Chapter 4

Linear Block Codes

For the construction of codes as well as for simple encoding and decoding schemes an algebraic structure is required for the set of codewords. The main criterion a linear code needs to fulfill is that the sum of two codewords is another codeword, whereas cyclic codes require additional algebraic structures which will be discussed in Chapter 5.

The required algebraic basics for linear codes are fairly easy. The numbers of correctable and detectable errors are determined by the minimum distance. In this chapter we will derive the basic relations between the minimum distance and the code parameters. Some of these results will also be partially valid for non-linear codes. Finally, we will introduce the weight distribution for calculating the error probabilities for error-detection and error-correction coding.

4.1 Structure of Linear Block Codes

According to Definition 1.4, the information symbols u_i and the encoded symbols a_i in an $(n, k, d_{\min})_q$ block code are q-ary with $u_i, a_i \in \mathbb{F}_q$, where the input alphabet $\mathcal{A}_{\text{in}}$ has been replaced by the Galois field $\mathbb{F}_q$. Therefore we will introduce Galois fields and vector spaces before defining linear codes.

4.1.1 Galois Fields and Vector Spaces

The Galois field $\mathbb{F}_q$ is a q-ary set on which the two binary operations $+$ (addition) and $-$ (multiplication) are defined. The corresponding inverse operations are subtraction and division, respectively. The rules satisfied by these arithmetic operations are similar to those for the set of real or rational numbers.

The advantage of Galois fields is that the finiteness of the set or alphabet corresponds to the finiteness of the technical systems. Also, there are no rounding or quantization errors as is the case with real numbers, but only exact results within the finite set. A disadvantage is the lack of an order relation, i.e., there is no "smaller than" or "greater than" relation between the elements of the Galois field.

A mathematical introduction to Galois fields can be found in the appendix Sections A.4 to A.8. Little knowledge of Galois fields is required for the simple linear and cyclic codes in Chapters 3 to 5. Thus, in the following, only the definition of Galois fields and a few examples are given.

Definition 4.1. *A* Galois field *(finite field)* $\mathbb{F}_q$ *is a* q*-ary set with two binary arithmetic operations, usually denoted* $+$ *(addition) and* $\cdot$ *(multiplication). The set* $\mathbb{F}_q$ *is* closed, *i.e.,* $x+y \in \mathbb{F}_q$ *and* $x\cdot y \in \mathbb{F}_q$ *for all* $x, y \in \mathbb{F}_q$*. The two operations must satisfy the following "usual" rules.*

1. Additive commutative law*:* $x + y = y + x$ *for all* $x, y \in \mathbb{F}_q$*.*

2. Additive associative law*:* $(x + y) + z = x + (y + z)$ *for all* $x, y, z \in \mathbb{F}_q$*.*

3. Additive identity*: there exists an element* $0 \in \mathbb{F}_q$ *such that* $x + 0 = x$ *for all* $x \in \mathbb{F}_q$*.*

4. Additive inverse Element*: for all* $x \in \mathbb{F}_q$ *there exists a unique element* $-x \in \mathbb{F}_q$ *such that* $x + (-x) = 0$*.*

5. Multiplicative commutative law*:* $x \cdot y = y \cdot x$ *for all* $x, y \in \mathbb{F}_q$*.*

6. Multiplicative associative law*:* $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ *for all* $x, y, z \in \mathbb{F}_q$*.*

7. Multiplicative identity*: there exists an element* $1 \in \mathbb{F}_q$ *such that* $x \cdot 1 = x$ *for all* $x \in \mathbb{F}_q$*.*

8. Multiplicative inverse element*: for all* $x \in \mathbb{F}_q\backslash\{\boldsymbol{0}\}$ *there exists a unique element* x^{-1} *with* $x \cdot x^{-1} = 1$*.*

9. Distributive law*:* $x \cdot (y + z) = x \cdot y + x \cdot z$ *for all* $x, y, z \in \mathbb{F}_q$*.*

The two identity elements 0 and 1 are also called neutral elements. With the properties (1) to (4) $\mathbb{F}_q$ *is also called an additive group and with the properties (5) to (8)* $\mathbb{F}_q\backslash\{0\}$ *is called a multiplicative group. We will use the following denotations:*

$$x + (-y) = x - y \quad , \quad x \cdot y = xy \quad , \quad x \cdot y^{-1} = \frac{x}{y}.$$

Finally, $\cdot$ *is to bind stronger than* $+$*. Subsequently, the operations in a Galois field are not distinguished from those in the field of real numbers, since the difference can usually be recognized from the context.*

Generally, $x \cdot 0 = 0$. There exist no zero divisors, i.e., $xy = 0$ is only possible if $x = 0$ or $y = 0$.

Galois fields $\mathbb{F}_q$ only exist for $q = p^m$, where p is prime and m is a positive integer. Thus q can only take on the values 2, 3, 4, 5, 7, 8, 9, 11, 13, 16, 17,... Only one unique Galois field exists for each q, because two Galois fields with the same cardinality are isomorph, i.e., by renaming the elements one Galois field emerges from the other (in other words they are similar in structure).

The most important cases are $q = 2$ (simple binary codes) and $q = 2^m$ (e.g., RS codes, see Chapter 7). Since $q = 2$ for simple codes, there is no need to understand $\mathbb{F}_{p^m}$ completely and introduce construction methods at this moment. As the statements in the following chapters are also valid for the more complicated codes over $\mathbb{F}_{p^m}$, all definitions and theorems are given for the general case. For now, we will accept that

$$1 + 1 = 0 \quad \text{in } \mathbb{F}_2 \text{ and } \mathbb{F}_{2^m} \text{ for all } m. \tag{4.1.1}$$

Example 4.1. For the simple case of $q = p$ prime, which is only interesting for Chapters 4 to 6, $\mathbb{F}_p$ consists of the non-negative integers $0, 1, 2, \ldots, p-1$, where the addition and multiplication are performed by modular arithmetic. Addition modulo p is usually denoted

$$x + y = z \quad \bmod p, \tag{4.1.2}$$

where z is obtained by adding up x and y using the standard integer addition in $\mathbb{Z}$ and dividing the result by p; z is the remainder of the division. The same applies to the multiplication.

If p is small, the arithmetic operations in $\mathbb{F}_p$ can be described in practice by tables.

(1) $\mathbb{F}_2$ is the most important case:

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1

For example $1 + 1 = 0$, $-1 = 1$, $-0 = 0$, $1^{-1} = 1$.

(2) Consider $\mathbb{F}_5$:

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

·	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

For example $-1 = 4$, $-2 = 3$, $2^{-1} = 3$, $3^{-1} = 2$, $4^{-1} = 4$.

(3) $\mathbb{F}_4$ is a Galois field, but it can not be defined by the modulo 4 operation:

$\cdot$	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

There exists no $x \in \mathbb{F}_4$ such that $2 \cdot x = 1$, i.e., 2^{-1} does not exist. ■

Definition 4.2. *The set of all n-tuples (also called blocks, vectors or words of length n) with components in $\mathbb{F}_q$ is denoted*

$$\mathbb{F}_q^n = \mathbb{F}_{p^m}^n = \{(x_0, \ldots, x_{n-1}) \mid x_0, \ldots, x_{n-1} \in \mathbb{F}_q\}.$$

Its cardinality is $|\mathbb{F}_q^n| = q^n$. An addition and a scalar multiplication are defined component-by-component, i.e., for $x, y \in \mathbb{F}_q^n$ and $\alpha \in \mathbb{F}_q$

$$\begin{array}{rcllcl} \boldsymbol{x} + \boldsymbol{y} & = & (x_0, \ldots, x_{n-1}) + (y_0, \ldots, y_{n-1}) & = & (x_0 + y_0, \ldots, x_{n-1} + y_{n-1}) \\ \alpha \cdot \boldsymbol{x} & = & \alpha \cdot (x_0, \ldots, x_{n-1}) & = & (\alpha x_0, \ldots, \alpha x_{n-1}). \end{array}$$

Hence, $\boldsymbol{x} + \boldsymbol{y} \in \mathbb{F}_q^n$ and $\alpha \cdot \boldsymbol{x} \in \mathbb{F}_q^n$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$ and all $\alpha \in \mathbb{F}_q$. The "usual" laws, which are listed in detail in Section A.5, are valid for these operations. Therefore $\mathbb{F}_q^n$ is a *vector space* or *linear space.* The same symbols are used for the operations in a vector space as well as in a Galois field and in the real field. A multiplication is not defined between two words.

Now, the equations (1.5.7) and (1.5.10) in Theorem 1.1 are intelligible. Furthermore, the Hamming distance is invariant to shifts:

$$d_H(\boldsymbol{x}, \boldsymbol{y}) = d_H(\boldsymbol{x} + \boldsymbol{z}, \boldsymbol{y} + \boldsymbol{z}). \tag{4.1.3}$$

Theorem 4.1. *In $\mathbb{F}_2^n$ and $\mathbb{F}_{2^m}^n$ arbitrary words $\boldsymbol{x}$ and $\boldsymbol{y}$ satisfy the following relations.*
(1) If $w_H(\boldsymbol{y})$ is even, then

$$w_H(\boldsymbol{x}) \textit{ even} \quad \Longleftrightarrow \quad w_H(\boldsymbol{x} + \boldsymbol{y}) \textit{ even.} \tag{4.1.4}$$

(2) If $w_H(\boldsymbol{y})$ is odd, then

$$w_H(\boldsymbol{x}) \textit{ even} \quad \Longleftrightarrow \quad w_H(\boldsymbol{x} + \boldsymbol{y}) \textit{ odd.} \tag{4.1.5}$$

For an $(n, k)_q$ block code, $\boldsymbol{u} = (u_0, \ldots, u_{k-1}) \in \mathbb{F}_q^k$ is valid for the information word and $\boldsymbol{a} = (a_0, \ldots, a_{n-1}) \in \mathbb{F}_q^n$ for the codeword and furthermore $\mathcal{C} \subseteq \mathbb{F}_q^n$ and $|\mathcal{C}| = q^k$ for the code itself.

4.1.2 Definition and Simple Properties of Linear Codes

After the preparations in the previous subsection, we will now turn to the definition of linear codes:

Definition 4.3. *An $(n,k)_q$ code $\mathcal{C}$ over $\mathbb{F}_q$ is called a* linear code*, if the sum of two codewords is another codeword:*

$$\boldsymbol{a},\boldsymbol{b} \in \mathcal{C} \quad \Longrightarrow \quad \boldsymbol{a}+\boldsymbol{b} \in \mathcal{C}.$$

In addition, non-binary codes with $q > 2$ must satisfy

$$\boldsymbol{a} \in \mathcal{C}, \alpha \in \mathbb{F}_q \quad \Longrightarrow \quad \alpha \cdot \boldsymbol{a} \in \mathcal{C}.$$

In other words, $\mathcal{C}$ is to be a vector space.

Consider some simple examples ($q = 2$):
$\mathcal{C} = \{000, 100, 010, 001\}$ is non-linear,
$\mathcal{C} = \{000, 110, 011, 111\}$ is non-linear,
$\mathcal{C} = \{000, 110, 101, 011\}$ is linear.

In practice, the restriction on linear codes does not imply a great loss – at least not with hard-decision decoding [139, 207]. Only few cases are known where the best linear code is worse than the best non-linear code. Therefore J.L.Massey [207] looks upon the investigation of non-linear codes as “a waste of time”. Nearly all codes used in practice are linear, so from now on we shall mostly consider linear codes. In contrast to this, Shannon information theory, as discussed in Chapter 3, is mainly based on non-linear random codes (see also Subsection 4.4.3 for further considerations). Some of the following propositions are valid for non-linear codes as well, though this will not be pointed out especially.

Example 4.2. **(1)** The $(n,1)_2$ code

$$\mathcal{C} = \{00\ldots0, 11\ldots1\} \tag{4.1.6}$$

is called a repetition code. The linearity is apparent. The code rate is $R = 1/n$ and the minimum distance is $d_{\min} = n$. Systematic encoding can use $u_0 \mapsto \boldsymbol{a} = (u_0, \ldots, u_0)$.

(2) The $(n, n-1)_2$ code

$$\mathcal{C} = \left\{ (a_0, \ldots, a_{n-1}) \;\middle|\; \sum_{i=0}^{n-1} a_i = 0 \right\} \tag{4.1.7}$$

is called a *parity-check code* (or single parity-check code, SPCC). This code is linear with $R = (n-1)/n = 1 - 1/n$. Since $000\ldots0$ and $110\ldots0$ are codewords, $d_{\min} = 2$. When systematically encoding with $(u_0, \ldots, u_{n-2}) \mapsto \boldsymbol{a} = (u_0, \ldots, u_{n-2}, u_0 + \cdots + u_{n-2})$, the sum of the information bits is attached as a parity-check bit. ■

Theorem 4.2. *A linear code $\mathcal{C}$ is invariant under additive shifts, i.e., $\mathcal{C}+\boldsymbol{b} = \{\boldsymbol{a}+\boldsymbol{b} \mid \boldsymbol{a} \in \mathcal{C}\} = \mathcal{C}$ for all $\boldsymbol{b} \in \mathcal{C}$.*

Due to $d_H(\boldsymbol{a},\boldsymbol{b}) = w_H(\boldsymbol{a}-\boldsymbol{b})$ the minimum distance of a code is equal to the minimum weight of the codewords, thus for determining $d_{\min}$, only q^k-1 words need to be considered instead of $q^k(q^k-1)$ pairs. This implies:

Theorem 4.3. *For a linear $(n,k,d_{\min})_q$ code $\mathcal{C}$ the minimum Hamming distance is equal to the minimum Hamming weight:*

$$\begin{aligned} d_{\min} &= \min\{d_H(\boldsymbol{a},\boldsymbol{b}) \mid \boldsymbol{a},\boldsymbol{b} \in \mathcal{C}, \boldsymbol{a} \neq \boldsymbol{b}\} \\ &= \min\{w_H(\boldsymbol{a}) \mid \boldsymbol{a} \in \mathcal{C}, \boldsymbol{a} \neq \boldsymbol{0}\}. \end{aligned} \tag{4.1.8}$$

4.2 Error Detection and Correction and Their Geometric Representations

A transmission using hard-decision demodulation, i.e., $\mathcal{A}_{\text{in}} = \mathcal{A}_{\text{out}} = \mathbb{F}_q$, can be modelled as a superposition of the codeword and an *error word* (or error pattern):

$$\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{e}. \tag{4.2.1}$$

Thus the received word $\boldsymbol{y}$ is composed of the sum of the transmitted word $\boldsymbol{a}$ and the error word $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{a} \in \mathbb{F}_q^n$. According to the linear structure, $\boldsymbol{y}$ being a codeword is equivalent to $\boldsymbol{e}$ being a codeword. Analogue to Section 1.6, there are the following possibilities:

$\boldsymbol{e} = \boldsymbol{0}$ Errorfree correct transmission.

$\boldsymbol{e} \in \mathcal{C}\backslash\{\boldsymbol{0}\}$ Falsification into a different codeword which can never be detected or corrected.

$\boldsymbol{e} \notin \mathcal{C}$ The error pattern is generally detectable and could perhaps be corrected by the decoder.

Definition 4.4. *An $(n,k)_q$ block code $\mathcal{C}$*
(1) corrects t errors *(also called t-error-correcting code), if the maximum-likelihood decoding returns the correct codeword for each error pattern $\boldsymbol{e}$ with $w_H(\boldsymbol{e}) \leq t$;*
(2) detects t' errors *(also called t'-error-detecting code), if for each error pattern $\boldsymbol{e} \neq \boldsymbol{0}$ with $w_H(\boldsymbol{e}) \leq t'$ the received word $\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{e}$ is not a codeword.*

These definitions indicate that *up to t* errors are corrected and up to t' errors are detected. It is crucial that *every* possible error pattern of weight $\leq t$ is correctable and that *every* possible error pattern of weight $\leq t'$ is detectable. Since the definitions refer to the errors being in arbitrary positions within the received word, the terms random-error-correcting code and random-error-detecting are also widely used. This also emphasizes the difference to the so-called burst-error-correcting codes and burst-error-detecting codes which will be discussed in Section 6.7.

4.2.1 Error Detection

Theorem 4.4. *An $(n, k, d_{\min})_q$ code detects $t' = d_{\min} - 1$ errors.*

Proof. According to Theorem 3.3, $\boldsymbol{e} \in \mathcal{C}\backslash\{\boldsymbol{0}\}$ inevitably implies that $w_H(\boldsymbol{e}) \geq d_{\min}$. In reverse: an error pattern $\boldsymbol{e}$ with $w_H(\boldsymbol{e}) \leq d_{\min}-1$ implies that $\boldsymbol{e} \notin \mathcal{C}\backslash\{\boldsymbol{0}\}$ and thus $\boldsymbol{e}$ is detected. ■

In the proof of Theorem 2.1 in Section 3.7 we had already introduced the concept of spheres:

Definition 4.5. *A* sphere $K_r(\boldsymbol{x})$ *of radius r centered around the word $\boldsymbol{x} \in \mathbb{F}_q^n$ is defined as the set of all words with a Hamming distance $\leq r$ from $\boldsymbol{x}$:*

$$K_r(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathbb{F}_q^n \mid d_H(\boldsymbol{x}, \boldsymbol{y}) \leq r\}. \tag{4.2.2}$$

$K_0(\boldsymbol{x}) = \{\boldsymbol{x}\}$ and $K_n(\boldsymbol{x}) = \mathbb{F}_q^n$ are apparent. Mostly, only spheres around the codewords are considered, but within the radius all words in $\mathbb{F}_q^n$ are in the spheres. According to combinatorics, there are an exact $\binom{n}{i}(q-1)^i$ words $\boldsymbol{y} \in \mathbb{F}_q^n$ with $d_H(\boldsymbol{x}, \boldsymbol{y}) = i$, leading to the following important result for the cardinalities of the spheres:

$$\left|K_r(\boldsymbol{x})\right| = \sum_{i=0}^{r} \binom{n}{i} (q-1)^i. \tag{4.2.3}$$

Example 4.3. Let us consider the $(3,1)_2$ repetition code $\mathcal{C} = \{000, 111\}$ with $d_{\min} = 3$. The spheres around the codewords are as follows.

$K_1(000) = \{000, 100, 010, 001\}$
$K_1(111) = \{111, 110, 101, 011\}$
$K_2(000) = \{000, 100, 010, 001, 110, 101, 011\}$
$K_2(111) = \{111, 110, 101, 011, 001, 010, 100\}$
$K_3(000) = K_3(111) = \mathbb{F}_2^3$

The sphere $K_2(100)$ around a non-codeword contains both codewords. ■

Figure 3.1 shows that every sphere of radius $d_{\min} - 1$ around a codeword can not contain another codeword. If two codewords $\boldsymbol{a}$ and $\boldsymbol{b}$ with $\boldsymbol{b} \in K_{d_{\min}-1}(\boldsymbol{a})$ existed, then $d_H(\boldsymbol{a}, \boldsymbol{b}) \leq d_{\min} - 1$, yet $d_{\min}$ is the minimum distance between two codewords.

If there is a maximum of $d_{\min} - 1$ errors, the received word lies within the sphere around the actual transmitted codeword. Since there is no other

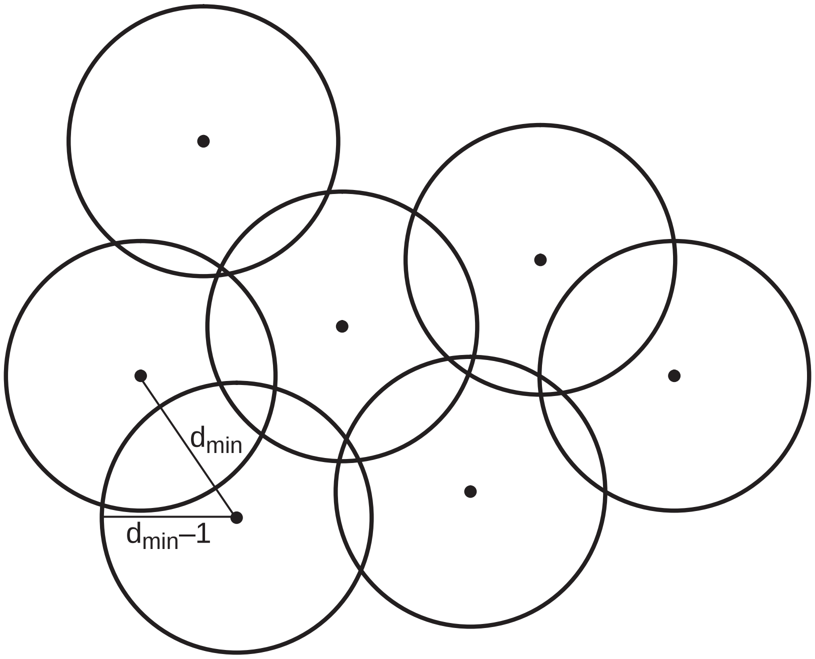

Figure 4.1. Spheres of radius $d_{\min} - 1$ around the codewords

codeword in this sphere, the received word can not be mistaken for a codeword – apart from the errorfree transmission, of course. However, several codewords can be in a sphere of radius $d_{\min} - 1$ around an arbitrary word.

According to Theorem 3.4, all $\sum_{i=0}^{d_{\min}-1} \binom{n}{i}(q-1)^i$ error patterns of a weight $\leq d_{\min}-1$ are detected. Error patterns of a higher weight can not all be detected, but the majority will be since every $\boldsymbol{e} \notin \mathcal{C}$ is detected and their number is $q^n - q^k$. The rate of detectable error patterns of higher weight will be more precisely determined for the cyclic codes in Section 5.7.

4.2.2 Error Correction

The following theorem, despite its easy proof, is one of the most important results delivered by the coding theory:

Theorem 4.5. *An $(n, k, d_{\min})_q$ code corrects t errors, if $2t + 1 \leq d_{\min}$. This inequality is equivalent to*

$$t = \lfloor (d_{\min} - 1)/2 \rfloor$$

where $\lfloor \lambda \rfloor$ denotes the largest integer $\leq \lambda$.

Proof. Let $\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{e}$ with $w_H(\boldsymbol{e}) \leq t$, then $d_H(\boldsymbol{y}, \boldsymbol{a}) \leq t$. Let $\boldsymbol{b} \in \mathcal{C}$ be arbitrary with $\boldsymbol{b} \neq \boldsymbol{a}$. The triangle inequality implies that

$$2t + 1 \leq d_{\min} \leq d_H(\boldsymbol{a}, \boldsymbol{b}) \leq d_H(\boldsymbol{a}, \boldsymbol{y}) + d_H(\boldsymbol{y}, \boldsymbol{b}) \leq t + d_H(\boldsymbol{y}, \boldsymbol{b}).$$

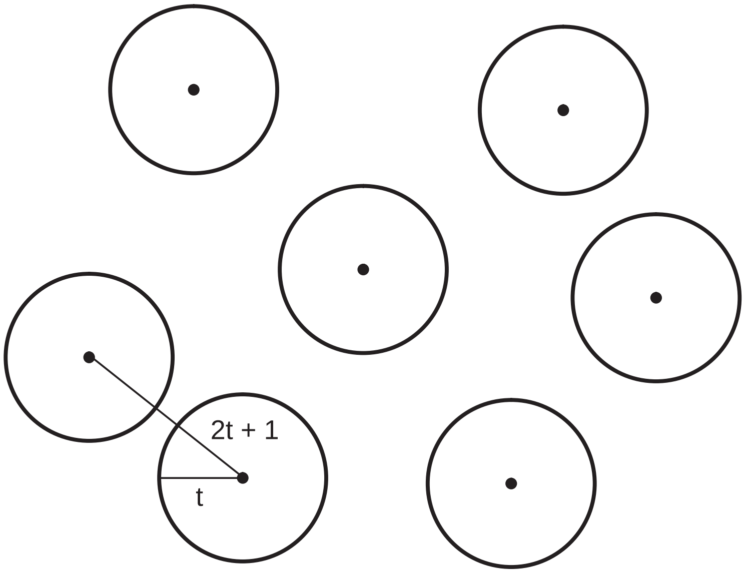

Figure 4.2. Decoding spheres of radius $t = \lfloor (d_{\min} - 1)/2 \rfloor$ around the codewords

Thus $d_H(\boldsymbol{y}, \boldsymbol{b}) \geq t + 1$. Therefore the distance of $\boldsymbol{a}$ from $\boldsymbol{y}$ is t at the most, whereas every other codeword has a distance of at least $t+1$ from $\boldsymbol{y}$. Therefore the ML decoder picks the correct codeword. ■

Figure 3.2 shows the disjoint spheres of radius t around the codewords. If during the transmission the codeword $\boldsymbol{a}$ is overlayed by t errors at the most, the received word $\boldsymbol{y}$ lies in $K_t(\boldsymbol{a})$ and has a distance of at least $t+1$ from every other codeword. Therefore the codeword with the smallest distance from $\boldsymbol{y}$ is uniquely determined. Thus the spheres of radius t around the codewords are also called *decoding spheres*. Received words with more than t errors are either in a wrong sphere and then decoded incorrectly or they are between the decoding spheres and are either decoded correctly or incorrectly.

Let us now consider spheres of radius t around arbitrary words, again there is only one codeword at the most within these spheres, otherwise because of $\boldsymbol{a}, \boldsymbol{b} \in K_t(\boldsymbol{x})$ there would be a contradiction:

$$2t + 1 \leq d_{\min} \leq d_H(\boldsymbol{a}, \boldsymbol{b}) \leq d_H(\boldsymbol{a}, \boldsymbol{x}) + d_H(\boldsymbol{x}, \boldsymbol{b}) \leq t + t = 2t.$$

Example 4.4. (1) Using the $(3, 1, 3)_2$ code $\mathcal{C} = \{000, 111\}$ with $t = 1$ the received words $000, 100, 010, 001$ are decoded into 000 and $111, 011, 101, 110$ are decoded into 111.

(2) Using the $(2, 1, 2)_2$ code $\mathcal{C} = \{00, 11\}$ with $t = 0$, the erroneous received words 01 and 10 are detected as corrupted, but can not be decoded correctly. In this case the ML decoder should decide randomly.

(3) Consider the slightly strange $(6, 2, 3)_2$ code with $t = 1$ given by

$$\mathcal{C} = \{\underbrace{00\ 0000}_{3}, \underbrace{10\ 1100}_{4}, \underbrace{01\ 0111}_{3}, \underbrace{11\ 1011}_{2}\}.$$

The first two bits are the information bits. Since $t = 1$ only one error can be corrected reliably. If $\boldsymbol{y} = 00\ 1011$ is received, the distance from the codewords are those given below the braces, therefore the ML-decoder decides on 11 1011, i.e., both information bits are corrected. ■

4.2.3 Combination of Error Correction and Error Detection

Error detection and error correction can also be performed simultaneously.

Theorem 4.6. *An $(n, k, d_{\text{min}})_q$ code can correct t errors and detect t' errors (with $t' \geq t$) simultaneously, if*

$$t + t' + 1 \leq d_{\text{min}}. \tag{4.2.4}$$

For $t = 0$ this corresponds to Theorem 3.4 and for $t = t'$ to Theorem 3.5.

In practice, a codeword is sought with a distance $\leq t$ from the received word. If such a codeword is found, it is the decoding result. If such a codeword is not found, all error patterns of a weight $\leq d_{\text{min}} - t - 1$ are detectable. This is also shown in Figure 4.3, where the hatched decoding spheres of radius t and the larger error-detection spheres of radius t' do not intersect if $t+t'+1 \leq d_{\text{min}}$. All received words within the t'-spheres but outside of the t-spheres are detected as erroneously.

Proof. Form the set $\mathcal{V} = \bigcup\limits_{\boldsymbol{b}\in\mathcal{C}} K_t(\boldsymbol{b})$ of correctable words with a maximum of t errors. Let $\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{e}$ be a received word with $w_H(\boldsymbol{e}) = f$ and $t < f \leq t'$. Assume $\boldsymbol{y} \in \mathcal{V}$, then there would exist a $\boldsymbol{b} \in \mathcal{C}$ with $\boldsymbol{y} \in K_t(\boldsymbol{b})$. Since $\boldsymbol{y}$ has a bigger distance from $\boldsymbol{a}$ than t, $\boldsymbol{b} \neq \boldsymbol{a}$, thus we conclude from the triangle inequality that

$$t + t' + 1 \leq d_{\text{min}} \leq d_H(\boldsymbol{a}, \boldsymbol{b}) \leq \underbrace{d_H(\boldsymbol{a}, \boldsymbol{y})}_{= f} + \underbrace{d_H(\boldsymbol{y}, \boldsymbol{b})}_{\leq t} \leq t' + t.$$

This is a contradiction, therefore $\boldsymbol{y} \notin \mathcal{V}$. So the received word is not in the set of correctable error patterns and thus is detected as erroneous. ■

Example 4.5. Consider the code $\mathcal{C} = \{000, 111\}$ with $d_{\text{min}} = 3$ once more. If $t = 1$ and $t' = 1$, 100 is decoded into 000. However, if $t = 0$ and $t' = 2$, 100 is detected as erroneous (emerging from 000 or 111). The combination $t = 1, t' = 2$ is impossible, because on the one hand 100 would be decoded into 000, on the other hand it would be detected as erroneous (emerging from 111). ■

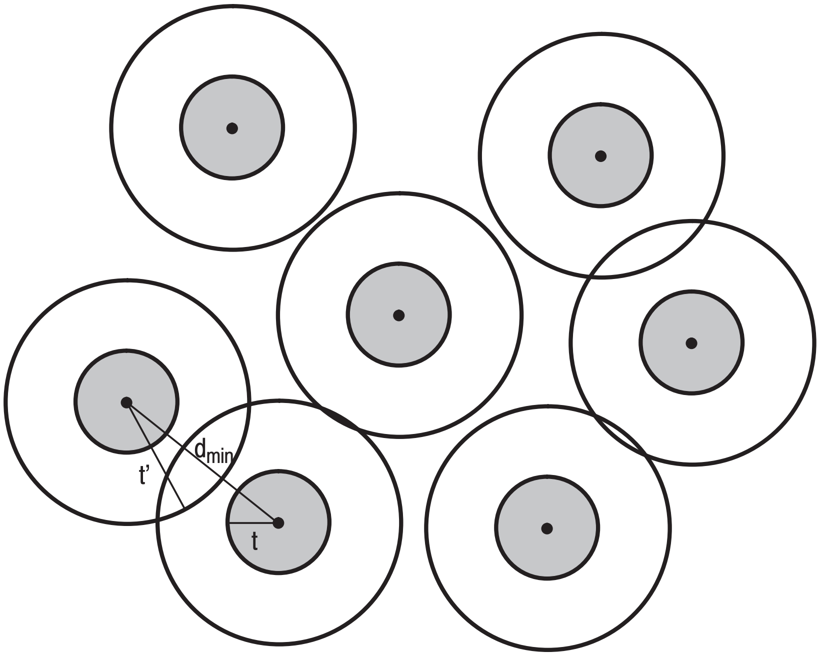

Figure 4.3. Decoding spheres of radius t and error-detection spheres of radius t' around the codewords

4.2.4 Decoding Rules for Hard-Decision Channels

Now, we will generalize the situation in Example 3.4(3). Let there be an $(n, k, d_{\min})_q$ code with $2t + 1 \leq d_{\min}$. The received word $\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{e}$ is to have $f = d_H(\boldsymbol{a}, \boldsymbol{y})$ errors. Using the ML estimation $\hat{\boldsymbol{a}}$, $f_{ML} = d_H(\boldsymbol{y}, \hat{\boldsymbol{a}})$ corrections are made in the received word. Two possible results have to be distinguished:

(a) If $f \leq t$, the ML estimation is correct with $\hat{\boldsymbol{a}} = \boldsymbol{a}$ and $f_{ML} = f$.

(b) If $f > t$, the ML estimation might be wrong with $\hat{\boldsymbol{a}} \neq \boldsymbol{a}$, and f_{ML} can become uncontrollably big (up to $f_{ML} = n$). The triangle inequality leads to $d_H(\boldsymbol{a}, \hat{\boldsymbol{a}}) \leq f + f_{ML}$, therefore the ML estimation may deviate in uncontrollably many positions from the transmitted codeword, although the code can only correct t errors.

For “safety reasons”, the correction during the decoding could be restricted to a maximum of $d_H(\boldsymbol{y}, \boldsymbol{a}) \leq t$ positions, which is actually used for the so-called BMD method:

Definition 4.6. *The* bounded-minimum-distance decoder *(BMD) is only defined for received words which lie in any decoding sphere. Let $\boldsymbol{y}$ be a received word. If a codeword $\boldsymbol{a}$ exists with $d_H(\boldsymbol{a}, \boldsymbol{y}) \leq t$, then $\boldsymbol{a}$ is the decoder output, of course, since $\boldsymbol{a}$ has minimum distance from $\boldsymbol{y}$. If there is no codeword $\boldsymbol{a}$ with $d_H(\boldsymbol{a}, \boldsymbol{y}) \leq t$, a decoder failure is declared.*

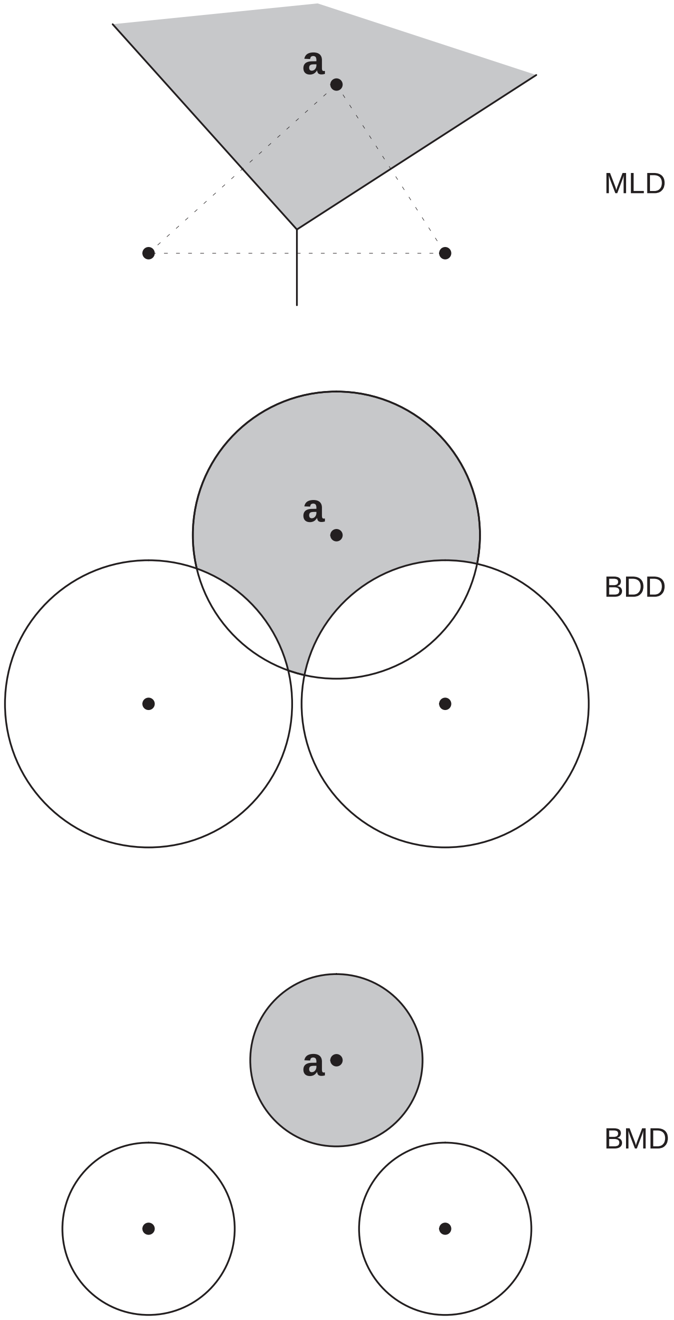

Figure 4.4. Comparison of Decoding Rules

Figure 3.3 shows a comparison of the three decoding methods MLD, BDD and BMD (assuming hard decisions). The bullets denote the codewords and $\boldsymbol{a}$ is the transmitted codeword. The hatched area surrounding $\boldsymbol{a}$ represents the *decision region* consisting of the received words which are decoded into $\boldsymbol{a}$. The decision regions can be denoted $\delta^{-1}(\boldsymbol{a})$ with the codeword estimator δ as given in (1.6.2). Generally, the decision regions do not intersect, but only for

MLD the whole space $\mathbb{F}_q^n$ of received words is completely covered by the decision regions. In contrast, for BDD and BMD the decision regions do not cover $\mathbb{F}_q^n$ completely, thus the decoder output is not defined for the received words which are not contained in any decision region (in other words the decoder might act uncontrollably which is always considered a decoder failure).

MLD (Maximum-Likelihood Decoder, see Theorem 1.3): For each received codeword the nearest codeword is chosen. In the 2-dimensional case, the decision region is bounded by the mid-perpendiculars of the triangles created by the codewords as end-points. In the n-dimensional case, it is bounded by the corresponding hyperplanes.

The MLD minimizes the word-error probability (under the assumption of equal a priori probabilities).

BDD (Bounded-Distance Decoder, see also the proof of the channel coding theorem in Section 2.7): Around each codeword a sphere of the same radius t according to (2.7.4) is created. Only received words which are in exactly one sphere are decoded into the center of the sphere. There is no decoding for words which are in no sphere or in more than one sphere. Thus, the decision regions are of a fairly complicated geometrical form.

Since the BDD is used for the proof of the channel coding theorem, it is hardly worse than the MLD.

BMD (Bounded-Minimum-Distance Decoder, see Definition 3.6): Around each codeword a sphere of the same radius $t = \lfloor (d_{\min} - 1)/2 \rfloor$ is created as the decision region. Thus, the spheres are definitely disjoint. Only received words which are in exactly one sphere are decoded into the center of the sphere. There is no decoding for words outside the spheres.

Thus, the BMD performs a "decoding up to half the minimum distance".

The BMD is, of course, worse than the MLD method, but the important advantage lies in the simpler realization of decoders for important code families like the RS and BCH codes (see Chapter 8 for more details). The comparison between MLD and BMD will be continued in Subsection 4.4.2.

More details of error-correction and error-detection with BMD decoders for the specific class of MDS codes (to be introduced in the next subsection) are covered in Subsection 8.1.4, where we will also consider the computation of $f_{ML} = d_H(\boldsymbol{y}, \hat{\boldsymbol{a}})$ in detail.

4.3 Bounds on Minimum Distance

After reading the previous section, the meaning of $d_{\min}$ and t is evident. But how do these values relate to the code parameters n, k and q?

4.3.1 The Singleton Bound and Maximum Distance Separable (MDS) Codes

Theorem 4.7 (Singleton bound). *A linear $(n, k, d_{\min})_q$ code satisfies*

$$d_{\min} \leq n - k + 1. \tag{4.3.1}$$

A code which satisfies this bound with equality is called a maximum distance separable *(MDS) code.*

Proof. All codewords differ in at least $d_{\min}$ positions. If the first $d_{\min} - 1$ positions of all codewords are deleted, the shortened codewords of length $n - d_{\min} + 1$ still differ. Thus there are q^k different shortened codewords in the space of the $q^{n-d_{\min}+1}$ shortened words. However, this is only possible if $k \leq n - d_{\min} + 1$. ■

Therefore $2t \leq n-k$ is required for any t-error-correcting code and $t' \leq n-k$ for any t'-error-detecting code. Hence, two parity-check bits are needed for the correction, but only one for the detection of an error.

The only binary MDS codes are the trivial $(n, 1, n)_2$ repetition codes. In comparison, the q-ary MDS codes, including the RS codes (see Chapter 7), are of great practical importance. Nevertheless, MDS codes do not represent the absolute optimum, thus the development of block codes has not yet reached its end. An MDS code could be improved by reducing the cardinality q of the alphabet with identical parameters n, k and $d_{\min}$. The channel coding theorem can not be proven by using MDS codes.

A very important property of the MDS codes is stated as follows.

Theorem 4.8. *In an $(n, k, n-k+1)_q$ MDS code the complete codeword is uniquely determined by an arbitrary combination of k codeword positions.*

Proof by contradiction. Assume that there are two different codewords $\boldsymbol{a}$ and $\boldsymbol{b}$, which are identical in k positions. Thus they differ in at most $n-k$ positions, which means $d_H(\boldsymbol{a}, \boldsymbol{b}) \leq n - k$. However, $n - k + 1$ is the minimum distance between the codewords, therefore the assumption is wrong. ■

4.3.2 The Hamming Bound and Perfect Codes

Theorem 4.9 (Hamming Bound, Sphere-Packing Bound). *A linear $(n, k, d_{\min})_q$ code which can correct up to t errors satisfies*

$$q^{n-k} \geq \sum_{r=0}^{t} \binom{n}{r} (q-1)^r \quad or \quad n - k \geq \log_q \left[\sum_{r=0}^{t} \binom{n}{r} (q-1)^r \right]. \tag{4.3.2}$$

In the specific case of $q = 2$ this reduces to

$$2^{n-k} \geq \sum_{r=0}^{t} \binom{n}{r} = 1 + n + \binom{n}{2} + \cdots + \binom{n}{t}. \tag{4.3.3}$$

A code which satisfies the Hamming bound with equality for a suitable integer t is called a perfect code. *In this case the decoding rules MLD and BMD are identical.*

Proof. The decoding spheres with the codewords as centers are mutually exclusive. Since there are q^k codewords, according to (3.2.8), the total number of words in all decoding spheres is exactly

$$q^k \cdot \sum_{r=0}^{t} \binom{n}{r} (q-1)^r.$$

This number must be smaller than or equal to the total number q^n of all words. Equality in the Hamming bound implies that the decoding spheres are so densely packed that they cover the whole space $\mathbb{F}_q^n$. ■

Theorem 3.9 is one of the most important practical results of coding theory. If a suitable code is chosen, the maximum error correction capability can be determined for any given numbers n, k and q. However, the Hamming bound is typically not very tight. If a bad code is chosen, t could be much smaller than promised by the Hamming bound.

The Hamming bound does not tell us anything about the existence of codes. If the Hamming bound is satisfied for a parameter combination n, k, t, q, the existence of an appropriate code with $d_{\min} \geq 2t+1$ is not necessarily guaranteed. Only the opposite case is given: if the parameter combination does not fit the Hamming bound, then in principle an appropriate code can not exist. This is also valid for the Singleton bound, the Elias bound (see Section 3.4.1) and the Plotkin bound (see Section 3.3.3).

Example 4.6. (1) The $(7, 4, 3)_2$ Hamming code in Example 1.2 with $d_{\min} = 3$ and $t = 1$ is perfect, because $2^{7-4} = 1 + 7$. There does not exist a word outside of the code $\mathcal{C}$, such that the distance from all 16 codewords is ≥ 2.

(2) The existence of the so-called $(23, 12, 7)_2$ *Golay code* with $t = 3$ (which was already displayed in Figure 1.10) will not be proven here. The only easy proof is to show that such a code is perfect:

$$2^{23-12} = 2048 = 1 + \binom{23}{1} + \binom{23}{2} + \binom{23}{3}.$$

(3) Consider a $(127, 113, d_{\min})_2$ code (which will turn out to be a BCH code in Chapter 7). We are looking for $d_{\min}$ under the assumption that the code has been constructed in the best possible way. The Hamming bound gives us

$$2^{127-113} = 16384 \geq \left\{ \begin{array}{lll} 1+127+\binom{127}{2} & = 8129 & \text{for } t=2 \\ 1+127+\binom{127}{2}+\binom{127}{3} & = 341504 & \text{for } t=3 \end{array} \right\}.$$

It follows that $t = 2$ and therefore $d_{\min} = 5$ or $d_{\min} = 6$. Thus, with 113 information bits and 14 parity-check bits either 2 errors can be corrected or 4 (maybe even 5) errors can be detected.

(4) According to the Hamming bound, a $(20, 10)_2$ code with $t = 2$ and a $(100, 50)_2$ code with $t = 11$ could exist where the code rate is 1/2. When repeated five times, a $(100, 50)$ code can emerge from the $(20, 10)$ code by separately encoding sections of 10 information bits as done with the $(20, 10)$ code. But the resulting code can correct neither 11 nor 10, but still only 2 errors, because 3 errors in a section of length 20 are not correctable.

Thus good codes with a big block length can not emerge from a code with a short block length by using the simple repetition method. ■

We emphasize that the term "perfect" code should not be taken too literally. Perfect codes are far from being the best of all possible error control codes, since the sphere-packing problem and the error-control problem are not entirely equivalent. Many codes which are highly imperfect (e.g., MDS and RS codes introduced in Chapter 8) provide more powerful error-control techniques than the perfect codes. Only few classes of perfect codes exist at all (we skip the time-consuming proof), namely the Hamming codes (see Section 4.4 for the general definition), some Golay codes (not covered in detail here) and binary repetition codes of odd block length:

Theorem 4.10. *The $(n, 1)_2$ repetition codes are perfect for an odd n.*

Proof. Let $n = d_{\min} = 2t + 1$ and $\boldsymbol{y} \in \mathbb{F}_2^n$. For $w_H(\boldsymbol{y}) \leq t$, $d_H(\boldsymbol{y}, \boldsymbol{0}) = w_H(\boldsymbol{y}) \leq t$ and for $w_H(\boldsymbol{y}) \geq t + 1$, $d_H(\boldsymbol{y}, \boldsymbol{1}) = n - w_H(\boldsymbol{y}) \leq t$. Therefore $K_t(\boldsymbol{0}) \cup K_t(\boldsymbol{1}) = \mathbb{F}_2^n$, thus the t-spheres around the codewords cover the whole space. ■

4.3.3 The Plotkin Bound

Theorem 4.11 (Plotkin Bound). *A linear $(n, k, d_{\min})_q$ code satisfies*

$$d_{\min} \leq \frac{n(q-1)q^{k-1}}{q^k - 1} \approx \frac{n(q-1)}{q}. \tag{4.3.4}$$

The approximation is valid for a large k.

Proof. Due to symmetry reasons, every code symbol takes on each of the q possible values with the same probability. Thus $(q-1)/q$ is the average weight of a code symbol and $n(q-1)/q$ is the average weight of a codeword. Omitting the all-zero word, the average weight of a codeword increases to

$$\frac{n(q-1)}{q} \cdot \frac{q^k}{q^k-1}.$$

This average weight is, of course, greater than the minimum weight. ■

4.3.4 The Gilbert-Varshamov Bound

The preceding Singleton, Hamming and Plotkin bounds provide so-called *upper bounds* for the minimum distance of a code, but the existence of such a code is not guaranteed. The upper bounds only prove that codes with certain combinations of parameters do not exist. This raises the question of what ranges of parameters are possible. Numerous specific answers will be given by the codes of the following chapters, however, a general answer is given by the Gilbert-Varshamov bound. This is a so-called *lower bound* which does guarantee the existence of a code, but similar to the channel coding theorem, the lower bound only guarantees the existence but does not provide a useful method for constructing codes which satisfy the Gilbert-Varshamov bound.

Theorem 4.12 (Gilbert-Varshamov Bound). *There always exists a linear $(n,k,d_{\min})_q$ code, if*

$$\sum_{r=0}^{d_{\min}-2} \binom{n-1}{r} (q-1)^r < q^{n-k}. \tag{4.3.5}$$

Various other forms of this bound are listed for example in [23].

Proof. In anticipation of Theorem 4.4, which we are still to derive, we will show that a parity-check matrix with n columns of length $n-k$ can be constructed, so that every selection of a set of $d_{\min}-1$ columns is linearly independent.

Select any non-zero $(n-k)$-tuple as the first column. Then select any non-zero $(n-k)$-tuple except for multiples of the first as the second column. The third column may be any $(n-k)$-tuple which is not a linear combination of the first two columns. If $n-1$ columns have been constructed, it is to be shown that an n-th column can be constructed.

In order to ensure that each selection of $d_{\min}-1$ out of the n columns is linearly independent, the n-th column is not to be a linear combination of any of $d_{\min}-2$ or fewer columns of the first $n-1$ columns.

The number of linear combinations of r out of $n-1$ columns is $\binom{n-1}{r}(q-1)^r$, thus there are $l = \sum_{r=1}^{d_{\min}-2} \binom{n-1}{r}(q-1)^r$ linear combinations of $\leq d_{\min} - 2$ columns out of $n-1$ columns. There are q^{n-k} possibilities for the n-th column, of which the l possibilities and the zero column are excluded, i.e., $q^{n-k} > l + 1$. ■

The proof of the Gilbert-Varshamov bound, like the proof of the channel coding theorem or the R_0 theorem, does not leave us with a useful construction method for good codes, for if the selection of the columns was bad, the codes are of a useless or chaotic structure.

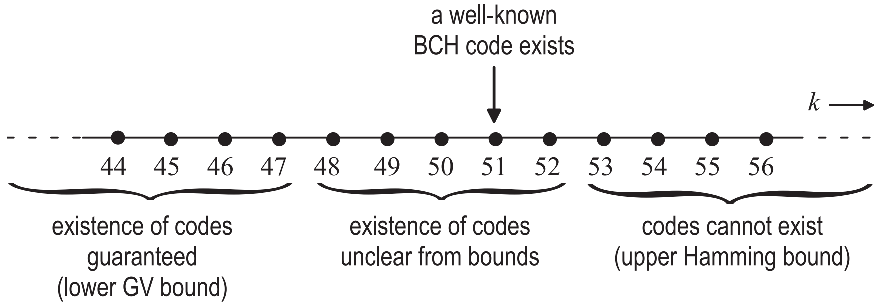

Figure 4.5. Gilbert-Varshamov and Hamming bounds for $(63, k, 5)_2$ codes

Example 4.7. Consider a $(63, k, 5)_2$ code. The correction of 2 errors is required and as few parity-check bits as possible are to be used. The Hamming bound implies that

$$\sum_{r=0}^{2} \binom{63}{r} = 1 + 63 + 1953 = 2017 \leq 2^{n-k}.$$

Therefore a code with only 11 parity-check bits might exist, thus $k \leq 52$. The Gilbert-Varshamov bound implies that

$$\sum_{r=0}^{3} \binom{62}{r} = 1 + 62 + 1891 + 37820 = 39774 < 2^{n-k}.$$

Thus the existence of a code with 16 parity-check bits is guaranteed, i.e., $k \geq 47$. There does actually exist a $(63, 51, 5)_2$ BCH code (see Table 7.1) which is fairly good so the search for a better code is hardly worth the effort. Both bounds and the BCH code are illustrated in Figure 4.4. ■

4.4 Asymptotic Bounds on Minimum Distance

4.4.1 Comparison of Bounds

Now, we will consider the limit of the minimum distance as $n \to \infty$, where the code rate R remains constant, thus implying that $k = Rn \to \infty$. For all upper or lower bounds in the last section there are asymptotic forms for which the so-called *distance rate* $d_{\min}/n$ converges to a limit as $n \to \infty$. The asymptotic bounds provide a relation between these limits of the distance rate and the code rate. In the following we will only consider binary codes.

Singleton bound: (3.3.1) implies that $d_{\min}/n \leq (n-k+1)/n \approx 1-R$ and therefore

$$R \;\leq\; 1 - \frac{d_{\min}}{n}. \tag{4.4.1}$$

Hamming bound: (3.3.2) implies that $1 - R \geq n^{-1} \log_2 \sum_{r=0}^{t} \binom{n}{r}$. According to Theorem A.1, the right side of this inequality converges to the binary entropy function by using $\lambda = t/n \approx d_{\min}/(2n)$:

$$R \;\leq\; 1 - H_2\left(\frac{d_{\min}}{2n}\right). \tag{4.4.2}$$

Plotkin bound: (3.3.4) merely implies that $1/2 \geq d_{\min}/n$. The derivation of the asymptotic form requires some additional considerations. Let $B(n,d) = 2^k$ be the maximum cardinality of a binary linear code with the block length n and the minimum distance d. First we need a proposition. For $d < n$

$$B(n,d) \leq 2 \cdot B(n-1,d). \tag{4.4.3}$$

Proof of this proposition. Let $\mathcal{C}$ be an $(n,k,d)_2$ code. The set $\mathcal{C}' = \{\boldsymbol{a} \in \mathcal{C} | a_{n-1} = 0\}$ of all codewords in $\mathcal{C}$ whose last symbol is 0 forms a subcode of $\mathcal{C}$ with a minimum distance of $d' \geq d$. It is fairly simple to prove that either $\mathcal{C}' = \mathcal{C}$ with $k' = k$ or $\mathcal{C}' \subset \mathcal{C}$ with $k' = k-1$ (see Problem 3.2). By eliminating the last component in $\mathcal{C}'$ an $(n-1,k',d')_2$ code $\mathcal{C}''$ is created. If $d' > d$, then $\mathcal{C}''$ can be degraded such that $d' = d$ applies. So, for each code $\mathcal{C}$ we can create a code $\mathcal{C}''$ of reduced block length with $|\mathcal{C}| = 2^k \leq 2 \cdot 2^{k'} = 2 \cdot |\mathcal{C}''| \leq 2B(n-1,d)$, completing the proof of the proposition. ■

Repeated application of (3.4.3) on an $(n,k,d)_2$ code with $n \geq 2d-1$ leads to:

$$\begin{aligned} 2^k &\leq B(n,d) \leq 2 \cdot B(n-1,d) \leq 4 \cdot B(n-2,d) \\ &\leq \ldots \;\; \leq 2^{n-(2d-1)} \cdot \underbrace{B(2d-1,d)}_{\leq 2d} \\ &\leq d \cdot 2^{n-2d+2}, \end{aligned}$$

because $2^k \leq 2d$ for a $(2d-1,k,d)_2$ code according to Theorem 3.11. By taking the logarithm, $k \leq \log_2 d + n - 2d + 2$, thus as $n \to \infty$ we finally obtain the result

$$R \;\leq\; 1 - 2\frac{d_{\min}}{n}. \tag{4.4.4}$$

Elias bound: This upper bound is tighter than any of the other upper bounds mentioned above. We will only quote the result without a proof [17, 105]

$$R \;\leq\; 1 - H_2\left(\frac{1-\sqrt{1-2\dfrac{d_{\min}}{n}}}{2}\right). \tag{4.4.5}$$

Gilbert-Varshamov bound: With (3.3.5) and by taking the logarithm of Theorem A.1 we obtain

$$n - k = \log_2 \sum_{r=0}^{d_{\min}-2} \binom{n-1}{r} \approx nH_2\left(\frac{d_{\min}-2}{n-1}\right) \approx nH_2\left(\frac{d_{\min}}{n}\right),$$

directly implying the asymptotic form of the lower bound

$$R \;\geq\; 1 - H_2\left(\frac{d_{\min}}{n}\right). \tag{4.4.6}$$

The various asymptotic bounds are displayed in Figure 3.4 for comparison where the code rate R is displayed as a function of the distance rate $d_{\min}/n$. All codes are asymptotically below the upper bounds, thus in particular below the Elias bound. There are also at least some good codes above the lower Gilbert-Varshamov bound, i.e., inside the hatched area. All codes below the hatched area are considered to be bad. Considering larger code rates the Hamming bound turns out to be nearly equal to the Elias bound. The asymptotic Singleton bound is fairly useless for $q = 2$, however, important codes attaining the Singleton bound for $q > 2$ are RS codes to be introduced in Chapter 8. The asymptotic properties of binary BCH codes are discussed in Subsection 8.2.4 and compared to the asymptotic bounds in Figure 8.16.

4.4.2 Asymptotically Good Codes

According to the Gilbert-Varshamov bound, at least the existence of so-called *families of asymptotically good codes* (n_s, k_s, d_s) with

$$\lim_{s\to\infty} \frac{k_s}{n_s} > 0 \quad \text{and} \quad \lim_{s\to\infty} \frac{d_s}{n_s} > 0 \tag{4.4.7}$$

is guaranteed. At first sight, this seems obvious and hardly meaningful, however, all known code families (apart from concatenated systems) do not have this property and are therefore *asymptotically bad*, which we will soon show. The explanation for this amazing fact has already been given in the discussion of the channel coding theorem in Section 2.2.

How is it possible that there are codes which attain the Singleton bound (MDS codes) or the Hamming bound (perfect codes), although the lower Elias

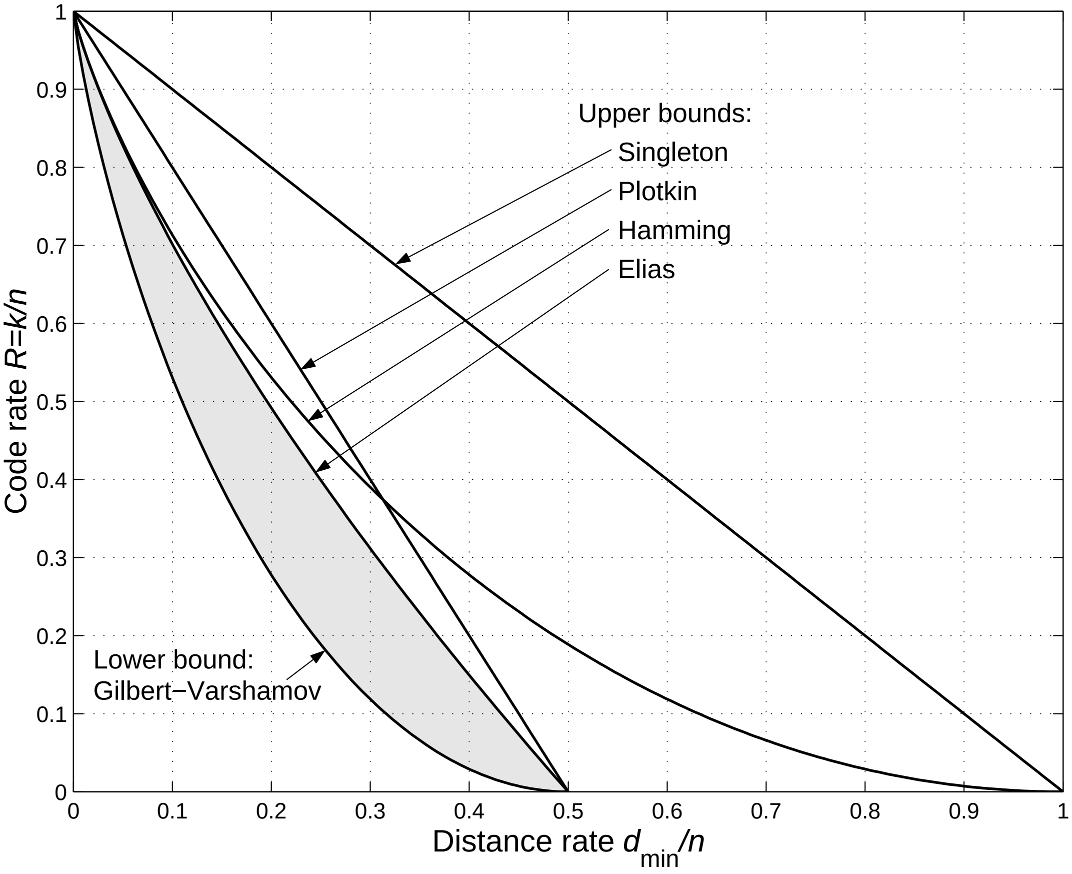

Figure 4.6. Asymptotic Bounds

bound seems to exclude the existence of such codes? As we have already mentioned, the cause is that in all known code families either the code rate or the distance rate converges to zero, which can be seen in the following examples.

MDS codes: The most important example for MDS codes are the RS codes according to Definition 7.1 which, assuming a fixed code rate, are of the form

$$(n, k, d_{\min})_q = (q-1, R(q-1), (1-R)(q-1)+1)_q.$$

We observe that $d_{\min}/n \to 1-R$ as $n \to \infty$, however, this is linked with $q \to \infty$. In the binary interpretation the minimum distance stays unchanged (see Chapter 7), thus

$$(n, k, d_{\min})_2 = ((q-1)\log_2 q, R(q-1)\log_2 q, (1-R)(q-1)+1)_2$$

with $d_{\min}/n \to 0$.

Perfect Codes: Asymptotically, the short Golay codes are of no interest. According to Theorem 4.10, Hamming codes are of the form

$$(n, k, d_{\min})_2 = (2^r-1, 2^r-r-1, 3)_2,$$

therefore $R \to 1$ and $d_{\min}/n \to 0$ as $r \to \infty$. According to Theorem 3.10, repetition codes have the form

$$(n, k, d_{\min})_2 = (2n+1, 1, 2n+1)_2$$

with $R \to 0$ and $d_{\min}/n = 1$ as $n \to \infty$.

The next considerations are associated with the proof of the noisy channel coding theorem from Section 3.x. Presume a BSC with the bit error probability p_e and BMD decoding according to Definition 3.6, thus error correction is only performed if there are less than $d_{\min}/2$ errors. At least the majority of all error patterns must be corrected to ensure that $P_w \to 0$ for the word error probability as $n \to \infty$ for the block length. The average number of errors within an error pattern is np_e, therefore $np_e < d_{\min}/2$ is required. This is equivalent to $p_e < d_{\min}/(2n)$, so

$$R_{\text{Hamming}} = 1 - H_2\left(\frac{d_{\min}}{2n}\right) \quad < \quad 1 - H_2(p_e) = C.$$

Since $R \le R_{\text{Elias}} \le R_{\text{Hamming}}$ for the asymptotic bounds, clearly $R \approx C$ for $P_w \to 0$ is not achievable. Thus the channel capacity can not be approached by using BMD decoding, thus BMD is significantly worse than the BDD and the MLD methods.

Conclusively, asymptotically good codes should enable an error correction over half the minimum distance with an efficient decoding scheme (as long as ideal maximum-likelihood decoding is impractical). When searching for asymptotically good codes this code property is as important as the maximization of the minimum distance.

A further simple consideration leads us to the fact that for most error patterns asymptotically good codes even correct $d_{\min}$ instead of $d_{\min}/2$ errors. On the one hand, for $R \approx C = 1 - H_2(p_e)$ the average number np_e of errors is corrected. On the other hand, we observe that $R = 1 - H_2(d_{\min}/n)$ according to the next theorem. The comparison leads to $p_e = d_{\min}/n$ or $d_{\min} = np_e$, thus for most error patterns even $d_{\min}$ instead of $d_{\min}/2$ errors are corrected.

4.4.3 Random Codes and the Gilbert-Varshamov Bound

Now, let us recollect the random coding argument as used for the proof of the noisy channel coding theorem. It is only possible to prove that $P_w \to 0$ as $n \to \infty$ for the mean value over all randomly chosen codes, but not for an individual code. This averaging technique can also be applied to the distance rate. The next theorem will state that for most codes the distance rate is close to the Gilbert-Varshamov bound, thus nearly half the codes are better than this bound. Only few codes greatly divert from the bound, i.e., only few codes have a significantly better or worse distance rate [105, 213].

Theorem 4.13. *For randomly chosen binary codes the averaged distance rate $d_{\min}/n$ asymptotically satisfies the Gilbert-Varshamov bound:*

$$R \;=\; 1 - H_2\left(\lim_{n\to\infty} \frac{E(d_{\min})}{n}\right). \tag{4.4.8}$$

Proof. It is to be shown that the cumulative distribution function of the distance rate converges to a unit step function as $n \to \infty$, where the position λ of the step is defined by $H_2(\lambda) = 1 - R$. To do this, the codewords are chosen randomly and statistically independent which, of course, implies a non-linear code. For $0 < \lambda < 1/2$ and $\boldsymbol{a}, \boldsymbol{b} \in \mathcal{C}$,

$$\begin{aligned} P\left(\frac{d_{\min}}{n} \geq \lambda\right) &= P\Big(d_H(\boldsymbol{a},\boldsymbol{b}) \geq \lambda n \;\text{ for all }\; \boldsymbol{b} \in \mathcal{C}\backslash\{\boldsymbol{a}\}\Big) \\ &= \prod_{\boldsymbol{b}\in\mathcal{C}\backslash\{\boldsymbol{a}\}} P(d_H(\boldsymbol{a},\boldsymbol{b}) \geq \lambda n) \\ &= \prod_{\boldsymbol{b}\in\mathcal{C}\backslash\{\boldsymbol{a}\}} P(\boldsymbol{a} \notin K_{\lambda n}(\boldsymbol{b})) \\ &= \prod_{\boldsymbol{b}\in\mathcal{C}\backslash\{\boldsymbol{a}\}} \big[1 - 2^{-n}|K_{\lambda n}(\boldsymbol{b})|\big]\,. \end{aligned}$$

The last equality results from $P(\boldsymbol{a} \in \mathcal{M}) = 2^{-n}|\mathcal{M}|$ as in (2.7.6). We use the notation $s_n = \sum_{i=0}^{\lambda n}\binom{n}{i} = |K_{\lambda n}(\boldsymbol{b})|$, implying that

$$P\left(\frac{d_{\min}}{n} \geq \lambda\right) = \Big[1 - 2^{-n}s_n\Big]^{2^k-1}.$$

By taking the logarithm,

$$\begin{aligned} \ln P\left(\frac{d_{\min}}{n} \geq \lambda\right) &= (2^k-1)\cdot\ln\Big(1 - \frac{s_n}{2^n}\Big) \\ &= (2^k-1)\frac{s_n}{2^n}\cdot\ln\left(\left[1 - \frac{1}{2^n/s_n}\right]^{2^n/s_n}\right). \end{aligned}$$

According to Theorem A.1,

$$\begin{aligned} \frac{2^n}{s_n} \;&\geq\; \frac{2^n}{2^{nH_2(\lambda)}} \;=\; 2^{n(1-H_2(\lambda))} \;\to\; \infty, \\ \frac{2^k-1}{2^n}s_n \;&\approx\; 2^{n(R-1+n^{-1}\log_2 s_n)} \;\to\; 2^{n(R-1+H_2(\lambda))} \end{aligned}$$

as $n \to \infty$. It follows that

$$\begin{aligned} \lim_{n\to\infty} \ln P\left(\frac{d_{\min}}{n} \geq \lambda\right) &= 2^{n(R-1+H_2(\lambda)}\cdot\ln(e^{-1}) \\ &= \left\{\begin{array}{ll} -2^{-\infty} & \text{for } R < 1 - H_2(\lambda) \\ -2^{\infty} & \text{for } R > 1 - H_2(\lambda) \end{array}\right\}, \end{aligned}$$

therefore

$$\lim_{n\to\infty} P\left(\frac{d_{\min}}{n} \geq \lambda\right) = \left\{\begin{array}{ll} 1 & \text{for } R < 1 - H_2(\lambda) \\ 0 & \text{for } R > 1 - H_2(\lambda) \end{array}\right\}. \tag{4.4.9}$$

Hence, for nearly all codes, $d_{\min}/n \geq \lambda$ for $R < 1 - H_2(\lambda)$ and $d_{\min}/n < \lambda$ for $R > 1 - H_2(\lambda)$. Therefore we can usually expect $d_{\min}/n = \lambda$ for $R = 1 - H_2(\lambda)$. A proof for linear codes can be found in [213]. ■

This theorem emphasizes the power of random codes, although as discussed in Section 2.2 there is no constructive method to achieve the Gilbert-Varshamov bound, at least not in the binary case. Yet, there is actually a certain inversion to Theorem 4.13 which we will give here refraining from all confusing mathematical details [155, 161]: if a code is no longer really random, but contains certain regular structures allowing a more compact description than a simple enumeration of the codewords, then on average the Gilbert-Varshamov bound can not be achieved.

4.5 The Weight Distribution

The minimum distance is the most important parameter of a block code, but for calculating the error probability and for some other applications, more information about the code properties as contained in the so-called weight distribution is required.

4.5.1 Definitions

Definition 4.7. *The* weight distribution *of a linear $(n, k, d_{\min})_q$ block code $\mathcal{C}$ is a vector with the parameters $A_0, \ldots, A_n$ where A_r denotes the number of codewords of Hamming weight r. The polynomial in Z with the A_r as coefficients is called the* weight enumerator *of the code $\mathcal{C}$:*

$$A(Z) = \sum_{r=0}^{n} A_r Z^r = \sum_{\boldsymbol{a}\in\mathcal{C}} Z^{w_H(\boldsymbol{a})}. \tag{4.5.1}$$

Another often used form of the weight enumerator is

$$W(X,Y) = \sum_{r=0}^{n} A_r X^{n-r} Y^r = \sum_{\boldsymbol{a}\in\mathcal{C}} X^{n-w_H(\boldsymbol{a})} Y^{w_H(\boldsymbol{a})}. \tag{4.5.2}$$

The indeterminates X, Y and Z are only formal placeholders. Computations with the weight enumerator always use integers – independent of the underlying Galois field $\mathbb{F}_q$ for the information and code symbols. The equalities in (3.5.1) and (3.5.2) are obvious. The two forms of weight enumerators are connected as follows:

$$A(Z) = W(1, Z), \qquad W(X,Y) = X^n A\left(\frac{Y}{X}\right). \tag{4.5.3}$$

The following properties are obvious:

$$A_0 = A(0) = 1, \quad A_n \le (q-1)^n, \tag{4.5.4}$$

$$A_r = 0 \;\text{ for }\; 0 < r < d_{\min}, \tag{4.5.5}$$

$$\sum_{r=0}^{n} A_r = A(1) = q^k. \tag{4.5.6}$$

For some codes the weight distribution is symmetric which can be equivalently described by the weight enumerator:

$$\begin{aligned} A_r = A_{n-r} \text{ for all } r \quad &\Longleftrightarrow \quad A(Z) = Z^n \cdot A(Z^{-1}) \\ &\Longleftrightarrow \quad W(X,Y) = W(Y,X). \end{aligned} \tag{4.5.7}$$

The weight distribution can be calculated in a closed form for few codes only, some of which are the Hamming and simplex codes (see Theorem 5.10 and (5.4.5)) and the MDS codes (see Theorem 8.3). Equivalent codes have identical weight distributions, since a permutation of codeword components does not change the weights.

Example 4.8. (1) With simple enumeration of the $(7,4,3)_2$ Hamming code in Example 1.2, $A_0 = A_7 = 1$ and $A_3 = A_4 = 7$. The weight distribution is symmetric with $A(Z) = 1 + Z^7 + 7(Z^3 + Z^4) = Z^7 \cdot A(Z^{-1})$.

(2) For the $(n,1,n)_2$ repetition code, it is obvious that $A(Z) = 1 + Z^n$.

(3) Consider the $(n, n-1, 2)_2$ parity-check code. The 2^{n-1} information words of length $k = n-1$ have a binomial weight distribution, i.e., there are $\binom{n-1}{r}$ information words of weight r. When attaching the parity-check bit, information words of an even weight are extended by a zero (the weight is unchanged) and information words of an odd weight are extended by a one (the weight is increased to the even value). Therefore $A_{2r} = \binom{n-1}{2r} + \binom{n-1}{2r-1} = \binom{n}{2r}$ and $A_{2r-1} = 0$ or

$$A_r = \left\{ \begin{array}{cl} \dbinom{n}{r} & \text{if } r \text{ is even} \\ 0 & \text{if } r \text{ is odd} \end{array} \right\}. \tag{4.5.8}$$

The property (3.5.6) can be easily verified with (A.2.2). ■

The weight distribution of some codes can be roughly approximated by the binomial distribution [83], i.e., for the binary case,

$$A_r \;\approx\; 2^{k-n} \binom{n}{r}. \tag{4.5.9}$$

However, for $A_{d_{\min}}$ this approximation is usually not suitable and (3.5.5) is not fulfilled. The preceding examples clearly show the limited applicability of the binomial approximation.

4.5.2 Random Codes

We will now consider a binary *random code* again (as in the proof of the channel coding theorem and as in Subsection 4.4.3), in which the encoded bits in all codewords are chosen randomly and statistically independent and where 0 and 1 turn up with an equal probability of 50%, respectively. Then the weights of the codewords have an exact binomial distribution according to (A.3.5) with $\epsilon = 0.5$. This distribution also occurs for the error pattern weight of a BSC with $p_e = 0.5$ according to (1.3.9). For a random code with the above definitions, (3.5.9) is exactly valid for the expected values of the weight distribution, so

$$E(A_r) \;=\; 2^{k-n}\binom{n}{r}. \tag{4.5.10}$$

However, this random code is not linear and the codewords may even be identical. Random codes with definitely different codewords and systematic random codes are discussed in [179].

In the case of linear random codes we will prove in Theorem 5.2 that the average weight distribution has the exact form

$$E(A_r) = \left\{ \begin{array}{ll} 1 & \text{for } r = 0 \\ 2^{k-n}\left[\binom{n}{r} - \binom{n-k}{r}\right] & \text{for } 1 \le r \le n-k \\ 2^{k-n}\binom{n}{r} & \text{for } n-k < r \le n \end{array} \right\}. \tag{4.5.11}$$

The relations (3.5.4) to (3.5.6) can easily be verified.

4.6 Error-Detection Performance

In this section we will exclusively consider error-detection codes. The following section will discuss the computation of the actual error probability which is always related to error-correction codes.

The probability of an undetected error pattern can be exactly calculated with the help of the weight distribution, however, the result can still have surprising properties:

Theorem 4.14 (Error Detection). *Let a linear $(n, k, d_{\min})_q$ code have the weight distribution $A_0, \ldots, A_n \;\leftrightarrow\; A(Z)$. For the transmission via the q-ary symmetric hard-decision DMC with the symbol-error probability p_e, the probability P_{ue} of an undetected error pattern (also called undetected word-error proba-*

bility) can be calculated exactly as

$$P_{ue} = P(\boldsymbol{e} \in \mathcal{C}\backslash\{\boldsymbol{0}\}) = \sum_{r=d_{\min}}^{n} A_r \left(\frac{p_e}{q-1}\right)^r (1-p_e)^{n-r} \tag{4.6.1}$$

$$= (1-p_e)^n \left[A\left(\frac{p_e}{(q-1)(1-p_e)}\right) - 1 \right].$$

For small p_e *and* $q = 2$*,*

$$P_{ue} \approx \sum_{r=d_{\min}}^{n} A_r \cdot p_e^r = A(p_e) - 1 \tag{4.6.2}$$

$$\approx A_{d_{\min}} \cdot p_e^{d_{\min}}. \tag{4.6.3}$$

In the binary case P_{ue} *typically approaches its maximum at* $p_e = 1/2$ *and is exponentially limited by the number of parity-check bits:*

$$P_{ue} \;\leq\; P_{ue}(p_e = 1/2) \;=\; \frac{2^k - 1}{2^n} \;\leq\; 2^{-(n-k)}. \tag{4.6.4}$$

However, one must be cautious since there are also so-called improper codes *with* $P_{ue} \gg 2^{-(n-k)}$ *for* $p_e \ll 1/2$*. Therefore (3.6.4) is sometimes also called the* fallacious upper bound *[205].*

Proof. (1) The equality in (3.6.1) is directly implied by

$$A\left(\frac{p_e}{(q-1)(1-p_e)}\right) = 1 + \sum_{r=1}^{n} A_r \left(\frac{p_e}{q-1}\right)^r (1-p_e)^{-r}.$$

It does not matter whether the sum starts at $r = 1$ or $r = d_{\min}$. For a small p_e, (3.6.2) and (3.6.3) follow directly from (3.6.1), and (3.6.4) follows directly from (3.6.1) with the help of (3.5.6).

(2) Proof of (3.6.1). Let $\boldsymbol{a}_\nu = (a_{\nu,0}, \ldots, a_{\nu,n-1})$ with $0 \leq \nu \leq q^k - 1$ be an enumeration of the codewords with $\boldsymbol{a}_0 = \boldsymbol{0}$. Then

$$P_{ue} = P(\boldsymbol{e} \in \mathcal{C}\backslash\{\boldsymbol{0}\}) = \sum_{\nu=1}^{q^k-1} P(\boldsymbol{e} = \boldsymbol{a}_\nu).$$

Obviously $\boldsymbol{e} = \boldsymbol{a}_\nu$ means that

$$\begin{array}{llll} e_j = a_{\nu,j} = 0 & \text{with} \;\; P(e_j = 0) = 1 - p_e & \text{in} \;\; n - w_H(\boldsymbol{a}_\nu) & \text{positions,} \\ e_j = a_{\nu,j} \neq 0 & \text{with} \;\; P(e_j \neq 0) = p_e & \text{in} \;\; w_H(\boldsymbol{a}_\nu) & \text{positions.} \end{array}$$

The probability that e_j will take on a specific non-zero value is $p_e/(q-1)$. Therefore, similar to (3.5.1) and (3.5.2),

$$P_{ue} = \sum_{\nu=1}^{q^k-1} (1-p_e)^{n-w_H(\boldsymbol{a}_\nu)} \left(\frac{p_e}{q-1}\right)^{w_H(\boldsymbol{a}_\nu)}$$

$$= \sum_{r=1}^{n} A_r \cdot (1-p_e)^{n-r} \left(\frac{p_e}{q-1}\right)^r,$$

where $\nu = 0$ and $r = 0$ correspond to the all-zero word. ■

Example 4.9. In the following, the error-detection performance is computed for six different codes and shown in the Figures 3.5a,b.

(1) For the $(n, 1, n)_2$ repetition code the definition of P_{ue} implies that $P_{ue} = P(\boldsymbol{e} = 11\ldots1) = p_e^n$. With $A(Z) = 1 + Z^n$ we get the same result

$$P_{ue} = (1-p_e)^n \left(A\left(\frac{p_e}{1-p_e}\right) - 1 \right) = (1-p_e)^n \frac{p_e^n}{(1-p_e)^n} = p_e^n.$$

(2) For the $(7, 3, 4)_2$ simplex code introduced in the next chapter (see Definition 4.5 and (4.4.3)), the weight distribution gives us the result of

$$P_{ue} = (1-p_e)^7 \left(1 + 7\left(\frac{p_e}{1-p_e}\right)^4 - 1 \right) = 7(1-p_e)^3 p_e^4.$$

(3) For the $(7, 4, 3)_2$ Hamming code in Example 3.8(1)

$$P_{ue} = 7p_e^3(1-p_e)^4 + 7p_e^4(1-p_e)^3 + p_e^7 = 7p_e^3 - 21p_e^4 + 21p_e^5 - 7p_e^6 + p_e^7.$$

(4) For the $(n, n-1, 2)_2$ parity-check code with $p_e = 1/2$ and (4.6.4) it can be shown that $P_{ue} \approx 1/2$. According to (3.5.8),

$$P_{ue} = \sum_{r=1}^{\lfloor n/2 \rfloor} \binom{n}{2r} p_e^{2r} (1-p_e)^{n-2r}.$$

For a small p_e, $P_{ue} \approx \dfrac{n(n-1)}{2} p_e^2$.

(5) Let us now consider a $(2k, k, 1)_2$ code which is intentionally poorly chosen by setting all parity-check bits to zero in order to obtain a simple and clearly arranged example of an improper code. Thus the weight distribution is not affected by the parity-checks, $A_r = \binom{k}{r}$ for $0 \leq r \leq k$ and $A_r = 0$ for $r > k$, hence $A(Z) = \sum_r \binom{k}{r} Z^r = (Z+1)^k$ according to (A.2.2). For the BSC with the bit-error probability p_e, according to (3.6.1),

$$P_{ue} = (1-p_e)^{2k} \left(\left(\frac{p_e}{1-p_e} + 1 \right)^k - 1 \right) = (1-p_e)^k \left(1 - (1-p_e)^k \right).$$

In particular for specific values of p_e,

$$P_{ue} = \left\{ \begin{array}{ll} 2^{-k}(1-2^{-k}) \approx 0 & \text{for} \quad p_e = 1/2 \\ 1/4 & \text{for} \quad p_e = 1 - 2^{-1/k} \approx 0 \end{array} \right\}.$$

Thus the probability of undetected errors is small for a high BSC error rate, but high for a small BSC error rate. Therefore this code violates the bound (3.6.4).

(6) The uncoded data transmission can formally be expressed by an $(n, n, 1)_2$ code. From (1.3.6) we know that for the BSC $P_{ue} = 1 - (1-p_e)^n$. The same result derives from $A_r = \binom{n}{r}$ and $A(Z) = (1+Z)^n$. ■

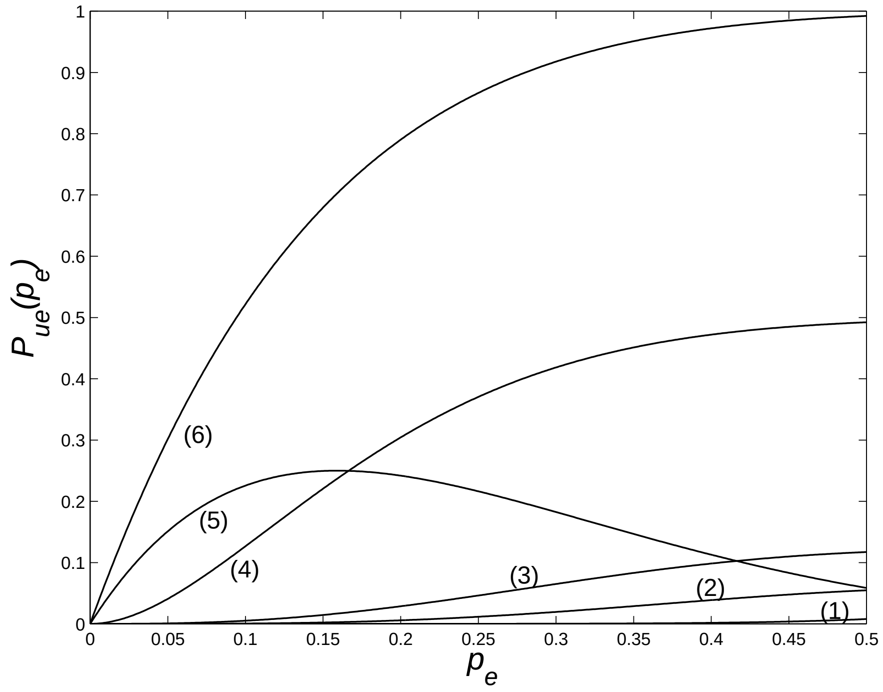

Figure 4.7a. Undetected error probability (linearly scaled axes)

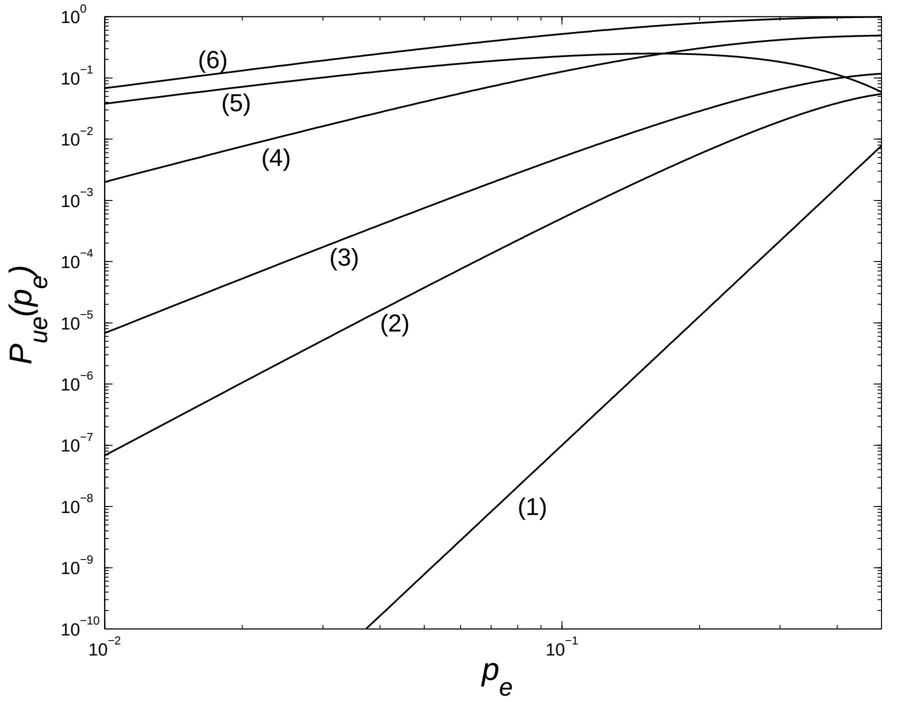

Figure 4.7b. Undetected error probability (both axes logarithmically scaled)

The Figures 3.5a and 3.5b show the performance results for the six codes discussed in Example 3.9, where the block length is always chosen as 7 or 8. The labeling of the graphs is as in Example 4.9: (1) = $(7,1,7)_2$ repetition code, (2) = $(7,3,4)_2$ simplex code, (3) = $(7,4,3)_2$ Hamming code, (4) = $(7,6,2)_2$ parity-check code, (5) = $(8,4,1)_2$ improper code, (6) = $(7,7,1)_2$ uncoded transmission. In both figures the same codes are used, however, in Figure 3.5a both axes are linearly scaled and in Figure 3.5b both axes are logarithmically scaled. From (1) to (6) the number of parity-check bits decreases, so the performance is decreased, too. The deviating behaviour of the improper code (5) becomes obvious, P_{ue} approaches its maximum at $p_e = 1 - 2^{-1/4} \approx 0.16$ according to Example 3.9(5).

Surprisingly, we can not derive generally valid statements from the seemingly simple formula (3.6.1). The bound in (3.6.4) is generally only valid for $p_e = 1/2$. In [200, 234] and in other publications various codes are examined as to whether they provide a smaller P_{ue} for a smaller p_e. Usually this is guaranteed for most codes, however, there remain some improper codes including the widely used $(63,24)_2$ BCH code (see Chapter 8) which actually does not satisfy the fallacious bound (3.6.4). For most applications the cyclic redundancy check (CRC) codes, discussed in Section 5.6, are used for error detection. Further widely-used error-detection codes, particularly the shortened Hamming codes, are covered in [180, 181]. The textbook [69] focuses on error-detection codes only.

4.7 Error-Correction Performance

As seen in the previous section there is a single formula for the undetected word-error probability which is based on the weight distribution of the code. However, for the word-error probability at the output of an error-correction decoder several cases have to be distinguished. An exact calculation is only possible for hard-decision BMD decoding, whereas in all other cases only bounds are available. These bounds are influenced by the channel properties as well as by the code properties as contained in the weight distribution. In the following subsections, always assuming channels with binary input, we will first discuss the hard-decision DMC, then the general DMC and finally the AWGN with ideal soft decisions.

4.7.1 Performance Bounds for Hard-Decision Decoding

For hard-decision decoding the received word $\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{e}$ will be interpreted as in (3.2.6) as the sum of the transmitted codeword $\boldsymbol{a}$ and an error word $\boldsymbol{e}$. The probability of an erroneous transmission (error-event probability)

$$P_{ee} = P(\boldsymbol{e} \neq \boldsymbol{0}) = P(\text{error pattern unequal to } 0) \tag{4.7.1}$$

was given in (1.3.6) and the undetected word-error probability

$$P_{ue} = P(\boldsymbol{e} \in \mathcal{C}\backslash\{\boldsymbol{0}\}) \tag{4.7.2}$$

was calculated in Section 3.6. Now, for error-correcting codes, we are interested in the word-error probability at decoder output (or word-error rate), according to (1.6.1),

$$P_w = P(\hat{\boldsymbol{a}} \neq \boldsymbol{a}) = P(\text{decoder error}). \tag{4.7.3}$$

In other words, P_w is the probability of the error patterns causing received words which are not or wrongly decoded. The word-error probability is expected to depend on the decoding method used.

Theorem 4.15 (Error Correction). *For a linear $(n, k, d_{\min})_q$ code with $t = \lfloor (d_{\min} - 1)/2 \rfloor$ and transmission over a q-ary symmetric hard-decision DMC with the symbol-error probability p_e, the word-error probability after maximum-likelihood decoding (MLD) is bounded above by*

$$P_w \leq 1 - \sum_{r=0}^{t} \binom{n}{r} p_e^r (1-p_e)^{n-r} = \sum_{r=t+1}^{n} \binom{n}{r} p_e^r (1-p_e)^{n-r}. \tag{4.7.4}$$

In particular, for bounded-minimum-distance decoding (BMD) as well as for perfect codes this bound is tight and thus P_w can be computed exactly. In general, the error probability P_{cs} for the q-ary information or encoded symbols is bounded above by

$$P_{cs} \leq \sum_{r=t+1}^{n} \min\left\{1, \frac{r+t}{k}\right\} \binom{n}{r} p_e^r (1-p_e)^{n-r}. \tag{4.7.5}$$

For a small p_e the bounds become very tight and provide useful approximations:

$$P_w \lessapprox \binom{n}{t+1} p_e^{t+1}, \qquad P_{cs} \lessapprox \min\left\{1, \frac{d_{\min}}{k}\right\} \cdot P_w. \tag{4.7.6}$$

Proof. Since the MLD method is better than the BMD method, we only need to prove the equality in (4.7.1) for the BMD. A correct BMD decoding takes place if and only if a maximum of t errors occurs:

$$\begin{aligned} P_w &= 1 - P(\text{correct decoding}) \\ &= 1 - P(w_H(\boldsymbol{e}) \leq t) = P(w_H(\boldsymbol{e}) \geq t+1) \\ &= 1 - \sum_{r=0}^{t} P(w_H(\boldsymbol{e}) = r) = \sum_{r=t+1}^{n} P(w_H(\boldsymbol{e}) = r). \end{aligned}$$

According to (1.3.12), the number of errors in a word of length n is binomially distributed:

$$P(w_H(\boldsymbol{e}) = r) = \binom{n}{r} p_e^r (1-p_e)^{n-r}.$$

Thus (3.7.1) is proven, the equality on the right side of (3.7.1) also derives directly from the binomial formula (A.2.2). For the q-ary symbols we have

$$\begin{aligned} P_{cs} &= \frac{1}{k} E(\text{number of } q\text{-ary symbol errors per decoded word}) \\ &= \frac{1}{k} \sum_{r=t+1}^{n} E(\text{number of symbol errors per decoded word} \mid w_H(\boldsymbol{e}) = r) \\ &\qquad\qquad \cdot P(w_H(\boldsymbol{e}) = r) \\ &\leq \frac{1}{k} \sum_{r=t+1}^{n} \min\{k, r+t\} \cdot P(w_H(\boldsymbol{e}) = r), \end{aligned}$$

since the number of q-ary symbol errors per word is limited to the number k of q-ary information symbols, though on the other hand limited to $r + t$, because for the BMD decoder

$$w_H(\boldsymbol{a}, \hat{\boldsymbol{a}}) \;\leq\; \underbrace{w_H(\boldsymbol{a}, \boldsymbol{y})}_{=\,r} + \underbrace{w_H(\boldsymbol{y}, \hat{\boldsymbol{a}})}_{\leq\, t}.$$

For a small p_e the first summand dominates the binomial sum on the right side of (3.7.1). For P_{cs}, we observe that $(r + t)/k = (2t + 1)/k = d_{\min}/k$. However, for a bigger p_e the error probability in (3.7.3) might be so low, that the upper bound could change into a lower bound. ■

To prove this theorem the ML decoding was degraded to BMD decoding. In this case the only influence is the minimum distance of the code, therefore the weight distribution of the code does not occur in Theorem 4.15. We write P_b instead of P_{cs} for the binary case with $q = 2$. The bit-error and word-error rates in Figures 1.10, 1.11, 4.5 as well as the RS and BCH graphs in Subsections 8.1.5 and 8.2.3 were computed with the results of Theorem 4.15.

The result (4.7.6) has already been used in Section 1.7 for the derivation of the asymptotic coding gain for hard-decision decoding, with the result that $G_{\text{a,hard}} = 10 \cdot \log_{10}(R(t+1))$ dB. With (4.7.6) the approximation for the relation between the bit and word-error probability is justified again.

One should pay attention when numerically evaluating the binomial sum in (3.7.1). For small values of p_e the left side approximately is of the form $1 - (1 - P_w) = P_w$ which requires a very high numerical resolution. These problems disappear when using the right side of (3.7.1).

Example 4.10. (1) According to Example 3.6(1), the $(7, 4, 3)_2$ Hamming code with $t = 1$ is perfect and therefore by using (3.7.1),

$$\begin{aligned} P_w &= 1 - \binom{7}{0} p_e^0 (1 - p_e)^7 - \binom{7}{1} p_e^1 (1 - p_e)^6 \\ &= 1 - (1 - p_e)^7 - 7p_e(1 - p_e)^6 \end{aligned}$$

$$
\begin{aligned}
&= 1 - (1 - 7p_e + 21p_e^2 - p_e^3 \ldots) - 7p_e(1 - 6p_e + p_e^2 \ldots) \\
&\approx 21p_e^2 = \binom{7}{2} p_e^2.
\end{aligned}
$$

This justifies the approximation in (3.7.3). The asymptotic coding gain is $G_{\text{a,hard}} = 10 \cdot \log_{10}(4/7 \cdot 2) = 0.6$ dB.

(2) According to Theorem 3.10, the $(n, 1, n)_2$ repetition code with $n = 2t+1$ is perfect and of course $P_b = P_w$. According to the left side of (3.7.3), we can approximate $P_b = P_w \approx \binom{2t+1}{t+1} p_e^{t+1}$ for a small p_e. The exact evaluation of (3.7.1) for the three cases

$$
p_e = \left\{ \begin{array}{ll} 0.00001 & \text{for } n = 1 \\ 0.0018 & \text{for } n = 3 \\ 0.010 & \text{for } n = 5 \end{array} \right\}
$$

always leads to $P_b = P_w = 10^{-5}$. Thus with a bigger block length or a smaller code rate a bad channel can be compensated for. If the BSC emerges from a binary quantized AWGN with $p_e = Q(\sqrt{2RE_b/N_0})$, then the repetition code turns out to be bad, since

$$
\frac{E_b}{N_0} = \left\{ \begin{array}{rl} 9.6 \text{ dB} & \text{for } n = 1 \\ 11.0 \text{ dB} & \text{for } n = 3 \\ 11.3 \text{ dB} & \text{for } n = 5 \end{array} \right\}
$$

is required for $P_b = P_w = 10^{-5}$. Hence, encoding causes degradations or costs signal power instead of saving signal power. Correspondingly, the asymptotic coding gain with $G_{\text{a,hard}} = -1.8$ dB for $n = 3$ or -2.2 dB for $n = 5$ turns out to be negative. Thus trivial codes like the repetition code prove to be useless or even disadvantageous. ■

4.7.2 Performance Bounds for the General DMC Based on the Union Bound

Now, we will consider the general DMC with binary input ($q = 2$). The error probability for block codes with ML decoding turns out to be dependent on the

- *code properties*, given by the weight distribution (see Definition 3.7) and the
- *channel properties*, given by the Bhattacharyya bound γ (see Definition 2.4) evaluated as

$$
\gamma = \left\{ \begin{array}{ll} \sqrt{4p_e(1-p_e)} & \text{BSC} \\ e^{-E_c/N_0} & \text{binary AWGN channel} \end{array} \right\}
$$

for our two standard channels according to (3.2.13) and (3.2.15).

In contrast to hard-decision decoding, the error probability now depends on the complete weight distribution of the code and we can only derive upper bounds for the word-error probability:

Theorem 4.16 (Union Bound for the General DMC). *For a linear $(n, k, d_{\min})_2$ code with the weight distribution $A_0, \ldots, A_n \leftrightarrow A(Z)$ and transmission over the DMC with the Bhattacharyya bound γ, the word-error probability after maximum-likelihood decoding (MLD) is bounded by*

$$P_w \leq \sum_{r=d_{\min}}^{n} A_r \gamma^r \ = \ A(\gamma) - 1. \tag{4.7.7}$$

For good channels with a small γ, P_w is approximately bounded by the dominant term in the summation,

$$P_w \lessapprox A_{d_{\min}} \cdot \gamma^{d_{\min}}. \tag{4.7.8}$$

A general weak upper bound without knowledge of the weight distribution is given by

$$P_w \ \leq \ (2^k - 1) \cdot \gamma^{d_{\min}}. \tag{4.7.9}$$

For both standard channels, the general bound is

$$P_w \leq \left\{ \begin{array}{ll} \displaystyle\sum_{r=d_{\min}}^{n} A_r \cdot \sqrt{4p_e(1-p_e)}^{\,r} & \textit{BSC} \\ \displaystyle\sum_{r=d_{\min}}^{n} A_r \cdot e^{-r \cdot E_c/N_0} & \textit{binary AWGN channel} \end{array} \right\}. \tag{4.7.10}$$

Proof. The idea is similar to the proof of the R_0 theorem in Section 2.8, though this time we will give a detailed derivation to make the method of the union bound clear. Let $\mathcal{C} = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{2^k}\}$ be an enumeration of the code. With the MLD rule the decoder is defined for all received words so that a wrong decoding is equivalent to a wrong estimation for the codeword. The decision region of $\boldsymbol{a}_i$ is

$$\mathcal{M}_i = \Big\{ \boldsymbol{y} \ \Big| \ P(\boldsymbol{y}|\boldsymbol{a}_i) \geq P(\boldsymbol{y}|\boldsymbol{b}) \text{ for all } \boldsymbol{b} \in \mathcal{C} \Big\}.$$

The complementary set is composed of a union of non-disjoint sets as follows,

$$\begin{aligned} \overline{\mathcal{M}_i} = \mathcal{A}_{\text{out}}^n \backslash \mathcal{M}_i &= \Big\{ \boldsymbol{y} \ \Big| \ \text{there exists } j \neq i \text{ with } P(\boldsymbol{y}|\boldsymbol{a}_j) > P(\boldsymbol{y}|\boldsymbol{a}_i) \Big\} \\ &= \bigcup_{\substack{j=1 \\ j \neq i}}^{2^k} \underbrace{\Big\{ \boldsymbol{y} \ \Big| \ P(\boldsymbol{y}|\boldsymbol{a}_j) > P(\boldsymbol{y}|\boldsymbol{a}_i) \Big\}}_{= \mathcal{M}_{i,j}}. \end{aligned}$$

We presume the codeword $\boldsymbol{a}_i$ has been transmitted. The first step is to derive an upper bound for the probability P_i of the ML decoder deciding on one of the other $2^k - 1$ codewords $\boldsymbol{a}_j$. This is achieved by using the union bound method (A.3.3) for the sum of the probabilities $P(\boldsymbol{y} \in \mathcal{M}_{i,j} | \boldsymbol{a}_i)$:

$$\begin{aligned} P_i &= P(\text{decoding error} \mid \boldsymbol{a}_i \text{ transmitted}) \\ &= P(\boldsymbol{y} \notin \mathcal{M}_i \mid \boldsymbol{a}_i \text{ transmitted}) \\ &= P\left(\boldsymbol{y} \in \bigcup_{\substack{j=1 \\ j\neq i}}^{2^k} \mathcal{M}_{i,j} \,\middle|\, \boldsymbol{a}_i \text{ transmitted} \right) \\ &\leq \sum_{\substack{j=1 \\ j\neq i}}^{2^k} P(\boldsymbol{y} \in \mathcal{M}_{i,j} \mid \boldsymbol{a}_i \text{ transmitted}). \end{aligned}$$

The probabilities $P(\boldsymbol{y} \in \mathcal{M}_{i,j} | \boldsymbol{a}_i)$ are the error probabilities for a code only consisting of the two codewords $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$. For these so-called *2-codeword-error probabilities* the denotation $P(\boldsymbol{a}_i \to \boldsymbol{a}_j)$ is often used, since a decoding error at this point means the decision on $\boldsymbol{a}_j$ instead of $\boldsymbol{a}_i$. The second step is to derive an upper bound for the 2-codeword-error probability by using the Bhattacharyya bound (an exact calculation without approximations will be given in Theorem 3.17 for the special case of the AWGN channel):

$$\begin{aligned} P(\boldsymbol{y} \in \mathcal{M}_{i,j} | \boldsymbol{a}_i) &= \sum_{\boldsymbol{y} \in \mathcal{M}_{i,j}} P(\boldsymbol{y}|\boldsymbol{a}_i) \\ &\leq \sum_{\boldsymbol{y} \in \mathcal{M}_{i,j}} P(\boldsymbol{y}|\boldsymbol{a}_i) \sqrt{\frac{P(\boldsymbol{y}|\boldsymbol{a}_j)}{P(\boldsymbol{y}|\boldsymbol{a}_i)}} \\ &= \sum_{\boldsymbol{y} \in \mathcal{M}_{i,j}} \sqrt{P(\boldsymbol{y}|\boldsymbol{a}_i) P(\boldsymbol{y}|\boldsymbol{a}_j)} \\ &\leq \sum_{\boldsymbol{y} \in \mathcal{A}_{\text{out}}^n} \sqrt{P(\boldsymbol{y}|\boldsymbol{a}_i) P(\boldsymbol{y}|\boldsymbol{a}_j)} \\ &= \sum_{\boldsymbol{y} \in \mathcal{A}_{\text{out}}^n} \prod_{r=0}^{n-1} \sqrt{P(y_r|a_{i,r}) P(y_r|a_{j,r})} \quad \text{according to (1.3.2)} \\ &= \prod_{r=0}^{n-1} \underbrace{\sum_{y \in \mathcal{A}_{\text{out}}} \sqrt{P(y|a_{i,r}) P(y|a_{j,r})}}_{= J(a_{i,r}, a_{j,r})} \quad \text{according to Lemma 2.1.} \end{aligned}$$

According to Definition 2.4, $J(a_{i,r}, a_{j,r}) = 1$ for $a_{i,r} = a_{j,r}$ and $J(a_{i,r}, a_{j,r}) = \gamma$ for $a_{i,r} \neq a_{j,r}$ is valid. Therefore $\prod_r J(a_{i,r}, a_{j,r}) = \gamma^{d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)}$ and thus

$$P_i \leq \sum_{\substack{j=1 \\ j\neq i}}^{2^k} \gamma^{d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)} = \sum_{j=1}^{2^k} \gamma^{d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)} - 1.$$

The distance distribution can be replaced by the weight distribution and then similar to Definition 3.7, we have

$$P_i \leq \sum_{\boldsymbol{a} \in \mathcal{C}} \gamma^{w_H(\boldsymbol{a})} - 1 = \sum_{r=d_{\min}}^{n} A_r \gamma^r.$$

This upper bound for P_i is independent of i, therefore, as in the proofs in Sections 2.7 and 2.8, we finally obtain the main result (3.8.1):

$$P_w = P(\text{decoding error}) = \sum_{i=1}^{2^k} P_i \cdot P(\boldsymbol{a}_i \text{ transmitted}) \leq \sum_{r=d_{\min}}^{n} A_r \gamma^r.$$

Due to $\gamma \leq 1$ the general weak upper bound (3.8.3) follows directly from (3.8.1). The other approximations (3.8.2) and (3.8.4) are also obvious. ■

The union bound is only useful for good channels with a small γ, because the bound for bad channels can become greater than 1. For a sufficiently small γ, $A_{d_{\min}} \gamma^{d_{\min}}$ is larger than $A_r \gamma^r$ for $r > d_{\min}$ irrespective of the weight distribution. So asymptotically for $\gamma \to 0$ the higher coefficients of the complete path enumerator and even $A_{d_{\min}}$ are meaningless.

According to (3.8.4), $P_w \approx \text{const} \cdot p_e^{d_{\min}/2}$ for the BSC. For an odd $d_{\min}$, $t = (d_{\min} - 1)/2$ and therefore $t + 1 > d_{\min}/2$. Thus for hard-decision decoding Theorem 3.15 is tighter than the union bound, which therefore turns out to be asymptotically not exact. Considering the general applicability of the union bound, this is not surprising.

4.7.3 Performance Bounds for the AWGN Channel with Soft-Decision Decoding

For the AWGN channel with binary input and ideal soft-decision decoding the union bound can be tightened to:

Theorem 4.17 (AWGN). *For a linear $(n, k, d_{\min})_2$ code with the weight distribution $A_0, \ldots, A_n$ and transmission over the binary AWGN channel with ideal soft decision, the word-error probability P_w is bounded by*

$$P_w \leq \sum_{r=d_{\min}}^{n} A_r Q\left(\sqrt{2r\frac{E_c}{N_0}}\right) = \sum_{r=d_{\min}}^{n} A_r Q\left(\sqrt{2Rr\frac{E_b}{N_0}}\right). \tag{4.7.11}$$

For good channels with a high E_b/N_0, P_w is approximately bounded by the dominant term in the summation

$$P_w \lessapprox A_{d_{\min}} Q\left(\sqrt{2Rd_{\min}\frac{E_b}{N_0}}\right). \tag{4.7.12}$$

For $E_b/N_0 \to \infty$, (3.8.6) is even asymptotically exact. A general weak upper bound without knowledge of the weight distribution is given by

$$P_w \le (2^k - 1) \cdot Q\left(\sqrt{2Rd_{\min}\frac{E_b}{N_0}}\right). \tag{4.7.13}$$

Proof. Let $\mathcal{C} = \{\boldsymbol{a}_1, \dots, \boldsymbol{a}_{2^k}\}$ be an enumeration of the code and let $\boldsymbol{a}_i = (a_{i,0}, \dots, a_{i,n-1})$ be transmitted and $\boldsymbol{y} = (y_0, \dots, y_{n-1}) = \boldsymbol{a}_i + \boldsymbol{\nu}$ be received. The components in the noise word $\boldsymbol{\nu} = (\nu_0, \dots, \nu_{n-1})$ are statistically independent with the variance $\sigma^2 = N_0/2$. The encoded bits are in $\{+\sqrt{E_c}, -\sqrt{E_c}\}$ according to definition 1.3. As in the proof of Theorem 3.16 we will consider the decision region of $\boldsymbol{a}_i$ for which, according to Theorem 1.4

$$\mathcal{M}_i = \left\{\boldsymbol{y} \;\middle|\; \|\boldsymbol{y} - \boldsymbol{a}_i\| \le \|\boldsymbol{y} - \boldsymbol{b}\| \text{ for all } \boldsymbol{b} \in \mathcal{C}\right\}.$$

The complementary set is composed of the sets

$$\mathcal{M}_{i,j} = \left\{\boldsymbol{y} \;\middle|\; \|\boldsymbol{y} - \boldsymbol{a}_j\| < \|\boldsymbol{y} - \boldsymbol{a}_i\|\right\}.$$

As in Theorem 3.16 the union bound gives us

$$P_i = P(\text{decoding error} \mid \boldsymbol{a}_i \text{ transm.}) \le \sum_{\substack{j=1\\ j\neq i}}^{2^k} P(\boldsymbol{y} \in \mathcal{M}_{i,j} \mid \boldsymbol{a}_i \text{ transm.}).$$

In constrast to Theorem 3.16, the 2-codeword-error probability can now be exactly computed for the AWGN channel:

$$\begin{aligned}
&P(\boldsymbol{y} \in \mathcal{M}_{i,j} \mid \boldsymbol{a}_i \text{ transm.})\\
&\quad = P\Big(\|\boldsymbol{y} - \boldsymbol{a}_j\| < \|\boldsymbol{y} - \boldsymbol{a}_i\| \;\Big|\; \boldsymbol{a}_i \text{ transm.}\Big)\\
&\quad = P\Big(\|\boldsymbol{\nu} + \boldsymbol{a}_i - \boldsymbol{a}_j\|^2 < \|\boldsymbol{\nu}\|^2\Big)\\
&\quad = P\left(\sum_{r=0}^{n-1}\Big(\nu_r^2 + 2\nu_r(a_{i,r} - a_{j,r}) + (a_{i,r} - a_{j,r})^2\Big) < \sum_{r=0}^{n-1}\nu_r^2\right)\\
&\quad = P\left(\sum_{r=0}^{n-1}\nu_r(a_{i,r} - a_{j,r}) < -\frac{1}{2}\sum_{r=0}^{n-1}(a_{i,r} - a_{j,r})^2\right).
\end{aligned}$$

In this case the probability only refers to the noise but not to the encoded bits. For $a_{i,r} \neq a_{j,r}$, $(a_{i,r} - a_{j,r})^2 = (2\sqrt{E_c})^2$, thus for the summation over r,

$$\sum_r (a_{i,r} - a_{j,r})^2 = 4d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)E_c.$$

Furthermore, the random variable $\sum_r \nu_r(a_{i,r} - a_{j,r})$ has a normal distribution with the mean value 0 and the variance

$$\sum_r \frac{N_0}{2}(a_{i,r} - a_{j,r})^2 = 2d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)N_0E_c.$$

Similar to (A.3.11) or (A.3.12), the Gaussian Q-function arises:

$$\begin{aligned} P(\boldsymbol{y} \in \mathcal{M}_{i,j} \mid \boldsymbol{a}_i \text{ transm.}) &= P\left(\frac{\sum\limits_{r=0}^{n-1} \nu_r(a_{i,r} - a_{j,r})}{\sqrt{2d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)N_0E_c}} < -\frac{2d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)E_c}{\sqrt{2d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)N_0E_c}} \right) \\ &= Q\left(\sqrt{2d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)\frac{E_c}{N_0}} \right). \end{aligned}$$

Again similar to Theorem 3.16 the overall result for P_i is

$$P_i \le \sum_{\substack{j=1 \\ j\ne i}}^{2^k} Q\left(\sqrt{2d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)\frac{E_c}{N_0}} \right) = \sum_{r=d_{\min}}^{n} A_r Q\left(\sqrt{2r\frac{E_c}{N_0}} \right).$$

As in Theorem 3.16 the same bound follows for P_w. The bounds (3.8.6) and (3.8.7) obviously follow from (3.8.5). We still need to prove the asymptotic accuracy of (3.8.6). For this a lower bound can be derived for P_i by using the 2-codeword-error probabilities:

$$\begin{aligned} P_i &= P\left(\boldsymbol{y} \in \bigcup_{\substack{j=1 \\ j\ne i}}^{2k} \mathcal{M}_{i,j} \;\middle|\; \boldsymbol{a}_i \text{ transmitted} \right) \\ &\ge \max_{\substack{j=1,\dots,2^k \\ j\ne i}} P(\boldsymbol{y} \in \mathcal{M}_{i,j} \mid \boldsymbol{a}_i \text{ transmitted}) \\ &= \max_{\substack{j=1,\dots,2^k \\ j\ne i}} Q\left(\sqrt{2d_H(\boldsymbol{a}_i, \boldsymbol{a}_j)\frac{E_c}{N_0}} \right) \\ &= Q\left(\sqrt{2d_{\min}\frac{E_c}{N_0}} \right). \end{aligned}$$

(The last equality follows from the shape of the Gaussian Q-function shown in Figure A.2). The same lower bound is also valid for P_w. Thus, on the whole, as $E_b/N_0 \to \infty$:

$$Q\left(\sqrt{2Rd_{\min}\frac{E_b}{N_0}} \right) \le P_w \le A_{d_{\min}} Q\left(\sqrt{2Rd_{\min}\frac{E_b}{N_0}} \right).$$

The influence of the constant $A_{d_{\min}}$ asymptotically disappears, thus the equality in (3.8.6) is proven. ■

According to (3.8.6) and (A.3.18), $P_b \approx \text{const} \cdot P_w \approx \text{const} \cdot e^{-Rd_{\min} \cdot E_b/N_0}$ for AWGN channels with soft-decision decoding, which has already been used in (1.7.10) to derive the asymptotic coding gain with the result that $G_{\text{a,soft}} = 10 \cdot \log_{10}(Rd_{\min})$ dB.

Example 4.11. Again consider the perfect $(7,4,3)_2$ Hamming code with $A_0 = A_7 = 1$, $A_3 = A_4 = 7$, $R = 4/7$, $d_{\min} = 3$ and $t = 1$. For this code, P_w is depicted over E_b/N_0 in Figure 3.5 to demonstrate the various previously derived bounds for hard- and soft-decision decoding.

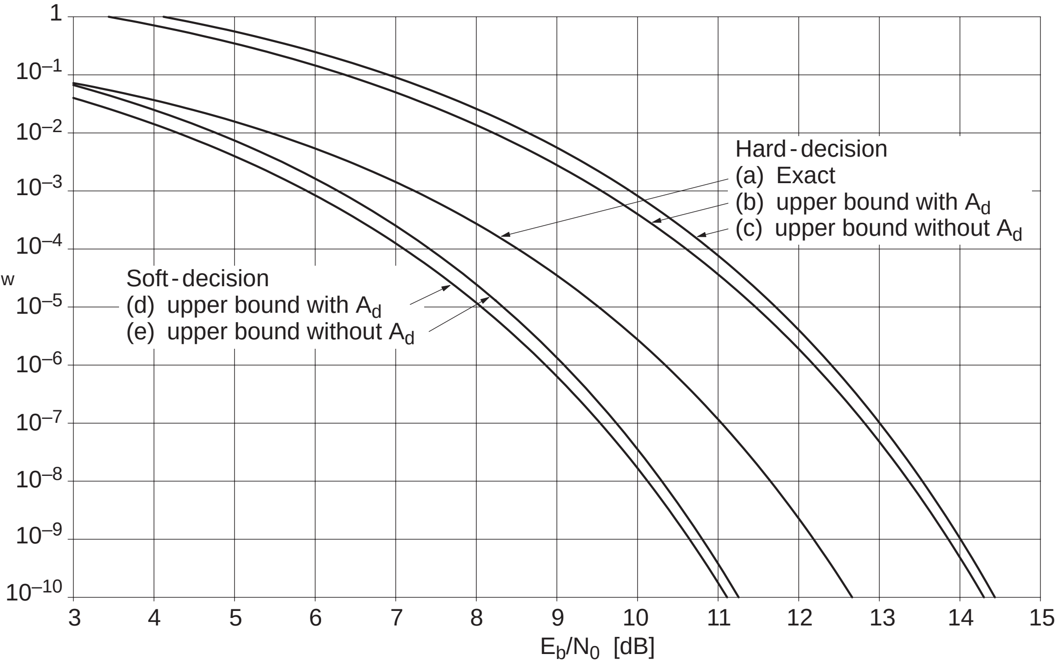

Figure 4.8. Various performance bounds demonstrated on the $(7,4)_2$ Hamming code

For hard decisions $p_e = Q(\sqrt{2RE_b/N_0})$ is valid. According to Theorem 3.15, the exact result is

$$P_w = 1 - (1-p_e)^7 - 7p_e(1-p_e)^6 \quad \text{(curve a)}$$

with $G_{\text{a,hard}} = 10 \cdot \log_{10}(R(t+1)) = 10 \cdot \log_{10}(4/7 \cdot 2) = 0.58$ dB. The upper bound according to (3.8.4) is

$$P_w \le 7\sqrt{4p_e(1-p_e)}^3 + 7\sqrt{4p_e(1-p_e)}^4 + \sqrt{4p_e(1-p_e)}^7 \quad \text{(curve b)}$$

with $G_{\text{a,hard}} = 10 \cdot \log_{10}(Rd_{\min}/2) = 10 \cdot \log_{10}(4/7 \cdot 3/2) = -0.67$ dB, i.e., the union bound has an asymptotic error of 1.25 dB for hard decisions. The general weak upper bound according to (3.8.3) is

$$P_w \le 15\sqrt{4p_e(1-p_e)}^3 \quad \text{(curve c).}$$

Now, for soft decisions according to (3.8.5),

$$P_w \le 7Q\left(\sqrt{2\frac{4}{7}3\frac{E_b}{N_0}}\right) + 7Q\left(\sqrt{2\frac{4}{7}4\frac{E_b}{N_0}}\right) + Q\left(\sqrt{2\frac{4}{7}7\frac{E_b}{N_0}}\right) \quad \text{(curve d)}$$

with $G_{\text{a,soft}} = 10 \cdot \log_{10}(Rd_{\text{min}}) = 10 \cdot \log_{10}(4/7 \cdot 3) = 2.34$ dB. The general weak upper bound according to (3.8.7) is

$$P_w \le 15Q\left(\sqrt{2\frac{4}{7}3\frac{E_b}{N_0}}\right) \quad \text{(curve e).}$$

In this example the gain by using soft-decision decoding is at least $2.34 - 0.58 = 1.76$ dB, though this can only be realized for a small P_w. For a small E_b/N_0 the curves (d) and (e) intersect with curve (a), since the union bounds are upper bounds and only the graph of (a) is exact. ■

4.8 Problems

4.1. Prove Theorem 3.1 by rearranging the components.

4.2. Prove that for a linear binary code the number of codewords beginning with 0 is equal to the number of codewords beginning with 1 as long as there is at least one codeword beginning with 1.

4.3. Prove that always one of the following statements applies for a linear binary code. Either every codeword has an even weight or codewords with an odd and even weight occur the same number of times.

4.4. How many different linear $(3, 2)_2$ block codes exist, if codes with a constant zero component in each codeword are excluded?

4.5. Construct a linear $(3, 2)_3$ block code which contains the words 101 and 120. Is the code unambiguous?

4.6. Determine the maximum minimum distance for the $(4, 2)_2$ block codes directly as well as with the Hamming bound.

4.7. A *palindrome* is a symmetric word, i.e., its letters taken in reverse order result in the same word. Let $\mathcal{C}$ be the set of all palindromes over $\mathbb{F}_2$ of length n. Is $\mathcal{C}$ a linear block code? Determine d_{min} and $|\mathcal{C}|$.

4.8. Let $\mathcal{C}$ be a linear (n,k,d) block code. Construct the new code $\mathcal{C}' = \{\boldsymbol{a} \in \mathcal{C} | w_H(\boldsymbol{a}) \text{ even}\}$. Is $\mathcal{C}'$ linear? Determine (n',k',d') for $\mathcal{C}'$.

4.9. Given that there are 4 parity-check bits to correct 1 error per codeword, how many information bits, at best, can be protected in the binary case?

4.10. Let a linear $(n,2)_2$ block code be able to correct 2 errors. Determine the minimum block length and a possible code and interpret the result.

4.11. Prove that a perfect $(11,6)_3$ code can exist. How many errors are correctable? Such a code does actually exist and is called a *ternary Golay code*. The perfect binary $(23,12,7)_2$ Golay code has already been discussed in Figure 1.10 and Example 3.6(2).

4.12. Repeat Example 3.7 for a $(127,k,5)_2$ BCH code and compare the result with Table 7.1.

4.13. Are the binary block codes $(63,31,7)$, $(63,45,7)$, $(127,109,7)$ possible? Compare the result with Table 7.1.

4.14. Which combinations for correcting and detecting errors are possible for an $(n,k,8)_2$ block code?

4.15. According to Table 7.1, a $(255,k,11)_2$ code exists with $k = 215$. Compare the code parameters with the Hamming and the Gilbert-Varshamov bounds and their asymptotic forms.

4.16. For a linear $(n,k)_2$ code, of which none of the codeword components are always equal to zero, prove that

$$\sum_{\boldsymbol{a}\in\mathcal{C}} w_H(\boldsymbol{a}) \;=\; n2^{k-1} \;=\; \sum_{r=1}^{n} rA_r \;=\; A'(1) \tag{4.8.1}$$

(A' being the derivate of the complete path enumerator).

4.17. Using the fact that the repetition code is perfect, prove that

$$\sum_{r=0}^{t} \binom{2t+1}{r} = 4^t. \tag{4.8.2}$$

How does this relate to (A.2.2)?

4.18. A $(63,36)_2$ BCH code can correct up to 5 errors. With 9 blocks of a $(7,4)_2$ Hamming code put together, a $(63,36)_2$ code emerges whose error correction capability is to be compared to that of the BCH code. Determine the bit-error probability for the BCH code at $p_e = 10^{-3}$ approximately, as well as the asymptotic coding gain.

4.19. Determine the upper bound for the word-error probability of the perfect $(15, 11, 3)_2$ Hamming code over the BSC with $p_e = 10^{-2}$. How exact is the bound? Determine an upper bound for soft decisions (corresponding to $p_e = 10^{-2}$) with as little effort as possible and interpret the result.

4.20. Consider the $(n, 2)_2$ code $\mathcal{C} = \{000...0, 100...0, 011...1, 111...1\}$. Prove that $P_{ue} \geq 2^{-nH_2(1/n)}$ at $p_e = 1/n$ and compare it to $p_e = 1/2$ for $n = 100$.

4.21. Given random codes with the average weight distribution according to (3.5.10), prove the co-called *Massey bound* [205] for the undetected error probability where the averaging is with respect to the individual worst case BSC bit-error probability for every code:

$$E\Big(\max_{p_e} P_{ue}\Big) \;\leq\; n \cdot 2^{-(n-k)}. \tag{4.8.3}$$

4.22. Prove that the probability of undetected BSC error patterns when averaging over all linear random codes with a weight distribution according to (3.5.11) is given by

$$E(P_{ue}) \;=\; 2^{-(n-k)}(1 - (1 - p_e)^k). \tag{4.8.4}$$

Conclude from this that error-detection codes exist which fulfil the bound (3.6.4) and moreover that on average about every second randomly chosen linear error-detection code fulfils this bound.

4.23. Prove the inequality $\lim\limits_{n\to\infty} d_n \leq 2$ for every arbitrary series of $(n, n - m, d_n)_2$ codes with a fixed number m of parity-check bits.

4.24. For the mean value over all random codes prove the following asymptotic properties as $n \to \infty$. For a fixed number of information bits $d_{\min}/n \to 1/2$ applies and for a fixed number of parity-check bits $d_{\min}/n \to 0$ applies.

4.25. Let $\mathcal{C}$ be a not necessarily linear $(n, k, d_{\min})_2$ code. Let $B_d(\boldsymbol{a})$ be the number of all codewords of distance d from the codeword $\boldsymbol{a}$. Obviously $\sum\limits_{d=0}^{n} B_d(\boldsymbol{a}) = 2^k$. Prove that $\sum\limits_{\boldsymbol{a}\in\mathcal{C}} B_d(\boldsymbol{a})$ is equal to the number of all pairs of codewords of distance d. Prove the following generalization of Theorem 3.14 for error-detection decoding.

$$P_{ue} \;=\; \sum_{d=1}^{n} \tilde{A}_d \cdot p_e^d (1 - p_e)^{n-d}, \quad \text{where } \tilde{A}_d = 2^{-k} \cdot \sum_{\boldsymbol{a}\in\mathcal{C}} B_d(\boldsymbol{a}). \tag{4.8.5}$$

4.26. (Based on the contribution of H.A.Loeliger in [22]). Let $\mathcal{E} \subseteq \mathbb{F}_q^n$ be an arbitrary set of error patterns with $\boldsymbol{0} \in \mathcal{E}$. A linear $(n, k)_q$ block code $\mathcal{C}$ is said to

(1) correct all error patterns in $\mathcal{E}$, if $\boldsymbol{a}+\boldsymbol{e} \neq \boldsymbol{a}'+\boldsymbol{e}'$ for all $\boldsymbol{a}, \boldsymbol{a}' \in \mathcal{C}$ with $\boldsymbol{a} \neq \boldsymbol{a}'$ and all $\boldsymbol{e}, \boldsymbol{e}' \in \mathcal{E}$.

(2) detect all error patterns in $\mathcal{E}$, if $\boldsymbol{a}+\boldsymbol{e} \notin \mathcal{C}$ for all $\boldsymbol{a} \in \mathcal{C}$ and all $\boldsymbol{e} \in \mathcal{E} \backslash \{\boldsymbol{0}\}$. Prove that this is equivalent to $\mathcal{C} \cap \mathcal{E} = \{\boldsymbol{0}\}$.

Show that these definitions are generalizations of Definition 3.4. Consider the set $\Delta\mathcal{E} = \{\boldsymbol{e}-\boldsymbol{e}' | \boldsymbol{e}, \boldsymbol{e}' \in \mathcal{E}\}$ and prove the following equivalences:

$$\begin{aligned} & \mathcal{C} \text{ corrects all error patterns in } \mathcal{E} \\ \Longleftrightarrow\ & \mathcal{C} \text{ detects all error patterns in } \Delta\mathcal{E} \\ \Longleftrightarrow\ & \mathcal{C} \cap \Delta\mathcal{E} = \{\boldsymbol{0}\}. \end{aligned}$$

Prove that

$$\Delta K_t(\boldsymbol{a}) = K_{2t}(\boldsymbol{0}).$$

Prove the following generalization of the Hamming bound. If $\mathcal{C}$ corrects all error patterns in $\mathcal{E}$, then

$$H_q(\mathcal{E}) = \frac{1}{n}\log_q |\mathcal{E}| \le 1 - R. \tag{4.8.6}$$

For the BSC with the bit-error probability p_e prove the asymptotic result

$$H_q(\mathcal{E}) = H_2\left(\frac{d_{\min}}{2n}\right) = 1 - R \tag{4.8.7}$$

where H_2 describes the binary entropy function according to (A.2.3).