Optimal Ultrasmall Block-Codes for Binary Discrete Memoryless Channels

Po-Ning Chen, *Senior Member, IEEE*, Hsuan-Yin Lin, *Student Member, IEEE*, and Stefan M. Moser, *Senior Member, IEEE*

***Abstract*—Optimal block-codes (in the sense of minimum average error probability, using maximum likelihood decoding) 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), both with arbitrary cross-over probabilities. For the ZC, the optimal code structure for an arbitrary finite blocklength is derived in the cases of two, three, and four codewords and conjectured in the case of five codewords. For the BSC, the optimal code structure for an arbitrary finite blocklength is derived in the cases of two and three codewords and conjectured in the case of four codewords. For a general BAC, the best codebooks under the assumption of a threshold decoder are derived for the case of two codewords. The derivation of these optimal codes relies on a new approach of constructing and analyzing the codebook matrix not rowwise (codewords), but *column-wise*. This new tool leads to an elegant definition of interesting code families that is recursive in the blocklength n and admits their *exact* analysis of error performance. This allows for a comparison of the average error probability between all possible codebooks.**

***Index Terms*—Binary asymmetric channel (BAC), binary symmetric channel (BSC), finite blocklength, flip codes, maximum likelihood (ML) decoder, minimum average error probability, optimal codes, weak flip codes, Z-channel (ZC).**

I. 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, but used a proof technique based on random coding that ensures the codes' existence. 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. In consequence, having large blocklength might not be an option, but 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.

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

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

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

Our approach in this 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 *ultrasmall block-codes*. By this reduction of the transmission rates, our results are directly applicable even for very short blocklengths. In contrast to [2] that provides 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 [7]. There the authors try to describe the empirical distribution of good codes (i.e., of codes that approach capacity with vanishing error probability)

Manuscript received March 20, 2012; accepted December 22, 2012. Date of publication August 07, 2013; date of current version October 16, 2013. This work was supported by the National Science Council under Grants NSC 97–2221–E–009–003–MY3 and NSC 100–2221–E–009–068–MY3. Parts of this paper were presented at the 2011 IEEE Information Theory Workshop.

The authors are with the Department of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: qponing@gmail.com; lin.hsuanyin@ieee.org; stefan.moser@ieee.org).

Communicated by I. Kontoyiannis, Associate Editor At Large.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2013.2276893

and show that for a large enough blocklength, the empirical distribution of certain good codes converges in the sense of divergence to a set of input distributions that maximize the input-output mutual information. Note, however, that [7] again focuses on the asymptotic regime, while our focus lies on finite blocklength, and not capacity-achieving codes.

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

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

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

Our main contributions are as follows:

1) We provide first fundamental insights into the performance analysis of *optimal nonlinear code design* for the BAC. Note that there exists a vast literature about linear codes, their properties and good linear design (e.g., [12]). Some Hamming-distance related topics of nonlinear codes are addressed in [13].[1]
2) We provide new insights in the optimal code construction for the BAC for an arbitrary finite blocklength n and for $\mathsf{M} = 2$ codewords.

[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 [14].

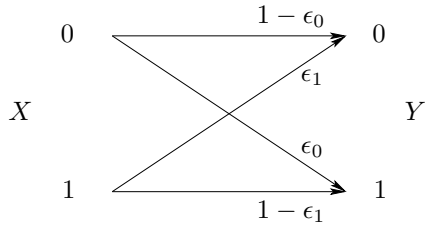

Fig. 1. Binary asymmetric channel (or BAC).

3) We provide optimal code constructions for the ZC for an arbitrary finite blocklength n and for $\mathsf{M} = 2$, 3 and 4 codewords. For the BSC, we provide optimal code constructions for an arbitrary finite blocklength n and for $\mathsf{M} = 2$ and 3 codewords and locally optimal code constructions for $\mathsf{M} = 4$ codewords.
4) We propose a new approach to the design and analysis of block-codes: instead of focusing on the codewords (i.e., the rows in the codebook matrix), we look at the codebook matrix in a *column-wise* manner.

The remainder of this paper is structured as follows: we end this introduction with some comments about our notation and will then introduce our channel models in Section II. After some more preliminaries in Section III, Section IV contains a very short example showing that the analysis of even such simple channel models is nontrivial and often nonintuitive. Section V then presents new code definitions that will be used for our main results. In Section VI, we review some important previous work. Section VII–IX then contain our main results. In Section VII, we analyze the BAC for two codewords, Section VIII takes a closer look at the ZC, and in Section IX we investigate the BSC. Many of the lengthy proofs have been moved to the appendix. We conclude in Section X.

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 codeword $\mathbf{x}$ or the received sequence $\mathbf{y}$. Otherwise our used notation follows the main stream. We use capital letters for random quantities, e.g., X, and small letters for realizations, e.g., x; 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., D.

II. 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 flipped into a 1, and ϵ_1 denotes the probability that a 1 is flipped into a 0, see Fig. 1.

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

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

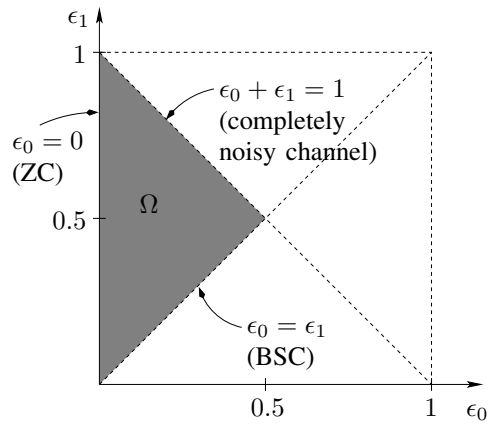

Fig. 2. Region of possible choices of the channel parameters ϵ_0 and ϵ_1 of a BAC. The shaded area corresponds to the interesting area according to (1)–(3).

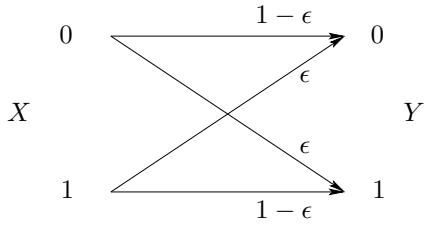

Fig. 3. Binary symmetric channel.

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

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

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 $\epsilon_0 \le 1 - \epsilon_1$.

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

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

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

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

The following three definitions are commonly used.

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

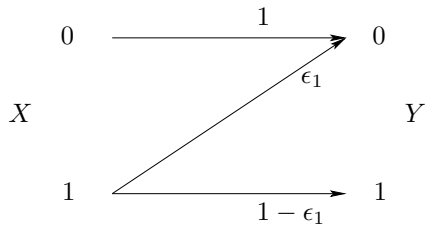

Fig. 4. Z-channel.

A codebook is called *linear* if it can be seen as a subspace of the n-dimensional vector space over the channel input alphabet.[2]

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 a $\mathcal{C}^{(\mathsf{M},n)}$ code:

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

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

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

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

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

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

and on the corresponding *average success probability*[3] $P_{\mathrm{c}}\big(\mathcal{C}^{(\mathsf{M},n)}\big)$.

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

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

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 a given (M, n) coding scheme, we define the *decoding region* $\mathcal{D}_m^{(\mathsf{M},n)}$ as the set of n-vectors $\mathbf{y}$ that are decoded to the message m:

$$\mathcal{D}_m^{(\mathsf{M},n)} \triangleq \{\mathbf{y} \colon g(\mathbf{y}) = m\}. \tag{9}$$

Moreover, we also make the following definitions.

Definition 4: By $d_{\alpha,\beta}(\mathbf{x}_m, \mathbf{y})$ we denote the number of positions j, where $x_{m,j} = \alpha$ and $y_j = \beta$. For $m \ne m'$,

[2]Being a subspace, linear codes usually are represented by a generator matrix, which is basically a basis of the subspace. As we are not interested in linear codes in particular, but in both linear and nonlinear codes, we will not use this in this paper.

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

the *joint composition* $q_{\alpha,\beta}(m,m')$ of two codewords $\mathbf{x}_m$ and $\mathbf{x}_{m'}$ is defined as

$$q_{\alpha,\beta}(m,m') \triangleq \frac{d_{\alpha,\beta}(\mathbf{x}_m,\mathbf{x}_{m'})}{n}. \tag{10}$$

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

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

Remark 5: It is usual to write the codebook $\mathcal{C}^{(\mathsf{M},n)}$ as an $\mathsf{M}\times n$ matrix with its M rows corresponding to the M codewords:

$$\mathcal{C}^{(\mathsf{M},n)} = \begin{pmatrix} — \ \mathbf{x}_1 \ — \\ \vdots \\ — \ \mathbf{x}_{\mathsf{M}} \ — \end{pmatrix} = \begin{pmatrix} | & | & & | \\ \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_n \\ | & | & & | \end{pmatrix}. \tag{11}$$

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

Remark 6: 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 $\mathcal{C}^{(\mathsf{M},n)}$ 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.

III. Preliminaries

A. Error Probability of the BAC

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

$$P_{Y|X}^n(\mathbf{y}|\mathbf{x}_m) = (1-\epsilon_0)^{d_{0,0}(\mathbf{x}_m,\mathbf{y})} \cdot \epsilon_0^{d_{0,1}(\mathbf{x}_m,\mathbf{y})} \cdot \epsilon_1^{d_{1,0}(\mathbf{x}_m,\mathbf{y})} \cdot (1-\epsilon_1)^{d_{1,1}(\mathbf{x}_m,\mathbf{y})} \tag{12}$$

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

$$P_{Y|X}^n(\mathbf{y}|\mathbf{x}) \triangleq \prod_{j=1}^{n} P_{Y|X}(y_j|x_j). \tag{13}$$

Considering that

$$n = d_{0,0}(\mathbf{x}_m,\mathbf{y}) + d_{0,1}(\mathbf{x}_m,\mathbf{y}) + d_{1,0}(\mathbf{x}_m,\mathbf{y}) + d_{1,1}(\mathbf{x}_m,\mathbf{y}) \tag{14}$$

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

$$P_{\mathrm{e}}\big(\mathcal{C}^{(\mathsf{M},n)}\big) = \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})\neq m}} P_{Y|X}^n(\mathbf{y}|\mathbf{x}_m) \tag{15}$$

$$= \frac{(1-\epsilon_0)^n}{\mathsf{M}} \sum_{\mathbf{y}} \sum_{\substack{m=1 \\ m\neq g(\mathbf{y})}}^{\mathsf{M}} \left(\frac{\epsilon_0}{1-\epsilon_0}\right)^{d_{0,1}(\mathbf{x}_m,\mathbf{y})} \cdot \left(\frac{\epsilon_1}{1-\epsilon_0}\right)^{d_{1,0}(\mathbf{x}_m,\mathbf{y})} \left(\frac{1-\epsilon_1}{1-\epsilon_0}\right)^{d_{1,1}(\mathbf{x}_m,\mathbf{y})} \tag{16}$$

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

B. Error (and Success) Probability of the BSC

In the special case of a BSC, (16) simplifies to

$$P_{\mathrm{e}}\big(\mathcal{C}^{(\mathsf{M},n)}\big) = \frac{(1-\epsilon)^n}{\mathsf{M}} \sum_{\mathbf{y}} \sum_{\substack{m=1 \\ m\neq g(\mathbf{y})}}^{\mathsf{M}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_{\mathrm{H}}(\mathbf{x}_m,\mathbf{y})}. \tag{17}$$

The success probability is accordingly[4]

$$P_{\mathrm{c}}\big(\mathcal{C}^{(\mathsf{M},n)}\big) = \frac{(1-\epsilon)^n}{\mathsf{M}} \sum_{\mathbf{y}} \sum_{\substack{m=1 \\ m= g(\mathbf{y})}}^{\mathsf{M}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_{\mathrm{H}}(\mathbf{x}_m,\mathbf{y})}. \tag{18}$$

C. Error (and Success) Probability of the ZC

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

$$P_{\mathrm{c}}\big(\mathcal{C}^{(\mathsf{M},n)}\big) = \frac{1}{\mathsf{M}} \sum_{\mathbf{y}} \sum_{\substack{m=1 \\ m= g(\mathbf{y})}}^{\mathsf{M}} \mathsf{I}\{d_{0,1}(\mathbf{x}_m,\mathbf{y})=0\} \left(\frac{\epsilon_1}{1-\epsilon_1}\right)^{d_{1,0}(\mathbf{x}_m,\mathbf{y})} \cdot (1-\epsilon_1)^{d_{1,1}(\mathbf{x}_m,\mathbf{y})+d_{1,0}(\mathbf{x}_m,\mathbf{y})} \tag{19}$$

$$= \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})= m}} \mathsf{I}\{d_{0,1}(\mathbf{x}_m,\mathbf{y})=0\} \left(\frac{\epsilon_1}{1-\epsilon_1}\right)^{d_{1,0}(\mathbf{x}_m,\mathbf{y})} \cdot (1-\epsilon_1)^{w_{\mathrm{H}}(\mathbf{x}_m)}. \tag{20}$$

The error probability formula is accordingly

$$P_{\mathrm{e}}\big(\mathcal{C}^{(\mathsf{M},n)}\big) = \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y})\neq m}} \mathsf{I}\{d_{0,1}(\mathbf{x}_m,\mathbf{y})=0\} \cdot \left(\frac{\epsilon_1}{1-\epsilon_1}\right)^{d_{1,0}(\mathbf{x}_m,\mathbf{y})} (1-\epsilon_1)^{w_{\mathrm{H}}(\mathbf{x}_m)}. \tag{21}$$

[4]Note that the second summation contains only one value and could be replaced by an indicator function.

D. Pairwise Hamming Distance

The minimum Hamming distance is a well-known and often used quality criterion of a codebook, see, e.g., [12], [13]. In [13, Ch. 2], the maximum minimum Hamming distance for a given code $\mathcal{C}^{(\mathsf{M},n)}$ is discussed including important results like the Plotkin bound and Levenshtein's theorem. (For more details about upper and lower bounds to the average error probability, see also Section VI.) 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 IX-C).

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

Definition 7: Given a codebook $\mathcal{C}^{(\mathsf{M},n)}$ with codewords $\mathbf{x}_m$, $1 \le m \le \mathsf{M}$, we define the *pairwise Hamming distance vector* of length $\frac{1}{2}(\mathsf{M}-1)\mathsf{M}$ as follows:

$$\begin{aligned}&\mathbf{d}\big(\mathcal{C}^{(\mathsf{M},n)}\big)\\ &\quad\triangleq \Big(d_{\mathrm{H}}(\mathbf{x}_1,\mathbf{x}_2),\\ &\qquad d_{\mathrm{H}}(\mathbf{x}_1,\mathbf{x}_3),\ d_{\mathrm{H}}(\mathbf{x}_2,\mathbf{x}_3),\\ &\qquad d_{\mathrm{H}}(\mathbf{x}_1,\mathbf{x}_4),\ d_{\mathrm{H}}(\mathbf{x}_2,\mathbf{x}_4),\ d_{\mathrm{H}}(\mathbf{x}_3,\mathbf{x}_4),\\ &\qquad \ldots,\\ &\qquad d_{\mathrm{H}}(\mathbf{x}_1,\mathbf{x}_{\mathsf{M}}),\ d_{\mathrm{H}}(\mathbf{x}_2,\mathbf{x}_{\mathsf{M}}),\ldots,\ d_{\mathrm{H}}(\mathbf{x}_{\mathsf{M}-1},\mathbf{x}_{\mathsf{M}})\Big). \quad (22)\end{aligned}$$

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

IV. 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-over probability $\epsilon = 0.4$, $\mathsf{M} = 4$, and a blocklength $n = 4$. Then consider the following codes:[5]

$$\mathcal{C}_1^{(4,4)} = \begin{pmatrix}0&0&0&0\\0&0&0&1\\1&1&1&0\\1&1&1&1\end{pmatrix}, \quad \mathcal{C}_2^{(4,4)} = \begin{pmatrix}0&0&0&0\\0&0&1&1\\1&1&0&0\\1&1&1&1\end{pmatrix}. \quad (23)$$

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. It is quite common to believe that $\mathcal{C}_2^{(4,4)}$ shows a better performance. This intuition is based on Gallager's famous performance bound [6, Ex. 5.19]:

$$P_{\mathrm{e}}\big(\mathcal{C}^{(\mathsf{M},n)}\big) \le (\mathsf{M}-1)e^{-d_{\min}(\mathcal{C}^{(\mathsf{M},n)})\log\frac{1}{\sqrt{4\epsilon(1-\epsilon)}}}. \quad (24)$$

However, the exact average error probability as given in (17) actually can be evaluated as $P_{\mathrm{e}}\big(\mathcal{C}_1^{(4,4)}\big) \approx 0.6112$ and $P_{\mathrm{e}}\big(\mathcal{C}_2^{(4,4)}\big) = 0.64$. Hence, even though the minimum Hamming distance of the first codebook is smaller, its overall performance is superior to the second codebook!

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

$$P_{\mathrm{e}}\big(\mathcal{C}^{(\mathsf{M},n)*}\big) \le P_{\mathrm{e}}\big(\mathcal{C}^{(\mathsf{M},n)}\big) \quad (25)$$

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

V. Flip Codes and Weak Flip Codes

We next introduce some special codebooks that will prove instrumental in developing the optimal codes.

Definition 8: The *flip code of type* t, $\mathcal{C}_t^{(2,n)}$, 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:

$$\mathcal{C}_t^{(2,n)} = \begin{pmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{pmatrix} \triangleq \begin{pmatrix}\mathbf{x}\\ \bar{\mathbf{x}}\end{pmatrix} = \begin{pmatrix}0&\cdots&0&\underbrace{1&\cdots&1}_{t\text{ columns}}\\ 1&\cdots&1&0&\cdots&0\end{pmatrix}. \quad (26)$$

Defining the column vectors

$$\left\{\mathbf{c}_1^{(2)} \triangleq \begin{pmatrix}0\\1\end{pmatrix},\ \mathbf{c}_2^{(2)} \triangleq \begin{pmatrix}1\\0\end{pmatrix}\right\} \quad (27)$$

we see that a flip code of type t is given by a codebook matrix consisting of first $n-t$ 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 $\mathcal{C}_t^{(2,n)}$, but in different order are equivalent to $\mathcal{C}_t^{(2,n)}$. 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 (27) are called *candidate columns*. They are flipped versions of each other, therefore also the name of the code.

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. Motivated by (27) and noting that these candidate columns have an equal number of zeros and ones, we give the following definition.

Definition 9: For an $\mathsf{M} \ge 2$, 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 $\lceil\frac{\mathsf{M}}{2}\rceil$.

Accordingly, a weak flip column contains an equal or at least almost equal number of zeros and ones. Note, however, that only $\mathbf{c}_1^{(2)}$ in (27) is a weak flip column.

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

Definition 10: A *weak flip code* is defined by a codebook matrix that is constructed solely by weak flip columns.

Note that for $\mathsf{M} = 2$, only the flip code of type 0 also is a weak flip code, all other flip codes are not weak flip codes, i.e., the definition of weak flip codes is only useful for $\mathsf{M} > 2$.

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

Definition 11: A *weak flip code of type* (t_2,t_3), $\mathcal{C}_{t_2,t_3}^{(\mathsf{M},n)}$, with $\mathsf{M} = 3$ or $\mathsf{M} = 4$ codewords is defined by a codebook matrix

[5]We will see in Section V that both codes are *weak flip codes*. In this example, $\mathcal{C}_1^{(4,4)} = \mathcal{C}_{1,0}^{(4,4)}$ and $\mathcal{C}_2^{(4,4)} = \mathcal{C}_{2,0}^{(4,4)}$ according to Definition 11 given later.

consisting of first $t_1 \triangleq n - t_2 - t_3$ 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

$$\left\{ \mathbf{c}_1^{(3)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \mathbf{c}_2^{(3)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \mathbf{c}_3^{(3)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\} \tag{28}$$

or

$$\left\{ \mathbf{c}_1^{(4)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \mathbf{c}_2^{(4)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \mathbf{c}_3^{(4)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \right\} \tag{29}$$

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

$$[t_1, t_2, t_3] \tag{30}$$

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

$$\mathcal{D}_{t_2,t_3;m}^{(\mathsf{M},n)} \triangleq \{\mathbf{y} \colon g(\mathbf{y}) = m\} \tag{31}$$

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

Note that, as already discussed in Remark 6, the order of these columns does not matter with regard to 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 $\mathbf{x}_1, \ldots, \mathbf{x}_{\mathsf{M}}$.

An interesting subfamily of weak flip codes is defined as follows.

Definition 12: A *fair weak flip code* of type $(t_2, t_3), \mathcal{C}_{t_2,t_3}^{(\mathsf{M},n)}$, with $\mathsf{M} = 3$ or $\mathsf{M} = 4$ codewords satisfies that

$$t_1 = t_2 = t_3. \tag{32}$$

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

$$t_2 = \left\lfloor \frac{n+1}{3} \right\rfloor, \quad t_3 = \left\lfloor \frac{n}{3} \right\rfloor. \tag{33}$$

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

The following lemma follows straightforwardly from the respective definitions. We therefore omit its proof.

Lemma 13: The pairwise Hamming distance vector of the weak flip code $\mathcal{C}_{t_2,t_3}^{(\mathsf{M},n)}$ for $\mathsf{M} = 3$ or $\mathsf{M} = 4$ is given as follows:

$$\mathbf{d}\big(\mathcal{C}_{t_2,t_3}^{(3,n)}\big) = (t_2 + t_3,\ t_1 + t_3,\ t_1 + t_2) \tag{34}$$

$$\mathbf{d}\big(\mathcal{C}_{t_2,t_3}^{(4,n)}\big) = (t_2 + t_3,\ t_1 + t_3,\ t_1 + t_2,\ t_1 + t_2,\ t_1 + t_3,\ t_2 + t_3). \tag{35}$$

VI. Previous Work

A. SGB Bounds on the Average Error Probability

In [15], Shannon, Gallager, and Berlekamp derive upper and lower bounds on the average error probability of a given code used on a DMC. We next quickly review their results.

Definition 14: For $0 < s < 1$, we define

$$\mu_{\alpha,\beta}(s) \triangleq \log \sum_y P_{Y|X}(y|\alpha)^{1-s} P_{Y|X}(y|\beta)^s. \tag{36}$$

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

$$\mathsf{D}^{(\mathrm{DMC})}(m, m') \triangleq -\min_{0 \le s \le 1} \sum_\alpha \sum_\beta q_{\alpha,\beta}(m, m') \mu_{\alpha,\beta}(s) \tag{37}$$

with $q_{\alpha,\beta}(m, m')$ given in Definition 4.

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

Definition 15: The *minimum discrepancy* $\mathsf{D}_{\min}^{(\mathrm{DMC})}(\mathcal{C}^{(\mathsf{M},n)})$ for a codebook is the minimum value of $\mathsf{D}^{(\mathrm{DMC})}(m, m')$ over all pairs of codewords. The *maximum minimum discrepancy* is the maximum value of $\mathsf{D}_{\min}^{(\mathrm{DMC})}(\mathcal{C}^{(\mathsf{M},n)})$ over all possible $\mathcal{C}^{(\mathsf{M},n)}$ codebooks: $\max_{\mathcal{C}^{(\mathsf{M},n)}} \mathsf{D}_{\min}^{(\mathrm{DMC})}(\mathcal{C}^{(\mathsf{M},n)})$.

Theorem 16 (SGB Bounds on Average Error Probability [15]): For an arbitrary DMC, the average error probability $P_{\mathrm{e}}\big(\mathcal{C}^{(\mathsf{M},n)}\big)$ of a given code $\mathcal{C}^{(\mathsf{M},n)}$ with M codewords and blocklength n is upper- and lower-bounded as follows:

$$\frac{1}{4\mathsf{M}} e^{-n\left(\mathsf{D}_{\min}^{(\mathrm{DMC})}(\mathcal{C}^{(\mathsf{M},n)}) + \sqrt{\frac{2}{n}} \log \frac{1}{P_{\min}}\right)} \le P_{\mathrm{e}}\big(\mathcal{C}^{(\mathsf{M},n)}\big) \le (\mathsf{M} - 1) e^{-n\mathsf{D}_{\min}^{(\mathrm{DMC})}(\mathcal{C}^{(\mathsf{M},n)})} \tag{38}$$

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

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

The bounds (38) are tight enough to derive the *error exponent* of the DMC (for a fixed number M of codewords).

Theorem 17 ([15]): The error exponent of a DMC for a fixed number M of codewords

$$\mathsf{E}_{\mathsf{M}} \triangleq \overline{\lim_{n\to\infty}} \max_{\mathcal{C}^{(\mathsf{M},n)}} \left\{ -\frac{1}{n} \log P_{\mathrm{e}}\big(\mathcal{C}^{(\mathsf{M},n)}\big) \right\} \tag{39}$$

is given as

$$\mathsf{E}_{\mathsf{M}} = \lim_{n\to\infty} \max_{\mathcal{C}^{(\mathsf{M},n)}} \mathsf{D}_{\min}^{(\mathrm{DMC})}\big(\mathcal{C}^{(\mathsf{M},n)}\big). \tag{40}$$

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

$$\mathsf{E}_2 = \max_{\mathcal{C}^{(2,n)}} \mathsf{D}_{\min}^{(\mathrm{DMC})}\big(\mathcal{C}^{(2,n)}\big) = \max_{\alpha,\beta} \left\{ -\min_{0 \le s \le 1} \mu_{\alpha,\beta}(s) \right\}. \tag{41}$$

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

Definition 18: A *pairwise reversible channel* is a DMC that has $\frac{\mathrm{d}}{\mathrm{d}s}\mu_{\alpha,\beta}(s)\big|_{s=\frac{1}{2}} = 0$ for any inputs α, β.

Clearly, the BSC is a pairwise reversible channel.

Note that it is easy to compute the pairwise discrepancy of a linear code on a pairwise reversible channel, so linear codes are quite suitable for computing (38).

Theorem 19 ([15]): For pairwise reversible channels with $\mathsf{M} > 2$,

$$\mathsf{E}_{\mathsf{M}} = \frac{1}{\mathsf{M}(\mathsf{M}-1)} \max_{\substack{\mathsf{M}_x \text{ s.t.} \\ \sum_x \mathsf{M}_x = \mathsf{M}}} \sum_{\substack{\text{all input} \\ \text{letters } x}} \sum_{\substack{\text{all input} \\ \text{letters } x'}} \mathsf{M}_x \mathsf{M}_{x'} \cdot \left(-\log \sum_y \sqrt{P_{Y|X}(y|x) P_{Y|X}(y|x')} \right) \quad (42)$$

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

We would like to emphasize that while Shannon *et al.* proved that fair weak flip codes achieve the error exponent, they did not investigate the error performance of fair weak flip codes for finite n. As we will show later, fair weak flip codes might be strictly suboptimal for finite n (see also [16]).

B. Gallager Bound

Another famous bound is by Gallager [6].

Theorem 20 ([6]): For an arbitrary DMC, there exists a code $\mathcal{C}^{(\mathsf{M},n)}$ with $\mathsf{M} = \lfloor e^{n\mathsf{R}} \rfloor$ such that

$$P_{\mathrm{e}}\big(\mathcal{C}^{(\mathsf{M},n)}\big) \le e^{-n\mathsf{E}_{\mathrm{G}}(\mathsf{R})} \quad (43)$$

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

$$\mathsf{E}_{\mathrm{G}}(\mathsf{R}) = \max_{Q(\cdot)} \max_{0 \le \rho \le 1} \big\{ \mathsf{E}_0(\rho, Q) - \rho \mathsf{R} \big\} \quad (44)$$

with

$$\mathsf{E}_0(\rho, Q) \triangleq -\log \left(\sum_y \left(\sum_x Q(x) P_{Y|X}(y|x)^{\frac{1}{1+\rho}} \right)^{1+\rho} \right). \quad (45)$$

C. PPV Bounds for the BSC

In [2], Polyanskiy, Poor, and Verdú present upper and lower bounds on the optimal average error probability for finite blocklength for the BSC. The upper bound is based on *random coding*. It is the exact random coding error expression for the BSC by using an alternative way compared to [17].

Theorem 21 (PPV Upper Bound [17, Th. 2], [2, Th. 32]): If the codebook $\mathcal{C}^{(\mathsf{M},n)}$ is created at random based on a uniform distribution, the expected average error probability (averaged over all codewords and all codebooks) satisfies

$$\mathsf{E}_{\mathcal{C}^{(\mathsf{M},n)}}\big[P_{\mathrm{e}}\big(\mathcal{C}^{(\mathsf{M},n)}\big)\big] = 1 - 2^{n-n\mathsf{M}} \sum_{i=0}^{n} \binom{n}{i} \epsilon^i (1-\epsilon)^{n-i} \cdot \left(\sum_{m=0}^{\mathsf{M}-1} \frac{1}{m+1} \binom{\mathsf{M}-1}{m} \binom{n}{i}^m \left(\sum_{j=i+1}^{n} \binom{n}{j} \right)^{\mathsf{M}-1-m} \right). \quad (46)$$

Note that there must exist a codebook whose average error probability achieves (46), so Theorem 21 provides a general achievable upper bound on the error probability, although we do not know the concrete code structure.

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

Theorem 22 (PPV Lower Bound [6, p. 163, Eq. (5.8.19)], [2, Theorem 35]): Any codebook $\mathcal{C}^{(\mathsf{M},n)}$ satisfies

$$P_{\mathrm{e}}\big(\mathcal{C}^{(\mathsf{M},n)}\big) \ge \left(\binom{n}{\mathsf{N}} - \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \mathsf{A}_{m,\mathsf{N}} \right) \epsilon^{\mathsf{N}} (1-\epsilon)^{n-\mathsf{N}} + \sum_{j=\mathsf{N}+1}^{n} \binom{n}{j} \epsilon^j (1-\epsilon)^{n-j} \quad (47)$$

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

$$\mathsf{A}_{m,j} = \begin{cases} \binom{n}{j} & 0 \le j \le \mathsf{N}-1 \\ 0 & \mathsf{N}+1 \le j \le n \end{cases} \quad (48)$$

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

$$\mathsf{M} \sum_{j=0}^{\mathsf{N}-1} \mathsf{A}_{m,j} + \sum_{m=1}^{\mathsf{M}} \mathsf{A}_{m,\mathsf{N}} = 2^n \quad (49)$$

$$0 < \sum_{m=1}^{\mathsf{M}} \mathsf{A}_{m,\mathsf{N}} \le \mathsf{M} \binom{n}{\mathsf{N}}. \quad (50)$$

VII. Analysis of the BAC

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

A. Optimal Codes

Theorem 23: Consider a BAC and a blocklength n. Then, irrespective of the channel parameters ϵ_0 and ϵ_1, there exists a choice of t, $0 \le t \le \lfloor \frac{n}{2} \rfloor$, such that the flip code of type t, $\mathcal{C}_t^{(2,n)}$, is optimal in the sense that it minimizes the average error probability.

Proof: Consider an arbitrary code with $\mathsf{M} = 2$ codewords and a blocklength $n + j$, and assume that this code is not a

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

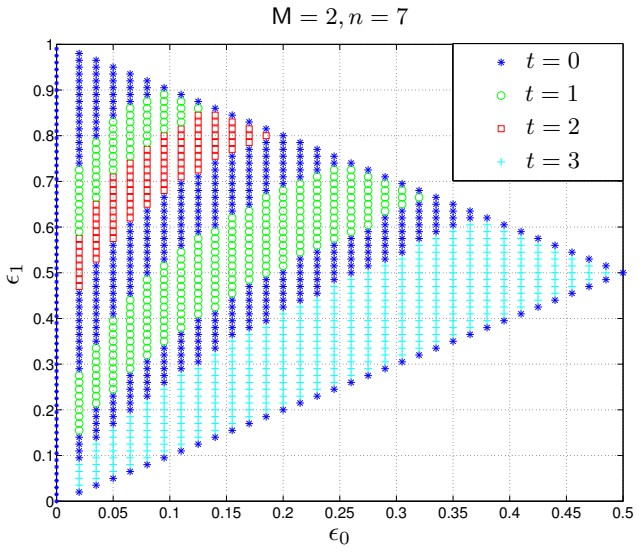

Fig. 5. Optimal codebooks on a BAC: the optimal choice of the parameter t for different values of ϵ_0 and ϵ_1 for a fixed blocklength $n = 7$.

flip code, but that it has a number j of positions where both codewords have the same symbol. An optimal decoder will ignore these j positions completely. Hence, the performance of this code will be identical to a flip code of length n. Now, change this code in the j positions with identical symbol such that the code becomes a flip code of length $n + j$. If we use a suboptimal decoder that ignores these j positions we still keep the same performance. However, an ML decoder can potentially improve the performance, i.e., we have

$$P_{\mathrm{e}}\big(\mathcal{C}^{(\mathsf{M},n+j)}_{\text{not flip}}\big)_{\text{ML decoder}} = P_{\mathrm{e}}\big(\mathcal{C}^{(\mathsf{M},n+j)}_{\text{flip}}\big)_{\text{suboptimal decoder}} \tag{51}$$

$$\geq P_{\mathrm{e}}\big(\mathcal{C}^{(\mathsf{M},n+j)}_{\text{flip}}\big)_{\text{ML decoder}}. \tag{52}$$

An alternative proof is shown in Appendix A-B. While this proof is more elaborate, it turns out to be useful for the derivation of Theorem 25. ■

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

We would like to point out that the exact choice of t is not obvious and depends strongly on n, ϵ_0, and ϵ_1. As an example, the optimal choices of t are shown in Fig. 5 for $n = 7$ as a function of ϵ_0 and ϵ_1. 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 29 it will be shown that the flip code of type 0 is optimal on the ZC; and in Theorem 36 it will be shown that the flip codes are optimal on the BSC for any choice of t. We defer the exact treatment of the ZC and the BSC to Section VIII and IX, respectively.

B. 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, $\mathcal{C}_t^{(2,n)}$, the LLR is given as

$$\begin{aligned}
&\log\left(\frac{P^n_{Y|X}(\mathbf{y}|\mathbf{x}_1)}{P^n_{Y|X}(\mathbf{y}|\mathbf{x}_2)}\right)\\
&= \log\left(\frac{\left(\frac{\epsilon_0}{1-\epsilon_0}\right)^{d_{0,1}(\mathbf{x}_1,\mathbf{y})}\left(\frac{\epsilon_1}{1-\epsilon_0}\right)^{d_{1,0}(\mathbf{x}_1,\mathbf{y})}}{\left(\frac{\epsilon_0}{1-\epsilon_0}\right)^{t-d_{1,0}(\mathbf{x}_1,\mathbf{y})}\left(\frac{\epsilon_1}{1-\epsilon_0}\right)^{n-t-d_{0,1}(\mathbf{x}_1,\mathbf{y})}}\right.\\
&\qquad\left.\cdot\frac{\left(\frac{1-\epsilon_1}{1-\epsilon_0}\right)^{t-d_{1,0}(\mathbf{x}_1,\mathbf{y})}}{\left(\frac{1-\epsilon_1}{1-\epsilon_0}\right)^{d_{0,1}(\mathbf{x}_1,\mathbf{y})}}\right) \qquad (53)\\
&= \big(t - d_{0,1}(\mathbf{x}_1,\mathbf{y}) - d_{1,0}(\mathbf{x}_1,\mathbf{y})\big)\log\left(\frac{1-\epsilon_1}{\epsilon_0}\right)\\
&\quad + \big(n - t - d_{0,1}(\mathbf{x}_1,\mathbf{y}) - d_{1,0}(\mathbf{x}_1,\mathbf{y})\big)\log\left(\frac{1-\epsilon_0}{\epsilon_1}\right) \qquad (54)\\
&= (t-d)\log\left(\frac{1-\epsilon_1}{\epsilon_0}\right) + (n-t-d)\log\left(\frac{1-\epsilon_0}{\epsilon_1}\right) \qquad (55)\\
&\triangleq \mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d) \qquad (56)
\end{aligned}$$

where we have defined

$$d \triangleq d_{0,1}(\mathbf{x}_1,\mathbf{y}) + d_{1,0}(\mathbf{x}_1,\mathbf{y}) = d_{\mathrm{H}}(\mathbf{x}_1,\mathbf{y}) \tag{57}$$

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

$$\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)\ \begin{cases}\geq 0 \implies g(\mathbf{y}) = 1\\ < 0 \implies g(\mathbf{y}) = 2.\end{cases} \tag{58}$$

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

This seems to point towards a simplification of (58): instead of computing the log-likelihood ratio, we only need to consider d. This indeed is the case. From Properties 2) and 3) of Proposition 40 in Appendix A-A, it follows directly that the ML decision rule for a flip code is a *threshold rule*.

Theorem 24 (Threshold Rule): For every flip code $\mathcal{C}_t^{(2,n)}$ and every BAC $(\epsilon_0,\epsilon_1) \in \Omega$, there exists a *threshold* ℓ, $t \leq \ell \leq \lfloor\frac{n-1}{2}\rfloor$, such that the ML decision rule can be stated as

$$g(\mathbf{y}) = \begin{cases}1 & \text{if } 0 \leq d \leq \ell\\ 2 & \text{if } \ell+1 \leq d \leq n.\end{cases} \tag{59}$$

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

$$\begin{aligned}\Omega_{\ell,t}^{(n)} \triangleq \big\{(\epsilon_0,\epsilon_1)\colon &\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,\ell) \geq 0 \text{ and}\\ &\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,\ell+1) \leq 0\big\}.\end{aligned} \tag{60}$$

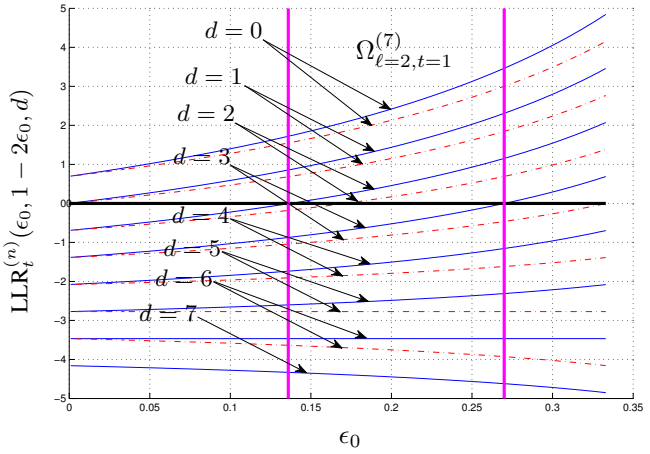

Fig. 6. Log-likelihood ratio $\mathrm{LLR}_t^{(n)}(\epsilon_0, \epsilon_1 = 1-2\epsilon_0, d)$ for $\mathcal{C}_1^{(2,n)}$ (i.e., $t=1$) as a function of ϵ_0 for different values of d. The solid blue lines correspond to $n=7$, the dashed red lines to $n=6$. Observe that for $n=7$ and $\epsilon_0 \in [0.136, 0.270]$ (i.e., the region between the two vertical purple lines), the threshold for the optimal ML decision rule is $\ell=2$, see Theorem 24.

C. Best Codes for a Fixed Decision Rule

Our original goal was to find the optimal code for a given channel (ϵ_0, ϵ_1). We have shown that this is equivalent to finding an optimal t. Unfortunately, this search is difficult because the borders between the regions of different optimal t (see, e.g., Fig. 5) are defined by the combined influences of two different forces: when varying (ϵ_0, ϵ_1), either the optimal code $\mathcal{C}_t^{(2,n)}$ changes, but the optimal threshold ℓ remains the same, or the optimal choice of ℓ changes, too. Hence, a joint optimization of t and ℓ is necessary.

We now simplify the problem by fixing the decision rule (i.e., the threshold ℓ) and then search for the *best* code $\mathcal{C}_t^{(2,n)}$ 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 links the roots of the LLR-function with choices of parameters for which two different codes have identical error probability. This will then allow us to find the boundaries where the best codes under a fixed decision rule change from $t+1$ to t (see also Fig. 8 below).

Theorem 25: 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

$$P_{\mathrm{e}}^{(\ell)}\big(\mathcal{C}_t^{(2,n)}\big) - P_{\mathrm{e}}^{(\ell)}\big(\mathcal{C}_{t+1}^{(2,n)}\big) = 0 \tag{61}$$

are identical to the roots of

$$\mathrm{LLR}_t^{(n-1)}(\epsilon_0, \epsilon_1, \ell) = 0 \tag{62}$$

where $P_{\mathrm{e}}^{(\ell)}\big(\mathcal{C}_t^{(2,n)}\big)$ denotes the error probability of code $\mathcal{C}_t^{(2,n)}$ 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 (61) holds; and for a fixed $\epsilon_1 \in \Omega$, there exists at most one $\epsilon_0 \in \Omega$ such that (61) holds. This means that if (61) has a solution, then this solution is unique for a fixed ϵ_0 or ϵ_1.

Proof: See Appendix A-C. ∎

Using Theorem 25 and Proposition 40, we can now state conditions on t such that $\mathcal{C}_t^{(2,n)}$ is best under a fixed decision rule ℓ.

Corollary 26: Fix a blocklength n and a decision rule ℓ. Then the flip code of type t, $\mathcal{C}_t^{(2,n)}$, is uniquely best for a fixed decision rule ℓ if and only if (ϵ_0, ϵ_1) belongs to

$$\big\{(\epsilon_0, \epsilon_1)\colon \mathrm{LLR}_t^{(n-1)}(\epsilon_0, \epsilon_1, \ell) > 0 \text{ and } \mathrm{LLR}_{t-1}^{(n-1)}(\epsilon_0, \epsilon_1, \ell) < 0\big\}. \tag{63}$$

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

Proof: From (140) in the proof of Theorem 25 in Appendix A-C and from assumption (1) it follows that

$$\mathrm{LLR}_t^{(n-1)}(\epsilon_0, \epsilon_1, \ell) > 0 \iff P_{\mathrm{e}}^{(\ell)}\big(\mathcal{C}_t^{(2,n)}\big) < P_{\mathrm{e}}^{(\ell)}\big(\mathcal{C}_{t+1}^{(2,n)}\big). \tag{64}$$

As we know from Proposition 40 that $\mathrm{LLR}_t^{(n-1)}(\epsilon_0, \epsilon_1, \ell)$ is increasing in t, this means that if both (64) and

$$\mathrm{LLR}_{t-1}^{(n-1)}(\epsilon_0, \epsilon_1, \ell) < 0 \tag{65}$$

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

We illustrate Corollary 26 by an example. We fix $n=7$, $\ell=2$, $\epsilon_1=0.5$, and let ϵ_0 increase from 0 to $\min\{\epsilon_1, 1-\epsilon_1\} = 0.5$, see Fig. 7. Starting with $t=3$, we check that

$$\mathrm{LLR}_2^{(6)}(\epsilon_0, 0.5, 2) > 0 \tag{66}$$

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

$$\mathrm{LLR}_1^{(6)}(\epsilon_0, 0.5, 2) < 0 \tag{67}$$

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

1) $\mathcal{C}_2^{(2,7)}$ is best in $\big\{\epsilon_0\colon \mathrm{LLR}_2^{(6)}(\epsilon_0, 0.5, 2) > 0$ and $\mathrm{LLR}_1^{(6)}(\epsilon_0, 0.5, 2) < 0\big\}$;
2) $\mathcal{C}_1^{(2,7)}$ is best in $\big\{\epsilon_0\colon \mathrm{LLR}_1^{(6)}(\epsilon_0, 0.5, 2) > 0$ and $\mathrm{LLR}_0^{(6)}(\epsilon_0, 0.5, 2) < 0\big\}$;
3) $\mathcal{C}_0^{(2,7)}$ is best in $\big\{\epsilon_0\colon \mathrm{LLR}_0^{(6)}(\epsilon_0, 0.5, 2) > 0\big\}$.

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

Corollary 26 shows that for a fixed decision rule ℓ, the choice of the best code parameter t depending on the given parameters n, ϵ_0, and ϵ_1 is much easier than the choice of the jointly optimal t and ℓ for a globally optimal code. In particular, we have the following regular structure.

Corollary 27: Fix a blocklength n and a decision rule ℓ, and consider a BAC. If we increase ϵ_0 or decrease ϵ_1, then the best value of t is nonincreasing.

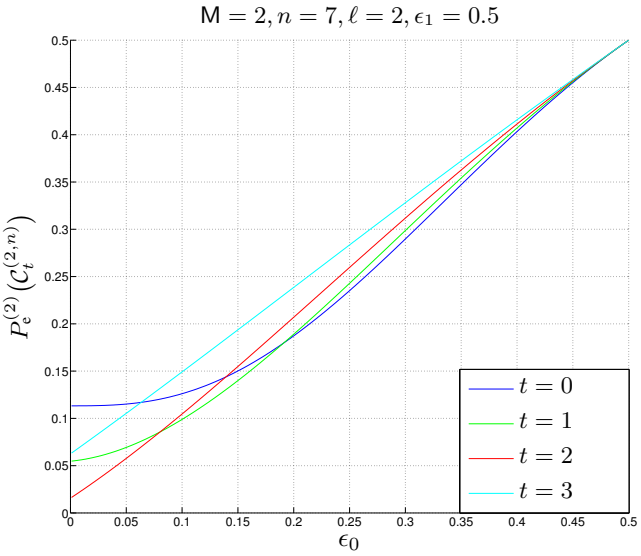

Fig. 7. Error probabilities of all possible flip codes $\mathcal{C}_t^{(2,n)}$ as a function of the channel parameter ϵ_0, for a fixed blocklength $n = 7$, $\epsilon_1 = 0.5$, and a fixed decision rule $\ell = 2$. For any ϵ_0, the best code is the one with the smallest error probability value.

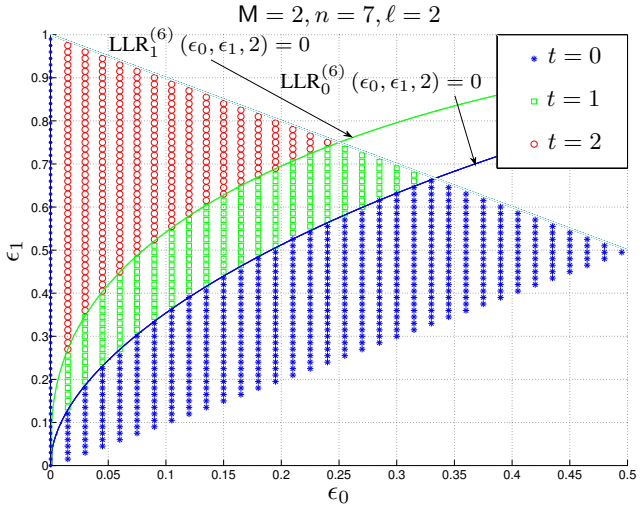

Fig. 8. Best codebooks on a BAC for a fixed decision rule: for all possible (ϵ_0, ϵ_1) this plot shows the best choice of the code parameter t. The blocklength is $n = 7$ and the decision rule is $\ell = 2$.

More sloppily we can say that when we are moving inside of Ω (see Fig. 2) to the right or downwards, the best t will either remain the same or be reduced by 1. This is in stark contrast to the picture of the regions of optimal codes where the optimal t changes in a seemingly random manner. For an illustration, compare the best codes for a fixed decision rule $\ell = 2$ in Fig. 8 with the corresponding globally optimal regions of Fig. 5.

Even more importantly, Theorem 25 also allows us to locate the exact location of some of the boundaries between the different areas of *globally optimal* codes (Fig. 5).

Corollary 28: Consider the boundary between two areas of globally optimal codes (as, e.g., shown in Fig. 5). If the optimal decision rule on both sides of the boundary takes the same value ℓ and if the optimal code on the left is $t + 1$, while the optimal code on the right is t, then this boundary is

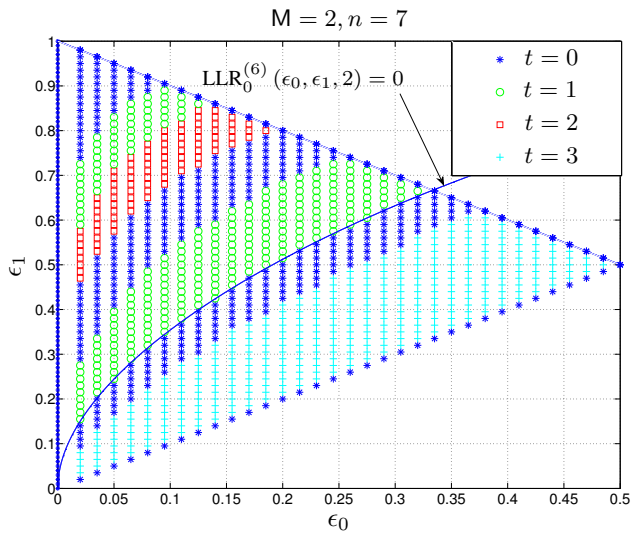

Fig. 9. Globally optimal codebooks on a BAC for a blocklength $n = 7$ (identical to Fig. 5). The shown boundary between $t = 1$ and $t = 0$ is identical to the corresponding boundary given in Fig. 8, where a fixed decision rule $\ell = 2$ has been assumed.

identical to a corresponding boundary in the situation with a fixed decision rule ℓ. In particular, this boundary is given by the roots of $\mathrm{LLR}_t^{(n-1)}(\epsilon_0, \epsilon_1, \ell)$.

We again show the example of $n = 7$ from Fig. 5: in Fig. 9 the same plot is shown including a boundary that is identical to a boundary given in Fig. 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. Such a search can be summarized by the following algorithm.

- Step 0: Fix a channel (ϵ_0, ϵ_1) and find the best t under the fixed decision rule $\ell = 0$ and its corresponding error probability $p \triangleq P_{\mathrm{e}}^{(0)}\big(\mathcal{C}_t^{(2,n)}\big)$. Then set $\ell \triangleq 1$.
- Step 1: Find the best t_{temp} under a fixed decision rule ℓ and the corresponding error probability $P_{\mathrm{e}}^{(\ell)}\big(\mathcal{C}_{t_{\mathrm{temp}}}^{(2,n)}\big)$.
- Step 2: Check whether $P_{\mathrm{e}}^{(\ell)}\big(\mathcal{C}_{t_{\mathrm{temp}}}^{(2,n)}\big) < p$. If yes, set $t \triangleq t_{\mathrm{temp}}$ and $p \triangleq P_{\mathrm{e}}^{(\ell)}\big(\mathcal{C}_{t_{\mathrm{temp}}}^{(2,n)}\big)$.
- 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).

VIII. Analysis of the ZC

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

A. Optimal Codes With Two Codewords ($\mathsf{M} = 2$)

Theorem 29: 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, $\mathcal{C}_0^{(2,n)}$. It has an error probability

$$P_{\mathrm{e}}\big(\mathcal{C}_0^{(2,n)}\big) = \frac{1}{2}\epsilon_1^n. \tag{68}$$

Proof: Due to Theorem 23, we can restrict our search to flip codes of some type t, $\mathcal{C}_t^{(2,n)}$, 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 $\mathbf{y} = \mathbf{0}$:

$$\min\left\{P_{Y|X}^n(\mathbf{y}|\mathbf{x}_1), P_{Y|X}^n(\mathbf{y}|\mathbf{x}_2)\right\} = \begin{cases} 0 & \text{if } \mathbf{y} \neq \mathbf{0} \\ \epsilon_1^{\max\{w_{\mathrm{H}}(\mathbf{x}_1), w_{\mathrm{H}}(\mathbf{x}_2)\}} & \text{if } \mathbf{y} = \mathbf{0}. \end{cases} \tag{69}$$

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

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

B. Optimal Codes With Three or Four Codewords ($\mathsf{M} = 3, 4$)

Before we describe how we address the optimal codes with 3 or 4 codewords for a ZC, we first show that an optimal code must contain the all-zero vector $\mathbf{0}$ as a codeword.

Theorem 30 (Sufficient Set of Candidate Columns for the ZC): For a ZC, for any blocklength n, and for an arbitrary number M of codewords, an optimal codebook must contain the all-zero codeword $\mathbf{0}$.

Proof: See Appendix B-A. ∎

Next we show that the weak flip codes of type (t_2, t_3) are optimal codes with three or four codewords for a ZC.

Theorem 31: For a ZC and for any $n \geq 2$, an optimal codebook with three codewords $\mathsf{M} = 3$ or four codewords $\mathsf{M} = 4$ is the weak flip code of type $(t^*, 0)$, $\mathcal{C}_{t^*,0}^{(\mathsf{M},n)}$, with

$$t^* \triangleq \left\lfloor \frac{n}{2} \right\rfloor. \tag{70}$$

Moreover, the optimal code achieves the average error probability

$$P_{\mathrm{e}}\big(\mathcal{C}_{t^*,0}^{(\mathsf{M},n)}\big) = \begin{cases} \frac{1}{3}\big(\epsilon_1^{t^*} + \epsilon_1^{n-t^*}\big) & \text{if } \mathsf{M} = 3 \\ \frac{1}{4}\big(2\epsilon_1^{t^*} + 2\epsilon_1^{n-t^*} - \epsilon_1^n\big) & \text{if } \mathsf{M} = 4. \end{cases} \tag{71}$$

Proof: See Appendix B-B. ∎

Similarly to the case of $\mathsf{M} = 2$, we see that for $\mathsf{M} = 4$ the optimal code given in Theorem 31 is linear. Also note that from the discussion in Appendix B-B 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.

For $\mathsf{M} = 3$, the optimal codes are not unique. Indeed any choice of t_2 and t_3 with $t_2 + t_3 = t^*$ is optimal.

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 $P_{\mathrm{e}}\big(\mathcal{C}_{5,0}^{(4,10)}\big) \approx 2.43 \cdot 10^{-3}$. If we increase the blocklength to $n = 20$, we already achieve an average error probability $P_{\mathrm{e}}\big(\mathcal{C}_{10,0}^{(4,20)}\big) \approx 5.90 \cdot 10^{-6}$. The asymptotic behavior of the optimal error probability for n going to infinity will be discussed in next section.

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

We start with the following lemma.

Lemma 32: Fix some arbitrary integers $\mathsf{M} \geq 2$, $n \geq 1$, and $\gamma \geq 1$. Consider a DMC and a code $\mathcal{C}^{(\mathsf{M},n)}$ for this DMC with M codewords and blocklength n, and create a new code $\mathcal{C}^{(\mathsf{M},n+\gamma)}$ by appending γ arbitrary column vectors to the codebook matrix of $\mathcal{C}^{(\mathsf{M},n)}$. Then the average success probability of this new code cannot be smaller than the success probability of the original code:

$$P_{\mathrm{c}}\big(\mathcal{C}^{(\mathsf{M},n+\gamma)}\big) \geq P_{\mathrm{c}}\big(\mathcal{C}^{(\mathsf{M},n)}\big). \tag{72}$$

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

$$P_{\mathrm{c}}\big(\mathcal{C}^{(\mathsf{M},n)}\big) = \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \sum_{\mathbf{y}^{(n)} \in \mathcal{D}_m^{(n)}} P_{\mathbf{Y}|\mathbf{X}}\big(\mathbf{y}^{(n)}\big|\mathbf{x}_m^{(n)}\big). \tag{73}$$

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

$$\mathbf{x}_m^{(n+\gamma)} = \big[\mathbf{x}_m^{(n)}\ \mathbf{x}_m^{(\gamma)}\big] \tag{74}$$

$$\triangleq \big(x_{m,1}\ \ x_{m,2}\ \cdots\ x_{m,n}\ \ x_{m,n+1}\ \cdots\ x_{m,n+\gamma}\big) \tag{75}$$

and likewise the extended received vector by

$$\mathbf{y}^{(n+\gamma)} = \big[\mathbf{y}^{(n)}\ \mathbf{y}^{(\gamma)}\big] \triangleq \big(y_1\ \ y_2\ \cdots\ y_{n+\gamma}\big). \tag{76}$$

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

$$\frac{P_{\mathbf{Y}|\mathbf{X}}\left(\big[\mathbf{y}^{(n)}\ \mathbf{y}^{(\gamma)}\big]\Big|\big[\mathbf{x}_{m'}^{(n)}\ \mathbf{x}_{m'}^{(\gamma)}\big]\right)}{P_{\mathbf{Y}|\mathbf{X}}\left(\big[\mathbf{y}^{(n)}\ \mathbf{y}^{(\gamma)}\big]\Big|\big[\mathbf{x}_m^{(n)}\ \mathbf{x}_m^{(\gamma)}\big]\right)} \geq 1. \tag{77}$$

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

$$P_{\mathrm{c}}\big(\mathcal{C}^{(\mathsf{M},n+\gamma)}\big) = \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \left(\sum_{\mathbf{y}^{(n)} \in \mathcal{D}_m^{(n)}} P_{\mathbf{Y}|\mathbf{X}}\big(\mathbf{y}^{(n)}\big|\mathbf{x}_m^{(n)}\big) \cdot \underbrace{\sum_{\mathbf{y}^{(\gamma)} \in \mathcal{Y}^\gamma} P_{\mathbf{Y}|\mathbf{X}}\big(\mathbf{y}^{(\gamma)}\big|\mathbf{x}_m^{(\gamma)}\big)}_{=1} \right) \tag{78}$$

$$= P_{\mathrm{c}}\big(\mathcal{C}^{(\mathsf{M},n)}\big) \tag{79}$$

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

$$P_{\mathrm{c}}\big(\mathcal{C}^{(\mathsf{M},n+\gamma)}\big) = P_{\mathrm{c}}\big(\mathcal{C}^{(\mathsf{M},n)}\big) + \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \sum_{\substack{\mathbf{y}^{(n+\gamma)} \\ \text{s.t. } \mathbf{y}^{(n)} \in \mathcal{D}_m^{(n)} \\ \text{but } \mathbf{y}^{(n+\gamma)} \in \mathcal{D}_{m'}^{(n+\gamma)}}} \left(P_{\mathbf{Y}|\mathbf{X}}\Big(\mathbf{y}^{(n+\gamma)}\Big|\mathbf{x}_{m'}^{(n+\gamma)}\Big) - P_{\mathbf{Y}|\mathbf{X}}\big(\mathbf{y}^{(n+\gamma)}\big|\mathbf{x}_m^{(n+\gamma)}\big) \right) \tag{80}$$

$$\triangleq P_{\mathrm{c}}\big(\mathcal{C}^{(\mathsf{M},n)}\big) + \Delta\Psi\big(\mathcal{C}^{(\mathsf{M},n+\gamma)}\big). \tag{81}$$

The proof of Lemma 32 is completed by noting that, from (77), $\Delta\Psi\big(\mathcal{C}^{(\mathsf{M},n+\gamma)}\big)$ is always nonnegative. ∎

Definition 33: The term $\Delta\Psi\big(\mathcal{C}^{(\mathsf{M},n+\gamma)}\big)$ in (81) is called *total probability increase for a step-size* γ and describes the amount by which the average success probability of the code $\mathcal{C}^{(\mathsf{M},n)}$ grows when γ column vectors are appended to its codebook matrix.

Lemma 34: For a ZC, for any $n \geq 2$, and for $1 \leq t \leq \lfloor \frac{n}{2} \rfloor$, consider the weak flip code of type $(t, 0)$ with four codewords $\mathsf{M} = 4$, $\mathcal{C}_{t,0}^{(4,n)}$, and append a column to the codebook matrix to create a new code of length $n+1$. Then the total probability increase is maximized if, among all possible $2^4 = 16$ columns, we choose $\mathbf{c}_2^{(4)}$. If $t < \lfloor \frac{n}{2} \rfloor$, or if n is odd and $t = \lfloor \frac{n}{2} \rfloor$, then this choice is unique.

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

Proof: See Appendix B-C. ∎

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

$$\mathcal{C}_{t,0}^{(4,n)} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{x} \\ \bar{\mathbf{x}} \\ \mathbf{1} \end{pmatrix} \tag{82}$$

with $\mathbf{x}$ and $\bar{\mathbf{x}}$ defined in (26).

From the recursive technique that we have used in the derivation of Lemma 34 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 35: The optimal codebooks defined in Theorem 31 for $\mathsf{M} = 3$ and $\mathsf{M} = 4$ can be constructed recursively in the blocklength n by adding a column that yields the maximum total probability increase. We start with an optimal codebook for $n = 2$:

$$\mathcal{C}_{\mathrm{ZC}}^{(\mathsf{M},2)*} = \begin{pmatrix} \mathbf{c}_1^{(\mathsf{M})} & \mathbf{c}_2^{(\mathsf{M})} \end{pmatrix}. \tag{83}$$

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

$$\begin{cases} \mathbf{c}_1^{(\mathsf{M})} & \text{if } n \bmod 2 = 1 \\ \mathbf{c}_2^{(\mathsf{M})} & \text{if } n \bmod 2 = 0. \end{cases} \tag{84}$$

Proof: We only need to show that the constructed codes from (84) are equivalent to the optimal codes given in Theorem 31. The optimal code for $\mathsf{M} = 4$ and $n = 2$ is trivial and given by (83). Next assume that for blocklength n, $\mathcal{C}_{\lfloor \frac{n}{2} \rfloor,0}^{(4,n)}$ is optimal. From Lemma 34 we know that the largest total probability increase is achieved when adding column $\mathbf{c}_2^{(4)}$. Now note that for n even with $t = \frac{n}{2}$, adding the column $\mathbf{c}_2^{(4)}$ to the code $\mathcal{C}_{\frac{n}{2},0}^{(4,n)}$ will result in a code that is equivalent to $\mathcal{C}_{\lfloor \frac{n+1}{2} \rfloor,0}^{(4,n+1)}$: we only need to exchange the roles of the second and third codeword and then re-order the columns. For n odd with $t = \lfloor \frac{n}{2} \rfloor$, adding the column $\mathbf{c}_2^{(4)}$ to the code $\mathcal{C}_{\lfloor \frac{n}{2} \rfloor,0}^{(4,n)}$ results in $\mathcal{C}_{\frac{n+1}{2},0}^{(4,n+1)}$.

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

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

We conclude this section by a remark. While it is very intuitive to construct the codes recursively, i.e., to start from an optimal code for n and then to add one column that maximizes the total probability increase, unfortunately, from a proof perspective, such a recursive construction only guarantees local optimality: one still needs a proof (Theorem 31) that the achieved code of blocklength $n + 1$ is globally optimum.

C. Error Exponents

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

$$\mathsf{D}_{\min}^{(\mathrm{ZC})}\big(\mathcal{C}_{t^*,0}^{(\mathsf{M},n)}\big) = \begin{cases} -\frac{1}{2}\log\epsilon_1 & \text{if } n \bmod 2 = 0 \\ -\frac{\lfloor \frac{n}{2} \rfloor}{n}\log\epsilon_1 & \text{if } n \bmod 2 = 1 \end{cases} \tag{85}$$

we can now compute the error exponents:

$$\mathsf{E}_3 = \mathsf{E}_4 = -\frac{1}{2}\log\epsilon_1. \tag{86}$$

Note that the minimum discrepancy $\mathsf{D}_{\min}^{(\mathrm{ZC})}(\mathcal{C}^{(\mathsf{M},n)})$ for the generalized fair weak flip code for every n is

$$\mathsf{D}_{\min}^{(\mathrm{ZC})}\Big(\mathcal{C}_{\lfloor \frac{n+1}{3} \rfloor, \lfloor \frac{n}{3} \rfloor}^{(\mathsf{M},n)}\Big) = \begin{cases} -\frac{1}{3}\log\epsilon_1 & \text{if } n \bmod 3 = 0 \\ -\frac{\lfloor \frac{n}{3} \rfloor + 1}{n}\log\epsilon_1 & \text{if } n \bmod 3 = 1 \\ -\frac{\lfloor \frac{n}{3} \rfloor + 1}{n}\log\epsilon_1 & \text{if } n \bmod 3 = 2. \end{cases} \tag{87}$$

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

Since we now know the optimal code structure and its performance, it is interesting to compare it to the known bounds described in Section VI. Fig. 10 and Fig. 11 compare some SGB bounds and the Gallager bound with the exact performance of the optimal code (for $\mathsf{M} = 3$ and $\mathsf{M} = 4$ codewords, respectively). Besides the Gallager bound, we plot the SGB lower bound based on the optimal code structure (thereby making sure that this lower bound is valid generally), and we show two SGB upper bounds: one that is based on the optimal code design and one that is based on the fair weak flip code used by Shannon *et al.*

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

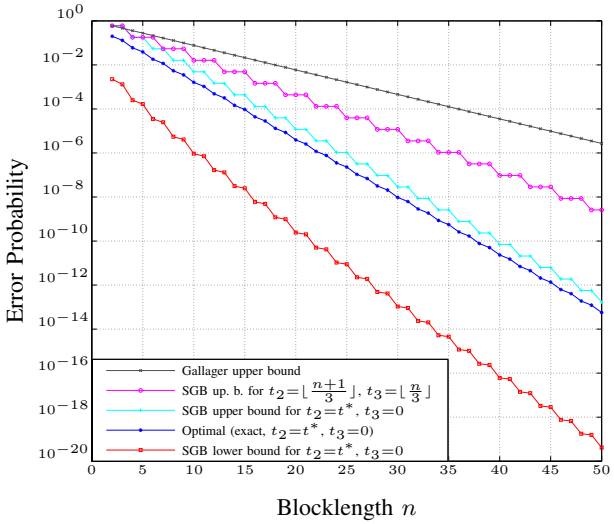

Fig. 10. Exact value of, and bounds on, the performance of an optimal code with $\mathsf{M} = 3$ codewords on the ZC with $\epsilon_1 = 0.3$ as a function of the blocklength n.

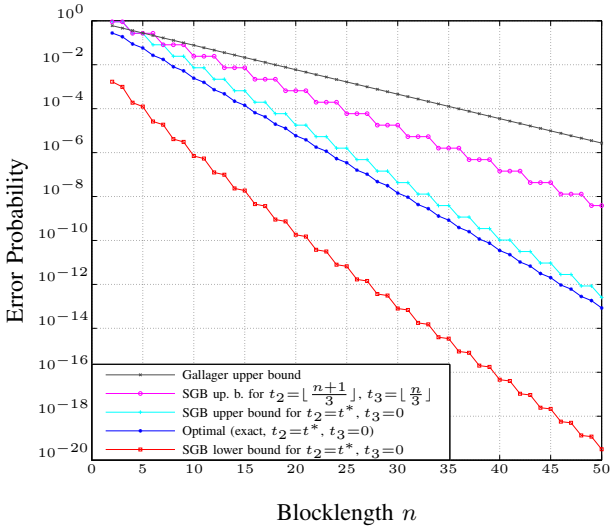

Fig. 11. Exact value of, and bounds on, the performance of an optimal code with $\mathsf{M} = 4$ codewords on the ZC with $\epsilon_1 = 0.3$ as a function of the blocklength n.

E. Conjectured Optimal Codes With Five Codewords ($\mathsf{M} = 5$)

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

We define the following five weak flip column vectors:

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

$$\left. \mathbf{c}_4^{(5)} \triangleq \begin{pmatrix} 0\\0\\1\\1\\1 \end{pmatrix}, \mathbf{c}_5^{(5)} \triangleq \begin{pmatrix} 0\\1\\0\\1\\1 \end{pmatrix} \right\}. \tag{88}$$

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

$$\mathcal{C}_{\mathrm{ZC}}^{(5,8)*} = \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_2^{(5)} & \mathbf{c}_3^{(5)} & \mathbf{c}_1^{(5)} & \mathbf{c}_2^{(5)} & \mathbf{c}_3^{(5)} & \mathbf{c}_4^{(5)} & \mathbf{c}_5^{(5)} \end{pmatrix}. \tag{89}$$

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

$$\begin{cases} \begin{pmatrix} \mathbf{c}_4^{(5)} & \mathbf{c}_5^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 0 \\ \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_2^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 2 \\ \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_3^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 4 \\ \begin{pmatrix} \mathbf{c}_3^{(5)} & \mathbf{c}_4^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 6 \\ \begin{pmatrix} \mathbf{c}_2^{(5)} & \mathbf{c}_5^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 8. \end{cases} \tag{90}$$

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

$$\mathcal{C}_{\mathrm{ZC}}^{(5,9)*} = \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_2^{(5)} & \mathbf{c}_3^{(5)} & \mathbf{c}_4^{(5)} & \mathbf{c}_5^{(5)} & \mathbf{c}_1^{(5)} & \mathbf{c}_2^{(5)} & \mathbf{c}_1^{(5)} & \mathbf{c}_3^{(5)} \end{pmatrix} \tag{91}$$

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

$$\begin{cases} \begin{pmatrix} \mathbf{c}_3^{(5)} & \mathbf{c}_4^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 1 \\ \begin{pmatrix} \mathbf{c}_2^{(5)} & \mathbf{c}_5^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 3 \\ \begin{pmatrix} \mathbf{c}_4^{(5)} & \mathbf{c}_5^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 5 \\ \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_2^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 7 \\ \begin{pmatrix} \mathbf{c}_1^{(5)} & \mathbf{c}_3^{(5)} \end{pmatrix} & \text{if } n \bmod 10 = 9. \end{cases} \tag{92}$$

Note that the recursive structure in (90) and (92) 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 (88) exactly twice. For $n < 10$, the optimal code structure goes through some transient states.

IX. Analysis of the BSC

A. Optimal Codes With Two Codewords ($\mathsf{M} = 2$)

Theorem 36: 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 23 we already know that there must exist a flip code that is optimal. Moreover, Theorem 23 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 $(1\ 0)^{\mathsf{T}}$. 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. ∎

B. Optimal Codes With Three or Four Codewords ($\mathsf{M} = 3, 4$)

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

Theorem 37: For a BSC with arbitrary cross-over probability $0 \le \epsilon < \frac{1}{2}$, the optimal code with three codewords $\mathsf{M} = 3$ or four codewords $\mathsf{M} = 4$ and with a blocklength $n = 2$ is

$$\mathcal{C}_{\text{BSC}}^{(\mathsf{M},2)\Diamond} = \left(\mathbf{c}_1^{(\mathsf{M})} \quad \mathbf{c}_2^{(\mathsf{M})}\right). \tag{93}$$

If we recursively construct a locally optimal codebook with three codewords $\mathsf{M} = 3$ or four codewords $\mathsf{M} = 4$ and with a blocklength $n \ge 3$ by using $\mathcal{C}_{\text{BSC}}^{(\mathsf{M},n-1)\Diamond}$ and appending a new column, the total probability increase is maximized by the following choice of appended columns:

$$\begin{cases} \mathbf{c}_1^{(\mathsf{M})} & \text{if } n \bmod 3 = 0 \\ \mathbf{c}_3^{(\mathsf{M})} & \text{if } n \bmod 3 = 1 \\ \mathbf{c}_2^{(\mathsf{M})} & \text{if } n \bmod 3 = 2. \end{cases} \tag{94}$$

Proof: See Appendix C-A. ∎

While Theorem 37 only guarantees local optimality for the given recursive construction, much points to that the given construction indeed is globally optimum. Indeed, we can prove this for the case $\mathsf{M} = 3$.

Theorem 38: For a BSC and for any $n \ge 2$, the weak flip code of type (t_2^*, t_3^*), $\mathcal{C}_{t_2^*,t_3^*}^{(3,n)}$, where

$$t_2^* \triangleq \left\lfloor \frac{n+1}{3} \right\rfloor, \quad t_3^* \triangleq \left\lfloor \frac{n-1}{3} \right\rfloor \tag{95}$$

is an optimal codebook with three codewords $\mathsf{M} = 3$. Note that the recursively constructed code of Theorem 37 is equivalent to the optimal code given here:

$$\mathcal{C}_{\text{BSC}}^{(3,n)\Diamond} \equiv \mathcal{C}_{t_2^*,t_3^*}^{(3,n)}. \tag{96}$$

Proof: See Appendix C-B. ∎

Using the shorthands

$$k \triangleq \left\lfloor \frac{n}{3} \right\rfloor, \quad p \triangleq \frac{\epsilon}{1-\epsilon} < 1 \tag{97}$$

the code parameters of these optimal codes can be written as

$$[t_1^*, t_2^*, t_3^*] = \begin{cases} [k+1, k, k-1] & \text{if } n \bmod 3 = 0 \\ [k+1, k, k] & \text{if } n \bmod 3 = 1 \\ [k+1, k+1, k] & \text{if } n \bmod 3 = 2 \end{cases} \tag{98}$$

and the exact average success probability can be derived recursively in the blocklength n: starting with

$$3P_{\text{c}}\big(\mathcal{C}_{t_2^*,t_3^*}^{(3,2)}\big) = (1-\epsilon)^2 \cdot (3+p) \tag{99}$$

we have[7]

$$P_{\text{c}}\big(\mathcal{C}_{t_2^*,t_3^*}^{(3,n)}\big) = P_{\text{c}}\big(\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) + \frac{1}{3}\sum_{a_3=0}^{k-1}\sum_{a_2=0}^{a_3}\binom{k}{a_2}\binom{k}{a_2}\binom{k-1}{a_3} \cdot (1-\epsilon)^n\left(p^{2k-1-a_3} - p^{2k-a_3}\right) \tag{100}$$

if $n = 3k$;

$$P_{\text{c}}\big(\mathcal{C}_{t_2^*,t_3^*}^{(3,n)}\big) = P_{\text{c}}\big(\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) + \frac{1}{3}\sum_{a_2=1}^{k}\sum_{a_1=1}^{a_2}\binom{k+1}{a_1}\binom{k}{a_2}\binom{k-1}{a_1-1} \cdot (1-\epsilon)^n\left(p^{2k-a_2} - p^{2k+1-a_2}\right) \tag{101}$$

if $n = 3k+1$; and

$$P_{\text{c}}\big(\mathcal{C}_{t_2^*,t_3^*}^{(3,n)}\big) = P_{\text{c}}\big(\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) + \frac{1}{3}\sum_{a_1=1}^{k+1}\sum_{a_3=0}^{a_1-1}\binom{k+1}{a_1}\binom{k}{a_3}\binom{k}{a_3} \cdot (1-\epsilon)^n\left(p^{2k+1-a_1} - p^{2k+2-a_1}\right) \tag{102}$$

if $n = 3k+2$.

The average success probability of $\mathcal{C}_{\text{BSC}}^{(4,n)\Diamond}$ can be expressed in a similar manner.

Note that for $\mathsf{M} = 2$, the optimal codes given in Theorem 36 can be linear or nonlinear. For $\mathsf{M} = 4$, by the definition of the weak flip code of type (t_2, t_3), the locally optimal codes $\mathcal{C}_{\text{BSC}}^{(4,n)\Diamond}$ are linear. As mentioned, there exists strong evidence that these codes are also globally optimal. Indeed, it can be shown that among all *linear* codes with four codewords, they are optimal.

We also would like to point out the regularity of the code design in Theorem 37 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 38 shows that when $\mathsf{M} = 3$, the received vector $\mathbf{y}$ farthest from the three codewords is

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

which corresponds to the choice of the fourth codeword $\mathbf{x}_4$ in $\mathcal{C}_{\text{BSC}}^{(4,n)\Diamond}$.

C. Pairwise Hamming Distance Structure

As already mentioned in Section III-D, it is quite common in conventional coding theory to use the *minimum Hamming distance* or the *weight enumerating function (WEF)* of a code as a design and quality criterion [12]. This is motivated by the equivalence of Hamming weight and Hamming distance for linear codes, and by the union bound that converts the search for the global error probability into pairwise error probabilities. Since we are interested in the globally optimal code design and the best performance achieved by an ML decoder, we can neither use the union bound, nor can we *a priori* restrict our search to linear codes. Note that for most values of M, linear codes do not even exist.[8]

We would like to come back to the example shown in Section IV and further deepen our analysis of the minimum Hamming distance of our optimal codes on the very symmetric BSC. Although, as (17) shows, the error probability

[7]For a meaning of the counters a_i, see the explanations before (234) in Appendix C.

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

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 37 with the following different weak flip code $\mathcal{C}_{\text{subopt}}^{(\mathsf{M},n)}$ with code parameters

$$[t_1,t_2,t_3]=\begin{cases}[k,k,k] & \text{if } n \bmod 3=0\\ [k+1,k+1,k-1] & \text{if } n \bmod 3=1\\ [k+2,k,k] & \text{if } n \bmod 3=2.\end{cases} \tag{104}$$

This code can be constructed from the optimal code $\mathcal{C}_{\text{BSC}}^{(\mathsf{M},n-1)*}$ by appending a suboptimal column[9] and—based on a closer inspection of the proof of Theorem 37—can be shown to be strictly suboptimal.

Recalling Lemma 13, we compute the pairwise Hamming distance vector of the optimal code for $\mathsf{M}=3$:

$$\mathbf{d}\big(\mathcal{C}_{\text{BSC}}^{(3,n)*}\big)=\begin{cases}(2k-1,2k,2k+1) & \text{if } n \bmod 3=0\\ (2k,2k,2k+1) & \text{if } n \bmod 3=1\\ (2k+1,2k+1,2k+2) & \text{if } n \bmod 3=2\end{cases} \tag{105}$$

i.e.,

$$d_{\min}\big(\mathcal{C}_{\text{BSC}}^{(3,n)*}\big)=\begin{cases}2k-1 & \text{if } n \bmod 3=0\\ 2k & \text{if } n \bmod 3=1\\ 2k+1 & \text{if } n \bmod 3=2.\end{cases} \tag{106}$$

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

$$\mathbf{d}\big(\mathcal{C}_{\text{BSC}}^{(4,n)\diamondsuit}\big)=\begin{cases}(2k-1,2k,2k+1,2k+1,2k,2k-1) & \text{if } n \bmod 3=0\\ (2k,2k,2k+1,2k+1,2k,2k) & \text{if } n \bmod 3=1\\ (2k+1,2k+1,2k+2,2k+2,2k+1,2k+1) & \text{if } n \bmod 3=2\end{cases} \tag{107}$$

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

Comparing this with the suboptimal code (104) now yields for $\mathsf{M}=3$:

$$\mathbf{d}\big(\mathcal{C}_{\text{subopt}}^{(3,n)}\big)=\begin{cases}(2k,2k,2k) & \text{if } n \bmod 3=0\\ (2k,2k,2k+2) & \text{if } n \bmod 3=1\\ (2k,2k+2,2k+2) & \text{if } n \bmod 3=2\end{cases} \tag{108}$$

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

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

[9]The choice of column depends on n.

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

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

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

We have shown the following proposition.

Proposition 39: On a BSC for $\mathsf{M}=3$ or $\mathsf{M}=4$ and for all n with $n \bmod 3=0$ or $n \bmod 3=1$, codes that maximize the minimum Hamming distance $d_{\min}\big(\mathcal{C}^{(\mathsf{M},n)}\big)$ can be strictly suboptimal. This is not true in the case of $n \bmod 3=2$.

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

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

We again provide a comparison between the performance of the optimal code to the known bounds of Section VI.

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

$$\mathsf{E}_3=\mathsf{E}_4=-\frac{2}{3}\log\sqrt{4\epsilon(1-\epsilon)}. \tag{110}$$

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

$$\mathsf{D}_{\min}^{(\text{BSC})}\Big(\mathcal{C}_{\lfloor\frac{n+1}{3}\rfloor,\lfloor\frac{n}{3}\rfloor}^{(\mathsf{M},n)}\Big)=\begin{cases}-\frac{2}{3}\log\sqrt{4\epsilon(1-\epsilon)} & \text{if } n \bmod 3=0\\ -\frac{\lfloor\frac{n}{3}\rfloor+\lfloor\frac{n+1}{3}\rfloor}{n}\log\sqrt{4\epsilon(1-\epsilon)} & \text{if } n \bmod 3=1\\ -\frac{\lfloor\frac{n}{3}\rfloor+\lfloor\frac{n+1}{3}\rfloor}{n}\log\sqrt{4\epsilon(1-\epsilon)} & \text{if } n \bmod 3=2.\end{cases} \tag{111}$$

Figs. 12 and 13 compare the exact optimal performance for $\mathsf{M}=3$ and $\mathsf{M}=4$, respectively, with some bounds: the SGB upper bound based on the weak flip code used by Shannon *et al.*,[10] the SGB lower bound based on the weak flip code (which is suboptimal, but achieves the optimal $\mathsf{D}_{\min}^{(\text{DMC})}$ and is therefore a generally valid lower bound), the Gallager upper bound, and also the PPV upper and lower bounds.

We can see that the PPV upper bound is tighter to the exact optimal performance than the SGB upper bound. Note, however, that only the SGB upper bound exhibits the correct error exponent as n is large enough. It is shown in [18] that, for n going to infinity, the random coding (PPV) upper bound tends to the Gallager exponent for $R=0$ [6], which is of course not necessarily equal to E_{M} for finite M.

Concerning the lower bounds, we see that the PPV lower bound (meta-converse) is much better for finite n than the SGB bound. However, for n large enough, its exponential growth will approach that of the sphere-packing bound [15], which does not equal to E_{M} either.

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

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

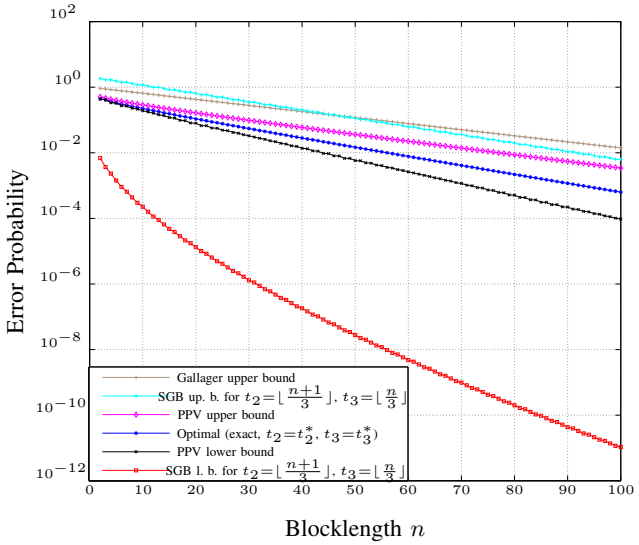

Fig. 12. Exact value of, and bounds on, the performance of an optimal code with $\mathsf{M} = 3$ codewords on the BSC with $\epsilon = 0.3$ as a function of the blocklength n.

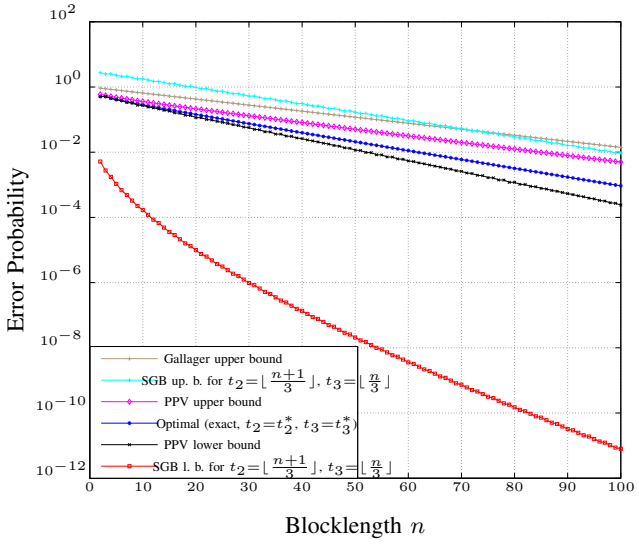

Fig. 13. Exact value of, and bounds on, the performance of an optimal code with $\mathsf{M} = 4$ codewords on the BSC with $\epsilon = 0.3$ as a function of the blocklength n.

X. Conclusions

We have studied the optimal code design of ultrasmall 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)*. For the ZC and for an arbitrary finite blocklength n, we have derived an optimal code design with four or less messages and we have conjectured an optimal code design with five messages. For the BSC and for an arbitrary finite blocklength n, we have derived an optimal code design with two or three messages and we have conjectured an optimal code design with four messages.

Note that since the optimal codes we proposed do not depend on the cross-over probability of the channel, the optimal codes remain the same even if the channel is nonergodic or nonstationary. Also note that the optimal weak flip codes are by definition coset codes: the $\mathsf{M} = 3$ nonlinear code is always a coset of the $\mathsf{M} = 4$ linear code. However, they are not fixed composition codes.

We have introduced a new way of generating these codes recursively by using a column-wise build-up of the codebook matrix. This column view of the codebook turns out to be a more powerful tool for analysis than the standard rowwise 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) [14]. 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 have shown 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 V (and in particular the subfamily *fair weak flip codes*) turns out to have many interesting properties. A first closer investigation of some of these properties and the relation of these codes to linear codes can be found in [14].

Appendix A
Derivations Concerning the BAC

A. LLR Function

Proposition 40 (Properties of $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$*):*

1) If $\epsilon_0+\epsilon_1=1$, then $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)=0$ irrespective of d, t, or n.
2) $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$ is a nonincreasing function in d for every n, t:
$$\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d) \le \mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d-1), \quad 1 \le d \le n. \tag{112}$$
3) For certain values of d, the value of $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$ is always nonnegative (or always nonpositive) for all ϵ_0 and ϵ_1:
$$\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d) \begin{cases} \ge 0 & \text{if } 0 \le d \le t \\ \lesseqgtr 0 & \text{if } t < d \le \left\lfloor \frac{n}{2} \right\rfloor \text{ (depending on } \epsilon_0,\ \epsilon_1) \\ \le 0 & \text{if } \left\lfloor \frac{n}{2} \right\rfloor < d \le n. \end{cases} \tag{113}$$
4) $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$ is a nondecreasing function in n for fixed t, d, and (ϵ_0,ϵ_1).
5) $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$ is a nondecreasing function in t for fixed n, d, and (ϵ_0,ϵ_1).
6) For $0 \le d \le n$,
$$\mathrm{LLR}_t^{(n+1)}(\epsilon_0,\epsilon_1,d+1) < \mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d). \tag{114}$$

Proof: These properties follow quite easily from the definition of $\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)$ and the relations (1)–(3). We only show a proof of the second property:

$$\begin{aligned}&\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d-1)-\mathrm{LLR}_t^{(n)}(\epsilon_0,\epsilon_1,d)\\&\quad=\log\left(\frac{1-\epsilon_1}{\epsilon_0}\right)+\log\left(\frac{1-\epsilon_0}{\epsilon_1}\right)\geq 0.\end{aligned}\tag{115}$$

■

B. Alternative Proof of Theorem 23

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 23 is proven once we have shown that

$$P_{\mathrm{e}}\big(\mathcal{C}_t^{(2,n-1)}\big)\geq\max\Big\{P_{\mathrm{e}}\big(\mathcal{C}_t^{(2,n)}\big),P_{\mathrm{e}}\big(\mathcal{C}_{t+1}^{(2,n)}\big)\Big\}.\tag{116}$$

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

$$\mathcal{C}_t^{(2,n-1)}=\begin{pmatrix}\mathbf{x}_1^{(n-1)}\\ \mathbf{x}_2^{(n-1)}\end{pmatrix}\tag{117}$$

we can derive two nontrivial length-n codes:

$$\mathcal{C}_t^{(2,n)}=\begin{pmatrix}\big[\mathbf{x}_1^{(n-1)}\ 0\big]\\ \big[\mathbf{x}_2^{(n-1)}\ 1\big]\end{pmatrix},\quad \mathcal{C}_{t+1}^{(2,n)}=\begin{pmatrix}\big[\mathbf{x}_1^{(n-1)}\ 1\big]\\ \big[\mathbf{x}_2^{(n-1)}\ 0\big]\end{pmatrix}.\tag{118}$$

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 Theorem 24, the optimal ML decision rule for $\mathcal{C}_t^{(2,n)}$ can be expressed as

$$g(\mathbf{y})=\begin{cases}1 & \text{if } 0\leq d^{(n)}\leq\ell^{(n)}\\ 2 & \text{if } \ell^{(n)}+1\leq d^{(n)}\leq n.\end{cases}\tag{119}$$

The threshold satisfies $0\leq\ell^{(n)}\leq\big\lfloor\frac{n-1}{2}\big\rfloor$. Note that when $\ell^{(n)}=\big\lfloor\frac{n-1}{2}\big\rfloor$, 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 $\mathcal{C}_t^{(2,n-1)}$ be $\ell^{(n-1)}$. We will now argue that the threshold for $\mathcal{C}_t^{(2,n)}$ and $\mathcal{C}_{t+1}^{(2,n)}$ (compare with (118)) must satisfy

$$\ell^{(n)}=\ell^{(n-1)}\quad\text{or}\quad\ell^{(n)}=\ell^{(n-1)}+1.\tag{120}$$

Consider first the code $\mathcal{C}_t^{(2,n)}$ 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 $\mathcal{C}_t^{(n-1)}$) 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 $\mathcal{C}_t^{(2,n-1)}$ was $\mathbf{x}_1^{(n-1)}$.

Second, again considering code $\mathcal{C}_t^{(2,n)}$, 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 $\mathcal{C}_t^{(2,n-1)}$) 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 $\mathcal{C}_t^{(2,n-1)}$ was $\mathbf{x}_1^{(n-1)}$.

The same arguments also hold for the other code $\mathcal{C}_{t+1}^{(2,n)}$. 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 $P_{\mathrm{e}}\big(\mathcal{C}_t^{(2,n-1)}\big)\geq P_{\mathrm{e}}\big(\mathcal{C}_t^{(2,n)}\big)$.

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

$$P_{\mathrm{e}}=\frac{1}{2}\sum_{\substack{\mathbf{y}\\ g(\mathbf{y})=2}}P_{Y|X}^n(\mathbf{y}|\mathbf{x}_1)+\frac{1}{2}\sum_{\substack{\mathbf{y}\\ g(\mathbf{y})=1}}P_{Y|X}^n(\mathbf{y}|\mathbf{x}_2).\tag{121}$$

For $\mathcal{C}_t^{(2,n-1)}$, (121) can be written as follows:

$$\begin{aligned}&2P_{\mathrm{e}}\big(\mathcal{C}_t^{(2,n-1)}\big)\\&\quad=\sum_{\substack{\mathbf{y}^{(n-1)}\\ \ell^{(n-1)}+1\leq d^{(n-1)}\leq n-1}}P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\\&\qquad+\sum_{\substack{\mathbf{y}^{(n-1)}\\ 0\leq d^{(n-1)}\leq\ell^{(n-1)}}}P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)\qquad(122)\\&\quad=\sum_{\substack{\mathbf{y}^{(n-1)}\\ \ell^{(n-1)}+1\leq d^{(n-1)}\leq n-1}}P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)P_{Y|X}(1|0)\\&\qquad+\sum_{\substack{\mathbf{y}^{(n-1)}\\ \ell^{(n-1)}+1\leq d^{(n-1)}\leq n-1}}P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)P_{Y|X}(0|0)\\&\qquad+\sum_{\substack{\mathbf{y}^{(n-1)}\\ 0\leq d^{(n-1)}\leq\ell^{(n-1)}}}P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)P_{Y|X}(1|1)\\&\qquad+\sum_{\substack{\mathbf{y}^{(n-1)}\\ 0\leq d^{(n-1)}\leq\ell^{(n-1)}}}P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)P_{Y|X}(0|1)\qquad(123)\\&\quad=\sum_{\substack{\mathbf{y}^{(n-1)}\\ \ell^{(n-1)}+2\leq d^{(n)}\leq n}}P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_1^{(n)}\Big)\\&\qquad+\sum_{\substack{\mathbf{y}^{(n-1)}\\ \ell^{(n-1)}+1\leq d^{(n)}\leq n-1}}P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big)\\&\qquad+\sum_{\substack{\mathbf{y}^{(n-1)}\\ 1\leq d^{(n)}\leq\ell^{(n-1)}+1}}P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big)\end{aligned}$$

$$+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0\le d^{(n)}\le \ell^{(n-1)}}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_2^{(n)}\Big). \tag{124}$$

Here, in (123) 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 (124) we combine the terms together using the definition of $\mathcal{C}_t^{(2,n)}$ according to (26) (and (118)).

We can now distinguish the two cases (120):

(i) If the decision rule for $\mathcal{C}_t^{(2,n)}$ is unchanged, i.e., $\ell^{(n)}=\ell^{(n-1)}$, we only need to take care of the third summation in (124) that contains some terms that will now be decoded differently. We split this sum up into two parts:

$$\begin{aligned}&\sum_{\substack{\mathbf{y}^{(n-1)} \\ 1\le d^{(n)}\le \ell^{(n-1)}+1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big)\\ &\quad= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n)}=\ell^{(n-1)}+1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big)\\ &\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 1\le d^{(n)}\le \ell^{(n-1)}}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big).\end{aligned} \tag{125}$$

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

$$\frac{P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_1^{(n)}\Big)}{P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big)}\le 1. \tag{126}$$

Hence, we have

$$\begin{aligned}&2P_{\mathrm{e}}\big(\mathcal{C}_t^{(2,n-1)}\big)\\ &\quad\ge \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+2\le d^{(n)}\le n}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_1^{(n)}\Big)\\ &\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+1\le d^{(n)}\le n-1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big)\\ &\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n)}=\ell^{(n-1)}+1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_1^{(n)}\Big)\\ &\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 1\le d^{(n)}\le \ell^{(n-1)}}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\mathbf{x}_2^{(n)}\Big)\\ &\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0\le d^{(n)}\le \ell^{(n-1)}}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_2^{(n)}\Big)\\ &\quad= \sum_{\substack{\mathbf{y}^{(n)} \\ \ell^{(n-1)}+1\le d^{(n)}\le n}} P_{Y|X}^n\big(\mathbf{y}^{(n)}\big|\mathbf{x}_1^{(n)}\big)\\ &\qquad+ \sum_{\substack{\mathbf{y}^{(n)} \\ 0\le d^{(n)}\le \ell^{(n-1)}}} P_{Y|X}^n\big(\mathbf{y}^{(n)}\big|\mathbf{x}_2^{(n)}\big)\end{aligned} \tag{127, 128}$$

$$= 2P_{\mathrm{e}}\big(\mathcal{C}_t^{(2,n)}\big). \tag{129}$$

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

$$\begin{aligned}&\sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+1\le d^{(n)}\le n-1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big)\\ &\quad= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n)}=\ell^{(n-1)}+1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big)\\ &\qquad+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell^{(n-1)}+2\le d^{(n)}\le n-1}} P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big).\end{aligned} \tag{130}$$

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

$$\frac{P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_1^{(n)}\Big)}{P_{Y|X}^n\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\mathbf{x}_2^{(n)}\Big)}\ge 1. \tag{131}$$

The rest of the argument now is analogous to Case (i). This proves that $P_{\mathrm{e}}\big(\mathcal{C}_t^{(2,n-1)}\big)\ge P_{\mathrm{e}}\big(\mathcal{C}_t^{(2,n)}\big)$. The remaining proof of $P_{\mathrm{e}}\big(\mathcal{C}_t^{(2,n-1)}\big)\ge P_{\mathrm{e}}\big(\mathcal{C}_{t+1}^{(2,n)}\big)$ is similar and omitted.

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

The reason why we show this long derivation in addition to the compact proof given in Section VII-A is the expression (124) that explicitly states the error probability as a function of the ML decoder threshold. In the sequel of (124), we had to make a case distinction depending on what the correct ML decoder looks like. In the proof of Theorem 25 in the following section, we will assume that the decoder is fixed, which will allow us to make even better use of (124).

C. Proof of Theorem 25

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

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

Following the same structure as in (124), we write the error probability of $\mathcal{C}_t^{(2,n)}$ for the given decoding rule ℓ as follows:

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

$$+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \le d^{(n-1)} \le \ell}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\big[\mathbf{x}_2^{(n-1)}\ 1\big]\Big)$$
$$+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \le d^{(n-1)} \le \ell-1}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\big[\mathbf{x}_2^{(n-1)}\ 1\big]\Big) \tag{132}$$
$$= \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell+1 \le d^{(n-1)} \le n-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)(1-\epsilon_0+\epsilon_0)$$
$$+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\epsilon_0$$
$$+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \le d^{(n-1)} \le \ell-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)(\epsilon_1+1-\epsilon_1)$$
$$+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)\epsilon_1. \tag{133}$$

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

$$2P_{\mathrm{e}}^{(\ell)}\big(\mathcal{C}_{t+1}^{(2,n)}\big)$$
$$= \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell+1 \le d^{(n-1)} \le n-1}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\big[\mathbf{x}_1^{(n-1)}\ 1\big]\Big)$$
$$+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell \le d^{(n-1)} \le n-1}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\big[\mathbf{x}_1^{(n-1)}\ 1\big]\Big)$$
$$+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \le d^{(n-1)} \le \ell}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 1\big]\Big|\big[\mathbf{x}_2^{(n-1)}\ 0\big]\Big)$$
$$+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \le d^{(n-1)} \le \ell-1}} P_{Y|X}^{n}\Big(\big[\mathbf{y}^{(n-1)}\ 0\big]\Big|\big[\mathbf{x}_2^{(n-1)}\ 0\big]\Big) \tag{134}$$
$$= \sum_{\substack{\mathbf{y}^{(n-1)} \\ \ell+1 \le d^{(n-1)} \le n-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)(1-\epsilon_1+\epsilon_1)$$
$$+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\epsilon_1$$
$$+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ 0 \le d^{(n-1)} \le \ell-1}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)(\epsilon_0+1-\epsilon_0)$$
$$+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)\epsilon_0. \tag{135}$$

Subtracting (135) from (133) then yields

$$2P_{\mathrm{e}}^{(\ell)}\big(\mathcal{C}_{t}^{(2,n)}\big) - 2P_{\mathrm{e}}^{(\ell)}\big(\mathcal{C}_{t+1}^{(2,n)}\big)$$
$$= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\epsilon_0$$
$$+ \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)\epsilon_1$$
$$- \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\epsilon_1$$
$$- \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)\epsilon_0 \tag{136}$$
$$= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} \Big(P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big) - P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)\Big)(\epsilon_1-\epsilon_0) \tag{137}$$
$$= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big) \cdot \left(1 - \frac{P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_1^{(n-1)}\big)}{P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big)}\right)(\epsilon_1-\epsilon_0) \tag{138}$$
$$= \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big) \cdot \Big(1 - e^{\mathrm{LLR}_t^{(n-1)}(\epsilon_0,\epsilon_1,\ell)}\Big)(\epsilon_1-\epsilon_0) \tag{139}$$
$$= \Big(1 - e^{\mathrm{LLR}_t^{(n-1)}(\epsilon_0,\epsilon_1,\ell)}\Big) \cdot (\epsilon_1-\epsilon_0) \sum_{\substack{\mathbf{y}^{(n-1)} \\ d^{(n-1)} = \ell}} P_{Y|X}^{n-1}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}_2^{(n-1)}\big) \tag{140}$$

where in (139) we make use of our assumption that $\mathcal{C}_t^{(2,n-1)}$ 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, $2P_{\mathrm{e}}^{(\ell)}\big(\mathcal{C}_t^{(2,n)}\big) - 2P_{\mathrm{e}}^{(\ell)}\big(\mathcal{C}_{t+1}^{(2,n)}\big)$ can only be zero if

$$\mathrm{LLR}_t^{(n-1)}(\epsilon_0,\epsilon_1,\ell) = 0. \tag{141}$$

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 (141) is satisfied. The same is true if we fix ϵ_1 and search for an ϵ_0.

APPENDIX B
DERIVATIONS CONCERNING THE ZC

A. Proof of Theorem 30

Consider a general codebook matrix $\mathcal{C}^{(\mathsf{M},n)}$ with codewords $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{\mathsf{M}}$. Considering Remark 6, we can assume without loss of generality that

$$w \triangleq w_{\mathrm{H}}(\mathbf{x}_1) \le w_{\mathrm{H}}(\mathbf{x}_2) \le \cdots \le w_{\mathrm{H}}(\mathbf{x}_{\mathsf{M}}) \tag{142}$$

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

$$\mathbf{x}_1 = (0\ \cdots\ 0\ \underbrace{1\ \cdots\ 1}_{w \text{ pos.}}). \tag{143}$$

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

We note that for any $\mathbf{y} = (0\ \cdots\ 0\ y_{n-w+1}\ \cdots\ y_n)$ with $0 \le w_{\mathrm{H}}(\mathbf{y}) \le w$, and for every codeword $\mathbf{x}_m$, $1 \le m \le \mathsf{M}$,

the conditional channel law can be expressed as

$$P_{X|Y}(\mathbf{y}|\mathbf{x}_m) = \mathsf{I}\{d_{0,1}(\mathbf{x}_m,\mathbf{y})=0\}\,\epsilon_1^{w_{\mathrm{H}}(\mathbf{x}_m)-d_{1,1}(\mathbf{x}_1,\mathbf{y})} \cdot (1-\epsilon_1)^{d_{1,1}(\mathbf{x}_1,\mathbf{y})} \tag{144}$$

where $\mathsf{I}\{\cdot\}$ denotes again the indicator function, and where we make explicit use of the shape of $\mathbf{x}_1$ (143), the structure of the considered received vector $\mathbf{y}$ and the assumption (142). Note that the value of (144)—if positive—only depends on $\mathbf{x}_m$ via its Hamming weight. Hence,

$$\begin{aligned}&\max\left\{P_{Y|X}^n(\mathbf{y}|\mathbf{x}_1),\ldots,P_{Y|X}^n(\mathbf{y}|\mathbf{x}_m),\ldots,P_{Y|X}^n(\mathbf{y}|\mathbf{x}_{\mathsf{M}})\right\}\\ &= \max\Big\{\epsilon_1^{w-d_{1,1}(\mathbf{x}_1,\mathbf{y})}(1-\epsilon_1)^{d_{1,1}(\mathbf{x}_1,\mathbf{y})},\\ &\quad \mathsf{I}\{d_{0,1}(\mathbf{x}_2,\mathbf{y})=0\}\,\epsilon_1^{w_{\mathrm{H}}(\mathbf{x}_2)-d_{1,1}(\mathbf{x}_1,\mathbf{y})}(1-\epsilon_1)^{d_{1,1}(\mathbf{x}_1,\mathbf{y})},\\ &\quad \ldots,\\ &\quad \mathsf{I}\{d_{0,1}(\mathbf{x}_{\mathsf{M}},\mathbf{y})=0\}\,\epsilon_1^{w_{\mathrm{H}}(\mathbf{x}_{\mathsf{M}})-d_{1,1}(\mathbf{x}_1,\mathbf{y})}(1-\epsilon_1)^{d_{1,1}(\mathbf{x}_1,\mathbf{y})}\Big\} \qquad (145)\\ &= \epsilon_1^{w-d_{1,1}(\mathbf{x}_1,\mathbf{y})}(1-\epsilon_1)^{d_{1,1}(\mathbf{x}_1,\mathbf{y})} \qquad (146)\end{aligned}$$

where (146) follows from (142). Since when transmitting $\mathbf{x}_1$, the received sequence cannot have any ones in the first $n-w$ positions, this now shows that the optimal decoding region for the first codeword is

$$\mathcal{D}_1^{(\mathsf{M},n)} = \Big\{\mathbf{y}\colon \mathbf{y} = \big(\underbrace{0\ \cdots\ 0}_{n-w\text{ pos.}}\ \underbrace{y_{n-w+1}\ \cdots\ y_n}_{w\text{ pos.}}\big)\Big\} \tag{147}$$

which yields the conditional success probability

$$\psi_1 = \sum_{d=0}^{w}\binom{w}{d}\epsilon_1^d\cdot(1-\epsilon_1)^{w-d} = 1. \tag{148}$$

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

Note that this increase is strictly larger than zero if there exists some other codeword that has one or more ones in the last w positions.

B. Proof of Theorem 31

The proof of Theorem 31 is based on an exact expression of the average success probability as a function of the numbers of candidate columns t_i. The problem is then transformed into an optimization problem.

We first consider the easier case of $\mathsf{M}=3$. By Theorem 30 and because the all-zero column can be ignored (based on the argument used in the proof of Theorem 23), we can restrict our search to the candidate columns given in (28). Hence, for any blocklength n, with $t_1+t_2+t_3=n$, consider an arbitrary codebook $\mathcal{C}_{t_2,t_3}^{(3,n)}$ and, without loss of generality, assume that

$$w_{\mathrm{H}}(\mathbf{x}_1)\le w_{\mathrm{H}}(\mathbf{x}_2)\le w_{\mathrm{H}}(\mathbf{x}_3). \tag{149}$$

Moreover, note that

$$w_{\mathrm{H}}(\mathbf{x}_1)=0,\quad w_{\mathrm{H}}(\mathbf{x}_2)=t_2+t_3,\quad w_{\mathrm{H}}(\mathbf{x}_3)=t_1+t_3 \tag{150}$$

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

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

Defining $t\triangleq t_2+t_3$ and using a derivation similar to (145)–(147), we further realize that

$$\mathcal{D}_{t_2,t_3;2}^{(3,n)} = \Big\{\mathbf{y}\colon \mathbf{y} = \big(\underbrace{0\ \cdots\ 0}_{n-t\text{ pos.}}\ \underbrace{y_{n-t+1}\ \cdots\ y_n}_{t\text{ pos.}}\big) \text{ with } 1\le w_{\mathrm{H}}(\mathbf{y})\le t\Big\} \tag{151}$$

and

$$\psi_2 = 1-\epsilon_1^t. \tag{152}$$

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

$$\begin{aligned}\mathcal{D}_{t_2,t_3;3}^{(3,n)} &= \{0,1\}^n\setminus\big(\mathcal{D}_{t_2,t_3;2}^{(3,n)}\cup\{\mathbf{0}\}\big) \qquad (153)\\ &= \Big\{\mathbf{y}\colon [\mathbf{y}^{(t_1)}\ \mathbf{0}^{(t_2)}\ \mathbf{y}^{(t_3)}] \text{ with } 1\le w_{\mathrm{H}}(\mathbf{y}^{(t_1)})\le t_1,\\ &\qquad 0\le w_{\mathrm{H}}(\mathbf{y}^{(t_3)})\le t_3\Big\} \qquad (154)\end{aligned}$$

with

$$\begin{aligned}\psi_3 &= \left(\sum_{d=0}^{t_1-1}\epsilon_1^d(1-\epsilon_1)^{t_1-d}\right)\cdot\left(\sum_{d=0}^{t_3}\epsilon_1^d(1-\epsilon_1)^{t_3-d}\right) \qquad (155)\\ &= \big(1-\epsilon_1^{t_1}\big)\cdot 1. \qquad (156)\end{aligned}$$

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

$$3P_{\mathrm{c}}\big(\mathcal{C}_{t_2,t_3}^{(3,n)}\big) = 1+(1-\epsilon_1^t)+(1-\epsilon_1^{n-t}). \tag{157}$$

The proof for the case $\mathsf{M}=3$ is now completed by showing that the average success probability (157) is maximized by the choice $t^*=\lfloor n/2\rfloor$. Note that the exact choice of t_2 and t_3 is irrelevant as long as $t_2+t_3=t^*$.

In the case of $\mathsf{M}=4$, we cannot only rely on the candidate columns in (29), but unfortunately need to consider totally seven candidate columns:[11]

$$\left\{\mathbf{c}_1^{(4)}\triangleq\begin{pmatrix}0\\0\\0\\1\end{pmatrix},\ \mathbf{c}_2^{(4)}\triangleq\begin{pmatrix}0\\0\\1\\0\end{pmatrix},\ \mathbf{c}_3^{(4)}\triangleq\begin{pmatrix}0\\0\\1\\1\end{pmatrix},\ \mathbf{c}_4^{(4)}\triangleq\begin{pmatrix}0\\1\\0\\0\end{pmatrix},\ \mathbf{c}_5^{(4)}\triangleq\begin{pmatrix}0\\1\\0\\1\end{pmatrix},\ \mathbf{c}_6^{(4)}\triangleq\begin{pmatrix}0\\1\\1\\0\end{pmatrix},\ \mathbf{c}_7^{(4)}\triangleq\begin{pmatrix}0\\1\\1\\1\end{pmatrix}\right\}. \tag{158}$$

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

$$w_{\mathrm{H}}(\mathbf{x}_1)\le w_{\mathrm{H}}(\mathbf{x}_2)\le w_{\mathrm{H}}(\mathbf{x}_3)\le w_{\mathrm{H}}(\mathbf{x}_4). \tag{159}$$

[11] By the argument shown in the proof of Theorem 23 and by Theorem 30, the other nine columns can be excluded.

Also note that

$$w_{\mathrm{H}}(\mathbf{x}_1) = 0 \tag{160}$$
$$w_{\mathrm{H}}(\mathbf{x}_2) = t_4 + t_5 + t_6 + t_7 \tag{161}$$
$$w_{\mathrm{H}}(\mathbf{x}_3) = t_2 + t_3 + t_6 + t_7 \tag{162}$$
$$w_{\mathrm{H}}(\mathbf{x}_3) = t_1 + t_3 + t_5 + t_7 \tag{163}$$

and, as a result, $t_4 + t_5 \leq t_2 + t_3$ and $t_2 + t_6 \leq t_1 + t_5$. Again, we investigate the decoding regions with the corresponding success probabilities.

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

$$\psi_1 = 1, \quad \psi_2 = 1 - \epsilon_1^{t_4+t_5+t_6+t_7}. \tag{164}$$

Then, we have

$$\mathcal{D}_3^{(4,n)} = \Big\{\mathbf{y}^{(n)} \colon \mathbf{y}^{(n)} = \big[\mathbf{0}^{(t_1)}\ \mathbf{y}^{(t_2+t_3)}\ \mathbf{0}^{(t_4+t_5)}\ \mathbf{y}^{(t_6+t_7)}\big] \text{ with } 1 \leq w_{\mathrm{H}}\big(\mathbf{y}^{(t_2+t_3)}\big) \leq t_2 + t_3, \text{ and } 0 \leq w_{\mathrm{H}}(\mathbf{y}^{(t_6+t_7)}) \leq t_6 + t_7\Big\} \tag{165}$$

with

$$\psi_3 = 1 - \epsilon_1^{t_2+t_3}. \tag{166}$$

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

$$\mathcal{D}_4^{(4,n)} = \mathcal{P} \setminus \Big(\mathcal{D}_2^{(4,n)} \cup \mathcal{D}_3^{(4,n)}\Big) \tag{167}$$

where

$$\mathcal{P} \triangleq \Big\{\mathbf{y}^{(n)} \colon \mathbf{y}^{(n)} = \big[\mathbf{y}^{(t_1)}\ \mathbf{0}^{(t_2)}\ \mathbf{y}^{(t_3)}\ \mathbf{0}^{(t_4)}\ \mathbf{y}^{(t_5)}\ \mathbf{0}^{(t_6)}\ \mathbf{y}^{t_7)}\big] \text{ with } 1 \leq w_{\mathrm{H}}(\mathbf{y}^{(n)}) \leq t_1 + t_3 + t_5 + t_7\Big\}. \tag{168}$$

Hence

$$\begin{aligned}\psi_4 &= \sum_{\mathbf{y}\in\mathcal{P}} P_{Y|X}(\mathbf{y}|\mathbf{x}_4) - \sum_{\mathbf{y}\in\mathcal{P}\cap\mathcal{D}_2^{(4,n)}} P_{Y|X}(\mathbf{y}|\mathbf{x}_4) \\ &\quad - \sum_{\mathbf{y}\in(\mathcal{P}\setminus\mathcal{D}_2^{(4,n)})\cap\mathcal{D}_3^{(4,n)}} P_{Y|X}(\mathbf{y}|\mathbf{x}_4) \qquad (169)\\ &= \left(\sum_{d=0}^{t_1+t_3+t_5+t_7-1} \epsilon_1^d(1-\epsilon_1)^{t_1+t_3+t_5+t_7-d}\right) \\ &\quad - \epsilon_1^{t_1+t_3}\left(\sum_{d=0}^{t_5+t_7-1} \epsilon_1^d(1-\epsilon_1)^{t_5+t_7-d}\right) \\ &\quad - \epsilon_1^{t_1+t_5}\left(\sum_{d=1}^{t_3-1} \epsilon_1^d(1-\epsilon_1)^{t_3-d}\right)\left(\sum_{d=1}^{t_7} \epsilon_1^d(1-\epsilon_1)^{t_7-d}\right) \qquad (170)\\ &= \big(1-\epsilon_1^{t_1+t_3+t_5+t_7}\big) - \epsilon_1^{t_1+t_3}\big(1-\epsilon_1^{t_5+t_7}\big) \\ &\quad - \epsilon_1^{t_1+t_5}\big(1-\epsilon_1^{t_3}\big) \qquad (171)\\ &= 1 - \epsilon_1^{t_1+t_3} - \epsilon_1^{t_1+t_5}(1-\epsilon_1^{t_3}) \qquad (172)\end{aligned}$$

where

$$\begin{aligned}&\mathcal{P} \cap \mathcal{D}_2^{(4,n)} \\ &= \Big\{\mathbf{y}^{(n)} \colon \mathbf{y}^{(n)} = \big[\mathbf{0}^{(t_1)}\ \mathbf{0}^{(t_2)}\ \mathbf{0}^{(t_3)}\ \mathbf{0}^{(t_4)}\ \mathbf{y}^{(t_5)}\ \mathbf{0}^{(t_6)}\ \mathbf{y}^{t_7)}\big] \\ &\qquad \text{with } 1 \leq w_{\mathrm{H}}(\mathbf{y}) \leq t_5 + t_7\Big\} \qquad (173)\end{aligned}$$

and

$$\begin{aligned}&\big(\mathcal{P} \setminus \mathcal{D}_2^{(4,n)}\big) \cap \mathcal{D}_3^{(4,n)} \\ &= \Big\{\mathbf{y}^{(n)} \colon \mathbf{y}^{(n)} = \big[\mathbf{0}^{(t_1)}\ \mathbf{0}^{(t_2)}\ \mathbf{y}^{(t_3)}\ \mathbf{0}^{(t_4)}\ \mathbf{0}^{(t_5)}\ \mathbf{0}^{(t_6)}\ \mathbf{y}^{(t_7)}\big] \\ &\qquad \text{with } 1 \leq w_{\mathrm{H}}(\mathbf{y}^{(t_3)}) \leq t_3, 0 \leq w_{\mathrm{H}}(\mathbf{y}^{(t_7)}) \leq t_7\Big\}. \qquad (174)\end{aligned}$$

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

$$\begin{aligned}4P_{\mathrm{c}}\big(\mathcal{C}^{(4,n)}\big) &= 1 + \big(1 - \epsilon_1^{n-(t_1+t_2+t_3)}\big) + \big(1 - \epsilon_1^{t_2+t_3}\big) \\ &\quad + \big(1 - \epsilon_1^{t_1+t_3} - \epsilon_1^{t_1+t_5}(1-\epsilon_1^{t_3})\big) \qquad (175)\end{aligned}$$

and is maximized for

$$\mathbf{t}^* = \left[0, 0, \left\lfloor \frac{n}{2} \right\rfloor, 0, \left\lceil \frac{n}{2} \right\rceil, 0, 0\right]. \tag{176}$$

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

C. Proof of Lemma 34

We apply (157) and (175) to the weak flip code of type $(t, 0)$.

Corollary 41: On a ZC, for $\mathsf{M} = 3$ or $\mathsf{M} = 4$, and for any $n \geq 2$, the optimal decoding regions $\mathcal{D}_{t,0;m}^{(\mathsf{M},n)}$ for the weak flip code of type $(t, 0)$, $\mathcal{C}_{t,0}^{(\mathsf{M},n)}$, for $1 \leq t \leq \left\lfloor \frac{n}{2} \right\rfloor$, are

$$\mathcal{D}_{t,0;1}^{(\mathsf{M},n)} = \{\mathbf{0}\} \tag{177}$$
$$\mathcal{D}_{t,0;2}^{(\mathsf{M},n)} = \Big\{\mathbf{y} \colon \mathbf{y} = \big(\underbrace{0\ \cdots\ 0}_{\substack{n-t\\ \text{pos.}}}\ \underbrace{y_{n-t+1}\ \cdots\ y_n}_{t\text{ pos.}}\big) \text{ with } 1 \leq w_{\mathrm{H}}(\mathbf{y}) \leq t\Big\} \tag{178}$$
$$\mathcal{D}_{t,0;3}^{(\mathsf{M},n)} = \Big\{\mathbf{y} \colon \mathbf{y} = \big(\underbrace{y_1\ \cdots\ y_{n-t}}_{n-t\text{ pos.}}\ \underbrace{0\ \cdots\ 0}_{t\text{ pos.}}\big) \text{ with } 1 \leq w_{\mathrm{H}}(\mathbf{y}) \leq n - t\Big\} \tag{179}$$
$$\mathcal{D}_{t,0;4}^{(4,n)} = \{0,1\}^n \setminus \bigcup_{m=1}^{3} \mathcal{D}_{t,0;m}^{(4,n)}. \tag{180}$$

The corresponding average success probabilities are

$$3P_{\mathrm{c}}\big(\mathcal{C}_{t,0}^{(3,n)}\big) = 1 + \big(1 - \epsilon_1^t\big) + \big(1 - \epsilon_1^{n-t}\big) \tag{181}$$
$$\begin{aligned}4P_{\mathrm{c}}\big(\mathcal{C}_{t,0}^{(4,n)}\big) &= 1 + \big(1 - \epsilon_1^t\big) + \big(1 - \epsilon_1^{n-t}\big) \\ &\quad + \big(1 - \epsilon_1^{n-t}\big) - \epsilon_1^t\big(1 - \epsilon_1^{n-t}\big). \qquad (182)\end{aligned}$$

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

We start with $\mathsf{M} = 4$ and recall that we can restrict our search to the seven columns given in (158). To prove Lemma 34 we append an additional bit to all four codewords of $\mathcal{C}_{t,0}^{(4,n)}$ as follows:

$$\begin{pmatrix}[\mathbf{0}\ \ x_{1,n+1}]\\ [\mathbf{x}\ \ x_{2,n+1}]\\ [\bar{\mathbf{x}}\ \ x_{3,n+1}]\\ [\mathbf{1}\ \ x_{4,n+1}]\end{pmatrix} \tag{183}$$

where $x_{m,n+1} \in \{0,1\}$ and where $\mathbf{x}$ and $\bar{\mathbf{x}}$ are given in (26) with $t \in \{1,2,\ldots,\lfloor \frac{n}{2} \rfloor\}$. We denote[12] this new code by $\mathcal{C}^{(4,n+1)}$. We now need to establish the decoding regions for the new code $\mathcal{C}^{(4,n+1)}$. If we simply extend the decoding regions given in (177)–(180) 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

$$\begin{aligned}
&\psi_m\big(\mathcal{C}^{(4,n+1)}\big) \\
&\quad = \psi_m\big(\mathcal{C}_{t,0}^{(4,n)}\big) \cdot P_{Y|X}(0|x_{m,n+1}) \\
&\qquad + \psi_m\big(\mathcal{C}_{t,0}^{(4,n)}\big) \cdot P_{Y|X}(1|x_{m,n+1}) && (184)\\
&\quad = \psi_m\big(\mathcal{C}_{t,0}^{(4,n)}\big) \cdot \big(P_{Y|X}(0|x_{m,n+1}) + P_{Y|X}(1|x_{m,n+1})\big) && (185)\\
&\quad = \psi_m\big(\mathcal{C}_{t,0}^{(4,n)}\big). && (186)
\end{aligned}$$

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

First 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.

Second, 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:

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

$$\begin{aligned}
&P_{Y|X}^{n+1}\big(\mathbf{0}^{(n+1)}\big|\mathbf{x}_1^{(n+1)} = \big[\mathbf{0}^{(n)}\ 1\big]\big) \\
&\quad = \epsilon_1 \geq P_{Y|X}^{n+1}\big(\mathbf{0}^{(n+1)}\big|\mathbf{x}_m^{(n+1)}\big), \quad \forall\, m = 2,3,4 && (187)
\end{aligned}$$

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

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

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

[12]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-B, i.e., we use superscripts (n) to clarify length and affiliation.

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

$$\begin{aligned}
\Delta\psi_2 &\triangleq \psi_2\big(\mathcal{C}^{(4,n+1)}\big) - \psi_2\big(\mathcal{C}_{t,0}^{(4,n)}\big) && (188)\\
&= \sum_{\mathbf{y}^{(n)} \in \mathcal{D}_{t,0;1}^{(4,n)}} P_{Y|X}^{n+1}\Big(\big[\mathbf{y}^{(n)}\ 1\big]\Big|\big[\mathbf{0}^{(n-t)}\ \mathbf{1}^t\ 1\big]\Big) \\
&\qquad \cdot (x_{2,n+1} - x_{1,n+1})^+ && (189)\\
&= \epsilon_1^t(1-\epsilon_1)(x_{2,n+1} - x_{1,n+1})^+ && (190)
\end{aligned}$$

where

$$(x)^+ \triangleq x \cdot \mathsf{I}\{x \geq 0\} = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0. \end{cases} \qquad (191)$$

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

$$\begin{aligned}
\Delta\psi_3 &= \epsilon_1^{n-t}(1-\epsilon_1)(x_{3,n+1} - x_{1,n+1})^+ && (192)\\
\Delta\psi_4 &= \epsilon_1^{n}(1-\epsilon_1)(x_{4,n+1} - x_{1,n+1})^+. && (193)
\end{aligned}$$

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 (190), (192), and (193) as follows:

$$\begin{aligned}
\Delta\psi_2 &= \epsilon_1^t(1-\epsilon_1)(x_{2,n+1} - x_{1,n+1})^+ && (194)\\
\Delta\psi_3 &= \epsilon_1^{n-t}(1-\epsilon_1)(x_{3,n+1} - x_{2,n+1} - x_{1,n+1})^+ && (195)\\
\Delta\psi_4 &= \epsilon_1^{n}(1-\epsilon_1) \\
&\qquad \cdot (x_{4,n+1} - x_{3,n+1} - x_{2,n+1} - x_{1,n+1})^+. && (196)
\end{aligned}$$

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

$$\begin{aligned}
\Delta\psi_4 &= \sum_{\mathbf{y}^{(n)} \in \mathcal{D}_{t,0;2}^{(4,n)}} P_{Y|X}^{n+1}\big(\big[\mathbf{y}^{(n)}\ 1\big]\big|\big[\mathbf{1}^{(n)}\ 1\big]\big) \\
&\qquad \cdot (x_{4,n+1} - x_{2,n+1})^+ && (197)\\
&= \sum_{d=0}^{t-1} \binom{t}{d} (1-\epsilon_1)^{t-d}\epsilon_1^{n-t+d}(1-\epsilon_1) \\
&\qquad \cdot (x_{4,n+1} - x_{2,n+1})^+ && (198)\\
&= \big(\epsilon_1^{n-t} - \epsilon_1^n\big)(1-\epsilon_1)(x_{4,n+1} - x_{2,n+1})^+. && (199)
\end{aligned}$$

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

$$\begin{aligned}
\Delta\psi_4 &= \sum_{\mathbf{y}^{(n)} \in \mathcal{D}_{t,0;3}^{(4,n)}} P_{Y|X}^{n+1}\big(\big[\mathbf{y}^{(n)}\ 1\big]\big|\big[\mathbf{1}^{(n)}\ 1\big]\big) \\
&\qquad \cdot (x_{4,n+1} - x_{3,n+1})^+ && (200)\\
&= \big(\epsilon_1^{t} - \epsilon_1^n\big)(1-\epsilon_1)(x_{4,n+1} - x_{3,n+1})^+. && (201)
\end{aligned}$$

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 (194), (195), (196), (199),

and (201), is as follows:

$$\begin{cases} x_{2,n+1} - x_{1,n+1} = 1 \\ x_{4,n+1} - x_{2,n+1} = 0 \\ x_{4,n+1} - x_{3,n+1} = 1 \end{cases} \implies \begin{cases} x_{1,n+1} = 0 \\ x_{2,n+1} = 1 \\ x_{3,n+1} = 0 \\ x_{4,n+1} = 1 \end{cases} \tag{202}$$

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

$$\Delta\Psi\big(\mathcal{C}_{t+1,0}^{(4,n+1)}\big) = \frac{1}{4}\epsilon_1^t(1-\epsilon_1) + \frac{1}{4}\big(\epsilon_1^t - \epsilon_1^n\big)(1-\epsilon_1) \tag{203}$$

$$= \frac{1}{4}\big(2\epsilon_1^t - \epsilon_1^n\big)(1-\epsilon_1). \tag{204}$$

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

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

APPENDIX C
DERIVATIONS CONCERNING THE BSC

A. Proof of Theorem 37

We first consider the case $\mathsf{M} = 3$. Our proof is based on induction in n. We start with a locally optimal code of length $n-1$ and then prove that appending a column according to the choice given in Theorem 37 will result in a new locally optimal code that maximizes the total probability increase. We rely on a couple of observations that for clarity are summarized here:

1) The proof that the length-2 code given in (93) is optimal is straightforward and omitted.
2) We do not need to worry about any other codebook columns than those given in (28) because first the all-zero and the all-one column can be neglected by an argument similar to Theorem 23, and because second 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.
3) Due to (18) and Lemma 13, the average success probability of a weak flip code of parameters $[t_1, t_2, t_3]$ remains unchanged with respect to any permutation of the code parameters. Hence, without loss of generality, we may assume that $t_1 \geq t_2 \geq t_3$.
4) We need to distinguish three cases in the induction from $n-1$ to n, depending on whether $n \bmod 3 = 0$, 1, or 2.

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

$$d_m^{(n)}(\mathbf{y}) \triangleq d_{\mathrm{H}}(\mathbf{x}_m, \mathbf{y}), \quad m = 1, \ldots, \mathsf{M} \tag{205}$$

and

$$\mathbf{d}^{(n)}(\mathbf{y}) \triangleq \big(d_1^{(n)}(\mathbf{y}), d_2^{(n)}(\mathbf{y}), \ldots, d_{\mathsf{M}}^{(n)}(\mathbf{y})\big). \tag{206}$$

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 $\mathbf{d}\big(\mathcal{C}^{(\mathsf{M},n)}\big)$, which compares all possible pairing combinations of the codewords of a codebook $\mathcal{C}^{(\mathsf{M},n)}$.

We also remind the reader that $k \triangleq \lfloor \frac{n}{3} \rfloor$ and $p \triangleq \frac{\epsilon}{1-\epsilon}$. Using these shorthands, we can describe the ML decoding rule for a BSC quite simply as

$$g(\mathbf{y}) = \operatorname*{argmin}_{1 \leq m \leq \mathsf{M}} \big\{d_m^{(n)}(\mathbf{y})\big\}. \tag{207}$$

We start with an observation about a basic property of the weak flip code given in (95).

Claim 42: For the weak flip code of (95), $\mathcal{C}_{t_2^*,t_3^*}^{(3,n)}$, the largest received Hamming distance between any $\mathbf{y}$ and the nearest codeword is given by the minimum Hamming distance of the codebook:

$$\max_{\mathbf{y}} \min_{j \in \{1,2,3\}} d_j^{(n)}(\mathbf{y}) = d_{\min}\big(\mathcal{C}_{t_2^*,t_3^*}^{(3,n)}\big). \tag{208}$$

Proof: It is not too difficult to see that a $\mathbf{y}$ that achieves the maximum in (208) 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:

$$\mathbf{y}_{\max} \triangleq (\underbrace{1 \cdots 1}_{t_1^*}\ \underbrace{1 \cdots 1}_{t_2^*}\ \underbrace{0 \cdots 0}_{t_3^*}). \tag{209}$$

Then,

$$\max_{\mathbf{y}} \min_{j \in \{1,2,3\}} d_j^{(n)}(\mathbf{y})$$

$$= \min\big\{d_1^{(n)}(\mathbf{y}_{\max}), d_2^{(n)}(\mathbf{y}_{\max}), d_3^{(n)}(\mathbf{y}_{\max})\big\} \tag{210}$$

$$= \min\{t_1^* + t_2^*, t_1^* + t_3^*, t_2^* + t_3^*\} \tag{211}$$

$$= \min\Big\{d_{\mathrm{H}}\big(\mathbf{x}_2^{(n)}, \mathbf{x}_3^{(n)}\big), d_{\mathrm{H}}\big(\mathbf{x}_1^{(n)}, \mathbf{x}_3^{(n)}\big), d_{\mathrm{H}}\big(\mathbf{x}_1^{(n)}, \mathbf{x}_2^{(n)}\big)\Big\} \tag{212}$$

$$= d_{\min}\big(\mathcal{C}_{t_2^*,t_3^*}^{(3,n)}\big). \tag{213}$$

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

Next note that the (length-3) pairwise Hamming distance vector of any code $\mathcal{C}^{(3,n-1)}$ will have exactly two 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 $\mathcal{C}^{(3,n)}$. For example, if we add $\mathbf{c}_1^{(3)}$, then

$$\mathbf{d}\big(\mathcal{C}^{(3,n)}\big) = \Big(d_{\mathrm{H}}\big(\mathbf{x}_1^{(n-1)}, \mathbf{x}_2^{(n-1)}\big), d_{\mathrm{H}}\big(\mathbf{x}_1^{(n-1)}, \mathbf{x}_3^{(n-1)}\big) + 1,$$
$$d_{\mathrm{H}}\big(\mathbf{x}_2^{(n-1)}, \mathbf{x}_3^{(n-1)}\big) + 1\Big). \tag{214}$$

We are now ready for our induction proof.

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

$$[t_1^*, t_2^*, t_3^*] = [k, k, k-1] \tag{215}$$

$$\mathbf{d}\big(\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) = (2k-1, 2k-1, 2k) \tag{216}$$

$$d_{\min}\big(\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) = 2k-1. \tag{217}$$

The corresponding success probability formula looks as follows:

$$3P_{\mathrm{c}}\big(\mathcal{C}^{(3,n-1)}_{t_2^*,t_3^*}\big) = \sum_{m=1}^{3} \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k,k-1;m}} P^{n-1}_{Y|X}\big(\mathbf{y}^{(n-1)}\big|\mathbf{x}^{(n-1)}_m\big) \tag{218}$$

$$= (1-\epsilon)^{n-1} \sum_{m=1}^{3} \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k,k-1;m}} \left(\frac{\epsilon}{1-\epsilon}\right)^{d_{\mathrm{H}}(\mathbf{x}^{(n-1)}_m,\mathbf{y}^{(n-1)})} \tag{219}$$

$$= (1-\epsilon)^{n-1} \left(\sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k,k-1;1}} p^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})} + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k,k-1;2}} p^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})} + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k,k-1;3}} p^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})} \right) \tag{220}$$

$$= (1-\epsilon)^{n} \left(\sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k,k-1;1}} p^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})} + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k,k-1;1}} p^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})+1} + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k,k-1;2}} p^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})} + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k,k-1;2}} p^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})+1} + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k,k-1;3}} p^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})} + \sum_{\mathbf{y}^{(n-1)}\in\mathcal{D}^{(3,n-1)}_{k,k-1;3}} p^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})+1} \right) \tag{221}$$

where in the last equality we used the trick to write

$$1 = (1-\epsilon)\left(1+\frac{\epsilon}{1-\epsilon}\right) = (1-\epsilon)(1+p). \tag{222}$$

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

$$[t_1,t_2,t_3] = [k,k,k] \tag{223}$$

$$\mathbf{d}\big(\mathcal{C}^{(3,n)}\big) = (2k,2k,2k) \tag{224}$$

$$d_{\min}\big(\mathcal{C}^{(3,n)}\big) = 2k. \tag{225}$$

Note that we can rewrite (221) in the following way:

$$3P_{\mathrm{c}}\Big(\mathcal{C}^{(3,n-1)}_{t_2^*,t_3^*}\Big) = (1-\epsilon)^{n} \left(\sum_{\mathbf{y}^{(n)}\in\left[\mathcal{D}^{(3,n-1)}_{k,k-1;1}\ 0\right]} p^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})} + \sum_{\mathbf{y}^{(n)}\in\left[\mathcal{D}^{(3,n-1)}_{k,k-1;1}\ 1\right]} p^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})+1} + \sum_{\mathbf{y}^{(n)}\in\left[\mathcal{D}^{(3,n-1)}_{k,k-1;2}\ 1\right]} p^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})} + \sum_{\mathbf{y}^{(n)}\in\left[\mathcal{D}^{(3,n-1)}_{k,k-1;2}\ 0\right]} p^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})+1} + \sum_{\mathbf{y}^{(n)}\in\left[\mathcal{D}^{(3,n-1)}_{k,k-1;3}\ 1\right]} p^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})} + \sum_{\mathbf{y}^{(n)}\in\left[\mathcal{D}^{(3,n-1)}_{k,k-1;3}\ 0\right]} p^{d_3^{(n-1)}(\mathbf{y}^{(n-1)})+1} \right). \tag{226}$$

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

$$3P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}\big) = (1-\epsilon)^{n} \left(\sum_{\mathbf{y}^{(n)}\in\mathcal{D}^{(3,n)}_1} p^{d_1^{(n)}(\mathbf{y}^{(n)})} + \sum_{\mathbf{y}^{(n)}\in\mathcal{D}^{(3,n)}_2} p^{d_2^{(n)}(\mathbf{y}^{(n)})} + \sum_{\mathbf{y}^{(n)}\in\mathcal{D}^{(3,n)}_3} p^{d_3^{(n)}(\mathbf{y}^{(n)})} \right) \tag{227}$$

where we use $\mathcal{D}^{(3,n)}_m$ to denote the decoding region of the new code $\mathcal{C}^{(3,n)}$. In order to be able to compare (226) with (227), we need to be able to compare $\mathcal{D}^{(3,n-1)}_{k,k-1;m}$ 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 $\mathbf{c}_3^{(3)} = (0\ 1\ 1)^{\mathsf{T}}$ to the code of length $n-1$, it is obvious that

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k,k-1;1} \implies \big[\mathbf{y}^{(n-1)}\ 0\big] \in \mathcal{D}^{(3,n)}_1;\ d_1^{(n)}\big(\mathbf{y}^{(n)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \tag{228}$$

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k,k-1;2} \implies \big[\mathbf{y}^{(n-1)}\ 1\big] \in \mathcal{D}^{(3,n)}_2;\ d_2^{(n)}\big(\mathbf{y}^{(n)}\big) = d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \tag{229}$$

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k,k-1;3} \implies \big[\mathbf{y}^{(n-1)}\ 1\big] \in \mathcal{D}^{(3,n)}_3;\ d_3^{(n)}\big(\mathbf{y}^{(n)}\big) = d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big). \tag{230}$$

The other three cases in (226) are more problematic. For example,

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}^{(3,n-1)}_{k,k-1;1} \implies \big[\mathbf{y}^{(n-1)}\ 1\big] \in \mathcal{D}^{(3,n)}_1 \text{ or } \mathcal{D}^{(3,n)}_2 \text{ or } \mathcal{D}^{(3,n)}_3 \tag{231}$$

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

To be able to investigate the different possible cases depending on $d_m^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$, we introduce the shorthand

$$d \triangleq \min_{m\in\{1,2,3\}} d_m^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \tag{232}$$

to denote the distance to the closest codeword (which is the first codeword in this case because we investigate $\mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;1}^{(3,n-1)}$) and another shorthand d^+ to denote any value *strictly* larger than d. The received Hamming distance vector can now take one out of four possible forms, e.g., in the currently investigated situation of (232) where $d = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$:

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d,d,d) \text{ or } (d,d,d^+) \text{ or } (d,d^+,d) \text{ or } (d,d^+,d^+). \tag{233}$$

Since the code has been extended by $\mathbf{c}_3^{(3)} = (0\ 1\ 1)^\intercal$, it follows that if we append a 1 to $\mathbf{y}^{(n-1)}$, then only the first component of $\mathbf{d}^{(n)}\big(\mathbf{y}^{(n)}\big)$ will be increased by 1 in comparison to $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$, while the second and third component remain unchanged. This means that in the fourth case in (233), the new vector $\big[\mathbf{y}^{(n-1)}\ 1\big]$ will belong to $\mathcal{D}_1^{(3,n)}$, while in the other cases it will belong to $\mathcal{D}_2^{(3,n)}$ or $\mathcal{D}_3^{(3,n)}$. However, we will show next that the first and the second case in (233) 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 42 it follows that $d \leq 2k-1$.

Now we describe $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ using $\mathbf{y}_{\max}^{(n-1)}$ defined analogously to (209). To that goal we define a_i to be the number of positions where $\mathbf{y}^{(n-1)}$ differs from $\mathbf{y}_{\max}^{(n-1)}$ when we only consider the t_i^* positions corresponding to $\mathbf{c}_i^{(3)}$, i.e., $0 \leq a_i \leq t_i^*$, $i = 1,2,3$. So, each received vector $\mathbf{y}^{(n-1)}$ can now be described by a_1, a_2, and a_3 (e.g., the all-zero vector $\mathbf{y} = \mathbf{0}$ has $a_1 = t_1^*$, $a_2 = t_2^*$, and $a_3 = 0$).

Then, for every $\mathbf{y}^{(n-1)}$, we define a matrix

$$\begin{pmatrix} t_1^* - a_1 & t_2^* - a_2 & a_3 \\ t_1^* - a_1 & a_2 & t_3^* - a_3 \\ a_1 & t_2^* - a_2 & t_3^* - a_3 \end{pmatrix} = \begin{pmatrix} k - a_1 & k - a_2 & a_3 \\ k - a_1 & a_2 & k - 1 - a_3 \\ a_1 & k - a_2 & k - 1 - a_3 \end{pmatrix} \tag{234}$$

from which the received Hamming distance vector $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ can be computed as follows:

$$\begin{pmatrix} d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \end{pmatrix} = \begin{pmatrix} k - a_1 & k - a_2 & a_3 \\ k - a_1 & a_2 & k - 1 - a_3 \\ a_1 & k - a_2 & k - 1 - a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \tag{235}$$

It is straightforward to prove the following claim.

Claim 43: There exists no integer solution (a_1, a_2, a_3), $0 \leq a_1 \leq k$, $0 \leq a_2 \leq k$, $0 \leq a_3 \leq k-1$, that satisfies

$$\begin{pmatrix} k - a_1 & k - a_2 & a_3 \\ k - a_1 & a_2 & k - 1 - a_3 \\ a_1 & k - a_2 & k - 1 - a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d \\ d \end{pmatrix} \text{ or } \begin{pmatrix} d \\ d \\ d^+ \end{pmatrix} \text{ or } \begin{pmatrix} d \\ d^+ \\ d \end{pmatrix} \tag{236}$$

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

$$\begin{pmatrix} k - a_1 & k - a_2 & a_3 \\ k - a_1 & a_2 & k - 1 - a_3 \\ a_1 & k - a_2 & k - 1 - a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d^+ \\ d \\ d \end{pmatrix}. \tag{237}$$

Proof: Omitted. ∎

Recalling our discussion after (233), it now follows from (236) that

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;1}^{(3,n-1)} \implies \big[\mathbf{y}^{(n-1)}\ 1\big] \in \mathcal{D}_1^{(3,n)};\ d_1^{(n)}\big(\mathbf{y}^{(n)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) + 1. \tag{238}$$

Similarly, we can argue for the second problematic case of (226):

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;2}^{(3,n-1)} \implies \big[\mathbf{y}^{(n-1)}\ 0\big] \in \mathcal{D}_1^{(3,n)} \text{ or } \mathcal{D}_2^{(3,n)} \tag{239}$$

depending on the exact value of $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$. Note that $\big[\mathbf{y}^{(n-1)}\ 0\big] \notin \mathcal{D}_3^{(3,n)}$ because we have added a 1 to the third codeword. If we append a 0 to $\mathbf{y}^{(n-1)}$, then the second and the third component of $\mathbf{d}^{(n)}\big(\mathbf{y}^{(n)}\big)$ will be increased by 1 in comparison to $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$, while the first component remains unchanged. Again, the received Hamming distance vector can take one out of four possible forms:

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d,d,d) \text{ or } (d,d,d^+) \text{ or } (d^+,d,d) \text{ or } (d^+,d,d^+). \tag{240}$$

In the first two cases, $\big[\mathbf{y}^{(n-1)}\ 0\big]$ will change to $\mathcal{D}_1^{(3,n)}$, in the other two cases, it will remain in $\mathcal{D}_2^{(3,n)}$. However, both the first and the second case are not possible according to (236). Hence, $\big[\mathbf{y}^{(n-1)}\ 0\big]$ will remain in the second decoding region.

Finally,

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;3}^{(3,n-1)} \implies \big[\mathbf{y}^{(n-1)}\ 0\big] \in \mathcal{D}_1^{(3,n)} \text{ or } \mathcal{D}_3^{(3,n)} \tag{241}$$

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

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d,d,d) \text{ or } (d^+,d,d) \text{ or } (d,d^+,d) \text{ or } (d^+,d^+,d). \tag{242}$$

In the first and third case, $\big[\mathbf{y}^{(n-1)}\ 0\big]$ will change to $\mathcal{D}_1^{(3,n)}$, while in the other two cases, it will remain in $\mathcal{D}_3^{(3,n)}$. Again, the first and the third case are not possible according to (236).

Hence, we have shown that

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;1}^{(3,n-1)} \implies \big[\mathbf{y}^{(n-1)}\ 1\big] \in \mathcal{D}_1^{(3,n)};\ d_1^{(n)}\big(\mathbf{y}^{(n)}\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) + 1 \tag{243}$$

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;2}^{(3,n-1)} \implies \big[\mathbf{y}^{(n-1)}\ 0\big] \in \mathcal{D}_2^{(3,n)};\ d_2^{(n)}\big(\mathbf{y}^{(n)}\big) = d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) + 1 \tag{244}$$

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;3}^{(3,n-1)} \implies \big[\mathbf{y}^{(n-1)}\ 0\big] \in \mathcal{D}_3^{(3,n)};\ d_3^{(n)}\big(\mathbf{y}^{(n)}\big) = d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) + 1. \tag{245}$$

Together with (228)–(230), this proves that the success probability of (227) is identical to the success probability of (226). So in spite of increasing the length $n-1$ by 1, we have not improved our performance.

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

$$[t_1, t_2, t_3] = [k+1, k, k-1] \tag{246}$$
$$\mathbf{d}\big(\mathcal{C}^{(3,n)}\big) = (2k-1, 2k, 2k+1) \tag{247}$$
$$d_{\min}\big(\mathcal{C}^{(3,n)}\big) = 2k-1. \tag{248}$$

One of the three problematic cases now is

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{k,k-1;2}^{(3,n-1)} \implies \big[\mathbf{y}^{(n-1)}\ 1\big] \in \mathcal{D}_2^{(3,n)} \text{ or } \mathcal{D}_3^{(3,n)} \tag{249}$$

depending on the exact value of $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ given in (240). 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 $\mathcal{D}_2^{(3,n)}$. According to Claim 43, 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

$$d_3^{(n)}\big(\mathbf{y}^{(n)}\big) = d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \tag{250}$$

without the additional increase by 1. This then means that the success probability of (227) is strictly larger than the success probability of $\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}$ because

$$p^{d_3^{(n)}(\mathbf{y}^{(n)})} = p^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})} > p^{d_2^{(n-1)}(\mathbf{y}^{(n-1)})+1} \tag{251}$$

and the choice of $\mathbf{c}_1^{(3)}$ is effective.

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

Appending $\mathbf{c}_2^{(3)}$*:* Finally, we look at the case when we append $\mathbf{c}_2^{(3)} = (0\ 1\ 0)^\intercal$. The new code has the following parameters:

$$[t_1, t_2, t_3] = [k, k+1, k-1] \tag{252}$$
$$\mathbf{d}\big(\mathcal{C}^{(3,n)}\big) = (2k, 2k-1, 2k+1) \tag{253}$$
$$d_{\min}\big(\mathcal{C}^{(3,n)}\big) = 2k-1. \tag{254}$$

We realize that these code parameters simply are a permutation of the parameters of the case when we append $\mathbf{c}_1^{(3)}$. Hence, the investigation will not fundamentally change, and we find an identical performance. So, both choices of vectors $\mathbf{c}_1^{(3)}$ and $\mathbf{c}_2^{(3)}$ are optimal. We decide to choose $\mathbf{c}_1^{(3)}$ for keeping the ordering $t_1 \geq t_2 \geq t_3$.

2) Case II: Step from $n-1 = 3k$ *to* $n = 3k+1$*:* In this case, we start with the code $\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}$ with parameters

$$[t_1^*, t_2^*, t_3^*] = [k+1, k, k-1] \tag{255}$$
$$\mathbf{d}\big(\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) = (2k-1, 2k, 2k+1) \tag{256}$$
$$d_{\min}\big(\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) = 2k-1. \tag{257}$$

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

$$[t_1, t_2, t_3] = [k+2, k, k-1] \tag{258}$$
$$\mathbf{d}\big(\mathcal{C}^{(3,n)}\big) = (2k-1, 2k+1, 2k+2) \tag{259}$$
$$d_{\min}\big(\mathcal{C}^{(3,n)}\big) = 2k-1. \tag{260}$$

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

$$[t_1, t_2, t_3] = [k+1, k+1, k-1] \tag{261}$$
$$\mathbf{d}\big(\mathcal{C}^{(3,n)}\big) = (2k, 2k, 2k+2) \tag{262}$$
$$d_{\min}\big(\mathcal{C}^{(3,n)}\big) = 2k. \tag{263}$$

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

$$[t_1, t_2, t_3] = [k+1, k, k] \tag{264}$$
$$\mathbf{d}\big(\mathcal{C}^{(3,n)}\big) = (2k, 2k+1, 2k+1) \tag{265}$$
$$d_{\min}\big(\mathcal{C}^{(3,n)}\big) = 2k. \tag{266}$$

The corresponding investigation of possible situations now reads as follows.

Claim 44: There exists no integer solution (a_1, a_2, a_3), $0 \leq a_1 \leq k+1$, $0 \leq a_2 \leq k$, $0 \leq a_3 \leq k-1$, that satisfies

$$\begin{pmatrix} k+1-a_1 & k-a_2 & a_3 \\ k+1-a_1 & a_2 & k-1-a_3 \\ a_1 & k-a_2 & k-1-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d \\ d \end{pmatrix} \text{ or } \begin{pmatrix} d \\ d \\ d^+ \end{pmatrix} \text{ or } \begin{pmatrix} d^+ \\ d \\ d \end{pmatrix} \tag{267}$$

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

$$\begin{pmatrix} k+1-a_1 & k-a_2 & a_3 \\ k+1-a_1 & a_2 & k-1-a_3 \\ a_1 & k-a_2 & k-1-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d^+ \\ d \end{pmatrix}. \tag{268}$$

The investigation is similar and shows that appending $\mathbf{c}_2^{(3)}$ is strictly suboptimal, while appending $\mathbf{c}_1^{(3)}$ and $\mathbf{c}_3^{(3)}$ are equivalent and optimal. Note that a more detailed examination can be found in Appendix C-B.

3) Case III: Step from $n-1 = 3k+1$ *to* $n = 3k+2$*:* In this case, we start with the code $\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}$ with parameters

$$[t_1^*, t_2^*, t_3^*] = [k+1, k, k] \tag{269}$$
$$\mathbf{d}\big(\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) = (2k, 2k+1, 2k+1) \tag{270}$$
$$d_{\min}\big(\mathcal{C}_{t_2^*,t_3^*}^{(3,n-1)}\big) = 2k. \tag{271}$$

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

$$[t_1, t_2, t_3] = [k+2, k, k] \tag{272}$$
$$\mathbf{d}\big(\mathcal{C}^{(3,n)}\big) = (2k, 2k+2, 2k+2) \tag{273}$$
$$d_{\min}\big(\mathcal{C}^{(3,n)}\big) = 2k. \tag{274}$$

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

$$[t_1, t_2, t_3] = [k+1, k+1, k] \tag{275}$$
$$\mathbf{d}\big(\mathcal{C}^{(3,n)}\big) = (2k+1, 2k+1, 2k+2) \tag{276}$$
$$d_{\min}\big(\mathcal{C}^{(3,n)}\big) = 2k+1. \tag{277}$$

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

$$[t_1, t_2, t_3] = [k+1, k, k+1] \tag{278}$$
$$\mathbf{d}\big(\mathcal{C}^{(3,n)}\big) = (2k+1, 2k+2, 2k+1) \tag{279}$$
$$d_{\min}\big(\mathcal{C}^{(3,n)}\big) = 2k+1. \tag{280}$$

The corresponding investigation of possible situations now reads as follows.

Claim 45: There exists no integer solution (a_1, a_2, a_3), $0 \le a_1 \le k+1$, $0 \le a_2 \le k$, $0 \le a_3 \le k$, that satisfies

$$\begin{pmatrix} k+1-a_1 & k-a_2 & a_3 \\ k+1-a_1 & a_2 & k-a_3 \\ a_1 & k-a_2 & k-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d \\ d \end{pmatrix} \text{ or } \begin{pmatrix} d^+ \\ d \\ d \end{pmatrix} \text{ or } \begin{pmatrix} d \\ d^+ \\ d \end{pmatrix} \tag{281}$$

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

$$\begin{pmatrix} k+1-a_1 & k-a_2 & a_3 \\ k+1-a_1 & a_2 & k-a_3 \\ a_1 & k-a_2 & k-a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} d \\ d \\ d^+ \end{pmatrix}. \tag{282}$$

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:

$$\begin{pmatrix} d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_4^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \end{pmatrix} = \begin{pmatrix} t_1-a_1 & t_2-a_2 & a_3 \\ t_1-a_1 & a_2 & t_3-a_3 \\ a_1 & t_2-a_2 & t_3-a_3 \\ a_1 & a_2 & a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \tag{283}$$

The derivation follows then exactly the same lines as for $\mathsf{M} = 3$. The 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 11 like, e.g., $\mathbf{c} = (0\ 0\ 0\ 1)^\mathsf{T}$ and prove that they are strictly suboptimal. The details are omitted.

B. Proof of Theorem 38

This proof will use the same approach as Appendix C-A but is much more elaborate. Unlike for the ZC, we do not have a closed-form expression for the exact average success probability for given a general codebook $\mathcal{C}^{(\mathsf{M},n)}$. Hence, to solve the global optimization problem for discrete variables, we still use the method based on induction in n. In contrast to Theorem 37, and to be able to compare the total probability increase for all possible codebooks, we use the recursive construction in blocklength n for not only the locally optimal codebooks $\mathcal{C}_{\text{BSC}}^{(3,n)\Diamond}$ given in Theorem 37, but also other locally optimal codebooks.

We again first consider the case $\mathsf{M} = 3$, and for brevity, we only discuss the case of $n = 3k$. We summarize some important observations for our long proof:

1) A principal lemma shows how to simplify the recursive construction in the blocklength n by fixing one of the code parameters.
2) Because $t_1 = n - t_2 - t_3$ and by fixing the code parameter t_3, the only free discrete variable left is t_2. We try to find the best code parameters $[t_1^\Diamond, t_2^\Diamond, t_3]$ by examining all possible code parameters for the given t_3.
3) We will list all possible best code parameters when we fix the code parameter t_3.
4) Finally, we allow t_3 to be a free discrete variable again and then prove that the optimal code parameters are equal to $[t_1^*, t_2^*, t_3^*] = [k+1, k, k-1]$.

The following lemma describes the optimal strategy for appending a new nth column to a given code $\mathcal{C}_{t_2,t_3}^{(3,n-1)}$ under the constraint that one of the code parameters must remain fixed.

Lemma 46: For $n \ge 3$, consider the general code parameters $[t_1, t_2, t_3]$ with $t_1 \ge t_2 \ge t_3$, and $t_1 + t_2 + t_3 = n - 1$. Fix one of the code parameters and append a new nth column being one of the remaining two other column types. The following choice will result in a maximal total probability increase:

1) If t_3 is fixed, append

$$\begin{cases} \mathbf{c}_1^{(3)} & \text{if } (t_1 - t_3) \text{ is even and } (t_2 - t_3) \text{ is odd} \\ \mathbf{c}_2^{(3)} & \text{if } (t_2 - t_3) \text{ is even} \\ \mathbf{c}_1^{(3)} \equiv \mathbf{c}_2^{(3)} & \text{if } (t_1 - t_3) \text{ and } (t_2 - t_3) \text{ are odd.} \end{cases} \tag{284}$$

2) If t_2 is fixed, append

$$\begin{cases} \mathbf{c}_1^{(3)} & \text{if } (t_1 - t_2) \text{ is even but } (t_2 - t_3) \text{ is odd} \\ \mathbf{c}_3^{(3)} & \text{if } (t_2 - t_3) \text{ is even} \\ \mathbf{c}_1^{(3)} \equiv \mathbf{c}_3^{(3)} & \text{if } (t_1 - t_2) \text{ and } (t_2 - t_3) \text{ are odd.} \end{cases} \tag{285}$$

3) If t_1 is fixed, append

$$\begin{cases} \mathbf{c}_2^{(3)} & \text{if } (t_1 - t_2) \text{ is even and } (t_1 - t_3) \text{ is odd} \\ \mathbf{c}_3^{(3)} & \text{if } (t_1 - t_3) \text{ is even} \\ \mathbf{c}_2^{(3)} \equiv \mathbf{c}_3^{(3)} & \text{if } (t_1 - t_2) \text{ and } (t_1 - t_3) \text{ are odd.} \end{cases} \tag{286}$$

Proof: Analogously to (234), the general code parameters $[t_1, t_2, t_3]$ with received Hamming distances $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$

can be computed as follows

$$\begin{pmatrix} d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \end{pmatrix} = \begin{pmatrix} t_1 - a_1 & t_2 - a_2 & a_3 \\ t_1 - a_1 & a_2 & t_3 - a_3 \\ a_1 & t_2 - a_2 & t_3 - a_3 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \tag{287}$$

Next we clarify the optimal decoding regions $\mathcal{D}_m^{(3,n)}$ depending on the appended n-th column.

Appending $\mathbf{c}_1^{(3)}$*:* Following the discussion in Appendix C-A, we know that one of the problematic cases is

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{t_2,t_3;1}^{(3,n-1)} \implies \big[\mathbf{y}^{(n-1)}\ 1\big] \in \mathcal{D}_1^{(3,n)} \text{ or } \mathcal{D}_3^{(3,n)}. \tag{288}$$

Depending on the exact value of $d_m^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ in (233), the first and third case in (233) will cause $\big[\mathbf{y}^{(n-1)}\ 1\big]$ to change to $\mathcal{D}_3^{(3,n)}$.

The other two cases are

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{t_2,t_3;2}^{(3,n-1)} \implies \big[\mathbf{y}^{(n-1)}\ 1\big] \in \mathcal{D}_2^{(3,n)} \text{ or } \mathcal{D}_3^{(3,n)} \tag{289}$$

and

$$\text{if } \mathbf{y}^{(n-1)} \in \mathcal{D}_{t_2,t_3;3}^{(3,n-1)} \implies \big[\mathbf{y}^{(n-1)}\ 0\big] \in \mathcal{D}_1^{(3,n)} \text{ or } \mathcal{D}_2^{(3,n)} \text{ or } \mathcal{D}_3^{(3,n)}. \tag{290}$$

For the situation of (289), the first and third case in (240) will cause $\big[\mathbf{y}^{(n-1)}\ 1\big]$ to change to $\mathcal{D}_3^{(3,n)}$. For the situation of (290), except the fourth case, all other cases in (242) will cause $\big[\mathbf{y}^{(n-1)}\ 0\big]$ to change to $\mathcal{D}_1^{(3,n)}$ or $\mathcal{D}_2^{(3,n)}$. Note that the third case in (233) and third case in (242) are identical, and that without loss of generality, the length-$(n-1)$ received vectors that with equal optimality can be put in $\mathcal{D}_{t_2,t_3;1}^{(3,n-1)}$ or $\mathcal{D}_{t_2,t_3;3}^{(3,n-1)}$ are assigned to $\mathcal{D}_{t_2,t_3;1}^{(3,n-1)}$. We then only compute the increase in success probability from $\mathcal{D}_{t_2,t_3;1}^{(3,n-1)}$ to $\mathcal{D}_3^{(3,n)}$ when $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d, d^+, d)$.

Finally, to figure out what the total probability increase is when appending $\mathbf{c}_1^{(3)}$, the only cases that we have to take into account are

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d,d,d) \text{ or } (d,d^+,d) \text{ or } (d^+,d,d). \tag{291}$$

Hence—similarly to the derivations for (250) and following—we study the conditions that result in an integer solution (a_1, a_2, a_3) with a corresponding increase in success probability. We first investigate the second case:

$$d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = d = d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \tag{292}$$

$$\implies (t_1 - a_1) + a_3 = a_1 + (t_3 - a_3) \tag{293}$$

$$\implies (t_1 - t_3) = 2(a_1 - a_3) \tag{294}$$

$$\implies a_1 - a_3 = \frac{t_1 - t_3}{2}. \tag{295}$$

There exist integer solutions a_i if $t_1 - t_3$ is even. On the other hand, there exist no integer solutions a_i if $t_1 - t_3$ is odd. Similarly, in the third case of (291), no integer solutions a_i exist if $t_1 - t_2$ is even. Hence, we have shown that integer solutions exist such that

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = \begin{cases} (d, d^+, d) \\ (d^+, d, d) \\ (d, d, d) \end{cases} \text{ if } \begin{cases} (t_1 - t_3) \text{ is even} \\ (t_1 - t_2) \text{ is even} \\ (t_1 - t_2),\ (t_2 - t_3),\ (t_1 - t_3) \text{ are all even.} \end{cases} \tag{296}$$

Appending $\mathbf{c}_2^{(3)}$*:* Using the same argument, we can also show that

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = \begin{cases} (d, d, d^+) \\ (d^+, d, d) \\ (d, d, d) \end{cases} \text{ if } \begin{cases} (t_2 - t_3) \text{ is even} \\ (t_1 - t_2) \text{ is even} \\ (t_1 - t_2),\ (t_2 - t_3),\ (t_1 - t_3) \text{ are all even.} \end{cases} \tag{297}$$

Appending $\mathbf{c}_3^{(3)}$*:* In this case, we can show that

$$\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = \begin{cases} (d, d, d^+) \\ (d, d^+, d) \\ (d, d, d) \end{cases} \text{ if } \begin{cases} (t_2 - t_3) \text{ is even} \\ (t_1 - t_3) \text{ is even} \\ (t_1 - t_2),\ (t_2 - t_3),\ (t_1 - t_3) \text{ are all even.} \end{cases} \tag{298}$$

In the following, we will investigate the case when we fix t_3. The other two cases of Lemma 46 are similar in principal and omitted.

To prove the first statement, we fix t_3 and only allow the code parameters t_1 or t_2 to be increased by 1. Comparing (296) (and its corresponding integer solutions (295)) with (297) (and its integer solutions), it can be quickly deduced that the cases where at least one of $(t_1 - t_3)$ and $(t_2 - t_3)$ is odd exhibit an obvious behavior. For example, if $(t_1 - t_3)$ is even and $(t_2 - t_3)$ odd, then the total probability increase when appending $\mathbf{c}_1^{(3)}$ is strictly larger than when appending $\mathbf{c}_2^{(3)}$. (Actually, in this situation, appending $\mathbf{c}_2^{(3)}$ results in an unchanged success probability because $(t_1 - t_2)$ cannot be even.)

The problematic case is when both $(t_1 - t_3)$ and $(t_2 - t_3)$ are even. Note that then there are only two possible values of the code parameters $[t_1, t_2, t_3]$: either all t_i are even or all t_i are odd. We are going to show that appending $\mathbf{c}_2^{(3)}$ will result in a larger total probability increase.

First we introduce the shorthands

$$u \triangleq \frac{t_1 - t_3}{2}, \quad v \triangleq \frac{t_2 - t_3}{2}, \quad \bar{d} \triangleq \frac{t_1 + t_2}{2} + t_3. \tag{299}$$

Note that in the special case of $t_1 = t_2$, appending $\mathbf{c}_1^{(3)}$ or $\mathbf{c}_2^{(3)}$ are equivalent since $[t_1 + 1, t_1, t_3] \equiv [t_1, t_1 + 1, t_3]$. Without loss of generality we can therefore assume that $t_1 > t_2$. Then we have $u > v$.

From (292)–(295), we obtain

$$d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = d = (t_2 - a_2) + \frac{t_1+t_3}{2} \tag{300}$$

$$= d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \tag{301}$$

$$d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = d^+ = a_2 + (t_1+t_3) - (a_1+a_3) \tag{302}$$

with the solutions (a_1, a_2, a_3) satisfying

$$a_1 = u + a_3, \quad a_2 \geq v + a_3. \tag{303}$$

Setting $a_2 \triangleq v + r$ with $r \geq a_3$ then yields

$$d = (t_2 - v - r) + \frac{t_1+t_3}{2} = \frac{t_1+t_2}{2} + t_3 - r \tag{304}$$

$$d^+ = v + r + (t_1+t_3) - (u + 2a_3)$$
$$= \frac{t_1+t_2}{2} + t_3 + (r - 2a_3). \tag{305}$$

Recalling that the range of the integer solutions a_i is $0 \leq a_i \leq t_i$, we now get the corresponding $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ as follows:

$$\begin{pmatrix} d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \end{pmatrix} = \begin{pmatrix} d \\ d^+ \\ d \end{pmatrix} = \begin{pmatrix} \bar{d} - r \\ \bar{d} + \big(r - 2a_3\big) \\ \bar{d} - r \end{pmatrix} \tag{306}$$

for $0 \leq a_3 \leq t_3$, $a_3 \leq r \leq \frac{t_2+t_3}{2}$.

In this situation, depending on the solutions of $(a_1, a_2, a_3) = (u + a_3, v + r, a_3)$, there are

$$\binom{t_1}{u+a_3}\binom{t_2}{v+r}\binom{t_3}{a_3} \tag{307}$$

different $\mathbf{y}^{(n-1)}$ with $d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = d$ such that

$$\mathbf{y}^{(n-1)} \in \mathcal{D}_{t_2,t_3;1}^{(3,n-1)} \implies [\mathbf{y}^{(n-1)}\ 1] \in \mathcal{D}_3^{(3,n)} \tag{308}$$

$$d_3^{(n)}\big([\mathbf{y}^{(n-1)}\ 1]\big) = d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = d. \tag{309}$$

The increase of success probability for each such $\mathbf{y}^{(n-1)}$ is then

$$p^{d_3^{(n)}([\mathbf{y}^{(n-1)}\ 1])} - p^{d_1^{(n-1)}(\mathbf{y}^{(n-1)})+1} = p^d - p^{d+1}. \tag{310}$$

Hence, the total probability increase for the new decoding region $\mathcal{D}_3^{(3,n)}$ is

$$\Delta\psi_3\big(\mathcal{C}_{t_2,t_3}^{(3,n)}\big) = \sum_{a_3=0}^{t_3} \sum_{r\geq a_3}^{\frac{t_2+t_3}{2}} \binom{t_1}{u+a_3}\binom{t_2}{v+r}\binom{t_3}{a_3} \cdot (1-\epsilon)^n \left(p^{\bar{d}-r} - p^{\bar{d}-r+1}\right) \tag{311}$$

$$= \sum_{r=0}^{\frac{t_2+t_3}{2}} \sum_{a_3=0}^{\min\{r,t_3\}} \binom{t_1}{u+a_3}\binom{t_2}{v+r}\binom{t_3}{a_3} \cdot (1-\epsilon)^n \big(p^{\bar{d}-r} - p^{\bar{d}-r+1}\big) \tag{312}$$

where in (311) we have interchanged the summations.

Note that in (306) we on purpose allow r and a_3 to be zero, even though this corresponds to $d = \bar{d} = d^+$. This slight misuse of notation allows us to incorporate the first case $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d, d, d)$ in (291) into the second case.

Finally, in the third case of $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = (d, d, d^+)$, we have

$$d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = d = (t_1 - a_1) + \frac{t_2+t_3}{2}$$
$$= d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \tag{313}$$

$$d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) = d^+ = a_1 + (t_2+t_3) - (a_2+a_3) \tag{314}$$

with solutions (a_1, a_2, a_3) satisfying

$$a_1 \triangleq u + r \geq u + a_3, \quad a_2 = v + a_3. \tag{315}$$

Consequently, the corresponding $\mathbf{d}^{(n-1)}\big(\mathbf{y}^{(n-1)}\big)$ is

$$\begin{pmatrix} d_1^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_2^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \\ d_3^{(n-1)}\big(\mathbf{y}^{(n-1)}\big) \end{pmatrix} = \begin{pmatrix} \bar{d} - r \\ \bar{d} - r \\ \bar{d} + \big(r - 2a_3\big) \end{pmatrix} \tag{316}$$

with $0 \leq a_3 \leq t_3$, $a_3 \leq r \leq \frac{t_1+t_3}{2}$.

As above, the total probability increase of the new decoding region $\mathcal{D}_2^{(3,n)}$ can now be derived as

$$\Delta\psi_2\big(\mathcal{C}_{t_2+1,t_3}^{(3,n)}\big) = \sum_{a_3=0}^{t_3} \sum_{r\geq a_3}^{\frac{t_1+t_3}{2}} \binom{t_1}{u+r}\binom{t_2}{v+a_3}\binom{t_3}{a_3} \cdot (1-\epsilon)^n \big(p^{\bar{d}-r} - p^{\bar{d}-r+1}\big) \tag{317}$$

$$= \sum_{r=0}^{\frac{t_1+t_3}{2}} \sum_{a_3=0}^{\min\{r,t_3\}} \binom{t_1}{u+r}\binom{t_2}{v+a_3}\binom{t_3}{a_3} \cdot (1-\epsilon)^n \big(p^{\bar{d}-r} - p^{\bar{d}-r+1}\big). \tag{318}$$

To complete the proof of Statement 1) of Lemma 46, it only remains to show that under the assumption $t_1 > t_2$ we have (318) $>$ (312). A first step towards this goal is achieved by Claim 47.

Claim 47: Let $t_1 > t_2$ be two nonnegative integers, both even or both odd, and let ν_1, ν_2 be two nonnegative integers with $\frac{t_2}{2} \geq \nu_1 > \nu_2 \geq 0$. Then

$$\binom{t_1}{\lceil \frac{t_1}{2} \rceil + \nu_1}\binom{t_2}{\lceil \frac{t_2}{2} \rceil + \nu_2} - \binom{t_1}{\lceil \frac{t_1}{2} \rceil + \nu_2}\binom{t_2}{\lceil \frac{t_2}{2} \rceil + \nu_1}$$
$$= \binom{t_1}{\lfloor \frac{t_1}{2} \rfloor - \nu_1}\binom{t_2}{\lceil \frac{t_2}{2} \rceil + \nu_2} - \binom{t_1}{\lceil \frac{t_1}{2} \rceil + \nu_2}\binom{t_2}{\lfloor \frac{t_2}{2} \rfloor - \nu_1} \tag{319}$$

$$= \binom{t_1}{\lfloor \frac{t_1}{2} \rfloor - \nu_1}\binom{t_2}{\lfloor \frac{t_2}{2} \rfloor - \nu_2} - \binom{t_1}{\lfloor \frac{t_1}{2} \rfloor - \nu_2}\binom{t_2}{\lfloor \frac{t_2}{2} \rfloor - \nu_1} \tag{320}$$

$$= \binom{t_1}{\lceil \frac{t_1}{2} \rceil + \nu_1}\binom{t_2}{\lfloor \frac{t_2}{2} \rfloor - \nu_2} - \binom{t_1}{\lfloor \frac{t_1}{2} \rfloor - \nu_2}\binom{t_2}{\lceil \frac{t_2}{2} \rceil + \nu_1} \tag{321}$$

$$> 0. \tag{322}$$

Proof: Note that the equality $\binom{t}{\lfloor \frac{t}{2} \rfloor - \nu} = \binom{t}{\lceil \frac{t}{2} \rceil + \nu}$ follows from the definition of the binomial coefficient. We then only need to prove (322) for the case that t_1, t_2 are both even. We

write

$$\binom{t_1}{\frac{t_1}{2}+\nu_1}\binom{t_2}{\frac{t_2}{2}+\nu_2} = \frac{\left(\frac{t_1}{2}-\nu_1+1\right)\cdots\left(\frac{t_1}{2}\right)}{\left(\frac{t_1}{2}+\nu_1\right)\cdots\left(\frac{t_1}{2}+1\right)}\cdot\frac{t_1!}{\frac{t_1}{2}!\frac{t_1}{2}!} \cdot\frac{\left(\frac{t_2}{2}-\nu_2+1\right)\cdots\left(\frac{t_2}{2}\right)}{\left(\frac{t_2}{2}+\nu_2\right)\cdots\left(\frac{t_2}{2}+1\right)}\cdot\frac{t_2!}{\frac{t_2}{2}!\frac{t_2}{2}!} \tag{323}$$

$$\binom{t_1}{\frac{t_1}{2}+\nu_2}\binom{t_2}{\frac{t_2}{2}+\nu_1} = \frac{\left(\frac{t_1}{2}-\nu_2+1\right)\cdots\left(\frac{t_1}{2}\right)}{\left(\frac{t_1}{2}+\nu_2\right)\cdots\left(\frac{t_1}{2}+1\right)}\cdot\frac{t_1!}{\frac{t_1}{2}!\frac{t_1}{2}!} \cdot\frac{\left(\frac{t_2}{2}-\nu_1+1\right)\cdots\left(\frac{t_2}{2}\right)}{\left(\frac{t_2}{2}+\nu_1\right)\cdots\left(\frac{t_2}{2}+1\right)}\cdot\frac{t_2!}{\frac{t_2}{2}!\frac{t_2}{2}!} \tag{324}$$

and divide (323) by (324). Since $t_1 > t_2$ and $\nu_1 > \nu_2$, we obtain

$$\frac{\binom{t_1}{\frac{t_1}{2}+\nu_1}\binom{t_2}{\frac{t_2}{2}+\nu_2}}{\binom{t_1}{\frac{t_1}{2}+\nu_2}\binom{t_2}{\frac{t_2}{2}+\nu_1}} = \frac{\left(\frac{t_1}{2}-\nu_1+1\right)\cdots\left(\frac{t_1}{2}-\nu_2\right)}{\left(\frac{t_1}{2}+\nu_1\right)\cdots\left(\frac{t_1}{2}+\nu_2+1\right)} > 1 \tag{325}$$

where the inequality follows from the fact that provided that $e < b < c$,

$$\frac{b}{c} > \frac{b-e}{c-e} \tag{326}$$

for b, c, e being positive integers. ∎

In the remainder of the proof, we only treat the case when t_i are all even. We subtract (312) from (318):

$$\begin{aligned}
&\Delta\psi_2\big(\mathcal{C}^{(3,n)}_{t_2+1,t_3}\big) - \Delta\psi_3\big(\mathcal{C}^{(3,n)}_{t_2,t_3}\big) \\
&\quad= \sum_{r=0}^{\frac{t_2+t_3}{2}}\sum_{a_3=0}^{\min\{r,t_3\}}\left[\binom{t_1}{u+r}\binom{t_2}{v+a_3} - \binom{t_1}{u+a_3}\binom{t_2}{v+r}\right] \\
&\qquad\qquad\cdot\binom{t_3}{a_3}(1-\epsilon)^n\big(p^{\bar{d}-r}-p^{\bar{d}-r+1}\big) \\
&\qquad+\sum_{r=\frac{t_2+t_3}{2}+1}^{\frac{t_1+t_3}{2}}\sum_{a_3=0}^{\min\{r,t_3\}}\binom{t_1}{u+r}\binom{t_2}{v+a_3}\binom{t_3}{a_3} \\
&\qquad\qquad\cdot(1-\epsilon)^n\left(p^{\bar{d}-r}-p^{\bar{d}-r+1}\right).
\end{aligned} \tag{327}$$

Observe that the second double-sum of (327) is strictly larger than zero. The first double-sum can be rewritten as follows:

$$\begin{aligned}
&\sum_{r=0}^{t_3}\sum_{a_3=0}^{r}\left[\binom{t_1}{u+r}\binom{t_2}{v+a_3} - \binom{t_1}{u+a_3}\binom{t_2}{v+r}\right] \\
&\qquad\cdot\binom{t_3}{a_3}(1-\epsilon)^n\big(p^{\bar{d}-r}-p^{\bar{d}-r+1}\big) \\
&+\sum_{r=t_3+1}^{\frac{t_2+t_3}{2}}\sum_{a_3=0}^{t_3}\left[\binom{t_1}{u+r}\binom{t_2}{v+a_3} - \binom{t_1}{u+a_3}\binom{t_2}{v+r}\right] \\
&\qquad\cdot\binom{t_3}{a_3}(1-\epsilon)^n\big(p^{\bar{d}-r}-p^{\bar{d}-r+1}\big).
\end{aligned} \tag{328}$$

In the first term of (328), the factor consisting only of binomial coefficients is equal to

$$\begin{aligned}
&\left[\binom{t_1}{\frac{t_1}{2}+r-\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+a_3-\frac{t_3}{2}}\right. \\
&\quad\left.-\binom{t_1}{\frac{t_1}{2}+a_3-\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+r-\frac{t_3}{2}}\right]\binom{t_3}{a_3}
\end{aligned} \tag{329}$$

with $t_3 \geq r \geq a_3 \geq 0$. For the case $r = a_3$, we have

$$\begin{aligned}
&\binom{t_1}{\frac{t_1}{2}+r-\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+a_3-\frac{t_3}{2}} \\
&\quad-\binom{t_1}{\frac{t_1}{2}+a_3-\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+r-\frac{t_3}{2}} = 0.
\end{aligned} \tag{330}$$

For the case $r + a_3 = t_3$, we have

$$\begin{aligned}
&\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-r}\binom{t_2}{\frac{t_2}{2}+\frac{t_3}{2}-r} \\
&\quad-\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-r}\binom{t_2}{\frac{t_2}{2}+r-\frac{t_3}{2}} = 0.
\end{aligned} \tag{331}$$

For the remaining cases $r - a_3 = 1, \ldots, t_3 - 1$, we will only illustrate the case $r - a_3 = 1$, i.e., $r = a_3 + 1$. Then (329) becomes

$$\begin{aligned}
&\left[\binom{t_1}{\frac{t_1}{2}+a_3+1-\frac{t_3}{2}}\binom{t_2}{\frac{t_1}{2}+a_3-\frac{t_3}{2}}\right. \\
&\quad\left.-\binom{t_1}{\frac{t_1}{2}+a_3-\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+a_3+1-\frac{t_3}{2}}\right]\binom{t_3}{a_3}
\end{aligned} \tag{332}$$

which for $a_3 \leq \frac{t_3}{2} - 1$ equals

$$\begin{aligned}
&\left[\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-a_3-1}\binom{t_2}{\frac{t_1}{2}+\frac{t_3}{2}-a_3}\right. \\
&\quad\left.-\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-a_3}\binom{t_2}{\frac{t_2}{2}+\frac{t_3}{2}-a_3-1}\right]\binom{t_3}{a_3} < 0
\end{aligned} \tag{333}$$

because of Claim 47. However, if $a_3 > \frac{t_3}{2} - 1$, then (332) equals

$$\begin{aligned}
&\left[\binom{t_1}{\frac{t_1}{2}+a_3-\frac{t_3}{2}+1}\binom{t_2}{\frac{t_1}{2}+a_3-\frac{t_3}{2}}\right. \\
&\quad\left.-\binom{t_1}{\frac{t_1}{2}+a_3-\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+a_3-\frac{t_3}{2}+1}\right]\binom{t_3}{a_3} > 0.
\end{aligned} \tag{334}$$

Recalling that the range of a_3 is from 0 to $t_3 - 1$, we now combine pairs from (333) and (334) as follows: in (333) we take the term corresponding to $a_3 = 0$ ($r = 1$), and in (334), we take the term corresponding to $a_3' = t_3 - 1$ ($r = t_3$). Adding them together yields

$$\begin{aligned}
&\left[\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-1}\binom{t_2}{\frac{t_1}{2}+\frac{t_3}{2}}\right. \\
&\quad\left.-\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+\frac{t_3}{2}-1}\right]\binom{t_3}{0} \\
&+\left[\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}}\binom{t_2}{\frac{t_1}{2}+\frac{t_3}{2}-1}\right. \\
&\quad\left.-\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-1}\binom{t_2}{\frac{t_2}{2}+\frac{t_3}{2}}\right]\binom{t_3}{t_3-1} \\
&=\left[\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}}\binom{t_2}{\frac{t_1}{2}+\frac{t_3}{2}-1}-\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-1}\binom{t_2}{\frac{t_2}{2}+\frac{t_3}{2}}\right] \\
&\quad\cdot\left[\binom{t_3}{t_3-1}-\binom{t_3}{0}\right] > 0
\end{aligned} \tag{335}$$

because $\binom{t_3}{t_3-1} > \binom{t_3}{0}$. Furthermore, since $\left(p^{\bar{d}-r} - p^{\bar{d}-r+1}\right)$ is strictly increasing in r,

$$\left[\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}}\binom{t_2}{\frac{t_1}{2}+\frac{t_3}{2}-1} - \binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}-1}\binom{t_2}{\frac{t_2}{2}+\frac{t_3}{2}}\right] \cdot \left[\binom{t_3}{t_3-1}\left(p^{\bar{d}-t_3} - p^{\bar{d}-t_3+1}\right) - \binom{t_3}{0}\left(p^{\bar{d}} - p^{\bar{d}+1}\right)\right] > 0. \quad (336)$$

Similarly, for the other cases with $a_3 + a_3' = t_3 - 1$, the terms of $a_3' > \frac{t_3}{2} - 1$ will always compensate for the terms of $a_3 \leq \frac{t_3}{2} - 1$. This shows that the whole summation still is larger than zero for the case of $r - a_3 = 1$.

The remaining cases $r - a_3 = 2, \ldots, t_3 - 1$ can be shown similarly.

The final step is showing that the second term of (328) is always strictly larger than zero, too. Again, we consider the range of the parameters: $t_3 \leq r \leq \frac{t_2+t_3}{2}$, $0 \leq a_3 \leq t_3$. For example, for $r = t_3 + 1$, (329) reads

$$\left[\binom{t_1}{\frac{t_1}{2}+\frac{t_3}{2}+1}\binom{t_2}{\frac{t_2}{2}+a_3-\frac{t_3}{2}} - \binom{t_1}{\frac{t_1}{2}+a_3-\frac{t_3}{2}}\binom{t_2}{\frac{t_2}{2}+\frac{t_3}{2}+1}\right]\binom{t_3}{a_3}. \quad (337)$$

Since $0 \leq a_3 \leq t_3$, we have $\left|a_3 - \frac{t_3}{2}\right| \leq \frac{t_3}{2}$ and therefore, by Claim 47, (337) is always lager than zero. We omit the remaining details.

This completes the proof that (318) $>$ (312).

The second and third statement of Lemma 46 can be proved in a similar way. ∎

From the proof of Lemma 46 we can also deduce that the total probability increase can be computed recursively for each case in Theorem 37.

Corollary 48: For a BSC and for any $n \geq 2$, the exact average success probability of an optimal code with three codewords $\mathsf{M} = 3$ can be derived recursively in blocklength n: we start with (99) and then apply (100)–(102).

Proof: It is quite simple to get the starting expression (99) for $n = 2$ from (93).

We only illustrate the calculation for the case $n-1 = 3k-1$ to $n = 3k$. The optimal code parameters for $n - 1 = 3k - 1$ is $[k, k, k-1]$. Since we are going to append $\mathbf{c}_1^{(3)}$ for $n = 3k$, the solutions of (a_1, a_2, a_3) for $\mathbf{d}^{(n-1)}\left(\mathbf{y}^{(n-1)}\right) = (d^+, d, d)$ satisfy

$$a_1 = a_2, \quad a_2 \leq a_3. \quad (338)$$

The corresponding $\mathbf{d}^{(n-1)}\left(\mathbf{y}^{(n-1)}\right)$ is

$$\begin{pmatrix} d_1^{(n-1)}\left(\mathbf{y}^{(n-1)}\right) \\ d_2^{(n-1)}\left(\mathbf{y}^{(n-1)}\right) \\ d_3^{(n-1)}\left(\mathbf{y}^{(n-1)}\right) \end{pmatrix} = \begin{pmatrix} 2k + \left(a_3 - 2a_2\right) \\ (2k-1) - a_3 \\ (2k-1) - a_3 \end{pmatrix} \quad (339)$$

for $0 \leq a_2 \leq t_2$, $a_2 \leq a_3 \leq t_3$.

Therefore, the total probability increase from the third decoding region $\mathcal{D}_3^{(3,n)}$ is

$$\Delta\Psi\left(\mathcal{C}_{k,k-1}^{(3,n)}\right) = \sum_{a_3=0}^{k-1}\sum_{a_2=0}^{a_3}\binom{k}{a_2}\binom{k}{a_2}\binom{k-1}{a_3} \cdot (1-\epsilon)^n\left(p^{2k-1-a_3} - p^{2k-a_3}\right). \quad (340)$$

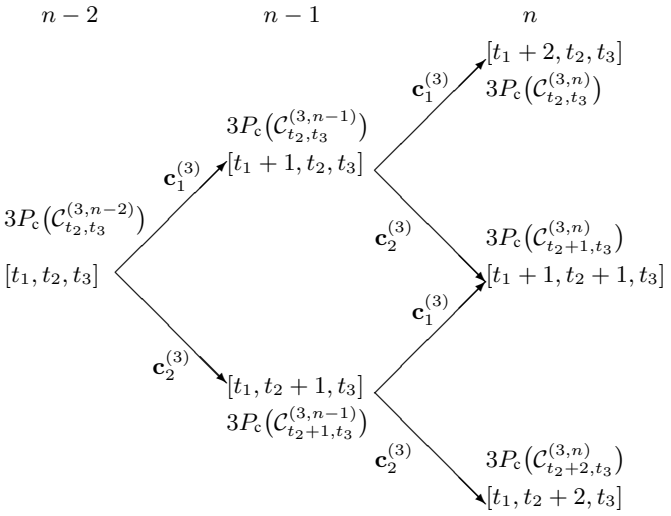

Fig. 14. All possible code parameters extensions from blocklength $n-2$ to n with fixed t_3.

The other two cases are similar to (312) and (318). ∎

Now we return to the proof of Theorem 38 and derive a strict monotonicity property.

Corollary 49: For a BSC and a blocklength $n \geq 2$, any code with parameters $[t_1, t_2, t_3]$ that satisfy $t_1 \geq t_2 + 2 \geq t_3$ satisfies

$$P_{\mathrm{c}}\left(\mathcal{C}_{t_2,t_3}^{(3,n)}\right) < P_{\mathrm{c}}\left(\mathcal{C}_{t_2+2,t_3}^{(3,n)}\right). \quad (341)$$

Proof: Consider the codebook $\mathcal{C}_{t_2,t_3}^{(3,n-2)}$ with code parameters $[t_1, t_2, t_3]$. From blocklength $n-2$ to blocklength n with a fixed number t_3, there are three possible code parameters extensions, as shown in Fig. 14. The condition of $t_1 \geq t_2+2$ is needed to make sure that at blocklength n, the code parameters ordering is still nonincreasing.

Using the same approach as in Lemma 46, we investigate four cases for blocklength $n-2$: $(t_1 - t_3)$ and $(t_2 - t_3)$ both even, $(t_1 - t_3)$ even and $(t_2 - t_3)$ odd, $(t_1 - t_3)$ odd and $(t_2 - t_3)$ even, and $(t_1 - t_3)$ and $(t_2 - t_3)$ both odd.

The two cases with one difference even and the other odd are straightforward. For example, if $(t_1 - t_3)$ is even and $(t_2 - t_3)$ is odd, then by Lemma 46 we have

$$P_{\mathrm{c}}\left(\mathcal{C}_{t_2,t_3}^{(3,n-1)}\right) > P_{\mathrm{c}}\left(\mathcal{C}_{t_2+1,t_3}^{(3,n-1)}\right). \quad (342)$$

Then both $(t_1 + 1 - t_3)$ and $(t_2 - t_3)$ are odd, and therefore

$$P_{\mathrm{c}}\left(\mathcal{C}_{t_2,t_3}^{(3,n)}\right) = P_{\mathrm{c}}\left(\mathcal{C}_{t_2+1,t_3}^{(3,n)}\right) \quad (343)$$

and (since both $(t_1 - t_3)$ and $(t_2 + 1 - t_3)$ are even) by Lemma 46

$$P_{\mathrm{c}}\left(\mathcal{C}_{t_2+2,t_3}^{(3,n)}\right) > P_{\mathrm{c}}\left(\mathcal{C}_{t_2+1,t_3}^{(3,n)}\right) \quad (344)$$

$$= P_{\mathrm{c}}\left(\mathcal{C}_{t_2,t_3}^{(3,n)}\right). \quad (345)$$

For the case when both $(t_1 - t_3)$ and $(t_2 - t_3)$ are even, we note from Lemma 46 that

$$P_{\mathrm{c}}\left(\mathcal{C}_{t_2+1,t_3}^{(3,n-1)}\right) > P_{\mathrm{c}}\left(\mathcal{C}_{t_2,t_3}^{(3,n-1)}\right) \quad (346)$$

and since (t_1+1-t_3) is even but (t_2-t_3) is odd, and (t_1-t_3) is even but (t_2+1-t_3) is odd, we have

$$P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{t_2,t_3}\big) = P_{\mathrm{c}}\big(\mathcal{C}^{(3,n-1)}_{t_2,t_3}\big) \tag{347}$$
$$< P_{\mathrm{c}}\big(\mathcal{C}^{(3,n-1)}_{t_2+1,t_3}\big) \tag{348}$$
$$= P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{t_2+2,t_3}\big). \tag{349}$$

Finally, for the case when both (t_1-t_3) and (t_2-t_3) are odd, we note that

$$P_{\mathrm{c}}\big(\mathcal{C}^{(3,n-1)}_{t_2,t_3}\big) = P_{\mathrm{c}}\big(\mathcal{C}^{(3,n-1)}_{t_2+1,t_3}\big). \tag{350}$$

Now since (t_1+1-t_3) and (t_2+1-t_3) are even and by assumption $(t_1+1-t_3) \geq (t_2+1-t_3)$, we can use a similar reasoning as given in the proof of Lemma 46 to show that

$$P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{t_2,t_3}\big) = P_{\mathrm{c}}\big(\mathcal{C}^{(3,n-1)}_{t_2,t_3}\big) + \Delta\Psi\big(\mathcal{C}^{(3,n)}_{t_2,t_3}\big) \tag{351}$$
$$< P_{\mathrm{c}}\big(\mathcal{C}^{(3,n-1)}_{t_2+1,t_3}\big) + \Delta\Psi\big(\mathcal{C}^{(3,n)}_{t_2+2,t_3}\big) \tag{352}$$
$$= P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{t_2+2,t_3}\big). \tag{353}$$

■

Corollary 49 is useful for finding the optimized code parameters $[t_1^{\diamondsuit}, t_2^{\diamondsuit}, t_3]$ for a fixed t_3.

Corollary 50: For a BSC and a blocklength $n = 3k \geq 2$, consider a code with parameters $[t_1, t_2, t_3]$ satisfying $t_1 \geq t_2 \geq t_3$ and $t_3 = k-\kappa$, $0 \leq \kappa \leq k$. Then the best choice for t_2 is[13]

$$[t_1^{\diamondsuit}, t_2^{\diamondsuit}, k-\kappa] = \begin{cases} [k,k,k] & \text{if } \kappa = 0 \\ \left[k+\left\lceil \frac{\kappa}{2}\right\rceil, k+\left\lfloor \frac{\kappa}{2}\right\rfloor, k-\kappa\right] & \text{if } \kappa \bmod 4 = 1,2,3 \\ \left[k+\left\lceil \frac{\kappa}{2}\right\rceil+1, k+\left\lfloor \frac{\kappa}{2}\right\rfloor-1, k-\kappa\right] & \text{if } \kappa \bmod 4 = 0,\ \kappa \neq 0. \end{cases} \tag{354}$$

For the remaining derivations, we introduce the following shorthand:

$$\eta \triangleq \left\lfloor \frac{\kappa}{4} \right\rfloor. \tag{355}$$

Proof: Note that $n = 3k$, i.e., $t_1 + t_2 = 2k + \kappa$. From Corollary 49, we know that for fixed t_3 the average success probability is strictly increasing when t_2 grows to t_2+2. Also since $t_1 \geq t_2 \geq t_3 = k-\kappa$, the two possible best choices of code parameters can only be either

$$\left[k+\left\lceil \frac{\kappa}{2}\right\rceil, k+\left\lfloor \frac{\kappa}{2}\right\rfloor, k-\kappa\right] \tag{356}$$
$$\text{or} \quad \left[k+\left\lceil \frac{\kappa}{2}\right\rceil+1, k+\left\lfloor \frac{\kappa}{2}\right\rfloor-1, k-\kappa\right]. \tag{357}$$

In the case of $\kappa = 0$, the only possible code parameters is (356): $[k,k,k]$ because of $k-1 < k$.

We now illustrate the case of $\kappa \bmod 4 = 1$, i.e., $\kappa = 4\eta+1$. Then (356) and (357) become

$$[k+2\eta+1, k+2\eta, k-(4\eta+1)] \tag{358}$$
$$\text{or} \quad [k+2\eta+2, k+2\eta-1, k-(4\eta+1)]. \tag{359}$$

[13]Note that t_3 is fixed and t_1 is implicitly given as $t_1 = n-t_2-t_3$.

These two best choices both stem from the same $(n-1)$-code:

$$[k+2\eta+1, k+2\eta-1, k-(4\eta+1)]. \tag{360}$$

Since $k+2\eta+1-(k-(4\eta+1)) = 6\eta+2$ and $k+2\eta-1-(k-(4\eta+1)) = 6\eta$ both are even, by Lemma 46, we have

$$P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{k+2\eta,k-(4\eta+1)}\big) > P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{k+2\eta-1,k-(4\eta+1)}\big). \tag{361}$$

The proofs of the remaining cases are similar and omitted. ■

The clue to the proof of Theorem 38 is now the following claim, which is based on Corollary 50 and Lemma 46.

Claim 51: Among the codes given in (354), the average success probability is decreasing in κ, for all $\kappa \geq 1$. In particular, we have

$$P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{t_2^{\diamondsuit},k-\kappa}\big) \geq P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{t_2^{\diamondsuit},k-(\kappa+1)}\big). \tag{362}$$

Proof: In the case of $\kappa = 4\eta+1$, the best code parameters $[t_1^{\diamondsuit}, t_2^{\diamondsuit}, t_3]$ are

$$[k+2\eta+1, k+2\eta, k-(4\eta+1)] \tag{363}$$
$$[k+2\eta+1, k+2\eta+1, k-(4\eta+2)] \tag{364}$$

respectively. These two best code parameters stem from the $(n-1)$-code

$$[k+2\eta+1, k+2\eta, k-(4\eta+2)]. \tag{365}$$

Since both $k+2\eta+1-(k+2\eta) = 1$ and $k+2\eta+1-(k-(4\eta+2)) = 6\eta+3$ are odd, by Lemma 46,

$$P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{k+2\eta,k-(4\eta+1)}\big) = P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{k+2\eta+1,k-(4\eta+2)}\big). \tag{366}$$

In the case of $\kappa = 4\eta+2$, the best code parameters $[t_1^{\diamondsuit}, t_2^{\diamondsuit}, t_3]$ are

$$[k+2\eta+1, k+2\eta+1, k-(4\eta+2)] \tag{367}$$
$$[k+2\eta+2, k+2\eta+1, k-(4\eta+3)] \tag{368}$$

respectively, which both stem from

$$[k+2\eta+1, k+2\eta+1, k-(4\eta+3)]. \tag{369}$$

Since both $k+2\eta+1-(k+2\eta+1) = 0$ and $k+2\eta+1-(k-(4\eta+3)) = 6\eta+4$ are even,

$$P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{k+2\eta,k-(4\eta+2)}\big) > P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{k+2\eta+1,k-(4\eta+3)}\big). \tag{370}$$

The remaining cases are similar, and we omit the details. We obtain

$$P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{k+2\eta+1,k-(4\eta+3)}\big) = P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{k+2\eta+1,k-(4\eta+4)}\big) \tag{371}$$
$$P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{k+2\eta+1,k-(4\eta+4)}\big) > P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{k+2\eta+2,k-(4\eta+5)}\big). \tag{372}$$

This completes the proof. ■

Now note that in the proof of Theorem 37, we have shown that

$$P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{k,k}\big) < P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{k,k-1}\big). \tag{373}$$

Therefore and because (by Claim 51) $P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{t_2^{\diamondsuit},(k-\kappa)}\big)$ is decreasing in $\kappa \geq 1$, we see that $P_{\mathrm{c}}\big(\mathcal{C}^{(3,n)}_{k,k-1}\big)$ is the largest average success probability among all possible code parameters $[t_1, t_2, t_3]$ that satisfy $t_1 \geq t_2 \geq t_3$.

Note that according to (366), for $n = 3k$, there are two global optimal choices of code parameters: $[t_1^*, t_2^*, t_3^*] = [k+1, k, k-1]$ and $[k+1, k+1, k-2]$.

The cases of $n = 3k+1$ and $3k+2$, the arguments are similar and omitted.

REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423 and 623–656, Jul. and Oct. 1948.

[2] Y. Polyanskiy, H. V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[3] M. C. Gursoy, "Throughput analysis of buffer-constrained wireless systems in the finite blocklength regime," in *Proc. IEEE Int. Conf. Commun.*, Kyoto, Japan, Jun. 5–9, 2011, pp. 1–5.

[4] T. J. Riedl, T. P. Coleman, and A. C. Singer, "Finite block-length achievable rates for queuing timing channels," in *Proc. IEEE Inf. Theory Workshop*, Paraty, Brazil, Oct. 16–20, 2011, pp. 200–204.

[5] A. Martinez and A. Guillén i Fàbregas, "Saddlepoint approximation of random coding bounds," in *Proc. Inf. Theory Appl. Worksh., Univ. California*, San Diego, CA, USA, Feb. 6–11, 2011, pp. 1–6.

[6] R. G. Gallager, *Information Theory and Reliable Communication*. New York, NY, USA: Wiley, 1968.

[7] S. Shamai (Shitz) and S. Verdú, "The empirical distribution of good codes," *IEEE Trans. Inf. Theory*, vol. 43, no. 3, pp. 836–846, May 1997.

[8] C.-L. Wu, P.-N. Chen, Y. S. Han, and Y.-X. Zheng, "On the coding scheme for joint channel estimation and error correction over block fading channels," in *Proc. IEEE Int. Symp. Personel, Indoor Mobile Radio Commun.*, Tokyo, Japan, Sep. 13–16, 2009, pp. 1272–1276.

[9] M. Dohler, R. W. Heath Jr., A. Lozano, C. B. Papadias, and R. A. Valenzuela, "Is the PHY layer dead?" *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 159–165, Apr. 2011.

[10] J. N. Laneman, "On the distribution of mutual information," in *Proc. Inf. Theory Appl. Worksh., Univ. California*, San Diego, CA, USA, Feb. 6–10 2006.

[11] D. Buckingham and M. C. Valenti, "The information-outage probability of finite-length codes over AWGN channels," in *Proc. Annu. Conf. Inf. Sci. Syst.*, Princeton, NJ, USA, Mar. 19–21, 2008, pp. 390–395.

[12] S. Lin and D. J. Costello, Jr., *Error Control Coding*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2004.

[13] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: North-Holland, 1977.

[14] P.-N. Chen, H.-Y. Lin, and S. M. Moser, "Weak flip codes and applications to optimal code design on the binary erasure channel," in *Proc. 50th Allerton Conf. Commun., Control Comput.*, Monticello, IL, USA, Oct. 1–5, 2012, pp. 160–167.

[15] C. E. Shannon, R. G. Gallager, and E. R. Berlekamp, "Lower bounds to error probability for coding on discrete memoryless channels," *Inf. Contr.*, pp. 522–552, May 1967, part II.

[16] P.-N. Chen, H.-Y. Lin, and S. M. Moser, "Equidistant codes meeting the Plotkin bound are not optimal on the binary symmetric channel," in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, Turkey, Jul. 7–13, 2013, pp. 3015–3019.

[17] S. J. MacMullan and O. M. Collins, "A comparison of known codes, random codes, and the best codes," *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 3009–3022, Oct. 1998.

[18] Y. Polyanskiy, "Saddle point in the minimax converse for channel coding," *IEEE Trans. Inf. Theory*, vol. 59, no. 5, pp. 2576–2595, May 2013.

Po-Ning Chen (S'93–M'95–SM'01) was born in Taipei, R.O.C., in 1963. He received the B.S. and M.S. degrees in electrical engineering from the National Tsing-Hua University, Taiwan, in 1985 and 1987, respectively, and the Ph.D. degree in electrical engineering from University of Maryland, College Park, in 1994. From 1985 to 1987, he was with Image Processing Laboratory in National Tsing-Hua University, where he worked on the recognition of Chinese characters. During 1989, he was with Star Tech. Inc., where he focused on the development of finger-print recognition systems. After the reception of the Ph.D. degree in 1994, he joined Wan Ta Technology Inc. as a vice general manager, conducting several projects on Point-of-Sale systems. In 1995, he became a research staff in Advanced Technology Center, Computer and Communication Laboratory, Industrial Technology Research Institute in Taiwan, where he led a project on Java-based Network Managements.

Since 1996, he has been an Associate Professor in the Department of Communications Engineering at the National Chiao-Tung University, Taiwan, and was promoted to a full professor in 2001. He was elected to be the Chair of the IEEE Communications Society Taipei Chapter in 2006 and 2007, during which the IEEE ComSoc Taipei Chapter won the 2007 IEEE ComSoc Chapter Achievement Awards (CAA) and 2007 IEEE ComSoc Chapter of the Year (CoY). He has served as the chairman of the Department of Communications Engineering, National Chiao-Tung University, during 2007–2009. Dr. Chen received the annual Research Awards from the National Science Council, Taiwan, R.O.C., five years in a row since 1996. He then received the 2000 Young Scholar Paper Award from Academia Sinica, Taiwan. His Experimental Handouts for the course of Communication Networks Laboratory have been awarded as the Annual Best Teaching Materials for Communications Education by the Ministry of Education, Taiwan, R.O.C., in 1998. He has been selected as the Outstanding Tutor Teacher of the National Chiao-Tung University in 2002. He was also the recipient of the Distinguished Teaching Award from the College of Electrical and Computer Engineering, National Chiao-Tung University, Taiwan, in 2003. His research interests generally lie in information and coding theory, large deviation theory, distributed detection and sensor networks.

Hsuan-Yin Lin (S'09) was born in Taiwan (R.O.C.). He received the B.S. major degree in electrical engineering and minor degree in mathematics from the National Tsing-Hua University, Taiwan, in 2007. After one year study for a M.S. degree in the Institute of Communications Engineering, National Chiao Tung University, Hsinchu, Taiwan, he jumped directly into the study of a Ph.D. degree in the same department, in 2008. He is currently finishing his last Ph.D. year working jointly with advisor Prof. Stefan M. Moser and co-advisor Prof. Po-Ning Chen.

During January to October 2012, Mr. Lin was a visit scholar in Information Theory and Coding (ITC) Group at the Department of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain. His research interests lie in information theory, finite blocklength information theory, and optimal coding for finite blocklength.

Stefan M. Moser (S'01–M'05–SM'10) was born in Switzerland. He received the diploma (M.Sc.) in electrical engineering (with distinction) in 1999, the M.Sc. degree in industrial management (M.B.A.) in 2003, and the Ph.D. degree (Dr. sc. techn.) in the field of information theory in 2004, all from ETH Zurich, Switzerland.

From 1999 to 2003, he was a Research and Teaching Assistant, and from 2004 to 2005, he was a Senior Research Assistant with the Signal and Information Processing Laboratory, ETH Zurich. Since 2005, he has been an Assistant Professor, since 2008 an Associate Professor, and since 2012 a Professor with the Department of Electrical and Computer Engineering, National Chiao Tung University (NCTU), Hsinchu, Taiwan. His research interests are in information theory and digital communications.

Dr. Moser is recipient of the Wu Ta-You Memorial Award by the National Science Council of Taiwan in 2012, and the Best Paper Award for Young Scholars by the IEEE Communications Society Taipei and Tainan Chapters and the IEEE Information Theory Society Taipei Chapter in 2009. Further he received various awards from National Chiao Tung University, e.g., awards for excellent teaching in 2007 and 2013, and he was presented with the Willi Studer Award of ETH and the ETH Medal both in 1999, and with the Sandoz (Novartis) Basler Maturandenpreis in 1993.