound on the achievable error probability derived from the union bound or the mutual information density bound and its statistics [15], [16].

To simplify our analysis, we have restricted ourselves for the moment to a binary discrete memoryless channel, that we call in its most general form *binary asymmetric channel (BAC)*. The two most important special cases of the BAC, the *binary symmetric channel (BSC)* and the *Z-channel (ZC)*, are then investigated more in detail.

Our main contributions are as follows:

- We provide first fundamental insights in *optimal nonlinear code design* for the BAC. Note that there exists a vast literature about linear codes, their properties and good linear design (e.g., [17]), but nonlinear codes are mostly unexplored.[1]

- We provide optimal code constructions for BSC and ZC for an arbitrary finite blocklength n and for $\mathsf{M} = 2$, 3 and 4 codewords. For the ZC we also conjecture an optimal design for $\mathsf{M} = 5$.

- We provide new insights in the optimal code construction for the general BAC for an arbitrary finite blocklength n and for $\mathsf{M} = 2$ codewords.

- We propose a new approach to the design and analysis of block-codes: instead of focusing at the codewords (i.e., the rows in the codebook matrix), we look at the codebook matrix in a *column-wise* manner.

[1] Note that some of the code designs proposed in this paper actually have interesting "linear-like" properties and can be considered as generalizations of linear codes with 2^k codewords to codes with a general number of codewords M. For more details see [18].

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

As is common in coding theory, vectors (denoted by bold face Roman letters, e.g., $\mathbf{x}$) are row-vectors. However, for simplicity of notation and to avoid a large number of transpose-signs, we slightly misuse this notational convention for one special case: any vector $\mathbf{c}$ is a column-vector. It should be always clear from the context because these vectors are used to build codebook matrices and are therefore also conceptually quite different from the transmitted codewords $\mathbf{x}$ or the received sequence $\mathbf{y}$. Otherwise our used notation follows the main stream. We use capital letters for random quantities and small letters for realizations; sets are denoted by a calligraphic font, e.g., $\mathcal{D}$; and constants are depicted by Greek letters, small Romans or a special font, e.g., M.

2 Channel Model and System Description

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

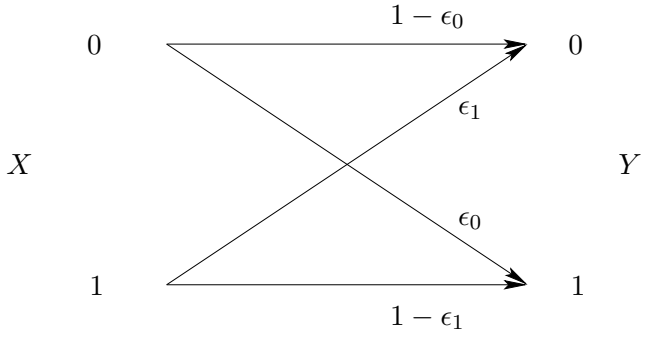

Figure 1: The binary asymmetric channel (BAC).

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

$$0 \le \epsilon_0 \le \epsilon_1 \le 1, \tag{1}$$

$$\epsilon_0 \le 1 - \epsilon_0, \tag{2}$$

$$\epsilon_0 \le 1 - \epsilon_1. \tag{3}$$

Note that in the case when $\epsilon_0 > \epsilon_1$, we simply flip all zeros to ones and vice versa to get an equivalent channel with $\epsilon_0 \le \epsilon_1$. For the case when $\epsilon_0 > 1 - \epsilon_0$, we flip the output Y, i.e., change all output zeros to ones and ones to zeros, to get an equivalent channel with $\epsilon_0 \le 1 - \epsilon_0$. Note that (2) can be simplified to $\epsilon_0 \le \frac{1}{2}$ and is actually

implied by (1) and (3). And for the case when $\epsilon_0 > 1 - \epsilon_1$, we flip the input X to get an equivalent channel that satisfies $\epsilon_0 \leq 1 - \epsilon_1$.

We have depicted the region of possible choices of the parameters ϵ_0 and ϵ_1 in Figure 2. The region of interesting choices given by (1)–(3) is denoted by Ω.

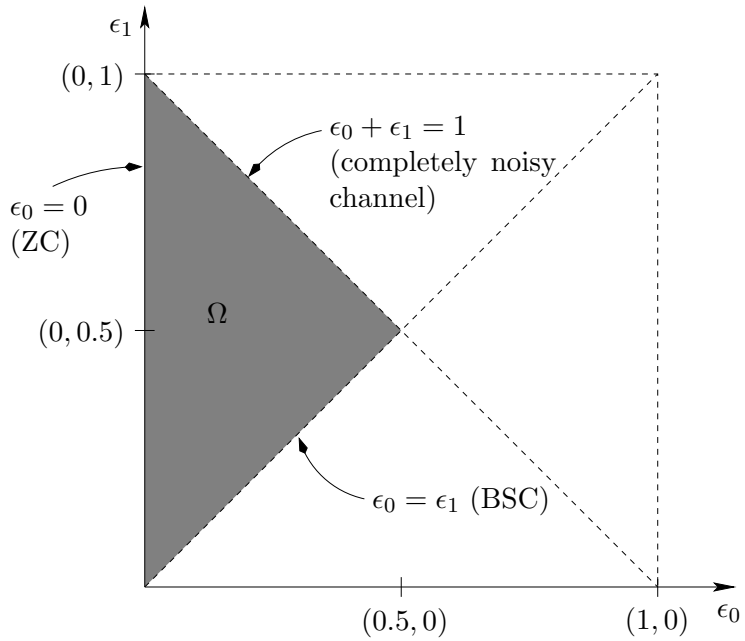

Figure 2: Region of possible choices of the channel parameters ϵ_0 and ϵ_1 of a BAC. The shaded area corresponds to the interesting area according to (1)–(3).

Note that the boundaries of Ω correspond to three special cases: The *binary symmetric channel (BSC)* has equal cross-over probabilities $\epsilon_0 = \epsilon_1 = \epsilon$, see Figure 3. As discussed above (see (2)) and without loss of generality, we assume that $\epsilon \leq \frac{1}{2}$.

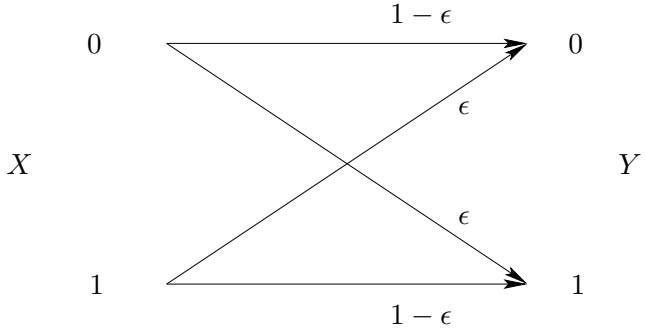

Figure 3: The binary symmetric channel (BSC).

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

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

The following three definitions are commonly used.

Definition 1. 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 ($m = 1, \ldots, \mathsf{M}$), an encoder that maps

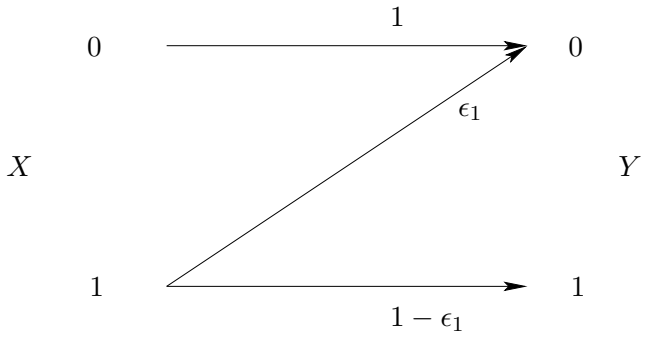

Figure 4: The Z-channel (ZC).

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 $\mathbf{y}$.

The performance of a coding scheme is described by its average probability of making a decoding error.

Definition 2. Given that message m has been sent, let λ_m be the *probability of a decoding error* of an (M, n) coding scheme $\mathscr{C}^{(\mathsf{M},n)}$:

$$\lambda_m\big(\mathscr{C}^{(\mathsf{M},n)}\big) \triangleq \Pr[g(\mathbf{Y}) \neq m \,|\, \mathbf{X} = \mathbf{x}_m] \tag{4}$$

$$= \sum_{\mathbf{y}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m)\mathrm{I}\{g(\mathbf{y}) \neq m\}, \tag{5}$$

where $\mathrm{I}\{\cdot\}$ is the indicator function whose value is 1 if the statement is correct and 0 otherwise. The *average error probability* $P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big)$ of an (M, n) coding scheme $\mathscr{C}^{(\mathsf{M},n)}$ is defined as

$$P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) \triangleq \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \lambda_m\big(\mathscr{C}^{(\mathsf{M},n)}\big). \tag{6}$$

Sometimes it will be more convenient to focus on the probability of *not* making any error, denoted *success probability* ψ_m:

$$\psi_m\big(\mathscr{C}^{(\mathsf{M},n)}\big) \triangleq \Pr[g(\mathbf{Y}) = m \,|\, \mathbf{X} = \mathbf{x}_m] \tag{7}$$

and on the corresponding *average success probability*[2] $P_{\mathrm{c}}\big(\mathscr{C}^{(\mathsf{M},n)}\big)$.

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

$$g(\mathbf{y}) \triangleq \operatorname*{argmax}_{1 \leq m \leq \mathsf{M}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m). \tag{8}$$

Note that for equally likely messages, an ML decoder is equivalent to a *maximum a posteriori (MAP)* decoder and is therefore optimal.

Definition 3. For given a coding scheme $\mathscr{C}^{(\mathsf{M},n)}$, we define the *decoding region* $\mathcal{D}_m$ corresponding to the mth codeword $\mathbf{x}_m$ as follows:

$$\mathcal{D}_m \triangleq \{\mathbf{y} \colon g(\mathbf{y}) = m\}. \tag{9}$$

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

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

Definition 4. It is usual to write the codebook $\mathscr{C}^{(\mathsf{M},n)}$ as an $\mathsf{M} \times n$ matrix with its M rows corresponding to the M codewords:

$$\mathscr{C}^{(\mathsf{M},n)} = \begin{pmatrix} — \, \mathbf{x}_1 \, — \\ \vdots \\ — \, \mathbf{x}_{\mathsf{M}} \, — \end{pmatrix} = \begin{pmatrix} | & | & & | \\ \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_n \\ | & | & & | \end{pmatrix}. \tag{10}$$

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

Remark 5. Since we assume equally likely messages, any permutation of rows only changes the assignment of codewords to messages and has no impact on the performance. We consider two codes with permuted rows as being *equal*, i.e., a code is actually a *set* of codewords, where the ordering of the codewords is irrelevant.

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

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

3 Preliminaries

3.1 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 $y_j = \beta$. Here, $\mathbf{x}_m = (x_{m,1}, x_{m,2}, \ldots, x_{m,n})$, for $m \in \{1, 2, \ldots, \mathsf{M}\}$, is the mth codeword and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ is the received sequence. The conditional probability of the received vector given the sent codeword can be written as

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

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

$$P_{Y|X}^n(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{n} P_{Y|X}(y_j|x_j). \tag{12}$$

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

be written as

$$P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) = \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y}) \neq m}} P_{Y|X}^{n}(\mathbf{y}|\mathbf{x}_m), \tag{13}$$

$$= \frac{(1-\epsilon_0)^n}{\mathsf{M}} \sum_{\mathbf{y}} \sum_{\substack{m=1 \\ m \neq g(\mathbf{y})}}^{\mathsf{M}} \left(\frac{\epsilon_0}{1-\epsilon_0}\right)^{d_{01}(\mathbf{x}_m,\mathbf{y})} \left(\frac{\epsilon_1}{1-\epsilon_0}\right)^{d_{10}(\mathbf{x}_m,\mathbf{y})} \left(\frac{1-\epsilon_1}{1-\epsilon_0}\right)^{d_{11}(\mathbf{x}_m,\mathbf{y})} \tag{14}$$

where $g(\mathbf{y})$ is the ML decision (8) for the observation $\mathbf{y}$.

3.2 Error (and Success) Probability of the BSC

In the special case of a BSC, (14) simplifies to

$$P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) = \frac{(1-\epsilon)^n}{\mathsf{M}} \sum_{\mathbf{y}} \sum_{\substack{m=1 \\ g(\mathbf{y}) \neq m}}^{\mathsf{M}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_{\mathrm{H}}(\mathbf{x}_m,\mathbf{y})} \tag{15}$$

where $d_{\mathrm{H}}(\cdot,\cdot) \triangleq d_{01}(\cdot,\cdot) + d_{10}(\cdot,\cdot)$ is the Hamming distance.

The success probability is accordingly

$$P_{\mathrm{c}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) = \frac{(1-\epsilon)^n}{\mathsf{M}} \sum_{\mathbf{y}} \sum_{\substack{m=1 \\ g(\mathbf{y}) = m}}^{\mathsf{M}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_{\mathrm{H}}(\mathbf{x}_m,\mathbf{y})}. \tag{16}$$

3.3 Error (and Success) Probability of the ZC

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

$$P_{\mathrm{c}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) = \frac{1}{\mathsf{M}} \sum_{\mathbf{y}} \sum_{\substack{m=1 \\ m = g(\mathbf{y})}}^{\mathsf{M}} \mathrm{I}\left\{d_{01}(\mathbf{x}_m,\mathbf{y}) = 0\right\} \epsilon_1^{d_{10}(\mathbf{x}_m,\mathbf{y})} (1-\epsilon_1)^{d_{11}(\mathbf{x}_m,\mathbf{y})} \tag{17}$$

$$= \frac{1}{\mathsf{M}} \sum_{\mathbf{y}} \sum_{\substack{m=1 \\ m = g(\mathbf{y})}}^{\mathsf{M}} \mathrm{I}\left\{d_{01}(\mathbf{x}_m,\mathbf{y}) = 0\right\} \left(\frac{\epsilon_1}{1-\epsilon_1}\right)^{d_{10}(\mathbf{x}_m,\mathbf{y})} (1-\epsilon_1)^{d_{11}(\mathbf{x}_m,\mathbf{y}) + d_{10}(\mathbf{x}_m,\mathbf{y})} \tag{18}$$

$$= \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y}) = m}} \mathrm{I}\left\{d_{01}(\mathbf{x}_m,\mathbf{y}) = 0\right\} \left(\frac{\epsilon_1}{1-\epsilon_1}\right)^{d_{10}(\mathbf{x}_m,\mathbf{y})} (1-\epsilon_1)^{w_{\mathrm{H}}(\mathbf{x}_m)}. \tag{19}$$

The error probability formula is accordingly

$$P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big) = \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y}) \neq m}} \mathrm{I}\left\{d_{01}(\mathbf{x}_m,\mathbf{y}) = 0\right\} \left(\frac{\epsilon_1}{1-\epsilon_1}\right)^{d_{10}(\mathbf{x}_m,\mathbf{y})} (1-\epsilon_1)^{w_{\mathrm{H}}(\mathbf{x}_m)}. \tag{20}$$

3.4 Pairwise Hamming Distance

The minimum Hamming distance is a well-known and often used quality criterion of a codebook. Unfortunately, a design based on the minimum Hamming distance can fail even for linear codes and even for a very symmetric channel like the BSC whose error probability performance is completely specified by the Hamming distances between codewords and received vectors (see also Section 8.3).

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

Definition 6. Given a codebook $\mathscr{C}^{(\mathsf{M},n)}$ with codewords $\mathbf{x}_m$, $1 \le m \le \mathsf{M}$, we define the *pairwise Hamming distance vector* $\mathbf{d}\big(\mathscr{C}^{(\mathsf{M},n)}\big)$ of length $\frac{1}{2}(\mathsf{M}-1)\mathsf{M}$ as

$$\begin{aligned}\mathbf{d}\big(\mathscr{C}^{(\mathsf{M},n)}\big) \triangleq \Big(& d_{\mathrm{H}}(\mathbf{x}_1,\mathbf{x}_2),\\ & d_{\mathrm{H}}(\mathbf{x}_1,\mathbf{x}_3),\ d_{\mathrm{H}}(\mathbf{x}_2,\mathbf{x}_3),\\ & d_{\mathrm{H}}(\mathbf{x}_1,\mathbf{x}_4),\ d_{\mathrm{H}}(\mathbf{x}_2,\mathbf{x}_4),\ d_{\mathrm{H}}(\mathbf{x}_3,\mathbf{x}_4),\\ & \ldots,\\ & d_{\mathrm{H}}(\mathbf{x}_1,\mathbf{x}_{\mathsf{M}}),\ d_{\mathrm{H}}(\mathbf{x}_2,\mathbf{x}_{\mathsf{M}}),\ldots,\ d_{\mathrm{H}}(\mathbf{x}_{\mathsf{M}-1},\mathbf{x}_{\mathsf{M}})\Big). \end{aligned} \tag{21}$$

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

4 Flip Codes and Weak Flip Codes

We next introduce some special codebooks that will be used later on.

Definition 7. The *flip code of type* t, $\mathscr{C}_t^{(2,n)}$, for $t \in \big\{0,1,\ldots,\lfloor \frac{n}{2} \rfloor\big\}$ is a code with $\mathsf{M}=2$ codewords defined by the following codebook matrix:

$$\mathscr{C}_t^{(2,n)} = \begin{pmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{pmatrix} \triangleq \begin{pmatrix}\mathbf{x}\\ \bar{\mathbf{x}}\end{pmatrix} = \begin{pmatrix}0 & \cdots & 0 & \overbrace{1 & \cdots & 1}^{t \text{ columns}}\\ 1 & \cdots & 1 & 0 & \cdots & 0\end{pmatrix}. \tag{22}$$

Defining the column vectors

$$\left\{\mathbf{c}_1^{(2)} \triangleq \begin{pmatrix}0\\1\end{pmatrix},\ \mathbf{c}_2^{(2)} \triangleq \begin{pmatrix}1\\0\end{pmatrix}\right\}, \tag{23}$$

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

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

The columns given in the set in (23) are called *candidate columns*. We see that they are flipped versions of each other, therefore also the name of the code.

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

Definition 8. For an $\mathsf{M} > 2$, a length-M candidate column $\mathbf{c}$ is called a *weak flip column* if its first component is 0 and its Hamming weight equals to $\lfloor \frac{\mathsf{M}}{2} \rfloor$ or $\lceil \frac{\mathsf{M}}{2} \rceil$.

We see that a weak flip column contains an equal or at least almost equal number of zeros and ones. Note, however, that the candidate columns of the flip code in (23) are not weak flip columns, i.e., the definition of weak flip columns is only meaningful for $\mathsf{M} > 2$.

Based on these weak flip columns we define the family of *weak flip codes*.

Definition 9. A *weak flip code* is defined by a codebook matrix that is constructed solely by weak flip columns.

For $\mathsf{M} = 3$ or $\mathsf{M} = 4$, we define the weak flip codes more specifically as follows.

Definition 10. A *weak flip code of type* (t_2, t_3), $\mathscr{C}_{t_2,t_3}^{(\mathsf{M},n)}$, with $\mathsf{M} = 3$ or $\mathsf{M} = 4$ codewords is defined by a codebook matrix that consists of[3] firstly $t_1 \triangleq n - t_2 - t_3$ columns $\mathbf{c}_1^{(\mathsf{M})}$, then t_2 columns $\mathbf{c}_2^{(\mathsf{M})}$, and finally t_3 columns $\mathbf{c}_3^{(\mathsf{M})}$, where

$$\left\{ \mathbf{c}_1^{(3)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \ \mathbf{c}_2^{(3)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \ \mathbf{c}_3^{(3)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\} \tag{24}$$

or

$$\left\{ \mathbf{c}_1^{(4)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \ \mathbf{c}_2^{(4)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \ \mathbf{c}_3^{(4)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \right\}, \tag{25}$$

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

$$[t_1, t_2, t_3] \tag{26}$$

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

$$\mathcal{D}_{t_2,t_3;m}^{(\mathsf{M},n)} \triangleq \{\mathbf{y} \colon g(\mathbf{y}) = m\} \tag{27}$$

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

Lemma 11. *The pairwise Hamming distance vector of the weak flip code* $\mathscr{C}_{t_2,t_3}^{(\mathsf{M},n)}$ *for* $\mathsf{M} = 3$ *or* $\mathsf{M} = 4$ *is given as follows:*

$$\mathbf{d}\big(\mathscr{C}_{t_2,t_3}^{(3,n)}\big) = (t_2 + t_3, \ t_1 + t_3, \ t_1 + t_2), \tag{28}$$

$$\mathbf{d}\big(\mathscr{C}_{t_2,t_3}^{(4,n)}\big) = (t_2 + t_3, \ t_1 + t_3, \ t_1 + t_2, \ t_1 + t_2, \ t_1 + t_3, \ t_2 + t_3). \tag{29}$$

[3] Note that, as already discussed in Remark 5, the order of these columns does not matter when regarding the performance of the code. However, in order to make sure that the code is well-defined, we require here the order of the candidate columns to be exactly as given (i.e., all columns $\mathbf{c}_1^{(\mathsf{M})}$ together, then all $\mathbf{c}_2^{(\mathsf{M})}$ in the middle, and all $\mathbf{c}_3^{(\mathsf{M})}$ on the right of the codebook matrix). Thereby we also clearly and uniquely specify the codewords $\mathbf{x}_1, \ldots, \mathbf{x}_{\mathsf{M}}$.

5 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 simple example before we summarize our main results.

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

$$\mathscr{C}_{1,0}^{(4,4)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \quad \mathscr{C}_{2,0}^{(4,4)} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}. \tag{30}$$

We observe that while both codes are linear, the first code has a minimum Hamming distance 1, and the second has a minimum Hamming distance 2. Assuming an ML decoder, the average error probability as given in (15) can now be evaluated as $P_{\mathrm{e}}\big(\mathscr{C}_{1,0}^{(4,4)}\big) \approx 0.6112$ and $P_{\mathrm{e}}\big(\mathscr{C}_{2,0}^{(4,4)}\big) = 0.64$. Hence, even though the minimum Hamming distance of the first codebook is smaller, its overall performance is superior to the second codebook!

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

$$P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)*}\big) \leq P_{\mathrm{e}}\big(\mathscr{C}^{(\mathsf{M},n)}\big), \tag{31}$$

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

6 Analysis of the BAC

We start with results that hold for the general BAC. In this section we will restrict ourselves to two codewords $\mathsf{M} = 2$.

6.1 The Optimal Decision Rule for Flip Codes

Having only two codewords, the ML decision rule can be expressed using the *log-likelihood ratio (LLR)*. For the flip code of type t, $\mathscr{C}_t^{(2,n)}$, in Definition 7, the LLR is given as follows:

$$\begin{aligned}
&\log\left(\frac{P_{Y|X}^n(\mathbf{y}|\mathbf{x}_1)}{P_{Y|X}^n(\mathbf{y}|\mathbf{x}_2)}\right) \\
&= \log\left(\frac{(1-\epsilon_0)^n \left(\frac{\epsilon_0}{1-\epsilon_0}\right)^{d_{01}(\mathbf{x}_1,\mathbf{y})} \left(\frac{\epsilon_1}{1-\epsilon_0}\right)^{d_{10}(\mathbf{x}_1,\mathbf{y})} \left(\frac{1-\epsilon_1}{1-\epsilon_0}\right)^{d_{11}(\mathbf{x}_1,\mathbf{y})}}{(1-\epsilon_0)^n \left(\frac{\epsilon_0}{1-\epsilon_0}\right)^{d_{01}(\mathbf{x}_2,\mathbf{y})} \left(\frac{\epsilon_1}{1-\epsilon_0}\right)^{d_{10}(\mathbf{x}_2,\mathbf{y})} \left(\frac{1-\epsilon_1}{1-\epsilon_0}\right)^{d_{11}(\mathbf{x}_2,\mathbf{y})}}\right) && (32) \\
&= \log\left(\frac{\left(\frac{\epsilon_0}{1-\epsilon_0}\right)^{d_{01}(\mathbf{x}_1,\mathbf{y})} \left(\frac{\epsilon_1}{1-\epsilon_0}\right)^{d_{10}(\mathbf{x}_1,\mathbf{y})} \left(\frac{1-\epsilon_1}{1-\epsilon_0}\right)^{t-d_{10}(\mathbf{x}_1,\mathbf{y})}}{\left(\frac{\epsilon_0}{1-\epsilon_0}\right)^{t-d_{10}(\mathbf{x}_1,\mathbf{y})} \left(\frac{\epsilon_1}{1-\epsilon_0}\right)^{n-t-d_{01}(\mathbf{x}_1,\mathbf{y})} \left(\frac{1-\epsilon_1}{1-\epsilon_0}\right)^{d_{01}(\mathbf{x}_1,\mathbf{y})}}\right) && (33) \\
&= \big(t - d_{01}(\mathbf{x}_1,\mathbf{y}) - d_{10}(\mathbf{x}_1,\mathbf{y})\big) \log\left(\frac{1-\epsilon_1}{\epsilon_0}\right) \\
&\quad + \big(n - t - d_{01}(\mathbf{x}_1,\mathbf{y}) - d_{10}(\mathbf{x}_1,\mathbf{y})\big) \log\left(\frac{1-\epsilon_0}{\epsilon_1}\right) && (34) \\
&= (t-d)\log\left(\frac{1-\epsilon_1}{\epsilon_0}\right) + (n-t-d)\log\left(\frac{1-\epsilon_0}{\epsilon_1}\right) && (35) \\
&\triangleq \mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d), && (36)
\end{aligned}$$

where we have defined

$$d \triangleq d_{01}(\mathbf{x}_1, \mathbf{y}) + d_{10}(\mathbf{x}_1, \mathbf{y}) = d_{\mathrm{H}}(\mathbf{x}_1, \mathbf{y}) \tag{37}$$

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

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

$$\mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, d) \begin{cases} \geq 0 & \Longrightarrow \ g(\mathbf{y}) = 1, \\ < 0 & \Longrightarrow \ g(\mathbf{y}) = 2. \end{cases} \tag{38}$$

Recall that ϵ_0 and ϵ_1 are parameters describing the channel (BAC), t and n describe the codebook (flip code $\mathscr{C}_t^{(2,n)}$), and $0 \leq d \leq n$ describes the received vector $\mathbf{y}$ (with respect to the first codeword). As an example, the log-likelihood ratio $\mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, d)$ is depicted in Figure 5 as a function of ϵ_0 (with $\epsilon_1 = 1 - 2\epsilon_0$) for the flip code $\mathscr{C}_1^{(2,n)}$ in the cases of $n = 6$ and $n = 7$.

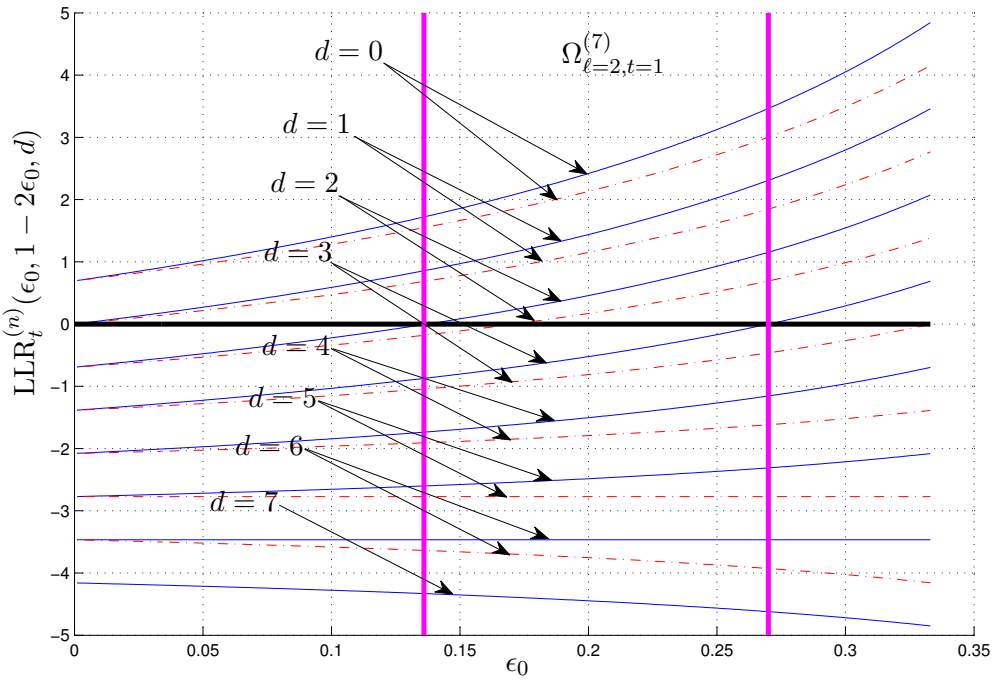

Figure 5: The log-likelihood ratio $\mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1 = 1 - 2\epsilon_0, d)$ for $\mathscr{C}_1^{(2,n)}$ (i.e., $t = 1$) as a function of ϵ_0 for different values of d. The solid blue lines correspond to $n = 7$, the dashed red lines to $n = 6$. We see that for $n = 7$ and $\epsilon_0 \in [0.136, 0.270]$ (i.e., the region between the two vertical purple lines), the threshold for the optimal ML decision rule (see Corollary 13) is $\ell = 2$.

We next state some important properties of $\mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, d)$.

Proposition 12 (Properties of $\mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, d)$).

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

$$\mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, d) \leq \mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, d-1), \quad \forall\, 1 \leq d \leq n. \tag{39}$$

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

$$\mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, d) \begin{cases} \geq 0 & \text{if } 0 \leq d \leq t, \\ \lesseqgtr 0 & \text{if } t < d \leq \left\lfloor \frac{n}{2} \right\rfloor, \text{ depending on } \epsilon_0 \text{ and } \epsilon_1, \\ \leq 0 & \text{if } \left\lfloor \frac{n}{2} \right\rfloor < d \leq n. \end{cases} \tag{40}$$

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

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

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

$$\mathrm{LLR}_t^{(n+1)}(\epsilon_0, \epsilon_1, d+1) < \mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, d). \tag{41}$$

Proof. Omitted. □

From Proposition 12 now follows directly that the ML decision rule for a flip code is a *threshold rule*.

Corollary 13 (Threshold Rule). *For every given flip code $\mathscr{C}_t^{(2,n)}$ and every BAC $(\epsilon_0, \epsilon_1) \in \Omega$, there exists a* ***threshold*** *ℓ, $t \leq \ell \leq \left\lfloor \frac{n-1}{2} \right\rfloor$, such that the ML decision rule can be stated as*

$$g(\mathbf{y}) = \begin{cases} 1 & \text{if } 0 \leq d \leq \ell, \\ 2 & \text{if } \ell + 1 \leq d \leq n. \end{cases} \tag{42}$$

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

$$\Omega_{\ell,t}^{(n)} \triangleq \left\{(\epsilon_0, \epsilon_1) \colon \mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, \ell) \geq 0 \text{ and } \mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1, \ell+1) \leq 0\right\}. \tag{43}$$

6.2 Optimal Codes

Theorem 14. *Consider a BAC and a blocklength n. Then, irrespective of the channel parameters ϵ_0 and ϵ_1, there exists a choice of t, $0 \leq t \leq \left\lfloor \frac{n}{2} \right\rfloor$, such that the flip code of type t, $\mathscr{C}_t^{(2,n)}$, is optimal in the sense that it minimizes the average error probability.*

Proof. See Appendix A.1. □

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, note that the theorem does not exclude the possibility that another code might exist that also is optimal and that has an identical symbol in both codewords at a given position.

We would like to point out that the exact choice of t is not obvious and depends strongly on n, ϵ_0, and ϵ_1. As an example, the optimal choices of t are shown in Figure 6 for $n = 7$. We see that depending on the channel parameters, the optimal value of t changes. Note that for a completely noisy channel ($\epsilon_1 = 1 - \epsilon_0$), the choice of t is irrelevant since the probability of error is $\frac{1}{2}$ for any code. Moreover, in Theorem 24 it will be shown that the flip codes are optimal on the BSC for any choice of t.

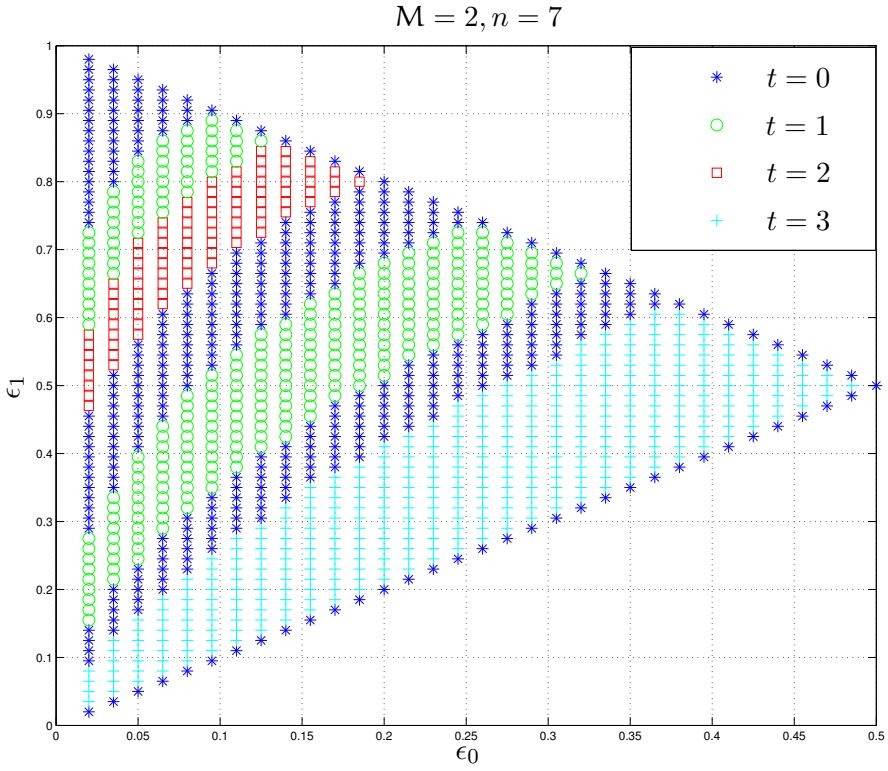

Figure 6: Optimal codebooks on a BAC: the optimal choice of the parameter t for different values of ϵ_0 and ϵ_1 for a fixed blocklength $n = 7$.

6.3 Optimal Codes for a Fixed Decision Rule

Our original goal was to find the optimal code for a given channel (ϵ_0, ϵ_1). We have shown that this is equivalent in finding an optimal t. Unfortunately, this search is difficult because the borders between the regions of different optimal t (see, e.g., Figure 6) are defined by the combined influences of two different forces: when varying (ϵ_0, ϵ_1) either the optimal code $\mathscr{C}_t^{(2,n)}$ changes, but the optimal threshold ℓ remains the same, or the optimal choice of ℓ changes, too. Hence, a joint optimization of t and ℓ is necessary.

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

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

Theorem 15. *Fix a blocklength n, a code parameter $0 \le t \le \lfloor \frac{n}{2} \rfloor$, and a decision rule threshold ℓ. Then the roots (ϵ_0, ϵ_1) of*

$$2P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_t^{(2,n)}\big) - 2P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_{t+1}^{(2,n)}\big) = 0 \tag{44}$$

are identical to the roots of

$$\mathrm{LLR}_t^{(n-1)}(\epsilon_0, \epsilon_1, \ell) = 0, \tag{45}$$

where $P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_t^{(2,n)}\big)$ *denotes the error probability of code* $\mathscr{C}_t^{(2,n)}$ *decoded under the decision threshold* ℓ*. Moreover, for a fixed* $\epsilon_0 \in \Omega$*, there exists at most one* $\epsilon_1 \in \Omega$ *such that* (44) *holds; and for a fixed* $\epsilon_1 \in \Omega$*, there exists at most one* $\epsilon_0 \in \Omega$ *such that* (44) *holds.*

Proof. See Appendix A.2. □

Using Theorem 15 and Proposition 12, we can now state conditions on t such that $\mathscr{C}_t^{(2,n)}$ is optimal under a fixed decision rule ℓ.

Corollary 16. *Fix a blocklength* n *and a decision rule* ℓ*. Then the flip code of type* t*,* $\mathscr{C}_t^{(2,n)}$*, is optimal for a fixed decision rule* ℓ *if* (ϵ_0, ϵ_1) *belongs to*

$$\Big\{(\epsilon_0,\epsilon_1)\colon \mathrm{LLR}_t^{(n-1)}(\epsilon_0,\epsilon_1,\ell) > 0 \text{ and } \mathrm{LLR}_{t-1}^{(n-1)}(\epsilon_0,\epsilon_1,\ell) < 0\Big\}. \tag{46}$$

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

Proof. From (107) in the proof of Theorem 15 in Appendix A.2 and from assumption (1) it follows that if

$$\mathrm{LLR}_t^{(n-1)}(\epsilon_0,\epsilon_1,\ell) > 0, \tag{47}$$

then

$$P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_t^{(2,n)}\big) < P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_{t+1}^{(2,n)}\big). \tag{48}$$

As we know from Proposition 12 that $\mathrm{LLR}_t^{(n-1)}(\epsilon_0,\epsilon_1,\ell)$ is increasing in t, this means that if both (47) and

$$\mathrm{LLR}_{t-1}^{(n-1)}(\epsilon_0,\epsilon_1,\ell) < 0 \tag{49}$$

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

We illustrate Corollary 16 by an example. We fix $n = 7$, $\ell = 2$, $\epsilon_1 = 0.5$, and let ϵ_0 increase from 0 to $\min\{\epsilon_1, 1-\epsilon_1\} = 0.5$, see Figure 7. Starting with $t = 3$, we check that

$$\mathrm{LLR}_2^{(6)}(\epsilon_0, 0.5, 2) > 0, \tag{50}$$

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

$$\mathrm{LLR}_1^{(6)}(\epsilon_0, 0.5, 2) < 0 \tag{51}$$

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

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

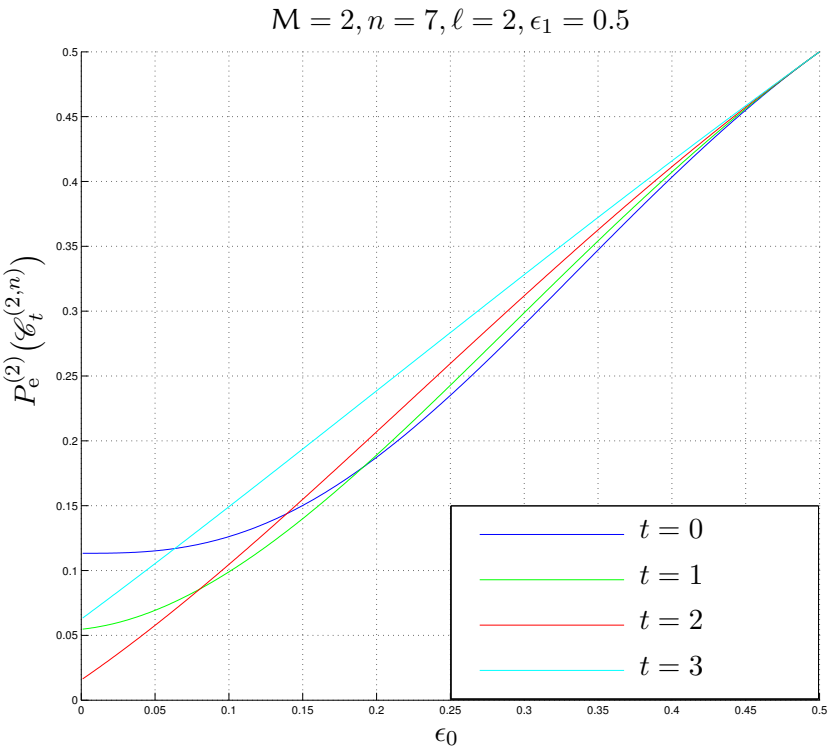

Figure 7: The error probabilities of all possible flip codes $\mathscr{C}_t^{(2,n)}$ as a function of the channel parameter ϵ_0, for a fixed blocklength $n = 7$, $\epsilon_1 = 0.5$, and a fixed decision rule $\ell = 2$. For any ϵ_0, the optimal code is the one with the smallest error probability value.

In Figure 7 the error probabilities of the various flip codes are shown as a function of ϵ_0. The optimal choices of t for all values of $(\epsilon_0, \epsilon_1) \in \Omega$ for $n = 7$ and $\ell = 2$ are shown in Figure 8.

Corollary 16 shows that for a fixed decision rule ℓ, the choice of the optimal code parameter t depending on the given parameters n, ϵ_0, and ϵ_1 is much easier than the choice of the jointly optimal t and ℓ for a globally optimal code. In particular, we have the following regular structure.

Corollary 17. *Fix a blocklength n and a decision rule ℓ, and consider a BAC. If we increase ϵ_0 or decrease ϵ_1, then the optimal value of t is nonincreasing.*

More sloppily we can say that when we are moving inside of Ω (see Figure 2) to the right or downwards, the optimal t will either remain the same or be reduced by 1. This means that the picture of the regions of optimal codes is much more regular without seemingly random jumps between different t. For an illustration compare the optimal codes for a fixed decision rule $\ell = 2$ in Figure 8 with the corresponding globally optimal regions of Figure 6.

Even more importantly, Theorem 15 also allows us to locate the exact location of some of the boundaries between the different areas of *globally optimal* codes (Figure 6)!

Corollary 18. *Consider the boundary between two areas of globally optimal codes (as, e.g., shown in Figure 6). If the optimal decision rule on both sides of the*

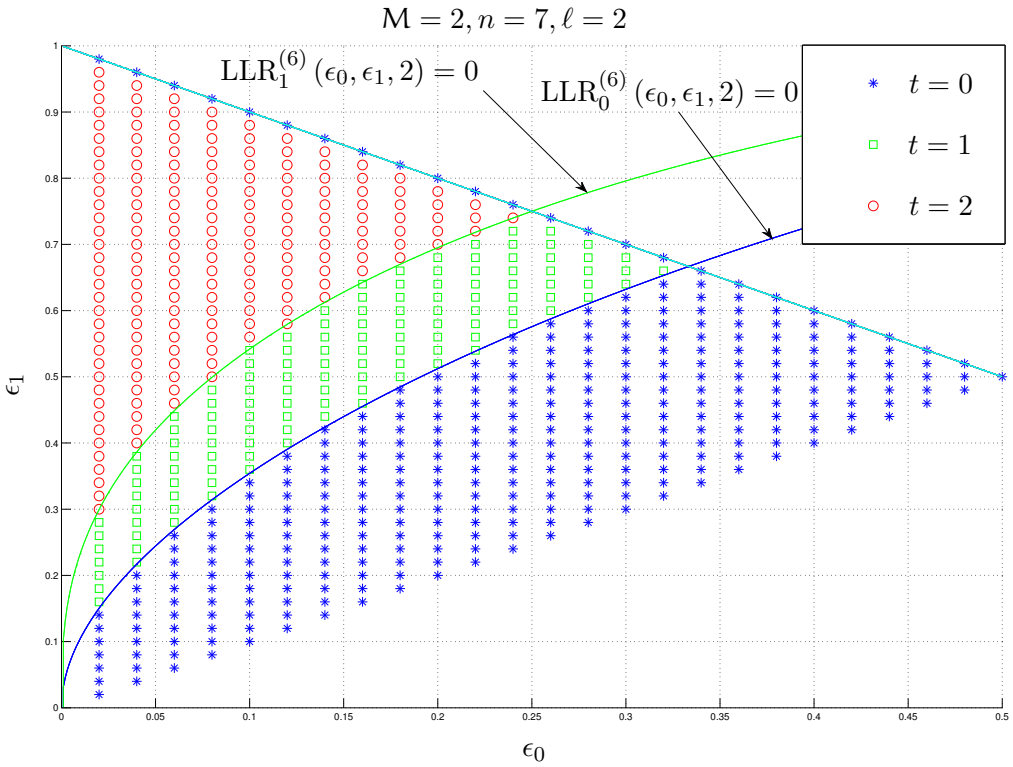

Figure 8: Optimal codebooks on a BAC for a fixed decision rule: for all possible (ϵ_0, ϵ_1) this plot shows the optimal choice of the code parameter t. The blocklength is $n = 7$ and the decision rule is $\ell = 2$.

boundary takes the same value ℓ and if the optimal code on the left is $t+1$, while the optimal code on the right is t, then this boundary is identical to a corresponding boundary in the situation with a fixed decision rule ℓ. In particular, this boundary is given by the roots of $\mathrm{LLR}_t^{(n-1)}(\epsilon_0, \epsilon_1, \ell)$.

We again show the example of $n = 7$ from Figure 6: in Figure 9 the same plot is shown including a boundary that is identical to a boundary given in Figure 8.

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

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

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

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

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

7 Analysis of the ZC

In this section we investigate the special case of a ZC more in detail.

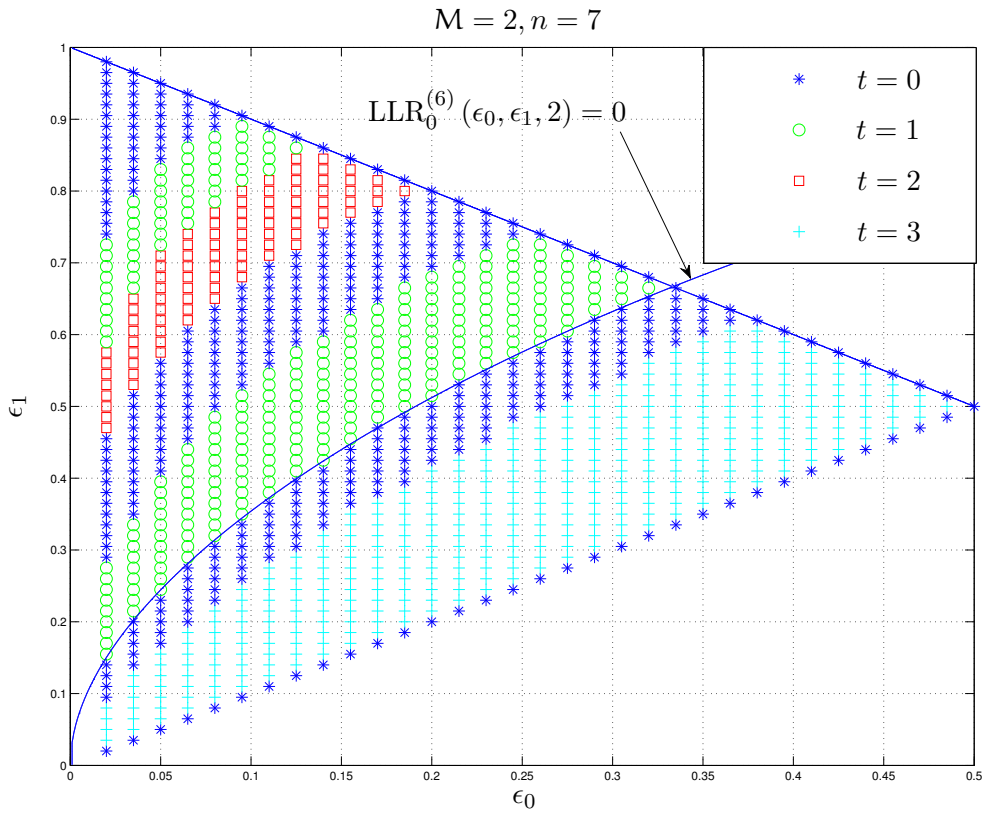

Figure 9: Globally optimal codebooks on a BAC for a blocklength $n = 7$ (identical to Figure 6). The shown boundary between $t = 1$ and $t = 0$ is identical to the corresponding boundary given in Figure 8, where we had assumed a fixed decision rule $\ell = 2$.

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

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

$$P_{\mathrm{e}}\big(\mathscr{C}_0^{(2,n)}\big) = \frac{1}{2}\epsilon_1^n. \tag{52}$$

Proof. Due to Theorem 14, we can restrict our search to flip codes of some type t, $\mathscr{C}_t^{(2,n)}$, i.e., $\mathbf{x}_2 = \bar{\mathbf{x}}$ is the flipped version of $\mathbf{x}_1 = \mathbf{x}$.

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

$$\min\big\{P_{Y|X}^n(\mathbf{y}|\mathbf{x}_1), P_{Y|X}^n(\mathbf{y}|\mathbf{x}_2)\big\} = \begin{cases} 0 & \text{if } \mathbf{y} \neq \mathbf{0}, \\ \epsilon_1^{\max\{w_{\mathrm{H}}(\mathbf{x}_1), w_{\mathrm{H}}(\mathbf{x}_2)\}} & \text{if } \mathbf{y} = \mathbf{0}. \end{cases} \tag{53}$$

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

We see that the optimal code is actually linear. Moreover, from the proof it also follows that it is the unique optimal code design.

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

We start with two auxiliary results about some weak flip codes on a ZC.

Lemma 20. *Consider the weak flip codes of type*[4] $(t, 0)$, $\mathscr{C}_{t,0}^{(\mathsf{M},n)}$, *for* $1 \le t \le \lfloor \frac{n}{2} \rfloor$. *Then for a ZC and for any* $n \ge 2$, *the optimal decoding regions* $\mathcal{D}_{t,0;m}^{(\mathsf{M},n)}$ *for* $\mathscr{C}_{t,0}^{(\mathsf{M},n)}$ *with three codewords* $\mathsf{M} = 3$ *or four codewords* $\mathsf{M} = 4$ *are*

$$\mathcal{D}_{t,0;1}^{(\mathsf{M},n)} = \{\mathbf{0}\}, \tag{54}$$

$$\mathcal{D}_{t,0;2}^{(\mathsf{M},n)} = \big\{\mathbf{y} \colon \mathbf{y} = (\underbrace{0 \ \cdots \ 0}_{\substack{n-t \\ \text{comp.}}} \ \underbrace{y_{n-t+1} \ \cdots \ y_n}_{t \text{ comp.}}) \text{ with } 1 \le w_{\mathrm{H}}(\mathbf{y}) \le t\big\}, \tag{55}$$

$$\mathcal{D}_{t,0;3}^{(\mathsf{M},n)} = \big\{\mathbf{y} \colon \mathbf{y} = (\underbrace{y_1 \ \cdots \ y_{n-t}}_{n-t \text{ comp.}} \ \underbrace{0 \ \cdots \ 0}_{t \text{ comp.}}) \text{ with } 1 \le w_{\mathrm{H}}(\mathbf{y}) \le n - t\big\}, \tag{56}$$

$$\mathcal{D}_{t,0;4}^{(4,n)} = \{0,1\}^n \setminus \bigcup_{m=1}^{3} \mathcal{D}_{t,0;m}^{(4,n)}. \tag{57}$$

The corresponding average success probabilities are

$$3P_{\mathrm{c}}\big(\mathscr{C}_{t,0}^{(3,n)}\big) = 1 + \sum_{d=0}^{t-1} \binom{t}{d} (1-\epsilon_1)^{t-d} \cdot \epsilon_1^d + \sum_{d=0}^{n-t-1} \binom{n-t}{d} (1-\epsilon_1)^{n-t-d} \cdot \epsilon_1^d; \tag{58}$$

$$\begin{aligned} 4P_{\mathrm{c}}\big(\mathscr{C}_{t,0}^{(4,n)}\big) = 1 &+ \sum_{d=0}^{t-1} \binom{t}{d} (1-\epsilon_1)^{t-d} \cdot \epsilon_1^d + \sum_{d=0}^{n-t-1} \binom{n-t}{d} (1-\epsilon_1)^{n-t-d} \cdot \epsilon_1^d \\ &+ \sum_{d=0}^{n-1} \left(\binom{n}{d} - \binom{n-t}{d-t} - \binom{t}{d-(n-t)} \right) (1-\epsilon_1)^{n-d} \cdot \epsilon_1^d. \end{aligned} \tag{59}$$

Proof. See Appendix B.1. □

Note that even though the space of received n-vectors $\mathbf{y}$ is identical for both $\mathsf{M} = 3$ and $\mathsf{M} = 4$, the code $\mathscr{C}_{t,0}^{(3,n)}$ does not contain the all-one codeword. Therefore, if we use $\mathscr{C}_{t,0}^{(3,n)}$, all received sequences in $\mathcal{D}_{t,0;4}^{(4,n)}$ have zero probability of occurring. We therefore do not need to include them into any decoding region.

Lemma 21. *For a ZC, for any* $n \ge 2$, *and for* $1 \le t \le \lfloor \frac{n}{2} \rfloor$, *consider the weak flip code of type* $(t, 0)$ *with four codewords* $\mathsf{M} = 4$, $\mathscr{C}_{t,0}^{(4,n)}$. *If we append a new column to create a new code of length* $n+1$, *an optimal (in the sense of resulting in the largest average success probability) choice among all possible* $2^4 = 16$ *columns is* $\mathbf{c}_2^{(4)}$. *If* $t < \lfloor \frac{n}{2} \rfloor$, *or if* n *is odd and* $t = \lfloor \frac{n}{2} \rfloor$, *then this choice is unique.*

If we consider only three codewords $\mathsf{M} = 3$, *i.e., if we consider* $\mathscr{C}_{t,0}^{(3,n)}$, *then appending* $\mathbf{c}_2^{(3)}$ *or* $\mathbf{c}_3^{(3)}$ *are equally optimal.*

Proof. See Appendix B.2. □

We are now ready for the main result of this section.

Theorem 22. *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* $(t^*, 0)$, $\mathscr{C}_{t^*,0}^{(\mathsf{M},n)}$, *with*

$$t^* \triangleq \left\lfloor \frac{n}{2} \right\rfloor. \tag{60}$$

[4] Note that here we only consider weak flip codes that do not use the candidate column $\mathbf{c}_3^{(\mathsf{M})}$.

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 choices of codewords. The optimal code is

$$\mathscr{C}_{1,0}^{(4,2)} = \begin{pmatrix} \mathbf{c}_1^{(4)} & \mathbf{c}_2^{(4)} \end{pmatrix}. \tag{61}$$

Next assume that for blocklength n, $\mathscr{C}_{\lfloor \frac{n}{2} \rfloor,0}^{(4,n)}$ is optimal. From Lemma 21 we know that it is optimal to add column $\mathbf{c}_2^{(4)}$. Now note that for n even with $t = \frac{n}{2}$, adding the column $\mathbf{c}_2^{(4)}$ to the code $\mathscr{C}_{\frac{n}{2},0}^{(4,n)}$ will result in a code that is equivalent to $\mathscr{C}_{\lfloor \frac{n+1}{2} \rfloor,0}^{(4,n+1)}$ by exchanging the roles of the second and third codeword and re-ordering the columns. For n odd with $t = \lfloor \frac{n}{2} \rfloor$, adding the column $\mathbf{c}_2^{(4)}$ to the code $\mathscr{C}_{\lfloor \frac{n}{2} \rfloor,0}^{(4,n)}$ results in $\mathscr{C}_{\frac{n+1}{2},0}^{(4,n+1)}$.

Hence, we see that $\mathscr{C}_{\lfloor \frac{n+1}{2} \rfloor,0}^{(4,n+1)}$ is still optimal. The claim now follows by induction.

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

$$\begin{pmatrix} \mathbf{c}_1^{(3)} & \mathbf{c}_3^{(3)} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \mathbf{c}_1^{(3)} & \mathbf{c}_2^{(3)} \end{pmatrix} \tag{62}$$

are optimal codebooks for $n = 2$. An optimal way of extending these codes according to Lemma 21 is to add columns $\mathbf{c}_2^{(3)}$ or $\mathbf{c}_3^{(3)}$.

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

Similar to the case of $\mathsf{M} = 2$, we see that for $\mathsf{M} = 4$ the optimal code given in Theorem 22 is linear. Also note that from Lemma 21 it follows that for even n, these linear codes are the unique optimal codes, while for odd n there are other (also nonlinear) designs that achieve the same optimal performance.

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

$$\mathscr{C}_{t,0}^{(4,n)} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{x} \\ \bar{\mathbf{x}} \\ \mathbf{1} \end{pmatrix} \tag{63}$$

with $\mathbf{x}$ and $\bar{\mathbf{x}}$ defined in (22).

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

From the recursive proof-technique that we have used in the derivation of Theorem 22 and that is based on the addition of columns to the codebook matrix, it immediately follows that our optimal codes can be constructed recursively in n. Concretely, we have the following corollary.

Corollary 23. *The optimal codebooks defined in Theorem 22 for* $\mathsf{M} = 3$ *and* $\mathsf{M} = 4$ *can be constructed recursively in blocklength* n*. We start with an optimal codebook for* $n = 2$*:*

$$\mathscr{C}_{\mathrm{ZC}}^{(\mathsf{M},2)*} = \begin{pmatrix} \mathbf{c}_1^{(\mathsf{M})} & \mathbf{c}_2^{(\mathsf{M})} \end{pmatrix}. \tag{64}$$

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

$$\begin{cases} \mathbf{c}_1^{(\mathsf{M})} & \text{if } n \bmod 2 = 1, \\ \mathbf{c}_2^{(\mathsf{M})} & \text{if } n \bmod 2 = 0. \end{cases} \tag{65}$$

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

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

We define the following five column vectors (all are weak flip columns!):

$$\left\{ \mathbf{c}_1^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \mathbf{c}_2^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \mathbf{c}_3^{(5)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \mathbf{c}_4^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \mathbf{c}_5^{(5)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \right\}. \tag{66}$$

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

$$\mathscr{C}_{\mathrm{ZC}}^{(5,8)*} = \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_2^{(5)} & \mathbf{c}_3^{(5)} & \mathbf{c}_1^{(5)} & \mathbf{c}_2^{(5)} & \mathbf{c}_3^{(5)} & \mathbf{c}_4^{(5)} & \mathbf{c}_5^{(5)} \end{pmatrix}. \tag{67}$$

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

$$\begin{cases} \begin{pmatrix} \mathbf{c}_4^{(5)} & \mathbf{c}_5^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 0, \\ \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_2^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 2, \\ \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_3^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 4, \\ \begin{pmatrix} \mathbf{c}_3^{(5)} & \mathbf{c}_4^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 6, \\ \begin{pmatrix} \mathbf{c}_2^{(5)} & \mathbf{c}_5^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 8. \end{cases} \tag{68}$$

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

$$\mathscr{C}_{\mathrm{ZC}}^{(5,9)*} = \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_2^{(5)} & \mathbf{c}_3^{(5)} & \mathbf{c}_4^{(5)} & \mathbf{c}_5^{(5)} & \mathbf{c}_1^{(5)} & \mathbf{c}_2^{(5)} & \mathbf{c}_1^{(5)} & \mathbf{c}_3^{(5)} \end{pmatrix} \tag{69}$$

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

$$\begin{cases} \begin{pmatrix} \mathbf{c}_3^{(5)} & \mathbf{c}_4^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 1, \\ \begin{pmatrix} \mathbf{c}_2^{(5)} & \mathbf{c}_5^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 3, \\ \begin{pmatrix} \mathbf{c}_4^{(5)} & \mathbf{c}_5^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 5, \\ \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_2^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 7, \\ \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_3^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 9. \end{cases} \tag{70}$$

Note that the recursive structure in (68) and (70) 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 (66) exactly twice. For $n < 10$ the optimal code structure goes through some transient states.

8 Analysis of the BSC

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

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

Proof. From Theorem 14 we already know that there must exist a flip code that is optimal. Moreover, Theorem 14 also shows that the all-zero and the all-one column in a codebook matrix is strictly suboptimal. So, we only have two possible choices of candidate columns: $(0\ 1)^\mathsf{T}$ and $(1\ 0)^\mathsf{T}$. However, by the symmetry of a BSC, both columns will result in an identical performance. Therefore every flip code has the same performance, i.e., all of them must be optimal. □

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

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^*), $\mathscr{C}_{t_2^*, t_3^*}^{(\mathsf{M},n)}$, where*

$$t_2^* \triangleq \left\lfloor \frac{n-1}{3} \right\rfloor, \quad t_3^* \triangleq \left\lfloor \frac{n+1}{3} \right\rfloor. \tag{71}$$

Using the shorthand

$$k \triangleq \left\lfloor \frac{n}{3} \right\rfloor, \tag{72}$$

the code parameters of these optimal codes can be written as

$$[t_1^*, t_2^*, t_3^*] = \begin{cases} [k+1, k-1, k] & \text{if } n \bmod 3 = 0, \\ [k+1, k, k] & \text{if } n \bmod 3 = 1, \\ [k+1, k, k+1] & \text{if } n \bmod 3 = 2. \end{cases} \tag{73}$$

Proof. See Appendix C.1. □

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 the 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.

We also would like to point out the regularity of the optimal code design in Theorem 25 that repeats in n with a period of 3. For $\mathsf{M} = 5$, we expect a similar behavior, but with a period that is larger than 3.

Moreover, a closer inspection of the proof of Theorem 25 shows that when $\mathsf{M} = 3$, the received vector $\mathbf{y}$ farthest from the three codewords is

$$\mathbf{y} = (\underbrace{1 \cdots 1}_{t_1^*}\ \underbrace{1 \cdots 1}_{t_2^*}\ \underbrace{0 \cdots 0}_{t_3^*}), \tag{74}$$

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

Again, the proof of Theorem 25 implicitly proves that the given optimal codes can be constructed recursively in n.

Corollary 26. *The optimal codebooks defined in Theorem 25 for $\mathsf{M} = 3$ and $\mathsf{M} = 4$ can be constructed recursively in the blocklength n. We start with an optimal codebook for $n = 2$:*

$$\mathscr{C}_{\mathrm{BSC}}^{(\mathsf{M},2)*} = \begin{pmatrix} \mathbf{c}_1^{(\mathsf{M})} & \mathbf{c}_3^{(\mathsf{M})} \end{pmatrix}. \tag{75}$$

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

$$\begin{cases} \mathbf{c}_1^{(\mathsf{M})} & \text{if } n \bmod 3 = 0, \\ \mathbf{c}_2^{(\mathsf{M})} & \text{if } n \bmod 3 = 1, \\ \mathbf{c}_3^{(\mathsf{M})} & \text{if } n \bmod 3 = 2. \end{cases} \tag{76}$$

8.3 Pairwise Hamming Distance Structure

As already mentioned in Section 3.4, 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 [17]. 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![5]

We would like to come back to the example shown in Section 5 and further deepen our analysis of the minimum Hamming distance of our optimal codes on the very symmetric BSC. Although, as (16) shows, the error probability performance of a BSC is completely specified by the Hamming distance between codewords and received vectors, we will now demonstrate that a design based on the minimum Hamming distance can fail, even for the very symmetric BSC and even for linear codes. In the case of a more general (and not symmetric) BAC, this will be even more pronounced.

We compare the optimal codes given in Theorem 25 with the following different weak flip code $\mathscr{C}_{\mathrm{subopt}}^{(\mathsf{M},n)}$ with code parameters

$$[t_1, t_2, t_3] = \begin{cases} [k, k, k] & \text{if } n \bmod 3 = 0, \\ [k+1, k-1, k+1] & \text{if } n \bmod 3 = 1, \\ [k+2, k, k] & \text{if } n \bmod 3 = 2. \end{cases} \tag{77}$$

This code can be constructed from the optimal code $\mathscr{C}_{\mathrm{BSC}}^{(\mathsf{M},n-1)*}$ by appending a suboptimal column[6] and—based on a closer inspection of the proof of Theorem 25 and Corollary 26—can be shown to be strictly suboptimal.

Recalling Lemma 11, we compute the pairwise Hamming distance vector of the optimal code for $\mathsf{M} = 3$:

$$\mathbf{d}\big(\mathscr{C}_{\mathrm{BSC}}^{(3,n)*}\big) = \begin{cases} (2k-1, 2k+1, 2k) & \text{if } n \bmod 3 = 0, \\ (2k, 2k+1, 2k) & \text{if } n \bmod 3 = 1, \\ (2k+1, 2k+2, 2k+1) & \text{if } n \bmod 3 = 2, \end{cases} \tag{78}$$

i.e.,

$$d_{\min}\big(\mathscr{C}_{\mathrm{BSC}}^{(3,n)*}\big) = \begin{cases} 2k-1 & \text{if } n \bmod 3 = 0, \\ 2k & \text{if } n \bmod 3 = 1, \\ 2k+1 & \text{if } n \bmod 3 = 2. \end{cases} \tag{79}$$

[5] Interestingly, a subfamily of the weak flip codes can be shown to have many linear-like properties. For more details see [18].

[6] The choice of column depends on n.

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

$$\mathbf{d}\big(\mathscr{C}_{\mathrm{BSC}}^{(4,n)*}\big) = \begin{cases} (2k-1, 2k+1, 2k, 2k, 2k+1, 2k-1) & \text{if } n \bmod 3 = 0, \\ (2k, 2k+1, 2k, 2k, 2k+1, 2k) & \text{if } n \bmod 3 = 1, \\ (2k+1, 2k+2, 2k+1, 2k+1, 2k+2, 2k+1) & \text{if } n \bmod 3 = 2, \end{cases} \tag{80}$$

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

We compare this with the suboptimal code (77). For $\mathsf{M} = 3$:

$$\mathbf{d}\big(\mathscr{C}_{\mathrm{subopt}}^{(3,n)}\big) = \begin{cases} (2k, 2k, 2k) & \text{if } n \bmod 3 = 0, \\ (2k, 2k+2, 2k) & \text{if } n \bmod 3 = 1, \\ (2k, 2k+2, 2k+2) & \text{if } n \bmod 3 = 2, \end{cases} \tag{81}$$

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

$$\mathbf{d}\big(\mathscr{C}_{\mathrm{subopt}}^{(4,n)}\big) = \begin{cases} (2k, 2k, 2k, 2k, 2k, 2k) & \text{if } n \bmod 3 = 0, \\ (2k, 2k+2, 2k, 2k, 2k+2, 2k) & \text{if } n \bmod 3 = 1, \\ (2k, 2k+2, 2k+2, 2k+2, 2k+2, 2k) & \text{if } n \bmod 3 = 2, \end{cases} \tag{82}$$

with also $d_{\min}\big(\mathscr{C}_{\mathrm{subopt}}^{(3,n)}\big) = 2k$ in all cases.

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

By adapting the construction of the strictly suboptimal code $\mathscr{C}_{\mathrm{subopt}}^{(\mathsf{M},n)}$, 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* $d_{\min}\big(\mathscr{C}^{(\mathsf{M},n)}\big)$ *can be strictly suboptimal. This is not true in the case of* $n \bmod 3 = 2$*.*

As a matter of fact, using a result from [18], one can show that on a BSC for $\mathsf{M} = 3$ or $\mathsf{M} = 4$ and in the case of $n \bmod 3 = 0$, all codes that maximize the minimum Hamming distance are strictly suboptimal.

9 Conclusion

We have studied optimal code design of ultra-small block-codes for the most general binary discrete memoryless channel, the so-called *binary asymmetric channel (BAC)*. For an arbitrary finite blocklength n, we have analyzed the structure of optimal codes with two codewords.

We then have put special emphasis on the two most important special cases of binary channels, the *Z-channel (ZC)* and the *binary symmetric channel (BSC)*. There, again for an arbitrary finite blocklength n, we have derived an optimal code design with four or less messages. In the case of the ZC, we have also conjectured an optimal code design with five messages.

We have introduced a new way of generating these codes recursively by using a column-wise build-up of the codebook matrix. This column view of the codebook turns out to be far more powerful for analysis than the standard row-wise view (i.e.,

the analysis based on the codewords). We believe that the recursive construction of codes may be extended to a higher number of codewords and also to more complex channel models. Indeed, we have achieved some first promising results for the binary erasure channel (BEC) [18]. Note, however, that in these more complex situations we might need a recursion from n to $n+\gamma$ with a step-size $\gamma > 1$.

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

Finally, we would like to point out that the family of weak flip codes defined in Section 4 (and in particular a subfamily called *fair weak flip codes* [18]) turn out to have many interesting properties. A first closer investigation of some of these properties and these codes' relation to linear codes can be found in [18].

A Derivations concerning the BAC

A.1 Proof of Theorem 14

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

We therefore only need to show that increasing n will always allow us to find a new flip code with a better performance. In other words, Theorem 14 is proven once we have shown that

$$P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n-1)}\big) \geq \max\left\{P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n)}\big), P_{\mathrm{e}}\big(\mathscr{C}_{t+1}^{(2,n)}\big)\right\}. \tag{83}$$

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

$$\mathscr{C}_t^{(2,n-1)} = \begin{pmatrix} \mathbf{x}_1^{(n-1)} \\ \mathbf{x}_2^{(n-1)} \end{pmatrix} \tag{84}$$

we can derive two nontrivial length-n codes:

$$\mathscr{C}_t^{(2,n)} = \begin{pmatrix} [\mathbf{x}_1^{(n-1)} \ 0] \\ [\mathbf{x}_2^{(n-1)} \ 1] \end{pmatrix}, \quad \mathscr{C}_{t+1}^{(2,n)} = \begin{pmatrix} [\mathbf{x}_1^{(n-1)} \ 1] \\ [\mathbf{x}_2^{(n-1)} \ 0] \end{pmatrix}. \tag{85}$$

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

Since in the following we are going to compare different flip codes of either length $n-1$ or n, we need to clarify our notation. So for the received vectors $\mathbf{y}^{(n)}$ we use a superscript (n) to denote their length, and for the codewords $\mathbf{x}_m^{(n)}$, optimal decoding threshold $\ell^{(n)}$, and the Hamming distance $d^{(n)}$ between a received sequence and the first codeword we use the superscript (n) to denote their affiliation with the corresponding code of length n. Hence, as shown in Corollary 13, the optimal ML decision rule for $\mathscr{C}_t^{(2,n)}$ can be expressed as

$$g(\mathbf{y}) = \begin{cases} 1 & \text{if } 0 \leq d^{(n)} \leq \ell^{(n)}, \\ 2 & \text{if } \ell^{(n)}+1 \leq d^{(n)} \leq n. \end{cases} \tag{86}$$

The threshold satisfies $0 \leq \ell^{(n)} \leq \left\lfloor \frac{n-1}{2} \right\rfloor$. Note that when $\ell^{(n)} = \left\lfloor \frac{n-1}{2} \right\rfloor$, the decision rule is equivalent to a majority rule. Also note that when n is even and $d^{(n)} = \frac{n}{2}$,

the decisions for $\mathbf{x}_1^{(n)}$ and $\mathbf{x}_2^{(n)}$ are equally likely, i.e., without loss of generality we then always decode to $\mathbf{x}_2^{(n)}$.

So let the threshold for $\mathscr{C}_t^{(2,n-1)}$ be $\ell^{(n-1)}$. We will now argue that the threshold for $\mathscr{C}_t^{(2,n)}$ and $\mathscr{C}_{t+1}^{(2,n)}$ (compare with (85)) must satisfy

$$\ell^{(n)} = \ell^{(n-1)} \quad \text{or} \quad \ell^{(n)} = \ell^{(n-1)} + 1. \tag{87}$$

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

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

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

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

$$P_{\mathrm{e}} = \frac{1}{2} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})=2}} P_{Y|X}^n(\mathbf{y}|\mathbf{x}_1) + \frac{1}{2} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})=1}} P_{Y|X}^n(\mathbf{y}|\mathbf{x}_2). \tag{88}$$

For $\mathscr{C}_t^{(2,n-1)}$, (88) can be written as follows:

$$\begin{aligned} 2P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n-1)}\big) = & \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+1 \leq d^{(n-1)} \leq n-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big) \\ & + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \leq d^{(n-1)} \leq \ell^{(n-1)}}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big) \qquad (89) \\ = & \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+1 \leq d^{(n-1)} \leq n-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big) P_{Y|X}(1|0) \\ & + \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+1 \leq d^{(n-1)} \leq n-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big) P_{Y|X}(0|0) \\ & + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \leq d^{(n-1)} \leq \ell^{(n-1)}}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big) P_{Y|X}(1|1) \\ & + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \leq d^{(n-1)} \leq \ell^{(n-1)}}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big) P_{Y|X}(0|1) \qquad (90) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+2 \le d^{(n)} \le n}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_1^{(n)}\Big) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+1 \le d^{(n)} \le n-1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 1 \le d^{(n)} \le \ell^{(n-1)}+1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \le d^{(n)} \le \ell^{(n-1)}}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_2^{(n)}\Big). \qquad (91)
\end{aligned}
$$

Here, in (90) 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 (91) we combine the terms together using the definition of $\mathscr{C}_t^{(2,n)}$ according to (22) (and (85)).

We can now distinguish the two cases (87):

(i) If the decision rule for $\mathscr{C}_t^{(2,n)}$ is unchanged, i.e., $\ell^{(n)} = \ell^{(n-1)}$, we only need to take care of the third summation in (91) that contains some terms that will now be decoded differently. We split this sum up into two parts:

$$
\begin{aligned}
&\sum_{\substack{\mathbf{y}^{(n-1)} \\ 1 \le d^{(n)} \le \ell^{(n-1)}+1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big) \\
&\quad = \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n)} = \ell^{(n-1)}+1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big) \\
&\qquad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 1 \le d^{(n)} \le \ell^{(n-1)}}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big). \qquad (92)
\end{aligned}
$$

Since we have assumed that $\ell^{(n)} = \ell^{(n-1)}$, we know that for all $\mathbf{y}^{(n-1)}$ with $d^{(n-1)} = \ell^{(n-1)}$ the length-n received vector $\big[\mathbf{y}^{(n-1)}\ 1\big]$ has $d^{(n)} = \ell^{(n-1)} + 1 = \ell^{(n)} + 1$ and will be decoded to $\mathbf{x}_2^{(n)}$. Hence we must have

$$
\frac{P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_1^{(n)}\Big)}{P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big)} \le 1. \qquad (93)
$$

Hence, we have

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

$$
\begin{aligned}
&+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 1 \le d^{(n)} \le \ell^{(n-1)}}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big) \\
&+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \le d^{(n)} \le \ell^{(n-1)}}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_2^{(n)}\Big) \qquad (94) \\
&= \sum_{\substack{\mathbf{y}^{(n)} \\ \ell^{(n-1)}+1 \le d^{(n)} \le n}} P_{Y|X}^n\big(\mathbf{y}^{(n)}\big|\mathbf{x}_1^{(n)}\big) \\
&+ \sum_{\substack{\mathbf{y}^{(n)} \\ 0 \le d^{(n)} \le \ell^{(n-1)}}} P_{Y|X}^n\big(\mathbf{y}^{(n)}\big|\mathbf{x}_2^{(n)}\big) \qquad (95) \\
&= 2P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n)}\big). \qquad (96)
\end{aligned}
$$

(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 (91) that contains some terms that will now be decoded differently. Again, we split this sum into two parts:

$$
\begin{aligned}
&\sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+1 \le d^{(n)} \le n-1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big) \\
&= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n)} = \ell^{(n-1)}+1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big) \\
&\quad + \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+2 \le d^{(n)} \le n-1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big). \qquad (97)
\end{aligned}
$$

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

$$
\frac{P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big)}{P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_2^{(n)}\Big)} \ge 1. \qquad (98)
$$

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

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

We remark that while in general $P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n-1)}\big) \ge P_{\mathrm{e}}\big(\mathscr{C}_t^{(2,n)}\big)$, we only achieve equality if n is even and $\ell^{(n-1)} = \big\lfloor \frac{n-1}{2} \big\rfloor$.

A.2 Proof of Theorem 15

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

Note that since we are using a similar approach as in Appendix A.1, we also apply the notation introduced there, i.e., we use a superscript (n) to denote length and affiliation.

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

$$
\begin{aligned}
2P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_t^{(2,n)}\big) = & \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell+1\le d^{(n-1)}\le n-1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\big[\mathbf{x}_1^{(n-1)}\ 0\big]\Big) \\
& + \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell\le d^{(n-1)}\le n-1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\big[\mathbf{x}_1^{(n-1)}\ 0\big]\Big) \\
& + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0\le d^{(n-1)}\le \ell}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\big[\mathbf{x}_2^{(n-1)}\ 1\big]\Big) \\
& + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0\le d^{(n-1)}\le \ell-1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\big[\mathbf{x}_2^{(n-1)}\ 1\big]\Big) \qquad (99) \\
= & \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell+1\le d^{(n-1)}\le n-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)(1-\epsilon_0+\epsilon_0) \\
& + \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)}=\ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\epsilon_0 \\
& + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0\le d^{(n-1)}\le \ell-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)(\epsilon_1+1-\epsilon_1) \\
& + \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)}=\ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)\epsilon_1. \qquad (100)
\end{aligned}
$$

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

$$
\begin{aligned}
2P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_{t+1}^{(2,n)}\big) = & \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell+1\le d^{(n-1)}\le n-1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\big[\mathbf{x}_1^{(n-1)}\ 1\big]\Big) \\
& + \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell\le d^{(n-1)}\le n-1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\big[\mathbf{x}_1^{(n-1)}\ 1\big]\Big) \\
& + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0\le d^{(n-1)}\le \ell}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\big[\mathbf{x}_2^{(n-1)}\ 0\big]\Big) \\
& + \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0\le d^{(n-1)}\le \ell-1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\big[\mathbf{x}_2^{(n-1)}\ 0\big]\Big) \qquad (101) \\
= & \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell+1\le d^{(n-1)}\le n-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)(1-\epsilon_1+\epsilon_1) \\
& + \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)}=\ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\epsilon_1
\end{aligned}
$$

$$+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \le d^{(n-1)} \le \ell-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)(\epsilon_0 + 1 - \epsilon_0)$$
$$+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)\epsilon_0. \tag{102}$$

Subtracting (102) from (100) then yields

$$\begin{aligned}
&2P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_t^{(2,n)}\big) - 2P_{\mathrm{e}}^{(\ell)}\big(\mathscr{C}_{t+1}^{(2,n)}\big) \\
&\quad= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\epsilon_0 + \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)\epsilon_1 \\
&\qquad- \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\epsilon_1 - \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)\epsilon_0 \qquad (103) \\
&\quad= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} \Big(P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big) - P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\Big)(\epsilon_1 - \epsilon_0) \qquad (104) \\
&\quad= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)\left(1 - \frac{P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)}{P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)}\right)(\epsilon_1 - \epsilon_0) \qquad (105) \\
&\quad= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)\Big(1 - e^{\mathrm{LLR}_t^{(n-1)}(\epsilon_0,\epsilon_1,\ell)}\Big)(\epsilon_1 - \epsilon_0) \qquad (106) \\
&\quad= \Big(1 - e^{\mathrm{LLR}_t^{(n-1)}(\epsilon_0,\epsilon_1,\ell)}\Big)(\epsilon_1 - \epsilon_0) \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big), \qquad (107)
\end{aligned}$$

where in (106) we make use of our assumption that $\mathscr{C}_t^{(2,n-1)}$ is decoded also using the same threshold ℓ.

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

$$\mathrm{LLR}_t^{(n-1)}(\epsilon_0, \epsilon_1, \ell) = 0. \tag{108}$$

From the definition of the log-likelihood ratio, we see that if we fix ϵ_0, then there exists at most one ϵ_1 such that (108) is satisfied. The same is true if we fix ϵ_1 and search for an ϵ_0.

B Derivations concerning the ZC

B.1 Proof of Lemma 20

We start with $\mathsf{M} = 4$. We consider the weak flip code of type $(t, 0)$, $\mathscr{C}_{t,0}^{(4,n)}$, where $1 \le t \le \lfloor \frac{n}{2} \rfloor$, and denote the decoding region of the mth codeword by $\mathcal{D}_{t,0;m}^{(4,n)}$. 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 versions of the first two.

From $P_{Y|X}(0|0) = 1$, the first codeword will always induce the received vector $\mathbf{y} = \mathbf{0}$; so, $\mathcal{D}_{t,0;1}^{(4,n)}$ consists only of the all-zero vector.

Next note that for any $\mathbf{y} = (0 \ \cdots \ 0 \ y_{n-t+1} \ \cdots \ y_n)$ with $1 \le w_{\mathrm{H}}(\mathbf{y}) \le t$, we have

$$\begin{aligned}
&\max\left\{P_{Y|X}^n(\mathbf{y}|\mathbf{x}_1), P_{Y|X}^n(\mathbf{y}|\mathbf{x}_2), P_{Y|X}^n(\mathbf{y}|\mathbf{x}_3), P_{Y|X}^n(\mathbf{y}|\mathbf{x}_4)\right\} \\
&\quad = \max\left\{0, P_{Y|X}^n(\mathbf{y}|\mathbf{x}_2), 0, P_{Y|X}^n(\mathbf{y}|\mathbf{x}_4)\right\} && (109)\\
&\quad = P_{Y|X}^n(\mathbf{y}|\mathbf{x}_2) && (110)\\
&\quad = (1-\epsilon_1)^{t-d_{\mathrm{H}}(\mathbf{x}_2,\mathbf{y})} \cdot \epsilon_1^{d_{\mathrm{H}}(\mathbf{x}_2,\mathbf{y})}, && (111)
\end{aligned}$$

where $0 \le d_{\mathrm{H}}(\mathbf{x}_2, \mathbf{y}) \le t-1$ and where the last step follows because we assume that $0 < \epsilon_1 \le \frac{1}{2}$ and that $w_{\mathrm{H}}(\mathbf{x}_4) = n > t = w_{\mathrm{H}}(\mathbf{x}_2)$. Hence, these vectors belong to $\mathcal{D}_{t,0;2}^{(4,n)}$.

A similar argument can be used for the description of decoding regions $\mathcal{D}_{t,0;3}^{(4,n)}$ and $\mathcal{D}_{t,0;4}^{(4,n)}$ (the latter is most easily described as consisting of all vectors that do not belong to the first three decoding regions). This proves (54)–(57).

Using (54)–(57) in (19) will then lead to (59).

A similar argument proves (54)–(56) for $\mathsf{M} = 3$, which then can be used together with (19) to derive (58).

We would like to add the following observation:

$$\begin{aligned}
\left|\mathcal{D}_{t,0;1}^{(4,n)}\right| &= 1; && (112)\\
\left|\mathcal{D}_{t,0;2}^{(4,n)}\right| &= \sum_{d=0}^{t-1}\binom{t}{d} = 2^t - 1; && (113)\\
\left|\mathcal{D}_{t,0;3}^{(4,n)}\right| &= \sum_{d=0}^{n-t-1}\binom{n-t}{d} = 2^{n-t} - 1; && (114)\\
\left|\mathcal{D}_{t,0;4}^{(4,n)}\right| &= \sum_{i=1}^{n-t}\sum_{j=1}^{t}\binom{n-t}{i}\binom{t}{j} && (115)\\
&= \sum_{d=0}^{n-1}\left(\binom{n}{d} - \binom{n-t}{d-t} - \binom{t}{d-(n-t)}\right) && (116)\\
&= 2^n - (2^t-1) - (2^{n-t}-1) - 1 = 2^n - 2^t - 2^{n-t} + 1. && (117)
\end{aligned}$$

Here, (115) denotes the number that both t and $n-t$ positions in their respective blocks contain at least one 1, while the expression in the sum of (116) denotes the number of sequences containing d zeros that are not part of $\mathcal{D}_{t,0;2}^{(4,n)}$ or $\mathcal{D}_{t,0;3}^{(4,n)}$.

B.2 Proof of Lemma 21

We start with $\mathsf{M} = 4$. Note that there are actually only 14 possible columns that we can choose from as the $(n+1)$th column because the all-zero and all-one columns are clearly suboptimal since in these two cases, an optimal decoder will simply ignore the $(n+1)$th received digit. This can be proven formally using an approach similar to the derivation shown in Appendix A.1.

To prove Lemma 21 we append an additional bit to all four codewords of $\mathscr{C}_{t,0}^{(4,n)}$ as follows:

$$\begin{pmatrix}
[\mathbf{0} & x_{1,n+1}]\\
[\mathbf{x} & x_{2,n+1}]\\
[\bar{\mathbf{x}} & x_{3,n+1}]\\
[\mathbf{1} & x_{4,n+1}]
\end{pmatrix} \qquad (118)$$

where $x_{m,n+1} \in \{0,1\}$ and where $\mathbf{x}$ and $\bar{\mathbf{x}}$ are given in (22) with $t \in \{1,2,\ldots,\lfloor \frac{n}{2} \rfloor\}$. We denote this new code by $\mathscr{C}^{(4,n+1)}$.

Note that again we use a proof technique that uses a given code to create a new code by adding a column to the codebook matrix. We therefore again use the notation introduced in Appendix A.1, i.e., we use superscripts (n) to clarify length and affiliation.

We now need to establish the decoding regions for the new code $\mathscr{C}^{(4,n+1)}$. If we simply extend the decoding regions given (54)–(57) by one bit, $\big[\mathcal{D}_{t,0;m}^{(4,n)}\ 0\big] \cup \big[\mathcal{D}_{t,0;m}^{(4,n)}\ 1\big]$, for $m = 1,2,3,4$, then we retain the same success probability because

$$\begin{aligned}
\psi_m\big(\mathscr{C}^{(4,n+1)}\big) &= \psi_m\big(\mathscr{C}_{t,0}^{(4,n)}\big) \cdot P_{Y|X}(0|x_{m,n+1}) + \psi_m\big(\mathscr{C}_{t,0}^{(4,n)}\big) \cdot P_{Y|X}(1|x_{m,n+1}) && (119)\\
&= \psi_m\big(\mathscr{C}_{t,0}^{(4,n)}\big) \cdot \big(\underbrace{P_{Y|X}(0|x_{m,n+1}) + P_{Y|X}(1|x_{m,n+1})}_{=1}\big) && (120)\\
&= \psi_m\big(\mathscr{C}_{t,0}^{(4,n)}\big). && (121)
\end{aligned}$$

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

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

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

- $\big[\mathcal{D}_{t,0;1}^{(4,n)}\ 0\big]$: The decoding region $\big[\mathcal{D}_{t,0;1}^{(4,n)}\ 0\big]$ only contains one vector: the all-zero vector. We have
$$P_{Y|X}^{n+1}\Big(\mathbf{0}^{(n+1)}\Big|\mathbf{x}_1^{(n+1)} = \big[\mathbf{0}^{(n)}\ 1\big]\Big) = \epsilon_1 \geq P_{Y|X}^{n+1}\big(\mathbf{0}^{(n+1)}\big|\mathbf{x}_m^{(n+1)}\big), \quad \forall\, m = 2,3,4, \tag{122}$$
independent of the choices of $x_{m,n+1}$, $m = 2,3,4$. Hence, we decide for $m = 1$.

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

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

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

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

$$P_{Y|X}^{n}\big(\mathbf{y}^{(n)}\big|\mathbf{0}^{(n)}\big) = P_{Y|X}^{n}\Big(\mathbf{y}^{(n)}\Big|\big[\mathbf{0}^{(n-t)}\ \mathbf{1}^{(t)}\big]\Big) = P_{Y|X}^{n}\Big(\mathbf{y}^{(n)}\Big|\big[\mathbf{1}^{(n-t)}\ \mathbf{0}^{(t)}\big]\Big) = 0. \tag{123}$$

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

- $\left[\mathcal{D}_{t,0;1}^{(4,n)}\ 1\right]$: If we assign these vectors (actually, it's only one) to the new decoding region $\mathcal{D}_{t,0;2}^{(4,n+1)}$, the conditional success probability for $m = 2$ is increased by

$$\begin{aligned}\Delta\psi_2 &\triangleq \psi_2\big(\mathscr{C}^{(4,n+1)}\big) - \psi_2\big(\mathscr{C}_{t,0}^{(4,n)}\big) && (124)\\ &= \sum_{\mathbf{y}^{(n)}\in\mathcal{D}_{t,0;1}^{(4,n)}} P_{Y|X}^{n+1}\Big(\big[\mathbf{y}^{(n)}\ 1\big]\Big|\big[\mathbf{0}^{(n-t)}\ \mathbf{1}^t\ 1\big]\Big)\cdot(x_{2,n+1}-x_{1,n+1})^+ && (125)\\ &= \epsilon_1^t\cdot(1-\epsilon_1)\cdot(x_{2,n+1}-x_{1,n+1})^+, && (126)\end{aligned}$$

  where

$$(x)^+ = \begin{cases} x & \text{if } x\geq 0,\\ 0 & \text{if } x<0.\end{cases} \qquad (127)$$

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

  Similarly, we compute

$$\begin{aligned}\Delta\psi_3 &= \epsilon_1^{n-t}\cdot(1-\epsilon_1)\cdot(x_{3,n+1}-x_{1,n+1})^+; && (128)\\ \Delta\psi_4 &= \epsilon_1^{n}\cdot(1-\epsilon_1)\cdot(x_{4,n+1}-x_{1,n+1})^+. && (129)\end{aligned}$$

  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, in order to represent this choice of ordering, we rewrite (126), (128), and (129) as follows:

$$\begin{aligned}\Delta\psi_2 &= \epsilon_1^t\cdot(1-\epsilon_1)\cdot(x_{2,n+1}-x_{1,n+1})^+, && (130)\\ \Delta\psi_3 &= \epsilon_1^{n-t}\cdot(1-\epsilon_1)\cdot(x_{3,n+1}-x_{2,n+1}-x_{1,n+1})^+, && (131)\\ \Delta\psi_4 &= \epsilon_1^{n}\cdot(1-\epsilon_1)\cdot(x_{4,n+1}-x_{3,n+1}-x_{2,n+1}-x_{1,n+1})^+. && (132)\end{aligned}$$

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

$$\begin{aligned}\Delta\psi_4 &= \sum_{\mathbf{y}^{(n)}\in\mathcal{D}_{t,0;2}^{(4,n)}} P_{Y|X}^{n+1}\Big(\big[\mathbf{y}^{(n)}\ 1\big]\Big|\big[\mathbf{1}^{(n)}\ 1\big]\Big)\cdot(x_{4,n+1}-x_{2,n+1})^+ && (133)\\ &= \sum_{d=0}^{t-1}\binom{t}{d}(1-\epsilon_1)^{t-d}\cdot\epsilon_1^{n-t+d}\cdot(1-\epsilon_1)\cdot(x_{4,n+1}-x_{2,n+1})^+ && (134)\\ &= \big(\epsilon_1^{n-t}-\epsilon_1^n\big)\cdot(1-\epsilon_1)\cdot(x_{4,n+1}-x_{2,n+1})^+. && (135)\end{aligned}$$

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

$$\begin{aligned}\Delta\psi_4 &= \sum_{\mathbf{y}^{(n)}\in\mathcal{D}_{t,0;3}^{(4,n)}} P_{Y|X}^{n+1}\Big(\big[\mathbf{y}^{(n)}\ 1\big]\Big|\big[\mathbf{1}^{(n)}\ 1\big]\Big)\cdot(x_{4,n+1}-x_{3,n+1})^+ && (136)\\ &= \big(\epsilon_1^{t}-\epsilon_1^n\big)\cdot(1-\epsilon_1)\cdot(x_{4,n+1}-x_{3,n+1})^+. && (137)\end{aligned}$$

For $\epsilon_1^t > \epsilon_1^{n-t} > \epsilon_1^n$, we can now conclude that the unique best solution for the choice of $x_{m,n+1}$, yielding the largest increase in success probability in (130), (131), (132),

(135), and (137), is as follows:

$$\begin{cases} x_{2,n+1} - x_{1,n+1} = 1, \\ x_{4,n+1} - x_{2,n+1} = 0, \\ x_{4,n+1} - x_{3,n+1} = 1 \end{cases} \quad \Longrightarrow \quad \begin{cases} x_{1,n+1} = 0, \\ x_{2,n+1} = 1, \\ x_{3,n+1} = 0, \\ x_{4,n+1} = 1, \end{cases} \tag{138}$$

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

$$\Delta P_{\mathrm{c}} = \frac{1}{4}\epsilon_1^t(1-\epsilon_1) + \frac{1}{4}\big(\epsilon_1^t - \epsilon_1^n\big)(1-\epsilon_1) = \frac{1}{4}\big(2\epsilon_1^t - \epsilon_1^n\big)(1-\epsilon_1). \tag{139}$$

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

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

C Derivations concerning the BSC

C.1 Proof of Theorem 25

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

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 (75) is optimal is straightforward and omitted.
- We do not need to worry about any other codebook columns than those given in (24) because firstly the all-zero and the all-one column can be neglected by the same argument as used in the proof of Theorem 14, 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.

Note that once we apply the notation introduced in Appendix A.1, i.e., we use a superscript (n) to denote length and affiliation. Moreover, we introduce the following shorthands:

$$d_m^{(n)}(\mathbf{y}) \triangleq d_{\mathrm{H}}(\mathbf{x}_m, \mathbf{y}), \quad m = 1, \ldots, \mathsf{M}, \tag{140}$$

and

$$\mathbf{d}^{(n)}(\mathbf{y}) \triangleq \big(d_1^{(n)}(\mathbf{y}), d_2^{(n)}(\mathbf{y}), \ldots, d_{\mathsf{M}}^{(n)}(\mathbf{y})\big). \tag{141}$$

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

We also remind the reader that $k \triangleq \lfloor \frac{n}{3} \rfloor$.

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

$$g(\mathbf{y}) = \underset{1\le m\le \mathsf{M}}{\operatorname{argmin}} \left\{ d_m^{(n)}(\mathbf{y}) \right\}. \tag{142}$$

We start with an observation about a basic property of the weak flip code given in (71).

Claim 28. *For the weak flip code of (71), $\mathscr{C}_{t_2^*,t_3^*}^{(3,n)}$, the largest received Hamming distance between any $\mathbf{y}$ and the nearest codeword is given by the minimum Hamming distance of the codebook:*

$$\max_{\mathbf{y}} \min_{j\in\{1,2,3\}} d_j^{(n)}(\mathbf{y}) = d_{\min}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n)}\big). \tag{143}$$

Proof. It is not too difficult to see that a $\mathbf{y}$ that achieves the maximum in (143) should have t_1^* ones, t_2^* ones, and t_3^* zeros in the positions where the optimal codebook consists of $\mathbf{c}_1^{(3)}$, $\mathbf{c}_2^{(3)}$, and $\mathbf{c}_3^{(3)}$, respectively:

$$\mathbf{y}_{\max} \triangleq (\underbrace{1\ \cdots\ 1}_{t_1^*}\ \underbrace{1\ \cdots\ 1}_{t_2^*}\ \underbrace{0\ \cdots\ 0}_{t_3^*}). \tag{144}$$

Then,

$$\begin{aligned}
\max_{\mathbf{y}} \min_{j\in\{1,2,3\}} d_j^{(n)}(\mathbf{y}) &= \max_{\mathbf{y}} \min\left\{ d_1^{(n)}(\mathbf{y}), d_2^{(n)}(\mathbf{y}), d_3^{(n)}(\mathbf{y}) \right\} && (145)\\
&= \min\left\{ d_1^{(n)}(\mathbf{y}_{\max}), d_2^{(n)}(\mathbf{y}_{\max}), d_3^{(n)}(\mathbf{y}_{\max}) \right\} && (146)\\
&= \min\{t_1^*+t_2^*, t_1^*+t_3^*, t_2^*+t_3^*\} && (147)\\
&= \min\left\{ d_{\mathrm{H}}\big(\mathbf{x}_2^{(n)}, \mathbf{x}_3^{(n)}\big), d_{\mathrm{H}}\big(\mathbf{x}_1^{(n)}, \mathbf{x}_3^{(n)}\big), d_{\mathrm{H}}\big(\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)}\big) \right\} && (148)\\
&= d_{\min}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n)}\big). && (149)
\end{aligned}$$

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

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

$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = \Big(d_{\mathrm{H}}\big(\mathbf{x}_1^{(n-1)}, \mathbf{x}_2^{(n-1)}\big), d_{\mathrm{H}}\big(\mathbf{x}_1^{(n-1)}, \mathbf{x}_3^{(n-1)}\big) + 1, d_{\mathrm{H}}\big(\mathbf{x}_2^{(n-1)}, \mathbf{x}_3^{(n-1)}\big) + 1 \Big). \tag{150}$$

We are now ready for our induction proof.

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

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

$$\begin{aligned}
&\text{code parameters:} && [t_1^*, t_2^*, t_3^*] = [k, k-1, k]; && (151)\\
&\text{pairwise Hamming distance vector:} && \mathbf{d}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) = (2k-1, 2k, 2k-1); && (152)\\
&\text{minimum Hamming distance:} && d_{\min}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) = 2k-1. && (153)
\end{aligned}$$

The corresponding success probability formula looks as follows:

$$3P_{\mathrm{c}}\Big(\mathscr{C}^{(3,n-1)}_{t_2^*,t_3^*}\Big) = \sum_{m=1}^{3} \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k-1,k;m}} P^{n-1}_{Y|X}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}^{(n-1)}_m\big) \tag{154}$$

$$= (1-\epsilon)^{n-1} \sum_{m=1}^{3} \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k-1,k;m}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_{\mathrm{H}}(\mathbf{x}^{(n-1)}_m,\mathbf{y}^{(n-1)})} \tag{155}$$

$$\begin{aligned} = (1-\epsilon)^{n-1} \Bigg(& \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k-1,k;1}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})} \\ & + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k-1,k;2}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})} \\ & + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k-1,k;3}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})} \Bigg) \end{aligned} \tag{156}$$

$$\begin{aligned} = (1-\epsilon)^{n} \Bigg(& \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k-1,k;1}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})} \\ & + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k-1,k;1}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})+1} \\ & + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k-1,k;2}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})} \\ & + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k-1,k;2}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})+1} \\ & + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k-1,k;3}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})} \\ & + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k-1,k;3}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})+1} \Bigg), \end{aligned} \tag{157}$$

where in the last equality we used the trick to write

$$1 = (1-\epsilon)\left(1+\frac{\epsilon}{1-\epsilon}\right). \tag{158}$$

1. **Appending $\mathbf{c}_2^{(3)}$:** We now build a new length-n (weak flip) code $\mathscr{C}^{(3,n)}$ from the given code $\mathscr{C}^{(3,n-1)}_{t_2^*,t_3^*}$ by appending $\mathbf{c}_2^{(3)} = (0\ 1\ 0)^{\mathsf{T}}$. 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:

$$[t_1, t_2, t_3] = [k, k, k]; \tag{159}$$

$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k, 2k, 2k); \tag{160}$$

$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k. \tag{161}$$

Now note that we can rewrite (157) in the following way:

$$\begin{aligned} 3P_{\mathrm{c}}\Big(\mathscr{C}^{(3,n-1)}_{t_2^*,t_3^*}\Big) = (1-\epsilon)^n \Bigg(& \sum_{\mathbf{y}^{(n)} \in \left[\mathcal{D}^{(3,n-1)}_{k-1,k;1}\ 0\right]} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})} \\ & + \sum_{\mathbf{y}^{(n)} \in \left[\mathcal{D}^{(3,n-1)}_{k-1,k;1}\ 1\right]} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})+1} \\ & + \sum_{\mathbf{y}^{(n)} \in \left[\mathcal{D}^{(3,n-1)}_{k-1,k;2}\ 1\right]} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})} \\ & + \sum_{\mathbf{y}^{(n)} \in \left[\mathcal{D}^{(3,n-1)}_{k-1,k;2}\ 0\right]} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})+1} \\ & + \sum_{\mathbf{y}^{(n)} \in \left[\mathcal{D}^{(3,n-1)}_{k-1,k;3}\ 0\right]} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})} \\ & + \sum_{\mathbf{y}^{(n)} \in \left[\mathcal{D}^{(3,n-1)}_{k-1,k;3}\ 1\right]} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})+1} \Bigg). \end{aligned} \tag{162}$$

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

$$\begin{aligned} & 3P_{\mathrm{c}}\big(\mathscr{C}^{(3,n)}\big) \\ & \quad = (1-\epsilon)^n \Bigg(\sum_{\mathbf{y}^{(n)} \in \mathcal{D}^{(3,n)}_1} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_1^{(n)}(\mathbf{y}^{(n)})} + \sum_{\mathbf{y}^{(n)} \in \mathcal{D}^{(3,n)}_2} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_2^{(n)}(\mathbf{y}^{(n)})} \\ & \qquad + \sum_{\mathbf{y}^{(n)} \in \mathcal{D}^{(3,n)}_3} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_3^{(n)}(\mathbf{y}^{(n)})} \Bigg), \end{aligned} \tag{163}$$

where we use $\mathcal{D}^{(3,n)}_m$ to denote the decoding region of the new code $\mathscr{C}^{(3,n)}$. In order to be able to compare (162) with (163), we need to be able to compare $\mathcal{D}^{(3,n-1)}_{k-1,k;m}$ with $\mathcal{D}^{(3,n)}_m$ and $d_m^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ with $d_m^{(n)}\big(\mathbf{y}^{(n)}\big)$. Note that every $\mathbf{y}^{(n)}$ can be uniquely written as some $\mathbf{y}^{(n-1)}$ plus an appended 0 or 1.

Since we have appended $\mathbf{c}_2^{(3)} = (0\ 1\ 0)^{\mathsf{T}}$ to the code of length $n-1$, it is obvious that

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k-1,k;1} \implies \big[\mathbf{y}^{(n-1)}\ 0\big] \in \mathcal{D}^{(3,n)}_1; \quad d_1^{(n)}\big(\mathbf{y}^{(n)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big);$$

(164)

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k-1,k;2}^{(3,n-1)} \implies \left[\mathbf{y}^{(n-1)}\ 1\right] \in \mathcal{D}_2^{(3,n)};\quad d_2^{(n)}\big(\mathbf{y}^{(n)}\big) = d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big);$$

(165)

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k-1,k;3}^{(3,n-1)} \implies \left[\mathbf{y}^{(n-1)}\ 0\right] \in \mathcal{D}_3^{(3,n)};\quad d_3^{(n)}\big(\mathbf{y}^{(n)}\big) = d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big).$$

(166)

The problems are the other three cases. For example,

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k-1,k;1}^{(3,n-1)} \implies \left[\mathbf{y}^{(n-1)}\ 1\right] \in \mathcal{D}_1^{(3,n)} \text{ or } \mathcal{D}_2^{(3,n)}, \tag{167}$$

depending on the exact value of $d_m^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$. Note that $\left[\mathbf{y}^{(n-1)}\ 1\right] \notin \mathcal{D}_3^{(3,n)}$ because we have added a 0 to the third codeword. To be able to investigate the different possible cases depending on $d_m^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$, we introduce a shorthand

$$d \triangleq \min_{m\in\{1,2,3\}} d_m^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \tag{168}$$

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:

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d,d,d) \text{ or } (d,d,d^+) \text{ or } (d,d^+,d) \text{ or } (d,d^+,d^+). \tag{169}$$

If we append a 1 to $\mathbf{y}^{(n-1)}$, then the first and the third component of $\mathbf{d}^{(n)}\big(\mathbf{y}^{(n)}\big)$ will be increased by 1 in comparison to $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$, while the second component remains unchanged. This means that in the third and fourth case in (169), the new vector $\left[\mathbf{y}^{(n-1)}\ 1\right]$ will belong to $\mathcal{D}_1^{(3,n)}$, while in the first and second case it will belong to $\mathcal{D}_2^{(3,n)}$. However, we will show next that the first and the second case can never occur!

To show this, first of all note that $d \geq k$ because the codebook's minimum Hamming distance between codewords is $2k-1$ and therefore it is not possible that a vector $\mathbf{y}^{(n-1)}$ has a distance to two (or more) codewords that is smaller than k. Also, from Claim 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 analogously to (144). To that goal we define a_m to be the number of positions where $\mathbf{y}^{(n-1)}$ differs from $\mathbf{y}_{\max}^{(n-1)}$ when we only consider the t_m^* positions corresponding to $\mathbf{c}_m^{(3)}$, i.e., $0 \leq a_m \leq t_m^*$, $m = 1,2,3$. For example, the all-zero vector $\mathbf{y} = \mathbf{0}$ has $a_1 = t_1^*$, $a_2 = t_2^*$, and $a_3 = 0$.

Then we define a matrix

$$\begin{pmatrix} t_1^*-a_1 & t_2^*-a_2 & a_3 \\ t_1^*-a_1 & a_2 & t_3^*-a_3 \\ a_1 & t_2^*-a_2 & t_3^*-a_3 \end{pmatrix} = \begin{pmatrix} k-a_1 & k-1-a_2 & a_3 \\ k-a_1 & a_2 & k-a_3 \\ a_1 & k-1-a_2 & k-a_3 \end{pmatrix} \tag{170}$$

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

$$\begin{pmatrix} d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \end{pmatrix} = \begin{pmatrix} k-a_1 & k-1-a_2 & a_3 \\ k-a_1 & a_2 & k-a_3 \\ a_1 & k-1-a_2 & k-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \tag{171}$$

It is straightforward to prove the following claim.

Claim 29. *There exists no integer solution* (a_1, a_2, a_3), $0 \le a_1 \le k$, $0 \le a_2 \le k-1$, $0 \le a_3 \le k$, *that satisfies*

$$\begin{pmatrix} k-a_1 & k-1-a_2 & a_3 \\ k-a_1 & a_2 & k-a_3 \\ a_1 & k-1-a_2 & k-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d \\ d \end{pmatrix} \text{ or } \begin{pmatrix} d \\ d \\ d^+ \end{pmatrix} \text{ or } \begin{pmatrix} d^+ \\ d \\ d \end{pmatrix} \tag{172}$$

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

$$\begin{pmatrix} k-a_1 & k-1-a_2 & a_3 \\ k-a_1 & a_2 & k-a_3 \\ a_1 & k-1-a_2 & k-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d^+ \\ d \end{pmatrix}. \tag{173}$$

Hence, we have shown that

$$\begin{aligned} &\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k-1,k;1}^{(3,n-1)} \\ &\qquad \Longrightarrow \left[\mathbf{y}^{(n-1)}\ 1\right] \in \mathcal{D}_1^{(3,n)}; \quad d_1^{(n)}\big(\mathbf{y}^{(n)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) + 1. \end{aligned} \tag{174}$$

Similarly,

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k-1,k;2}^{(3,n-1)} \Longrightarrow \left[\mathbf{y}^{(n-1)}\ 0\right] \in \mathcal{D}_1^{(3,n)} \text{ or } \mathcal{D}_2^{(3,n)} \text{ or } \mathcal{D}_3^{(3,n)}, \tag{175}$$

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

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d,d,d) \text{ or } (d,d,d^+) \text{ or } (d^+,d,d) \text{ or } (d^+,d,d^+). \tag{176}$$

In the fourth case $\left[\mathbf{y}^{(n-1)}\ 0\right]$ will remain in $\mathcal{D}_2^{(3,n)}$, in all other three cases it will change to another decision region. However all these three cases are not possible according to (172).

Finally,

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k-1,k;3}^{(3,n-1)} \Longrightarrow \left[\mathbf{y}^{(n-1)}\ 1\right] \in \mathcal{D}_2^{(3,n)} \text{ or } \mathcal{D}_3^{(3,n)}, \tag{177}$$

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

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d,d,d) \text{ or } (d^+,d,d) \text{ or } (d,d^+,d) \text{ or } (d^+,d^+,d). \tag{178}$$

In the first and second case $\left[\mathbf{y}^{(n-1)}\ 1\right]$ will change to $\mathcal{D}_2^{(3,n)}$, in the other two cases it will remain in $\mathcal{D}_3^{(3,n)}$. Again, the first and the second case are not possible according to (172).

Hence, we have shown that

$$\begin{aligned} &\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k-1,k;1}^{(3,n-1)} \\ &\qquad \Longrightarrow \left[\mathbf{y}^{(n-1)}\ 1\right] \in \mathcal{D}_1^{(3,n)}; \quad d_1^{(n)}\big(\mathbf{y}^{(n)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) + 1; \end{aligned} \tag{179}$$

$$\begin{aligned} &\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k-1,k;2}^{(3,n-1)} \\ &\qquad \Longrightarrow \left[\mathbf{y}^{(n-1)}\ 0\right] \in \mathcal{D}_2^{(3,n)}; \quad d_2^{(n)}\big(\mathbf{y}^{(n)}\big) = d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) + 1; \end{aligned} \tag{180}$$

$$\begin{aligned} &\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k-1,k;3}^{(3,n-1)} \\ &\qquad \Longrightarrow \left[\mathbf{y}^{(n-1)}\ 1\right] \in \mathcal{D}_3^{(3,n)}; \quad d_3^{(n)}\big(\mathbf{y}^{(n)}\big) = d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) + 1. \end{aligned} \tag{181}$$

But this proves that the success probability of (163) is identical to the success probability of (162)! 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 $\mathbf{c}_1^{(3)} = (0\ 0\ 1)^\mathsf{T}$. The new code has the following parameters:

$$[t_1, t_2, t_3] = [k+1, k-1, k]; \tag{182}$$
$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k-1, 2k+1, 2k); \tag{183}$$
$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k-1. \tag{184}$$

One of the three problematic cases now is

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k-1,k;1}^{(3,n-1)} \implies \big[\mathbf{y}^{(n-1)}\ 1\big] \in \mathcal{D}_1^{(3,n)} \text{ or } \mathcal{D}_3^{(3,n)}, \tag{185}$$

depending on the exact value of $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ given in (169). If we append a 1 to $\mathbf{y}^{(n-1)}$, the first and the second component of $\mathbf{d}^{(n)}\big(\mathbf{y}^{(n)}\big)$ will be increased by 1 in comparison to $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$, while the third component remains unchanged. This means that in the first and third case the new vector $\big[\mathbf{y}^{(n-1)}\ 1\big]$ will belong to $\mathcal{D}_3^{(3,n)}$, while in the second and the fourth case it will belong to $\mathcal{D}_1^{(3,n)}$. According to Claim 29, the third case is possible and does happen. If $\big[\mathbf{y}^{(n-1)}\ 1\big] \in \mathcal{D}_3^{(3,n)}$, then we have that

$$d_3^{(n)}\big(\mathbf{y}^{(n)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \tag{186}$$

without the additional increase by 1. This then means that the success probability of (163) is strictly larger than the success probability of $\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}$ because

$$\left(\frac{\epsilon}{1-\epsilon}\right)^{d_3^{(n)}\left(\mathbf{y}^{(n)}\right)} = \left(\frac{\epsilon}{1-\epsilon}\right)^{d_1^{(n-1)}\left(\mathbf{y}^{(n-1)}\right)} > \left(\frac{\epsilon}{1-\epsilon}\right)^{d_1^{(n-1)}\left(\mathbf{y}^{(n-1)}\right)+1}, \tag{187}$$

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:

$$[t_1, t_2, t_3] = [k, k-1, k+1]; \tag{188}$$
$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k, 2k+1, 2k-1); \tag{189}$$
$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k-1. \tag{190}$$

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 $\mathbf{c}_1^{(3)}$.

C.1.2 Case ii: Step from $n-1 = 3k$ to $n = 3k+1$:

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

code parameters: $$[t_1^*, t_2^*, t_3^*] = [k+1, k-1, k]; \tag{191}$$

pairwise Hamming distance vector: $$\mathbf{d}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) = (2k-1, 2k+1, 2k); \tag{192}$$

mininum Hamming distance: $$d_{\min}\big(\mathscr{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) = 2k-1. \tag{193}$$

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

$$[t_1, t_2, t_3] = [k+2, k-1, k]; \tag{194}$$
$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k-1, 2k+2, 2k+1); \tag{195}$$
$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k-1. \tag{196}$$

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

$$[t_1, t_2, t_3] = [k+1, k, k]; \tag{197}$$
$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k, 2k+1, 2k+1); \tag{198}$$
$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k. \tag{199}$$

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

$$[t_1, t_2, t_3] = [k+1, k-1, k+1]; \tag{200}$$
$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k, 2k+2, 2k); \tag{201}$$
$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k. \tag{202}$$

The corresponding investigation of possible situations now reads as follows.

Claim 30. *There exists no integer solution (a_1, a_2, a_3), $0 \le a_1 \le k$, $0 \le a_2 \le k-1$, $0 \le a_3 \le k$, that satisfies*

$$\begin{pmatrix} k+1-a_1 & k-1-a_2 & a_3 \\ k+1-a_1 & a_2 & k-a_3 \\ a_1 & k-1-a_2 & k-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d \\ d \end{pmatrix} \text{ or } \begin{pmatrix} d \\ d^+ \\ d \end{pmatrix} \text{ or } \begin{pmatrix} d \\ d \\ d^+ \end{pmatrix} \tag{203}$$

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

$$\begin{pmatrix} k+1-a_1 & k-1-a_2 & a_3 \\ k+1-a_1 & a_2 & k-a_3 \\ a_1 & k-1-a_2 & k-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d^+ \\ d \\ d \end{pmatrix}. \tag{204}$$

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.

C.1.3 Case iii: Step from $n-1 = 3k+1$ to $n = 3k+2$:

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

code parameters:	$[t_1^*, t_2^*, t_3^*] = [k+1, k, k];$	(205)
pairwise Hamming distance vector:	$\mathbf{d}\big(\mathscr{C}^{(3,n-1)}_{t_2^*,t_3^*}\big) = (2k, 2k+1, 2k+1);$	(206)
mininum Hamming distance:	$d_{\min}\big(\mathscr{C}^{(3,n-1)}_{t_2^*,t_3^*}\big) = 2k.$	(207)

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

$$[t_1, t_2, t_3] = [k+2, k, k]; \tag{208}$$
$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k, 2k+2, 2k+2); \tag{209}$$
$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k. \tag{210}$$

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

$$[t_1, t_2, t_3] = [k+1, k+1, k]; \tag{211}$$
$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k+1, 2k+1, 2k+2); \tag{212}$$
$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k+1. \tag{213}$$

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

$$[t_1, t_2, t_3] = [k+1, k, k+1]; \tag{214}$$
$$\mathbf{d}\big(\mathscr{C}^{(3,n)}\big) = (2k+1, 2k+2, 2k+1); \tag{215}$$
$$d_{\min}\big(\mathscr{C}^{(3,n)}\big) = 2k+1. \tag{216}$$

The corresponding investigation of possible situations now reads as follows.

Claim 31. *There exists no integer solution* (a_1, a_2, a_3), $0 \le a_1 \le k$, $0 \le a_2 \le k-1$, $0 \le a_3 \le k$, *that satisfies*

$$\begin{pmatrix} k+1-a_1 & k-a_2 & a_3 \\ k+1-a_1 & a_2 & k-a_3 \\ a_1 & k-a_2 & k-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d \\ d \end{pmatrix} \text{ or } \begin{pmatrix} d^+ \\ d \\ d \end{pmatrix} \text{ or } \begin{pmatrix} d \\ d^+ \\ d \end{pmatrix} \tag{217}$$

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

$$\begin{pmatrix} k+1-a_1 & k-a_2 & a_3 \\ k+1-a_1 & a_2 & k-a_3 \\ a_1 & k-a_2 & k-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d \\ d^+ \end{pmatrix}. \tag{218}$$

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

This completes the proof for $\mathsf{M} = 3$.

Finally, we turn to the case $\mathsf{M} = 4$. We note that the fourth codeword for $\mathsf{M} = 4$ is exactly the furthest received vector for $\mathsf{M} = 3$. We can therefore adapt the computation of the received Hamming distance vector as follows:

$$\begin{pmatrix} d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_4^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \end{pmatrix} = \begin{pmatrix} t_1-a_1 & t_2-a_2 & a_3 \\ t_1-a_1 & a_2 & t_3-a_3 \\ a_1 & t_2-a_2 & t_3-a_3 \\ a_1 & a_2 & a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \tag{219}$$

The derivation follows then exactly the same lines as for $\mathsf{M} = 3$. The only main difference is that we need to investigate more different columns. Actually, we need to investigate also some columns that have not been named in Definition 10 like, e.g., $\mathbf{c} = (0\ 0\ 0\ 1)^\mathsf{T}$ and prove that they are strictly suboptimal. The details are omitted.

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] 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.

[3] Ebrahim MolavianJazi and J. Nicholas Laneman, "Discrete memoryless multiple access channel in the finite blocklength regime," March 2012, submitted to *IEEE International Symposium on Information Theory (ISIT)*.

[4] Jakob Hoydis, Romain Couillet, Pablo Piantanida, and Mérouane Debbah, "A random matrix approach to the finite blocklength regime of mimo fading channels," March 2012, submitted to *IEEE International Symposium on Information Theory (ISIT)*.

[5] Mustafa Cenk Gursoy, "Throughput analysis of buffer-constrained wireless systems in the finite blocklength regime," in *Proceedings IEEE International Conference on Communications (ICC)*, Kyoto, Japan, June 5–9, 2011.

[6] Deli Qiao, Mustafa Cenk Gursoy, and Senem Velipasalar, "Channel coding over multiple coherence blocks with queueing constraints," in *Proceedings IEEE International Conference on Communications (ICC)*, Kyoto, Japan, June 5–9, 2011.

[7] Thomas J. Riedl, Todd P. Coleman, and Andrew C. Singer, "Finite block-length achievable rates for queuing timing channels," in *Proceedings IEEE Information Theory Workshop (ITW)*, Paraty, Brazil, October 16–20, 2011, pp. 200–204.

[8] Victoria Kostina and Sergio Verdú, "Fixed-length lossy compression in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 99, 2012.

[9] Alfonso Martinez and Albert Guillén i Fàbregas, "Saddlepoint approximation of randomcoding bounds," in *Proceedings Information Theory and Applications Workshop (ITA), University of California*, San Diego, USA, February 6–11, 2011.

[10] Alfonso Martinez and Albert Guillén i Fàbregas, "Randomcoding bounds for threshold decoders: Error exponent and saddlepoint approximation," in *Proceedings IEEE International Symposium on Information Theory (ISIT)*, St. Petersburg, Russia, July 31 – August 5, 2011, pp. 2899–2903.

[11] Albert Guillén i Fàbregas, Ingmar Land, and Alfonso Martinez, "Extremes of random coding error exponents," in *Proceedings IEEE International Symposium on Information Theory (ISIT)*, St. Petersburg, Russia, July 31 – August 5, 2011, pp. 2896–2898.

[12] Robert G. Gallager, *Information Theory and Reliable Communication*. New York: John Wiley & Sons, 1968.

[13] Shlomo Shamai (Shitz) and Sergio Verdú, "The empirical distribution of good codes," *IEEE Transactions on Information Theory*, vol. 43, no. 3, pp. 836–846, May 1997.

[14] 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.

[15] J. Nicholas Laneman, "On the distribution of mutual information," in *Proceedings Information Theory and Applications Workshop (ITA), University of California*, San Diego, USA, February 6–10, 2006.

[16] David Buckingham and Matthew C. Valenti, "The information-outage probability of finite-length codes over AWGN channels," in *Proceedings Annual Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, USA, March 19–21, 2008, pp. 390–395.

[17] Shu Lin and Daniel J. Costello, Jr., *Error Control Coding*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2004.

[18] Po-Ning Chen, Hsuan-Yin Lin, and Stefan M. Moser, "Weak flip codes and applications to optimal block-codes on the binary erasure channel," May 2012, submitted. [Online]. Available: http://moser.cm.nctu.edu.tw/publications.html