Ultra-Small Block-Codes for Binary Discrete Memoryless Channels

Po-Ning Chen, Hsuan-Yin Lin, and Stefan M. Moser
Department of Electrical Engineering
National Chiao Tung University (NCTU)
Hsinchu, Taiwan
Email: qponing@mail.nctu.edu.tw, {lin.hsuanyin, stefan.moser}@ieee.org

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

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

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, in particular he did not tell us much about the design of codes apart from the fact that good codes need to have large blocklength.

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

During the last years there has been an increased interests in the theoretical understanding of finite-length coding, see for example [2], [3]. There are several possible approaches on how one can approach the problem of finite-length codes. In [3] the authors fix an acceptable error probability and a finite blocklength and then try to find bounds on the possible transmission rates. In another approach, one fixes the transmission rate and studies how the error probability depends on the blocklength (i.e., one basically studies error exponents, but for relatively small n) [2]. Both approaches are related to Shannon's ideas in the sense that they try to make fundamental statements of what is possible and what not. The exact manner in which these systems have to be built is ignored on purpose.

Our approach in this 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 an only very small fixed number of codewords: so called *ultra-small block-codes*. For such codes we try to find a best possible design that minimizes the average error probability. Hence, we put a big emphasis on finding insights in how to actually design an optimal system.

There are interesting applications for such codes. For example, in the situation of establishing an initial connection in a wireless link, the amount of information that needs to be transmitted during the setup of the link is very much limited to usually only a couple of bits. However, these bits need to be transmitted in very short time (e.g., blocklength in the range of $n = 20$ to $n = 30$) with the highest possible reliability [4]. Note that while the motivation of this work focuses on rather smaller values of n, our results nevertheless hold for arbitrary finite n.

The study of ultra-small block-codes is interesting not only because of the above mentioned direct applications, but because their analytic description is a first step to a better fundamental understanding of optimal *nonlinear* coding schemes (with ML decoding) and of their performance based on the true error probability rather than an upper bound on the error probability derived from the union bound. To simplify our analysis, we have restricted ourselves for the moment to binary discrete memoryless channels.

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

The remainder of this paper is structured as follows: after some comments about our notation we will introduce the

channel models in Section II. In Section III we will give some code definitions that will be used for the main results that are summarized in Section IV. The proofs are omitted for space reasons [5]. Finally, Section V contains a discussion about the optimal code structure for the BSC.

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

II. Channel Models

The most general binary discrete memoryless channel is the so-called *binary asymmetric channel (BAC)*. It has a probability ϵ_0 that an input 0 will be flipped into a 1 and a (possible different) probability ϵ_1 for a flip from 1 to 0. In this paper we will exclusively focus on two special cases of the BAC. The *binary symmetric channel (BSC)* has equal cross-over probability $\epsilon_0 = \epsilon_1 = \epsilon$, see Fig. 1. For symmetry reasons and without loss of generality, we assume that $\epsilon < \frac{1}{2}$.

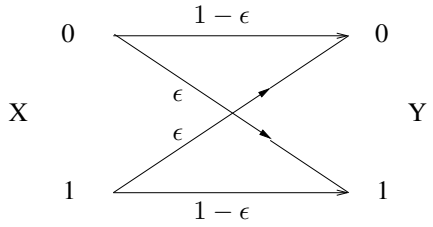

Fig. 1. The binary symmetric channel (BSC).

The *Z-channel (ZC)* will never distort an input 0, i.e., $\epsilon_0 = 0$. But the input 1 is flipped to 0 with probability $\epsilon_1 < 1$, see Fig. 2.

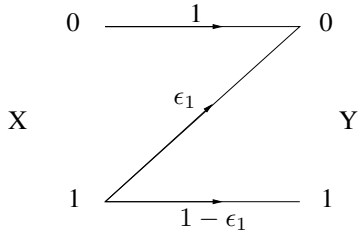

Fig. 2. The Z-channel (ZC).

We quickly review the following commonly used definitions.

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

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

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

$$\lambda_m^{(n)} \triangleq \Pr[g(\mathbf{Y}) \neq m \mid \mathbf{X} = \mathbf{x}_m] \tag{1}$$

$$= \sum_{\mathbf{y}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m) I\{g(\mathbf{y}) \neq m\}, \tag{2}$$

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

$$I\{\text{statement}\} \triangleq \begin{cases} 1 & \text{if statement is true,} \\ 0 & \text{if statement is wrong.} \end{cases} \tag{3}$$

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

$$P_{\mathrm{e}}^{(n)} \triangleq \frac{1}{\mathsf{M}} \sum_{m=1}^{\mathsf{M}} \lambda_m^{(n)}. \tag{4}$$

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

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

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

The definition of the average success probability[1] $P_{\mathrm{c}}^{(n)}$ is accordingly.

Definition 3: For a given codebook $\mathcal{C}$, we define the *decoding region* $\mathcal{D}_m$ corresponding to the m-th codeword $\mathbf{x}_m$ as follows:

$$\mathcal{D}_m \triangleq \{\mathbf{y} \colon g(\mathbf{y}) = m\}. \tag{7}$$

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

$$g(\mathbf{y}) \triangleq \arg \max_{1 \le m \le \mathsf{M}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m) \tag{8}$$

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

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

$$\mathcal{C}^{(\mathsf{M},n)} = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{\mathsf{M}} \end{pmatrix}. \tag{9}$$

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 completely equivalent to the first in the sense that it has the exact same error probability. Similarly, since we assume equally likely messages, any permutation of rows only changes the assignment of codewords to messages and has no impact on the performance. Therefore, in the remainder of this

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

paper, we will always consider such equivalent codes as being the same. In particular, when we speak of *unique design* we do not exclude the always possible permutations of columns and rows.

III. Some Binary Codes

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

Definition 4: The *flip code of type* t for $t \in \{0, 1, \ldots, \lfloor \frac{n}{2} \rfloor\}$ is a code with $\mathsf{M} = 2$ codewords defined by the following codebook matrix $\mathcal{C}_t^{(2,n)}$:

$$\mathcal{C}_t^{(2,n)} \triangleq \begin{pmatrix} \mathbf{x} \\ \bar{\mathbf{x}} \end{pmatrix} = \begin{pmatrix} 0 & \cdots & 0 & \overbrace{1 & \cdots & 1}^{t \text{ columns}} \\ 1 & \cdots & 1 & 0 & \cdots & 0 \end{pmatrix}. \tag{10}$$

Defining the column vectors

$$\left\{ \mathbf{c}_1^{(2)} \triangleq \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \mathbf{c}_2^{(2)} \triangleq \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}, \tag{11}$$

we see that a flip code of type t is given by a codebook matrix that consists of $(n-t)$ columns $\mathbf{c}_1^{(2)}$ and t columns $\mathbf{c}_2^{(2)}$. We again remind the reader that due to the memorylessness of the BSC and the ZC, the order of the columns of any codebook matrix is irrelevant. Moreover, we would like to point out that while the flip code of type 0 corresponds to a repetition code, the general flip code of type t with $t > 0$ is neither a repetition code nor is it even linear.

Definition 5: A *weak flip code of type* (t_2, t_3) for $\mathsf{M} = 3$ or $\mathsf{M} = 4$ codewords is defined by a codebook matrix $\mathcal{C}_{t_2,t_3}^{(\mathsf{M},n)}$ that consists of $t_1 \triangleq (n - t_2 - t_3)$ columns $\mathbf{c}_1^{(\mathsf{M})}$, t_2 columns $\mathbf{c}_2^{(\mathsf{M})}$, and t_3 columns $\mathbf{c}_3^{(\mathsf{M})}$, where

$$\left\{ \mathbf{c}_1^{(3)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \mathbf{c}_2^{(3)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \mathbf{c}_3^{(3)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right\} \tag{12}$$

or

$$\left\{ \mathbf{c}_1^{(4)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \mathbf{c}_2^{(4)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \mathbf{c}_3^{(4)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \right\}, \tag{13}$$

respectively.[2]

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

Example 6: Assume a BSC with cross probability $\epsilon = 0.4$, $\mathsf{M} = 4$, and a blocklength $n = 4$. Then consider the following two weak flip codes:

$$\mathcal{C}_{1,0}^{(4,4)} \triangleq \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \quad \mathcal{C}_{2,0}^{(4,4)} \triangleq \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}. \tag{14}$$

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

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

$$P_{\mathrm{e}}^{(n)}(\mathcal{C}) = \frac{1}{\mathsf{M}} \sum_{m=1}^{4} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y}) \neq m}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}_m) \tag{15}$$

$$= \frac{(1-\epsilon)^4}{4} \sum_{m=1}^{4} \sum_{\substack{\mathbf{y} \\ g(\mathbf{y}) \neq m}} \left(\frac{\epsilon}{1-\epsilon} \right)^{d_{\mathrm{H}}(\mathbf{x}_m, \mathbf{y})}, \tag{16}$$

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

If evaluated, we get an error probability $P_{\mathrm{e}}^{(n)} = 0.6112$ for $\mathcal{C}_{1,0}^{(4,4)}$ and 0.64 for $\mathcal{C}_{2,0}^{(4,4)}$. Hence, even though the minimum Hamming distance of the first codebook is smaller, its overall performance is superior to the second codebook! $\diamondsuit$

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

$$P_{\mathrm{e}}^{(n)}\big(\mathcal{C}^{(\mathsf{M},n)*}\big) \leq P_{\mathrm{e}}^{(n)}\big(\mathcal{C}^{(\mathsf{M},n)}\big), \tag{17}$$

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

IV. Main Results

A. Optimal Codes on ZC

Theorem 7: For a ZC and for any $n \geq 1$, an optimal codebook with two codewords $\mathsf{M} = 2$ is the flip codebook of type 0, $\mathcal{C}_0^{(2,n)}$. It has an error probability

$$P_{\mathrm{e}}^{(n)}\big(\mathcal{C}_0^{(2,n)}\big) = \frac{1}{2}\epsilon_1^n. \tag{18}$$

Lemma 8: For a ZC and for any $n \geq 2$, the average success probabilities of the weak flip code of type $(t, 0)$, $1 \leq t \leq \lfloor \frac{n}{2} \rfloor$, with three codewords $\mathsf{M} = 3$ or four codewords $\mathsf{M} = 4$ are

$$3P_{\mathrm{c}}^{(n)}\big(\mathcal{C}_{t,0}^{(3,n)}\big) = 1 + \sum_{d=0}^{t-1} \binom{t}{d} (1-\epsilon_1)^{t-d} \epsilon_1^d + \sum_{d=0}^{(n-t)-1} \binom{n-t}{d} (1-\epsilon_1)^{(n-t)-d} \epsilon_1^d; \tag{19}$$

$$4P_{\mathrm{c}}^{(n)}\big(\mathcal{C}_{t,0}^{(4,n)}\big) = 1 + \sum_{d=0}^{t-1} \binom{t}{d} (1-\epsilon_1)^{t-d} \epsilon_1^d + \sum_{d=0}^{n-t-1} \binom{n-t}{d} (1-\epsilon_1)^{(n-t)-d} \epsilon_1^d + \sum_{d=0}^{n-1} \left[\binom{n}{d} - \binom{n-t}{d-t} - \binom{t}{d-(n-t)} \right] (1-\epsilon_1)^{n-d} \epsilon_1^d. \tag{20}$$

Moreover, these average success probabilities are increasing with t.

Theorem 9: 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 $(t^*, 0)$ with $t^* \triangleq \lfloor \frac{n}{2} \rfloor$:

$$\mathcal{C}_{\text{ZC}}^{(\mathsf{M},n)*} = \mathcal{C}_{t^*,0}^{(\mathsf{M},n)}. \tag{21}$$

Note that for $\mathsf{M} = 2$ and $\mathsf{M} = 4$, the optimal codes given in Theorem 7 and Theorem 9 are linear. The proof of Theorem 9 shows that for even n, these linear codes are the unique optimal codes. For odd n, there are other (also nonlinear) designs that achieve the same optimal performance.

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

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

$$\mathcal{C}_{t,0}^{(4,n)} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{x} \\ \bar{\mathbf{x}} \\ \mathbf{1} \end{pmatrix} \tag{22}$$

with $\mathbf{x}$ defined in (10).

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

Corollary 10: The optimal codebooks defined in Theorem 9 for $\mathsf{M} = 3$ and 4 can be constructed recursively in the blocklength n. We start with an optimal codebook for $n = 2$:

$$\mathcal{C}_{\text{ZC}}^{(\mathsf{M},2)*} = \left(\mathbf{c}_1^{(\mathsf{M})}, \mathbf{c}_2^{(\mathsf{M})}\right). \tag{23}$$

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

$$\begin{cases} \mathbf{c}_1^{(\mathsf{M})} & \text{if } n \bmod 2 = 1, \\ \mathbf{c}_2^{(\mathsf{M})} & \text{if } n \bmod 2 = 0. \end{cases} \tag{24}$$

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

The idea of designing an optimal code recursively promises to be a very powerful approach. However, note that for larger values of M, the recursion might need a step-size larger than 1. In the following we conjecture an optimal code construction for a ZC in the case of five codewords $\mathsf{M} = 5$ with a different recursive design for n odd and n even.

We define the following five column vectors:

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

$$\left. \mathbf{c}_4^{(5)} \triangleq \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \ \mathbf{c}_5^{(5)} \triangleq \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} \right\}. \tag{25}$$

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

$$\mathcal{C}_{\text{ZC}}^{(5,8)*} = \left(\mathbf{c}_1^{(5)}, \mathbf{c}_2^{(5)}, \mathbf{c}_3^{(5)}, \mathbf{c}_1^{(5)}, \mathbf{c}_2^{(5)}, \mathbf{c}_3^{(5)}, \mathbf{c}_4^{(5)}, \mathbf{c}_5^{(5)}\right). \tag{26}$$

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

$$\begin{cases} \left(\mathbf{c}_4^{(5)}, \mathbf{c}_5^{(5)}\right) & \text{if } n \bmod 10 = 0, \\ \left(\mathbf{c}_1^{(5)}, \mathbf{c}_2^{(5)}\right) & \text{if } n \bmod 10 = 2, \\ \left(\mathbf{c}_1^{(5)}, \mathbf{c}_3^{(5)}\right) & \text{if } n \bmod 10 = 4, \\ \left(\mathbf{c}_3^{(5)}, \mathbf{c}_4^{(5)}\right) & \text{if } n \bmod 10 = 6, \\ \left(\mathbf{c}_2^{(5)}, \mathbf{c}_5^{(5)}\right) & \text{if } n \bmod 10 = 8. \end{cases} \tag{27}$$

For n odd we have

$$\mathcal{C}_{\text{ZC}}^{(5,9)*} = \left(\mathbf{c}_1^{(5)}, \mathbf{c}_2^{(5)}, \mathbf{c}_3^{(5)}, \mathbf{c}_4^{(5)}, \mathbf{c}_5^{(5)}, \mathbf{c}_1^{(5)}, \mathbf{c}_2^{(5)}, \mathbf{c}_1^{(5)}, \mathbf{c}_3^{(5)}\right). \tag{28}$$

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

$$\begin{cases} \left(\mathbf{c}_3^{(5)}, \mathbf{c}_4^{(5)}\right) & \text{if } n \bmod 10 = 1, \\ \left(\mathbf{c}_2^{(5)}, \mathbf{c}_5^{(5)}\right) & \text{if } n \bmod 10 = 3, \\ \left(\mathbf{c}_4^{(5)}, \mathbf{c}_5^{(5)}\right) & \text{if } n \bmod 10 = 5, \\ \left(\mathbf{c}_1^{(5)}, \mathbf{c}_2^{(5)}\right) & \text{if } n \bmod 10 = 7, \\ \left(\mathbf{c}_1^{(5)}, \mathbf{c}_3^{(5)}\right) & \text{if } n \bmod 10 = 9. \end{cases} \tag{29}$$

Note that the recursive structure in (27) and (29) 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 (25) exactly twice. For $n < 10$ the optimal code structure goes through some transient states.

C. Optimal Codes on BSC

Theorem 11: 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 $t \in \{0, 1, \ldots, \lfloor \frac{n}{2} \rfloor\}$.

Theorem 12: For any $n \geq 2$, define

$$t_2^* \triangleq \left\lfloor \frac{n-1}{3} \right\rfloor, \quad t_3^* \triangleq \left\lfloor \frac{n+1}{3} \right\rfloor. \tag{30}$$

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

$$\mathcal{C}_{\text{BSC}}^{(\mathsf{M},n)*} = \mathcal{C}_{t_2^*,t_3^*}^{(\mathsf{M},n)}. \tag{31}$$

Note that for $\mathsf{M} = 2$, the optimal codes given in Theorem 11 can be linear or nonlinear. For $\mathsf{M} = 4$, by the definition of weak flip code of type (t_2, t_3), the optimal codes in Theorem 12 are linear. However, due to the strong symmetry

of the BSC, there also exist nonlinear codes with the same optimal performance.

Moreover, note that one can learn from the proof of Theorem 12 that the received vector $\mathbf{y}$ that is farthest from the three codewords when $\mathsf{M} = 3$ is

$$\mathbf{y} = (\underbrace{1, 1, \cdots, 1}_{t_1^*}, \underbrace{1, 1, \cdots, 1}_{t_2^*}, \underbrace{0, 0, \ldots, 0}_{t_3^*}). \tag{32}$$

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

Corollary 13: The optimal codebooks defined in Theorem 9 for $\mathsf{M} = 3$ and $\mathsf{M} = 4$ can be constructed recursively in the blocklength n. We start with an optimal codebook for $n = 2$:

$$\mathcal{C}_{\text{BSC}}^{(\mathsf{M},2)*} = \left(\mathbf{c}_1^{(\mathsf{M})}, \mathbf{c}_3^{(\mathsf{M})}\right). \tag{33}$$

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

$$\begin{cases} \mathbf{c}_1^{(\mathsf{M})} & \text{if } n \bmod 3 = 0, \\ \mathbf{c}_2^{(\mathsf{M})} & \text{if } n \bmod 3 = 1, \\ \mathbf{c}_3^{(\mathsf{M})} & \text{if } n \bmod 3 = 2. \end{cases} \tag{34}$$

V. Pairwise Hamming Distance Structure

It is quite common in conventional coding theory to use the *minimum Hamming distance* or the *weight enumerating function (WEF)* of a code as a design and quality criterion [6]. This is motivated by the equivalence of Hamming weight and Hamming distance for linear codes, and by the union bound that converts the search for the global error probability into pairwise error probabilities. Since we are interested in the globally optimal code design and the best performance achieved by an ML decoder, we can neither use the union bound, nor can we a priori restrict our search to linear codes. Note that for most values of M, linear codes do not even exist!

In order to demonstrate that these commonly used design criteria do not work when searching for an *optimal* code, we will now investigate the minimum Hamming distance of an optimal code. Although, as (16) shows, the error probability performance of a BSC is completely specified by the Hamming distance between codewords and received vectors, it turns out that a design based on the minimum Hamming distance can fail, even for the very symmetric BSC and even for linear codes. Recall that we have seen a first glimpse of this behavior in Example 6. In the case of a more general (and not symmetric) BAC, this is even more pronounced [5]. In general one has to rely on the *pairwise Hamming distance vector*.

Definition 14: Given a codebook $\mathcal{C}^{(\mathsf{M},n)}$ with codewords $\mathbf{x}_i$ we define the *pairwise Hamming distance vector* $\mathbf{d}^{(\mathsf{M},n)}$ of length $\frac{(\mathsf{M}-1)\mathsf{M}}{2}$ as

$$\begin{aligned}\mathbf{d}^{(\mathsf{M},n)} \triangleq \Big(&d_{12}^{(n)},\ d_{13}^{(n)}, d_{23}^{(n)},\ d_{14}^{(n)}, d_{24}^{(n)}, d_{34}^{(n)}, \ldots, \\ &d_{1\mathsf{M}}^{(n)}, d_{2\mathsf{M}}^{(n)}, \ldots, d_{(\mathsf{M}-1)\mathsf{M}}^{(n)}\Big)\end{aligned} \tag{35}$$

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

For $\mathsf{M} = 3$ or $\mathsf{M} = 4$, we know from Theorem 12 that the optimal code $\mathcal{C}_{\text{BSC}}^{(\mathsf{M},n)*}$ consists of t_2^* columns $\mathbf{c}_2^{(\mathsf{M})}$, t_3^* columns $\mathbf{c}_3^{(\mathsf{M})}$, and $t_1^* \triangleq n - t_2^* - t_3^*$ columns $\mathbf{c}_1^{(\mathsf{M})}$. Using the shorthand $k \triangleq \lfloor \frac{n}{3} \rfloor$, we can write this as

$$(t_1^*, t_2^*, t_3^*) = \begin{cases} (k+1, k-1, k) & \text{if } n \bmod 3 = 0, \\ (k+1, k, k) & \text{if } n \bmod 3 = 1, \\ (k+1, k, k+1) & \text{if } n \bmod 3 = 2. \end{cases} \tag{36}$$

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

$$(t_1, t_2, t_3) = \begin{cases} (k, k, k) & \text{if } n \bmod 3 = 0, \\ (k+1, k-1, k+1) & \text{if } n \bmod 3 = 1, \\ (k+2, k, k) & \text{if } n \bmod 3 = 2. \end{cases} \tag{37}$$

This code can actually be constructed from the optimal code $\mathcal{C}_{\text{BSC}}^{(\mathsf{M},n-1)*}$ by appending a corresponding column (depending on n). In fact, by Corollary 13, we can prove that this second weak flip code is strictly suboptimal.

Next note that the pairwise Hamming distance vector of any weak flip code $\mathcal{C}_{t_2,t_3}^{(\mathsf{M},n)}$ can be computed readily from the code parameters $(t_1 \triangleq n - t_2 - t_3, t_2, t_3)$ as follows:

$$\mathbf{d}^{(3,n)} = \big(t_2 + t_3, t_1 + t_3, t_1 + t_2\big), \tag{38}$$

$$\mathbf{d}^{(4,n)} = \big(t_2 + t_3, t_1 + t_3, t_1 + t_2, t_1 + t_2, t_1 + t_3, t_2 + t_3\big). \tag{39}$$

If we now compare the pairwise Hamming distance vector of $\mathcal{C}_{\text{BSC}}^{(\mathsf{M},n)*}$ and $\mathcal{C}_{\text{subopt}}^{(\mathsf{M},n)}$ for $n \bmod 3 = 0$, we see that the minimum Hamming distance of the optimal code is $2k-1$ and therefore strictly smaller than the minimum Hamming distance $2k$ of the suboptimal code. By adapting the construction of the strictly suboptimal code $\mathcal{C}_{\text{subopt}}^{(\mathsf{M},n)}$, a similar statement can be made for the case when $n \bmod 3 = 1$.

We have the following proposition.

Proposition 15: On a BSC for $\mathsf{M} = 3$ or $\mathsf{M} = 4$ and for all n with $n \bmod 3 = 0$ or $n \bmod 3 = 1$, the codes that maximize the minimum Hamming distance $d_{\text{min}}^{(n)}$ can be strictly suboptimal. This is not true in the case of $n \bmod 3 = 2$.

Acknowledgments

This work was supported by the National Science Council under NSC 97-2221-E-009-003-MY3.

References

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

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

[3] Y. Polyanskiy, V. H. 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.

[4] 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. Pers., Indoor and Mob. Radio Commun.*, Tokyo, Japan, Sep. 13–16, 2009, pp. 1272–1276.

[5] P.-N. Chen, H.-Y. Lin, and S. M. Moser, "On ultra-small block-codes for binary discrete memoryless channels," Aug. 2011, in prep.

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