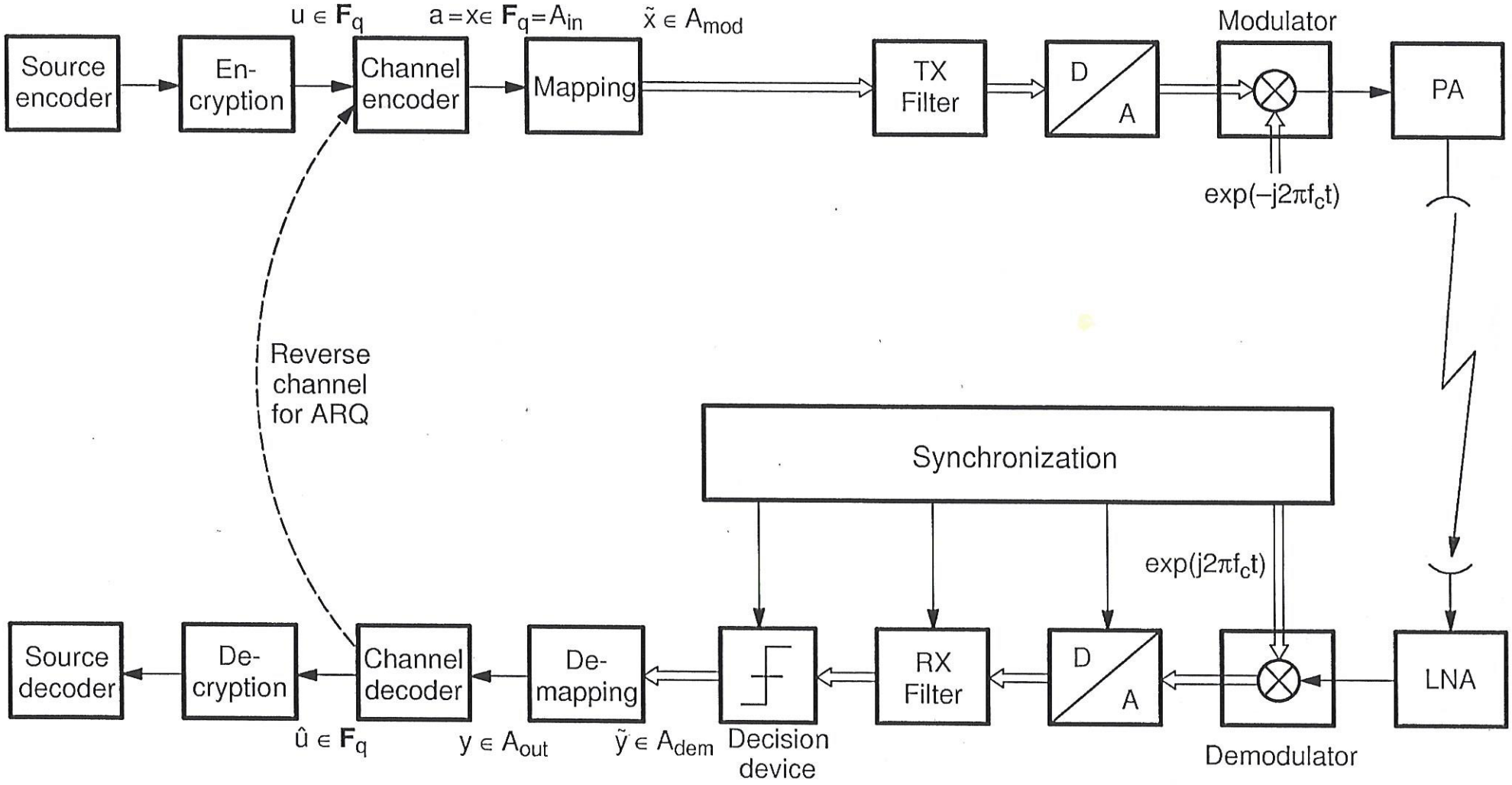


Typisches Blockdiagramm einer drahtlosen Übertragung

Beispiel für bedingte Wahrscheinlichkeiten: Medizinisches Testverfahren

$p = P(\text{Person krank}) = 0.002$

Testversagen:

$a = P(\text{Test negativ} \mid \text{Person krank}) = 0.01$

$b = P(\text{Test positiv} \mid \text{Person gesund}) = 0.001$

$$P(\text{irrtümlich krank}) = P(\text{Person gesund} \mid \text{Test positiv})$$

$$= \frac{P(\text{gesund} \wedge \text{positiv})}{P(\text{positiv})} = \frac{P(\text{positiv} \wedge \text{gesund})}{P(\text{positiv} \wedge \text{gesund}) + P(\text{positiv} \wedge \text{krank})}$$

$$= \frac{P(\text{positiv} \mid \text{gesund}) \cdot P(\text{gesund})}{P(\text{positiv} \mid \text{gesund}) \cdot P(\text{gesund}) + P(\text{positiv} \mid \text{krank}) \cdot P(\text{krank})}$$

$$= \frac{b(1-p)}{b(1-p) + (1-a)p} = 0.3351 \approx \begin{cases} b/p & \text{für } b \ll p \\ 1 & \text{für } b \gg p \end{cases}$$

$$P(\text{irrtümlich gesund}) = P(\text{Person krank} \mid \text{Test negativ})$$

$$= \frac{P(\text{negativ} \wedge \text{krank})}{P(\text{negativ})}$$

$$= \frac{P(\text{negativ} \mid \text{krank}) \cdot P(\text{krank})}{P(\text{negativ} \mid \text{gesund}) P(\text{gesund}) + P(\text{negativ} \mid \text{krank}) \cdot P(\text{krank})}$$

$$= \frac{ap}{(1-b)(1-p) + ap} \approx 0.00002$$

Zahlenbeispiel 1 Mio Personen

	krank	gesund	
negativ	20	997002	997022
positiv	1980	998	2978
	2000	998000	1 000 000

$$P(\text{gesund} \mid \text{positiv}) = \frac{998}{2978} = 0.3351$$

$$P(\text{krank} \mid \text{negativ}) = \frac{20}{997022} = 0.00002$$

Wahrscheinlichkeit, daß Sie auf Ihrem nächsten Flug entführt werden	$5{,}5 \cdot 10^{-6}$
Wahrscheinlichkeit für 6 Richtige im Lotto	$7{,}1 \cdot 10^{-8}$
Jährliche Wahrscheinlichkeit, von einem Blitz getroffen zu werden	10^{-7}
Risiko, von einem Meteoriten erschlagen zu werden	$1{,}6 \cdot 10^{-12}$
Anzahl der Moleküle in einem Mol (bei Gasen 22,4 Liter)	$6{,}023 \cdot 10^{23}$
Anzahl der Atome der Erde	10^{51} (2^{170})
Anzahl der Atome in der Sonne	10^{57} (2^{190})
Anzahl der Atome in unserer Galaxis 10^{67} (2^{223})	
Anzahl der Atome im Weltall (ohne dunkle Materie)	10^{77} (2^{265})
Zeit bis zur nächsten Eiszeit	14.000 (2^{14}) Jahre
Zeit, bis die Sonne zu einer Nova wird	10^{9} (2^{30}) Jahre
Alter der Erde	10^{9} (2^{30}) Jahre
Alter des Universums	10^{10} (2^{34}) Jahre
Wenn das Weltall geschlossen ist:	
Lebensdauer des Weltalls	10^{11} (2^{37}) Jahre
Wenn das Weltall offen ist:	
Zeit bis sich die Planeten aus den Sonnensystemen lösen	10^{15} (2^{50}) Jahre
Zeit bis sich die Sterne aus den Galaxienverbänden lösen	10^{19} (2^{64}) Jahre

Viele der in diesem Buch auftretenden Zahlen überschreiten diese physikalischen Werte. So würde z. B. ein Computer, der pro Sekunde 2.000.000.000 IDEA-Verschlüsselungen berechnen kann, zum Durchprobieren aller 2^{128} möglichen Schlüssel

$$\frac{2^{128}}{2 \cdot 10^{9} \frac{1}{\text{sek}} \cdot 3{,}2 \cdot 10^{7} \frac{\text{sek}}{\text{Jahr}}} \approx \frac{3{,}4 \cdot 10^{38}}{6{,}4 \cdot 10^{16}} \text{ Jahre} \approx 5{,}3 \cdot 10^{21} \text{ Jahre}$$

Wenn das Weltall geschlossen ist, läßt sich diese Berechnung nicht mehr vollständig durchführen.

where the inequality below the brace is obtained as follows: by simply remodelling

$$\eta \geq \sqrt{\alpha} = \frac{\sqrt{\alpha(\alpha+\beta)} - \alpha}{\sqrt{\alpha+\beta} - \sqrt{\alpha}}$$

we obtain

$$\underbrace{\eta(\sqrt{\alpha+\beta} - \sqrt{\alpha}) - \sqrt{\alpha(\alpha+\beta)} + \alpha + \eta^2/2 + \beta/2}_{= (\eta + \sqrt{\alpha+\beta} - \sqrt{\alpha})^2/2} \geq \eta^2/2 + \beta/2.$$

Thus (A.4.18) is proved. This inequality is used for the proof of Theorem 11.1. ■

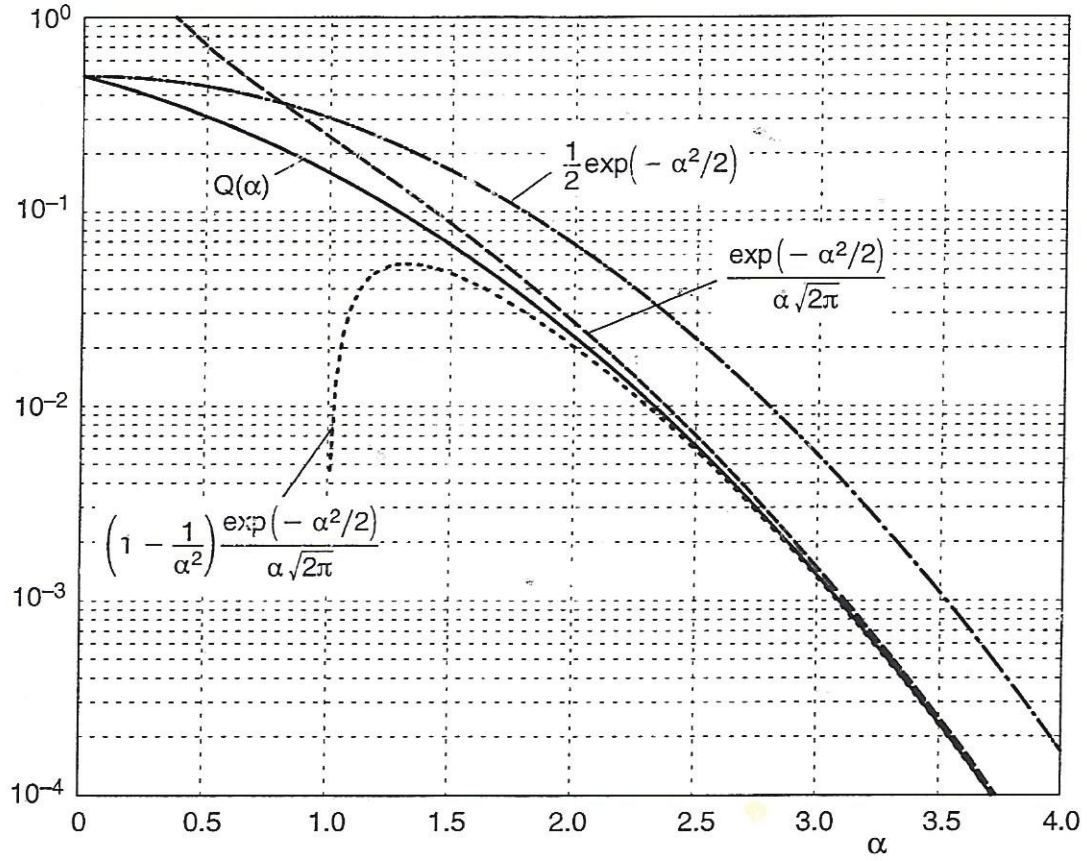


Figure A.5. Bounds for the complementary Gaussian error function $Q(\alpha)$

For large values of α there exists an extremely precise approximation [75, 83] which is not only of theoretical interest but also of great practical importance for numerical computations:

$$\left(1 - \frac{1}{\alpha^2}\right) \frac{e^{-\alpha^2/2}}{\alpha\sqrt{2\pi}} < Q(\alpha) < \frac{e^{-\alpha^2/2}}{\alpha\sqrt{2\pi}} \quad \text{for } \alpha > 0. \tag{A.4.19}$$

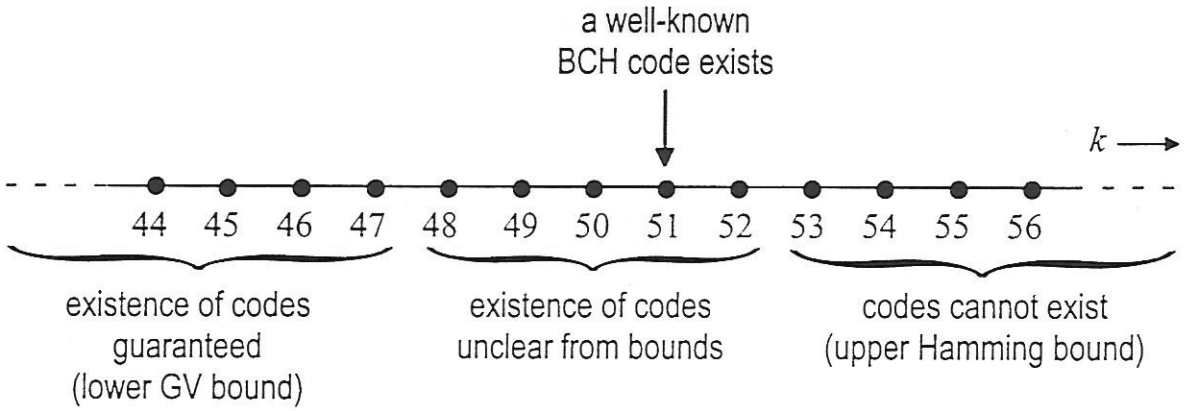


Figure 4.5. Gilbert-Varshamov and Hamming bounds for $(63, k, 5)_2$ codes

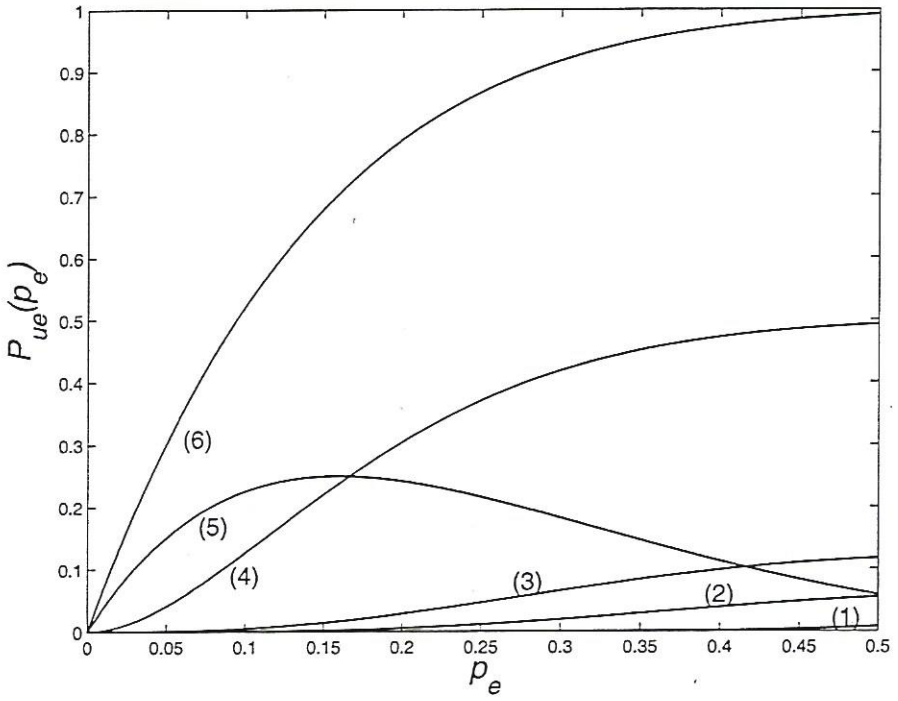


Figure 4.7a. Undetected error probability (linearly scaled axes)

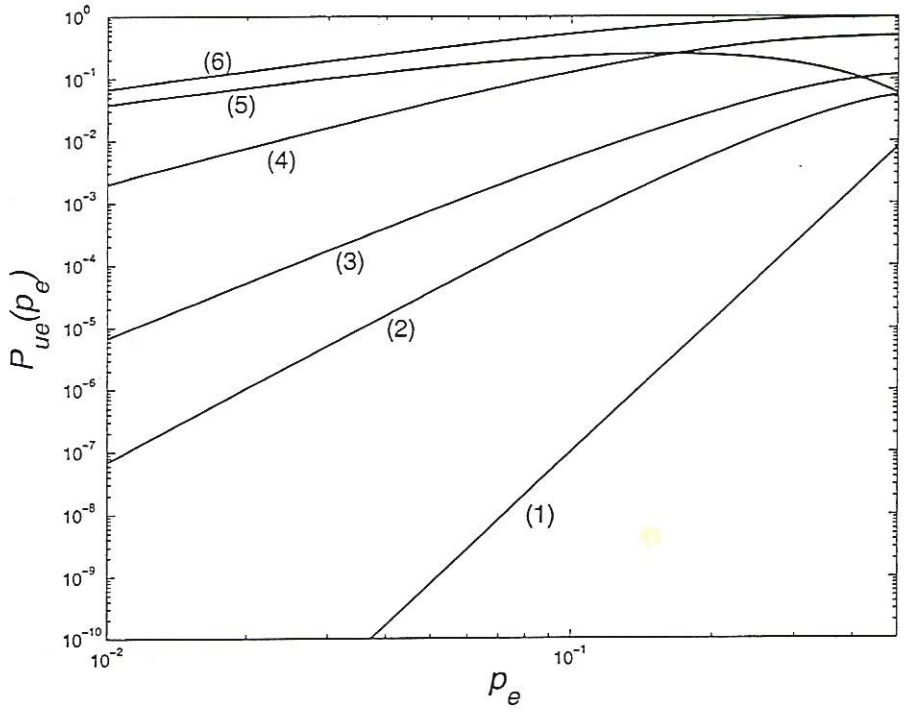


Figure 4.7b. Undetected error probability (both axes logarithmically scaled)

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

5.4.3 The Colored Hats Puzzle

The colored hats puzzle is a nice example for the successful application of Hamming codes to problems which do not seem to be connected to communications or coding. This puzzle was first published as an article which appeared in the Science Times section of the New York Times of April 10th, 2001.

The puzzle is stated as follows: Each player of a team is randomly and independently assigned to wear a colored hat (either red=0 or blue=1). Each player views the colors of his other teammates (but can not see his own color), and then tries to guess the color of his own hat. No communication is allowed between the players except for a strategy session before the game begins. It is allowed that some players do not guess and remain neutral. The team wins a prize if at least one player guesses his own color correctly and no player guesses incorrectly. Vice versa, the team loses if there are no guesses or at least one player guesses incorrectly.

On the first view, the team seems to have a chance of winning of only 50%. However, with a smart strategy, the chance of winning is almost 100%. More precisely, if the number of players has the form $n = 2^r - 1$, then the chance of winning is $n/(n+1)$.

The smart strategy is defined as follows. Let $\boldsymbol{y} = (y_0, \ldots, y_{n-1})$ be a vector representing the colors of the n hats. Let $\mathcal{C}$ be the $(2^r-1, 2^r-r-1, 3)_2$ Hamming code. By viewing his teammates, the i-th player knows the two vectors

$$\begin{aligned} \boldsymbol{a}_i &= (y_0, \ldots, y_{i-1}, 0, y_{i+1}, \ldots, y_{n-1}), \\ \boldsymbol{b}_i &= (y_0, \ldots, y_{i-1}, 1, y_{i+1}, \ldots, y_{n-1}) \end{aligned}$$

and guesses the color of his own hat as follows:

$$\begin{aligned} \boldsymbol{a}_i \notin \mathcal{C} \text{ and } \boldsymbol{b}_i \notin \mathcal{C} &\quad\Rightarrow\quad \text{neutral} \\ \boldsymbol{a}_i \in \mathcal{C} \text{ and } \boldsymbol{b}_i \notin \mathcal{C} &\quad\Rightarrow\quad \text{guess 1} \\ \boldsymbol{a}_i \notin \mathcal{C} \text{ and } \boldsymbol{b}_i \in \mathcal{C} &\quad\Rightarrow\quad \text{guess 0} \end{aligned}$$

The fourth case $\boldsymbol{a}_i \in \mathcal{C}$ and $\boldsymbol{b}_i \in \mathcal{C}$ is not possible since $d_H(\boldsymbol{a}_i, \boldsymbol{b}_i) = 1$, however, $\mathcal{C}$ has a minimum Hamming distance of 3.

Now we compute the chance of winning. Two cases have to be distinguished. Firstly, if $\boldsymbol{y} \in \mathcal{C}$, then all players guess incorrectly and the team loses. Secondly, we consider the case of $\boldsymbol{y} \notin \mathcal{C}$. Since $\mathcal{C}$ is perfect, there exists exactly one $\boldsymbol{c} \in \mathcal{C}$ with $d_H(\boldsymbol{y}, \boldsymbol{c}) = 1$. Let l be the position where the two vectors differ.

The l-th player guesses as follows: if $\boldsymbol{a}_l \in \mathcal{C}$, then he guesses 1 and $\boldsymbol{a}_l \neq \boldsymbol{y}$ since $\boldsymbol{y} \notin \mathcal{C}$. Hence, $\boldsymbol{a}_l = \boldsymbol{c} \in \mathcal{C}$ and $\boldsymbol{b}_l = \boldsymbol{y}$ and so his guess is correct. The same arguments also show a correct guess in case of $\boldsymbol{b}_l \in \mathcal{C}$. All other s-th players with $s \neq l$ remain neutral since $\boldsymbol{a}_s \notin \mathcal{C}$ and $\boldsymbol{b}_s \notin \mathcal{C}$.

In summary, in case of $\boldsymbol{y} \notin \mathcal{C}$ one player guesses correctly and all other players remain neutral. Hence

$$P(\boldsymbol{y} \notin \mathcal{C}) = \frac{2^n - 2^k}{2^n} = 1 - 2^{-(n-k)} = 1 - 2^{-r} = 1 - (n+1)^{-1} = \frac{n}{n+1}$$

is the teams's winning chance.

BCH-Codes in Bild 3.4 (Asymptot. Schranken)

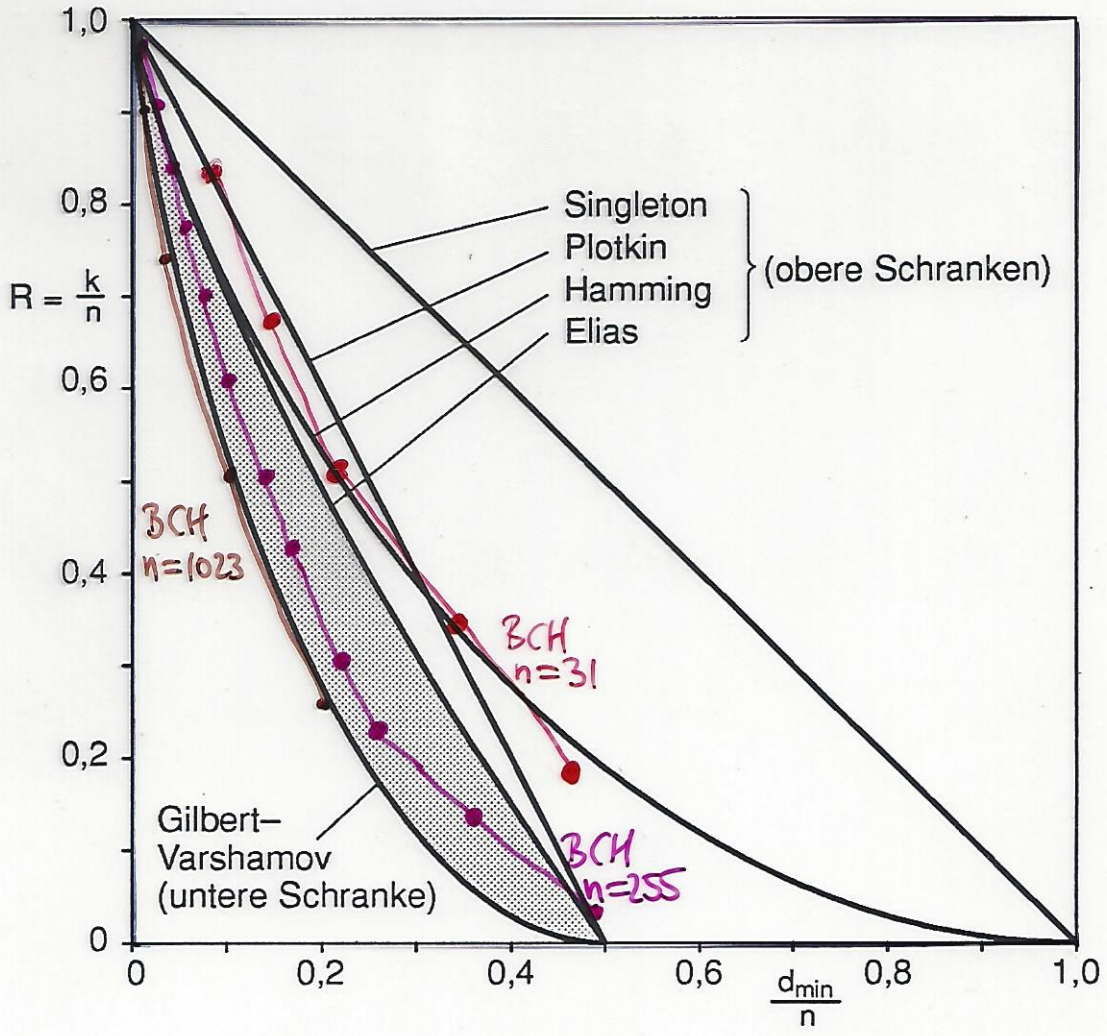


Bild 3.4. Asymptotische Schranken

Alle Codes liegen unter den oberen Schranken, insbesondere also unter der Elias-Schranke, und es gibt einige gute Codes oberhalb der unteren Gilbert-Varshamov-Schranke, d.h. im schraffierten Bereich.

Unterhalb des schraffierten Bereiches liegende Codes sind als schlecht anzusehen.

Nach der Gilbert-Varshamov-Schranke ist zumindest die Existenz von sogenannten *asymptotisch guten Codefamilien* (n_s, k_s, d_s) mit

$$\lim_{s\to\infty} \frac{k_s}{n_s} > 0 \quad \text{und} \quad \lim_{s\to\infty} \frac{d_s}{n_s} > 0 \tag{3.4.6}$$

garantiert. Aber alle bekannten Codefamilien erfüllen diese Eigenschaft nicht und sind damit *asymptotisch schlecht.*

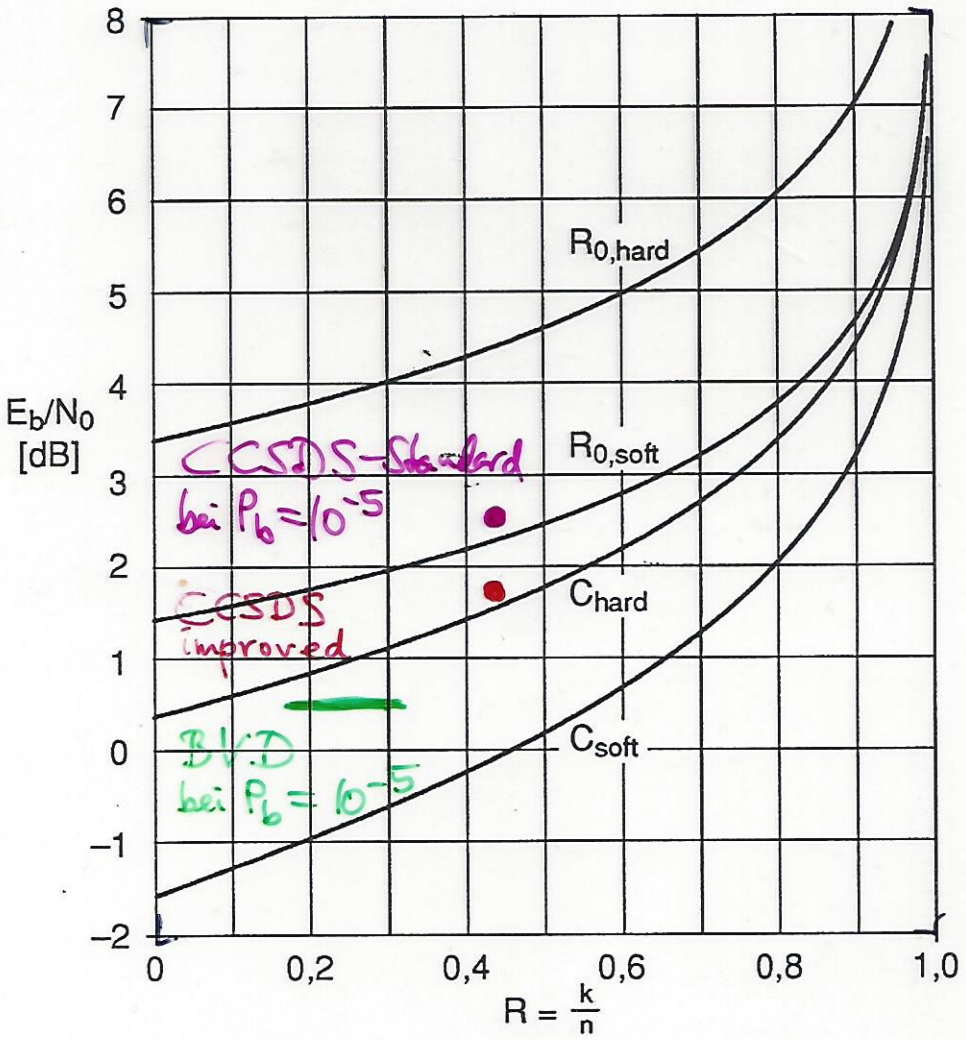


Bild 2.4. Notwendiges E_b/N_0 für $R = R_0$ bzw. $R = C$ beim AWGN ($q = 2$)

Folgerungen:

(1) Für E_b/N_0 kleiner als der linke Grenzwert ist keine Übertragung mit $R = R_0$ möglich! Ursache: Ein kleines R bedeutet ein große Bandbreite und damit ein kleines E_c gegenüber N_0. Damit wird jedes einzelne Codebit vom Rauschen weitgehend überdeckt und R_0 fällt sehr klein aus. Ab einer gewissen Grenze wird dann R_0 kleiner als R, so daß $R = R_0$ nicht mehr erreichbar ist.

(2) Der Abstand zwischen C_{soft} und C_{hard} ist nicht näherungsweise konstant wie bei den R_0-Kurven. Für $R \to 1$ laufen C_{hard} und $R_{0,\text{soft}}$ zusammen.

(3) Es ist kaum sinnvoll, $R < 1/2$ zu verwenden, da der Gewinn beim Übergang von $R = 1/2$ auf $R \to 0$ maximal nur rund 1 dB bezüglich R_0 beträgt!

If only burst errors of a length smaller than 4 are supposed for the channel, then the error pattern e_1 is more probable than e_2 and even more probable than e_3, in spite of $w_H(e_3) < w_H(e_1)$.

If the error pattern contains a maximum of t single errors or one cyclic burst error of length t, then the number of possible error patterns including the errorfree case is

$$L_t = \left\{ \begin{array}{ll} \displaystyle\sum_{r=0}^{t} \binom{n}{r} (q-1)^r & \text{single error (number of} \leq t) \\ & \\ 1 + n(q-1)q^{t-1} & \text{cyclic burst error (length} \leq t) \end{array} \right\}. \qquad (6.6.3)$$

The number of single errors was already determined in Theorem 4.13. For cyclic burst errors as in (5.6.2), there exist $q-1$ possibilities for the first coefficient in $b(x)$ and q possibilities for the other $t-1$ coefficients. For i there are n possibilities in addition to the errorfree case. However, these considerations only apply for $2t < n+2$. The opposite $2t \geq n+2$ is of less practical significance and in this case only $L_t \leq 1 + n(q-1)q^{t-1}$ is valid.

Example 6.7. Consider a non-cyclic linear $(6,2,3)_2$ code with

$$\mathcal{C} = \{000000, 111000, 000111, 111111\}.$$

The burst error 111000 is equal to a codeword and is not detected. Permutations lead to an equivalent code also with $d_{\min} = 3$,

$$\mathcal{C} = \{000000, 101010, 010101, 111111\}.$$

This code detects all $L_4 - 1 = 45 < 6 \cdot 2^3 = 48$ cyclic burst errors of a length smaller than or equal to 4 because these burst errors are not codewords:

100000	110000	101000	111000	100100	101100	110100	111100
010000	011000	010100	011100	010010	010110	011010	011110
001000	001100	001010	001110	001001	001011	001101	001111
000100	000110	000101	000111		100101	100110	100111
000010	000011	100010	100011		110010	010011	110011
000001	100001	010001	110001		011001	101001	111001.

Independent of the permutation $d_{\min} = 2$ arbitrary single errors are always detected in this example. ■

For the detection of burst errors equivalent codes are not equally suitable, especially if a non-cyclic code is compared to a cyclic code. Permutations of columns can cause a good code to turn into a bad one. The same goes for the correction of burst errors.

In contrast, for the detection or the correction of single errors, equivalent codes are to be considered to be equal.

RS- und BCH-Codes für Bündel- und Einzelfehler

---

es existieren (Begründung später)

BCH-Code $(63, 45, 7)_2$ $t=3$ Bits

RS-Code $(15, 11, 5)_{16} \hat{=} (60, 44, 5)_2$ $t=2$ ~~Bytes~~ Symbole

Beispiele

$e = (0000\ 1111\ 1111\ 0000 \ldots\ldots)$ RS Korrigiert BCH versagt

$e = (0001\ 0100\ 0100\ 0000 \ldots\ldots)$ RS versagt BCH Korrigiert

BCH Korrigiert 3 Einzelfehler, ein Bündelfehler bis zur Länge 3

RS .... 2 ..... 5...8

Fazit: BCH besser als RS bei Einzelfehlern

RS ..... BCH ... Bündelfehlern

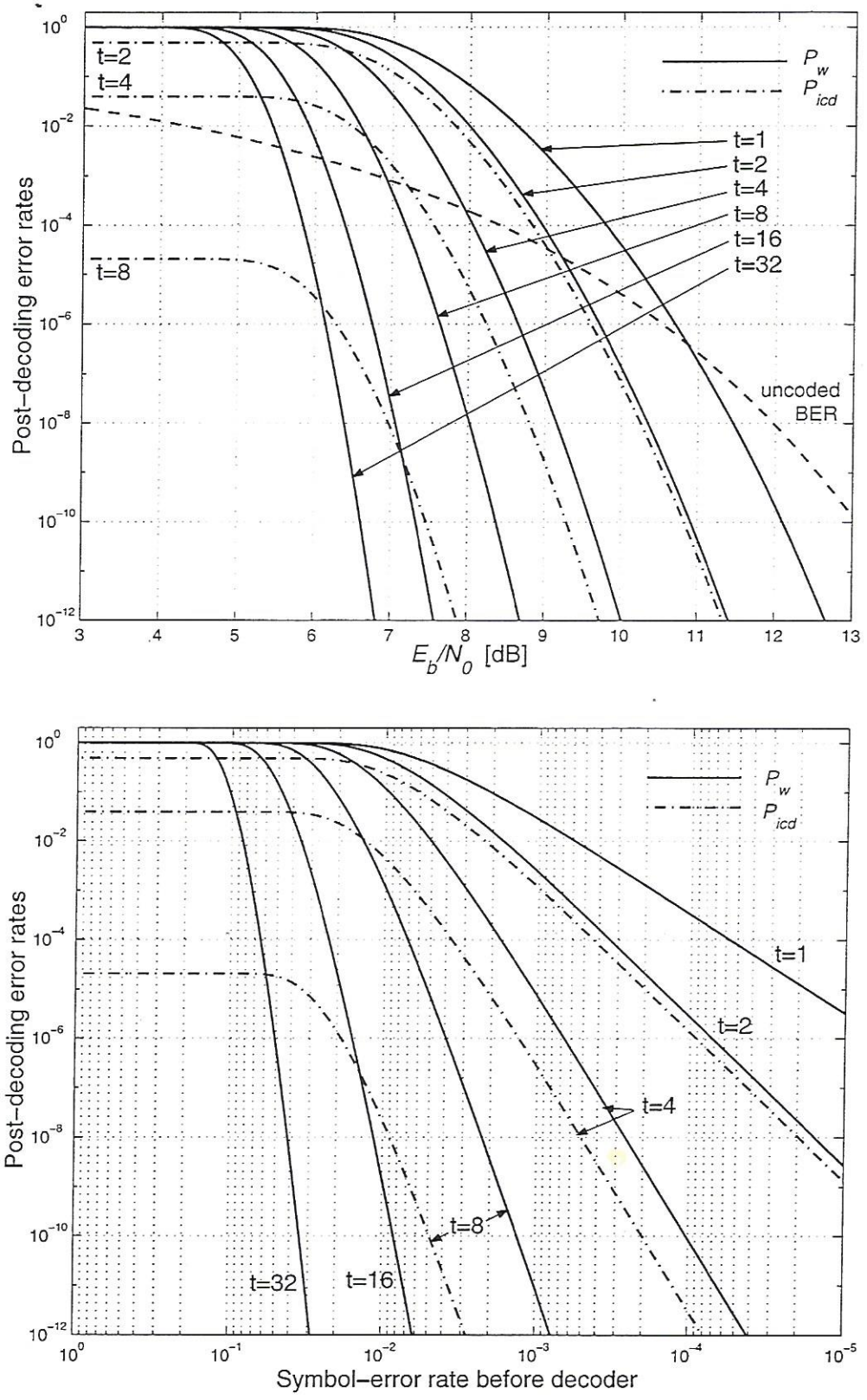


Figure 8.4. P_w and P_{icd} of $(255, 255-2t, 2t+1)_{256}$-RS codes for several t

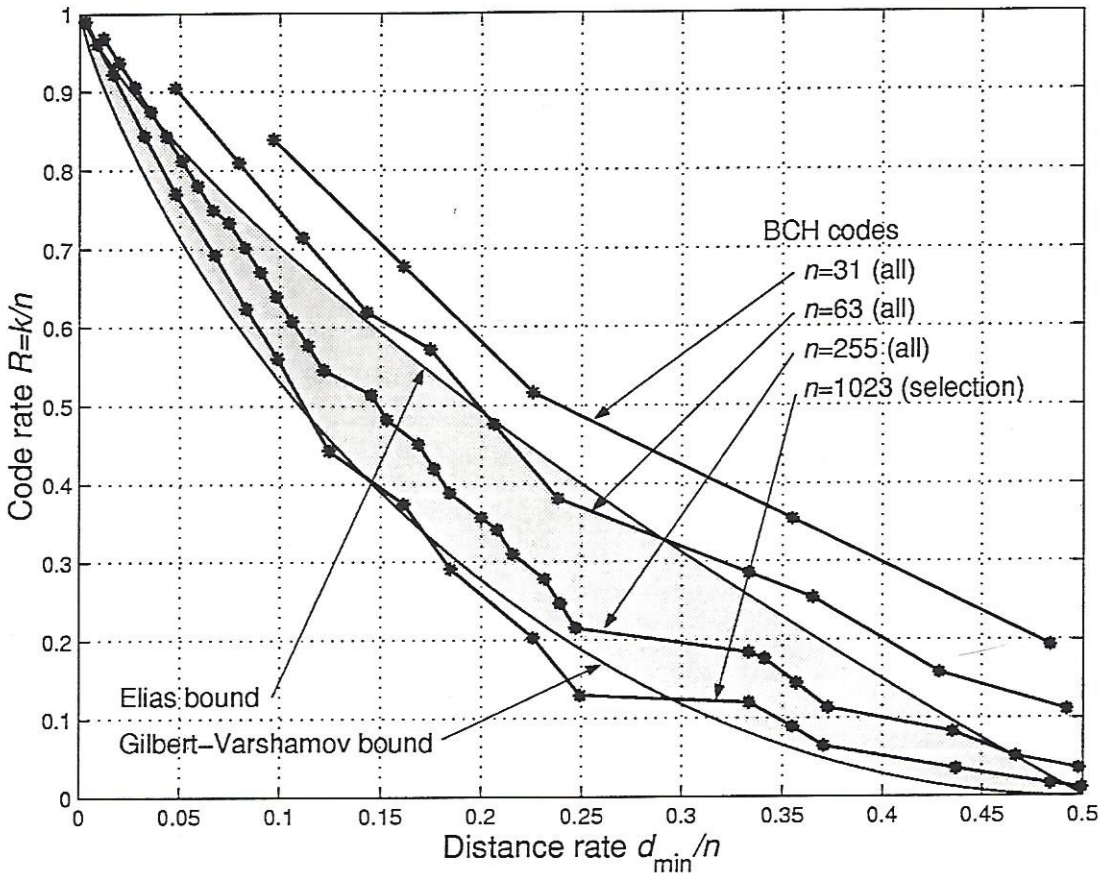


Figure 8.15. Comparison of some BCH codes with upper and lower asymptotic bounds

Asymptotically bad codes with $d/n \to 0$ at a constant rate and $n \to \infty$ may, of course, still have a coding gain going toward infinity, since

$$G_{\text{a,hard}} \approx 10 \cdot \log_{10}\left(\frac{R}{2} \cdot \frac{d}{n} \cdot n\right) \to \infty \quad \text{as} \quad n \to \infty$$

is possible, if d/n converges very slowly toward zero.

Although BCH codes are asymptotically bad their practical advantages still dominate: BCH codes exist for many parameter values, they are more powerful for short and medium block lengths than any other known code family, the costs of encoding and especially of decoding are relatively small. However, the disadvantages are:

- BCH codes can only be corrected up to half the designed distance by the usual, less time consuming decoding methods. It is of no use for decoding if the actual minimum distance is bigger. ML decoding is usually impossible, although it would mean enormous improvements [?].
- Furthermore the usual decoding methods only compute hard decisions. So the expected gain of 2 to 3 dB for soft decisions can not be achieved. For

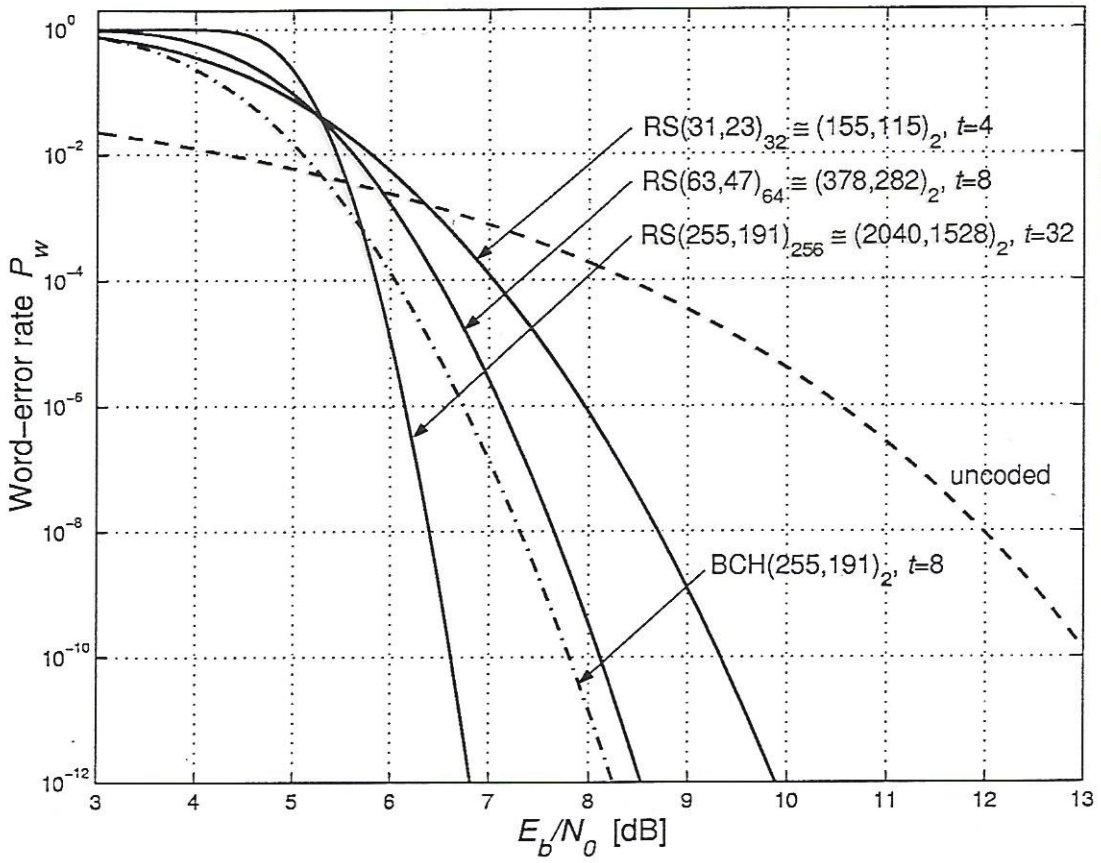


Figure 8.14. Comparison of a BCH code with some RS codes (all with $R \approx 3/4$)

Now, we will examine the influence block lengths have on the behaviour of BCH codes by taking a closer look at Tables 8.3 and 8.4. According to Table 8.3 the asymptotic coding gain $G_{\text{a,hard}} = 10 \cdot \log_{10}(R(t+1))$ grows with increasing block length, as expected on the grounds of Figures 8.9 to 8.11. Yet, at the same time the distance rate d/n decreases, as shown in Table 8.4. The distance rate of RS codes is much larger with $d_{\min} = 1 - R + 1/n \approx 1 - R$, and thus lies on the asymptotic Singleton bound, which is obvious since RS codes, being MDS codes, satisfy the equality of the Singleton bound.

Figure 8.15 shows a comparison of the binary BCH codes with the asymptotic upper Elias bound and the asymptotic lower Gilbert-Varshamov (GV) bound of Figure 4.6. For the block lengths 31, 63 and 255 all BCH codes are listed, however, for 1023 only a small number of representative BCH codes is given. Short BCH codes lie way above the GV bound, and for $n = 31$ even above the upper Elias bound (which is, of course, no contradiction, since for a small n, codes can have a quite different behaviour than in the asymptotic case of $n \to \infty$). For a bigger n the distance rate comes close to the GV bound and for $n > 1023$ even goes below the GV bound. So even in this representation BCH codes turn out to be asymptotically bad.

Kürzung von Codes allgemein (§4.5)

$(n, k, d)_q \rightarrow (n', k', d')_q$

$n - k = n' - k'$ $\quad n' < n,\ k' < k$

$d' \geq d$

- Beispiel RS mit $m = 8,\ t = 2$

  Basiscode $(255, 251, 5)_{256}$

  Gekürzte Codes $(28, 24, 5)_{256}$, $(32, 28, 5)_{256}$ } Compact Disk

- Alle Codes können
  - 2 Symbolfehler korrigieren
  - 1 Symbolfehler korrigieren + 0 Auslöschungen korrigieren
  - 0 + 2 Auslöschungen korrigieren
    + 4 Auslöschungen korrigieren
  - 4 Symbolfehler erkennen

Reed-Solomon (RS) codes

$$(n, k, d_{min})_q = (p^m - 1,\ p^m - 1 - 2t,\ 2t+1)_{p^m}$$
$$= (q-1,\ n-2t,\ d_{min})_q$$

also $n = q - 1 = p^m - 1$

$n - k = (q-1) - (n-2t) = 2t = d_{min} - 1$

Fehler- und Ausfallkorrektur für RS codes

Kanalmodell: $A_{in} = \mathbb{F}_q$ $\quad |A_{in}| = q$

$A_{out} = \mathbb{F}_q \cup \{?\}$ $\quad |A_{out}| = q+1$

$$y = a + e + v$$

y: bekannt, Empfangswort — a: unbekannt, Codewort — e: unbekannt, Fehlerwort — v: bekannt, Ausfallwortstelle (nimmt 0 oder ? an)

Mit ? werden bekannte Ausfälle markiert, d.h. von diesen Symbolen im Empfangswort macht der Decoder keinen Gebrauch

Theorem 7.9: Es werden τ Fehler und τ_v Ausfälle korrigiert, wenn gilt

$$2\tau + \tau_v \overset{!}{\leq} 2t = d_{min} - 1 = n - k$$

Compact Disc (CD)

Ziele: einfache Herstellung (Einzelfehler)
robust gegen Beschädigungen (Bündelfehler)
einfache CD-Player (Synchronisation)

Pit/Land Abfolge

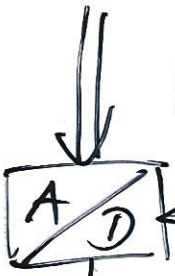
1-3 µm

Daten: Spurlänge = 5 km
Spurbreite = 0.6 µm
Spurabstand = 1.0 µm
Geschwindigkeit = 1.2 m/s (76 min)
"Kanalbit"länge = 0.3 µm

$\Rightarrow$ $13 \cdot 10^9$ Kanalbits auf einer CD
$4.3 \cdot 10^6$ Kanalbits/s

Bündelfehler von 1 mm betrifft $\approx$ 3000 falsche Kanalbits
" 7 " " 21 000 " "

Infobits insgesamt = 1.4 Mbit/s · 74 min = 6.2 Gbit
= 640 MByte



Stereosignal (bis 20 kHz)

44.1 kHz, 16 bit

Infobits 1,4112 Mbit/s

1 Abtastwert = 32 bit

6 Abtastwerte = 192 bit $\in \mathbb{F}_2^{192}$

= 24 · 8 bit $\in \mathbb{F}_{256}^{24}$

CIRC Encoder $R = 3/4 = \frac{24}{28} \cdot \frac{28}{32}$

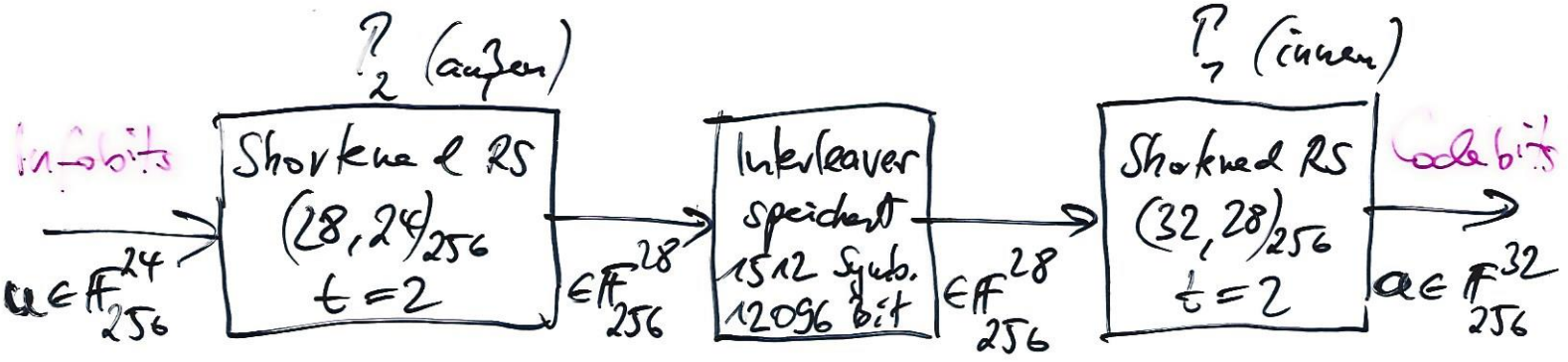
Codebits 1,8816 Mbit/s

6 Abtastwerte $\hat{=}$ 32 · 8 bit $\in \mathbb{F}_{256}^{32}$

EFM-Modulator $R = \frac{256}{588}$ (addiert Synchro, vermeidet Dauerlagen)

Kanalbits 4,3218 Mbit/s

CIRC-Encoder (Cross-Interleaved Reed-Solomon Code)



Korrektur von 4 Ausfällen

Nur Fehlererkennung (ev. 1 Fehler korrigieren)

Korrigiert 9408 Kanalbits (≙ 3mm)

≙ 40% Codebits (= 16·32·8/)

≙ 16 Komplett falsche P_1 – Codewörter, aufeinanderfolgend

Methode:

- In P_1 wird jedes verfälschte Wort mit hoher W. als falsch erkannt
- Jedes der 16 verfälschten Wörter wird mit 28 Ausfällen klassifiziert, insgesamt $16 \cdot 28 = 448$ Symbol-Ausfälle.
- Durch Interleaver werden die $16 \cdot 28$ Symbole auf 112 P_2-Wörter mit jeweils 4 Ausfällen verteilt
- P_2 korrigiert jedes der 112 Wörter korrekt

Modifikationen

Korrektur eines Fehlers in $\hat{?}_1$

$\Rightarrow$ Einzelfehler führen nicht zur Klassifizierung eines ganzen Wortes als ausgefallen

$\hat{?}_2$ wird entlastet

Sicherheit der Bündelfehler-Erkennung sinkt

Größerer Interleaver

$\Rightarrow$ Noch längere Bündelfehler sind korrigierbar, setzt aber größere fehlerfreie Zone zwischen den Bündelfehlern voraus.