Unterstützende Materialien zur Vorlesung

Verfahren zur Kanalcodierung – Teil 2

Prof. Dr. Bernd Friedrichs
KIT CEL

Inhalt

- Der Begriff des Codierungsgewinns
- Shannon Kanalcodierungstheorem, R_0-Theorem und Schlussfolgerungen
- Lineare Blockcodes
- Fehlererkennung und Fehlerkorrektur
- Decoder-Strategien (MLD, BMDD)

Die *Bit-Fehlerwahrscheinlichkeit* (BER, Bit Error Rate) bzw. *Symbol-Fehlerwahrscheinlichkeit* $P_b = P(\hat{a}_i \neq a_i)$ bezieht sich nur auf die Infosymbole und berücksichtigt nicht die Prüfsymbole. P_b und die Wort-Fehlerwahrscheinlichkeit $P_w = P(\hat{\boldsymbol{a}} \neq \boldsymbol{a})$ hängen in komplizierter Weise zusammen. Da die Anzahl der Fehler in einem fehlerhaft decodierten Wort zwischen 1 und k beträgt, gilt folglich

$$\frac{1}{k} \cdot P_w \leq P_b \leq P_w. \tag{1.7.1}$$

Normalerweise erweist sich die Näherung

$$P_b \approx \frac{d_{\min}}{k} \cdot P_w \tag{1.7.2}$$

als sinnvoll, die von $d_{\min}$ Fehlern pro falsch decodiertem Wort ausgeht.

Der Vergleich von Codes untereinander und mit der uncodierten Übertragung erfolgt oft anhand des AWGN-Kanalmodells mit $q = 2$ gemäß Definition 1.3. Dazu sei N_0 die einseitige Rauschleistungsdichte und E_b die Energie pro Infobit. Dann ist

$$E_c = R \cdot E_b \tag{1.7.3}$$

die Energie pro Codebit, die also um den Faktor R (Coderate) kleiner ausfällt, sofern man gleiche Sendeleistung unterstellt. Die Signalleistung muß dann aufgeteilt werden auf die Infobits und auf die Prüfbits, so daß pro Codebit weniger Energie verfügbar ist. Damit wächst die Wahrscheinlichkeit, daß die Codebits im Demodulator falsch bestimmt werden. Ein Codierungsgewinn ergibt sich nur, wenn die Korrekturfähigkeit des Codes diesen negativen Effekt überwiegt.

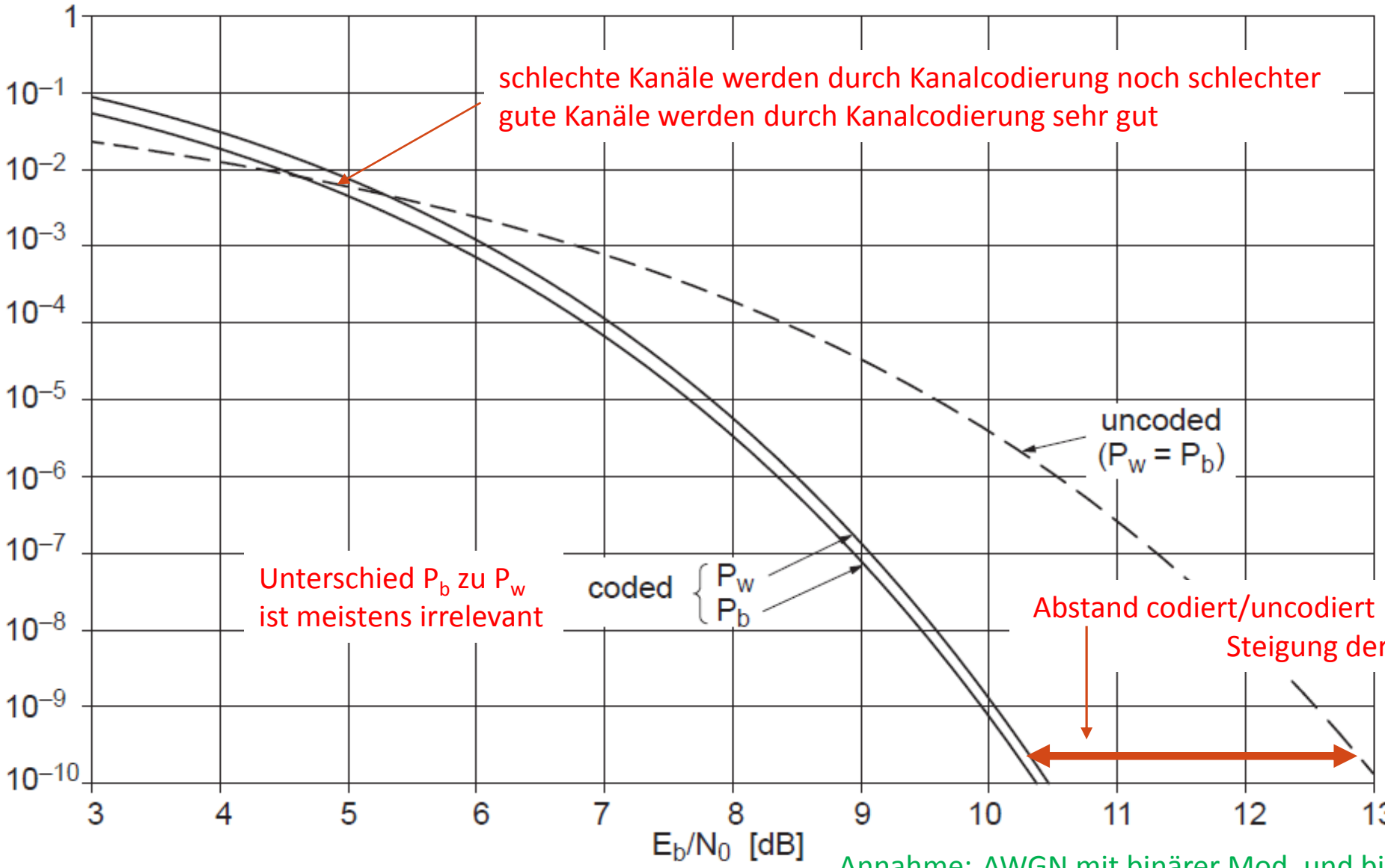


Bild 1.10. Fehlerwahrscheinlichkeit des $(23, 12)_2$-Golay-Codes (bei Hard-Decision)

In den Bildern 1.10 und 1.11 erfolgt nun anhand des AWGN-Modells ein quantitativer Vergleich zwischen codierter und uncodierter Übertragung, wobei P_w und P_b über E_b/N_0 aufgetragen werden. Die Kurve für die uncodierte Übertragung entspricht direkt Tabelle 1.1. Für die codierte Übertragung werden zwei *perfekte Codes* verwendet, bei denen als wesentlicher Vorteil die Wort-Fehlerwahrscheinlichkeit exakt berechnet werden kann (siehe dazu Satz 3.15).

Bild 1.10 zeigt am Beispiel des $(23, 12)_2$-Golay-Codes das prinzipielle Verhalten kanalcodierter Übertragungssysteme. Bei schlechten Kanälen ist die uncodierte Übertragung zunächst besser. Es gibt dann eine Schwelle (die hier bei etwa 5 dB liegt), von der an die codierte Übertragung besser wird und zu einem Codierungsgewinn (in dB) führt, der immer auf eine bestimmte Fehlerwahrscheinlichkeit P_w oder P_b bezogen wird. Bei $P_b = 10^{-6}$ beträgt dieser Codierungsgewinn etwa 2,0 dB. Unterhalb der Schwelle ergibt sich bei gleichem Signal/Rausch-Verhältnis eine kleinere Fehlerwahrscheinlichkeit bzw. bei gleicher Fehlerwahrscheinlichkeit kann die Sendeleistung reduziert werden. Bei einem sehr guten Kanal mit $E_b/N_0 \to \infty$ verlaufen die Kurven nahezu parallel und werden gleichzeitig immer steiler. Der Unterschied zwischen P_w und P_b ist dabei unbedeutend.

Der Abstand zwischen codierter und uncodierter Übertragung wird nicht beliebig groß bei $E_b/N_0 \to \infty$, sondern konvergiert gegen einen Grenzwert, der jetzt berechnet werden soll. Zur Unterscheidung wird deshalb die Energie pro Infobit bei der uncodierten Übertragung mit E_b' und bei der codierten Übertragung mit E_b bezeichnet.

Für die uncodierte Übertragung ergibt sich P_b direkt als BSC-Bitfehlerwahrscheinlichkeit. Nach (1.3.12) und (A.3.18) gilt:

$$P_{b,\text{unc}} = p_e = Q\left(\sqrt{\frac{2E_b'}{N_0}}\right) \approx \text{const} \cdot e^{-E_b'/N_0}. \tag{1.7.4}$$

Wie später in Satz 3.15 noch gezeigt wird, gilt für die codierte Übertragung mit Hard-Decision für großes E_b/N_0 näherungsweise:

$$P_{w,\text{cod}} \approx \text{const} \cdot p_e^{t+1}. \tag{1.7.5}$$

Dabei ist const eine vom Code abhängige Konstante, p_e ist die BSC-Bitfehlerwahrscheinlichkeit der Codebits zu $E_c = RE_b$ und $t = \lfloor (d_{\min} - 1)/2 \rfloor$ entspricht etwa der halben Minimaldistanz (mit $\lfloor \lambda \rfloor$ wird die größte ganze Zahl $\leq \lambda$ bezeichnet). Nach (1.7.2) und (A.3.18) gilt also:

$$\begin{aligned} P_{b,\text{cod}} &\approx \text{const} \cdot P_{w,\text{cod}} \\ &\approx \text{const} \cdot p_e^{t+1} \\ &\approx \text{const} \cdot \left[Q\left(\sqrt{\frac{2RE_b}{N_0}}\right)\right]^{t+1} \\ &\approx \text{const} \cdot e^{-R(t+1)\cdot E_b/N_0}. \end{aligned} \tag{1.7.6}$$

Der Codierungsgewinn wird auf gleiche Bitfehlerraten bezogen: $P_{b,\text{unc}} = P_{b,\text{cod}}$. Hieraus folgt:

$$\text{const} \cdot e^{-E_b'/N_0} = \text{const} \cdot e^{-R(t+1)\cdot E_b/N_0}. \tag{1.7.7}$$

Die Konstanten sind allenfalls linear von E_b/N_0 bzw. t abhängig und können somit beim Logarithmieren für großes E_b/N_0 vernachlässigt werden:

$$\frac{E_b'}{N_0} \approx R(t+1) \cdot \frac{E_b}{N_0}. \tag{1.7.8}$$

Daraus ergibt sich der *asymptotische Codierungsgewinn* (asymptotic coding gain) für Hard-Decision als:

$$G_{\text{a,hard}} = 10 \cdot \log_{10}(R(t+1)) \quad \text{dB}. \tag{1.7.9}$$

Für *Soft-Decision* wird später in Satz 3.17 gezeigt:

$$P_{w,\text{cod}} \approx \text{const} \cdot e^{-Rd_{\min} \cdot E_b/N_0}. \tag{1.7.10}$$

Der Vergleich mit (1.7.4) ergibt den asymptotischen Codierungsgewinn für Soft-Decision:

$$G_{\text{a,soft}} = 10 \cdot \log_{10}(Rd_{\min}) \quad \text{dB}. \tag{1.7.11}$$

Durch Soft-Decision ergibt sich prinzipiell ein asymptotischer Gewinn von etwa 3 dB (mit $t+1 \approx d_{\min}/2$):

$$G_{\text{a,soft}} \approx G_{\text{a,hard}} + 3{,}01 \text{ dB}. \tag{1.7.13}$$

Bei "realistischen" Werten von E_b/N_0 bzw. "mittleren" Werten von P_b beträgt der Gewinn durch Soft-Decision allerdings üblicherweise nur rund 2 dB.

Golay-Code $(23,12,7)_2$ $G_{a,hard} = 10*\log_{10}(12/23*(3+1))$ = 3.20 dB

Hamming-Code $(7,4,3)_2$ $G_{a,hard} = 10*\log_{10}(4/7*(1+1))$ = 0.58 dB

$G_{a,soft} = 10*\log_{10}(4/7*3)$ = 2.34 dB

Wiederholungs-Code $(n,1,n)_2$ $G_{a,hard} \approx 10*\log_{10}(1/n*n/2)$ = -3.01 dB !!!

Der Wiederholungscode hat zwar die bestmögliche Minimaldistanz,
aber in Kombination mit der kleinstmöglichen Coderate führt das zu einem verheerenden Codierungsgewinn.

Fazit:

- Die isolierte Maximierung von d_{min} oder R ist sinnlos,
- sondern das Produkt $d_{min}*R$ ist zu maximieren durch intelligente Wahl des Codes

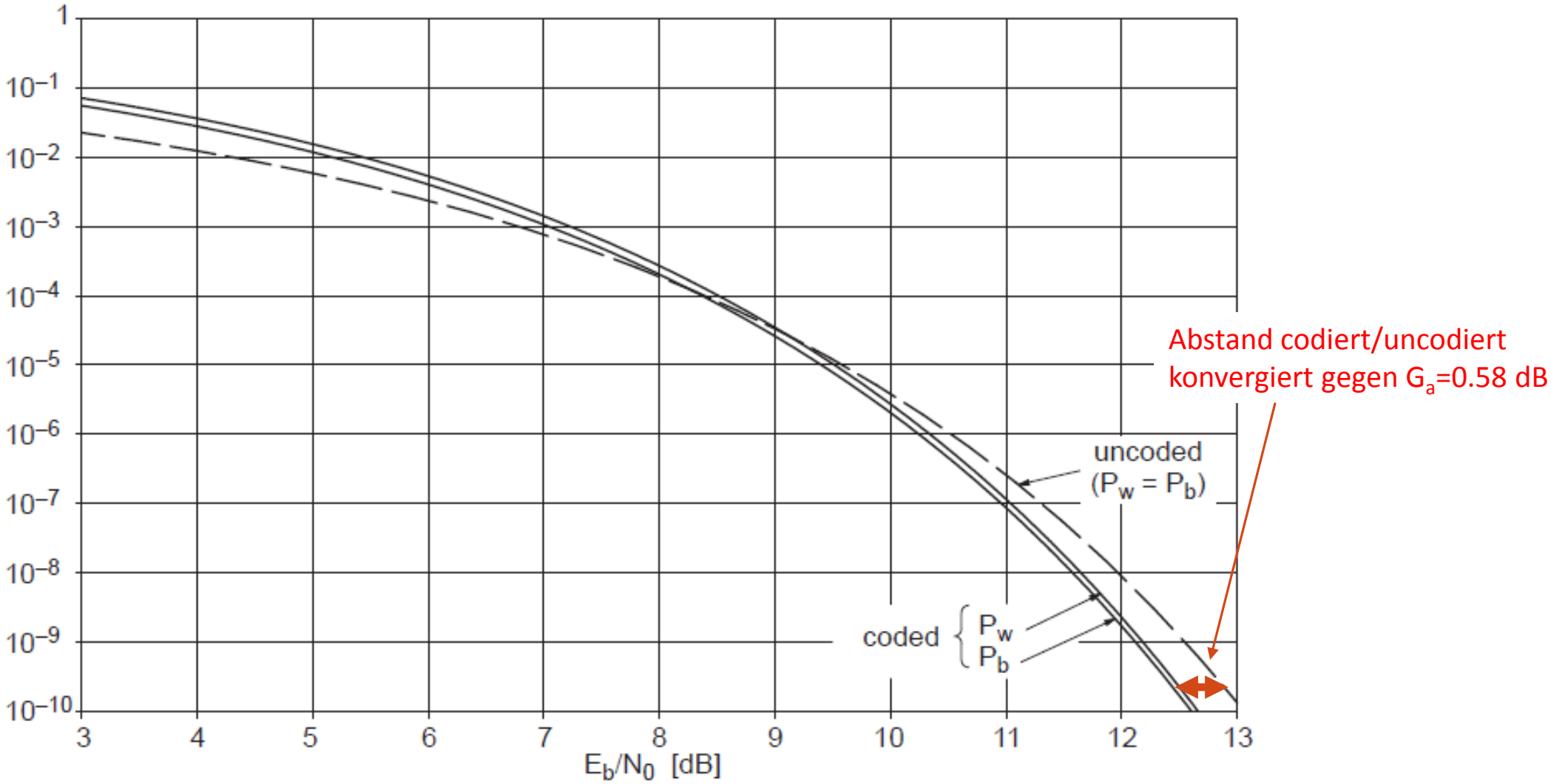


Bild 1.11. Fehlerwahrscheinlichkeit des $(7, 4)_2$-Hamming-Codes (bei Hard-Decision)

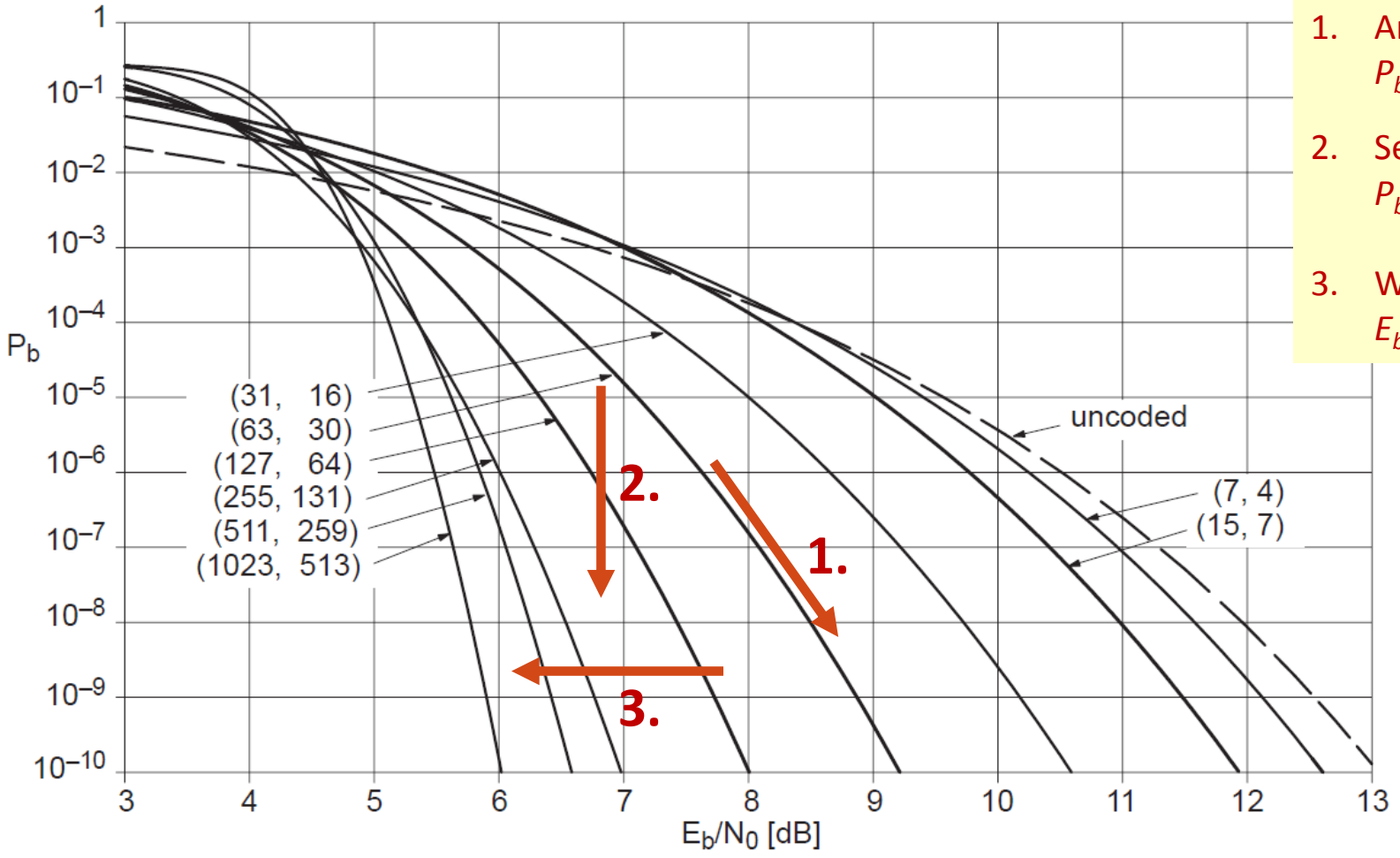


Bild 7.1. Fehlerwahrscheinlichkeit der $R = 1/2$-BCH-Codes

Potenzielle Trade-offs (am Beispiel einer Code-Familie)

1. An der Kurve entlang nach unten:
   P_b wird reduziert durch größeres E_b/N_0 bei n=const
2. Senkrecht nach unten:
   P_b wird reduziert durch größeres n bei E_b/N_0=const
3. Waagerecht nach links:
   E_b/N_0 wird reduziert durch größeres n bei P_b=const

Insbesondere mit Blockcodes wird unmittelbar der Grundgedanke deutlich, der hinter der Kanalcodierung steht. Es ist die gleiche Idee wie bei der Digitalisierung in der Nachrichtentechnik mit ähnlichen Vor- und Nachteilen:

Digitalisierung bedeutet Quantisierung der Symbole im Wertebereich: Wenn Δ die Quantisierungsbreite ist, so ist $\Delta/2$ der maximale Quantisierungsfehler. Daraus folgt:

Vorteil: Kleine Übertragungsfehler (kleiner als $\Delta/2$) werden völlig eliminiert, während bei der analogen Übertragung in jeder Verstärkerstufe das Signal/Rausch-Verhältnis prinzipiell immer schlechter wird.

Nachteil: Auch ohne Übertragungsfehler tritt immer ein mittlerer Quantisierungsfehler von $\sqrt{\Delta^2/12}$ auf. Große Übertragungsfehler (größer als $\Delta/2$) werden durch Zuordnung zur falschen Quantisierungsstufe noch vergrößert.

Fazit: Quantisierung nur dort, wo der Hauptanteil der Störungen kleiner als $\Delta/2$ ist.

Kanalcodierung bedeutet Quantisierung ganzer Symbolfolgen im Zeitbereich: Die $n-k$ Prüfstellen werden so gewählt, daß sich die verschiedenen Codewörter der Länge n an mindestens $d_{\min}$ Stellen unterscheiden. Das ist möglich, weil von den q^n möglichen Wörtern nur q^k Wörter auch tatsächlich Codewörter sind. Daraus folgt (wie in Abschnitt 3.2 noch detailliert gezeigt wird):

Vorteil: Bei weniger als $d_{\min}/2$ Übertragungsfehlern liegt das Empfangswort näher am gesendeten Wort als an allen anderen möglichen Codewörtern und kann somit richtig decodiert werden.

Nachteil: Bei mehr als $d_{\min}/2$ Übertragungsfehlern wird möglicherweise auf ein falsches Codewort entschieden und die Anzahl der Fehler erhöht sich durch die Codierung.

Fazit: Kanalcodierung nur dort, wo in der Mehrzahl weniger als $d_{\min}/2$ Fehler pro Codewort auftreten, d.h.: Kanalcodierung ist nur sinnvoll bei relativ guten Kanälen und bei hohen Anforderungen an die Zuverlässigkeit, während bei schlechten Kanälen eine Kanalcodierung nicht sinnvoll ist.

Die zentrale Idee der Kanalcodierung ist es, lange Infoblöcke in noch längere Codeblöcke zu transferieren. Das Ausmaß der hinzuzufügenden Redundanz ist abhängig von der Kanalqualität und der gewünschten Übertragungsqualität.

Eine weitere zentrale Idee bei Fading-Kanälen besteht darin, die Information zeitlich so weit zu verspreizen dass schlechte Zeitabschnitte durch gute Zeitabschnitte kompensiert werden können. Praktisch wird das aber durch Verarbeitungsaufwand und zulässige Verzögerungszeit begrenzt.

Definition 2.1. *Zwischen zwei Zufallsgrößen und insbesondere zwischen Input und Output des DMC kann die Transinformation (mutual information) definiert werden, die angibt, wieviel Information eine der beiden Zufallsgrößen über die andere vermittelt:*

$$I(x;y) = \sum_{x\in\mathcal{A}_{\text{in}},y\in\mathcal{A}_{\text{out}}} P(x)P(y|x)\log_2\frac{P(y|x)}{\sum_{x'\in\mathcal{A}_{\text{in}}}P(x')P(y|x')} \tag{2.1.6}$$

$$= \sum_{x\in\mathcal{A}_{\text{in}},y\in\mathcal{A}_{\text{out}}} P(x,y)\log_2\frac{P(x,y)}{P(x)P(y)} \tag{2.1.7}$$

$$= \underbrace{\sum_{y\in\mathcal{A}_{\text{out}}} P(y)\log_2\frac{1}{P(y)}}_{=H(y)} - \underbrace{\sum_{x\in\mathcal{A}_{\text{in}},y\in\mathcal{A}_{\text{out}}} P(x,y)\log_2\frac{1}{P(y|x)}}_{=H(y|x)} \tag{2.1.8}$$

$$= \underbrace{\sum_{x\in\mathcal{A}_{\text{in}}} P(x)\log_2\frac{1}{P(x)}}_{=H(x)} - \underbrace{\sum_{x\in\mathcal{A}_{\text{in}},y\in\mathcal{A}_{\text{out}}} P(x,y)\log_2\frac{1}{P(x|y)}}_{=H(x|y)}. \tag{2.1.9}$$

I(x;y) ist eine Funktion von $P_{y|x}$ und P_x
also ist C nur noch abhängig von $P_{y|x}$

Definition 2.2. *Für den DMC $(\mathcal{A}_{\text{in}}, \mathcal{A}_{\text{out}}, P_{y|x})$ ist die Kanalkapazität C definiert als das Maximum der Transinformation, wenn über alle möglichen Quellenstatistiken (Apriori-Wahrscheinlichkeiten) maximiert wird:*

$$C = \max_{P_x} I(x;y) \qquad \textit{Einheit: Infobit/Kanalbenutzung.} \tag{2.1.13}$$

Es gilt $0 \le C \le \log_2 q$ ($= 1$ bei binärer Übertragung). Bei $C = 0$ ist der Output vom Input statistisch unabhängig und es ist keine Information übertragbar. Bei $C = \log_2 q$ schaltet der Kanal transparent durch.

$$C = 1 + p_e \log_2(p_e) + (1 - p_e) \log_2(1 - p_e)$$

$$R_0 = 1 - \log_2\left(1 + \sqrt{4p_e(1 - p_e)}\right)$$

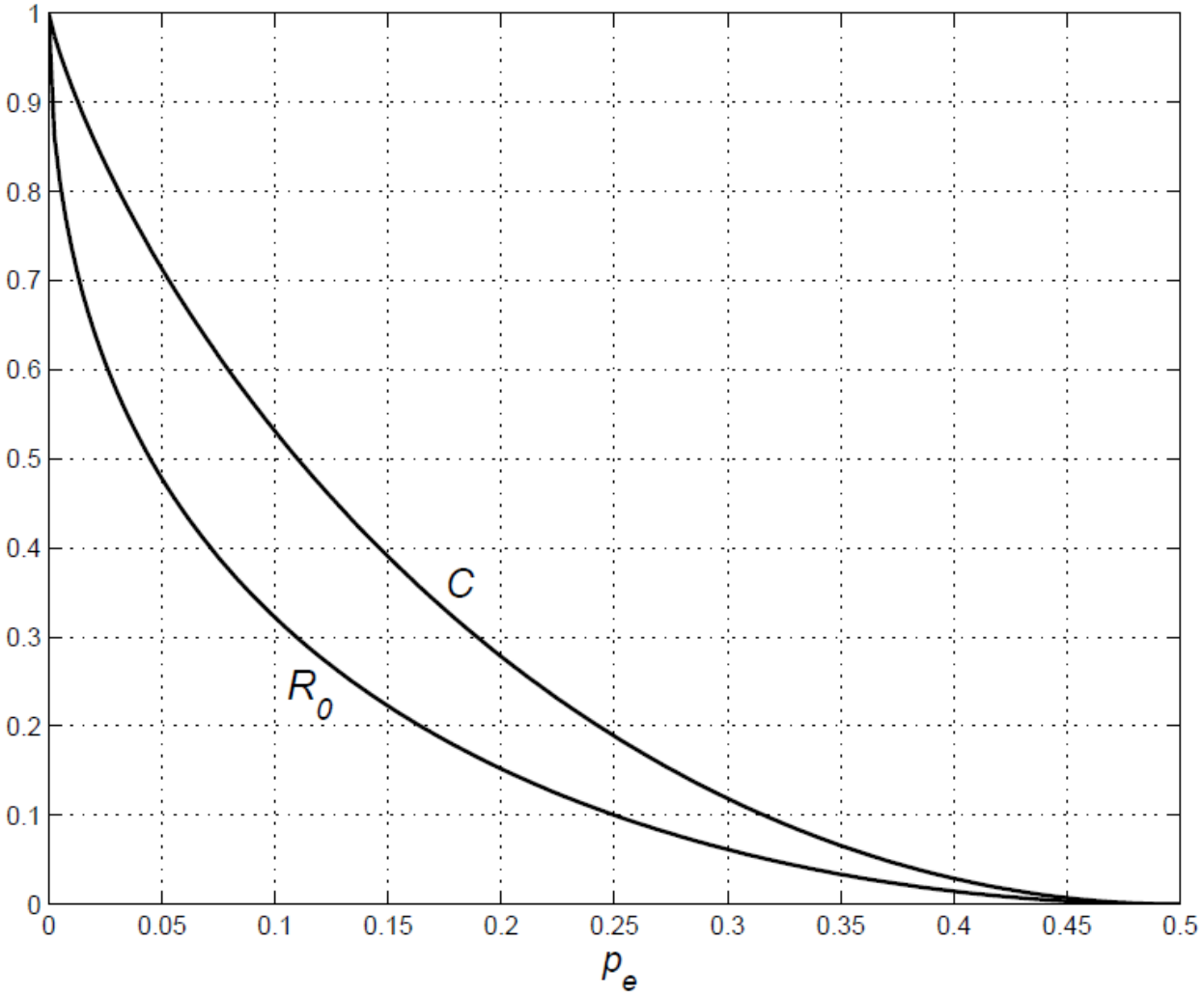


Bild 2.1. C und R_0 für den BSC

$$C = \frac{1}{2\sqrt{2\pi}} \int_{-\infty}^{\infty} \left(e^{-(\alpha - v)^2/2} \log_2 \frac{2}{1 + e^{-2\alpha v}} + e^{-(\alpha + v)^2/2} \log_2 \frac{2}{1 + e^{2\alpha v}} \right) d\alpha$$

$$R_0 = 1 - \log_2\left(1 + e^{-E_c/N_0}\right)$$

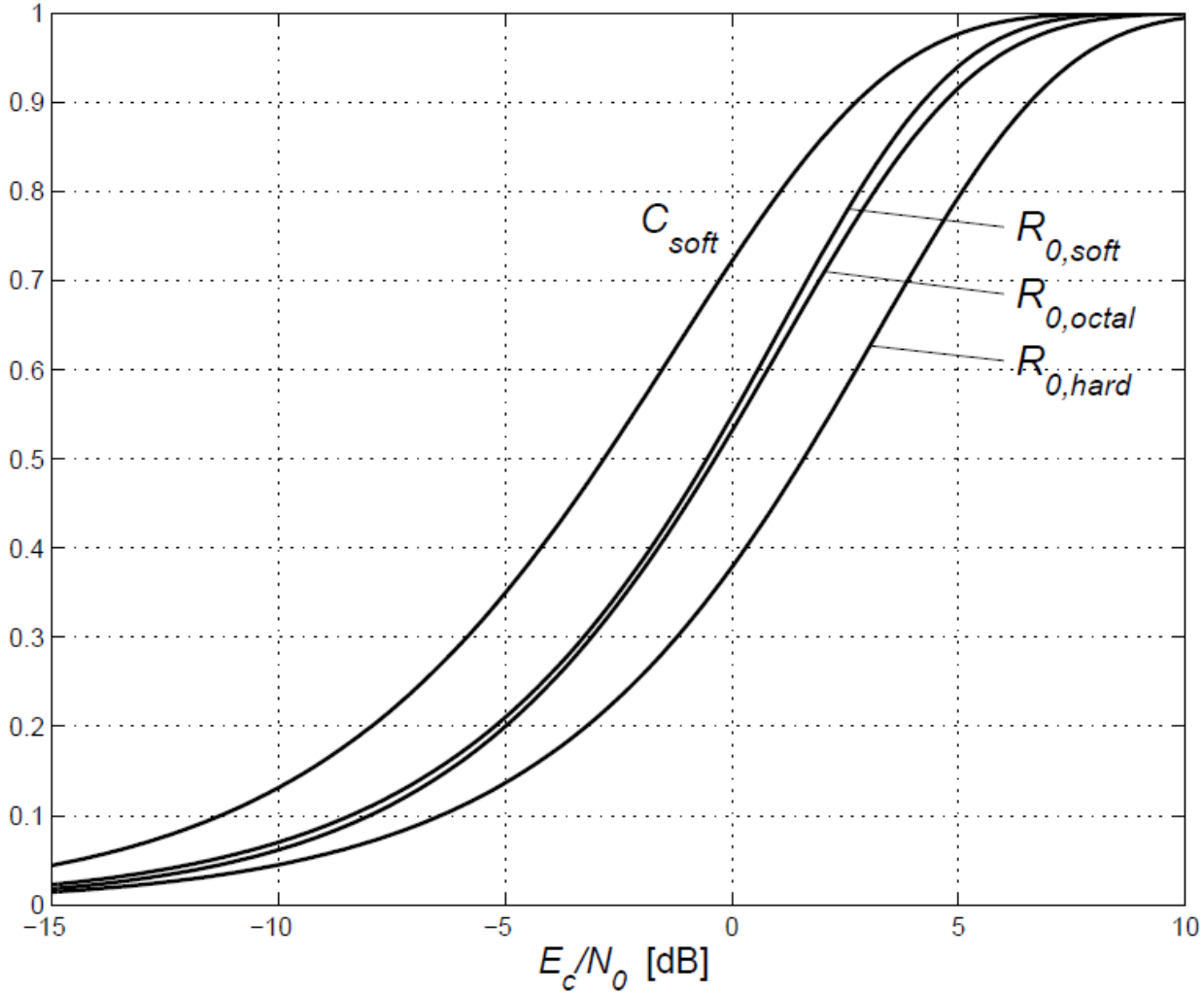


Bild 2.2. C und R_0 für den AWGN ($q = 2$)

Die Bedeutung der Kanalkapazität wird sofort klar aus dem berühmten Satz von Shannon (1948, noisy channel coding theorem), der die Grundlage für die digitale Kommunikation bildet:

Satz 2.1 (Kanalcodierungstheorem, 1.Fassung). *Es sei C die Kanalkapazität des DMC mit $q = |\mathcal{A}_{\text{in}}|$. Dann kann durch Verwendung von Kanalcodierung und Maximum-Likelihood-Decodierung die Wort-Fehlerwahrscheinlichkeit P_w beliebig klein gemacht werden, sofern die Coderate $R_b = R \cdot \log_2 q$ kleiner als C ist.*

Genauere Formulierung: Für jedes $\varepsilon > 0$ und $\varepsilon' > 0$ existiert ein $(n, k)_q$-Blockcode mit $R_b = k/n \cdot \log_2 q$, so daß gilt:

$$C - \varepsilon' \leq R_b < C \quad \textit{und} \quad P_w < \varepsilon.$$

Konverses Theorem: Bei $R_b > C$ kann P_w eine gewisse Grenze auch bei größtem Aufwand nie unterschreiten.

Dieses Ergebnis ist sicherlich überraschend: Die Kanaleigenschaften begrenzen nur die Übertragungsrate und den Durchsatz, aber nicht die Qualität der Übertragung. Bei immer höheren Qualitätsanforderungen muß also nicht die Datenrate reduziert werden oder der Kanal selbst verbessert werden, sondern es muß nur die Blocklänge des Codes und damit die Komplexität erhöht werden.

Speziell $q=2$, $R_b = R$

$$r_c = \frac{\text{Codebit}}{s} = \frac{\text{Kanalbenutzung}}{s}$$ bestimmt Bandbreite

$$r_b = \frac{\text{Infobit}}{s} = \frac{\text{Codebit} \cdot R}{s} = r_c \cdot R$$ = Durchsatz (throughput)

Durch r_c wird C bestimmt

Aus C folgt $R = \frac{k}{n} < C$

Damit folgt $r_b = r_c \cdot R < r_c \cdot C =: C^*$

Einheiten: $\left[\frac{\text{Infobit}}{\text{Kanalben.}}\right]$ $\left[\frac{\text{Infobit}}{s}\right]$

(3) Betrachte einen BSC mit $p_e = 0{,}01$, der mit $r_c = 1\,000\,000$ Bit/s benutzbar ist. Pro Sekunde werden im Mittel 990 000 Bit richtig und 10 000 Bit falsch empfangen. Ohne Kanalcodierung sind selbst wesentlich kleinere Infobitraten als 900 000 Bit/s nicht zuverlässig übertragbar. Die Kanalkapazität beträgt

$$C = 1 + 0{,}01 \cdot \log_2 0{,}01 + 0{,}99 \cdot \log_2 0{,}99 = 0{,}919$$

Bit/Kanalbenutzung bzw. $C^* = 919\,000$ Bit/s. Wenn nun $r_b = 900\,000$ Bit/s gewählt wird mit einer Coderate $R = 0{,}9$, dann ist durch Codierung weniger als 1 Fehler pro Sekunde oder eine noch kleinere Fehlerrate erreichbar. ■

Das Codierungstheorem ist ein reiner Existenzsatz und gibt keine Anleitung, wie die entsprechenden Blockcodes zu konstruieren sind. Shannon hat zum Beweis keine cleveren Codes konstruiert, sondern er hat die Codes einfach zufällig gewählt. Bei diesem sogenannten *Random Coding Argument* wird die Aussage für den Mittelwert über alle Blockcodes bewiesen und es gibt dann trivialerweise mindestens einen Code, der so gut ist wie der Mittelwert. Allerdings darf man hieraus nicht schließen, daß es sehr einfach wäre, entsprechende Codes zu finden. Tatsächlich ist keine binäre Codeklasse bekannt (abgesehen von verketteten Codes), so daß mit einer Folge von Blockcodes wachsender Blocklänge die Fehlerwahrscheinlichkeit gegen Null konvergiert, d.h.: *Fast alle Codes sind gut mit Ausnahme derjenigen, die wir kennen.*

Die Kanalkapazität C ist eine theoretische Schranke, von der die praktisch angewendeten bzw. realisierbaren Codierungsverfahren deutlich entfernt sind. Dagegen ist der sogenante R_0-Wert [37, 108, 123] mit vernünftigem Aufwand erreichbar,

Definition 2.3. *Der Wert des Fehlerexponenten an der Stelle $R_b = 0$ wird als R_0-Wert bezeichnet:*

$$R_0 = E_r(0) = \max_{P_x} \left[-\log_2 \sum_{y \in \mathcal{A}_{\text{out}}} \left(\sum_{x \in \mathcal{A}_{\text{in}}} P(x) \sqrt{P(y|x)} \right)^2 \right]. \qquad (2.3.1)$$

Die für R_0 auch übliche Bezeichnung *computional cut-off rate* oder R_{comp}

BSC: $R_0 = 1 - \log_2 \left(1 + \sqrt{4p_e(1-p_e)}\right)$

AWGN: $R_0 = 1 - \log_2 \left(1 + e^{-E_c/N_0}\right)$

Satz 2.3 (R_0-Theorem). *Für den DMC mit R_0 gilt: Es existiert immer ein $(n,k)_q$-Blockcode mit der Coderate $R_b = k/n \cdot \log_2 q < R_0$, so daß bei Maximum-Likelihood-Decodierung für die Wort-Fehlerwahrscheinlichkeit P_w gilt:*

$$P_w < 2^{-n(R_0 - R_b)}. \qquad (2.3.2)$$

Ähnlich wie beim Kanalcodierungstheorem kann hierbei eine Coderate R_b erreicht werden, die beliebig dicht an R_0 liegt.

Je näher also R_b an R_0 heranrückt, desto größer muß die Blocklänge n gewählt werden, um noch die gleiche Fehlerrate zu garantieren.

Betrachte den Grenzfall $R = C$ bzw. R_0

C bzw. R_0 $= \text{Funktion}\left(\frac{E_c}{N_0}\right)$ bzw. $= \text{Funktion}(p_E)$

$= \text{Funktion}\left(R \cdot \frac{E_b}{N_0}\right)$ $= \text{Funktion}(\text{Funktion}(\frac{E_c}{N_0}))$

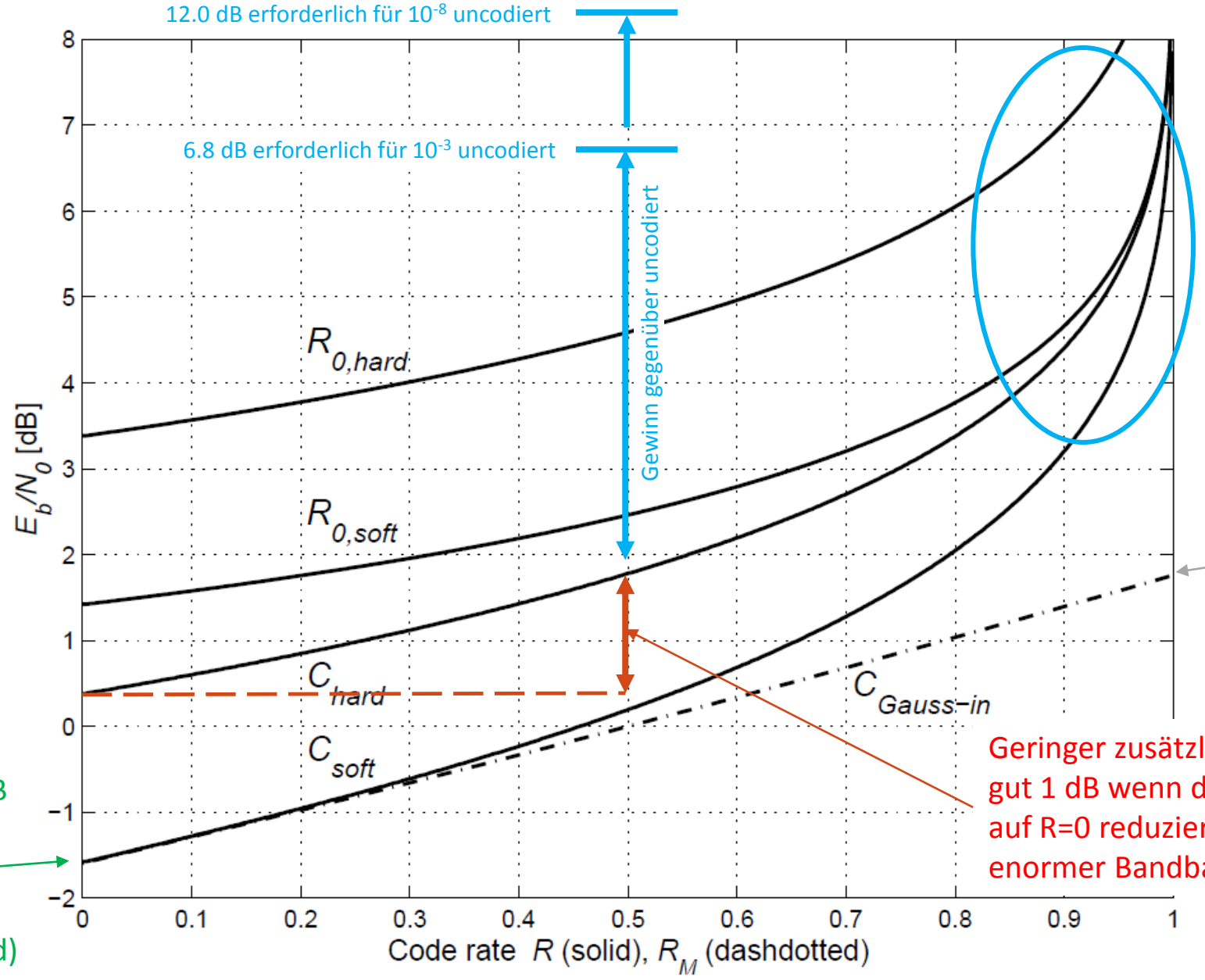
Mit $R = C$ bzw. R_0 ergibt sich $R = \text{Funktion}\left(R \cdot \frac{E_b}{N_0}\right)$,

also ein Zusammenhang zwischen R und $\frac{E_b}{N_0}$

Beispiel $R = R_{0,\text{soft}} = 1 - \log_2\left(1 + e^{-R E_b/N_0}\right)$

$$\rightarrow \quad \frac{E_b}{N_0} = -\frac{\ln\left(2^{1-R} - 1\right)}{R} \quad \rightarrow 2 \cdot \ln 2 \text{ für } R \rightarrow 0$$

Dieser Grenzwert kann auch aus Bild 2.4 abgelesen werden. Fazit: Für E_b/N_0 kleiner als 1,42 dB ist keine Übertragung mit $R = R_0$ möglich! Die Ursache kann wie folgt erklärt werden: Eine sehr kleine Coderate R bedeutet eine sehr große erforderliche Bandbreite (was nebenbei bemerkt praktisch sowieso unrealistisch ist) und damit eine sehr geringe Energie pro Codebit gegenüber der Rauschleistungsdichte. 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.



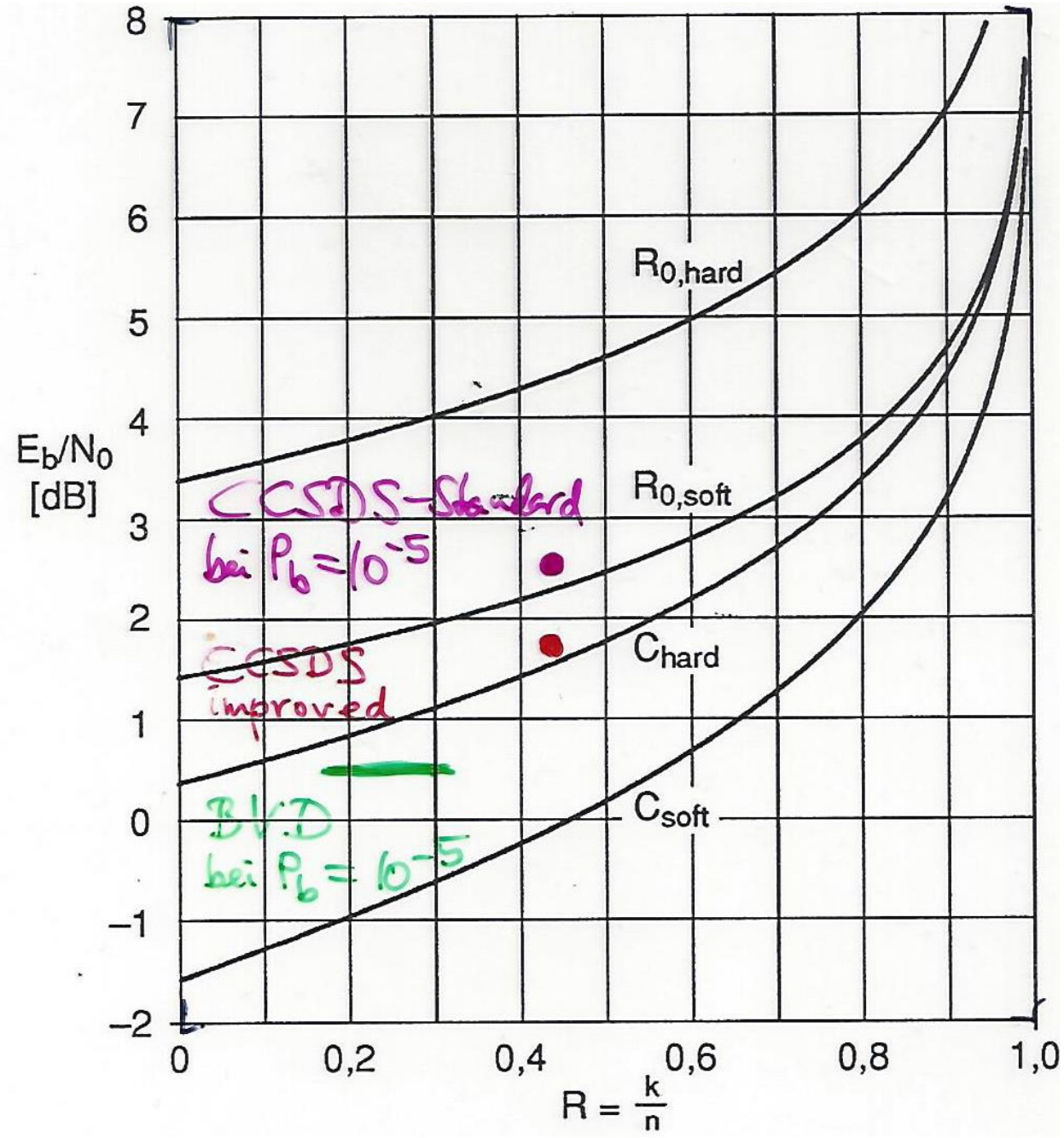
Große Gewinne gegenüber uncodierter Übertragung auch bei Coderaten nahe 1 ohne nennenswerte Bandbandbreitenexpansion

Nicht-binäre (q>2) Modulationsverfahren steigern den Gewinn (reduziertes E_b/N_0)

Shannon Grenze = -1.59 dB (unterhalb davon keine nahezu fehlerfreie Übertragung möglich, auch bei größtem Aufwand)

Geringer zusätzlicher Gewinn von nur gut 1 dB wenn die Coderate von R=1/2 auf R=0 reduziert wird (mit entsprechender enormer Bandbandbreitenexpansion)

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



CCSDS ist ein Standardisierungsgremium der Raumfahrtagenturen (NASA, ESA, DLR, CNES, CSA, JAXA, etc.)

Standardisierte Codierungsverfahren mit traditionellen Codes beruhen auf der Verkettung RS*CC (Details später). Moderne Verfahren beruhen aber auf SCCC/PCCC/LDPC und sind noch etwas besser.

Die Leistungsfähigkeit der RS*CC-Verfahren ist hier in Bezug auf die theoretischen Grenzen angegeben. Der Abstand zu den Grenzen ist klein:

- Bezogen wird das auf $P_b=10^{-5}$.
- Theoretisch wäre jedoch bis zur Grenze C_{soft} eine beliebig kleine Fehlerrate erreichbar (bei immensem Aufwand).
- Wenn für die Anwendungen (wie Bildübertragung) jedoch $P_b=10^{-5}$ ausreicht, ist der Vergleich fair.

Zusammenfassung wichtiger Größen mit ihren Einheiten (die Größen in Klammern beziehen sich auf den zeitdiskreten Kanal und sind hier nur zur Übersicht mit angegeben):

$R = k/n$	:	(Coderate)	[Infosymbol/Kanalbenutzung]
R_b	:	(Coderate)	[Infobit/Kanalbenutzung]
r_b	:	Infobitrate	[Infobit/s]
$r_c = r_b/R$	:	Codebitrate	[Codebit/s]
W	:	Bandbreite	[Hz]
N_0	:	eins. Rauschleist.dichte	[Watt/Hz=Joule]
$N = N_0 W$	:	Rauschleistung	[Watt]
S	:	Signalleistung	[Watt]
$E_b = S/r_b$	:	(Energie pro Infobit)	[Watt·s=Joule]
$E_c = RE_b$	:	(Energie pro Codebit)	[Joule]
$E_{cs} = R_b E_b$	:	Energie pro Codesymbol	[Joule]
C	:	(Kanalkapazität)	[Infobit/Kanalbenutzung]
C^*	:	Kanalkapazität	[Infobit/s]
C^*/W	:	Spektrale Bitrate	[Infobit/s/Hz=Infobit]

Satz 2.5 (Shannon-Hartley). *Für den bandbegrenzten AWGN mit wertkontinuierlichem Input ergibt sich die Kanalkapazität als*

d.h. wertkontinuierlicher Input
als theoretischer Grenzwert
hochstufiger Modulationsverfahren

$$C^* = W \cdot \log_2\left(1 + \frac{S}{N}\right) \qquad \textit{Einheit: Infobit/s}$$
$$= W \cdot \log_2\left(1 + \frac{E_b}{N_0} \cdot \frac{r_b}{W}\right). \qquad (2.6.2)$$

Bei $r_b < C^*$ ist mit entsprechendem Aufwand eine nahezu fehlerfreie Übertragung möglich. S/N und W setzen eine prinzipielle Grenze für den Durchsatz, aber nicht für die Übertragungsqualität.

Bemerkenswert ist hier, daß mit Kanalbandbreite, *Signal/Rausch-Abstand* (SNR, Signal-to-Noise Ratio) und Durchsatz die drei Schlüsselparameter einer Übertragung in einer einzigen Formel zusammengefaßt werden können.

Offensichtlich ist ein Austausch zwischen der Bandbreite und dem Signal/Rausch-Abstand möglich: Ein kleines S/N kann durch ein größeres W kompensiert werden. Dies gilt jedoch nur innerhalb gewisser Grenzen – insbesondere kann $E_b/N_0 \to 0$ nicht durch $W \to \infty$ ausgeglichen werden, wie sich schnell zeigt:

$$\lim_{W\to\infty} C^* = \lim_{W\to\infty} W \cdot \log_2\left(1+\frac{S}{N_0\cdot W}\right)$$
$$= \lim_{W\to\infty} \log_2\left(\left(1+\frac{S}{N_0\cdot W}\right)^W\right) = \log_2 \exp\left(\frac{S}{N_0}\right)$$
$$= \frac{1}{\ln 2}\cdot\frac{S}{N_0} = 1{,}44\cdot\frac{r_b\cdot E_b}{N_0}.$$

Wegen $r_b < C^*$ folgt hieraus erneut die *Shannon-Grenze* für E_b/N_0, die für eine zuverlässige Übertragung nicht unterschritten werden kann:

$$\frac{E_b}{N_0} > \ln 2 \cong -1{,}59 \text{ dB}. \qquad (2.6.4)$$

Beispiel 2.7. Beim klassischen analogen Telefonkanal über Wählleitungen beträgt die Bandbreite typischerweise $W = 3000$ Hz und der Signal/Rausch-Abstand $S/N = 30$ dB. Daraus ergibt sich die Kapazität

$$C^* = 3000\cdot\log_2(1+1000) = 29{,}901 \text{ kBit/s}.$$

Unter anderen Randbedingungen kann die Kanalkapazität auch größer oder kleiner ausfallen. Der Telefonkanal ist allerdings kein reiner AWGN-Kanal, da neben dem Rauschen auch andere Degradationen zu berücksichtigen sind.

Definition 2.5. *Als* spektrale Bitrate *wird* r_b/W *bezeichnet und als* normalisierte Kanalkapazität

$$\frac{C^*}{W} = \log_2\left(1 + \frac{S}{N}\right) \qquad \textit{Einheit: Infobit/s/Hz=Infobit}$$

$$= \log_2\left(1 + \frac{E_b}{N_0} \cdot \frac{r_b}{W}\right). \tag{2.6.5}$$

Für den Grenzfall $C^* = r_b$ gilt

$$\frac{C^*}{W} = \log_2\left(1 + \frac{E_b}{N_0} \cdot \frac{C^*}{W}\right) \qquad \text{bzw.} \qquad \frac{E_b}{N_0} = \frac{2^{C^*/W} - 1}{C^*/W} \tag{2.6.6}$$

sowie die Shannon-Grenze

$$\lim_{C^*/W \to 0} \frac{E_b}{N_0} = \ln 2 \cong -1{,}59 \text{ dB}. \tag{2.6.7}$$

Beispiel

C_1 sei ein $(n_1,k_1,d_1)_q$-Code
C_2 sei ein $(n_2,k_2,d_2)_q$-Code

Bilde den $(n,k,d)_q$-Code $C = C_1 \times C_2 = \{ (a_1,a_2) \mid a_1 \in C_1, a_2 \in C_2 \}$

Dann gilt: $n = n_1 + n_2$
$k = k_1 + k_2$
$d = \min\{d_1,d_2\}$ d.h. Kombination der schlechten Eigenschaften

Folgerungen

$\Rightarrow$ Gute Codes entstehen nicht durch simple Konstruktionen

$\Rightarrow$ Es sind zusätzliche mathematische Strukturen erforderlich wie

- Rechnen im Alphabet A_{in} (= endlicher Zahlenkörper = Galoisfeld = $\mathbf{F}_q$)
- Matrizen
- Polynome
- Diskrete Fouriertransformation
- Zustandsautomaten
- etc.

Bei einem $(n, k, d_{min})_q$-Blockcode gemäß Definition 1.4 sind sowohl die Infosymbole u_i wie die Codesymbole a_i jeweils q-stufig mit $u_i, a_i \in \mathbb{F}_q$. Das Galoisfeld $\mathbb{F}_q$ ist eine Menge mit q Elementen, in der eine Addition und eine Multiplikation (mit den inversen Operationen Subtraktion und Division) erklärt sind, und zwar derart, daß für die Rechenoperationen im Prinzip die gleichen Regeln gelten sollen wie bei den reellen oder rationalen Zahlen.

Der Vorteil von Galoisfeldern besteht darin, daß die Endlichkeit der Zahlenmenge bzw. Alphabete mit der Endlichkeit der technischen Systeme korrespondiert. Es gibt keine Rundungs- oder Quantisierungsfehler wie beim Rechnen mit reellen Zahlen, sondern immer eindeutige Ergebnisse aus der endlichen Zahlenmenge. Nachteilig ist allerdings, daß keine Ordnungsrelation erklärt werden kann, d.h. es gibt kein größer oder kleiner zwischen Elementen des Galoisfeldes.

Definition 3.1. *Mit $\mathbb{F}_q$ wird ein* Galoisfeld *(endlicher Körper, finite field) bezeichnet. $\mathbb{F}_q$ ist eine Menge mit q Elementen, zwischen denen zwei Rechenoperationen (Verknüpfungen) erklärt sind, die üblicherweise als Addition $+$ und Multiplikation $\cdot$ geschrieben werden. $\mathbb{F}_q$ soll* abgeschlossen *sein, d.h. für alle $x, y \in \mathbb{F}_q$ soll $x + y \in \mathbb{F}_q$ und $x \cdot y \in \mathbb{F}_q$ erfüllt sein. Ferner sollen die "üblichen" Rechenregeln gelten und insbesondere gibt es* neutrale Elemente *0 und 1 sowie* inverse Elemente *$-x$ und x^{-1}.*

Zusammenfassung aller Rechenregeln ($x, y, z \in \mathbb{F}_q$ beliebig):

(1) $x + y = y + x$ *(Kommutativgesetz der Addition)*
(2) $(x + y) + z = x + (y + z)$ *(Assoziativgesetz der Addition)*
(3) $x + 0 = x$ *(Existenz des add. neutralen Elementes (Nullelement))*
(4) *Zu x existiert $-x$ mit* $x + (-x) = 0$ *(Existenz des add. Inversen)*
(5) $x \cdot y = y \cdot x$ *(Kommutativgesetz der Multiplikation)*
(6) $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ *(Assoziativgesetz der Multiplikation)*
(7) $x \cdot 1 = x$ *(Existenz des mult. neutralen Elementes (Einselement))*
(8) *Zu $x \neq 0$ existiert x^{-1} mit* $x \cdot x^{-1} = 1$ *(Existenz des mult. Inversen)*
(9) $x \cdot (y + z) = x \cdot y + x \cdot z$ *(Distributivgesetz).*

Aufgrund von (1) bis (4) wird $\mathbb{F}_q$ auch als additive Gruppe und aufgrund von (5) bis (8) wird $\mathbb{F}_q \setminus \{0\}$ als multiplikative Gruppe bezeichnet. Es werden folgende Schreibweisen vereinbart:

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

Galoisfelder $\mathbb{F}_q$ existieren nur für $q = p^m$, wobei p eine Primzahl und m eine natürliche Zahl ist. Also kann q nur die Werte 2, 3, 4, 5, 7, 8, 9, 11, 13, 16, 17,... annehmen. Für jedes q existiert nur ein Galoisfeld in dem Sinne, daß zwei Galoisfelder gleicher Mächtigkeit isomorph (strukturgleich) sind, d.h. durch Umbenennung der Elemente geht das eine Galoisfeld aus dem anderen hervor.

Die mit Abstand wichtigsten Fälle sind $q = 2$ (einfache Binärcodes) und $q = 2^m$ (z.B. RS-Codes, siehe Kapitel 7).

$$\text{in } \mathbb{F}_2 \text{ und } \mathbb{F}_{2^m} \text{ gilt stets} \quad 1 + 1 = 0. \tag{3.1.1}$$

Beispiel 3.1. Für den einfachen und vorläufig auch ausschließlich praktisch interessanten Fall mit $q = p =$ Primzahl besteht $\mathbb{F}_p$ aus den natürlichen Zahlen $0, 1, 2, \ldots, p-1$, wobei Addition und Multiplikation modulo p erfolgen. Für kleines p können die Rechenoperationen in $\mathbb{F}_p$ durch *Ergebnistabellen* dargestellt werden:

(1) $\mathbb{F}_2$ ist der wichtigste Fall:

$+$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

Hier gilt beispielsweise $1 + 1 = 0$, $-1 = 1$, $-0 = 0$, $1^{-1} = 1$.

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

$+$	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

$\cdot$	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

Hier gilt beispielsweise $-1 = 4$, $-2 = 3$, $2^{-1} = 3$, $3^{-1} = 2$, $4^{-1} = 4$.

(3) $\mathbb{F}_4$ ist zwar ein Galoisfeld, kann aber nicht über die modulo 4 - Operation erklärt werden:

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

Hier gibt es kein $x \in \mathbb{F}_4$ mit $2 \cdot x = 1$, d.h. 2^{-1} existiert nicht. ■

Beweis daß $\mathbb{F}_p$ mit p=prim ein Galoisfeld ist
dazu ist zu zeigen daß zu $x \neq 0$ ein x' mit $x \cdot x' = 1$ existiert.

Nach dem Euklidischen Algorithmus kann der größte gemeinsame Teiler (GGT) zweier Zahlen aus $\mathbb{Z}$ als Linearkombination dieser Zahlen dargestellt werden: es existieren also x', p' mit

$$1 = GGT(x, p) = x \cdot x' + p \cdot p'$$

Also

$$1 = x \cdot x' \quad \text{modulo } p$$

Definition 3.2. *Die Menge aller n-Tupel (bzw. Blöcke, Vektoren, Wörter der Länge n) mit Komponenten aus $\mathbb{F}_q$ wird mit $\mathbb{F}_q^n = \mathbb{F}_{p^m}^n$ bezeichnet:*

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

Für die Mächtigkeit gilt natürlich $|\mathbb{F}_q^n| = q^n$. Zwischen den Wörtern wird eine komponentenweise Addition und Skalarmultiplikation erklärt, d.h. für alle $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$ und $\alpha \in \mathbb{F}_q$ gilt $\boldsymbol{x} + \boldsymbol{y} \in \mathbb{F}_q^n$ und $\alpha \cdot \boldsymbol{x} \in \mathbb{F}_q^n$ mit

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

Für diese Operationen gelten die "üblichen" Gesetze, die in Abschnitt A.5 nochmals zusammengestellt sind. Damit bildet $\mathbb{F}_q^n$ einen *Vektorraum* bzw. *linearen Raum*. Für die Verknüpfungen im Vektorraum werden die gleichen Operationszeichen verwendet wie im Galoisfeld und wie in den reellen Zahlen. Eine Multiplikation zwischen den Wörtern ist nicht definiert.

Verständlich sind nun auch die Aussagen (1.5.7) und (1.5.10) aus Satz 1.1. Ferner ist der Hammingabstand invariant gegenüber Verschiebungen:

$$d_H(\boldsymbol{x}, \boldsymbol{y}) = d_H(\boldsymbol{x} + \boldsymbol{z}, \boldsymbol{y} + \boldsymbol{z}). \tag{3.1.2}$$

Bei einem $(n,k)_q$-Blockcode gilt $\boldsymbol{u} = (u_0, \ldots, u_{k-1}) \in \mathbb{F}_q^k$ für das Infowort und $\boldsymbol{a} = (u_0, \ldots, a_{n-1}) \in \mathbb{F}_q^n$ für das Codewort sowie $\mathcal{C} \subseteq \mathbb{F}_q^n$, $|\mathcal{C}| = q^k$ für den Code.

Seit vielen Jahren die erste Frage in der mündlichen Prüfung:

Definition 3.3. *Der $(n,k)_q$-Code $\mathcal{C}$ über $\mathbb{F}_q$ heißt* linearer Code, *wenn mit zwei Codewörtern auch die Summe wieder ein Codewort ist:*

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

Für nicht-binäre Codes mit $q > 2$ wird zusätzlich gefordert:

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

Eine äquivalente Kennzeichnung ist, daß $\mathcal{C}$ ein Vektorraum sein soll.

Beispiele: $\mathcal{C} = \{000, 100, 010, 001\}$ ist nicht linear
$\mathcal{C} = \{000, 110, 011, 111\}$ ist nicht linear
$\mathcal{C} = \{000, 110, 101, 011\}$ ist linear.

Beispiel 3.2. **(1)** Als *Wiederholungscode* (repetition code) wird der $(n,1)_2$-Code

$$\mathcal{C} = \{00\ldots0, 11\ldots1\} \tag{3.1.5}$$

bezeichnet. Die Linearität ist offensichtlich. Es gilt $R = 1/n$ für die Coderate und $d_{\min} = n$ für die Minimaldistanz. Eine systematische Encodierung kann gemäß $u_0 \mapsto \boldsymbol{a} = (u_0, \ldots, u_0)$ erfolgen.

(2) Als *Parity Check Code* (auch: Single Parity Check Code, SPCC) wird der $(n, n-1)_2$-Code

$$\mathcal{C} = \left\{ (a_0, \ldots, a_{n-1}) \;\middle|\; \sum_{i=0}^{n-1} a_i = 0 \right\} \tag{3.1.6}$$

bezeichnet. Der Code ist linear mit $R = (n-1)/n = 1 - 1/n$. Da $000\ldots0$ und $110\ldots0$ jeweils Codewörter sind, folgt $d_{\min} = 2$. Bei einer systematischen Encodierung gemäß $(u_0, \ldots, u_{n-1}) \mapsto \boldsymbol{a} = (u_0, \ldots, u_{n-1}, u_0 + \cdots + u_{n-1})$ wird die Summe der Infobits als Prüfbit angehängt. ■

Satz 3.2. *Ein linearer Code $\mathcal{C}$ ist invariant gegenüber additiven Verschiebungen, d.h. für alle $\boldsymbol{b} \in \mathcal{C}$ gilt:* $\mathcal{C} + \boldsymbol{b} = \{\boldsymbol{a} + \boldsymbol{b} \mid \boldsymbol{a} \in \mathcal{C}\} = \mathcal{C}$.

Aus $d_H(\boldsymbol{a}, \boldsymbol{b}) = w_H(\boldsymbol{a} - \boldsymbol{b})$ folgt sofort, daß die Minimaldistanz eines Codes gleich dem minimalen Gewicht der Codewörter ist und somit sind für die Bestimmung von $d_{\min}$ jetzt also nicht mehr $q^k(q^k - 1)$ Paare zu betrachten, sondern nur noch $q^k - 1$ Wörter:

Satz 3.3. *Für die Minimaldistanz eines linearen $(n, k, d_{\min})_q$-Codes gilt:*

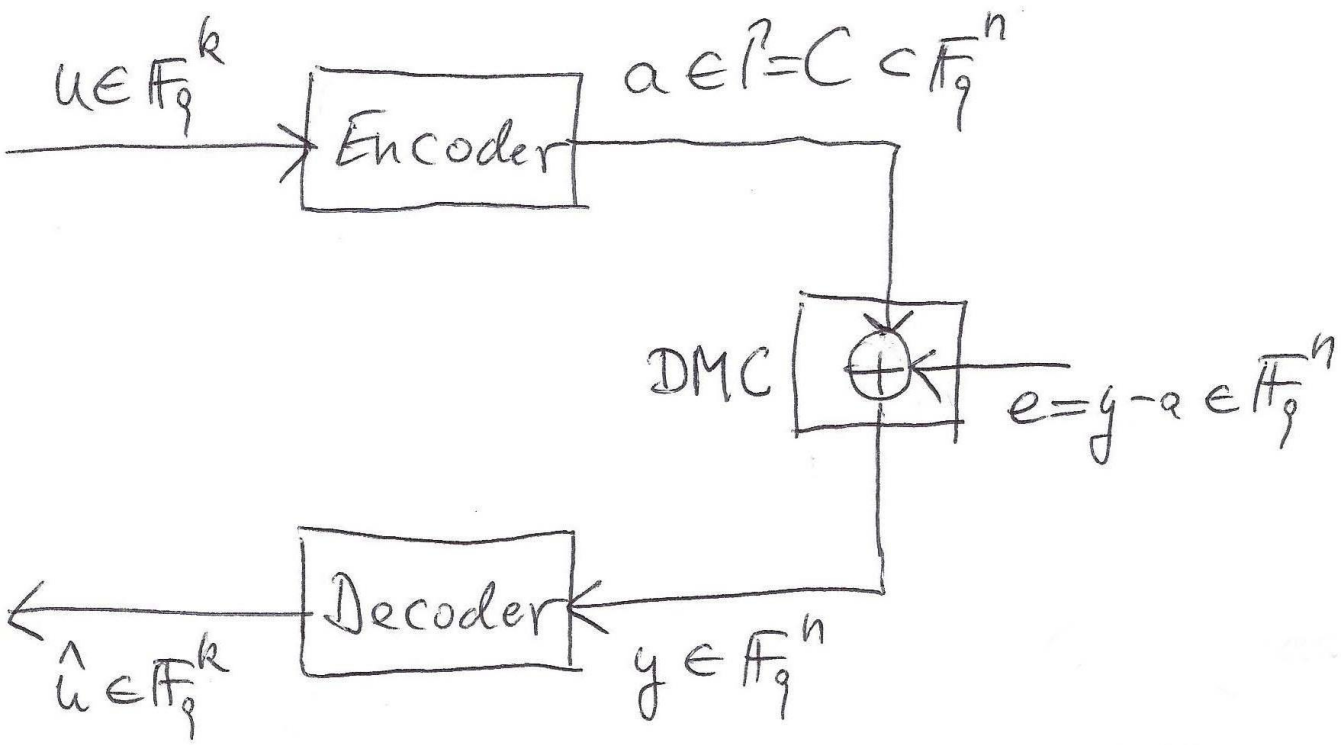
$$\begin{aligned} d_{\min} &= \min\{d_H(\boldsymbol{a}, \boldsymbol{b}) \mid \boldsymbol{a}, \boldsymbol{b} \in \mathcal{C}, \boldsymbol{a} \neq \boldsymbol{b}\} \\ &= \min\{w_H(\boldsymbol{a}) \mid \boldsymbol{a} \in \mathcal{C}, \boldsymbol{a} \neq \boldsymbol{0}\}. \end{aligned} \tag{3.1.7}$$

Bei Hard-Decision, also bei $\mathcal{A}_{\text{in}} = \mathcal{A}_{\text{out}} = \mathbb{F}_q$, kann die Übertragung interpretiert werden als Überlagerung des Sendewortes mit einem *Fehlerwort*:

$$\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{e}. \tag{3.2.1}$$

Das Empfangswort $\boldsymbol{y}$ ist also die Summe aus dem gesendeten Codewort $\boldsymbol{a}$ und dem Fehlerwort (Fehlermuster) $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{a} \in \mathbb{F}_q^n$. Aufgrund der linearen Struktur ist $\boldsymbol{y}$ genau dann ein Codewort, wenn $\boldsymbol{e}$ ein Codewort ist. In Analogie zu Abschnitt 1.6 gibt es folgende Möglichkeiten:

$\boldsymbol{e} = \boldsymbol{0}$	Fehlerfreie Übertragung.
$\boldsymbol{e} \in \mathcal{C} \setminus \{\boldsymbol{0}\}$	Verfälschung in ein anderes Codewort – das kann niemals erkannt oder korrigiert werden.
$\boldsymbol{e} \notin \mathcal{C}$	Die Verfälschung ist generell erkennbar und eventuell korrigierbar durch den Decoder.



DC ist gedächtnislos

⇕

Fehler in e sind statistisch unabhängig

⇓ ⇏

$w_H(e)$ ist binomialverteilt

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

wobei $p_e = P(y_i \neq a_i) = P(e_i \neq 0)$

Definition 3.5. *Als Kugel $K_r(\boldsymbol{x})$ vom Radius r um das Wort $\boldsymbol{x} \in \mathbb{F}_q^n$ wird die Menge aller Wörter verstanden, die von $\boldsymbol{x}$ eine Hammingdistanz $\leq r$ haben:*

$$K_r(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathbb{F}_q^n \mid d_H(\boldsymbol{x}, \boldsymbol{y}) \leq r\}. \tag{3.2.2}$$

Klar ist $K_0(\boldsymbol{x}) = \{\boldsymbol{x}\}$ und $K_n(\boldsymbol{x}) = \mathbb{F}_q^n$. Oftmals werden die Kugeln um die Codewörter betrachtet – zum Inhalt der Kugeln zählen aber immer alle Wörter aus $\mathbb{F}_q^n$. Aus der Kombinatorik ist bekannt, daß es genau $\binom{n}{i}(q-1)^i$ Wörter $\boldsymbol{y} \in \mathbb{F}_q^n$ mit $d_H(\boldsymbol{x}, \boldsymbol{y}) = i$ gibt. Somit folgt für die Mächtigkeiten der Kugeln das wichtige Resultat

$$\left|K_r(\boldsymbol{x})\right| = \sum_{i=0}^{r} \binom{n}{i}(q-1)^i. \tag{3.2.3}$$

Beispiel 3.3. Betrachte den $(3,1)_2$-Wiederholungscode $\mathcal{C} = \{000, 111\}$ mit $d_{\min} = 3$. Für die Kugeln gilt:

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

Definition 3.4. *Ein $(n,k)_q$-Blockcode $\mathcal{C}$*

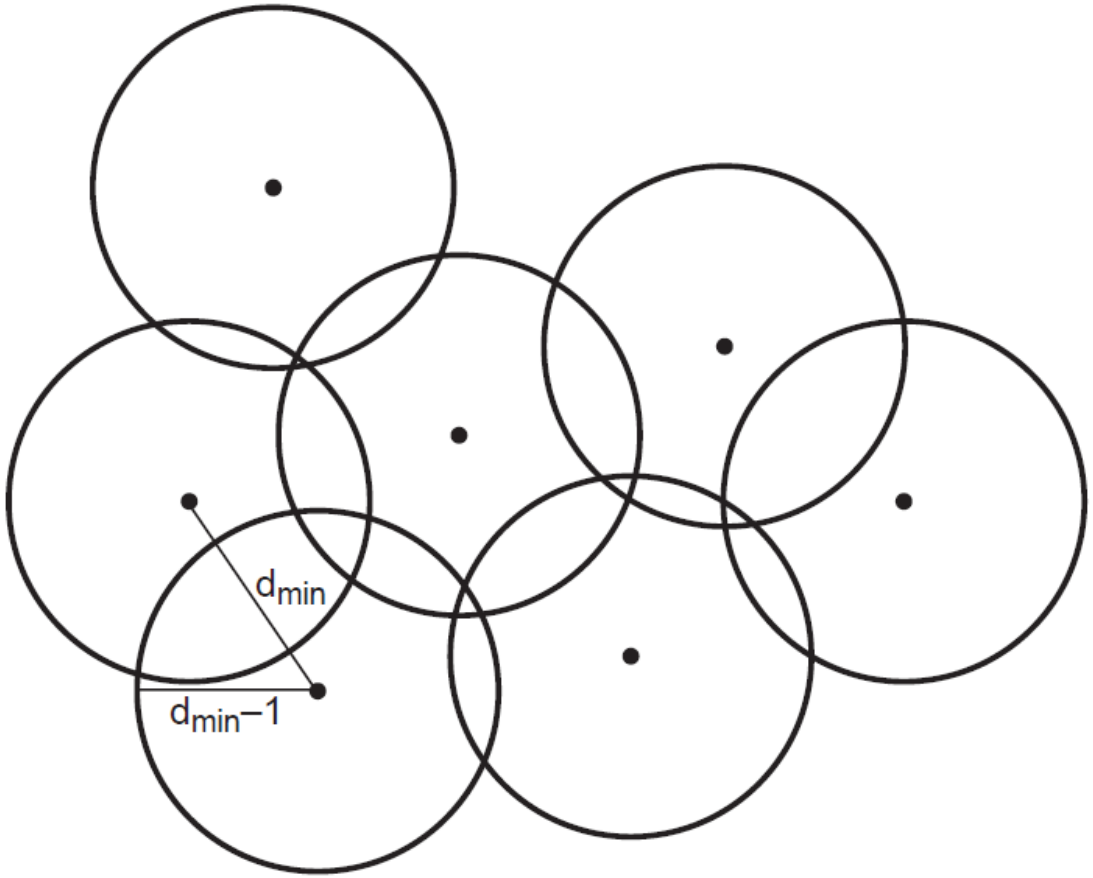
(1) korrigiert *t* Fehler, *wenn für jedes Fehlermuster $\boldsymbol{e}$ mit $w_H(\boldsymbol{e}) \leq t$ die Maximum-Likelihood-Decodierung das richtige Codewort liefert (error correction).*

(2) erkennt *t'* Fehler, *wenn für jedes Fehlermuster $\boldsymbol{e} \neq \boldsymbol{0}$ mit $w_H(\boldsymbol{e}) \leq t'$ das Empfangswort $\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{e}$ kein Codewort ist (error detection).*

In ausführlicherer Sprechweise werden also *bis zu* t Fehler korrigiert. Wichtig ist dabei, daß wirklich *jedes* prinzipiell mögliche Fehlermuster vom Gewicht $\leq t$ korrigierbar sein muß. Entsprechendes gilt natürlich auch für die Fehlererkennung.

Satz 3.4. *Ein* $(n, k, d_{\min})_q$*-Code erkennt* $t' = d_{\min} - 1$ *Fehler.*

Beweis: Nach Satz 3.3 folgt aus $\boldsymbol{e} \in \mathcal{C} \backslash \{\boldsymbol{0}\}$ zwangsläufig $w_H(\boldsymbol{e}) \geq d_{\min}$. Durch Umkehrung ergibt sich: Ein Fehlermuster mit $w_H(\boldsymbol{e}) \leq d_{\min} - 1$ impliziert $\boldsymbol{e} \notin \mathcal{C} \backslash \{\boldsymbol{0}\}$ und wird somit zwangsläufig erkannt. ■



Mit Bild 3.1 wird veranschaulicht, daß jede Kugel vom Radius $d_{\min} - 1$ um ein Codewort kein anderes Codewort enthält. Denn wenn zwei Codewörter $\boldsymbol{a}, \boldsymbol{b}$ existieren mit $\boldsymbol{b} \in K_{d_{\min}-1}(\boldsymbol{a})$, so würde $d_H(\boldsymbol{a}, \boldsymbol{b}) \leq d_{\min} - 1$ folgen. Jedoch ist $d_{\min}$ der minimale Abstand zwischen zwei verschiedenen Codewörtern.

Bei höchstens $d_{\min} - 1$ Fehlern liegt das Empfangswort in der Kugel um das tatsächlich gesendete Codewort. Da in dieser Kugel kein weiteres Codewort liegt, kann das Empfangswort mit keinem Codewort verwechselt werden – abgesehen natürlich von der fehlerfreien Übertragung. In einer Kugel vom Radius $d_{\min} - 1$ um ein beliebiges Wort können jedoch mehrere Codewörter liegen.

Bild 3.1. Kugeln vom Radius $d_{\min} - 1$ um die Codewörter

Satz 3.5. *Ein $(n, k, d_{\min})_q$-Code korrigiert t Fehler, sofern $2t + 1 \leq d_{\min}$ bzw. äquivalent $t = \lfloor (d_{\min} - 1)/2 \rfloor$ gilt.*

Beweis: Sei $\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{e}$ mit $w_H(\boldsymbol{e}) \leq t$. Dann gilt $d_H(\boldsymbol{y}, \boldsymbol{a}) \leq t$. Sei $\boldsymbol{b} \in \mathcal{C}$ beliebig mit $\boldsymbol{b} \neq \boldsymbol{a}$. Aus der Dreiecksungleichung ergibt sich

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

Somit folgt $d_H(\boldsymbol{y}, \boldsymbol{b}) \geq t + 1$. Also ist der Abstand von $\boldsymbol{a}$ zu $\boldsymbol{y}$ höchstens t, während jedes andere Codewort von $\boldsymbol{y}$ mindestens den Abstand $t+1$ hat. Somit wählt der ML-Decoder das richtige Codewort. ■

Bild 3.2 veranschaulicht die Situation für Kugeln vom Radius t um die Codewörter. Die Kugeln sind disjunkt. Wenn bei der Übertragung das Codewort $\boldsymbol{a}$ mit höchstens t Fehlern überlagert wird, so liegt das Empfangswort $\boldsymbol{y}$ in $K_t(\boldsymbol{a})$ und hat von jedem anderen Codewort mindestens den Abstand $t+1$. Somit ist das Codewort minimalen Abstandes zu $\boldsymbol{y}$ eindeutig bestimmt. Die Kugeln vom Radius t um die Codewörter werden deshalb auch *Decodierkugeln* genannt. Empfangswörter mit mehr als t Fehlern liegen entweder in einer falschen Decodierkugel und werden dann falsch decodiert oder sie liegen zwischen den Decodierkugeln und werden dann richtig oder falsch decodiert.

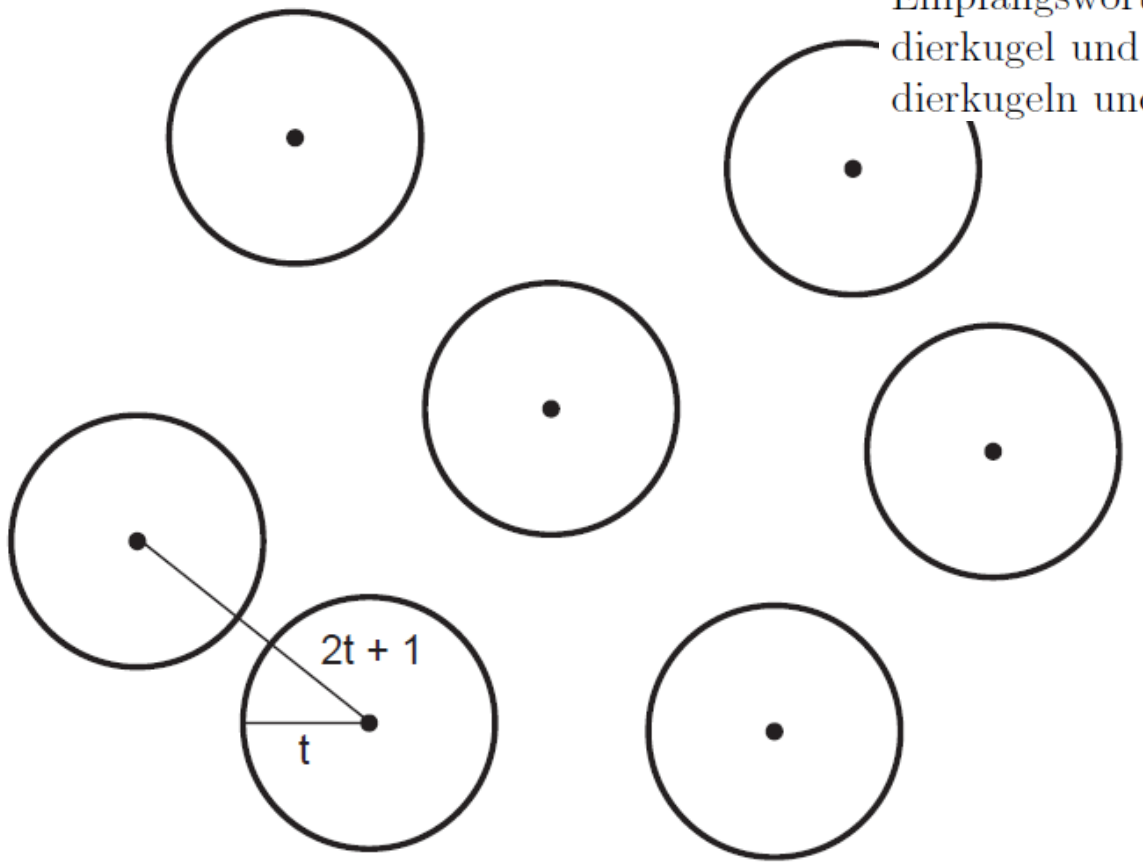


Bild 3.2. Decodierkugeln vom Radius $t = \lfloor (d_{\min} - 1)/2 \rfloor$ um die Codewörter

Satz 3.4. *Ein $(n, k, d_{\min})_q$-Code erkennt $t' = d_{\min} - 1$ Fehler.*

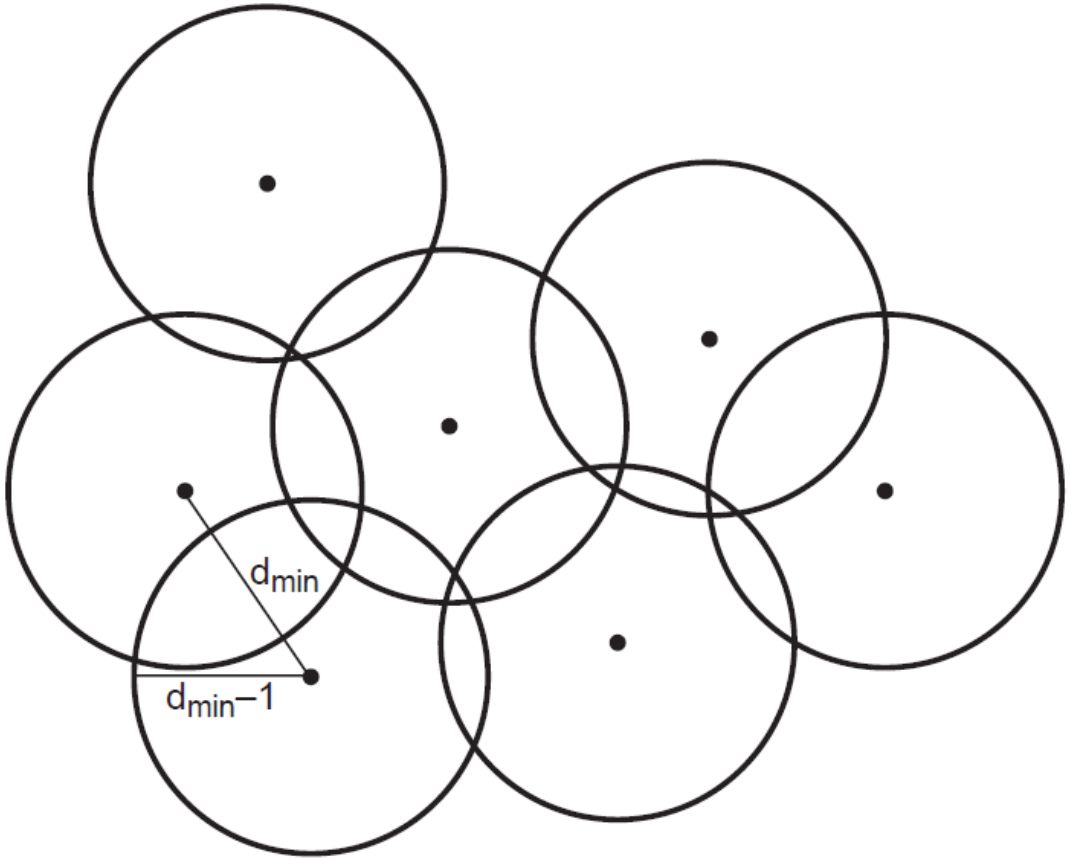


Bild 3.1. Kugeln vom Radius $d_{\min} - 1$ um die Codewörter

Satz 3.5. *Ein $(n, k, d_{\min})_q$-Code korrigiert t Fehler, sofern $2t + 1 \leq d_{\min}$*

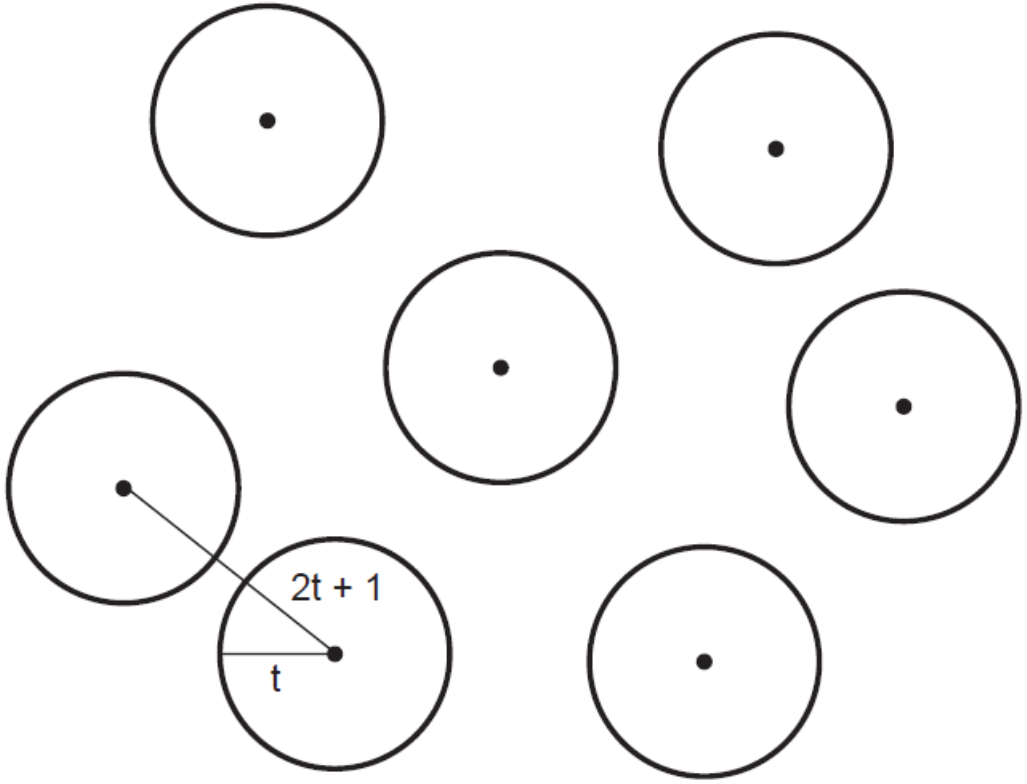


Bild 3.2. Decodierkugeln vom Radius $t = \lfloor (d_{\min} - 1)/2 \rfloor$ um die Codewörter

Dazu sei ein $(n, k, d_{\min})_q$-Code mit $2t+1 \leq d_{\min}$ gegeben. Das Empfangswort $\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{e}$ weise $f = d_H(\boldsymbol{a}, \boldsymbol{y})$ Fehler auf. Bei der ML-Schätzung $\hat{\boldsymbol{a}}$ erfolgen $f_{ML} = d_H(\boldsymbol{y}, \hat{\boldsymbol{a}})$ Korrekturen im Empfangswort. Es sind zwei Fälle zu unterscheiden:

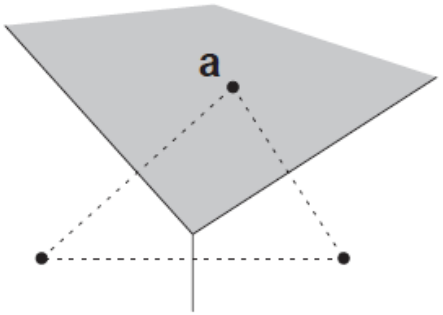
(a) Bei $f \leq t$ ist die ML-Schätzung richtig mit $\hat{\boldsymbol{a}} = \boldsymbol{a}$ und $f_{ML} = f$.

(b) Bei $f > t$ ist die ML-Schätzung eventuell falsch mit $\hat{\boldsymbol{a}} \neq \boldsymbol{a}$, und f_{ML} kann unkontrollierbar groß werden (bis zu $f_{ML} = n$). Nach der Dreiecksungleichung ergibt sich $d_H(\boldsymbol{a}, \hat{\boldsymbol{a}}) \leq f + f_{ML}$ und somit kann auch die ML-Schätzung an unkontrollierbar vielen Stellen vom gesendeten Codewort abweichen, obwohl der Code nur t Fehler korrigieren kann.

Aus "Sicherheitsgründen" könnte die Korrektur bei der Decodierung auf höchstens $d_H(\boldsymbol{y}, \boldsymbol{a}) \leq t$ Stellen begrenzt werden. Damit ergibt sich das sogenannte BMD-Prinzip:

Definition 3.6. *Beim* Begrenzten-Minimaldistanz-Decoder *(BMD, Bounded Minimum Distance Decoder) erfolgt nur dann eine Decodierung, wenn das Empfangswort innerhalb einer Decodierkugel liegt, d.h. wenn es zum Empfangswort ein Codewort im Abstand $\leq t$ gibt.*

Schulgeometrie:
Die Mittelhalbierenden eines Dreiecks schneiden sich in einem Punkt

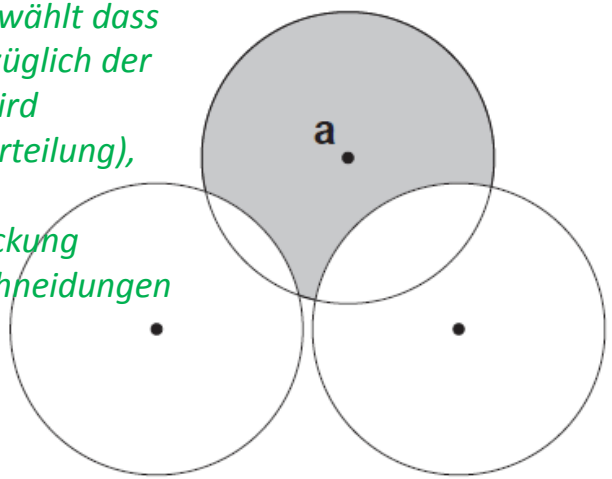


MLD

(Maximum-Likelihood-Decoder, siehe Satz 1.3): Für jedes Empfangswort wird nach dem nächstgelegenen Codewort gesucht. Der Entscheidungsbereich wird im 2-dim. Fall durch die Mittelhalbierenden begrenzt und im n-dim. Fall durch entsprechende Hyperflächen.

Das MLD-Prinzip minimiert die Wort-Fehlerwahrscheinlichkeit (unter der Annahme gleicher Apriori-Wahrscheinlichkeiten).

Der Kugelradius wird so gewählt dass der überdeckte Bereich abzüglich der Schnittmengen maximal wird (bewertet mit der Gauß-Verteilung),
Also:
Kugel klein ⇒ wenig Abdeckung
Kugel groß ⇒ viele Überschneidungen

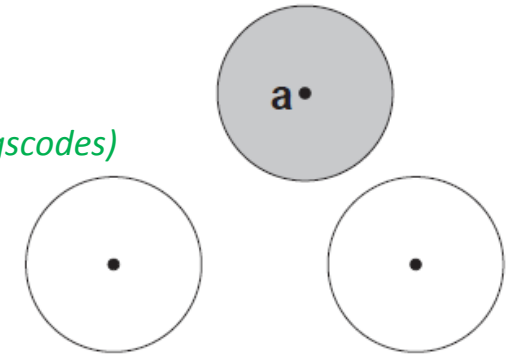


BDD

(Begrenzter-Distanz-Decoder, siehe den Beweis des Kanalcodierungstheorems in Abschnitt 2.7): Um jedes Codewort wird eine Kugel vom gleichen Radius t gemäß (2.7.4) gelegt. Es werden nur diejenigen Empfangswörter zum Kugelmittelpunkt decodiert, die in genau einer Kugel liegen. Keine Decodierung erfolgt für Wörter, die in keiner oder in mehreren Kugeln liegen. Die Entscheidungsbereiche sind also geometrisch von relativ komplizierter Form.
Da mit dem BDD das Kanalcodierungstheorem bewiesen wird, ist der BDD nur unwesentlich schlechter als der MLD.

Generell:
Schraffierte Fläche = Entscheidungsbereich für a

BMD typisch bei Blockcodes
(aber MLD typisch bei Faltungscodes)



BMD

(Begrenzter-Minimaldistanz-Decoder, siehe Definition 3.6): Um jedes Codewort wird als Entscheidungsbereich eine Kugel vom gleichen Radius $t = \lfloor (d_{\min} - 1)/2 \rfloor$ gelegt. Damit sind die Kugeln als disjunkt gewährleistet. Es werden nur diejenigen Empfangswörter zum Kugelmittelpunkt decodiert, die in einer Kugel liegen. Keine Decodierung erfolgt für Wörter, die außerhalb der Kugeln liegen.

Beim BMD erfolgt also eine "Decodierung bis zur halben Minimaldistanz".

Der BMD ist natürlich schlechter als die ML-Decodierung, aber die wesentlichen Vorteile des BMD liegen in der vereinfachten Realisierung des Decoders