Suboptimum Algorithm for Estimating Transmitted Code Bits for Binary Linear Block Codes and Symmetric Memoryless Channels

Vladimir B. Balakirsky

Euler Institute for Discrete Mathematics
and its Applications (EIDMA)
c/o Eindhoven University of Technology
P.O.Box 513, 5600 MB Eindhoven, The Netherlands
v.b.balakirsky@tue.nl

Abstract

We consider a class of data transmission systems where one has to construct estimates of the transmitted code bits and the quality is characterized by n error probabilities : the j–th bit error is defined as the event that the j–th bit at the output of the decoder differs from the j–th transmitted bit, where $j = \overline{1,n}$ and n denotes the code length. A decoding algorithm for binary linear block codes and binary symmetric channel is proposed, and this algorithm is then generalized to arbitrary binary–input symmetric channels including channels with additive white Gaussian noise. We transform the vector received at the output of the channel to a binary vector and find the codeword at the Hamming distance equal to the half of the minimal distance of the code or less, if possible. The result of the search is the only information about the code, which is taken into account by the decoder while constructing the estimates of the transmitted bits. The values of the bit error probabilities obtained by the decoder can be exactly expressed via the weight distributions of codewords. Examples of codes and channels where these probabilities are the smallest possible or close to the smallest possible are presented. [1]

[1]The work was supported by Philips Natlab Laboratories, Eindhoven, The Netherlands, and by the DFG fund, University of Essen, Germany.

1 Introduction and summary of results

We consider the following data transmission system. Suppose that the message of a source is one of binary vectors of length k generated according to a uniform distribution. Each message is encoded by corresponding codeword of a binary linear block (n, k) code. The codeword is transmitted over a binary–input memoryless channel. There are n decoders that receive the same vector at the output of the channel. The j–th decoder makes a decision about the j–th transmitted bit, and the j–th decoding error is defined as the event that this decision is wrong, $j = \overline{1, n}$. The minimization of the bit error probabilities is the task of a designer of a system.

If the code is systematic, then the vector of length k consisting of bits formed by the first k decoders is an estimate of the transmitted message. Therefore, the minimization of the average value of the first k bit error probabilities is equivalent to the minimization of the expected Hamming distance between the transmitted message and its estimate. Another wide class of data transmission schemes where the problem under consideration has to be studied is a class of iterative and Turbo–like approaches to the decoding ([1] and other papers).

The decoding rule, which minimizes the j–th bit error probability, can be easily discovered. Namely, the j–th decoder, having received a vector $\mathbf{y}$, has to compute two values, $S_j^{(0)}(\mathbf{y})$ and $S_j^{(1)}(\mathbf{y})$, where $S_j^{(x)}(\mathbf{y})$ is the sum of 2^{k-1} conditional probabilities of receiving the vector $\mathbf{y}$ when one of codewords having bit x at the j–th position was sent, $x = 0, 1$. If $S_j^{(0)}(\mathbf{y}) > S_j^{(1)}(\mathbf{y})$, then the decoder outputs 0, otherwise he outputs 1. A conventional way of organizing computations is based on the use of the BCJR algorithm [2]. This requires computations of conditional probabilities associated with nodes of the code trellis of a block code and has exponential complexity in the code length. The performance of the algorithm can be expressed via distance properties of 2^{n-k} cosets constructed for a given code by "the standard array representation" of the space $\{0, 1\}^n$ [3], [4], [5].

We develop an alternative approach to solving the decoding problem, and our results can be summarized as follows. *For any binary linear block code and any binary–input symmetric memoryless channel defined by the matrix of crossover probabilities* $\mathbf{W}$, *there is a decoding algorithm, for which we know the function* F *such that*

$$P_{\mathrm{b},j} = F\Big(j, \{M_{j,d}^{(0)}, M_{j,d}^{(1)}\}_{d=0}^{n}, \mathbf{W}\Big),$$

where $P_{\mathrm{b},j}$ *is the* j*–th bit error probability and* $M_{j,d}^{(x)}$ *is the number of codewords having the Hamming weight* d *and containing bit* x *at the* j*–th position,* $x = 0, 1$. *There are examples of (non–perfect) codes such that the algorithm is optimum for any binary symmetric channel. The algorithm requires the bounded distance decoding (BDD) algorithm for the given code and consists of three steps : (1) the received word is transformed to a binary vector; (2) the BDD–decoder has to find the closest codeword to this binary vector, which is located at the Hamming distance at most* $\lfloor (d_{\min} - 1)/2 \rfloor$, *if possible; (3) the result of the BDD is considered as a proposal by our decoder.*

Let us present the algorithm above for some well–known codes when the codeword is transmitted over a binary symmetric channel having crossover probability $p \in (0, 1/2)$. In these examples, $\varphi(\mathbf{y}) = (\varphi_1(\mathbf{y}), \ldots, \varphi_n(\mathbf{y}))$ denotes the vector of decisions of all decoders and $\mathbf{e}$ is the noise vector such that $\mathbf{y} \oplus \mathbf{e}$ is a codeword and the Hamming weight of the vector $\mathbf{e}$, denoted by $\mathrm{wt}(\mathbf{e})$, is the smallest possible.

1). The $(6,3)$ shortened Hamming code. If $p \in (0, 0.292)$ and $\mathrm{wt}(\mathbf{e}) \leq 1$, then set $\varphi(\mathbf{y}) = \mathbf{y} \oplus \mathbf{e}$. Otherwise, set $\varphi(\mathbf{y}) = \mathbf{y}$. The algorithm is optimum for all $p \in (0, 1/2)$.

2). The $(23,12)$ Golay code. Set $\varphi(\mathbf{y}) = \mathbf{y} \oplus \mathbf{e}$ if one of the following conditions hold : (a) $p \in (0, 0.117)$; (b) $p \in (0.117, 0.227)$ and $\mathrm{wt}(\mathbf{e}) \leq 2$; (c) $p \in (0.227, 0.278)$ and $\mathrm{wt}(\mathbf{e}) \leq 1$. Otherwise, set $\varphi(\mathbf{y}) = \mathbf{y}$. The algorithm is optimum for all $p \in (0, 1/2)$.

3). The $(15,7)$ BCH code. Set $\varphi(\mathbf{y}) = \mathbf{y} \oplus \mathbf{e}$ if one of the following conditions hold : (a) $p \in (0, 0.229)$ and $\mathrm{wt}(\mathbf{e}) \leq 2$. (b) $p \in (0.229, 0.317)$ and $\mathrm{wt}(\mathbf{e}) \leq 1$. Otherwise, set $\varphi(\mathbf{y}) = \mathbf{y}$. The algorithm is suboptimum, but its performance is very close to the optimum one. In particular, the optimum algorithm also sets $\varphi_j(\mathbf{y}) = y_j \oplus e_j$ for 13 noise vectors of weight 3 if $p = 0.15$ (the total number of noise vectors of weight 3 and $e_j = 1$ is equal to 91) and for 2 noise vectors of weight 2 if $p = 0.25$ (the total number of noise vectors of weight 2 and $e_j = 1$ is equal to 14).

Our results obtained by the algorithms above for the $(6,3)$ and $(23,12)$, codes are illustrated in Figure 1, 2, where we also show the characteristics of the maximum likelihood decoding, when the decoder outputs a codeword $\mathbf{y} \oplus \mathbf{e}$. Note that in these examples the decoder always outputs either the received word or the result of maximum likelihood decoding. In general case this is not true, i.e., the decoder can construct a vector, which differs from $\mathbf{y}$ and from $\mathbf{y} \oplus \mathbf{e}$.

The algorithms described above for several specific codes and binary symmetric channels are generalized to arbitrary binary–input symmetric memoryless channels in such a way that the decoder constructs a binary vector as a result of quantization of the received word and tries to find a codeword at the Hamming distance at most one half of the minimal distance of the code. The result of decoding is formed depending on the result of the search, the parameters of the channel and the received word.

An important model of channels having non-binary output alphabet is the model of additive white Gaussian noise (AWGN) channels where 0's and 1's are transmitted by signals having the amplitudes $-\sqrt{E}$ and $+\sqrt{E}$, respectively. Notice that in the examples for binary symmetric channels the decoding "did not take place" for all values of the crossover probability of the channel greater than some threshold value (0.292 for the $(6,3)$ code, 0.278 for the $(23,12)$ code, etc.). In the case of the AWGN channels one can assume a similar situation : for some signal–to–noise ratios one has to quantize the component of the received vector at the 0 level and output the result of the quantization. We show that this is not true, i.e., for any code (we assume that the generator matrix does not contain the all–zero columns) and any signal–to–noise ratio, the optimum algorithm the bit error probability will be always less if the decoder sometimes changes the result of quantization. For a suboptimum algorithm, which we propose, the statement is slightly weaker : if the minimal distance of the code is at least 3, then there are at most n signal–to–noise ratios when the decoder cannot improve the result of quantization. This statement is illustrated in Table 1, where p denotes the crossover probability of a binary symmetric channel "created" when the received components are quantized. The value of $1 - P_{\mathrm{b},j}(\varphi)/p$ shows the gain obtained with the use of the algorithm φ as compared to the trivial decoding for the $(6,3)$ code and for the $(23,12)$ code.

The paper is organized in the following way. After some notation in Section 2 we describe the decoding algorithm for binary symmetric channels and present its analysis in Section 3. These considerations are then extended to arbitrary binary–input symmetric memoryless channels in Section 3. The basic points of the proofs are discussed in the body of the paper, and all technical details are given in the Appendix.

2 Notation

A binary linear block (n, k) code is understood as a collection of 2^k binary vectors of length n. The mapping of the information vector $\mathbf{u} \in \{0,1\}^k$ to codeword (the encoding) is introduced as $\mathbf{uG} \triangleq ((\mathbf{uG})_1, \ldots, (\mathbf{uG})_n)$, where $\mathbf{G}$ is the $k \times n$ generator matrix. Let $d_{\min}$ denote the minimum distance of the code and let

$$t_{\min} \triangleq \lfloor (d_{\min} - 1)/2 \rfloor. \tag{2.1}$$

We also denote the number of codewords of the Hamming weight d containing bit x at the j–th position by

$$M_{j,d}^{(x)} \triangleq \Big| \Big\{ \mathbf{u} \in \{0,1\}^k : (\mathbf{uG})_j = x, \, \mathrm{wt}(\mathbf{uG}) = d \Big\} \Big|, \tag{2.2}$$

where $x = 0, 1$, $d = \overline{0, n}$, and $j = \overline{1, n}$.

Let $\mathcal{E}$ denote some set of representatives of the cosets $\bigcup_{\mathbf{u} \in \{0,1\}^k} \{\mathbf{uG} \oplus \mathbf{e}\}$ for the code defined by the generator matrix $\mathbf{G}$, which have the smallest Hamming weights, i.e.,

$$\bigcup_{\mathbf{u} \in \{0,1\}^k} \bigcup_{\mathbf{e} \in \mathcal{E}} \{\mathbf{uG} \oplus \mathbf{e}\} = \{0,1\}^n, \quad 2^k |\mathcal{E}| = 2^n,$$

and

$$\mathbf{e} \in \mathcal{E} \implies \mathrm{wt}(\mathbf{e}) = \min_{\mathbf{u} \in \{0,1\}^k} \mathrm{wt}(\mathbf{uG} \oplus \mathbf{e}).$$

For all $t = \overline{0, t_{\min}}$, denote

$$\hat{\mathcal{E}} \triangleq \bigcup_{t=0}^{t_{\min}} \{0,1\}_t^n, \tag{2.3}$$

where

$$\{0,1\}_t^n \triangleq \Big\{ \mathbf{e} \in \{0,1\}^n : \mathrm{wt}(\mathbf{e}) = t \Big\}. \tag{2.4}$$

Then $\hat{\mathcal{E}} \subseteq \mathcal{E}$ for any binary linear block code having the minimum distance $d_{\min}$.

3 Decoding algorithm for binary symmetric channels

Suppose that the codeword is transmitted over a binary symmetric channel having the crossover probability $p \in (0, 1/2)$, which will be referred to as a BSC(p). Then

$$W(\mathbf{y}|\mathbf{x}) \triangleq \prod_{j=1}^{n} W_{x_j}(y_j),$$

where $W_0(0) = W_1(1) = 1 - p$ and $W_0(1) = W_1(0) = p$, is the conditional probability to receive the vector $\mathbf{y} \in \{0,1\}^n$ when the vector $\mathbf{x} \in \{0,1\}^n$ was sent. Let us introduce the decoding as the mapping

$$\varphi = \Big\{ \, \varphi_j(\mathbf{y}) \in \{0,1\}, \mathbf{y} \in \{0,1\}^n \, \Big\}_{j=1}^{n}.$$

Then

$$P_{\mathrm{b},j}(\varphi) \triangleq \sum_{\mathbf{u}^* \in \{0,1\}^k} \; \sum_{\mathbf{y} : \, \varphi_j(\mathbf{y}) \neq (\mathbf{u}^*\mathbf{G})_j} \frac{1}{2^k} W(\mathbf{y}|\mathbf{u}^*\mathbf{G}) \tag{3.1}$$

is called *the j–th bit error probability*, $j = \overline{1,n}$. Note that *the decoded vector*

$$\varphi(\mathbf{y}) = (\varphi_1(\mathbf{y}), \ldots, \varphi_n(\mathbf{y})) \in \{0,1\}^n$$

can be not a codeword. If the code is systematic, i.e., if $(\mathbf{uG})_j = u_j$ for all $j = \overline{1,k}$, then $P_{\mathrm{b},1}(\varphi), \ldots, P_{\mathrm{b},k}(\varphi)$ are also the bit error probabilities for information bits.

The following notation will be used in the description of the decoding algorithm. For all $j = \overline{1,n}$ and $t = \overline{0, t_{\min}}$, let

$$\Theta_{j,t}(p) \triangleq \sum_{d=0}^{n} M_{j,d}^{(0)} \Gamma_t^{(d)}(p) + \sum_{d=1}^{n} M_{j,d}^{(1)} \Gamma_t^{(d-1)}(p), \tag{3.2}$$

$$\alpha_{j,t}(p) \triangleq \frac{1}{\Theta_{j,t}(p)} \sum_{d=1}^{n} M_{j,d}^{(1)} \Gamma_t^{(d-1)}(p), \tag{3.3}$$

$$\begin{aligned}\Gamma_t^{(d)}(p) &\triangleq \sum_{\mathbf{e}' \in \{0,1\}_t^{n-1}} W(1^d 0^{n-1-d} \oplus \mathbf{e}' \,|\, 0^{n-1}) \\ &= (1-p)^{n-1} \sum_{w=0}^{t} \binom{d}{w} \binom{n-1-d}{t-w} \left(\frac{p}{1-p}\right)^{d+t-2w},\end{aligned} \tag{3.4}$$

where $1^d 0^{n-1-d}$ is the binary vector of length $n-1$ consisting of the 1–run of length d followed by the 0–run of length $n-1-d$ and 0^{n-1} is the all–zero vector of length $n-1$; the set $\{0,1\}_t^{n-1}$ is defined by (2.4) with $n-1$ substituted for n; the function $\Gamma_t^{(d-1)}(p)$ is defined by (3.4) with $d-1$ substituted for d. We will also assume that $\mathbf{0}$ denotes the all–zero vector of length n, and χ stands for the indicator function : $\chi\{S\} = 1$ if the statement S is true and $\chi\{S\} = 0$ otherwise.

(B1) *Given a received vector* $\mathbf{y} \in \{0,1\}^n$, *find, if possible, the noise vector* $\mathbf{e} \in \{0,1\}^n$ *of the Hamming weight at most* $t_{\min}$ *such that* $\mathbf{y} \oplus \mathbf{e}$ *is a codeword.*

(B2a) *The noise vector* $\mathbf{e}$ *is found in step* (B1) : *denote* $t = \mathrm{wt}(\mathbf{e})$ *and set*

$$e_j = 0 \implies \varphi_j(\mathbf{y}) = \begin{cases} 1 \oplus y_j, & \textit{if } p > 1 - \alpha_{j,t}(p), \\ y_j, & \textit{otherwise} \end{cases}$$

$$e_j = 1 \implies \varphi_j(\mathbf{y}) = \begin{cases} 1 \oplus y_j, & \textit{if } p > \alpha_{j,t-1}(p), \\ y_j, & \textit{otherwise} \end{cases}$$

for all $j = \overline{1,n}$.

(B2b) *The noise vector* $\mathbf{e}$ *is not found in step* (B1) : *set* $\varphi(\mathbf{y}) = \mathbf{y}$.

The rules of the (B1)–(B2) algorithm can be interpreted in the following way. Having received a $y_j \in \{0,1\}$, the decoder knows that the transmitted bit is equal to y_j with probability $1-p$. Let the decoder also know that the Hamming weight of the most likely noise vector is equal to t. If $e_j = 1$, then the information obtained from other $n-1$ received bits shows that y_j should be inverted, and the value of $1 - \alpha_{j,t-1}(p)$ characterizes the reliability of this proposal. The decoder compares p and $\alpha_{j,t-1}(p)$ and makes a decision. If $e_j = 0$, then the reliability of inversion of y_j is characterized by the value of $\alpha_{j,t}(p)$. We could not find examples of codes and channels, for which $p > 1 - \alpha_{j,t}(p)$, but we also cannot prove that it is impossible.

The statement, which will be used in the analysis of the (B1)–(B2) algorithm, is given in a general form, because we also need such a formulation to address alternative algorithms.

Lemma 1. *Given a vector* $\mathbf{y} \in \{0,1\}^n$, *let* $\mathbf{e} \in \mathcal{E}$ *denote the noise vector such that* $\mathbf{y} \oplus \mathbf{e}$ *is a codeword. Fix any subsets* $\hat{\mathcal{E}}_1, \ldots, \hat{\mathcal{E}}_n$ *of the set* $\mathcal{E}$ *and define the decoding*

$$\hat{\varphi} = \Big\{ \hat{\varphi}_j(\mathbf{y}), \mathbf{y} \in \{0,1\}^n \Big\}_{j=1}^n$$

in such a way that

$$\hat{\varphi}_j(\mathbf{y}) = \begin{cases} 1 \oplus y_j, & \text{if } \mathbf{e} \in \hat{\mathcal{E}}_j, \\ y_j, & \text{if } \mathbf{e} \notin \hat{\mathcal{E}}_j \end{cases} \tag{3.5}$$

for all $j = \overline{1,n}$. *Then*

$$p - P_{\mathrm{b},j}(\hat{\varphi}) = \sum_{\mathbf{e} \in \hat{\mathcal{E}}_j} \Big[\omega_j^{(1 \oplus e_j)}(\mathbf{e}) - \omega_j^{(e_j)}(\mathbf{e}) \Big], \tag{3.6}$$

where

$$\omega_j^{(x)}(\mathbf{e}) \triangleq \sum_{\mathbf{u}:\, (\mathbf{uG})_j = x} W(\mathbf{uG} \oplus \mathbf{e} | \mathbf{0}), \quad x = 0, 1. \tag{3.7}$$

A possible decoding φ' for a BSC(p) is $\varphi'(\mathbf{y}) = \mathbf{y}$ for all $\mathbf{y} \in \{0,1\}^n$. Then $P_{\mathrm{b},j}(\varphi') = p$. Therefore we measure the gain in the performance for the decoding φ by the difference $p - P_{\mathrm{b},j}(\varphi)$.

Theorem 1. *If the decoding* φ *is defined by the* (B1)–(B2) *algorithm, then*

$$\begin{aligned} p - P_{\mathrm{b},j}(\varphi) \;\; &= \;\; \sum_{t=0}^{t_{\min}} \Theta_{j,t}(p) \max\Big\{ 0,\, p - (1 - \alpha_{j,t}(p)) \Big\} \\ &\quad + \sum_{t=1}^{t_{\min}} \Theta_{j,t-1}(p) \max\Big\{ 0,\, p - \alpha_{j,t-1}(p) \Big\} \end{aligned} \tag{3.8}$$

for all $j = \overline{1,n}$.

Proof : For all $j = \overline{1,n}$, let

$$\hat{\mathcal{E}}_j \triangleq \bigcup_{t=0}^{t_{\min}} \Big\{ \mathbf{e} \in \{0,1\}_t^n :\, e_j = 0 \,\Rightarrow\, p > 1 - \alpha_{j,t}(p);\; e_j = 1 \,\Rightarrow\, p > \alpha_{j,t-1}(p) \Big\}.$$

Then $\hat{\varphi} = \varphi$, as it follows from (B2) and (3.5). Therefore (3.6) can be expressed as

$$\begin{aligned} p - P_{\mathrm{b},j}(\varphi) \;\; &= \;\; \sum_{t=0}^{t_{\min}} \sum_{\mathbf{e} \in \{0,1\}_t^n : e_j = 0} \Big[\omega_j^{(1)}(\mathbf{e}) - \omega_j^{(0)}(\mathbf{e}) \Big] \chi\Big\{ p > 1 - \alpha_{j,t}(p) \Big\} \\ &\quad + \sum_{t=1}^{t_{\min}} \sum_{\mathbf{e} \in \{0,1\}_t^n : e_j = 1} \Big[\omega_j^{(0)}(\mathbf{e}) - \omega_j^{(1)}(\mathbf{e}) \Big] \chi\Big\{ p > \alpha_{j,t-1}(p) \Big\}. \end{aligned}$$

The proof is complete if we use the identities given in Proposition 1 below.

Proposition 1. *For all* $t = \overline{0, t_{\min}}$,

$$\frac{1}{\Theta_{j,t}(p)} \sum_{\mathbf{e} \in \{0,1\}_t^n : e_j = 0} \Big[\omega_j^{(1)}(\mathbf{e}) - \omega_j^{(0)}(\mathbf{e}) \Big] \;\; = \;\; p - (1 - \alpha_{j,t}(p)), \tag{3.9}$$

and, for all $t = \overline{1, t_{\min}}$,

$$\frac{1}{\Theta_{j,t-1}(p)} \sum_{\mathbf{e}\in\{0,1\}^n_t : e_j = 1} \Big[\omega_j^{(0)}(\mathbf{e}) - \omega_j^{(1)}(\mathbf{e}) \Big] \; = \; p - \alpha_{j,t-1}(p), \tag{3.10}$$

where the functions Θ *and* α *are defined in* (3.2)–(3.4).

□

We will discuss the performance of the (B1)–(B2) algorithm in comparison with the characteristics of the optimum decoding and the maximum likelihood decoding.

Theorem 2. *Denote*

$$\begin{aligned} \varphi^{(\mathrm{opt})} &= \Big\{ \varphi_j^{(\mathrm{opt})}(\mathbf{y}), \mathbf{y} \in \{0,1\}^n \Big\}_{j=1}^n, \\ \varphi^{(\mathrm{ML})} &= \Big\{ \varphi_j^{(\mathrm{ML})}(\mathbf{y}), \mathbf{y} \in \{0,1\}^n \Big\}_{j=1}^n, \end{aligned} \tag{3.11}$$

where

$$\left. \begin{array}{r} \mathbf{y} = \mathbf{uG} \oplus \mathbf{e} \\ \mathbf{e} \in \mathcal{E} \end{array} \right\} \implies \left\{ \begin{array}{l} \varphi_j^{(\mathrm{opt})}(\mathbf{y}) \triangleq y_j \oplus e_j \oplus \arg\max_{x=0,1} \omega_j^{(x)}(\mathbf{e}), \\ \varphi_j^{(\mathrm{ML})}(\mathbf{y}) \triangleq y_j \oplus e_j. \end{array} \right. \tag{3.12}$$

Then $\varphi^{(\mathrm{opt})}$ *is the optimum decoding in a sense that*

$$\min_{\varphi'} P_{\mathrm{b},j}(\varphi') = P_{\mathrm{b},j}(\varphi^{(\mathrm{opt})}), \tag{3.13}$$

where the minimization domain consists of all mappings $\varphi' : \{0,1\}^n \to \{0,1\}^n$. *Furthermore,*

$$p - P_{\mathrm{b},j}(\varphi^{(\mathrm{opt})}) = \sum_{\mathbf{e}\in\mathcal{E}} \max\Big\{ 0, \, \omega_j^{(1\oplus e_j)}(\mathbf{e}) - \omega_j^{(e_j)}(\mathbf{e}) \Big\}, \tag{3.14}$$

$$p - P_{\mathrm{b},j}(\varphi^{(\mathrm{ML})}) = \sum_{\mathbf{e}\in\mathcal{E}} \Big[\omega_j^{(1\oplus e_j)}(\mathbf{e}) - \omega_j^{(1)}(\mathbf{e}) \Big]. \tag{3.15}$$

To implement an optimal decoding procedure for minimizing the bit error probability in general case we should specify how the decoder has to process 2^{n-k} noise vectors $\mathbf{e} \in \mathcal{E}$, since the selection between $\omega_j^{(0)}(\mathbf{e})$ and $\omega_j^{(1)}(\mathbf{e})$ in (3.11) depends on $\mathbf{e}$. A simplification of this procedure can be obtained if we allow this selection to be dependent only on e_j and $\mathrm{wt}(\mathbf{e})$. If we also assume that the decoder is allowed to change the received word only when $\mathrm{wt}(\mathbf{e}) \le t_{\min}$, then we will also get a closed form expression for the j–th bit error probability since all noise vectors of the Hamming weights at most $t_{\min}$ belong to the set $\mathcal{E}$, and their contribution can be computed using the weight distributions of codewords, $\{M_{j,d}^{(0)}, M_{j,d}^{(1)}\}_{d=0}^n$. More specifically,

$$\begin{aligned} p - P_{\mathrm{b},j}(\varphi^{(\mathrm{opt})}) &\ge \sum_{\mathbf{e}\in\hat{\mathcal{E}}} \max\Big\{ 0, \, \omega_j^{(1\oplus e_j)}(\mathbf{e}) - \omega_j^{(e_j)}(\mathbf{e}) \Big\} \\ &= \sum_{t=0}^{t_{\min}} \sum_{\mathbf{e}\in\{0,1\}^n_t : e_j=0} \max\Big\{ 0, \, \omega_j^{(1)}(\mathbf{e}) - \omega_j^{(0)}(\mathbf{e}) \Big\} \\ &\quad + \sum_{t=1}^{t_{\min}} \sum_{\mathbf{e}\in\{0,1\}^n_t : e_j=1} \max\Big\{ 0, \, \omega_j^{(0)}(\mathbf{e}) - \omega_j^{(1)}(\mathbf{e}) \Big\} \end{aligned}$$

$$\geq \sum_{t=0}^{t_{\min}} \max\Big\{ 0, \sum_{\mathbf{e}\in\{0,1\}_t^n : e_j=0} \Big[\omega_j^{(1)}(\mathbf{e}) - \omega_j^{(0)}(\mathbf{e}) \Big] \Big\}$$

$$+ \sum_{t=1}^{t_{\min}} \max\Big\{ 0, \sum_{\mathbf{e}\in\{0,1\}_t^n : e_j=1} \Big[\omega_j^{(0)}(\mathbf{e}) - \omega_j^{(1)}(\mathbf{e}) \Big] \Big\}$$

$$= p - P_{\mathrm{b},j}(\varphi). \tag{3.16}$$

Thus, $p - P_{\mathrm{b},j}(\varphi)$ is a lower bound on $p - P_{\mathrm{b},j}(\varphi^{(\mathrm{opt})})$. This bound is tight if and only if both inequalities included in these transformations can be replaced with the equalities. Examples of codes, for which this is true for all $p \in (0, 1/2)$, include the $(6, 3)$ shortened Hamming code and the $(23, 12)$ Golay code. Let us explain the statement for the $(6, 3)$ code introduced by the generator matrix

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Denote $\mathbf{0} = 000000$, $\mathbf{1}_{1,6} = 100001$, and let $\mathbf{1}_j$ stand for the vector of length 6 having bit equal to 1 only in the j–th position, $j = \overline{1,6}$. The set of coset representatives of the smallest Hamming weights can be fixed as the set

$$\mathcal{E} = \{\mathbf{0}, \mathbf{1}_1, \mathbf{1}_2, \mathbf{1}_3, \mathbf{1}_4, \mathbf{1}_5, \mathbf{1}_6, \mathbf{1}_{1,6}\}.$$

The Hamming weights of the vectors $\mathbf{uG} \oplus \mathbf{e}$, where $\mathbf{u} \in \{0,1\}^3$ and $\mathbf{e} \in \mathcal{E}$, are given in Table 2. An analysis of these data leads to the following conclusions, where we assume that $q = 1 - p$.

1. For all $j = \overline{1,6}$,

$$\omega_j^{(0)}(\mathbf{0}) = p^0q^6 + 2p^3q^3 + p^4q^2, \;\; \omega_j^{(1)}(\mathbf{0}) = 2p^3q^3 + 2p^4q^2$$

and $\omega_j^{(0)}(\mathbf{0}) > \omega_j^{(1)}(\mathbf{0})$ for all $p \in (0, 1/2)$.

2. For all $j = \overline{1,6}$,

$$\omega_j^{(0)}(\mathbf{1}_j) = p^1q^5 + 2p^4q^2 + p^5q^1, \;\; \omega_j^{(1)}(\mathbf{1}_j) = 2p^2q^4 + 2p^3q^3,$$

and $\omega_j^{(0)}(\mathbf{1}_j) < \omega_j^{(1)}(\mathbf{1}_j)$ if and only if $p \in (0, \varepsilon_1)$, where $\varepsilon_1 = 0.2928\ldots$ is uniquely determined as the root to the equation

$$\varepsilon_1^1(1-\varepsilon_1)^5 + 2\varepsilon_1^4(1-\varepsilon_1)^2 + \varepsilon_1^5(1-\varepsilon_1)^1 = 2\varepsilon_1^2(1-\varepsilon_1)^4 + 2\varepsilon_1^3(1-\varepsilon_1)^3$$

belonging to the interval $(0, 1/2)$. If $j' \neq j$, then there can be two different functions $\omega_j^{(0)}(\mathbf{1}_{j'})$ and $\omega_j^{(1)}(\mathbf{1}_{j'})$. However, in both cases, $\omega_j^{(0)}(\mathbf{1}_{j'}) > \omega_j^{(1)}(\mathbf{1}_{j'})$ for all $p \in (0, 1/2)$.

3. For $j \in \{1, 6\}$,

$$\omega_j^{(0)}(\mathbf{1}_{1,6}) = p^2q^4 + 2p^3q^3 + p^6q^0, \;\; \omega_j^{(1)}(\mathbf{1}_{1,6}) = 2p^2q^4 + 2p^3q^3,$$

and for $j' \in \{2, 3, 4, 5\}$,

$$\omega_{j'}^{(0)}(\mathbf{1}_{1,6}) = 2p^2q^4 + 2p^3q^3, \;\; \omega_{j'}^{(1)}(\mathbf{1}_{1,6}) = p^2q^4 + 2p^3q^3 + p^6q^0.$$

Thus, $\omega_j^{(0)}(\mathbf{1}_{1,6}) < \omega_j^{(1)}(\mathbf{1}_{1,6})$ and $\omega_{j'}^{(0)}(\mathbf{1}_{1,6}) > \omega_{j'}^{(1)}(\mathbf{1}_{1,6})$ for all $p \in (0, 1/2)$.

Inequalities above show that $\omega_j^{(1\oplus e_j)}(\mathbf{e})$ *can be greater than* $\omega_j^{(e_j)}(\mathbf{e})$ *only if* $\mathbf{e} = \mathbf{1}_j$. Therefore, by (3.14), (3.15),

$$\begin{aligned} p - P_{\mathrm{b},j}(\varphi^{(\mathrm{opt})}) &= \max\Big\{ 0,\ (p^1q^5 + 2p^4q^2 + p^5q^1) - (2p^2q^4 + 2p^3q^3) \Big\}, \quad (3.17) \\ p - P_{\mathrm{b},j}(\varphi^{(\mathrm{ML})}) &= (p^1q^5 + 2p^4q^2 + p^5q^1) - (2p^2q^4 + 2p^3q^3) + (p^6q^0 - p^2q^4)\chi\{\, j \in \{1,6\}\,\} \end{aligned}$$

for all $j = \overline{1,6}$. On the other hand, the expression on the right–hand side of (3.17) is equal to $p - P_{\mathrm{b},j}(\varphi)$, and it can be obtained using (3.8) and the weight distributions of codewords (see Table 3). Thus, we can find the value of ε_1 and derive the decoding φ without inspection of the standard array of the space constructed for a given code. This inspection leads to a stronger statement that such an algorithm attains the optimum performance for all $p \in (0, 1/2)$.

The results above are illustrated in Figure 1.

4 Decoding algorithm for arbitrary binary–input symmetric memoryless channels

Let us extend considerations to the case when the channel has the output alphabet $\mathcal{Y} \triangleq \{0, \ldots, 2L-1\}$. Let the conditional probabilities $W_x(y), x \in \{0,1\}, y \in \mathcal{Y}$, be defined by the $2 \times (2L)$ matrix

$$\mathbf{W} = \begin{bmatrix} q(0) & \ldots & q(L-1) & p(L-1) & \ldots & p(0) \\ p(0) & \ldots & p(L-1) & q(L-1) & \ldots & q(0) \end{bmatrix},$$

where

$$q(y) > p(y) \geq 0,\ y = \overline{0, L-1};\quad \sum_{y=0}^{L-1} (p(y) + q(y)) = 1. \quad (4.1)$$

We will also assume that

$$|\mathcal{P}| = L, \quad (4.2)$$

where

$$\mathcal{P} \triangleq \bigcup_{y=0}^{L-1} \Big\{ \frac{p(y)}{p(y) + q(y)} \Big\}, \quad (4.3)$$

and denote

$$p \triangleq \sum_{y=0}^{L-1} p(y). \quad (4.4)$$

Channels having these properties will be referred to as *binary–input symmetric memoryless channels*. If $L = 1$, then the matrix $\mathbf{W}$ determines a BSC(p).

The decoding for arbitrary binary–input memoryless channels is the mapping

$$\varphi = \Big\{ \varphi_j(\mathbf{y}) \in \{0,1\}, \mathbf{y} \in \mathcal{Y}^n \Big\}_{j=1}^{n},$$

and the definition of the j–th bit error probability is the same as (3.1).

We will translate data transmission over a channel determined by the matrix $\mathbf{W}$ to the transmission over a time–varying binary symmetric channel and introduce the following notation.

(a) For all $y \in \{0, \dots, 2L-1\}$, let

$$z(y) \triangleq \begin{cases} 0, & \text{if } y \in \{0, \dots, L-1\}, \\ 1, & \text{if } y \in \{L, \dots, 2L-1\}, \end{cases} \tag{4.5}$$

$$\xi(y) \triangleq \begin{cases} p(y)/(p(y)+q(y)), & \text{if } y \in \{0, \dots, L-1\}, \\ p(y')/(p(y')+q(y')), & \text{if } y \in \{L, \dots, 2L-1\}, \end{cases} \tag{4.6}$$

where $y' = 2L-1-y$. For all $\mathbf{y} \in \mathcal{Y}^n$, let

$$\mathbf{z}(\mathbf{y}) \triangleq (z(y_1), \dots, z(y_n)), \quad \boldsymbol{\xi}(\mathbf{y}) \triangleq (\xi(y_1), \dots, \xi(y_n)). \tag{4.7}$$

(b) For all $y \in \{0, \dots, 2L-1\}$, let

$$\pi(\xi)\Big|_{\xi=\xi(y)} \triangleq p(y) + q(y). \tag{4.8}$$

(c) For all $\xi \in \mathcal{P}$ and $x \in \{0,1\}$, let

$$V_\xi(x) \triangleq |(x \oplus 1) - \xi| = \begin{cases} 1-\xi, & \text{if } x = 0, \\ \xi, & \text{if } x = 1, \end{cases} \tag{4.9}$$

where the set $\mathcal{P}$ is defined in (4.3).

(d) For all $\boldsymbol{\xi} \in \mathcal{P}^n$ and $\mathbf{x} \in \{0,1\}^n$, let

$$\Pi(\boldsymbol{\xi}) \triangleq \prod_{j=1}^{n} \pi(\xi_j), \quad V(\mathbf{x}|\boldsymbol{\xi}) \triangleq \prod_{j=1}^{n} V_{\xi_j}(x_j). \tag{4.10}$$

By (4.1)–(4.3) and (4.8), we conclude that $(\,\pi(\xi), \xi \in \mathcal{P}\,)$ is the probability distribution on the set $\mathcal{P}$. Furthermore, by (4.9), $(\,V_\xi(x), x \in \{0,1\}\,)$ is the conditional probability distribution on the set $\{0,1\}$ given $\xi \in \mathcal{P}$ and

$$V_\xi(x) = V_{1-\xi}(1 \oplus x). \tag{4.11}$$

Example. *Let* $L = 2$ *and*

$$\mathbf{W} = \begin{bmatrix} 0.70 & 0.20 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.20 & 0.70 \end{bmatrix}. \tag{4.12}$$

Then $(z(0), \dots, z(3)) = (0,0,1,1)$, $(\xi(0), \dots, \xi(3)) = (1/15, 1/5, 1/5, 1/15)$, $\mathcal{P} = \{1/15, 1/5\}$, $(\pi(1/15), \pi(1/5)) = (3/4, 1/4)$.

Remark 1. Notations above can be also used when $L = 1$. Then $(z(y), \xi(y)) = (y, p)$ and $(\mathbf{z}(\mathbf{y}), \boldsymbol{\xi}(\mathbf{y})) = (\mathbf{y}, \mathbf{p})$ for all $\mathbf{y} \in \{0,1\}^n$, where $\mathbf{p}$ is the vector of length n whose components are equal to p. Furthermore,

$$\Pi(\boldsymbol{\xi})\Big|_{\boldsymbol{\xi}=\mathbf{p}} = 1, \quad V(\mathbf{x}|\boldsymbol{\xi})\Big|_{\boldsymbol{\xi}=\mathbf{p}} = W(\mathbf{x}|\mathbf{0}) \tag{4.13}$$

for all $\mathbf{x} \in \{0,1\}^n$.

Our considerations of arbitrary symmetric channels are based on the following observations that can be explained for the channel defined in (4.12). Let us introduce the data transmission scheme given in Figure 3, where the j–th code bit is transmitted either over a BSC(1/15), or over a BSC(1/5). The selection is made by a jammer who chooses the first channel with probability 3/4 and chooses the second channel with probability 1/4. The jammer also informs the decoder about his choice by sending him the value of $\xi_j \in \{1/15, 1/5\}$. These operations are independently repeated for all $j = \overline{1,n}$. The decoder has to construct an estimate of the transmitted codeword for a given vector of pairs $((z_1,\xi_1),\ldots,(z_n,\xi_n))$. It is easy to see that this problem coincides with the original problem of estimating the transmitted codeword for a given vector $(y_1,\ldots,y_n)$ received at the output of the channel having the matrix of crossover probabilities (4.12). This is true because there are component–wise one–to–one mappings

$$0 \leftrightarrow (0,1/15), \quad 1 \leftrightarrow (0,1/5), \quad 2 \leftrightarrow (1,1/5), \quad 3 \leftrightarrow (1,1/15)$$

that translate $(y_1,\ldots,y_n)$ to the vector of pairs generated in the scheme in Figure 3. Nevertheless, such a translation to data transmission over a time–varying binary symmetric channel makes sense from technical point of view. Namely, we will fix a vector $\boldsymbol{\xi} \in \mathcal{P}^n$ and compute the j–th bit error probability when the 1–st code bit is transmitted over a BSC(ξ_1), …, the n–th code bit is transmitted over a BSC(ξ_n). The final result is obtained as a weighted sum of the results for all possible vectors $\boldsymbol{\xi}$, where the weight is defined as $\Pi(\boldsymbol{\xi})$. Notice also that

$$\sum_{\xi\in\mathcal{P}} \pi(\xi)\xi = p, \tag{4.14}$$

where p is defined in (4.4). It is easy to see that transformation of the received vector $\mathbf{y}$ to the vector $\mathbf{z}(\mathbf{y})$ defined in (4.6) corresponds to quantization of components of the vector $\mathbf{y}$. As a result, we "create" a BSC(p). Thus, the j–th bit error probability, which is attained using the optimal decoding rule, does not exceed the j–th bit error probability for a BSC(p), and we can measure the gain achievable with the decoding φ by the difference $p - P_{\mathrm{b},j}(\varphi)$.

Remark 2. By the assumption (4.2), the total number of pairs of vectors $(\mathbf{z},\boldsymbol{\xi}) \in \{0,1\}^n \times \mathcal{P}^n$ coincides with the total number of vectors $\mathbf{y} \in \mathcal{Y}^n$, since

$$2^n |\,\mathcal{P}^n\,| = (2L)^n,$$

and for any pair $(\mathbf{z},\boldsymbol{\xi})$ there is a unique $\mathbf{y} \in \mathcal{Y}^n$ such that $\mathbf{z}(\mathbf{y}) = \mathbf{z}$ and $\boldsymbol{\xi}(\mathbf{y}) = \boldsymbol{\xi}$, where the vectors $\mathbf{z}(\mathbf{y})$ and $\boldsymbol{\xi}(\mathbf{y})$ are defined in (4.5)–(4.7). We will write

$$\mathbf{y} = \mathbf{y}(\mathbf{z},\boldsymbol{\xi}) \iff \begin{cases} \mathbf{z}(\mathbf{y}) = \mathbf{z} \\ \boldsymbol{\xi}(\mathbf{y}) = \boldsymbol{\xi} \end{cases} \tag{4.15}$$

and address each vector of the set $\mathcal{Y}^n$ by corresponding pair of vectors $(\mathbf{z},\boldsymbol{\xi}) \in \{0,1\}^n \times \mathcal{P}^n$.

Let us introduce the following decoding algorithm.

(L0) *Given a received vector* $\mathbf{y} \in \mathcal{Y}^n$, *form the vectors* $\mathbf{z} = \mathbf{z}(\mathbf{y})$ *and* $\boldsymbol{\xi} = \boldsymbol{\xi}(\mathbf{y})$ *according to* (4.5)–(4.7) *and represent* $\mathbf{y}$ *as the vector* $\mathbf{y}(\mathbf{z},\boldsymbol{\xi})$.

(L1) *Find, if possible, the noise vector* $\mathbf{e} \in \{0,1\}^n$ *of the Hamming weight at most* $t_{\min}$ *such that* $\mathbf{z} \oplus \mathbf{e}$ *is a codeword.*

(L2a) *The noise vector* $\mathbf{e}$ *is found in step* (L1) : *denote* $t = \mathrm{wt}(\mathbf{e})$ *and set*

$$e_j = 0 \implies \varphi_j(\mathbf{y}(\mathbf{z},\boldsymbol{\xi})) = \begin{cases} 1 \oplus z_j, & \text{if } \xi_j > 1 - \alpha_{j,t}(p), \\ z_j, & \text{otherwise} \end{cases}$$

$$e_j = 1 \implies \varphi_j(\mathbf{y}(\mathbf{z},\boldsymbol{\xi})) = \begin{cases} 1 \oplus z_j, & \text{if } \xi_j > \alpha_{j,t-1}(p), \\ z_j, & \text{otherwise} \end{cases}$$

for all $j = \overline{1,n}$, *where* p *is defined in* (4.4).

(L2b) *The noise vector* $\mathbf{e}$ *is not found in step* (L1) : *set* $\varphi(\mathbf{y}(\mathbf{z},\boldsymbol{\xi})) = \mathbf{z}$.

The relationship between the decoding φ for arbitrary binary–input symmetric memoryless channels and the decoding for binary symmetric channels can be established on the basis of Proposition 2 and Lemma 2 given below.

Proposition 2. *For any pair of vectors* $(\mathbf{x},\mathbf{y}) \in \{0,1\}^n \times \mathcal{Y}^n$,

$$\left.\begin{array}{c} \mathbf{z}(\mathbf{y}) = \mathbf{z} \\ \boldsymbol{\xi}(\mathbf{y}) = \boldsymbol{\xi} \end{array}\right\} \implies W(\mathbf{y}|\mathbf{x}) = \Pi(\boldsymbol{\xi}) V(\mathbf{x} \oplus \mathbf{z}|\boldsymbol{\xi}), \tag{4.16}$$

where the vectors $\mathbf{z}(\mathbf{y}) \in \{0,1\}^n$ *and* $\boldsymbol{\xi}(\mathbf{y}) \in \mathcal{P}^n$ *are defined in* (4.5)–(4.7).

The result of Proposition 2 can be interpreted in such a way that the conditional probability distribution V plays the role of the conditional probability distribution W for a time–varying BSC whose crossover probabilities at each time instant are determined by components of the vector $\boldsymbol{\xi} \in \mathcal{P}^n$.

Lemma 2. *Given a vector* $\boldsymbol{\xi} \in \mathcal{P}^n$ *and a vector* $\mathbf{y} \in \mathcal{Y}^n$ *with* $(\mathbf{z}(\mathbf{y}), \boldsymbol{\xi}(\mathbf{y})) = (\mathbf{z}, \boldsymbol{\xi})$, *let* $\mathbf{e} \in \mathcal{E}$ *denote the noise vector such that* $\mathbf{z} \oplus \mathbf{e}$ *is a codeword. Fix any subsets* $\hat{\mathcal{E}}_1(\boldsymbol{\xi}), \ldots, \hat{\mathcal{E}}_n(\boldsymbol{\xi})$ *of the set* $\mathcal{E}$ *and define the decoding* $\hat{\varphi}$ *in such a way that*

$$\hat{\varphi}_j(\mathbf{y}(\mathbf{z},\boldsymbol{\xi})) = \begin{cases} 1 \oplus z_j, & \text{if } \mathbf{e} \in \hat{\mathcal{E}}_j(\boldsymbol{\xi}), \\ z_j, & \text{if } \mathbf{e} \notin \hat{\mathcal{E}}_j(\boldsymbol{\xi}) \end{cases}$$

for all $j = \overline{1,n}$. *Then*

$$p - P_{\mathrm{b},j}(\hat{\varphi}) = \sum_{\boldsymbol{\xi} \in \mathcal{P}^n} \Pi(\boldsymbol{\xi}) \sum_{\mathbf{e} \in \hat{\mathcal{E}}_j(\boldsymbol{\xi})} \left[v_j^{(1 \oplus e_j)}(\mathbf{e}|\boldsymbol{\xi}) - v_j^{(e_j)}(\mathbf{e}|\boldsymbol{\xi}) \right], \tag{4.17}$$

where

$$v_j^{(x)}(\mathbf{e}|\boldsymbol{\xi}) \triangleq \sum_{\mathbf{u}:\, (\mathbf{uG})_j = x} V(\mathbf{uG} \oplus \mathbf{e}|\boldsymbol{\xi}), \quad x = 0, 1. \tag{4.18}$$

In the proof of Lemma 2 we use the following transformations :

$$\mathbf{y} \in \mathcal{Y}^n \longrightarrow (\mathbf{z}, \boldsymbol{\xi}) \longrightarrow (\mathbf{u}, \mathbf{e}, \boldsymbol{\xi}) \in \{0,1\}^n \times \{0,1\}^n \times \mathcal{P}^n \tag{4.19}$$

and

$$W(\mathbf{y}|\mathbf{x}) \longrightarrow \Pi(\boldsymbol{\xi}) V(\mathbf{x} \oplus \mathbf{z}|\boldsymbol{\xi}) \longrightarrow \Pi(\boldsymbol{\xi}) V(\mathbf{x} \oplus \mathbf{uG} \oplus \mathbf{e}|\boldsymbol{\xi}). \tag{4.20}$$

Notice that if $L = 1$, then

$$v_j^{(x)}(\mathbf{e}|\boldsymbol{\xi})\Big|_{\boldsymbol{\xi} = \mathbf{p}} = \omega_j^{(x)}(\mathbf{e}), \quad x = 0, 1,$$

which agrees with (4.13).

Theorem 3. *If the decoding φ is defined by the* (L0)–(L2) *algorithm, then*

$$
\begin{aligned}
p - P_{\mathrm{b},j}(\varphi) \;=\; & \sum_{t=0}^{t_{\min}} \Theta_{j,t}(p) \sum_{\xi\in\mathcal{P}} \pi(\xi) \max\Big\{\, 0,\ \xi - (1-\alpha_{j,t}(p)) \,\Big\} \\
& + \sum_{t=1}^{t_{\min}} \Theta_{j,t-1}(p) \sum_{\xi\in\mathcal{P}} \pi(\xi) \max\Big\{\, 0,\ \xi - \alpha_{j,t-1}(p) \,\Big\} \qquad (4.21)
\end{aligned}
$$

for all $j = \overline{1,n}$.

The performance of our decoding can be compared with the performance of the optimum decoding.

Theorem 4. *For all $(\mathbf{z},\boldsymbol{\xi}) \in \{0,1\}^n \times \mathcal{P}^n$, set*

$$
\left.\begin{array}{r} \mathbf{z} = \mathbf{uG} \oplus \mathbf{e} \\ \mathbf{e} \in \mathcal{E} \end{array}\right\} \;\Longrightarrow\; \varphi_j^{(\mathrm{opt})}(\mathbf{y}(\mathbf{z},\boldsymbol{\xi})) \triangleq z_j \oplus e_j \oplus \arg\max_{x=0,1} \upsilon_j^{(x)}(\mathbf{e}|\boldsymbol{\xi}). \qquad (4.22)
$$

Then

$$
\varphi^{(\mathrm{opt})} = \Big\{\, \varphi_j^{(\mathrm{opt})}(\mathbf{y}), \mathbf{y} \in \mathcal{Y}^n \,\Big\}_{j=1}^{n}
$$

is the optimum decoding in a sense that (3.13) *holds when the minimization domain consists of all mappings $\varphi' : \mathcal{Y}^n \to \{0,1\}^n$. Furthermore,*

$$
p - P_{\mathrm{b},j}(\varphi^{(\mathrm{opt})}) \;=\; \sum_{\mathbf{e}\in\mathcal{E}} \sum_{\boldsymbol{\xi}\in\mathcal{P}^n} \Pi(\boldsymbol{\xi}) \max\Big\{\, 0,\ \upsilon_j^{(1\oplus e_j)}(\mathbf{e}|\boldsymbol{\xi}) - \upsilon_j^{(e_j)}(\mathbf{e}|\boldsymbol{\xi}) \,\Big\}. \qquad (4.23)
$$

We do not include the proof of Theorem 4 in the Appendix, since it can be obtained as a direct extension of the proof of Theorem 2.

Similarly to transformations (3.16) we can write

$$
\begin{aligned}
p - P_{\mathrm{b},j}(\varphi^{(\mathrm{opt})}) \;\geq\; & \sum_{t=0}^{t_{\min}} \sum_{\boldsymbol{\xi}\in\mathcal{P}^n} \Pi(\boldsymbol{\xi}) \max\Big\{\, 0,\ \sum_{\mathbf{e}\in\{0,1\}_t^n : e_j=0} \Big[\, \upsilon_j^{(1)}(\mathbf{e}|\boldsymbol{\xi}) - \upsilon_j^{(0)}(\mathbf{e}|\boldsymbol{\xi}) \,\Big] \Big\} \\
& + \sum_{t=1}^{t_{\min}} \sum_{\boldsymbol{\xi}\in\mathcal{P}^n} \Pi(\boldsymbol{\xi}) \max\Big\{\, 0,\ \sum_{\mathbf{e}\in\{0,1\}_t^n : e_j=1} \Big[\, \upsilon_j^{(0)}(\mathbf{e}|\boldsymbol{\xi}) - \upsilon_j^{(1)}(\mathbf{e}|\boldsymbol{\xi}) \,\Big] \Big\}
\end{aligned}
$$

and continue this inequality as follows :

$$
\begin{aligned}
p - P_{\mathrm{b},j}(\varphi^{(\mathrm{opt})}) \;\geq\; & \sum_{t=0}^{t_{\min}} \sum_{\xi_j\in\mathcal{P}} \pi(\xi_j) \max\Big\{\, 0,\ \sum_{\xi_i\in\mathcal{P},\, i\neq j} \frac{\Pi(\boldsymbol{\xi})}{\pi(\xi_j)} \sum_{\mathbf{e}\in\{0,1\}_t^n : e_j=0} \Big[\, \upsilon_j^{(1)}(\mathbf{e}|\boldsymbol{\xi}) - \upsilon_j^{(0)}(\mathbf{e}|\boldsymbol{\xi}) \,\Big] \Big\} \\
& + \sum_{t=1}^{t_{\min}} \sum_{\xi_j\in\mathcal{P}} \pi(\xi_j) \max\Big\{\, 0,\ \sum_{\xi_i\in\mathcal{P},\, i\neq j} \frac{\Pi(\boldsymbol{\xi})}{\pi(\xi_j)} \sum_{\mathbf{e}\in\{0,1\}_t^n : e_j=1} \Big[\, \upsilon_j^{(0)}(\mathbf{e}|\boldsymbol{\xi}) - \upsilon_j^{(1)}(\mathbf{e}|\boldsymbol{\xi}) \,\Big] \Big\} \\
=\; & p - P_{\mathrm{b},j}(\varphi).
\end{aligned}
$$

The last inequality has a bigger influence on the final result than the previous ones. In particular, we already cannot claim the optimality of the algorithm φ for the $(6,3)$ code and for the $(23,12)$ code. Moreover, if the channel is obtained by symmetric quantization of the output of the AWGN channel in $2L$ levels and $L > 1$, then the algorithm is non–optimum for any signal–to–noise ratio. However, we reach an essential simplification of the optimum decoding procedure by allowing the selection between $v_j^{(0)}(\mathbf{e}|\boldsymbol{\xi})$ and $v_j^{(1)}(\mathbf{e}|\boldsymbol{\xi})$ in (4.22) to be dependent only on ξ_j, e_j, and $\mathrm{wt}(\mathbf{e})$.

Let us generalize considerations above to data transmission over an AWGN channel where 0's and 1's are transmitted by signals having the amplitudes $-\sqrt{E}$ and $+\sqrt{E}$, respectively, when the variance of the noise is equal to $\sigma^2 = 1/2$. Namely, let

$$W_x(y) \;=\; \frac{1}{\sqrt{\pi}} \begin{cases} e^{-(y+\sqrt{E})^2}, & \text{if } x = 0, \\ e^{-(y-\sqrt{E})^2}, & \text{if } x = 1 \end{cases}$$

be the conditional probability of receiving a symbol $y \in (-\infty, +\infty)$ when $x \in \{0, 1\}$ was transmitted. If we assume that amplitude of the signal at the output of the demodulator is symmetrically quantized in $2L$ levels and $L \to \infty$, then notations (4.5), (4.6) become

$$z(y) \;=\; \begin{cases} 0, & \text{if } y \le 0, \\ 1, & \text{if } y > 0, \end{cases} \tag{4.24}$$

$$\xi(y) \;=\; \frac{1}{2\sqrt{\pi}W(y)} \begin{cases} e^{-(y-\sqrt{E})^2}, & \text{if } y \le 0, \\ e^{-(y+\sqrt{E})^2}, & \text{if } y > 0, \end{cases} \tag{4.25}$$

for all y, where

$$W(y) \triangleq \frac{1}{2\sqrt{\pi}} \Big[e^{-(y+\sqrt{E})^2} + e^{-(y-\sqrt{E})^2} \Big]. \tag{4.26}$$

Furthermore, the set $\mathcal{P}$ becomes the interval $[0, 1/2]$, and (4.8) is represented as

$$\pi(\xi(y)) = 2W(y), \quad \text{for all } y \in (-\infty, +\infty). \tag{4.27}$$

Notice also that, for any function $f(\xi)$ defined on the interval $[0, 1/2]$,

$$\int_0^{1/2} \pi(\xi) f(\xi) d\xi = 2 \int_{-\infty}^{0} W(y) f(\xi(y)) dy = \int_{-\infty}^{+\infty} W(y) f(\xi(y)) dy, \tag{4.28}$$

because $\xi(-y) = \xi(y)$ for all y.

Let us introduce the function

$$Q(c) \triangleq \frac{1}{\sqrt{\pi}} \int_0^{c+\sqrt{E}} e^{-y^2} dy \tag{4.29}$$

and notice that if the demodulator would make a hard decision about the transmitted bit at each time instant, then we obtain a binary symmetric channel having the crossover probability

$$p = \frac{1}{2} - Q(0). \tag{4.30}$$

Theorem 5. *For all $j = \overline{1,n}$ and $t = \overline{0, t_{\min}}$, let*

$$y_{j,t}(p) \triangleq \frac{1}{4\sqrt{E}} \Big| \ln \frac{\alpha_{j,t}(p)}{1 - \alpha_{j,t}(p)} \Big|. \tag{4.31}$$

If the decoding φ is specified by the (L0)–(L2) *algorithm with the vectors* $\mathbf{z}(\mathbf{y})$ *and* $\boldsymbol{\xi}(\mathbf{y})$ *whose components are defined in* (4.24), (4.25), *then*

$$\begin{aligned} p - P_{\mathrm{b},j}(\varphi) &= \sum_{t=0}^{t_{\min}} \Theta_{j,t}(p) \int_{-\infty}^{+\infty} W(y) \max\Big\{\, 0,\ \xi(y) - (1-\alpha_{j,t}(p))\,\Big\} dy \\ &\quad + \sum_{t=1}^{t_{\min}} \Theta_{j,t-1}(p) \int_{-\infty}^{+\infty} W(y) \max\Big\{\, 0,\ \xi(y) - \alpha_{j,t-1}(p)\,\Big\} dy \qquad (4.32) \\ &= \sum_{t=0}^{t_{\min}-1} \Theta_{j,t}(p) I_{j,t}(p) + \Theta_{j,t_{\min}}(p) \max\Big\{\, 0,\ I^{(0)}_{j,t_{\min}}(p)\,\Big\}, \qquad (4.33) \end{aligned}$$

where

$$I_{j,t}(p) \triangleq \begin{cases} I^{(0)}_{j,t}(p), & \text{if } \alpha_{j,t}(p) > 1/2, \\ 0, & \text{if } \alpha_{j,t}(p) = 1/2, \\ I^{(1)}_{j,t}(p), & \text{if } \alpha_{j,t}(p) < 1/2 \end{cases} \qquad (4.34)$$

and

$$\begin{aligned} I^{(0)}_{j,t}(p) &\triangleq \alpha_{j,t}(p) Q\Big(+y_{j,t}(p)\Big) + (1-\alpha_{j,t}(p)) Q\Big(-y_{j,t}(p)\Big) - Q(0), \\ I^{(1)}_{j,t}(p) &\triangleq (1-\alpha_{j,t}(p)) Q\Big(+y_{j,t}(p)\Big) + \alpha_{j,t}(p) Q\Big(-y_{j,t}(p)\Big) - Q(0). \end{aligned}$$

An example of the function $\xi(y)$ defined in (4.25), is illustrated in Figure 4, where we also show the values of $\alpha_{j,0}(p), \ldots, \alpha_{j,3}(p)$ for the $(23,12)$. code. As it is easy to see from (4.32), the values of y such that $\xi(y) > \alpha_{j,t-1}(p)$, give a positive contribution to the difference $p - P_{\mathrm{b},j}(\varphi)$. In general case, as it follows from (4.33), (4.34), $p - P_{\mathrm{b},j}(\varphi) = 0$ if and only if

$$\alpha_{j,t}(p) = 1/2, \quad t = \overline{0, t_{\min}-1} \qquad (4.35)$$

and $\alpha_{j,t_{\min}}(p) \geq 1/2$. Suppose that $d_{\min} \geq 3$. Then $t_{\min} \geq 1$ and, by (3.2), (3.3) we conclude that (4.35) is equivalent to the equalities

$$\sum_{d=0}^{n} M^{(0)}_{j,d} \Gamma^{(d)}_t(p) = \sum_{d=1}^{n} M^{(1)}_{j,d} \Gamma^{(d-1)}_t(p) \qquad (4.36)$$

for all $t = \overline{0, t_{\min}-1}$. Equality in (4.36) can take place for at most n values of $p \in (0, 1/2)$, since the both the left–hand side and the right–hand side contain polynomials of p, having the degrees at most n. Note also that the our requirement is stronger : the value of p should satisfy (4.36) for all $t = \overline{0, t_{\min}-1}$ and that p obtained from (4.30), is an irrational number. We do not know examples of codes such that these constraints are satisfy, but present the statement in the following form : *for any binary linear code having the minimal distance at least 3, there are at most n signal–to–noise ratios such that* $p - P_{\mathrm{b},j}(\varphi) = 0$.

Let us also note that our analysis leads to some conclusions for the BCJR algorithm for AWGN channels. Obviously, there are received vectors $\mathbf{y}$ such that the vector $\mathbf{e} \in \mathcal{E} \backslash \{\mathbf{0}\}$ is the result of their quantization and the probability $W(\mathbf{y}|\mathbf{0})$ is much greater than $W(\mathbf{y}|\mathbf{x})$, where $\mathbf{x}$ is any non–zero codeword. Then the vector $\mathbf{e}$ is also the result of decoding formed by the BCJR algorithm. Therefore an additional condition that there are at most n "bad" signal–to–noise ratios can be omitted for the optimum decoding. This fact can be also easily found from the analysis of the formula

$$
\begin{aligned}
& p - P_{\mathrm{b},j}(\varphi^{(\mathrm{opt})}) \\
= \; & \sum_{\mathbf{e}\in\mathcal{E}} \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \Big[\prod_{t=1}^{n} \pi(\xi(y_t)) \Big] \max\Big\{ 0, \; \upsilon_j^{(1\oplus e_j)}(\mathbf{e}|\boldsymbol{\xi}(\mathbf{y})) - \upsilon_j^{(e_j)}(\mathbf{e}|\boldsymbol{\xi}(\mathbf{y})) \Big\} dy_1 \ldots y_n,
\end{aligned}
$$

which can be derived as a direct generalization of (4.23). This expression gives the value of the probability $P_{\mathrm{b},j}(\varphi^{(\mathrm{opt})})$.

5 Conclusion

Designers of systems of information transmission with the decoder for minimizing the bit error probabilities mainly concern about the performance of the decoding algorithm and the complexity of necessary transformations. Due to our knowledge, the only available estimate of the performance of the BCJR algorithm requires the weight distributions of vectors belonging to all cosets constructed for a given code via "the standard array representation" of the space $\{0,1\}^n$ (see [6] for a good summary of known results). It is also not clear which parameters of the code are important for effective solving the problem. In particular, the minimum distance and the number non–zero codewords having the smallest weight specify only one term in the sums to be compared (these sums were mentioned in the Introduction).

Certainly, if the signal–to–noise ratio is big, then one can say that we run the maximum likelihood decoding and also obtain good results with the bit error probabilities criterion. However, the problem under considerations is different : it has an identification spirit [7] and requires different approaches. We illustrate this point by paying attention to the fact that the difference in the results of the optimum algorithm and the algorithm, which uses a bounded distance decoding procedure, is much smaller than similar characteristics of decoding for minimizing the word error probability. Furthermore, we can get the exact expression for the performance of the algorithm if the weight distributions of codewords are known contrary to the fact that these distributions (the code spectrum) are used as data needed to construct union–type upper bounds on the decoding error probability when the decoder has to output an estimate of the transmitted message.

If the signal–to–noise ratio is small and the rate of the code is greater than the capacity of the channel, then one cannot obtain good results with the maximum likelihood decoding (we refer to the strong converse to the coding theorem [8]). In addition to the discussion above, notice that we use the weight distributions of codewords to obtain improvement over the trivial decoding for all $p < 0.292$ for the $(6,3)$ code, $p < 0.278$ for the $(23,12)$ code, etc., – these values are much greater than the values of crossover probabilities such that the code spectrum is useful in the analysis of the performance of the maximum likelihood decoding. The data transmission systems where small signal–to–noise ratios are of interest include iterative decoding algorithms : the decoding is implemented to subcodes of a very long codes with the hope that relibilities of components of the received word will be increasing during iterations. We do not address these aspects in the current paper because they require the description of a decoding algorithm and its analysis for channels with memory. However we believe that our approach where the influence of all received components, except for the component we are interested in, is accumulated in one number (the smallest weight of possible noise vectors) can be developed for this purpose.

We do not discuss the complexity of the bounded distance decoding included in our algorithms and its comparison with the complexity of the BCJR algorithm. Let us however mention another interesting possibility. Our approach can be also used if we run the (B1)–(B2) algorithm and the (L0)–(L2) algorithm with any value $t'_{\min} < t_{\min}$. In particular, we can organize the decoding procedure in such a way that it relies on the bounded distance decoding within half of the constructive distance when a BCH code is used. As a result, having received a binary vector, the decoder finds the closest codeword, if possible, by the Berlekamp–Massey algorithm. This approach leads to an essential simplification of the decoding and simultaneously brings a good performance of the decoding for minimizing the bit error probabilities in some region of signal–to–noise ratios.

We believe that the translation of the decoding problem for arbitrary binary–input symmetric channels to the problem for time–varying binary symmetric channels is important. The word received at the output of the channels having binary output alphabet belongs to the same space as the code. Therefore one can use the "standard array representation" of the space and other algebraic tools for the decoding. The translation makes these tools also useful for arbitrary binary–input symmetric channels.

Appendix

A.1 Proof of Lemma 1

Using (3.1) with $\hat{\varphi}$ substituted for φ, the representation of the received vector $\mathbf{y}$ as $\mathbf{uG} \oplus \mathbf{e}$, where $\mathbf{e} \in \mathcal{E}$, and the definitions of $\omega_j^{(0)}(\mathbf{e}), \omega_j^{(1)}(\mathbf{e})$ in (3.7) we obtain

$$\begin{aligned}
P_{\mathrm{b},j}(\hat{\varphi}) &= \frac{1}{2^k} \sum_{\mathbf{y}} \sum_{\mathbf{u}^*} W(\mathbf{y}|\mathbf{u}^*\mathbf{G}) \chi\{\, \hat{\varphi}_j(\mathbf{y}) \neq (\mathbf{u}^*\mathbf{G})_j \,\} \\
&= \frac{1}{2^k} \sum_{\mathbf{e}\in\mathcal{E}} \sum_{\mathbf{u},\mathbf{u}^*} W(\mathbf{uG}\oplus\mathbf{e}|\mathbf{u}^*\mathbf{G}) \chi\{\, \hat{\varphi}_j(\mathbf{uG}\oplus\mathbf{e}) \neq (\mathbf{u}^*\mathbf{G})_j \,\} \\
&= \frac{1}{2^k} \sum_{\mathbf{e}\in\mathcal{E}} \sum_{\mathbf{u}} \Big[\chi\{\, \hat{\varphi}_j(\mathbf{uG}\oplus\mathbf{e}) = 0 \,\} \sum_{\mathbf{u}^*:\,(\mathbf{u}^*\mathbf{G})_j=1} W(\mathbf{uG}\oplus\mathbf{e}|\mathbf{u}^*\mathbf{G}) \\
&\qquad + \chi\{\, \hat{\varphi}_j(\mathbf{uG}\oplus\mathbf{e}) = 1 \,\} \sum_{\mathbf{u}^*:\,(\mathbf{u}^*\mathbf{G})_j=0} W(\mathbf{uG}\oplus\mathbf{e}|\mathbf{u}^*\mathbf{G}) \Big] \\
&= \frac{1}{2^k} \sum_{\mathbf{e}\in\mathcal{E}} \sum_{\mathbf{u}} \Big[\omega_j^{(1\oplus(\mathbf{uG})_j)}(\mathbf{e}) \chi\{\, \hat{\varphi}_j(\mathbf{uG}\oplus\mathbf{e}) = 0 \,\} \\
&\qquad + \omega_j^{((\mathbf{uG})_j)}(\mathbf{e}) \chi\{\, \hat{\varphi}_j(\mathbf{uG}\oplus\mathbf{e}) = 1 \,\} \Big]. \qquad \text{(A.1)}
\end{aligned}$$

The last equality in (A.1) is derived from the following observation : for all $x = 0, 1$,

$$\begin{aligned}
\sum_{\mathbf{u}^*:\,(\mathbf{u}^*\mathbf{G})_j=x} W(\mathbf{uG}\oplus\mathbf{e}|\mathbf{u}^*\mathbf{G}) &= \sum_{\mathbf{u}^*:\,(\mathbf{u}^*\mathbf{G})_j=x} W((\mathbf{u}^*\oplus\mathbf{u})\mathbf{G}\oplus\mathbf{e}|\mathbf{0}) \\
&= \sum_{\mathbf{u}':\,(\mathbf{u}'\mathbf{G})_j=x\oplus(\mathbf{uG})_j} W(\mathbf{u}'\mathbf{G}\oplus\mathbf{e}|\mathbf{0}) \\
&= \omega_j^{(x\oplus(\mathbf{uG})_j)}(\mathbf{e}).
\end{aligned}$$

If $\mathbf{e} \in \hat{\mathcal{E}}_j$, then $\hat{\varphi}_j(\mathbf{uG}\oplus\mathbf{e}) = 1 \oplus (\mathbf{uG})_j \oplus e_j$ and

$$\begin{aligned}
&\omega_j^{(1\oplus(\mathbf{uG})_j)}(\mathbf{e}) \chi\{\, \hat{\varphi}_j(\mathbf{uG}\oplus\mathbf{e}) = 0 \,\} + \omega_j^{((\mathbf{uG})_j)}(\mathbf{e}) \chi\{\, \hat{\varphi}_j(\mathbf{uG}\oplus\mathbf{e}) = 1 \,\} \\
=\ &\omega_j^{(e_j)}(\mathbf{e}) \chi\{\, \hat{\varphi}_j(\mathbf{uG}\oplus\mathbf{e}) = 0 \,\} + \omega_j^{(e_j)}(\mathbf{e}) \chi\{\, \hat{\varphi}_j(\mathbf{uG}\oplus\mathbf{e}) = 1 \,\} \\
=\ &\omega_j^{(e_j)}(\mathbf{e}).
\end{aligned}$$

If $\mathbf{e} \in \mathcal{E}\backslash\hat{\mathcal{E}}_j$, then $\hat{\varphi}_j(\mathbf{uG}\oplus\mathbf{e}) = (\mathbf{uG})_j \oplus e_j$ and

$$\begin{aligned}
&\omega_j^{(1\oplus(\mathbf{uG})_j)}(\mathbf{e}) \chi\{\, \hat{\varphi}_j(\mathbf{uG}\oplus\mathbf{e}) = 0 \,\} + \omega_j^{((\mathbf{uG})_j)}(\mathbf{e}) \chi\{\, \hat{\varphi}_j(\mathbf{uG}\oplus\mathbf{e}) = 1 \,\} \\
=\ &\omega_j^{(1\oplus e_j)}(\mathbf{e}) \chi\{\, \hat{\varphi}_j(\mathbf{uG}\oplus\mathbf{e}) = 0 \,\} + \omega_j^{(1\oplus e_j)}(\mathbf{e}) \chi\{\, \hat{\varphi}_j(\mathbf{uG}\oplus\mathbf{e}) = 1 \,\} \\
=\ &\omega_j^{(1\oplus e_j)}(\mathbf{e}).
\end{aligned}$$

As a result, (A.1) can be continued as follows :

$$\begin{aligned}
P_{\mathrm{b},j}(\hat{\varphi}) &= \frac{1}{2^k} \sum_{\mathbf{e}\in\hat{\mathcal{E}}_j} \sum_{\mathbf{u}} \omega_j^{(e_j)}(\mathbf{e}) + \frac{1}{2^k} \sum_{\mathbf{e}\in\mathcal{E}\backslash\hat{\mathcal{E}}_j} \sum_{\mathbf{u}} \omega_j^{(1\oplus e_j)}(\mathbf{e}) \\
&= \sum_{\mathbf{e}\in\hat{\mathcal{E}}_j} \omega_j^{(e_j)}(\mathbf{e}) + \sum_{\mathbf{e}\in\mathcal{E}\backslash\hat{\mathcal{E}}_j} \omega_j^{(1\oplus e_j)}(\mathbf{e}). \qquad \text{(A.2)}
\end{aligned}$$

On the other hand,

$$
\begin{aligned}
p &= \sum_{\mathbf{y}:\, y_j=1} W(\mathbf{y}|\mathbf{0}) \\
&= \sum_{\mathbf{e}\in\mathcal{E}} \sum_{\mathbf{u}} W(\mathbf{uG}\oplus\mathbf{e}|\mathbf{0})\chi\{\, (\mathbf{uG})_j \oplus e_j = 1\,\} \\
&= \sum_{\mathbf{e}\in\mathcal{E}} \sum_{\mathbf{u}:\, (\mathbf{uG})_j = 1\oplus e_j} W(\mathbf{uG}\oplus\mathbf{e}|\mathbf{0}) \\
&= \sum_{\mathbf{e}\in\mathcal{E}} \omega_j^{(1\oplus e_j)}(\mathbf{e}), \qquad \text{(A.3)}
\end{aligned}
$$

and $p - P_{\mathrm{b},j}(\hat{\varphi})$ is expressed by the sum on the right–hand side of (3.6).

A.2 Proof of Proposition 1

One can easily see that

$$
\left.\begin{aligned}
\mathrm{wt}(\mathbf{uG}) &= d \\
(\mathbf{uG})_j &= x \\
e_j &= e
\end{aligned}\right\} \implies W(\mathbf{uG}\oplus\mathbf{e}|\mathbf{0}) = W_0(x\oplus e)\sum_{d=0}^{n} M_{j,d}^{(x)}\Gamma_{t-e}^{(d-x)}(p).
$$

Therefore

$$
\begin{aligned}
\sum_{\mathbf{e}\in\{0,1\}_t^n:e_j=0} \omega_j^{(x)}(\mathbf{e}) &= W_0(x)\sum_{d=0}^{n} M_{j,d}^{(x)}\Gamma_{t}^{(d-x)}(p), \\
\sum_{\mathbf{e}\in\{0,1\}_t^n:e_j=1} \omega_j^{(x)}(\mathbf{e}) &= W_1(x)\sum_{d=1}^{n} M_{j,d}^{(x)}\Gamma_{t-1}^{(d-x)}(p).
\end{aligned}
$$

These equalities and notations (3.2)–(3.4) imply that for all $t = \overline{0, t_{\min}}$,

$$
\frac{1}{\Theta_{j,t}(p)} \sum_{\mathbf{e}\in\{0,1\}_t^n:e_j=0} \left[\omega_j^{(1)}(\mathbf{e}) - \omega_j^{(0)}(\mathbf{e})\right] = p\alpha_{j,t}(p) - (1-p)(1-\alpha_{j,t}(p)),
$$

and, for all $t = \overline{1, t_{\min}}$,

$$
\frac{1}{\Theta_{j,t-1}(p)} \sum_{\mathbf{e}\in\{0,1\}_t^n:e_j=1} \left[\omega_j^{(0)}(\mathbf{e}) - \omega_j^{(1)}(\mathbf{e})\right] = p(1-\alpha_{j,t-1}(p)) - (1-p)\alpha_{j,t-1}(p),
$$

and (3.9), (3.10) follow.

A.3 Proof of Theorem 2

For any decoding φ', using the arguments of the proof of Lemma 1 (see (A.1)) with φ' substituted for $\hat{\varphi}$, we obtain

$$
\begin{aligned}
P_{\mathrm{b},j}(\varphi') &= \frac{1}{2^k}\sum_{\mathbf{e}\in\mathcal{E}}\sum_{\mathbf{u}}\Big[\omega_j^{(1\oplus(\mathbf{uG})_j)}(\mathbf{e})\chi\{\,\varphi_j'(\mathbf{uG}\oplus\mathbf{e}) = 0\,\} \\
&\qquad + \omega_j^{((\mathbf{uG})_j)}(\mathbf{e})\chi\{\,\varphi_j'(\mathbf{uG}\oplus\mathbf{e}) = 1\,\}\Big] \qquad \text{(A.4)} \\
&\geq \frac{1}{2^k}\sum_{\mathbf{e}\in\mathcal{E}}\sum_{\mathbf{u}} \min\{\,\omega_j^{(1\oplus(\mathbf{uG})_j)}(\mathbf{e}),\, \omega_j^{((\mathbf{uG})_j)}(\mathbf{e})\,\} \qquad \text{(A.5)} \\
&= \sum_{\mathbf{e}\in\mathcal{E}} \min\{\,\omega_j^{(0)}(\mathbf{e}),\, \omega_j^{(1)}(\mathbf{e})\,\}. \qquad \text{(A.6)}
\end{aligned}
$$

Since $\mathbf{y} = \mathbf{uG} \oplus \mathbf{e}$ implies $(\mathbf{uG})_j = y_j \oplus e_j$, the inequality in (A.5) can be replaced with the equality if and only if

$$\varphi'_j(\mathbf{y}) = \begin{cases} 0, & \text{if } \omega_j^{(y_j \oplus e_j)}(\mathbf{e}) > \omega_j^{(1 \oplus y_j \oplus e_j)}(\mathbf{e}), \\ 1, & \text{otherwise.} \end{cases} \tag{A.7}$$

An inspection of 4 possible values for the pair (y_j, e_j) shows that (A.7) is equivalent to $\varphi'_j(\mathbf{y}) = \varphi_j^{(\mathrm{opt})}(\mathbf{y})$, where $\varphi_j^{(\mathrm{opt})}(\mathbf{y})$ is defined in (3.12). Thus, (3.13) is proved and

$$P_{\mathrm{b},j}(\varphi^{(\mathrm{opt})}) \quad = \quad \sum_{\mathbf{e} \in \mathcal{E}} \min\left\{ \omega_j^{(0)}(\mathbf{e}), \omega_j^{(1)}(\mathbf{e}) \right\}. \tag{A.8}$$

If $\varphi'_j = \varphi_j^{(\mathrm{ML})}$, then $\varphi'_j(\mathbf{uG} \oplus \mathbf{e}) = (\mathbf{uG})_j$ for all $\mathbf{e} \in \mathcal{E}$. Therefore, using (A.4) we have

$$\begin{aligned} P_{\mathrm{b},j}(\varphi^{(\mathrm{ML})}) \quad &= \quad \frac{1}{2^k} \sum_{\mathbf{e} \in \mathcal{E}} \sum_{\mathbf{u}} \left[\omega_j^{(1)}(\mathbf{e}) \chi\{ (\mathbf{uG})_j = 0 \} + \omega_j^{(1)}(\mathbf{e}) \chi\{ (\mathbf{uG})_j = 1 \} \right] \\ &= \quad \sum_{\mathbf{e} \in \mathcal{E}} \omega_j^{(1)}(\mathbf{e}) \end{aligned} \tag{A.9}$$

and using (A.3), (A.8), (A.9) we obtain (3.14), (3.15).

A.4 Proof of Proposition 2

Suppose that $\mathbf{z}(\mathbf{y}) = \mathbf{z}$, $\boldsymbol{\xi}(\mathbf{y}) = \boldsymbol{\xi}$ and denote $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)$, where

$$\eta_j \triangleq \frac{W_1(y_j)}{W_0(y_j) + W_1(y_j)}, \quad j = \overline{1, n}.$$

By (4.9),

$$V_{\eta_j}(x_j) = \left| (1 \oplus x_j) - \frac{W_1(y_j)}{W_0(y_j) + W_1(y_j)} \right|$$

and using (4.8) we have

$$V_{\eta_j}(x_j) = \frac{W_{x_j}(y_j)}{W_0(y_j) + W_1(y_j)} = \frac{W_{x_j}(y_j)}{\pi(\xi_j)},$$

which means that

$$W_{x_j}(y_j) = \pi(\xi_j) V_{\eta_j}(x_j). \tag{A.10}$$

However, $\eta_j = \xi_j$ if $z_j = 0$ and $\eta_j = 1 - \xi_j$ if $z_j = 1$, as it follows from (4.5). Hence

$$V_{\eta_j}(x_j) = V_{\xi_j}(x_j \oplus z_j)$$

by (4.11), and $V(\mathbf{x}|\boldsymbol{\eta}) = V(\mathbf{x} \oplus \mathbf{z}|\boldsymbol{\xi})$. Using this equality, notations (4.10), and taking the product on $j = \overline{1, n}$ in (A.10) we obtain (4.16).

A.5 Proof of Lemma 2

We use transformations (4.19), (4.20) and similarly to (A.1)–(A.2) write

$$
\begin{aligned}
P_{\mathrm{b},j}(\hat{\varphi}) &= \frac{1}{2^k} \sum_{\mathbf{e}\in\mathcal{E}} \sum_{\mathbf{u},\mathbf{u}^*} V(\mathbf{u}^*\mathbf{G} \oplus \mathbf{u}\mathbf{G} \oplus \mathbf{e}|\boldsymbol{\xi})\chi\{\, \hat{\varphi}_j(\mathbf{y}(\mathbf{u}\mathbf{G} \oplus \mathbf{e}, \boldsymbol{\xi})) \neq (\mathbf{u}^*\mathbf{G})_j \,\} \\
&= \sum_{\boldsymbol{\xi}\in\mathcal{P}^n} \Pi(\boldsymbol{\xi}) \Big[\sum_{\mathbf{e}\in\hat{\mathcal{E}}_j} \upsilon_j^{(e_j)}(\mathbf{e}|\boldsymbol{\xi}) + \sum_{\mathbf{e}\in\mathcal{E}\setminus\hat{\mathcal{E}}_j} \upsilon_j^{(1\oplus e_j)}(\mathbf{e}|\boldsymbol{\xi}) \Big],
\end{aligned} \tag{A.11}
$$

where $\upsilon_j^{(0)}(\mathbf{e}|\boldsymbol{\xi})$, $\upsilon_j^{(1)}(\mathbf{e}|\boldsymbol{\xi})$ are defined in (4.18).

Using the definition of the conditional probability distribution $V(\mathbf{x}|\boldsymbol{\xi})$ in (4.9), (4.10) one can check that

$$
\xi_j = \sum_{\mathbf{x}:\, x_j=1} V(\mathbf{x}|\boldsymbol{\xi}), \quad \text{for all } \boldsymbol{\xi} \in \mathcal{P}^n.
$$

This equality can be continued as follows :

$$
\begin{aligned}
\xi_j &= \sum_{\mathbf{e}\in\mathcal{E}} \sum_{\mathbf{u}} V(\mathbf{u}\mathbf{G} \oplus \mathbf{e}|\boldsymbol{\xi})\chi\{\, (\mathbf{u}\mathbf{G})_j \oplus e_j = 1 \,\} \\
&= \sum_{\mathbf{e}\in\mathcal{E}} \sum_{\mathbf{u}:\, (\mathbf{u}\mathbf{G})_j = 1\oplus e_j} V(\mathbf{u}\mathbf{G} \oplus \mathbf{e}|\boldsymbol{\xi}) \\
&= \sum_{\mathbf{e}\in\mathcal{E}} \upsilon_j^{(1\oplus e_j)}(\mathbf{e}|\boldsymbol{\xi})
\end{aligned}
$$

and, by (4.10), (4.14),

$$
\sum_{\xi_j\in\mathcal{P}} \pi(\xi_j)\xi_j = \sum_{\boldsymbol{\xi}\in\mathcal{P}^n} \Pi(\boldsymbol{\xi})\xi_j = p.
$$

Therefore

$$
p = \sum_{\boldsymbol{\xi}\in\mathcal{P}^n} \Pi(\boldsymbol{\xi}) \sum_{\mathbf{e}\in\mathcal{E}} \upsilon_j^{(1\oplus e_j)}(\mathbf{e}|\boldsymbol{\xi}), \tag{A.12}
$$

and (4.17) follows from (A.11), (A.12).

A.6 Proof of Theorem 5

A direct extension of the result of Theorem 4 brings the equalities

$$
\begin{aligned}
p - P_{\mathrm{b},j}(\varphi) &= \sum_{t=0}^{t_{\min}} \Theta_{j,t}(p) \int_0^{1/2} \pi(\xi) \max\Big\{\, 0,\, \xi - (1 - \alpha_{j,t}(p)) \,\Big\} d\xi \\
&\quad + \sum_{t=1}^{t_{\min}} \Theta_{j,t-1}(p) \int_0^{1/2} \pi(\xi) \max\Big\{\, 0,\, \xi - \alpha_{j,t}(p) \,\Big\} d\xi,
\end{aligned}
$$

and (4.32) follows from the identities (4.28).

Assuming that p and j are fixed, denote $y_t = y_{j,t}(p)$, $\alpha_t = \alpha_{j,t}(p)$ for all $t \in \{0, \ldots, t_{\min}\}$. Then (4.31) is expressed as

$$
y_t = \frac{1}{4\sqrt{E}} \Big| \ln \frac{\alpha_t}{1 - \alpha_t} \Big|
$$

and using (4.25), (4.26) one can see that

$$
\xi(y_t) = \min\Big\{\, 1 - \alpha_t,\, \alpha_t \,\Big\}.
$$

Furthermore

$$
\begin{aligned}
|\,y\,| < y_t \quad &\Longrightarrow \quad \xi(y) > \begin{cases} 1-\alpha_t, & \text{if } \alpha_t > 1/2, \\ \alpha_t, & \text{if } \alpha_t < 1/2, \end{cases} \\
|\,y\,| \geq y_t \quad &\Longrightarrow \quad \xi(y) \leq \begin{cases} 1-\alpha_t, & \text{if } \alpha_t \geq 1/2, \\ \alpha_t, & \text{if } \alpha_t \leq 1/2. \end{cases}
\end{aligned}
$$

Since

$$\xi(y) \leq 1/2, \quad \text{for all } y \in (-\infty, +\infty),$$

these observations lead to the following continuation of (4.32) :

$$
\begin{aligned}
p - P_{\mathrm{b},j}(\varphi) \quad = \quad & \sum_{t=0}^{t_{\min}} \Theta_{j,t}(p)\chi\Big\{\, \alpha_t > 1/2 \,\Big\} \int_{-y_t}^{+y_t} W(y)\Big(\, \xi(y) - (1-\alpha_{j,t}(p))\,\Big)dy \\
& + \sum_{t=1}^{t_{\min}} \Theta_{j,t-1}(p)\chi\Big\{\, \alpha_{t-1} < 1/2 \,\Big\} \int_{-y_t}^{+y_t} W(y)\Big(\, \xi(y) - \alpha_{j,t}(p)\,\Big)dy.
\end{aligned}
$$

The values of the integrals in this expression above can be computed using (4.25) and the function Q introduced in (4.29). The result of these computations is given in (4.33).

Table 1: The values of the bit error probabilities for the AWGN channel with $\sigma^2 = 1/2$ when the $(6,3)$ shortened Hamming code and the $(23,12)$ Golay code are used.

p	E	$1 - P_{\mathrm{b},j}(\varphi)/p$	
		(6,3) code	(23,12) code
0.45	0.0078	$0.421 \cdot 10^{-3}$	$0.615 \cdot 10^{-11}$
0.40	0.0320	$0.427 \cdot 10^{-2}$	$0.549 \cdot 10^{-7}$
0.35	0.0742	$0.174 \cdot 10^{-1}$	$0.114 \cdot 10^{-4}$
0.30	0.1374	$0.476 \cdot 10^{-1}$	$0.487 \cdot 10^{-3}$
0.25	0.2274	0.103	$0.741 \cdot 10^{-2}$
0.20	0.3541	0.192	$0.515 \cdot 10^{-1}$
0.15	0.5270	0.321	0.202
0.10	0.8211	0.498	0.514
0.05	1.3527	0.725	0.876

Table 2: The Hamming weights of the vectors $\mathbf{uG} \oplus \mathbf{e}$ for the $(6,3)$ shortened Hamming code.

	$\mathbf{e}$							
	$\mathbf{0}$	$\mathbf{1}_1$	$\mathbf{1}_2$	$\mathbf{1}_3$	$\mathbf{1}_4$	$\mathbf{1}_5$	$\mathbf{1}_6$	$\mathbf{1}_{1,6}$
$\mathbf{uG}$	$\mathrm{wt}(\mathbf{uG} \oplus \mathbf{e})$							
000000	0	1	1	1	1	1	1	2
001011	3	4	4	2	4	2	2	3
010101	3	4	2	4	2	4	2	3
011110	4	5	3	3	3	3	5	6
100110	3	2	4	4	2	2	4	3
101101	4	3	5	3	3	5	3	2
110011	4	3	3	5	5	3	3	2
111000	3	2	2	2	4	4	4	3

Table 3: The weight distributions of codewords of the $(6,3)$ shortened Hamming code, $j = \overline{1,6}$.

d	$M_{j,d}^{(0)}$	$M_{j,d}^{(1)}$
0	1	0
1	0	0
2	0	0
3	2	2
4	1	2
5	0	0
6	0	0

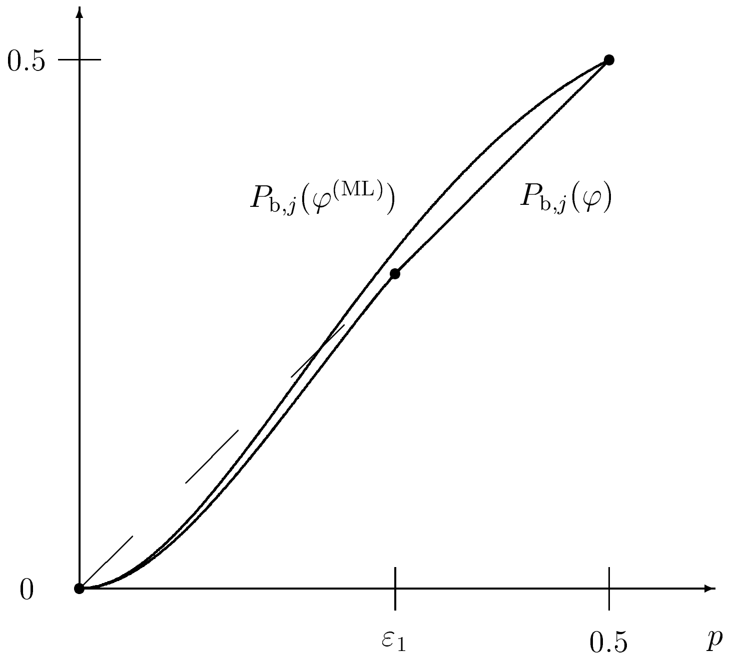

Figure 1: BSC(p) and the $(6,3)$ shortened Hamming code : the bit error probabilities for the decoding φ (the optimum decoding) and for the maximum likelihood decoding $\varphi^{(\mathrm{ML})}$ when $j \in \{1,6\}$ and the vector 100001 belongs to the set of coset representatives having the smallest Hamming weights; $\varepsilon_1 = 0.292$.

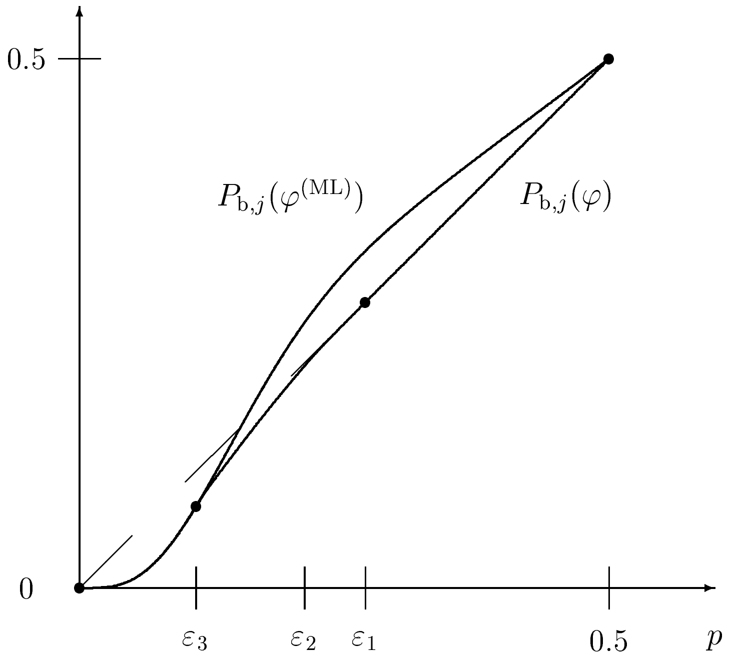

Figure 2: BSC(p) and the $(23, 12)$ Golay code : the bit error probabilities for the decoding φ (the optimum decoding) and for the maximum likelihood decoding $\varphi^{(\mathrm{ML})}$, where $(\varepsilon_1, \varepsilon_2, \varepsilon_3) = (0.278, 0.227, 0.117)$ and $j = \overline{1, 23}$.

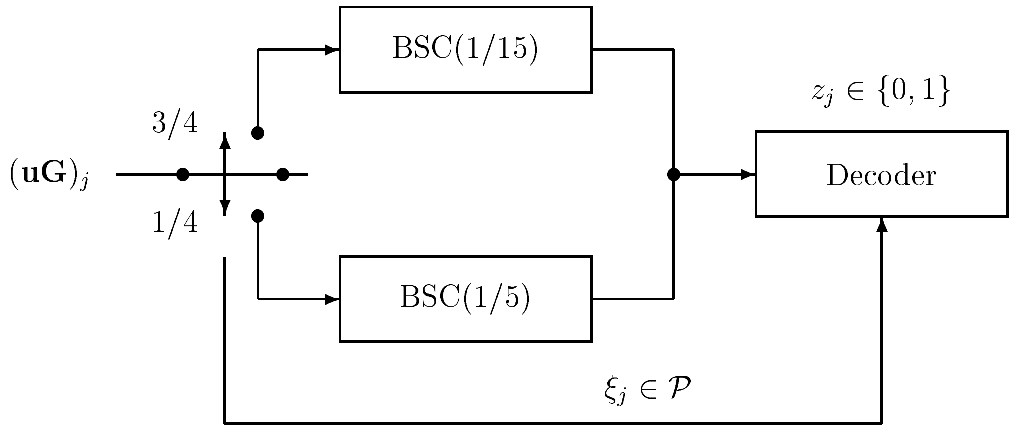

Figure 3: The scheme, which can be used to realize data transmission over a channel defined in (4.12).

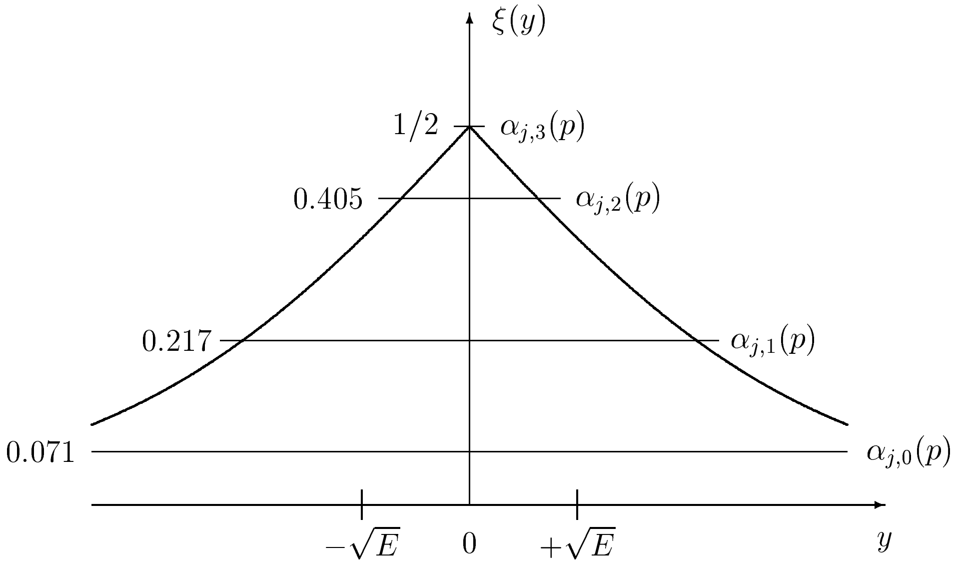

Figure 4: The function $\xi(y)$ for the AWGN channel when $E = 0.285$, $\sigma^2 = 1/2$, $p = 0.225$. The given values $\alpha_{j,0}(p), \ldots, \alpha_{j,3}(p)$ are computed for the $(23, 12)$ Golay code.

References

[1] J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and convolutional codes,” *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, 1996.

[2] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” *IEEE Trans. Inform. Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[3] L. A. Dunning, “Encoding and decoding for minimization of message symbol error rates in linear block codes,” *IEEE Trans. Inform. Theory*, vol. 33, no. 1, pp. 91–104, 1987.

[4] A. B. Kiely, J. T. Coffey, and M. R. Bell, “Optimal information bit decoding of linear block codes,” *IEEE Trans. Inform. Theory*, vol. 41, no. 1, pp. 130–140, 1995.

[5] S. Coffey and H. Chen, “Bit error probability and encoder structure,” in : *Coding, Communication and Broadcasting*, P. Farrel, M. Darnell, B. Honary (eds.), Research Studies Press Ltd., Baldock, Herdfordshire, England, pp. 29–38, 2000.

[6] A. Abedi, P. Chaudhari, and A. K. Khandani, “On some properties of the bit decoding algorithms,” in : *Proc. 7–th Canadian Workshop on Information Theory*, Vancouver, British Columbia, pp. 106–109, June 2001.

[7] R. Ahlswede, R., “Theory of information transfer,” Preprint. Submitted to *IEEE Trans. Inform. Theory*, 2001.

[8] S. Arimoto, “On the converse to the coding theorem for discrete memoryless channels,” *IEEE Trans. Inform. Theory*, vol. 19, no. 3, pp. 357–359, 1973.