UNIVERSITÄT DER BUNDESWEHR MÜNCHEN
Fakultät für Elektrotechnik und Informationstechnik

Stochastische Analysen und Algorithmen zur Soft Decodierung binärer linearer Blockcodes

Thomas F. Sturm

UNIVERSITÄT DER BUNDESWEHR MÜNCHEN
Fakultät für Elektrotechnik und Informationstechnik

Stochastische Analysen und Algorithmen zur Soft Decodierung binärer linearer Blockcodes

Thomas F. Sturm

Vorsitzender des Promotionsausschusses: Prof. Dr.-Ing. U. Appel
1. Berichterstatter: Prof. Dr.rer.nat. Dr.-Ing. S. Schäffler
2. Berichterstatter: Prof. Dr.rer.nat. C. Hillermeier

Tag der Prüfung: 24.7.2003

Mit der Promotion erlangter akademischer Grad:
Doktor-Ingenieur
(Dr.-Ing.)

Neubiberg, den 25. Juli 2003

Das Dokument wurde mit dem Satzsystem LaTeX 2ε erstellt.

Inhaltsverzeichnis

Abbildungsverzeichnis

Algorithmenverzeichnis

Notation

Allgemein:

$\mathbb{N}$	Menge der natürlichen Zahlen, $\{1,2,3,\ldots\}$
$\mathbb{N}_0$	Menge der natürlichen Zahlen mit Null, $\{0,1,2,3,\ldots\}$
$\mathbb{R}$	Menge der reellen Zahlen
$\mathbb{R}_0^+$	Menge der positiven reellen Zahlen mit Null, $\{x \in \mathbb{R};\ x \geq 0\}$
$\mathbb{R}^n$	der n-dimensionale reelle Vektorraum
$\mathcal{S}_n$	Gruppe der Permutationen in $\{1,\ldots,n\}$ (symmetrische Gruppe vom Grad n)
$\mathcal{P}(M)$	Potenzmenge einer Menge M
$\dim(.)$	Dimension eines Vektorraums
$\operatorname{rang}(.)$	Rang einer Matrix
A^{-1}	die zur Matrix A inverse Matrix
$A^{\top}$	die zur Matrix A transponierte Matrix
$A^{-\top} = (A^{-1})^{\top} = (A^{\top})^{-1}$	die zur Matrix A transponierte inverse Matrix
I_n	die n-dimensionale Einheitsmatrix

Wahrscheinlichkeitstheorie:

Ω	Basismenge, Ergebnismenge (S. 163)
$\bar{\mathbb{R}}$	Erweiterung von $\mathbb{R}$ um $\{\pm\infty\}$ (S. 163)
$\mathcal{B}, \mathcal{B}^n$	Borelsche σ-Algebra (S. 165)
$\bar{\mathcal{B}}$	Erweiterung der Borelschen σ-Algebra (S. 168)
$\mathcal{S}$	σ-Algebra (S. 164)
$\sigma(\mathcal{F})$	vom Mengensystem $\mathcal{F}$ erzeugte σ-Algebra (S. 165)
$\sigma(X)$	von der Zufallsvariable X erzeugte σ-Algebra (S. 179)
μ	Maß (S. 164)
λ, λ^n	Lebesgue-Borel-Maß (S. 165)
P	Wahrscheinlichkeitsmaß (S. 170)
P_X	Bildmaß der Zufallsvariable X (Verteilung von X) (S. 171)
$P(.\|.)$	Bedingte Wahrscheinlichkeit (S. 172, 46)
$(\Omega, \mathcal{S})$	Meßraum (S. 166)

$(\Omega, \mathcal{S}, \mu)$	Maßraum (S. 166)
$(\Omega, \mathcal{S}, P)$	Wahrscheinlichkeitsraum (S. 170)
$\mathbf{E}(.)$	Erwartungswert (S. 171)
$\mathbf{Var}(.)$	Varianz (S. 172)
$\mathbf{Cov}(.)$	Covarianz (S. 178)
$B(s,p)$	Binomialverteilung mit Parameter s, p (S. 175)
$\mathcal{N}(\mu, \Sigma)$	Normalverteilung mit Erwartungsvektor μ und Kovarianzmatrix Σ (S. 177)
I_A	Indikatorfunktion der Menge A (S. 167)
$\int f \, d\mu$	μ-Integral über die Funktion f (S. 169)

Codierung:

$k \in \mathbb{N}$	Codedimension (S. 20)
$n \in \mathbb{N}$	Codelänge (S. 20)
$(\{\pm 1\}, \oplus, \odot)$	Binärer Körper (S. 20)
$\oplus$	Addition im binären Körper (S. 20)
$\odot$	Multiplikation im binären Körper (S. 20)
$u \in \{\pm 1\}^k$	uncodiertes Wort (S. 20)
$c \in \{\pm 1\}^n$	codiertes Wort, Codewort (S. 20)
$\mathcal{C}$	Menge der Codewörter (S. 20)
$G \in \{\pm 1\}^{k,n}$	Generatormatrix (S. 23)
$J_1, \ldots, J_n$	charakterisierende Mengen (S. 23)
φ	Codierungsabbildung (S. 20)
(n, k, φ)	binärer linearer (n,k)-Blockcode (S. 20)
$d_{\text{ham}}(\varphi)$	Hamming-Distanz eines (n,k)-Blockcodes mit Codierungsabbildung φ (S. 21)
$\{\pm 1\}[x]$	Polynomring über dem Körper $(\{\pm 1\}, \oplus, \odot)$
$g(x) \in \{\pm 1\}[x]$	Generatorpolynom (S. 25)
$((n_1, k_1, \varphi_1), \ldots, (n_m, k_m, \varphi_m))$	verketteter binärer linearer (n,k)-Blockcode (S. 27)
$(7, 4, \varphi_{\text{bsp}})$	Beispiel für einen $(7,4)$-Blockcode (S. 23)

Decodierung:

δ_{HD}	Hard-Decision Decodierungsabbildung (S. 35)
δ_{SD}	Soft-Decision Decodierungsabbildung (S. 39)
δ_{SO}	Soft-Output Decodierungsabbildung (S. 49)
$\mathcal{K}$	n-Kanal (S. 31)
$\hat{\mathcal{K}}$	Superkanal bezüglich $(\varphi, \mathcal{K}, \delta_{\text{SO}})$ (S. 50)
$\mathcal{K}_c$	Zufallsvariable des Kanals zur Realisierung c der Kanaleingabe (S. 31)
f_c	Dichte des Bildmaßes von $\mathcal{K}_c$ (S. 33)

U Zufallsvariable der Kanaleingabe des Superkanals (S. 50)
C Zufallsvariable der Kanaleingabe (S. 31)
$\hat{C}$ diskrete Zufallsvariable der Kanalausgabe (S. 34)
Y stetige Zufallsvariable der Kanalausgabe (S. 31)
y Realisierung der stetigen Zufallsvariable der Kanalausgabe (Demodulationsergebnis) (S. 18)
σ^2 bitweise Varianz der AWGN-Kanalstörung (S. 33)
$w(u,y)$ Wahrschein. von $\{\omega \in \Omega;\ U(\omega) = u\}$ unter der Bedingung $\{\omega \in \Omega;\ Y(\omega) = y\}$ (S. 46)
$\Gamma^i(\alpha)$ Menge der Codewörter $\varphi(u)$, wobei $u_i = \alpha$ (S. 51)
$M^i(\alpha)$ Teilmenge des $\mathbb{R}^n$ (S. 51)

Soft-Output Decodierung terminierter Faltungscodes:

a Anzahl der Eingabeblöcke (S. 60)
b Anzahl der Eingabebits (S. 60)
d Anzahl der Ausgabebits pro Ausgabeblock (S. 60)
l Blocklänge des Schieberegisters (S. 60)
L Bitlänge des Schieberegisters (S. 60)
Q Anzahl der Zustandsübergänge (S. 60)
$M_1,\ldots,M_d \subseteq \{1,\ldots,L\}$ definierende Mengen (S. 60)
$(a,b,l,d,M_1,\ldots,M_d)$ terminierter (n,k)-Faltungscode (S. 60)
ψ Codierungsabbildung des Schieberegisterinhaltes (S. 60)
$S = \{\pm 1\}^L$ Menge der Zustände (S. 60)
$s_0 \in S$ Nullelement von S (S. 60)
$s_i^u \in S$ i-ter Schieberegisterzustand (S. 60)
$\tau(s)$ Letzter Bitblock eines Schieberegisterzustands s (S. 69)
$V = \{\pm 1\}^b$ Menge der Zustandsübergangszeichen (S. 60)
$v_0 \in V$ Nullelement von V (S. 60)
V_m Menge der zulässigen Zustandsübergangszeichen im m-ten Schritt (S. 74)
$V_j^i(\alpha)$ Einschränkung der Menge der zulässigen Zustandsübergangszeichen (S. 74)
U_m Menge der ersten m Komponenten zulässiger Eingabewörter (S. 74)
$U_Q^i(\alpha)$ Einschränkung der Menge der zulässigen Eingabewörter (S. 74)
T Zustandsübergangsfunktion (S. 61)
$\hat{T}$ Inverse Zustandsübergangsfunktion (S. 69)
$\mathcal{T}$ Knotenmenge eines Trellis-Diagramms (S. 68)
$(s,q) \in \mathcal{T}$ Knoten eines Trellis-Diagramms (S. 68)
$(\mathcal{T},T)$ Trellis-Diagramm (S. 68)
$\Delta F_q(s)$ Bewertungsfunktion für den Knoten (s,q) (S. 65)
$\mu_q(s)$ Multiplikator für den Knoten (s,q) (S. 74)

$A_\alpha^i(y)$	i-ter Teilterm der L-Werte zur Realisierung y (S. 66)
$\tilde{A}_m$	Hilfs-Abbildungen zur L-Wert Berechnung (S. 69)
A_m	Rekursive Abbildungen zur L-Wert Berechnung (S. 70)
B_m	Rekursive Abbildungen zur L-Wert Berechnung (S. 73)
W	Abbildung auf eine Zustandsmenge (S. 69)

Soft-Decision Decodierung binärer linearer Blockcodes:

F	Soft-Decision Zielfunktion (S. 92)
$A \in \{\pm 1\}^{k,k}$	Reguläre Matrix zur Quasi-Systematisierung (S. 95)
$\tilde{G} \in \{\pm 1\}^{k,n}$	Generatormatrix des quasi-systematisierten Codes (S. 95)
$\tilde{J}_1, \ldots, \tilde{J}_n$	charakterisierende Mengen des quasi-systematisierten Codes (S. 96)
$\mu \in \mathcal{S}_n$	Betragssortierung der Komponenten von y (S. 97)
τ	Indizes der Einheitsspalten von $\tilde{G}$ (S. 95)
$\rho \in \mathcal{S}_k$	Betragssortierung von k Komponenten von y (S. 97)
π	Projektionsabbildung (S. 97)
$\alpha_m \in \{1,\ldots,k\} \times \{1,\ldots,k\}$	Protokoll einer Zeilenaddition (S. 97)
$a \in \mathbb{N}_0$	Anzahl der Zeilenadditionen (S. 97)
A_{pq}	$k \times k$-Einheitsmatrix mit Extra-Element -1 in der p-q-Position (S. 97)
$\tilde{u}^0$	Startpunkt (transformiert) (S. 97)
$\tilde{F}$	transformierte Soft-Decision Zielfunktion (S. 105)
γ	Summennorm von y (S. 105)
$s_j(\tilde{u})$	Teilsumme der transformierten Soft-Decision Zielfunktion (S. 105)
$s_j^b(\tilde{u})$	Obere Schranke von $s_j(\tilde{u})$ (S. 105)
$\tilde{F}^b(\tilde{u})$	Untere Schranke von $\tilde{F}(\tilde{u})$ (S. 105)
K_b	Indexmenge vollständig bekannter $\tilde{J}_j$ im b-ten Schritt (S. 105)
$\hat{K}_b$	Vereinigungsmenge von Mengen K_b (S. 105)
L^b	Teilsumme von γ (S. 105)
γ^b	Summe der von $\tilde{u}$ unabhängigen Abschätzungen (S. 105)
$S^b(\tilde{u})$	Summe der bekannten $s_j(\tilde{u})$ (S. 105)
$\Delta^b(\tilde{u})$	Update von $S^b(\tilde{u})$ (S. 105)
$\tilde{u}_{\min}$	Minimierer der transformierten Soft-Decision Zielfunktion (S. 111)

Kapitel 1

Einleitung

Make it so.

(Jean-Luc Picard)

Bei der Übertragung von digitalen Nachrichten über analoge Medien, etwa Funkkanäle im Mobilfunk oder bei der Satellitenkommunikation, erhält der Empfänger diese Nachrichten nur in gestörter Form. Anstelle jedes Bits einer ursprünglichen Nachricht liegt dann dem Empfänger ein (verfälschter) analoger Wert („Soft-Wert“) vor, der im allgemeinen nicht einfach zum originalen Bit gerundet werden kann. Daher erweitern spezielle Kanal*codierer* des Senders jeweils einen Block u von k Bits der Nachricht zu einem redundanten, größeren Block c mit n Bits („Blockcodierung“). Die Aufgabe des Kanal*decodierers* des Empfängers besteht dann in der Rekonstruktion des ursprünglichen Blocks u aus k Bits mit Hilfe der n empfangenen Soft-Werte.

Die Rekonstruktion des Bitblocks, auch Wort genannt, ist mit einer gewissen Fehlerwahrscheinlichkeit behaftet, die von der Art der Codierung, der Störung auf dem Kanal und der Decodierungsmethode abhängt. Ausgehend von gegebenen binären linearen Blockcodes und Kanalstörungen soll nun die Fehlerwahrscheinlichkeit durch die Entwicklung von Soft Decodierungsverfahren minimiert werden, die im Gegensatz zu herkömmlichen (Hard Decodierungs-) Methoden den Informationsgehalt der empfangenen Soft-Werte bis zum Ende des Decodierungsprozesses vollständig verwenden. Die erwartete Verbesserung der Wortfehlerwahrscheinlichkeit bei der Decodierung begründet sich dabei auf Aussagen der Shannonschen Informationstheorie, siehe [Sha48, SW76, BB90].

In Abbildung 1.1 werden die technischen Auswirkungen einer solchen Verbesserung veranschaulicht. Dabei gibt es zwei Lesarten:

- Bei einer festen Kanalstörung (als Signal-to-Noise Ratio bezeichnet) werden weniger Decodierungsfehler begangen. Damit kommt es zu einer **Verbesserung der Empfangsqualität**.
- Bei einem festgelegten Fehlerniveau, auch als garantierter Quality-of-Service (QoS) bekannt, muß weniger Übertragungsenergie aufgewandt werden. Die **Energieeinsparung** kann dabei bis zu 3 Dezibel betragen, was einer Halbierung der Übertragungsenergie entspricht.

Der Schwerpunkt der vorliegenden Arbeit liegt auf der allgemeinen stochastischen Analyse eines Kanalmodells für Blockcodes und den verschiedenen Varianten der Decodierung von binären linearen Blockcodes ohne Spezialisierung auf bestimmte Codes. Ausgehend von dieser Analyse

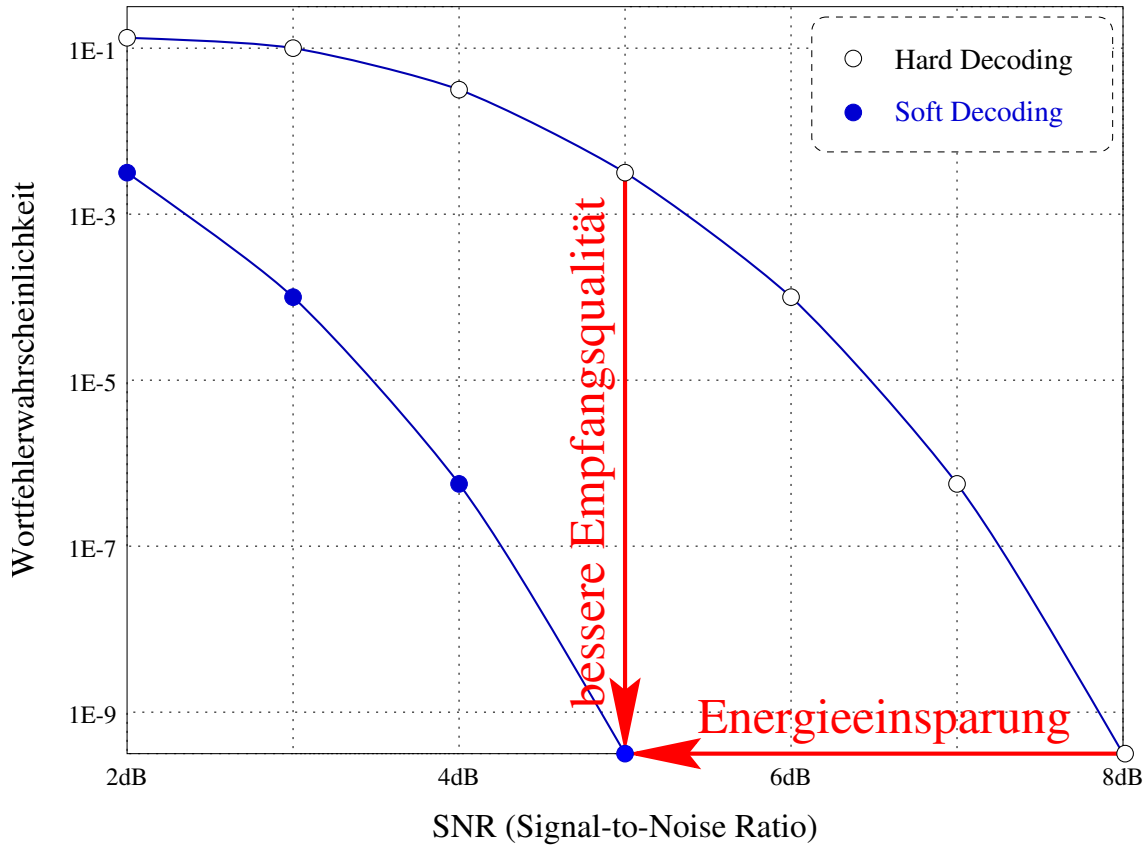

Abbildung 1.1: *Verbesserung der Empfangsqualität / Reduktion der Übertragungsenergie*

werden fehleroptimale Decodierungsverfahren abgeleitet, insbesondere ein Soft-Decision Algorithmus für bliebige binäre lineare Blockcodes und ein Algorithmus zur Soft-Output Decodierung für die Klasse der terminierten Faltungscodes.

Dreht man die Vorgehensweise um, d.h. konstruiert man spezielle Codes, die sich besonders für bestimmte Soft Decodierungsverfahren eignen, dann erhält man schnelle und approximativ fehleroptimale Paare von Codes und Decodierungsverfahren, wie die Low Density Parity Check Codes [Gal62, MN97] und Turbo-Codes [BM96, Rie97, BDMP98], welche mit iterativen Decodierungsverfahren behandelbar sind. In [Bos98, Bos99, Hub02, Hag02] wird ein Überblick über diese Codes und Methoden gegeben.

Im weiteren wird aber von keiner Spezialisierung auf diese oder andere Codes ausgegangen, sondern es wird die allgemeine Situation beliebiger linearer binärer Blockcodes analysiert und daraus werden allgemeine stochastische Aussagen abgeleitet.

In Kapitel 2 werden zunächst die benötigten Definitionen und mathematischen Grundlagen der digitalen Nachrichtenübertragung und der Kanalcodierung von binären linearen Blockcodes dargestellt. Insbesondere wird die technisch wichtige Verkettung von Blockcodes betrachtet.

Basierend auf einer stochastischen Kanalmodellierung werden in Kapitel 3 die prinzipiellen Decodierungsmethodiken klassifiziert und für jede Klasse werden Kriterien zur Konstruktion „bester“ Decodierungsverfahren in allgemeiner Form entwickelt. Die Behandlung der wichtigen verketteten Codes wird auf die Decodierung der Einzelcodes zurückgeführt, wobei die vollständige Informationsweitergabe zwischen den einzelnen Teilen der Decodierung durch eine Soft-Schnittstelle sichergestellt wird.

Für die Klasse der terminierten Faltungscodes wird in Kapitel 4 eine Soft-Output Decodierungsmethode entwickelt, bei der das Decodierungsergebnis ein „softer“ Vektor von Zuverlässigkeitswerten ist, der als Eingabe für ein nachgeschaltetes zweites Decodierungsverfahren dienen kann.

In Kapitel 5 wird ein Branch-and-Bound Verfahren zur Soft-Decision Decodierung auf Grundlage einer speziellen Quasi-Systematisierung eines beliebigen gegebenen binären linearen Blockcodes vorgestellt.

Das numerische Verhalten der entwickelten Soft Decodierungsverfahren wird in Kapitel 6 anhand von Referenzanwendungen untersucht. Durch den Vergleich mit Standardmethoden der Hard Decodierung wird die Effizienz der Soft Decodierung deutlich.

Ich möchte mich an dieser Stelle herzlich bei Prof. Dr. Dr. Stefan Schäffler bedanken für seine wertvollen fachlichen Anregungen und moralischen Stärkungen, für sein Wohlwollen und Vertrauen und nicht zuletzt auch für seine beständige Unterstützung auch in schwierigen Phasen. Bei Prof. Dr. Claus Hillermeier bedanke ich mich sehr für seine Hilfe und für die jahrelange freundschaftliche Zusammenarbeit. Weiter bedanke ich mich bei Herrn Prof. Dr. Albert Gilg und Herrn Dr. Werner Weber für ihr großes Interesse und ihre freundliche Anteilnahme.

Kapitel 2

Grundlagen

The Wheel of Time turns, and Ages come and pass, leaving memories that become legend. Legend fades to myth, and even myth is long forgotten when the Age that gave it birth comes again.

(Robert Jordan, „The Wheel of Time“)

2.1 Digitale Nachrichtenübertragung

Das Ziel der Kommunikation zwischen zwei Kommunikationspartnern ist das Austauschen von Information(en). Zur Übermittlung der Information von dem einen Partner, genannt Quelle, zum anderen Partner, genannt Senke, werden Nachrichten verwendet, die über ein Medium, genannt Kanal, übermittelt werden.

Die Information einer Nachricht ist subjektiv und technisch nicht faßbar, da sie in der Regel nur im Kontext der Kommunikationspartner verständlich ist. Eine Nachricht „FEUER“ etwa übermittelt eine andere Information zwischen den Kommunikationspartnern Brandmelder und Feuerwehr als zwischen den Kommunikationspartnern Offizier und Soldat.

Unter einer Nachricht hingegen wird im technischen Umfeld eine Zeichenfolge aus einem endlichen Zeichenvorrat verstanden, der im allgemeinen beliebig sein kann, etwa Schrift, Bilder, Symbole, Sprache, Klänge, etc. In der digitalen Nachrichtentechnik wird ein binärer Zeichenvorrat verwendet[1].

Bei der Übermittlung der Nachricht über einen physikalischen Kanal, etwa ein Glasfaserkabel, ein Koaxialkabel, ein Compact-Disc-Leselaser, eine Mobilfunk- oder Richtfunkstrecke, wird eine physikalische Repräsentation der Nachricht als Signal notwendig, welches während der Übermittlung physikalischen Störungen unterliegt. Daher erhält die Seite der Senke ein gestörtes Signal, aus dem sich die übertragene Nachricht je nach Störungsgrad eventuell nicht oder verfälscht (was schlimmer sein kann, vergleiche die Nachricht „FEUER“) rekonstruieren läßt.

Aus diesem Grund wird der Nachricht gezielt Redundanz hinzugefügt (durch die sogenannte Kanalcodierung, siehe Abschnitt 2.2) und diese mit übertragen. Mit Hilfe der Redundanz der empfangenen Signale ist es die Aufgabe des Kanaldecodierers, die ursprüngliche Nachricht zu rekonstruieren.

[1] Statt binären Zeichenvorräten (Alphabeten) können auch Alphabete verwendet werden, deren Elemente Wörter aus einem binären Alphabet sind. Die Konstruktion von Reed-Solomon-Codes beruht etwa auf solchen Symbolzeichensätzen, deren Elementzahl eine Zweierpotenz ist.

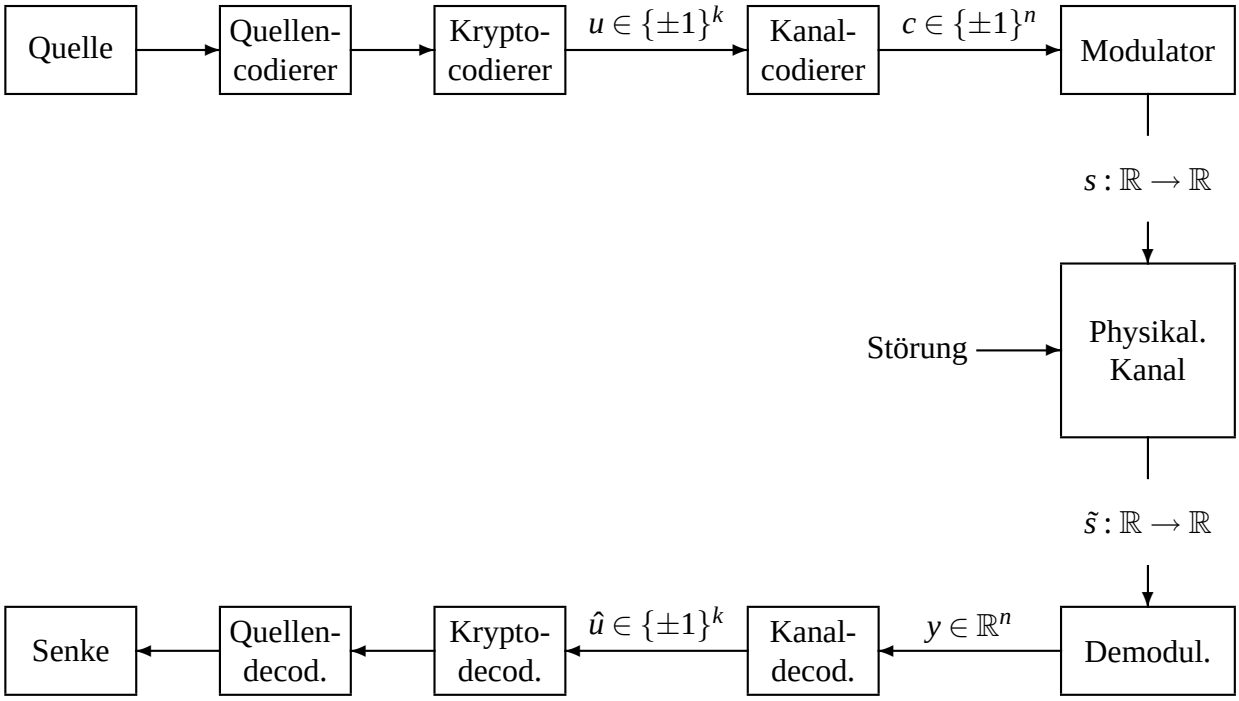

Abbildung 2.1: *Digitale Nachrichtenübertragung*

Die Abbildung 2.1 zeigt den schematischen Aufbau einer digitalen Nachrichtenübertragungsstrecke für Nachrichten, die mit Blockcodes codiert sind.

- **Quelle:** Von der Quelle aus sollen Nachrichten zur Senke geschickt werden. Die Darstellung der Nachricht an der Quelle ist beliebig und für die weiteren Betrachtungen nicht relevant. In Mobilfunkanwendungen kann zum Beispiel als Quelle ein Sprecher beziehungsweise Sprache angenommen werden.

- **Quellencodierer:** Der Quellencodierer hat die Aufgabe, die Nachrichten der Quelle so in digitale Wertefolgen (oder Zeichenketten im allgemeinen Fall) zu transformieren, daß weder Informationen verlorengehen noch unnötige Redundanzen codiert werden. Zur Reduktion der Datenmenge können Datenkomprimierungsalgorithmen angewandt werden.

- **Kryptocodierer:** Die optionale Komponente des Kryptocodierers verschlüsselt die vom Quellencodierer kommenden Nachrichten, um lesenden oder schreibenden (verfälschenden) Zugriff von Unbefugten zu verhindern, siehe dazu [Beu94]. Die Ausgabe $u \in \{\pm 1\}^k$ des Kryptocodierers ist ein Binärwort aus dem binären Zeichenvorrat $\{\pm 1\}$. Ein Kryptocodierer bietet die sogenannte „perfekte Sicherheit“, wenn alle kryptocodierten Wörter mit gleicher Wahrscheinlichkeit auftreten.

- **Kanalcodierer:** Da bei der Übertragung der Nachricht über den physikalischen Kanal mit Störungen zu rechnen ist, fügt der Kanalcodierer der Nachricht gezielt Redundanz hinzu, um dem Empfänger die Rekonstruktion der ursprünglichen Nachricht zu ermöglichen. Der Ausgang des Kanalcodierers ist ein kanalcodiertes Wort $c \in \{\pm 1\}^n$, $n \geq k$, welches mit Hilfe eines binären linearen Blockcodes[2] aus u erzeugt werden kann. Genaue Details der Kanalcodierung sind in Abschnitt 2.2 ab Seite 19 beschrieben.

[2] Wir betrachten hier nur binäre lineare Blockcodes. Im allgemeinen Fall kann eine Blockcodierung auch symbolbasiert und/oder nichtlinear erfolgen.

- **Modulator:** Die Übertragung der Nachricht über ein physikalisches Medium erfordert eine physikalische Repräsentation der Nachricht als zeitkontinuierliches Signal (hier dargestellt als Funktion $s : \mathbb{R} \to \mathbb{R}$). Verfahren zur digitalen Modulation, um ein Nachrichtensignal einem Trägersignal aufzuprägen, sind z.B. bei [DB96, Lük99, CCR01, Pro01] beschrieben. Bei der sogenannten Amplitudenumtastung werden die Bits zum Beispiel durch die Amplituden des Trägersignals moduliert (daher unter anderem auch die Bit-Darstellung als ± 1).

- **Physikalischer Kanal:** Der physikalische Kanal ist ein (weitestgehend beliebiges) physikalisches Medium, etwa ein Glasfaserkabel, ein Koaxialkabel, ein Compact-Disc-Leselaser, eine Mobilfunk- oder Richtfunkstrecke, etc. Bei der Übertragung des Signals s kommt es zu Störungen, so daß der Kanalausgang ein verfälschtes Signal $\tilde{s}$ ist.

- **Demodulator:** Der Demodulator ist das Gegenstück zum Modulator und demoduliert das empfangene Signal zu einem reellwertigen Vektor $y \in \mathbb{R}^n$. In diesen Vektor fließen das abgesandte kanalcodierte Wort $c \in \{\pm 1\}^n$ und die Kanalstörung ein. Da jetzt keine „harten“ Bits mehr vorliegen, spricht man auch von Soft-Werten.

- **Kanaldecodierer:** Die Aufgabe des Kanaldecodierers besteht in der Rekonstruktion der ursprünglichen Nachricht $u \in \{\pm 1\}^k$ unter Verwendung der in $y \in \mathbb{R}^n$ enthaltenen Informationen und dem Wissen über die Kanalcodierung und über die Eigenschaften des Störeinflusses. Der Ausgang $\hat{u} \in \{\pm 1\}^k$ des Kanaldecodierers sollte mit möglichst hoher Wahrscheinlichkeit mit der ursprünglichen Nachricht $u \in \{\pm 1\}^k$ identisch sein.

- **Kryptodecodierer:** Der Kryptodecodierer ist das Gegenstück zum Kryptocodierer, das heißt, die quellencodierte Nachricht wird hier wieder entschlüsselt.

- **Quellendecodierer:** Der Quellendecodierer bereitet die empfangene Nachricht für die Verarbeitung durch die Senke auf. Nimmt man erneut das Mobilfunkbeispiel, so könnte der Quellendecodierer etwa Sprache erzeugen.

- **Senke:** Die Senke ist der (beliebig geartete) Empfänger der Nachricht.

2.2 Kanalcodierung

Die Codierungstheorie setzt im allgemeinsten Fall keine speziellen Alphabete voraus, aus denen die Zeichen der Wörter der Codes genommen werden. In der Praxis der digitalen Nachrichtenübertragung und der Verarbeitung von Nachrichten in Computersystemen und integrierten Schaltungen wird aber stets ein digitales (zweielementiges, binäres) Alphabet verwendet[3]. Zusammen mit den notwendigen Verknüpfungen der Elemente dieser Menge wird also im weiteren ausschließlich der nachfolgend definierte binäre Körper betrachtet.

[3] Auch bei Symbolalphabeten der digitalen Nachrichtentechnik liegt ein binäres Basisalphabet zugrunde.

Definition 2.1 (binärer Körper)
Das Tripel $(\{\pm 1\}, \oplus, \odot)$ *sei der binäre Körper mit der Addition* $\oplus$ *und der Multiplikation* $\odot$*, die wie folgt definiert sind:*

$$\begin{aligned} -1 \oplus -1 &= 1, & -1 \odot -1 &= -1, \\ -1 \oplus 1 &= -1, & -1 \odot 1 &= 1, \\ 1 \oplus -1 &= -1, & 1 \odot -1 &= 1, \\ 1 \oplus 1 &= 1, & 1 \odot 1 &= 1. \end{aligned}$$

⌟

Bis auf Isomorphie gibt es nur einen binären Körper. Im Vergleich zu einer informationstechnisch oft üblichen $\{0,1\}$-Repräsentation korrespondiert -1 mit 1 und 1 mit 0.

Da $(\{\pm 1\}, \oplus, \odot)$ mit Definition 2.1 die Körpereigenschaften besitzt, ist $\{\pm 1\}^p$ für $p \in \mathbb{N}$ ein Vektorraum[4] über $(\{\pm 1\}, \oplus, \odot)$ und es können lineare Abbildung vom Vektorraum $\{\pm 1\}^k$ in den Vektorraum $\{\pm 1\}^n$ betrachtet werden.

Definition 2.2 (binärer linearer (n,k)-Blockcode)
Ein binärer linearer (n,k)*-Blockcode ist ein Tripel* (n,k,φ) *bestehend aus* $n,k \in \mathbb{N}$*,* $n \geq k$*, und einer injektiven linearen Abbildung*

$$\varphi : \{\pm 1\}^k \to \{\pm 1\}^n.$$

n heißt die Codelänge, k heißt die Codedimension.
Die Abbildung φ *heißt Codierungsabbildung.*
Ein $u \in \{\pm 1\}^k$ *heißt uncodiertes Wort und die Komponenten von u heißen Infobits.*
Ein $c \in \{\pm 1\}^n$ *heißt codiertes Wort oder Codewort und die Komponenten von c heißen Codebits.*
$\mathcal{C} := \varphi(\{\pm 1\}^k) = \{\varphi(u);\ u \in \{\pm 1\}^k\} \subseteq \{\pm 1\}^n$ *heißt die Menge der Codewörter.*

⌟

Falls nicht anders beschrieben, steht in der Folge „Blockcode“ immer für einen binären linearen (n,k)-Blockcode. Im Spezialfall $n = k$ ist $\varphi \in \mathcal{S}_n$, das heißt, die Codierungsabbildung sortiert in diesem Fall die Infobits um. Dieser Spezialfall wird als Interleaving bezeichnet und seine Bedeutung liegt in der Abschwächung von Bündelfehlern[5].

Wie oben definiert führt die Codierungsabbildung eines Blockcodes also eine Zeichenkette in eine andere (für $n > k$ längere) Zeichenkette über, die die ursprünglichen Zeichen in redundanter Form repräsentiert. Daher verwendet man eine solche Darstellung zur Übertragung über den Kanal, um mit Hilfe der Redundanz die ursprüngliche Nachricht mit hoher Wahrscheinlichkeit rekonstruieren zu können.

Definition 2.3 (Kanalcodierung)
Im technischen Zusammenhang der digitalen Nachrichtenübertragung (siehe Abbildung 2.1 auf Seite 18) nennt man einen binären linearen (n,k)*-Blockcode* (n,k,φ) *einen Kanalcode, wenn die uncodierten Wörter Eingang des* Kanalcodierers *sind und die mit* φ *codierten Wörter Ausgang des* Kanalcodierers *sind.*
Die Abbildung φ *heißt dann auch Kanalcodierungsabbildung.*
Ein $u \in \{\pm 1\}^k$ *heißt auch kryptocodiertes Wort.*
Ein $c \in \{\pm 1\}^n$ *heißt auch kanalcodiertes Wort.*

⌟

[4] Bei Blockcodes ist also aufgrund der fixen Länge die Menge der zulässigen Wörter ein Vektorraum.
[5] Vergleiche mit den gedächtnislosen Kanalmodellen in Definition 3.1 auf Seite 31.

Ein Kriterium für die Güte einer solchen Codierung ist die nachfolgend definierte Hamming-Distanz. Sie beschreibt, an wievielen Stellen sich zwei Codewörter des Blockcodes mindestens unterscheiden.

Definition 2.4 (Hamming-Abstand, Hamming-Distanz)

(i) *Für $n \in \mathbb{N}$ heißt die Metrik*[6]

$$d_{\text{ham}} : \{\pm 1\}^n \times \{\pm 1\}^n \to \mathbb{N}_0$$
$$(c, \hat{c}) \mapsto \frac{1}{2} \sum_{j=1}^{n} |c_j - \hat{c}_j|$$

Hamming-Abstand.

(ii) *Für $\mathcal{C} \subseteq \{\pm 1\}^n$, $n \in \mathbb{N}$, heißt*

$$d_{\text{ham}}(\mathcal{C}) := \min \{ d_{\text{ham}}(c, \hat{c});\ c, \hat{c} \in \mathcal{C},\ c \neq \hat{c} \}$$

die Hamming-Distanz oder die Minimaldistanz der Menge $\mathcal{C}$.

(iii) *Sei (n, k, φ) ein binärer linearer (n,k)-Blockcode. Dann heißt*

$$d_{\text{ham}}(\varphi) := d_{\text{ham}}(\varphi(\{\pm 1\}^k)) = \min \left\{ d_{\text{ham}}(\varphi(u), \varphi(\hat{u}));\ u, \hat{u} \in \{\pm 1\}^k,\ u \neq \hat{u} \right\}$$

die Hamming-Distanz oder die Minimaldistanz des binären linearen (n,k)-Blockcodes (n, k, φ).

Bei einem (n,k)-Blockcode, dessen Hamming-Distanz beispielsweise 3 sei, unterscheiden sich also alle Codewörter an mindestens drei Stellen. Daher kann der Empfänger einer verfälschten Nachricht diese korrekt decodieren, wenn an einer Stelle des Codeworts ein Bit falsch angekommen ist. Entsprechend lassen sich bei höheren Hamming-Distanzen auch mehrbitige Fehler erkennen und korrigieren.

Definition 2.5 (identischer Blockcode, äquivalenter Blockcode)
Seien (n, k, φ_1) und (n, k, φ_2) zwei binäre lineare (n,k)-Blockcodes.

(i) *(n, k, φ_1) und (n, k, φ_2) heißen identisch, falls*

$$\mathcal{C}_2 := \varphi_2(\{\pm 1\}^k) = \varphi_1(\{\pm 1\}^k) =: \mathcal{C}_1.$$

(ii) *(n, k, φ_1) und (n, k, φ_2) heißen äquivalent, falls es ein $\sigma \in \mathcal{S}_n$ gibt mit*

$$\varphi_2(\{\pm 1\}^k) = \sigma \circ \varphi_1(\{\pm 1\}^k).$$

[6] Bettet man den Körper $(\{\pm 1\}, \oplus, \odot)$ in den Körper $(\mathbb{R}, +, \cdot)$ ein, so haben die beiden Elemente des binären Körpers den Abstand 2 in $\mathbb{R}$. Daher wird in der Definition des Hamming-Abstands ein Faktor $\frac{1}{2}$ notwendig, damit die Zahl der unterschiedlichen Bits von c und $\hat{c}$ korrekt dargestellt wird.

Bei identischen Blockcodes sind also die Codewortmengen identisch, während bei äquivalenten Blockcodes die Codewortmengen identisch bis auf (eine feste) Vertauschung der Wortkomponenten sind.

Vielfach wird in der Codierungstheorie nicht zwischen Blockcodes unterschieden, die identisch oder äquivalent sind, da diese Codes viele Eigenschaften gemeinsam haben. Beispielsweise ist die wichtige Eigenschaft der Hamming-Distanz bei identischen und äquivalenten Codes gleich, wie man sich leicht überlegen kann.

An Stelle der Codewortmengen wie in Definition 2.5 können die beiden Beziehungen aber auch über die Codierungsabbildungen nachgewiesen werden.

Lemma 2.6 (identischer Blockcode, äquivalenter Blockcode über die Codierungsabbildung) *Seien (n,k,φ_1) und (n,k,φ_2) zwei binäre lineare (n,k)-Blockcodes.*

(i) *(n,k,φ_1) und (n,k,φ_2) sind identisch genau dann, wenn es einen Automorphismus ψ auf $\{\pm 1\}^k$ gibt mit*

$$\varphi_2 = \varphi_1 \circ \psi.$$

(ii) *(n,k,φ_1) und (n,k,φ_2) sind äquivalent, falls es einen Automorphismus ψ auf $\{\pm 1\}^k$ und ein $\sigma \in \mathcal{S}_n$ gibt mit*

$$\varphi_2 = \sigma \circ \varphi_1 \circ \psi.$$

Beweis. (i) „$\Longleftarrow$“: trivial.

„$\Longrightarrow$“: Seien (n,k,φ_1) und (n,k,φ_2) identisch. Zu den Einheitsvektoren $e_i \in \{\pm 1\}^k$ gibt es dann jeweils Elemente $a_i \in \{\pm 1\}^k$ mit

$$\varphi_2(e_i) = \varphi_1(a_i), \quad \text{für } i = 1,\ldots,k.$$

Da φ_2 injektiv ist und daher $\dim\langle \varphi_2(e_1),\ldots,\varphi_2(e_k)\rangle = k$, folgt $\dim\langle a_1,\ldots,a_k\rangle = k$, das heißt, $\{a_1,\ldots,a_k\}$ ist eine Basis von $\{\pm 1\}^k$.

Weiter gilt

$$\varphi_2(u) = \varphi_2\left(\bigoplus_{i=1}^{k} u_i \odot e_i\right) = \bigoplus_{i=1}^{k} u_i \odot \varphi_2(e_i) = \bigoplus_{i=1}^{k} u_i \odot \varphi_1(a_i) = \varphi_1\left(\bigoplus_{i=1}^{k} u_i \odot a_i\right) = \varphi_1(\psi(u))$$

mit $\psi(u) := \bigoplus_{i=1}^{k} u_i \odot a_i$ für $u \in \{\pm 1\}^k$. ψ ist linear und surjektiv, da $\{a_1,\ldots,a_k\}$ eine Basis von $\{\pm 1\}^k$ ist, also ist ψ ein Automorphismus auf $\{\pm 1\}^k$.

(ii) „$\Longleftarrow$“: trivial.

„$\Longrightarrow$“: Seien (n,k,φ_1) und (n,k,φ_2) äquivalent. Also gibt es ein $\sigma \in \mathcal{S}_n$, so daß $(n,k,\sigma\circ\varphi_1)$ und (n,k,φ_2) identisch sind. Mit (i) folgt dann die Existenz eines Automorphismus ψ auf $\{\pm 1\}^k$, so daß

$$\varphi_2(u) = \sigma \circ \varphi_1 \circ \psi(u), \quad \text{für alle } u \in \{\pm 1\}^k.$$

□

Eine gebräuchliche Darstellung der linearen Codierungsabbildung erfolgt über die sogenannte Generatormatrix.

Definition 2.7 (Generatormatrix)
Sei (n,k,φ) ein binärer linearer (n,k)-Blockcode. Die Matrix $G \in \{\pm 1\}^{k,n}$ mit der Eigenschaft

$$G^\top u = \varphi(u), \quad \text{für alle } u \in \{\pm 1\}^k,$$

heißt Generatormatrix des binären linearen (n,k)-Blockcodes.

Die Verwendung der transponierten Form der Matrix in der Definition folgt der traditionellen Darstellung in der Codierungstheorie. Die Existenz und Eindeutigkeit von G folgt sofort aus der Tatsache, daß φ eine lineare Abbildung auf einem endlichdimensionalen Vektorraum ist. Damit ergibt sich unmittelbar das folgende Lemma.

Lemma 2.8 (Codedefinition über die Generatormatrix)
(i) *Sei $G \in \{\pm 1\}^{k,n}$ die Generatormatrix eines binären linearen (n,k)-Blockcodes. Dann gilt*

$$\operatorname{rang}(G) = \dim\left(\left\{G^\top u;\ u \in \{\pm 1\}^k\right\}\right) = k.$$

(ii) *Sei $k,n \in \mathbb{N}$, $k < n$ und $G \in \{\pm 1\}^{k,n}$ eine Matrix mit* $\operatorname{rang}(G) = k$. *Dann ist mit*

$$\begin{aligned} \varphi : \{\pm 1\}^k &\to \{\pm 1\}^n, \\ u &\mapsto G^\top u, \end{aligned}$$

(n,k,φ) ein binärer linearer (n,k)-Blockcode.

Beweis. Die Aussage (i) folgt sofort aus der Injektivität der Codierungsabbildung. Umgekehrt impliziert der volle Rang der Matrix G in Aussage (ii) die Injektivität der Abbildung φ.

□

Als Beispiel betrachten wir einen binären linearen $(7,4)$-Blockcode $(7,4,\varphi_{\text{bsp}})$, der über die Generatormatrix

$$G_{\text{bsp}} = \begin{pmatrix} -1 & 1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & -1 & 1 & -1 \end{pmatrix} \in \{\pm 1\}^{4,7}$$

definiert sei.

Eine weitere wichtige Darstellung eines binären linearen Blockcodes erfolgt über charakterisierende (Teil-)Mengen.

Definition 2.9 (charakterisierende Mengen)
Sei (n,k,φ) ein binärer linearer (n,k)-Blockcode mit Generatormatrix $G \in \{\pm 1\}^{k,n}$. Die charakterisierenden Mengen $J_1,\ldots,J_n \subseteq \{1,\ldots,k\}$ sind dann definiert als

$$J_j := \{i \in \{1,\ldots,k\};\ G_{ij} = -1\}, \quad \text{für } 1 \le j \le n.$$

Die sieben charakterisierenden Mengen des Beispielcodes $(7,4,\varphi_{\text{bsp}})$ lauten also

$$J_1 = \{1\},\ J_2 = \{2\},\ J_3 = \{3\},\ J_4 = \{4\},\ J_5 = \{1,3,4\},\ J_6 = \{1,2,3\},\ J_7 = \{2,3,4\}.$$

Mit den jetzt vorhandenen Mitteln können wir ein Codewort auf verschiedene äquivalente Weisen darstellen.

Lemma 2.10 (Codewort-Darstellung)
Sei (n,k,φ) ein binärer linearer (n,k)-Blockcode mit Generatormatrix $G \in \{\pm 1\}^{k,n}$ und charakterisierenden Mengen $J_1,\ldots,J_n \subseteq \{1,\ldots,k\}$. Sei $u \in \{\pm 1\}^k$ ein uncodiertes Wort und $c \in \{\pm 1\}^n$ ein Codewort. Dann sind die folgenden Aussagen äquivalent:

(i)
$$c = \varphi(u),$$

(ii)
$$c = G^\top u,$$

(iii)
$$c_j = \bigoplus_{i \in J_j} u_i, \quad \text{für } 1 \le j \le n,$$

(iv)
$$c_j = \prod_{i \in J_j} u_i, \quad \text{für } 1 \le j \le n.$$

⌟

Beweis. Die Aussagen (i)-(iii) des Lemmas folgen sofort. Aussage (iv) gilt, wenn man $\{\pm 1\}$ als Teilmenge von $\mathbb{R}$ betrachtet.

□

Somit gilt für den Beispielcode $(7,4,\varphi_{\text{bsp}})$, daß

$$\begin{aligned}
c_1 &= u_1,\\
c_2 &= u_2,\\
c_3 &= u_3,\\
c_4 &= u_4,\\
c_5 &= u_1 \oplus u_3 \oplus u_4 = u_1 \cdot u_3 \cdot u_4,\\
c_6 &= u_1 \oplus u_2 \oplus u_3 = u_1 \cdot u_2 \cdot u_3,\\
c_7 &= u_2 \oplus u_3 \oplus u_4 = u_2 \cdot u_3 \cdot u_4,
\end{aligned}$$

wobei $\cdot$ die Multiplikationsverknüpfung von $(\mathbb{R},+,\cdot)$ nach Einbettung von $(\{\pm 1\},\oplus,\odot)$ in $(\mathbb{R},+,\cdot)$ ist.

Binäre lineare Blockcodes lassen sich gemäß ihrer Darstellung wie folgt klassifizieren.

Definition 2.11 (systematisch, quasi-systematisch, unsystematisch)
Sei (n,k,φ) ein binärer linearer (n,k)-Blockcode. (n,k,φ) heißt ein

(i) *systematischer Blockcode, falls für die charakterisierenden Mengen des Blockcodes gilt:*

$$J_j = \{j\}, \quad \text{für } j = 1,\ldots,k.$$

Systematische Blockcodes heißen auch separierbare Blockcodes.

(ii) *quasi-systematischer Blockcode, falls es eine Permutation $\sigma \in \mathcal{S}_n$ gibt, so daß $(n,k,\sigma\circ\varphi)$ ein systematischer Blockcode ist.*

(iii) *unsystematischer Blockcode, falls (n,k,φ) kein systematischer Blockcode ist.*

⌟

Aus der Definition der charakterisierenden Mengen folgt sofort, daß die ersten k Spalten der Generatormatrix eines systematischen binären linearen (n,k)-Blockcodes die Einheitsmatrix bilden. Bei quasi-systematischen Codes stehen die k Einheitsspalten an beliebigen Stellen der Generatormatrix.

Bei systematischen Blockcodes sind die ersten k Stellen des Codeworts $c \in \{\pm1\}^n$ identisch mit dem uncodierten Wort $u \in \{\pm1\}^k$, also

$$c_j = u_j, \quad \text{für alle } j = 1,\ldots,k.$$

Die restlichen Stellen c_j, $j = k+1,\ldots,n$, werden dann als Prüfstellen oder Paritätsstellen bezeichnet.

Der Beispielcode $(7,4,\varphi_{\text{bsp}})$ ist somit ein systematischer binärer linearer $(7,4)$-Blockcode.

In der Codierungstheorie nimmt die Konstruktion von Codes über Polynome einen sehr breiten Raum ein. Gewünschte Code-Eigenschaften lassen sich durch algebraische Polynom-Eigenschaften erzwingen. Da in der vorliegenden Arbeit Polynome ausschließlich zur Definition von Codierungsabbildungen Verwendung finden, benötigen wir lediglich eine Konstruktionsvorschrift.

Definition 2.12 (polynomerzeugter Code, Generatorpolynom)
Es seien $n,k \in \mathbb{N}$, $n \geq k$, und es sei $g(x) \in \{\pm1\}[x]$ ein Polynom vom Grad $n-k$, also

$$g(x) = g_0 \oplus g_1 \odot x \oplus g_2 \odot x^2 \oplus \ldots \oplus g_{n-k} \odot x^{n-k}, \quad \text{mit } g_j \in \{\pm1\},\ j = 0,\ldots,n-k,$$

wobei neben $g_{n-k} = -1$ zusätzlich gelte $g_0 = -1$.

Es heißt (n,k,φ) der durch das Polynom $g(x)$ erzeugte binäre lineare (n,k)-Blockcode, wenn für die Generatormatrix $G \in \{\pm1\}^{k,n}$ des Blockcodes gilt:

$$G := \begin{pmatrix} g_0 & g_1 & \ldots & \ldots & g_{n-k} & 1 & 1 & \ldots & 1 \\ 1 & g_0 & g_1 & \ldots & \ldots & g_{n-k} & 1 & \ldots & 1 \\ \vdots & \ddots & \ddots & \ddots & & & \ddots & \ddots & \vdots \\ 1 & \ldots & 1 & g_0 & g_1 & \ldots & \ldots & g_{n-k} & 1 \\ 1 & \ldots & \ldots & 1 & g_0 & g_1 & \ldots & \ldots & g_{n-k} \end{pmatrix}.$$

Das Polynom $g(x)$ heißt dann das Generatorpolynom des binären linearen (n,k)-Blockcodes (n,k,φ).

⌟

Da $g_0 = -1$ bzw. $g_{n-k} = -1$ und folglich $\operatorname{rang}(G) = k$, folgt mit Lemma 2.8 auf Seite 23, daß (n,k,φ) ein binärer linearer (n,k)-Blockcode ist. Zur Erzeugung „guter“ Codes (etwa zyklischer Codes) werden im allgemeinen noch weitere Eigenschaften des Generatorpolynoms gefordert.

Zur Erhöhung der Lesbarkeit werden Generatorpolynome meist als Polynome aus $\{0,1\}[x]$ dargestellt. Man betrachte zum Beispiel ein Generatorpolynom $g_{\text{bsp}}(x) \in \{\pm 1\}[x]$ definiert als

$$g_{\text{bsp}}(x) = -1 \oplus -1 \odot x \oplus +1 \odot x^2 \oplus -1 \odot x^3 = -1 \oplus x \oplus x^3.$$

In $\{0,1\}[x]$ entspricht

$$1 + x + x^3$$

dem Generatorpolynom $g_{\text{bsp}}(x)$ und wird in der Regel in dieser Form zur Darstellung verwendet.

Sei $(7,4,\tilde{\varphi}_{\text{bsp}})$ der durch $g_{\text{bsp}}(x)$ erzeugt $(7,4)$-Blockcode. Dann lautet die Generatormatrix dieses Blockcodes

$$\tilde{G}_{\text{bsp}} = \begin{pmatrix} -1 & -1 & 1 & -1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 & -1 & 1 & -1 \end{pmatrix}.$$

Lemma 2.13 (Systematisierung eines polynomerzeugten Codes)
Es seien $n,k \in \mathbb{N}$, $n \geq k$, und es sei $g(x) \in \{\pm 1\}[x]$ ein Polynom vom Grad $n-k$, also

$$g(x) = g_0 \oplus g_1 \odot x \oplus g_2 \odot x^2 \oplus \ldots \oplus g_{n-k} \odot x^{n-k}, \quad \text{mit } g_j \in \{\pm 1\},\ j = 0,\ldots,n-k,$$

wobei neben $g_{n-k} = -1$ zusätzlich gelte $g_0 = -1$, und es sei (n,k,φ) der durch das Polynom $g(x)$ erzeugte binäre lineare (n,k)-Blockcode. Dann gibt es genau einen zu (n,k,φ) identischen systematischen binären linearen (n,k)-Blockcode $(n,k,\hat{\varphi})$.

Beweis. „$\Longrightarrow$“: Unter den genannten Voraussetzungen ist die Existenz eines identischen systematischen binären linearen (n,k)-Blockcodes $(n,k,\hat{\varphi})$ nachzuweisen. Nach Definition 2.12 bilden die ersten k Spalten der Generatormatrix G von (n,k,φ) eine reguläre obere Dreiecksmatrix. Daher kann G ohne Spaltenvertauschungen durch reine Zeilenoperationen (Gauß-Diagonalisierung) in systematische Form transformiert werden, das heißt, es gibt eine reguläre Matrix $A \in \{\pm 1\}^{k,k}$, so daß $\hat{G}$ mit

$$\hat{G} = AG$$

die Generatormatrix eines systematischen Codes ist, also

$$\hat{\varphi}(u) = \hat{G}^\top u = (AG)^\top u = G^\top (A^\top u) = \varphi(\psi(u)) = \varphi \circ \psi(u),$$

wobei ψ der zu $A^\top$ gehörige Automorphismus ist. Somit ist der systematische Blockcode $(n,k,\hat{\varphi})$ identisch zu (n,k,φ).

„$\Longleftarrow$“: Seien $(n,k,\hat{\varphi})$ und $(n,k,\tilde{\varphi})$ zwei zu (n,k,φ) identische systematische binäre lineare (n,k)-Blockcodes. Also sind auch $(n,k,\hat{\varphi})$ und $(n,k,\tilde{\varphi})$ zueinander identisch, das heißt, es gibt einen Automorphismus ψ auf $\{\pm 1\}^k$ mit $\tilde{\varphi} = \hat{\varphi} \circ \psi$. Aufgrund der Systematik beider Blockcodes gilt

$$\begin{pmatrix} u \\ \bullet \end{pmatrix} = \tilde{\varphi}(u) = \hat{\varphi}(\psi(u)) = \begin{pmatrix} \psi(u) \\ \bullet \end{pmatrix}, \quad \text{für alle } u \in \{\pm 1\}^k,$$

also ist ψ die identische Abbildung und somit gilt $\hat{\varphi} \equiv \tilde{\varphi}$.

□

Hat man nicht die Voraussetzungen an den Blockcode, wie hier gegeben, so läßt sich dennoch zu jedem binären linearen (n,k)-Blockcode ein (nicht eindeutiger) äquivalenter systematischer Blockcode angeben und ein (nicht eindeutiger) identischer quasi-systematischer Blockcode.

Aufgrund der Eindeutigkeitsaussage von Lemma 2.13 läßt sich ein eindeutiger systematischer Code über ein Generatorpolynom erzeugen.

Definition 2.14 (systematischer polynomerzeugter Code)
Es seien $n,k \in \mathbb{N}$, $n \geq k$, und es sei $g(x) \in \{\pm 1\}[x]$ ein Polynom vom Grad $n-k$, also

$$g(x) = g_0 \oplus g_1 \odot x \oplus g_2 \odot x^2 \oplus \ldots \oplus g_{n-k} \odot x^{n-k}, \quad \text{mit } g_j \in \{\pm 1\},\ j = 0,\ldots,n-k,$$

wobei neben $g_{n-k} = -1$ zusätzlich gelte $g_0 = -1$.

Es heißt (n,k,φ) der durch das Polynom $g(x)$ erzeugte systematische binäre lineare (n,k)-Blockcode, wenn (n,k,φ) der eindeutige systematische binäre lineare (n,k)-Blockcode ist, der zu dem von $g(x)$ erzeugten binären linearen (n,k)-Blockcode identisch ist.

Im Beispiel läßt sich die Matrix $\tilde{G}_{\text{bsp}}$ durch Zeilenoperationen zur Matrix G_{bsp} des ersten Beispiels transformieren, das heißt,

$$G_{\text{bsp}} = A_{\text{bsp}} \tilde{G}_{\text{bsp}}$$

mit einer regulären Matrix $A_{\text{bsp}} \in \{\pm 1\}^{4,4}$. Somit ist $(7,4,\varphi_{\text{bsp}})$ der (eindeutige) durch das Polynom $g_{\text{bsp}}(x)$ erzeugte systematische binäre lineare $(7,4)$-Blockcode[7].

Für vertiefende und weitergehende Betrachtungen der Codierungstheorie sei auf [PW72, LC83, Fri95, HQ95, Jun95, Pro01, Roh95, Bos98, Bos99] verwiesen.

2.3 Verkettete Kanalcodierung

In der Praxis wird die Kanalcodierung oftmals nicht in einem, sondern in zwei oder mehr Codierungsschritten durchgeführt. Entsteht ein Code durch die Hintereinanderausführung einzelner Codes, so spricht man seit [For66] von einem verketteten Code.

Definition 2.15 (verketteter binärer linearer (n,k)-Blockcode)
Ein verketteter binärer linearer (n,k)-Blockcode ist ein Tupel $((n_1,k_1,\varphi_1),\ldots,(n_m,k_m,\varphi_m))$ bestehend aus m binären linearen (n_i,k_i)-Blockcodes mit $n,k,m,n_i,k_i \in \mathbb{N}$, $i = 1,\ldots,m$, und

$$k = k_1 \leq n_1 = k_2 \leq n_2 = k_3 \leq \ldots n_m = n.$$

Die Codierungsabbildung des verketteten Codes ist

$$\begin{aligned} \varphi:\ \{\pm 1\}^k &\to \{\pm 1\}^n, \\ u &\mapsto \varphi_m \circ \varphi_{m-1} \circ \ldots \circ \varphi_1(u). \end{aligned}$$

[7] $(7,4,\varphi_{\text{bsp}})$ ist der systematische $(7,4)$-BCH-Code. Die numerische Untersuchung dieses überschaubaren und simplen Beispiels ist ab Seite 130 aufgeführt.

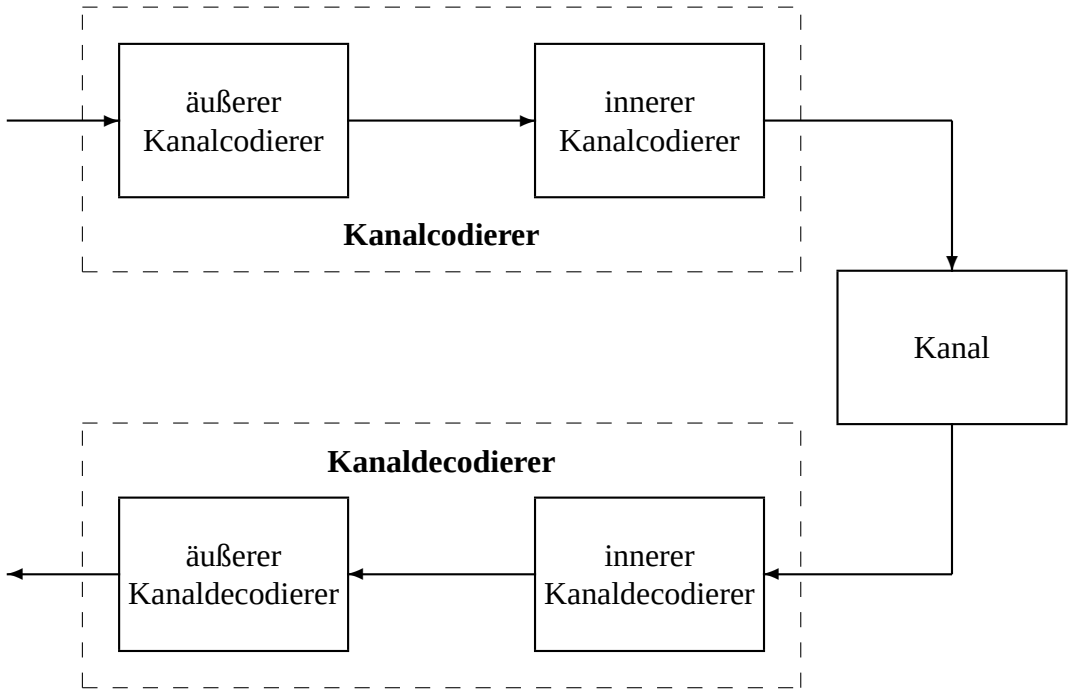

Abbildung 2.2: *Verkettete Kanalcodierung*

In der Formulierung der Definition ist die Aussage des folgenden einfachen Lemmas schon impliziert.

Lemma 2.16 (verketteter binärer linearer (n,k)-Blockcode)
Sei $((n_1,k_1,\varphi_1),\ldots,(n_m,k_m,\varphi_m))$ ein verketteter binärer linearer (n,k)-Blockcode. Dann ist $(n,k,\varphi_m\circ\varphi_{m-1}\circ\ldots\circ\varphi_1)$ ein binärer linearer (n,k)-Blockcode.

Beweis. Die Aussage ist klar, da alle φ_i, $i=1,\ldots,m$, injektive lineare Abbildungen sind und somit auch $\varphi_m\circ\varphi_{m-1}\circ\ldots\circ\varphi_1$ eine injektive lineare Abbildung ist.

□

In praktischen Beispielen scheint bisweilen eine Verallgemeinerung der obigen Definition verwendet zu werden, nämlich bei der sogenannten Punktierung von Codes (z.B. bei systematischen Codes oder Faltungscodes), bei der Stellen aus dem Codewort gestrichen werden (Verkürzung). Man darf allerdings das Paar Codierung+Verkürzung nicht fälschlicherweise als verketteten Code betrachten, sondern muß den Vorgang als **eine** Codierung ansehen, die dann notwendigerweise eine injektive Codierungsabbildung besitzen muß.

Besteht eine Kanalcodierung (vergleiche Definition 2.3 auf Seite 20) aus der Verkettung zweier Blockcodes, so spricht man oft von der Verkettung einer äußeren Kanalcodierung mit einer inneren Kanalcodierung, siehe Abbildung 2.2, wobei „außen“ und „innen“ bisweilen vom Betrachter abhängen. In der Abbildung wurden Modulator, physikalischer Kanal und Demodulator zu einem „Kanal“ zusammengefaßt.

Sind $\varphi_{\text{außen}}$ und φ_{innen} die Codierungsabbildungen der beiden entsprechenden Einzelcodes, so ist

$$\varphi=\varphi_{\text{innen}}\circ\varphi_{\text{außen}}$$

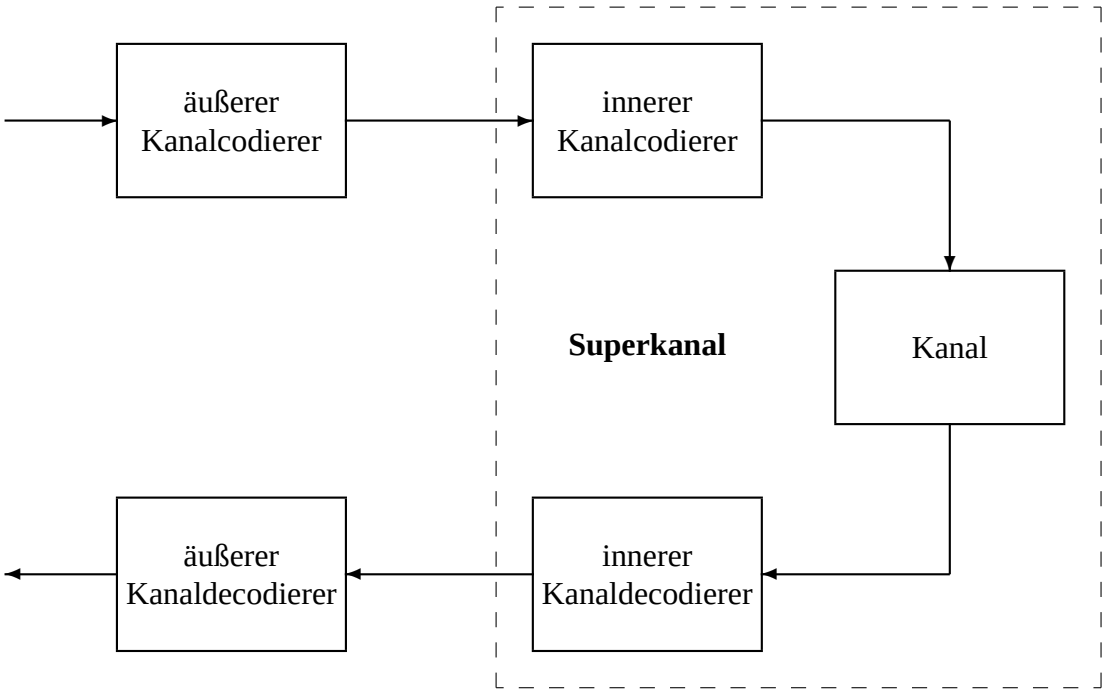

Abbildung 2.3: *Superkanal*

die Codierungsabbildung des verketteten Codes. In Abschnitt 3.6 ab Seite 55 wird detailiert auf die möglichen Eingänge und Ausgänge der Elemente in Abbildung 2.2 eingegangen.

Faßt man den inneren Kanalcodierer, den Kanal und den inneren Kanaldecodierer zu einer Einheit zusammen, siehe Abbildung 2.3, so erhält man einen Superkanal. Verwendet man geeignete Decodierungsalgorithmen (siehe Lemma 3.25 auf Seite 50 und Kapitel 4), so besitzt der Superkanal gegenüber dem ursprünglichen Kanal eine verringerte Kanalstörung[8], vergleiche auch [Fri95, BGH$^+$00]. Unter diesem Blickwinkel können also schlechte physikalische Eigenschaften des Kanals mit Hilfe der inneren Codierung/Decodierung ausgeglichen werden.

[8] Siehe dazu die numerischen Ergebnisse in Abschnitt 6.3.

Kapitel 3

Decodierung binärer linearer Blockcodes

...I will seek for the opening words. I once knew every spell in all tongues of Elves or Men or Orcs, that was ever used for such a purpose. I can still remember ten score of them without searching in my mind. But only a few trials, I think, will be needed; and I shall not have to call on Gimli for words of the secret dwarf-tongue that they teach to none. The opening words were Elvish, like the writing on the arch: that seems certain.

(Gandalf; J.R.R.Tolkien, „The Fellowship of the Ring“)

3.1 Stochastische Kanalmodellierung

Unter einer Decodierung versteht man den Vorgang der Rekonstruktion $\hat{u} \in \{\pm 1\}^k$ eines uncodierten Codewortes $u \in \{\pm 1\}^k$, vergleiche Abbildung 2.1 auf Seite 18. Wie in den Abschnitten 2.2 und 2.3 beschrieben, wird u zunächst zu einem Codewort $c \in \{\pm 1\}^n$ kanalcodiert und dann über einen Kanal zu einem Empfänger der Nachricht geschickt. Der Kanaldecodierer soll nun so konstruiert werden, daß mit dem empfangenen Vektor $y \in \mathbb{R}^n$ und dem Wissen über die Codierung und die Art der Störung auf dem Kanal die Rekonstruktion $\hat{u}$ nur mit einer minimalen Fehlerwahrscheinlichkeit nicht mit u übereinstimmt[1].

Dazu betrachten wir zunächst eine stochastische Kanalmodellierung, die uns die Mittel an die Hand geben soll, möglichst optimale Decodierungsmethoden zu entwickeln. Die folgende abstrakte Definition eines n-Kanals[2] umfaßt die in Abschnitt 2.2 betrachteten Komponenten *Modulator*, *physikalischer Kanal* und *Demodulator*. Eine schematische Darstellung dieser Aggregation ist durch Abbildung 3.1 gegeben. Auch der in Abbildung 2.3 auf Seite 29 motivierte Superkanal wird von nachfolgender Definition umfaßt.

Im folgenden werden alle Aussagen so allgemein wie möglich anhand beliebiger (stetiger) n-Kanäle getroffen. Erst dann erfolgt eine Spezialisierung auf den Standardfall der AWGN-Kanäle (vergleiche Definition 3.3). Damit kann diese Arbeit auch als Grundlage für die Spezialisierung auf andere Kanäle dienen.

[1] In der Praxis ist zudem die Echtzeitfähigkeit eines solchen Kanaldecodierers eine wichtige Randbedingung.

[2] Kanalmodelle, die die stochastischen Eigenschaften des physikalischen Kanals bzw. die Modulation und Demodulation betrachten sind z.B. in [DR87, Pät99, Pro01] zu finden

Definition 3.1 (n-Kanal, Kanaleingabe, Kanalausgabe)
Es sei $(\Omega, \mathcal{S}, P)$ ein Wahrscheinlichkeitsraum. Für $n \in \mathbb{N}$ heißt eine Abbildung

$$\mathcal{K}\colon\ \{\pm 1\}^n \times \Omega \to \mathbb{R}^n$$

ein n-Kanal, falls für jedes $c \in \{\pm 1\}^n$ die Abbildungen

$$\begin{aligned}\mathcal{K}_c\colon\ \Omega &\to \mathbb{R}^n,\\ \omega &\mapsto \mathcal{K}(c,\omega),\end{aligned}$$

n-dimensionale reelle Zufallsvariablen sind. $\mathcal{K}$ heißt ein gedächtnisloser n-Kanal, falls für jedes $c \in \{\pm 1\}^n$ die Komponenten von $\mathcal{K}_c$ stochastisch unabhängig sind.
Sind $C\colon\ \Omega \to \{\pm 1\}^n$ und $Y\colon\ \Omega \to \mathbb{R}^n$ zwei n-dimensionale Zufallsvariablen mit

$$Y(\omega) = \mathcal{K}\left(C(\omega), \omega\right), \quad \text{für alle } \omega \in \Omega,$$

und ist C stochastisch unabhängig von $\mathcal{K}_c$ für alle $c \in \{\pm 1\}^n$, dann heißt C die (Zufallsvariable der) Kanaleingabe und Y die (Zufallsvariable der) Kanalausgabe.

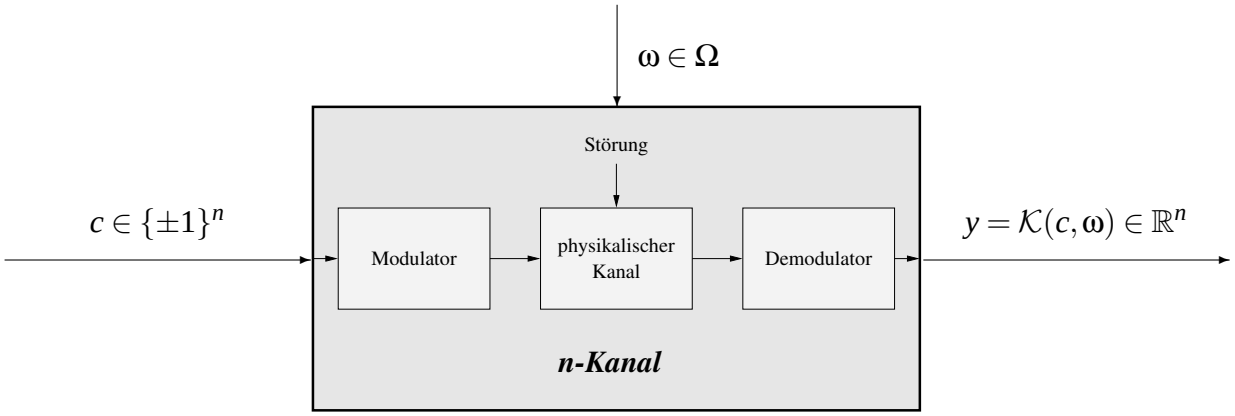

Abbildung 3.1: *n-Kanal*

Aus offensichtlichen Gründen ist die Kanalausgabe Y als Zufallsvariable modelliert, da in sie ja die Kanalstörung eingeht. Die Kanaleingabe C erscheint hier ebenso als Zufallsvariable, da dem Empfänger die Wahl eines speziellen $c \in \{\pm 1\}^n$ unbekannt ist (anderenfalls wäre die Decodierung auch unnötig).

Die Definition der Gedächtnislosigkeit bei n-Kanälen bezieht sich auf die stochastische Unabhängigkeit der Zufallsvariablen der einzelnen (übertragenen) Bits eines Codeworts[3]. Implizit ist durch die Kanaldefinition aber generell eine Gedächtnislosigkeit des Übertragungssystems bezogen auf ganze Codewörter enthalten, da die Übertragung eines Codeworts als unabhängig von der Übertragung vorhergehender Codewörter angesehen wird. Betrachtet man eine Sequenz von Codewortübertragungen, so ist zur Modellierung (in Analogie zur Stichprobendefinition in der Statistik) also eine Folge $\mathcal{K}^1, \mathcal{K}^2, \ldots, \mathcal{K}^m$ von n-Kanälen zu betrachten, bei denen für jedes $c \in \{\pm 1\}^n$ die $\mathcal{K}_c^i$ für alle $i \in \{1, \ldots, m\}$ identisch verteilt und stochastisch unabhängig sind.

[3]Da in der Realität Fehler oft in Bündeln auftreten, verwendet man Interleaving zum „Durchschütteln“ der Codebits, um so die stochastische Unabhängigkeit der zugrundeliegenden Zufallsvariablen zu approximieren.

Für genauere Betrachtungen sind Spezialisierungen und zusätzliche Annahmen über die Kanaleigenschaften notwendig. Man sollte sich stets bewußt sein, daß sinnvoll zu wählende Modellannahmen wichtige Entscheidungen treffen, um Decodiermethoden zu klassifizieren, zu entwerfen und zu bewerten, aber daß die Modellannahmen die praktische Situation lediglich approximieren und keinesfalls (vollständig) beschreiben. In späteren Kapiteln werden wir uns im besonderen zentral mit stetigen n-Kanälen beschäftigen.

Definition 3.2 (stetiger n-Kanal)
Es sei $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum. Für $n\in\mathbb{N}$ heißt eine Abbildung

$$\mathcal{K}:\ \{\pm1\}^n\times\Omega\to\mathbb{R}^n$$

ein stetiger n-Kanal, falls $\mathcal{K}$ ein n-Kanal ist und für jedes $c\in\{\pm1\}^n$ die Abbildungen

$$\begin{aligned}\mathcal{K}_c:\ \Omega&\to\mathbb{R}^n,\\ \omega&\mapsto\mathcal{K}(c,\omega),\end{aligned}$$

n-dimensionale absolutstetige reelle Zufallsvariablen mit reellwertigen[4] Dichten

$$f_c:\ \mathbb{R}^n\to\mathbb{R}_0^+,$$

der Bildmaße sind, das heißt,

$$P_{\mathcal{K}_c}(A):=P\left(\{\omega\in\Omega;\ \mathcal{K}_c(\omega)\in A\}\right)=P\left(\{\omega\in\Omega;\ \mathcal{K}(c,\omega)\in A\}\right)=\int\limits_A f_c\,d\lambda^n,$$

für alle $A\in\mathcal{B}^n$.

Der wichtigste stetige n-Kanal ist derjenige, bei dem sich die Kanalstörung über eine normalverteilte Zufallsvariable darstellen läßt.

Definition 3.3 (AWGN-Kanal)
Es sei $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum. Ein stetiger n-Kanal $\mathcal{K}$ heißt ein AWGN-Kanal[5] (Additive White Gaussian Noise), wenn es eine $\mathcal{N}(0,\sigma^2 I_n)$ normalverteilte Zufallsvariable $Z:\Omega\to\mathbb{R}^n$ gibt ($\sigma^2>0$) mit

$$\mathcal{K}(c,\omega)=c+Z(\omega),\quad \text{für alle } c\in\{\pm1\}^n,\ \omega\in\Omega.$$

σ^2 heißt dann auch die bitweise Varianz der Kanalstörung.

[4] Mit dem Satz A.32 von Radon-Nikodym existiert eine nichtnegative Dichte $\tilde{f}_c:\ \mathbb{R}^n\to\bar{\mathbb{R}}$. Da

$$\int\limits_{\mathbb{R}^n}\tilde{f}_c\,d\lambda^n=P_{\mathcal{K}_c}(\mathbb{R}^n)=1<\infty,$$

ist $\left\{x\in\mathbb{R}^n;\ \tilde{f}_c(x)=+\infty\right\}$ eine λ^n-Nullmenge und folglich kann die Dichte auch reellwertig gewählt werden. Daher ist die Dichteaussage in der Definition 3.2 nur eine Bezeichnungsfestlegung und keine Einschränkung.

[5] Üblicherweise wird der Begriff AWGN für Rauschprozesse in der Signaldarstellung physikalischer Kanäle verwendet [Pro01]. Nach Verknüpfung mit Modulationsmethoden und geeigneten Demodulationsverfahren ergibt sich daraus die hier verwendete n-Kanaldarstellung für Kanalcodierungsverfahren. Daher übertragen wir den Begriff auf die entsprechenden n-Kanäle.

Obwohl wir im allgemeinen immer von stetigen n-Kanälen bei der Modellierung des physikalischen Kanals ausgehen, benötigen wir für die weitere Darstellung bei verketteten Codes den Begriff des diskreten n-Kanals. Er gewinnt seine Bedeutung bei den Hard-Decision Methoden.

Definition 3.4 (diskreter n-Kanal)
Es sei $(\Omega, \mathcal{S}, P)$ ein Wahrscheinlichkeitsraum. Für $n \in \mathbb{N}$ heißt eine Abbildung

$$\mathcal{K}: \{\pm 1\}^n \times \Omega \to \{\pm 1\}^n$$

ein diskreter n-Kanal, falls $\mathcal{K}$ mit Erweiterung der Bildmenge auf den $\mathbb{R}^n$ ein n-Kanal ist und für jedes $c \in \{\pm 1\}^n$ die Abbildungen

$$\begin{aligned} \mathcal{K}_c: \Omega &\to \{\pm 1\}^n, \\ \omega &\mapsto \mathcal{K}(c, \omega), \end{aligned}$$

n-dimensionale diskrete Zufallsvariablen sind.
Sind $C: \Omega \to \{\pm 1\}^n$ und $\hat{C}: \Omega \to \{\pm 1\}^n$ zwei n-dimensionale Zufallsvariablen mit

$$\hat{C}(\omega) = \mathcal{K}(C(\omega), \omega), \quad \text{für alle } \omega \in \Omega,$$

und ist C stochastisch unabhängig von $\mathcal{K}_c$ für alle $c \in \{\pm 1\}^n$, dann heißt C die (Zufallsvariable der) Kanaleingabe und $\hat{C}$ die diskrete (Zufallsvariable der) Kanalausgabe. ⌟

Analog zum allgemeinen Fall heißt $\mathcal{K}$ diskreter gedächtnisloser n-Kanal (DMC, discrete memoryless channel), falls für jedes $c \in \{\pm 1\}^n$ die Komponenten von $\mathcal{K}_c$ stochastisch unabhängig sind.

Ist $\mathcal{K}$ ein stetiger n-Kanal, so läßt sich ein diskreter n-Kanal $\hat{\mathcal{K}}$ sofort durch Rundung der Kanalausgabe erzeugen:

$$\hat{\mathcal{K}}(c, \omega) := \begin{cases} +1, & \text{falls } \mathcal{K}(c, \omega) \geq 0 \\ -1, & \text{sonst} \end{cases}, \quad \text{für alle } c \in \{\pm 1\}^n,\ \omega \in \Omega.$$

Es ist sofort klar, daß der so konstruierte n-Kanal $\hat{\mathcal{K}}$ weniger Information als der ursprüngliche n-Kanal $\mathcal{K}$ übertragen kann, da beispielsweise die Komponentenwerte $+0.1$ und $+10.0$ gleichermaßen zu $+1$ gerundet werden, aber der genaue Wert der Komponenten eine wichtige Information im Codewortzusammenhang darstellt.

3.2 Klassifikation der Decodierungs-Methodiken

Je nach vorliegendem Kanalmodell und Decodierungsziel lassen sich drei prinzipielle Vorgehensweisen bei der Decodierung unterscheiden. Diese Klassifikation erfolgt nach den möglichen Kombinationen der Ein- und Ausgänge eines Kanaldecodierers, siehe Abbildung 3.2.

- **Hard-Decision Decodierung (Seite 35ff):**
  Bei der Hard-Decision Decodierung wird ein diskreter n-Kanal als Übertragungsmedium angenommen oder ein stetiger n-Kanal komponentenweise gerundet, obwohl der dabei entstehende Informationsverlust sehr hoch sein kann. Bei jeder Decodierung dient also ein „hartes“ Wort $\hat{c} \in \{\pm 1\}^n$ als Entscheidungsgrundlage für ein „hartes“ Decodierungsergebnis $\hat{u} \in \{\pm 1\}^k$. Daher bezeichnet man diesen Decodierungsvorgang als **Hard-Decision**. Bei allgemeinen Blockcodes ist diese Vorgehensweise bislang die Standardmethode zur Decodierung.

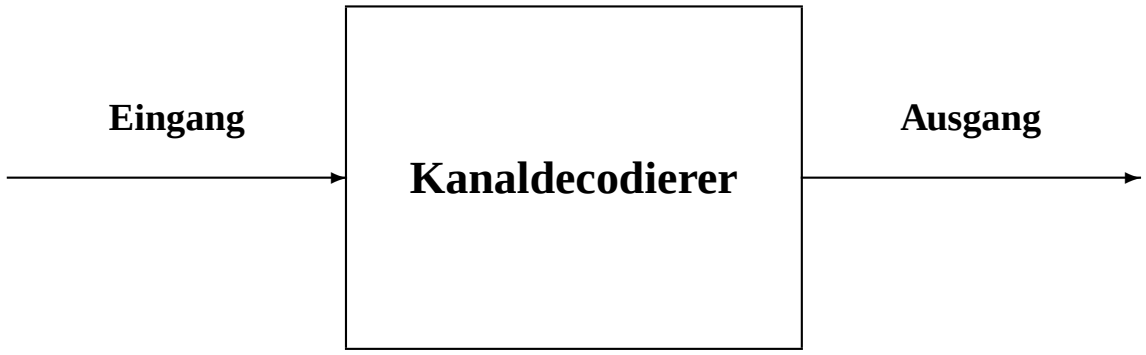

Abbildung 3.2: *Kanaldecodierer*

- **Soft-Decision Decodierung (Seite 39ff):**
  Es wird ein stetiger n-Kanal als Übertragungsmedium angenommen und bei jeder Decodierung die gesamte (Soft-)Information einer Realisierung $y \in \mathbb{R}^n$ der Kanalausgabe verwendet, um ein „hartes“ Decodierungsergebnis $\hat{u} \in \{\pm 1\}^k$ zu erzeugen. Daher bezeichnet man diesen Decodierungsvorgang als **Soft-Decision**.

- **Soft-Output Decodierung (Seite 49ff):**
  Unter den gleichen Voraussetungen wie bei der Soft-Decision Decodierung soll nicht nur ein Decodierungsergebnis $\hat{u} \in \{\pm 1\}^k$ erzeugt werden, sondern zusätzlich ein Vektor $x \in \mathbb{R}^k$ (**Soft-Outputs**), der komponentenweise ein Zuverlässigkeitsmaß für jede Komponente von $\hat{u}$ darstellt. Die Soft-Outputs können u.a. zur Fehlererkennung und bei der Decodierung verketteter Codes eingesetzt werden.

Die theoretisch vorhandene Variante, daß Soft-Outputs erzeugt werden sollen, wenn lediglich „harte“ Eingangsdaten für den Decodierer vorliegen, spielt technisch keine Rolle und kann als Spezialfall der Soft-Output Decodierung bei stetigen Kanälen angesehen werden.

3.3 Hard-Decision Decodierung

Bei einer Hard-Decision Decodierung steht zur Entscheidung für ein Wort $\hat{u} \in \{\pm 1\}^k$ ein „hartes“ Wort $\hat{c} \in \{\pm 1\}^n$ zur Verfügung. Die folgende Definition einer Decodierungsabbildung ist das Analogon zur Codierungsabbildung in Definition 2.2 auf Seite 20.

Definition 3.5 (Hard-Decision Decodierungsabbildung)
Es sei $(\Omega, \mathcal{S}, P)$ ein Wahrscheinlichkeitsraum, $n,k \in \mathbb{N}$, $n > k$. Weiter sei (n,k,φ) ein binärer linearer (n,k)-Blockcode und $\mathcal{K}$ ein diskreter gedächtnisloser n-Kanal. Eine Abbildung

$$\begin{aligned} \delta_{\mathrm{HD}} : \{\pm 1\}^n &\to \{\pm 1\}^k \cup \{¿\}, \\ \hat{c} &\mapsto \hat{u} = \delta_{\mathrm{HD}}(\hat{c}), \end{aligned}$$

die einer Realisierung $\hat{c}$ der Kanalausgabe ein decodiertes (uncodiertes) Wort $\hat{u} \in \{\pm 1\}^k$ oder ein ¿ zuordnet, heißt Hard-Decision Decodierungsabbildung.

Das technische Bauelement (oder der Algorithmus), welche(s/r) die Hard-Decision Decodierungsabbildung repräsentiert, bezeichnen wir mit Hard-Decision Decodierer, siehe Abbildung 3.3. Im

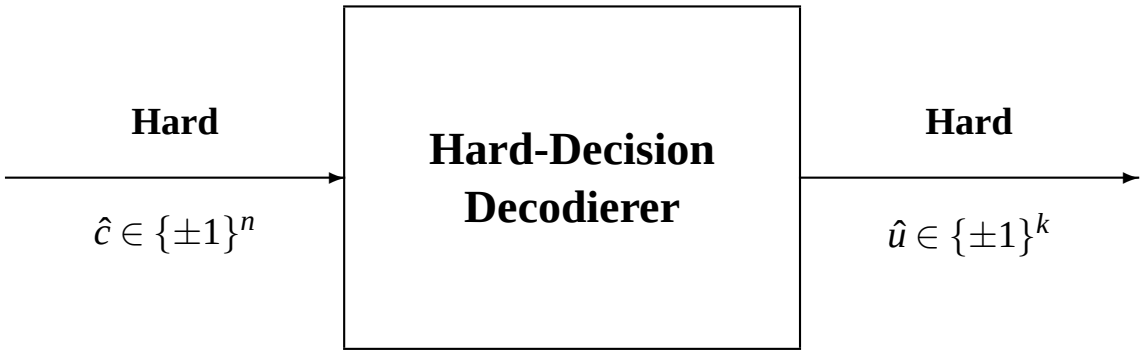

Abbildung 3.3: *Hard-Decision Decodierer*

Fall $\delta_{\mathrm{HD}}(\hat{c}) = ¿$ wird ein wissentlicher Decodierungsfehler[6] begangen, auf den ein technisches Gesamtsystem im Kontext reagieren könnte (etwa durch Verwerfen der Nachricht oder durch Neuanforderung der Nachricht bei entsprechenden Übertragungsprotokollen). Diese speziellen Kontexte sind aber nicht Gegenstand der vorliegenden Arbeit. Daher wird zur Vereinfachung der folgenden Decodierungsbeschreibung stets angenommen, daß $\delta_{\mathrm{HD}} : \{\pm 1\}^n \to \{\pm 1\}^k$.

Für die nachfolgend formulierten Aussagen werden eine Reihe von Voraussetzungen benötigt, die hier zusammengefaßt sind:

Voraussetzung 3.6 (Hard Decodierung)
Es sei $(\Omega, \mathcal{S}, P)$ ein Wahrscheinlichkeitsraum, $n, k \in \mathbb{N}$, $n > k$. Weiter sei (n, k, φ) ein binärer linearer (n,k)-Blockcode, $\mathcal{K}$ ein diskreter gedächtnisloser n-Kanal, $\hat{C} : \Omega \to \{\pm 1\}^n$ die Zufallsvariable der Kanalausgabe und $U : \Omega \to \{\pm 1\}^k$ eine von $\mathcal{K}_c$ für alle $c \in \{\pm 1\}^n$ stochastisch unabhängige Zufallsvariable mit

$$\hat{C}(\omega) = \mathcal{K}(\varphi(U(\omega)), \omega), \quad \text{für alle } \omega \in \Omega,$$

d.h., $\varphi(U)$ ist die Zufallsvariable der Kanaleingabe. Weiter gelte

$$P(\{\omega \in \Omega;\, U(\omega) = u\}) > 0 \quad \text{für alle } u \in \{\pm 1\}^k,$$
$$P\left(\{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\}\right) > 0 \quad \text{für alle } \hat{c} \in \{\pm 1\}^n.$$

Bei einem hinreichend realistischen Kanal wird man jedes Wort aus $\{\pm 1\}^n$ mit einer gewissen positiven Wahrscheinlichkeit empfangen. Außerdem ist hier angenommen, daß auch jedes uncodierte Wort mit einer positiven Wahrscheinlichkeit gesendet worden sein kann. Im weiteren Verlauf werden wir sogar die Gleichverteilung von U annehmen, welche bei Existenz eines Kryptocodierers (siehe Seite 18), der perfekte Sicherheit bietet, gegeben ist.

Definition 3.7 (Hard-Decision Decodierwortfehlerwahrscheinlichkeit)
Voraussetzung 3.6 auf Seite 36 sei erfüllt. Weiter sei $\delta_{\mathrm{HD}} : \{\pm 1\}^n \to \{\pm 1\}^k$ eine Hard-Decision Decodierungsabbildung. Dann heißt zu jedem $u \in \{\pm 1\}^k$

$$p_E(\delta_{\mathrm{HD}}, u) := 1 - \sum_{\hat{c} \in \{\pm 1\}^n;\ \delta_{\mathrm{HD}}(\hat{c}) = u} P\left(\{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\} | \{\omega \in \Omega;\, U(\omega) = u\}\right)$$

[6] vergleiche dazu auch Abschnitt 5.8 über Fehlererkennung.

die Decodier(wort)fehlerwahrscheinlichkeit der Nachricht u bei der Decodierungsabbildung δ_{HD}. Weiter heißt

$$p_E(\delta_{\mathrm{HD}}) := \sum_{u\in\{\pm1\}^k} p_E(\delta_{\mathrm{HD}},u)\cdot P\left(\{\omega\in\Omega;\, U(\omega)=u\}\right)$$

die mittlere Decodier(wort)fehlerwahrscheinlichkeit der Decodierungsabbildung δ_{HD}.

Ziel der Decodierung ist die Minimierung der mittleren Wortfehlerwahrscheinlichkeit. In nachfolgender Definition werden verschiedene Decodierungsabbildungen charakterisiert, die in der meist englischsprachigen Literatur gebräuchlich sind.

Definition 3.8 (Hard-Decision Minimalfehler-, ME-, MAP-, ML-Decodierung)
Voraussetzung 3.6 auf Seite 36 sei erfüllt.

(i) *Die Decodierung mit einer Decodierungsabbildung $\delta_{\mathrm{HD}}:\ \{\pm1\}^n\to\{\pm1\}^k$, die für jedes $\hat{c}\in\{\pm1\}^n$ die Eigenschaft*

$$\begin{aligned}&P\left(\{\omega\in\Omega;\, U(\omega)=\delta_{\mathrm{HD}}(\hat{c})\}\,|\,\left\{\omega\in\Omega;\,\hat{C}(\omega)=\hat{c}\right\}\right)\\ &\geq P\left(\{\omega\in\Omega;\, U(\omega)=u\}\,|\,\left\{\omega\in\Omega;\,\hat{C}(\omega)=\hat{c}\right\}\right)\quad \text{für alle } u\in\{\pm1\}^k,\end{aligned}$$

erfüllt, heißt MAP-Decodierung (Maximum a posteriori probability), Minimalfehler-Decodierung oder ME-Decodierung (Minimum error probability).

(ii) *Die Decodierung mit einer Decodierungsabbildung $\delta_{\mathrm{HD}}:\ \{\pm1\}^n\to\{\pm1\}^k$, die für jedes $\hat{c}\in\{\pm1\}^n$ die Eigenschaft*

$$\begin{aligned}&P\left(\left\{\omega\in\Omega;\,\hat{C}(\omega)=\hat{c}\right\}\,|\,\{\omega\in\Omega;\, U(\omega)=\delta_{\mathrm{HD}}(\hat{c})\}\right)\\ &\geq P\left(\left\{\omega\in\Omega;\,\hat{C}(\omega)=\hat{c}\right\}\,|\,\{\omega\in\Omega;\, U(\omega)=u\}\right)\quad \text{für alle } u\in\{\pm1\}^k,\end{aligned}$$

erfüllt, heißt ML-Decodierung (Maximum likelihood).

Die Begriffsbildung „Minimalfehler-Decodierung“ wird durch den folgenden Satz gerechtfertigt, der nachweist, daß bei dieser Abbildung tatsächlich die mittlere Decodierfehlerwahrscheinlichkeit minimal ist.

Satz 3.9 (Minimalfehler Hard-Decision Decodierung)
Voraussetzung 3.6 auf Seite 36 sei erfüllt. Weiter sei $\delta_{\mathrm{HD}}:\ \{\pm1\}^n\to\{\pm1\}^k$ eine Minimalfehler (MAP, ME) Hard-Decision Decodierungsabbildung. Dann gilt

$$p_E(\delta_{\mathrm{HD}})\leq p_E(\delta),$$

für jede Hard-Decision Decodierungsabbildung $\delta:\ \{\pm1\}^n\to\{\pm1\}^k$.
Ist U gleichverteilt, so gilt

$$p_E(\delta_{\mathrm{HD}}) = 1-\frac{1}{2^k}\sum_{\hat{c}\in\{\pm1\}^n} P\left(\{\omega\in\Omega;\,\hat{C}(\omega)=\hat{c}\}|\{\omega\in\Omega;\, U(\omega)=\delta_{\mathrm{HD}}(\hat{c})\}\right).$$

Beweis. Für jede Hard-Decision Decodierungsabbildung δ gilt nach dem MAP-Kriterium für jedes $\hat{c} \in \{\pm 1\}^n$ insbesondere

$$\begin{aligned} &P\left(\{\omega \in \Omega;\, U(\omega) = \delta_{\mathrm{HD}}(\hat{c})\} \,|\, \left\{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\right\}\right) \\ &\geq P\left(\{\omega \in \Omega;\, U(\omega) = \delta(\hat{c})\} \,|\, \left\{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\right\}\right). \end{aligned}$$

Damit folgt

$$\begin{aligned} p_E(\delta_{\mathrm{HD}}) &= \sum_{u \in \{\pm 1\}^k} p_E(\delta_{\mathrm{HD}}, u) \cdot P\left(\{\omega \in \Omega;\, U(\omega) = u\}\right) \\ &= \sum_{u \in \{\pm 1\}^k} \Bigg[P\left(\{\omega \in \Omega;\, U(\omega) = u\}\right) \\ &\quad - \sum_{\hat{c} \in \{\pm 1\}^n;\, \delta_{\mathrm{HD}}(\hat{c}) = u} P\left(\{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\} | \{\omega \in \Omega;\, U(\omega) = u\}\right) P\left(\{\omega \in \Omega;\, U(\omega) = u\}\right) \Bigg] \\ &= 1 - \sum_{\hat{c} \in \{\pm 1\}^n} \sum_{u \in \{\pm 1\}^k;\, \delta_{\mathrm{HD}}(\hat{c}) = u} P\left(\{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\} | \{\omega \in \Omega;\, U(\omega) = u\}\right) \\ &\qquad \cdot P\left(\{\omega \in \Omega;\, U(\omega) = u\}\right) \\ &= 1 - \sum_{\hat{c} \in \{\pm 1\}^n} P\left(\{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\} | \{\omega \in \Omega;\, U(\omega) = \delta_{\mathrm{HD}}(\hat{c})\}\right) P\left(\{\omega \in \Omega;\, U(\omega) = \delta_{\mathrm{HD}}(\hat{c})\}\right) \\ &= 1 - \sum_{\hat{c} \in \{\pm 1\}^n} P\left(\{\omega \in \Omega;\, U(\omega) = \delta_{\mathrm{HD}}(\hat{c})\} | \{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\}\right) P\left(\left\{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\right\}\right) \\ &\leq 1 - \sum_{\hat{c} \in \{\pm 1\}^n} P\left(\{\omega \in \Omega;\, U(\omega) = \delta(\hat{c})\} | \{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\}\right) P\left(\left\{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\right\}\right) \\ &= p_E(\delta) \end{aligned}$$

Ist U gleichverteilt, also $P\left(\{\omega \in \Omega;\, U(\omega) = \delta_{\mathrm{HD}}(\hat{c})\}\right) = \frac{1}{2^k}$ für alle $\hat{c} \in \{\pm 1\}^n$, so folgt aus dem eben Gezeigten unmittelbar

$$p_E(\delta_{\mathrm{HD}}) = 1 - \frac{1}{2^k} \sum_{\hat{c} \in \{\pm 1\}^n} P\left(\{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\} | \{\omega \in \Omega;\, U(\omega) = \delta_{\mathrm{HD}}(\hat{c})\}\right).$$

□

Unter geeigneten Voraussetzungen fallen die Begriffe Minimalfehler-Decodierung und Maximum-Likelihood-Decodierung zusammen.

Lemma 3.10 (Minimalfehler gleich ML)
Voraussetzung 3.6 auf Seite 36 sei erfüllt. Weiter sei U gleichverteilt und $\delta_{\mathrm{HD}} : \{\pm 1\}^n \to \{\pm 1\}^k$ sei eine Hard-Decision Decodierungsabbildung.
Es gilt: δ_{HD} ist genau dann eine Minimalfehler (MAP, ME) Hard-Decision Decodierungsabbildung, wenn δ_{HD} eine ML Hard-Decision Decodierungsabbildung ist.

Beweis. Da U gleichverteilt ist, gilt also

$$P\left(\{\omega \in \Omega;\, U(\omega) = u\}\right) = \frac{1}{2^k}, \quad \text{für alle } u \in \{\pm 1\}^k.$$

Damit gilt für jedes $\hat{c} \in \{\pm 1\}^n$:

$$\begin{aligned}
&P\left(\left\{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\right\} \mid \{\omega \in \Omega;\, U(\omega) = \delta_{\mathrm{HD}}(\hat{c})\}\right) \\
&= \frac{P\left(\left\{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\right\}\right)}{P\left(\{\omega \in \Omega;\, U(\omega) = \delta_{\mathrm{HD}}(\hat{c})\}\right)} P\left(\{\omega \in \Omega;\, U(\omega) = \delta_{\mathrm{HD}}(\hat{c})\} \mid \left\{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\right\}\right) \\
&= 2^k \cdot const \cdot P\left(\{\omega \in \Omega;\, U(\omega) = \delta_{\mathrm{HD}}(\hat{c})\} \mid \left\{\omega \in \Omega;\, \hat{C}(\omega) = \hat{c}\right\}\right).
\end{aligned}$$

Somit sind die Maximalitätskriterien äquivalent.

□

3.4 Soft-Decision Decodierung

Im Gegensatz zur Hard-Decision Decodierung ist die Entscheidungsgrundlage des Decodierers ein „softer“ Vektor $y \in \mathbb{R}^n$. Analog zur Hard-Decision läßt sich damit eine Decodierungsabbildung definieren.

Definition 3.11 (Soft-Decision Decodierungsabbildung)
Es sei $(\Omega, \mathcal{S}, P)$ ein Wahrscheinlichkeitsraum, $n, k \in \mathbb{N}$, $n > k$. Weiter sei (n, k, φ) ein binärer linearer (n,k)-Blockcode und $\mathcal{K}$ ein stetiger gedächtnisloser n-Kanal. Eine $\mathcal{B}^n$-$\mathcal{P}(\{\pm 1\}^k)$-meßbare Abbildung

$$\begin{aligned}
\delta_{\mathrm{SD}} :\ & \mathbb{R}^n \to \{\pm 1\}^k, \\
& y \mapsto \hat{u} = \delta_{\mathrm{SD}}(y),
\end{aligned}$$

die einer Realisierung y der Kanalausgabe ein decodiertes (uncodiertes) Wort $\hat{u} \in \{\pm 1\}^k$ zuordnet, heißt Soft-Decision Decodierungsabbildung.

⌟

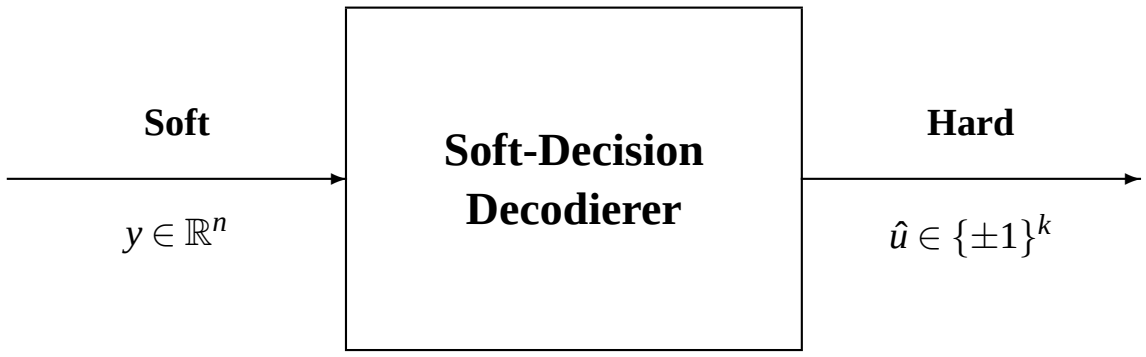

Abbildung 3.4: *Soft-Decision Decodierer*

Analog zur Hard-Decision Decodierung bezeichnen wir das technische Bauelement (oder den Algorithmus), welche(s/r) die Soft-Decision Decodierungsabbildung repräsentiert, mit Soft-Decision Decodierer, siehe Abbildung 3.4.

Im folgenden wollen wir analoge Begriffe zur Hard-Decision Decodierung einführen, die dann ebenfalls die Charakterisierung von fehlerminimalen Decodierungsmethoden erlauben.

Voraussetzung 3.12 (Soft Decodierung)
Es sei $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum, $n,k\in\mathbb{N}$, $n>k$. Weiter sei (n,k,φ) ein binärer linearer (n,k)-Blockcode, $\mathcal{K}$ ein stetiger gedächtnisloser n-Kanal, $Y:\ \Omega\to\mathbb{R}^n$ die Zufallsvariable der Kanalausgabe und $U:\ \Omega\to\{\pm1\}^k$ eine von $\mathcal{K}_c$ für alle $c\in\{\pm1\}^n$ stochastisch unabhängige Zufallsvariable mit

$$Y(\omega)=\mathcal{K}(\varphi(U(\omega)),\omega),\quad \text{für alle } \omega\in\Omega,$$

d.h., $\varphi(U)$ ist die Zufallsvariable der Kanaleingabe. Weiter gelte

$$P(\{\omega\in\Omega;\ U(\omega)=u\})>0,\quad \text{für alle } u\in\{\pm1\}^k.$$

Im Gegensatz zu Voraussetzung 3.6 auf Seite 36 gilt bei einem stetigen n-Kanal

$$P(\{\omega\in\Omega;\ Y(\omega)=y\})=0,\quad \text{für alle } y\in\mathbb{R}^n$$

und somit können wir die Konstruktion mit bedingten Wahrscheinlichkeiten wie in Abschnitt 3.3 nicht unmittelbar übernehmen. Statt dessen werden die Dichten des stetigen n-Kanals die Rolle der (positiven) Wahrscheinlichkeiten übernehmen.

Da

$$P(\{\omega\in\Omega;\ U(\omega)=u\})>0,\quad \text{für alle } u\in\{\pm1\}^k,$$

können wir aber die Decodierwortfehlerwahrscheinlichkeit, die uns als Bewertungskriterium für Decodierungsabbildungen dient, analog definieren.

Definition 3.13 (Soft-Decision Decodierwortfehlerwahrscheinlichkeit)
Voraussetzung 3.12 auf Seite 40 sei erfüllt. Weiter sei $\delta_{\mathrm{SD}}:\ \mathbb{R}^n\to\{\pm1\}^k$ eine Soft-Decision Decodierungsabbildung. Dann heißt zu jedem $u\in\{\pm1\}^k$

$$p_E(\delta_{\mathrm{SD}},u):=1-P\left(\left\{\omega\in\Omega;\ Y(\omega)\in\delta_{\mathrm{SD}}^{-1}(\{u\})\right\}\,|\,\{\omega\in\Omega;\ U(\omega)=u\}\right)$$

die Decodier(wort)fehlerwahrscheinlichkeit der Nachricht u bei der Decodierungsabbildung δ_{SD}, wobei $\delta_{\mathrm{SD}}^{-1}(\{u\})=\{x\in\mathbb{R}^n;\ \delta_{\mathrm{SD}}(x)=u\}$.
Weiter heißt

$$p_E(\delta_{\mathrm{SD}}):=\sum_{u\in\{\pm1\}^k}p_E(\delta_{\mathrm{SD}},u)\cdot P(\{\omega\in\Omega;\ U(\omega)=u\})$$

die mittlere Decodier(wort)fehlerwahrscheinlichkeit der Decodierungsabbildung δ_{SD}.

Da δ_{SD} als $\mathcal{B}^n$-$\mathcal{P}(\{\pm1\}^k)$-meßbar vorausgesetzt wurde, ist $\delta_{\mathrm{SD}}^{-1}(\{u\})\in\mathcal{B}^k$ und somit ist die Betrachtung der bedingten Wahrscheinlichkeiten in der Definition zulässig.

Der folgende wichtige Satz formuliert die Darstellung von bedingten Wahrscheinlichkeiten und Fehlerwahrscheinlichkeiten in Abhängigkeit von den Dichtefunktionen des stetigen n-Kanals.

Satz 3.14 (Dichte-Darstellung der Decodierwortfehlerwahrscheinlichkeit)
Voraussetzung 3.12 auf Seite 40 sei erfüllt.

(i) *Für alle $A\in\mathcal{B}^n$ gilt*

$$P(\{\omega\in\Omega;\ Y(\omega)\in A\}\,|\,\{\omega\in\Omega;\ U(\omega)=u\})=\int_A f_{\varphi(u)}d\lambda^n.$$

(ii) *Für alle* $A \in \mathcal{B}^n$ *mit* $P(\{\omega \in \Omega;\ Y(\omega) \in A\}) > 0$ *und* U *gleichverteilt gilt*

$$P(\{\omega \in \Omega;\ U(\omega) = u\} \mid \{\omega \in \Omega;\ Y(\omega) \in A\}) = \frac{\int\limits_A f_{\varphi(u)} d\lambda^n}{\sum\limits_{\tilde{u} \in \{\pm 1\}^k} \int\limits_A f_{\varphi(\tilde{u})} d\lambda^n}.$$

(iii) *Sei* $\delta_{\mathrm{SD}} : \mathbb{R}^n \to \{\pm 1\}^k$ *eine Soft-Decision Decodierungsabbildung. Dann gilt für alle* $u \in \{\pm 1\}^k$

$$p_E(\delta_{\mathrm{SD}}, u) = 1 - \int\limits_{\delta_{\mathrm{SD}}^{-1}(\{u\})} f_{\varphi(u)} d\lambda^n.$$

(iv) *Sei* $\delta_{\mathrm{SD}} : \mathbb{R}^n \to \{\pm 1\}^k$ *eine Soft-Decision Decodierungsabbildung und* U *gleichverteilt. Dann gilt*

$$p_E(\delta_{\mathrm{SD}}) = 1 - \frac{1}{2^k} \int\limits_{\mathbb{R}^n} f_{\varphi(\delta_{\mathrm{SD}}(x))}(x) d\lambda^n(x).$$

Beweis. Ad (i): Sei $A \in \mathcal{B}^n$. Dann gilt

$$\begin{aligned}
&P(\{\omega \in \Omega;\ Y(\omega) \in A\} \mid \{\omega \in \Omega;\ U(\omega) = u\}) \\
&= \frac{P(\{\omega \in \Omega;\ Y(\omega) \in A\} \cap \{\omega \in \Omega;\ U(\omega) = u\})}{P(\{\omega \in \Omega;\ U(\omega) = u\})} \\
&= \frac{P(\{\omega \in \Omega;\ \mathcal{K}(\varphi(U(\omega)), \omega) \in A\} \cap \{\omega \in \Omega;\ U(\omega) = u\})}{P(\{\omega \in \Omega;\ U(\omega) = u\})} \\
&= \frac{P(\{\omega \in \Omega;\ \mathcal{K}(\varphi(u), \omega) \in A\} \cap \{\omega \in \Omega;\ U(\omega) = u\})}{P(\{\omega \in \Omega;\ U(\omega) = u\})} \\
&= \frac{P\left(\left\{\omega \in \Omega;\ \mathcal{K}_{\varphi(u)}(\omega) \in A\right\} \cap \{\omega \in \Omega;\ U(\omega) = u\}\right)}{P(\{\omega \in \Omega;\ U(\omega) = u\})} \\
&= P\left(\left\{\omega \in \Omega;\ \mathcal{K}_{\varphi(u)}(\omega) \in A\right\} \mid \{\omega \in \Omega;\ U(\omega) = u\}\right) \\
&= P\left(\left\{\omega \in \Omega;\ \mathcal{K}_{\varphi(u)}(\omega) \in A\right\}\right) = \int\limits_A f_{\varphi(u)} d\lambda^n,
\end{aligned}$$

da $\mathcal{K}_{\varphi(u)}$ und U nach Voraussetzung 3.12 auf Seite 40 stochastisch unabhängig sind.

Ad (ii): Sei $A \in \mathcal{B}^n$ mit $P(\{\omega \in \Omega;\ Y(\omega) \in A\}) > 0$ und sei U gleichverteilt. Dann gilt

$$\begin{aligned}
&P(\{\omega \in \Omega;\ U(\omega) = u\} \mid \{\omega \in \Omega;\ Y(\omega) \in A\}) \\
&= \frac{P(\{\omega \in \Omega;\ U(\omega) = u\} \cap \{\omega \in \Omega;\ Y(\omega) \in A\})}{P(\{\omega \in \Omega;\ Y(\omega) \in A\})} \\
&= \frac{P(\{\omega \in \Omega;\ U(\omega) = u\}) P(\{\omega \in \Omega;\ Y(\omega) \in A\} \mid \{\omega \in \Omega;\ U(\omega) = u\})}{\sum\limits_{\tilde{u} \in \{\pm 1\}^k} P(\{\omega \in \Omega;\ U(\omega) = \tilde{u}\}) P(\{\omega \in \Omega;\ Y(\omega) \in A\} \mid \{\omega \in \Omega;\ U(\omega) = \tilde{u}\})} \\
&= \frac{P(\{\omega \in \Omega;\ Y(\omega) \in A\} \mid \{\omega \in \Omega;\ U(\omega) = u\})}{\sum\limits_{\tilde{u} \in \{\pm 1\}^k} P(\{\omega \in \Omega;\ Y(\omega) \in A\} \mid \{\omega \in \Omega;\ U(\omega) = \tilde{u}\})} \\
&= \frac{\int\limits_A f_{\varphi(u)} d\lambda^n}{\sum\limits_{\tilde{u} \in \{\pm 1\}^k} \int\limits_A f_{\varphi(\tilde{u})} d\lambda^n}.
\end{aligned}$$

Ad (iii): Die Aussage folgt unmittelbar aus (i).

Ad (iv): Sei $\delta_{SD}: \mathbb{R}^n \to \{\pm 1\}^k$ eine Soft-Decision Decodierungsabbildung und U gleichverteilt. Dann gilt

$$\begin{aligned} p_E(\delta_{SD}) &= \sum_{u\in\{\pm1\}^k} p_E(\delta_{SD},u)\cdot P(\{\omega\in\Omega;\, U(\omega)=u\}) \\ &= 1-\frac{1}{2^k}\sum_{u\in\{\pm1\}^k}\int\limits_{\delta_{SD}^{-1}(\{u\})} f_{\varphi(u)}d\lambda^n \\ &= 1-\frac{1}{2^k}\sum_{u\in\{\pm1\}^k}\int\limits_{\mathbb{R}^n} I_{\delta_{SD}^{-1}(\{u\})} f_{\varphi(u)}d\lambda^n \\ &= 1-\frac{1}{2^k}\int\limits_{\mathbb{R}^n}\sum_{u\in\{\pm1\}^k} I_{\delta_{SD}^{-1}(\{u\})}(x) f_{\varphi(u)}(x)d\lambda^n(x) \\ &= 1-\frac{1}{2^k}\int\limits_{\mathbb{R}^n} f_{\varphi(\delta_{SD}(x))}(x)d\lambda^n(x). \end{aligned}$$

□

Mit Hilfe der Dichtedarstellung können nun Decodierungsabbildungen, die minimale Decodierungsfehler produzieren, charakterisiert werden.

Definition 3.15 (Soft-Decision Minimalfehler-Decodierung)
Voraussetzung 3.12 auf Seite 40 sei erfüllt und U sei gleichverteilt. Die Decodierung mit einer Decodierungsabbildung $\delta_{SD}: \mathbb{R}^n \to \{\pm1\}^k$, die für jedes $y\in\mathbb{R}^n$ die Eigenschaft

$$f_{\varphi(\delta_{SD}(y))}(y) \geq f_{\varphi(u)}(y), \quad \text{für alle } u\in\{\pm1\}^k,$$

erfüllt, heißt Minimalfehler-Decodierung oder ME-Decodierung (Minimum error probability). ⌟

Der nachfolgende Satz verifiziert die Begriffsbildung der Definition.

Satz 3.16 (Minimalfehler Soft-Decision Decodierung)
Voraussetzung 3.12 auf Seite 40 sei erfüllt und U sei gleichverteilt.
Weiter sei $\delta_{SD}: \mathbb{R}^n \to \{\pm1\}^k$ eine Minimalfehler Soft-Decision Decodierungsabbildung. Dann gilt

$$p_E(\delta_{SD}) \leq p_E(\delta),$$

für jede Soft-Decision Decodierungsabbildung $\delta: \mathbb{R}^n \to \{\pm1\}^k$. ⌟

Beweis. Mit den Vorarbeiten von Satz 3.14 folgt der Beweis sehr schnell. Nach dem ME Kriterium von Definition 3.15 gilt insbesondere für jede Soft-Decision Decodierungsabbildung $\delta: \mathbb{R}^n \to \{\pm1\}^k$, daß für alle $x\in\mathbb{R}^n$

$$f_{\varphi(\delta_{SD}(x))}(x) \geq f_{\varphi(\delta(x))}(x).$$

Und damit folgt

$$\begin{aligned} p_E(\delta_{\mathrm{SD}}) &= 1 - \frac{1}{2^k} \int\limits_{\mathbb{R}^n} f_{\varphi(\delta_{\mathrm{SD}}(x))}(x) d\lambda^n(x) \\ &\leq 1 - \frac{1}{2^k} \int\limits_{\mathbb{R}^n} f_{\varphi(\delta(x))}(x) d\lambda^n(x) \\ &= p_E(\delta). \end{aligned}$$

□

Die Frage nach der Existenz von Minimalfehler Soft-Decision Decodierungsabbildungen ist noch offen, das heißt, es ist zu zeigen, daß eine $\mathcal{B}^n$-$\mathcal{P}(\{\pm 1\}^k)$-meßbare Abbildung existiert, die das ME Kriterium von Definition 3.15 erfüllt. Den Nachweis führt der folgende Satz, der zusätzlich die Stetigkeit der Kanaldichten voraussetzt. Diese Eigenschaft wird auch bei den später folgenden Betrachtungen benötigt und ist im Hauptanwendungsfall der AWGN-Kanäle gegeben.

Satz 3.17 (Existenz einer Minimalfehler Soft-Decision Decodierung)
Voraussetzung 3.12 auf Seite 40 sei erfüllt, U sei gleichverteilt und f_c sei stetig für alle $c \in \{\pm 1\}^n$. Dann existiert eine Minimalfehler Soft-Decision Decodierungsabbildung $\delta_{\mathrm{SD}} : \mathbb{R}^n \to \{\pm 1\}^k$. ⌟

Beweis. Der Beweis wird in konstruktiver Weise geführt. Die Abbildung

$$h_{u,\tilde{u}} : \mathbb{R}^n \to \mathbb{R},\ y \mapsto h_{u,\tilde{u}}(y) = f_{\varphi(u)}(y) - f_{\varphi(\tilde{u})}(y),$$

ist stetig für alle $u, \tilde{u} \in \{\pm 1\}^k$, da f_c sei stetig für alle $c \in \{\pm 1\}^n$ ist. Also ist

$$C_{u,\tilde{u}} := \{y \in \mathbb{R}^n;\ h_{u,\tilde{u}}(y) \geq 0\}$$

abgeschlossen und somit $C_{u,\tilde{u}} \in \mathcal{B}^n$ für alle $u, \tilde{u} \in \{\pm 1\}^k$.
Für alle $u \in \{\pm 1\}^k$ gilt dann

$$D_u := \left\{ y \in \mathbb{R}^n;\ f_{\varphi(u)}(y) \geq f_{\varphi(\tilde{u})}(y) \text{ für alle } \tilde{u} \in \{\pm 1\}^k \right\} = \bigcap_{\tilde{u} \in \{\pm 1\}^k} C_{u,\tilde{u}} \in \mathcal{B}^n,$$

und $\bigcup\limits_{u \in \{\pm 1\}^k} D_u = \mathbb{R}^n$.
Betrachte

$$\{\pm 1\}^k = \left\{ u^1, u^2, \ldots, u^{2^k} \right\}$$

und definiere

$$\begin{aligned} \hat{D}_{u^1} &:= D_{u^1} \in \mathcal{B}^n, \\ \hat{D}_{u^{i+1}} &:= \underbrace{D_{u^{i+1}}}_{\in \mathcal{B}^n} \setminus \underbrace{\left(\bigcup_{j=1}^{i} \hat{D}_{u^j} \right)}_{\in \mathcal{B}^n} \in \mathcal{B}^n, \quad \text{für } i = 1, \ldots, 2^k - 1. \end{aligned}$$

Dann ist $\mathbb{R}^n = \sum\limits_{u \in \{\pm 1\}^k} \hat{D}_u$.

Definiere schließlich

$$\begin{aligned}\delta_{\mathrm{SD}}:\ &\mathbb{R}^n \to \{\pm 1\}^k,\\ &y \mapsto u \quad \text{mit } u \text{ derart, daß } y \in \hat{D}_u.\end{aligned}$$

Um die $\mathcal{B}^n$-$\mathcal{P}(\{\pm 1\}^k)$-Meßbarkeit von δ_{SD} nachzuweisen, genügt es zu zeigen, daß $\delta_{\mathrm{SD}}^{-1}(\{u\}) \in \mathcal{B}^n$ für alle $u \in \{\pm 1\}^k$. Da nach Konstruktion

$$\delta_{\mathrm{SD}}^{-1}(\{u\}) = \hat{D}_u \in \mathcal{B}^n, \quad \text{für alle } u \in \{\pm 1\}^k,$$

ist δ_{SD} $\mathcal{B}^n$-$\mathcal{P}(\{\pm 1\}^k)$-meßbar und also eine Soft-Decision Decodierungsabbildung.
Für ein beliebiges $y \in \mathbb{R}^n$ betrachte $u := \delta_{\mathrm{SD}}(y)$, also

$$y \in \hat{D}_u \subseteq D_u = \left\{ x \in \mathbb{R}^n;\ f_{\varphi(u)}(x) \geq f_{\varphi(\tilde{u})}(x) \text{ für alle } \tilde{u} \in \{\pm 1\}^k \right\}.$$

Somit gilt

$$f_{\varphi(\delta_{\mathrm{SD}}(y))}(y) = f_{\varphi(u)}(y) \geq f_{\varphi(\tilde{u})}(y) \text{ für alle } \tilde{u} \in \{\pm 1\}^k,$$

das heißt, δ_{SD} ist eine Minimalfehler Soft-Decision Decodierungsabbildung.

□

Unter geeigneten Voraussetzungen läßt sich der Begriff der bedingten Wahrscheinlichkeit erweitern, so daß wir a posteriori Wahrscheinlichkeiten als Werkzeug verwenden können. Die Berechtigung zu einer solchen Erweiterung gibt der folgende Satz.

Satz 3.18 (Konvergenz von bedingten Wahrscheinlichkeiten)
Voraussetzung 3.12 auf Seite 40 sei erfüllt und U sei gleichverteilt. Weiter sei f_c für alle $c \in \{\pm 1\}^n$ stetig. Für jedes $u \in \{\pm 1\}^k$ und jedes $y \in \mathbb{R}^n$ mit

$$f_{\varphi(u)}(y) > 0,$$

und für alle Folgen $\{A_m\}_{m\in\mathbb{N}}$ mit $A_m \in \mathcal{B}^n$, $\lambda^n(A_m) > 0$, A_m kompakt, $A_m \supseteq A_{m+1}$, $m \in \mathbb{N}$, und $\bigcap_{m\in\mathbb{N}} A_m = \{y\}$ gilt

$$\lim_{m\to\infty} P\left(\{\omega \in \Omega;\ U(\omega) = u\} \,|\, \{\omega \in \Omega;\ Y(\omega) \in A_m\}\right) = \frac{f_{\varphi(u)}(y)}{\sum_{\tilde{u}\in\{\pm 1\}^k} f_{\varphi(\tilde{u})}(y)}.$$

⌟

Beweis. Definiere

$$\begin{aligned} h(x) &:= f_{\varphi(u)}(x), && \text{für alle } x \in \mathbb{R}^n,\\ g(x) &:= \sum_{\tilde{u}\in\{\pm 1\}^k} f_{\varphi(\tilde{u})}(x), && \text{für alle } x \in \mathbb{R}^n. \end{aligned}$$

Nach Voraussetzung sind $h, g:\ \mathbb{R} \to \mathbb{R}_0^+$ stetige Funktionen.

Sei $\varepsilon > 0$ mit $\varepsilon < \frac{1}{2}h(y) = \frac{1}{2} f_{\varphi(u)}(y)$ beliebig gegeben. Die Mengen

$$B_m^\varepsilon := \{x \in A_m;\ |h(x) - h(y)| \geq \varepsilon\} \cup \{x \in A_m;\ |g(x) - g(y)| \geq \varepsilon\}, \quad \text{für alle } m \in \mathbb{N},$$

sind kompakt, da die Mengen A_m kompakt sind und h, g stetige Funktionen sind, und es gilt

$$B_m^\varepsilon \supseteq B_{m+1}^\varepsilon, \quad \text{für alle } m \in \mathbb{N}.$$

Es gilt $y \notin B_m^\varepsilon$ und folglich

$$\bigcap_{m\in\mathbb{N}} B_m^\varepsilon \subseteq \bigcap_{m\in\mathbb{N}} A_m \setminus \{y\} = \emptyset.$$

Da die Mengen B_m^ε kompakt sind, gibt es ein $M_\varepsilon \in \mathbb{N}$, so daß

$$B_m^\varepsilon = \emptyset, \quad \text{für alle } m \geq M_\varepsilon,$$

und somit

$$|h(x)-h(y)| < \varepsilon,\ |g(x)-g(y)| < \varepsilon, \quad \text{für alle } x \in A_m,\ m \geq M_\varepsilon.$$

Insbesondere gilt somit

$$\int_{A_m} f_{\varphi(u)} d\lambda^n = \int_{A_m} h(x) d\lambda^n(x) \geq \int_{A_m} (h(y)-\varepsilon) d\lambda^n(x) \geq \frac{1}{2}\lambda^n(A_m)h(y) > 0, \quad \text{für alle } m \geq M_\varepsilon$$

und weiter gilt mit Satz 3.14

$$\begin{aligned}
&P(\{\omega \in \Omega;\ Y(\omega) \in A_m\}) \\
&= \sum_{\tilde{u}\in\{\pm1\}^k} P(\{\omega \in \Omega;\ U(\omega) = \tilde{u}\}) P(\{\omega \in \Omega;\ Y(\omega) \in A_m\} \,|\, \{\omega \in \Omega;\ U(\omega) = \tilde{u}\}) \\
&= \frac{1}{2^k} \sum_{\tilde{u}\in\{\pm1\}^k} \int_{A_m} f_{\varphi(\tilde{u})} d\lambda^n \\
&\geq \frac{1}{2^k} \int_{A_m} f_{\varphi(u)} d\lambda^n > 0, \quad \text{für alle } m \geq M_\varepsilon.
\end{aligned}$$

Damit sind die Voraussetzungen von Aussage (ii) in Satz 3.14 gegeben und es gilt

$$\begin{aligned}
w(u,A_m) :=& P(\{\omega \in \Omega;\ U(\omega) = u\} \,|\, \{\omega \in \Omega;\ Y(\omega) \in A_m\}) \\
&= \frac{\int_{A_m} f_{\varphi(u)} d\lambda^n}{\sum_{\tilde{u}\in\{\pm1\}^k} \int_{A_m} f_{\varphi(\tilde{u})} d\lambda^n} \\
&= \frac{\int_{A_m} f_{\varphi(u)} d\lambda^n}{\int_{A_m} \sum_{\tilde{u}\in\{\pm1\}^k} f_{\varphi(\tilde{u})} d\lambda^n} = \frac{\int_{A_m} h(x) d\lambda^n(x)}{\int_{A_m} g(x) d\lambda^n(x)}, \quad \text{für alle } m \geq M_\varepsilon.
\end{aligned}$$

Nun folgt

$$\begin{aligned}
\frac{-\varepsilon g(y)-\varepsilon h(y)}{(g(y)+\varepsilon)g(y)} &= \frac{\lambda^n(A_m)(h(y)-\varepsilon)}{\lambda^n(A_m)(g(y)+\varepsilon)} - \frac{h(y)}{g(y)} \\
&= \frac{\int\limits_{A_m}(h(y)-\varepsilon)d\lambda^n(x)}{\int\limits_{A_m}(g(y)+\varepsilon)d\lambda^n(x)} - \frac{h(y)}{g(y)} \leq \frac{\int\limits_{A_m} h(x)d\lambda^n(x)}{\int\limits_{A_m} g(x)d\lambda^n(x)} - \frac{h(y)}{g(y)} \\
&= w(u,A_m) - \frac{h(y)}{g(y)} \\
&\leq \frac{\int\limits_{A_m}(h(y)+\varepsilon)d\lambda^n(x)}{\int\limits_{A_m}(g(y)-\varepsilon)d\lambda^n(x)} - \frac{h(y)}{g(y)} = \frac{\lambda^n(A_m)(h(y)+\varepsilon)}{\lambda^n(A_m)(g(y)-\varepsilon)} - \frac{h(y)}{g(y)} \\
&= \frac{\varepsilon g(y)+\varepsilon h(y)}{(g(y)-\varepsilon)g(y)}, \quad \text{für alle } m \geq M_\varepsilon,
\end{aligned}$$

also

$$\left| w(u,A_m) - \frac{h(y)}{g(y)} \right| \leq \varepsilon \frac{g(y)+h(y)}{(g(y)-\varepsilon)g(y)} \leq \varepsilon \cdot 2\frac{g(y)+h(y)}{g^2(y)} \longrightarrow 0 \quad \text{für } \varepsilon \downarrow 0 \text{ und } m \geq M_\varepsilon.$$

Damit ist die Satzaussage nachgewiesen.

□

Die Aussage von Satz 3.18 berechtigt zur Aufstellung der nachfolgenden Erweiterung des Begriffs der bedingten Wahrscheinlichkeit unter den genannten Voraussetzungen.

Definition 3.19 (A posteriori Wahrscheinlichkeiten bei stetigen Kanälen)
Voraussetzung 3.12 auf Seite 40 sei erfüllt und U sei gleichverteilt. Weiter sei f_c für alle $c \in \{\pm 1\}^n$ stetig. Für jedes $u \in \{\pm 1\}^k$ und jedes $y \in \mathbb{R}^n$ mit

$$f_{\varphi(u)}(y) > 0,$$

definiere

$$w(u,y) := P\left(\{\omega \in \Omega;\ U(\omega) = u\} \,|\, \{\omega \in \Omega;\ Y(\omega) = y\}\right) := \frac{f_{\varphi(u)}(y)}{\sum\limits_{\tilde{u} \in \{\pm 1\}^k} f_{\varphi(\tilde{u})}(y)}$$

als die Wahrscheinlichkeit von $\{\omega \in \Omega;\ U(\omega) = u\}$ unter der Bedingung $\{\omega \in \Omega;\ Y(\omega) = y\}$ (bedingte Wahrscheinlichkeit, A Posteriori Wahrscheinlichkeit).

⌟

Wie bei der Hard-Decision Decodierung sind die Kriterien „Minimum Error Probability“ und „Maximum A Posteriori Probability“ unter geeigneten Voraussetzungen äquivalent. Diese Aussage wird in folgender Definition mit anschließendem Lemma festgehalten.

Definition 3.20 (Soft-Decision MAP-Decodierung)
Voraussetzung 3.12 auf Seite 40 sei erfüllt, U sei gleichverteilt und f_c sei für alle $c \in \{\pm 1\}^n$ stetig und positiv. Die Decodierung mit einer Decodierungsabbildung $\delta_{\mathrm{SD}} : \mathbb{R}^n \to \{\pm 1\}^k$, die für jedes $y \in \mathbb{R}^n$ die Eigenschaft

$$w(\delta_{\mathrm{SD}}(y),y) \geq w(u,y), \quad \text{für alle } u \in \{\pm 1\}^k,$$

erfüllt, heißt MAP-Decodierung (Maximum A Posteriori Probability).

⌟

Lemma 3.21 (Soft-Decision: Minimalfehler gleich MAP)
Voraussetzung 3.12 auf Seite 40 sei erfüllt, U sei gleichverteilt und f_c sei für alle $c \in \{\pm 1\}^n$ stetig und positiv. Weiter sei $\delta_{SD} : \mathbb{R}^n \to \{\pm 1\}^k$ eine Soft-Decision Decodierungsabbildung.
Es gilt: δ_{SD} ist genau dann eine Minimalfehler Soft-Decision Decodierungsabbildung, wenn δ_{SD} eine MAP Soft-Decision Decodierungsabbildung ist.

Beweis. Mit Definition 3.15, Definition 3.19 und Definition 3.20 folgt die Aussage sofort aus der Tatsache, daß der Nenner von

$$w(u,y) = \frac{f_{\varphi(u)}(y)}{\sum\limits_{\tilde{u}\in\{\pm 1\}^k} f_{\varphi(\tilde{u})}(y)}$$

unabhängig von der Wahl eines u ist.

□

Es läßt sich noch ein „handlicheres“ Kriterium zur Soft-Decision Decodierung aufstellen, welches unter geeigneten Voraussetzungen wiederum eine fehlerminimale Decodierung darstellt.

Definition 3.22 (Soft-Decision Minimaldistanz-Decodierung)
Voraussetzung 3.12 auf Seite 40 sei erfüllt und $\|.\|$ sei eine Norm auf $\mathbb{R}^n$. Die Decodierung mit einer Decodierungsabbildung $\delta_{SD} : \mathbb{R}^n \to \{\pm 1\}^k$, die für jedes $y \in \mathbb{R}^n$ die Eigenschaft

$$\|\varphi(\delta_{SD}(y)) - y\| \leq \|\varphi(u) - y\|, \quad \text{für alle } u \in \{\pm 1\}^k,$$

erfüllt, heißt Minimaldistanz-Decodierung bezüglich der Norm $\|.\|$.

Für jedes $y \in \mathbb{R}^n$ liegt also ein kombinatorisches Optimierungsproblem

$$\min_{u\in\{\pm 1\}^k} \|\varphi(u) - y\|,$$

vor, dessen globaler Minimierer als $\delta_{SD}(y)$ definiert wird.

Bei der wichtigsten Sorte von stetigen n-Kanälen, den AWGN-Kanälen, sind Minimaldistanz-Decodierung und Minimalfehler-Decodierung identisch.

Satz 3.23 (Soft-Decision bei einem AWGN-Kanal)
Voraussetzung 3.12 auf Seite 40 sei erfüllt, U sei gleichverteilt und $\mathcal{K}$ sei ein AWGN-Kanal mit bitweiser Varianz $\sigma^2 > 0$ der Kanalstörung.

(i) *Es gilt*

$$f_c(x) = \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp\left(-\frac{\|c-x\|_2^2}{2\sigma^2}\right), \quad \text{für alle } x \in \mathbb{R}^n.$$

(ii) *Es gilt für alle* $u \in \{\pm 1\}^k$ *und alle* $y \in \mathbb{R}^n$*:*

$$\begin{aligned} w(u,y) &= P\left(\{\omega \in \Omega;\, U(\omega) = u\} \,|\, \{\omega \in \Omega;\, Y(\omega) = y\}\right) \\ &= \frac{\exp\left(-\frac{\|\varphi(u)-y\|_2^2}{2\sigma^2}\right)}{\sum\limits_{\tilde{u}\in\{\pm1\}^k} \exp\left(-\frac{\|\varphi(\tilde{u})-y\|_2^2}{2\sigma^2}\right)} \\ &= \frac{1}{\sum\limits_{\tilde{u}\in\{\pm1\}^k} \exp\left(\frac{y^\top(\varphi(\tilde{u})-\varphi(u))}{\sigma^2}\right)}. \end{aligned}$$

(iii) *Weiter sei* $\delta_{\mathrm{SD}} : \mathbb{R}^n \to \{\pm 1\}^k$ *eine Soft-Decision Decodierungsabbildung.*
Es gilt: δ_{SD} *ist genau dann eine Minimalfehler Soft-Decision Decodierungsabbildung, wenn* δ_{SD} *eine Minimaldistanz Soft-Decision Decodierungsabbildung bezüglich* $\|.\|_2$ *(euklidische Norm) ist.*

□

Beweis. Ad (i): Mit Definition 3.3 auf Seite 33 gilt für eine $\mathcal{N}(0,\sigma^2 I_n)$ normalverteilte Zufallsvariable $Z : \Omega \to \mathbb{R}^n$, daß $\mathcal{K}_c(\omega) = c + Z(\omega)$. Folglich ist $\mathcal{K}_c$ eine $\mathcal{N}(c,\sigma^2 I_n)$ normalverteilte Zufallsvariable mit Dichte f_c, wobei

$$f_c(x) = \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp\left(-\frac{\|c-x\|_2^2}{2\sigma^2}\right), \quad \text{für alle } x \in \mathbb{R}^n.$$

Ad (ii): f_c ist stetig und $f_{\varphi(u)}(y) > 0$ für alle $u \in \{\pm 1\}^k$ und alle $y \in \mathbb{R}^n$. Nach Definition 3.19 existiert damit $w(u,y)$ für alle $u \in \{\pm 1\}^k$ und alle $y \in \mathbb{R}^n$ und es gilt

$$\begin{aligned} w(u,y) &= P\left(\{\omega \in \Omega;\, U(\omega) = u\} \,|\, \{\omega \in \Omega;\, Y(\omega) = y\}\right) \\ &= \frac{\exp\left(-\frac{\|\varphi(u)-y\|_2^2}{2\sigma^2}\right)}{\sum\limits_{\tilde{u}\in\{\pm1\}^k} \exp\left(-\frac{\|\varphi(\tilde{u})-y\|_2^2}{2\sigma^2}\right)} = \frac{1}{\sum\limits_{\tilde{u}\in\{\pm1\}^k} \exp\left(\frac{\|\varphi(u)-y\|_2^2-\|\varphi(\tilde{u})-y\|_2^2}{2\sigma^2}\right)} \\ &= \frac{1}{\sum\limits_{\tilde{u}\in\{\pm1\}^k} \exp\left(\frac{\|\varphi(u)\|_2^2-\|\varphi(\tilde{u})\|_2^2+2y^\top(\varphi(\tilde{u})-\varphi(u))}{2\sigma^2}\right)} = \frac{1}{\sum\limits_{\tilde{u}\in\{\pm1\}^k} \exp\left(\frac{n-n+2y^\top(\varphi(\tilde{u})-\varphi(u))}{2\sigma^2}\right)} \\ &= \frac{1}{\sum\limits_{\tilde{u}\in\{\pm1\}^k} \exp\left(\frac{y^\top(\varphi(\tilde{u})-\varphi(u))}{\sigma^2}\right)}. \end{aligned}$$

Ad (iii): Es wird

$$f_{\varphi(u)}(y) = \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp\left(-\frac{\|\varphi(u)-y\|_2^2}{2\sigma^2}\right)$$

bezüglich u genau dann maximal, wenn

$$\|\varphi(u)-y\|_2$$

minimal wird.

□

Unter den Voraussetzungen von Satz 3.23 gilt es nun ein kombinatorisches Optimierungsproblem zu lösen, wobei der numerische Aufwand zur Lösung stark von der Codierungsabbildung φ abhängt. In Kapitel 5 wird ein neues Verfahren vorgestellt, mit dem sich beliebige (n,k)-Blockcodes behandeln lassen.

3.5 Soft-Output Decodierung

Insbesondere im Zusammenhang mit verketteten Codes ist man nicht allein am „harten“ Decodierungsergebnis $\hat{u} \in \{\pm 1\}^k$ interessiert, sondern benötigt „softe“ Aussagen $x \in \mathbb{R}^k$ über die Zuverlässigkeit dieser Decodierung. Der Betrag der Komponenten x_i, $i = 1,\ldots,k$, von x soll dabei ein Zuverlässigkeitsmaß darstellen, das heißt, ist $x_i = 0$, so konnte keine Decodierungsentscheidung getroffen werden. Ist der Betrag von x_i hingegen sehr groß, so wurde die Decodierungsentscheidung als sehr sicher angesehen.

Man betrachtet also eine weitere Decodierungsabbildung mit einem k-dimensionalen reellen Vektor als Ergebnis.

Definition 3.24 (Soft-Output Decodierungsabbildung)
Es sei $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum, $n,k \in \mathbb{N}$, $n > k$. Weiter sei (n,k,φ) ein binärer linearer (n,k)-Blockcode und $\mathcal{K}$ ein stetiger gedächtnisloser n-Kanal. Eine stetige Abbildung

$$\begin{aligned}\delta_{\mathrm{SO}} :\ & \mathbb{R}^n \to \mathbb{R}^k,\\ & y \mapsto x = \delta_{\mathrm{SO}}(y),\end{aligned}$$

die einer Realisierung y der Kanalausgabe ein $x \in \mathbb{R}^k$ zuordnet, heißt Soft-Output Decodierungsabbildung.

⌟

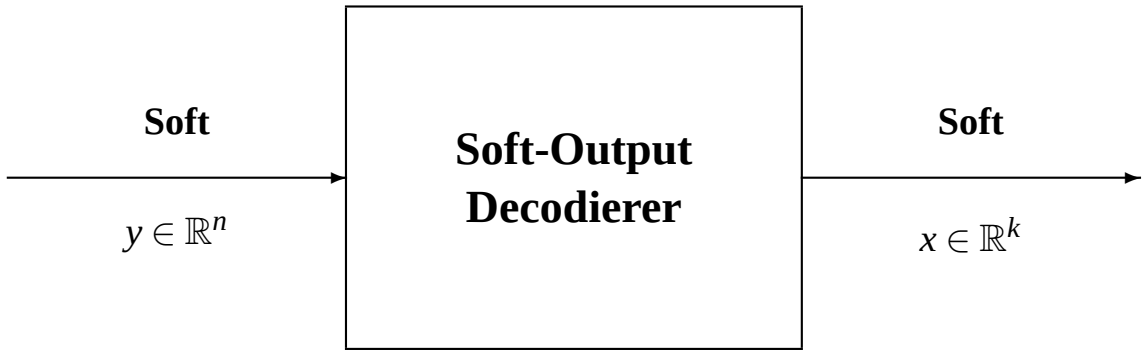

Abbildung 3.5: *Soft-Output Decodierer*

Analog zur Hard-Decision Decodierung und zur Soft-Decision Decodierung bezeichnen wir das technische Bauelement (oder den Algorithmus), welche(s/r) die Soft-Output Decodierungsabbildung repräsentiert, mit Soft-Output Decodierer, siehe Abbildung 3.5.

Bereits in Abschnitt 2.3 wurde der Begriff des Superkanals motiviert. Mit den bereitgestellten Werkzeugen kann man jetzt diesen Begriff präzisieren. Mit Hilfe einer Soft-Output Decodierungsabbildung δ_{SO} kann man eine Codierungsabbildung φ und einen n-Kanal $\mathcal{K}$ zu einem Superkanal

zusammenfassen. Im folgenden Lemma wird die Frage beantwortet, unter welchen Bedingungen dieser Superkanal einen k-Kanal darstellt und sogar selbst stetig ist.

Lemma 3.25 (Superkanal)
Es sei $(\Omega, \mathcal{S}, P)$ ein Wahrscheinlichkeitsraum, $n, k \in \mathbb{N}$, $n > k$. Weiter sei (n, k, φ) ein binärer linearer (n,k)-Blockcode, $\mathcal{K}$ ein stetiger n-Kanal, $Y : \Omega \to \mathbb{R}^n$ die Zufallsvariable der Kanalausgabe und $U : \Omega \to \{\pm 1\}^k$ eine von $\mathcal{K}_c$ für alle $c \in \{\pm 1\}^n$ stochastisch unabhängige Zufallsvariable mit

$$Y(\omega) = \mathcal{K}(\varphi(U(\omega)), \omega), \quad \text{für alle } \omega \in \Omega,$$

d.h., $\varphi(U)$ ist die Zufallsvariable der Kanaleingabe. Es sei $\delta_{\mathrm{SO}} : \mathbb{R}^n \to \mathbb{R}^k$ eine Soft-Output Decodierungsabbildung.

(i) *Dann ist*

$$\hat{\mathcal{K}} : \{\pm 1\}^k \times \Omega \to \mathbb{R}^k, \quad (u, \omega) \mapsto \delta_{\mathrm{SO}}(\mathcal{K}(\varphi(u), \omega))$$

ein k-Kanal. Der k-Kanal $\hat{\mathcal{K}}$ heißt dann Superkanal *bezüglich $(\varphi, \mathcal{K}, \delta_{\mathrm{SO}})$.*
U ist die Zufallsvariable der Kanaleingabe von $\hat{\mathcal{K}}$ und

$$X : \Omega \to \mathbb{R}^k, \quad \omega \mapsto \delta_{\mathrm{SO}}(\mathcal{K}(\varphi(U(\omega)), \omega))$$

ist die Zufallsvariable der Kanalausgabe von $\hat{\mathcal{K}}$.

(ii) *Ist das Bildmaß von λ^n unter der Abbildung δ_{SO} zusätzlich absolutstetig bezüglich λ^k mit reellwertiger Dichte $g_{\mathrm{SO}} : \mathbb{R}^k \to \mathbb{R}_0^+$, also*

$$\lambda^n(\delta_{\mathrm{SO}}^{-1}(A)) = \int_A g_{\mathrm{SO}} d\lambda^k, \quad \text{für alle } A \in \mathcal{B}^k,$$

dann ist $\hat{\mathcal{K}}$ ein stetiger k-Kanal.

Beweis. Ad (i): Betrachte für jedes $u \in \{\pm 1\}^k$ die Abbildungen

$$\hat{\mathcal{K}}_u : \Omega \to \mathbb{R}^k, \quad \omega \mapsto \hat{\mathcal{K}}(u, \omega).$$

Es ist

$$\hat{\mathcal{K}}_u(\omega) = \hat{\mathcal{K}}(u, \omega) = \delta_{\mathrm{SO}}(\mathcal{K}(\varphi(u), \omega)) = \delta_{\mathrm{SO}}\left(\mathcal{K}_{\varphi(u)}(\omega)\right) = \delta_{\mathrm{SO}} \circ \mathcal{K}_{\varphi(u)}(\omega), \quad \text{für alle } \omega \in \Omega.$$

$\mathcal{K}_{\varphi(u)}$ ist $\mathcal{S}$-$\mathcal{B}^n$-meßbar und δ_{SO} ist stetig und damit $\mathcal{B}^n$-$\mathcal{B}^k$-meßbar. Somit ist $\hat{\mathcal{K}}_u$ eine $\mathcal{S}$-$\mathcal{B}^k$-meßbare Zufallsvariable und $\mathcal{K}$ ein k-Kanal.

Die Zufallsvariable X der Kanalausgabe ist definiert über

$$X(\omega) := \hat{\mathcal{K}}(U(\omega), \omega) = \delta_{\mathrm{SO}}(\mathcal{K}(\varphi(U(\omega)), \omega)), \quad \text{für alle } \omega \in \Omega.$$

Ad (ii): Sei das Bildmaß von λ^n unter der Abbildung δ_{SO} absolutstetig bezüglich λ^k mit reellwertiger Dichte $g_{SO}: \mathbb{R}^k \to \mathbb{R}_0^+$, also

$$\lambda^n(\delta_{SO}^{-1}(A)) = \int_A g_{SO} d\lambda^k, \quad \text{für alle } A \in \mathcal{B}^k,$$

dann gilt für alle $u \in \{\pm 1\}^k$, $c := \varphi(u)$ und für alle $A \in \mathcal{B}^k$:

$$\begin{aligned}
P_{\hat{\mathcal{K}}_u}(A) &= P\left(\left\{\omega \in \Omega;\ \hat{\mathcal{K}}_u(\omega) \in A\right\}\right) \\
&= P\left(\{\omega \in \Omega;\ \delta_{SO} \circ \mathcal{K}_c(\omega) \in A\}\right) \\
&= P\left(\left\{\omega \in \Omega;\ \mathcal{K}_c(\omega) \in \delta_{SO}^{-1}(A)\right\}\right) \\
&= P_{\mathcal{K}_c}\left(\delta_{SO}^{-1}(A)\right) = \int\limits_{\delta_{SO}^{-1}(A)} f_c d\lambda^n \\
&\leq \sup\left\{f_c(y);\ y \in \delta_{SO}^{-1}(A)\right\} \cdot \lambda^n\left(\delta_{SO}^{-1}(A)\right) \\
&\leq \sup\left\{f_c(y);\ y \in \delta_{SO}^{-1}(A)\right\} \cdot \sup\{g_{SO}(x);\ x \in A\} \cdot \lambda^k(A)
\end{aligned}$$

und somit ist $P_{\hat{\mathcal{K}}_u}$ absolutstetig bezüglich λ^k und $\mathcal{K}$ ist ein stetiger k-Kanal.

□

Jede Komponente der Soft-Output Decodierung soll einen Zuverlässigkeitswert für das jeweilige Bit des Decodierungsergebnisses darstellen. Aus diesem Grund betrachtet man nicht die Wortfehlerwahrscheinlichkeit wie in Abschnitt 3.4, sondern nimmt die Bitfehlerwahrscheinlichkeit als Bewertungskriterium für verschiedene Soft-Output Decodierungsabbildungen.

Definition 3.26 (Soft-Decision Bitfehlerwahrscheinlichkeit)
Voraussetzung 3.12 auf Seite 40 sei erfüllt. Weiter sei $\delta_{SD}: \mathbb{R}^n \to \{\pm 1\}^k$ eine Soft-Decision Decodierungsabbildung. Dann heißt zu $i \in \{1,\ldots,k\}$

$$p_{E,bit}^i(\delta_{SD}) := P\left(\{\omega \in \Omega;\ [\delta_{SD}(Y(\omega))]_i \neq U_i(\omega)\}\right)$$

Bitfehlerwahrscheinlichkeit der Decodierungsabbildung δ_{SD} für das i-te Bit.

⌟

Auch die Bitfehlerwahrscheinlichkeit läßt sich mit Hilfe der Dichtefunktionen des Kanals darstellen.

Lemma 3.27 (Dichtedarstellung der Soft-Decision Bitfehlerwahrscheinlichkeit)
Voraussetzung 3.12 auf Seite 40 sei erfüllt, U sei gleichverteilt und $\delta_{SD}: \mathbb{R}^n \to \{\pm 1\}^k$ sei eine Soft-Decision Decodierungsabbildung. Dann gilt für alle $i \in \{1,\ldots,k\}$

$$p_{E,bit}^i(\delta_{SD}) = 1 - \frac{1}{2^k} \sum_{\alpha \in \{\pm 1\}} \int\limits_{M^i(\alpha)} \sum_{c \in \Gamma^i(\alpha)} f_c d\lambda^n$$

mit

$$M^i(\alpha) := \{y \in \mathbb{R}^n;\ [\delta_{SD}(y)]_i = \alpha\}, \quad \Gamma^i(\alpha) := \left\{\varphi(u);\ u \in \{\pm 1\}^k,\ u_i = \alpha\right\}$$

für alle $i \in \{1,\ldots,k\}$ und $\alpha \in \{\pm 1\}$.

⌟

Beweis. Es gilt für alle $i \in \{1,\dots,k\}$ mit Satz 3.14 auf Seite 40

$$\begin{aligned}
p^i_{E,bit}(\delta_{SD}) &= P(\{\omega\in\Omega;\ [\delta_{SD}(Y(\omega))]_i \neq U_i(\omega)\}) \\
&= 1 - P(\{\omega\in\Omega;\ [\delta_{SD}(Y(\omega))]_i = U_i(\omega)\}) \\
&= 1 - \sum_{u\in\{\pm1\}^k} P(\{\omega\in\Omega;\ [\delta_{SD}(Y(\omega))]_i = u_i\} \,|\, \{\omega\in\Omega;\ U(\omega)=u\}) \\
&\qquad\qquad \cdot P(\{\omega\in\Omega;\ U(\omega)=u\}) \\
&= 1 - \frac{1}{2^k} \sum_{u\in\{\pm1\}^k} P\left(\left\{\omega\in\Omega;\ Y(\omega)\in M^i(u_i)\right\} \,|\, \{\omega\in\Omega;\ U(\omega)=u\}\right) \\
&= 1 - \frac{1}{2^k} \sum_{u\in\{\pm1\}^k} \int\limits_{M^i(u_i)} f_{\varphi(u)}\, d\lambda^n \\
&= 1 - \frac{1}{2^k} \sum_{\alpha\in\{\pm1\}} \int\limits_{M^i(\alpha)} \sum_{u\in\{\pm1\}^k,\ u_i=\alpha} f_{\varphi(u)}\, d\lambda^n \\
&= 1 - \frac{1}{2^k} \sum_{\alpha\in\{\pm1\}} \int\limits_{M^i(\alpha)} \sum_{c\in\Gamma^i(\alpha)} f_c\, d\lambda^n .
\end{aligned}$$

□

Von einer Soft-Output Decodierung, die als Ausgang eines Superkanals und als Eingang für einen nachgeschalteten zweiten Decodierer dient, wird man erwarten, daß die komponentenweise Rundung auf ±1 ein bitfehlerminimales Decodierungsergebnis darstellt. Falls

$$\begin{aligned}
&P(\{\omega\in\Omega;\ U_i(\omega)=+1\} \,|\, \{\omega\in\Omega;\ Y(\omega)=y\}) \\
=&P(\{\omega\in\Omega;\ U_i(\omega)=-1\} \,|\, \{\omega\in\Omega;\ Y(\omega)=y\}),
\end{aligned}$$

so kann keine Aussage über das i-te Bit getroffen werden und die i-te Komponente des Soft-Outputs sollte daher 0 sein. Zur Bewertung der Zuverlässigkeit der Decodierungsentscheidung ist das Verhältnis

$$\frac{P(\{\omega\in\Omega;\ U_i(\omega)=+1\} \,|\, \{\omega\in\Omega;\ Y(\omega)=y\})}{P(\{\omega\in\Omega;\ U_i(\omega)=-1\} \,|\, \{\omega\in\Omega;\ Y(\omega)=y\})}$$

das natürliche Maß. Das logarithmierte Verhältnis spiegelt dann alle Anforderungen an Soft-Outputs wider und wird als L-Wert bezeichnet.

Definition 3.28 (L-Wert Soft-Output Decodierung)
Voraussetzung 3.12 auf Seite 40 sei erfüllt, U sei gleichverteilt und f_c sei für alle $c\in\{\pm1\}^n$ stetig und positiv. Für $i\in\{1,\dots,k\}$ und $y\in\mathbb{R}^n$ heißt

$$L(U_i|y) := \ln\left(\frac{P(\{\omega\in\Omega;\ U_i(\omega)=+1\} \,|\, \{\omega\in\Omega;\ Y(\omega)=y\})}{P(\{\omega\in\Omega;\ U_i(\omega)=-1\} \,|\, \{\omega\in\Omega;\ Y(\omega)=y\})}\right)$$

der i-te L-Wert, wobei

$$\begin{aligned}
&P(\{\omega\in\Omega;\ U_i(\omega)=\alpha\} \,|\, \{\omega\in\Omega;\ Y(\omega)=y\}) \\
&= \sum_{u\in\{\pm1\}^k,\ u_i=\alpha} P(\{\omega\in\Omega;\ U(\omega)=u\} \,|\, \{\omega\in\Omega;\ Y(\omega)=y\}),
\end{aligned}$$

mit $\alpha \in \{\pm 1\}$.
Die Soft-Output Decodierungsabbildung

$$\begin{aligned} \delta_{SO} : \mathbb{R}^n &\to \mathbb{R}^k, \\ y &\mapsto \begin{pmatrix} L(U_1|y) \\ \vdots \\ L(U_k|y) \end{pmatrix} \end{aligned}$$

heißt L-Wert Soft-Output Decodierungsabbildung.

Die in der Definition implizit angenommene Stetigkeit von δ_{SO} wird durch den folgenden Satz nachgewiesen. Außerdem wird bewiesen, daß die komponentenweise Rundung der L-Werte auf ± 1 tatsächlich eine bitfehlerminimale Soft-Decision Decodierung ist.

Satz 3.29 (Eigenschaften der L-Wert Soft-Output Decodierung)
Voraussetzung 3.12 auf Seite 40 sei erfüllt, U sei gleichverteilt und f_c sei für alle $c \in \{\pm 1\}^n$ stetig und positiv. $\delta_{SO} : \mathbb{R}^n \to \mathbb{R}^k$ sei die L-Wert Soft-Output Decodierungsabbildung. Dann gilt:

(i) *Für alle $i \in \{1,\dots,k\}$ und $y \in \mathbb{R}^n$ ist*

$$[\delta_{SO}(y)]_i = L(U_i|y) = \ln\left(\frac{\sum\limits_{c\in\Gamma^i(+1)} f_c(y)}{\sum\limits_{c\in\Gamma^i(-1)} f_c(y)}\right)$$

mit

$$\Gamma^i(\alpha) = \left\{\varphi(u);\ u \in \{\pm 1\}^k,\ u_i = \alpha\right\}, \quad \text{für alle } \alpha \in \{\pm 1\}.$$

(ii) *Die Soft-Decision Decodierungsabbildung $\delta_{SD} : \mathbb{R}^n \to \{\pm 1\}^k$ sei die gerundete L-Wert Soft-Output Decodierungsabbildung, also*

$$[\delta_{SD}(y)]_i := \begin{cases} +1, & \text{für } [\delta_{SO}(y)]_i = L(U_i|y) \geq 0, \\ -1, & \text{sonst} \end{cases} \quad \text{für } i \in \{1,\dots,k\},\ y \in \mathbb{R}^n.$$

Dann gilt

$$p^i_{E,bit}(\delta_{SD}) \leq p^i_{E,bit}(\delta), \quad \text{für alle } i \in \{1,\dots,k\},$$

für jede Soft-Decision Decodierungsabbildung $\delta : \mathbb{R}^n \to \{\pm 1\}^k$ und

$$p^i_{E,bit}(\delta_{SD}) = 1 - \frac{1}{2^k}\int\limits_{\mathbb{R}^n} \max\left\{\sum_{c\in\Gamma^i(+1)} f_c(x), \sum_{c\in\Gamma^i(-1)} f_c(x)\right\} d\lambda^n(x).$$

Beweis. Ad (i): Für alle $i \in \{1,\dots,k\}$ und $y \in \mathbb{R}^n$ gilt mit Definition 3.19 auf Seite 46

$$\begin{aligned} L(U_i|y) &= \ln\left(\frac{P(\{\omega\in\Omega;\ U_i(\omega)=+1\}\,|\,\{\omega\in\Omega;\ Y(\omega)=y\})}{P(\{\omega\in\Omega;\ U_i(\omega)=-1\}\,|\,\{\omega\in\Omega;\ Y(\omega)=y\})}\right) \\ &= \ln\left(\frac{\sum\limits_{u\in\{\pm1\}^k,\ u_i=+1} P(\{\omega\in\Omega;\ U(\omega)=u\}\,|\,\{\omega\in\Omega;\ Y(\omega)=y\})}{\sum\limits_{u\in\{\pm1\}^k,\ u_i=-1} P(\{\omega\in\Omega;\ U(\omega)=u\}\,|\,\{\omega\in\Omega;\ Y(\omega)=y\})}\right) \\ &= \ln\left(\frac{\sum\limits_{u\in\{\pm1\}^k,\ u_i=+1} f_{\varphi(u)}(y)}{\sum\limits_{u\in\{\pm1\}^k,\ u_i=-1} f_{\varphi(u)}(y)}\right) = \ln\left(\frac{\sum\limits_{c\in\Gamma^i(+1)} f_c(y)}{\sum\limits_{c\in\Gamma^i(-1)} f_c(y)}\right) \end{aligned}$$

Ad (ii): Es ist $M^i(-1) = \mathbb{R}^n \setminus M^i(+1)$ Daher wird

$$p^i_{E,bit}(\delta_{SD}) = 1 - \frac{1}{2^k} \sum_{\alpha\in\{\pm1\}} \int\limits_{M^i(\alpha)} \sum_{c\in\Gamma^i(\alpha)} f_c\, d\lambda^n$$

minimal, falls $M^i(+1)$ so gewählt werden kann, daß

$$M^i(+1) = \left\{ y\in\mathbb{R}^n;\ \sum_{c\in\Gamma^i(+1)} f_c(y) \geq \sum_{c\in\Gamma^i(-1)} f_c(y) \right\},$$

denn dann ist

$$p^i_{E,bit}(\delta_{SD}) = 1 - \frac{1}{2^k} \int\limits_{\mathbb{R}^n} \max\left\{ \sum_{c\in\Gamma^i(+1)} f_c(x),\ \sum_{c\in\Gamma^i(-1)} f_c(x) \right\} d\lambda^n(x).$$

Für die gerundete L-Wert Soft-Output Decodierungsabbildung δ_{SD} gilt nun, daß

$$\begin{aligned} M^i(+1) &= \{y\in\mathbb{R}^n;\ [\delta_{SD}(y)]_i = +1\} = \{y\in\mathbb{R}^n;\ L(U_i|y)\geq 0\} \\ &= \left\{ y\in\mathbb{R}^n;\ \ln\left(\frac{\sum\limits_{c\in\Gamma^i(+1)} f_c(y)}{\sum\limits_{c\in\Gamma^i(-1)} f_c(y)}\right) \geq 0 \right\} \\ &= \left\{ y\in\mathbb{R}^n;\ \sum_{c\in\Gamma^i(+1)} f_c(y) \geq \sum_{c\in\Gamma^i(-1)} f_c(y) \right\} \end{aligned}$$

und somit ist die Satzaussage bewiesen.

□

Die Rundung der 0 auf $+1$, die auch in den folgenden Kapiteln verwendet wird, ist eine willkürliche Wahl. Da jeder Soft-Output aber mit Wahrscheinlichkeit 0 einen bestimmten reellen Wert annimmt, ist die Art der Wahl irrelevant.

Da δ_{SO} als Verknüpfung stetiger Funktionen selbst stetig ist, ist die L-Wert Soft-Output Decodierung tatsächlich eine Soft-Output Decodierung.

Abschließend wird wieder der besonders wichtige Spezialfall eines AWGN-Kanals betrachtet.

Korollar 3.30 (L-Werte bei einem AWGN-Kanal)
Voraussetzung 3.12 auf Seite 40 sei erfüllt, U sei gleichverteilt und $\mathcal{K}$ sei ein AWGN-Kanal mit bitweiser Varianz $\sigma^2 > 0$ der Kanalstörung. $\delta_{SO} : \mathbb{R}^n \to \mathbb{R}^k$ sei die L-Wert Soft-Output Decodierungsabbildung. Für alle $i \in \{1,\ldots,k\}$ und $y \in \mathbb{R}^n$ ist dann

$$[\delta_{SO}(y)]_i = L(U_i|y) = \ln\left(\frac{\sum\limits_{c\in\Gamma^i(+1)} \exp\left(-\frac{\|c-y\|_2^2}{2\sigma^2}\right)}{\sum\limits_{c\in\Gamma^i(-1)} \exp\left(-\frac{\|c-y\|_2^2}{2\sigma^2}\right)}\right) = \ln\left(\frac{\sum\limits_{c\in\Gamma^i(+1)} \exp\left(\frac{y^\top c}{\sigma^2}\right)}{\sum\limits_{c\in\Gamma^i(-1)} \exp\left(\frac{y^\top c}{\sigma^2}\right)}\right)$$

mit

$$\Gamma^i(\alpha) = \left\{\varphi(u);\ u \in \{\pm 1\}^k,\ u_i = \alpha\right\}, \quad \textit{für alle } \alpha \in \{\pm 1\}.$$

Beweis. Die erste Darstellung folgt sofort aus Satz 3.29 unter Verwendung von Satz 3.23 auf Seite 47, wobei lediglich Konstanten gekürzt werden.

Die zweite Darstellung folgt mit der Umformung

$$\exp\left(-\frac{\|c-y\|_2^2}{2\sigma^2}\right) = \exp\left(-\frac{\|c\|_2^2+\|y\|_2^2-2y^\top c}{2\sigma^2}\right) = \exp\left(-\frac{n+\|y\|_2^2}{2\sigma^2}\right)\exp\left(\frac{y^\top c}{\sigma^2}\right).$$

□

Obwohl die in Korollar 3.30 dargestellten Formeln eine kurze geschlossene Form aufweisen, ist die Auswertung in der Regel numerisch sehr aufwendig, da pro L-Wert alle 2^k Codewörter betrachtet werden müssen. In Kapitel 4 wird ein Verfahren vorgestellt, mit dem sich bei einer bestimmten wichtigen Sorte von Codierungsabbildungen dieser numerische Aufwand drastisch reduzieren läßt.

3.6 Decodierung verketteter Codes

Bei verketteten binären linearen (n,k)-Blockcodes werden zwei oder mehr Codierungsabbildungen zu einer Gesamtcodierungsabbildung verknüpft, siehe Definition 2.15 auf Seite 27.

Ein Beispiel aus dem Mobilfunk sind die GSM-Kontrollkanäle (GSM = Global System for Mobile communications), siehe [GSM96a, GSM96b, DB96], die einen verketteten binären linearen $(456,184)$-Blockcode $((224,184,\varphi_1),(456,224,\varphi_2))$ verwenden[7]. Dabei ist $(224,184,\varphi_1)$ ein spezieller systematischer Blockcode (ein sogenannter Fire-Code) und $(456,224,\varphi_2)$ ein terminierter Faltungscode[8]. Vor der Übertragung über den (physikalischen) Kanal werden die Bits dann noch umsortiert (Interleaving), um Bündelfehler abzuschwächen. Diese übliche Maßnahme soll den tatsächlichen physikalischen Kanal einem theoretischen AWGN-Kanal ähnlicher machen, um die Effizienz der für AWGN-Kanäle konstruierten Decodierungsmethoden zu verbessern.

[7] In der Spezifikation [GSM96b] besteht die Ausgabe des ersten Codierers / die Eingabe des zweiten Codierers aus 228 Bits. Der Grund ist aber nur eine andere Zuordnung der Terminierungsbits des Faltungscodes als in der hier vorliegenden Terminologie.

[8] Terminierte Faltungscodes werden in Abschnitt 4.2 ab Seite 60 definiert.

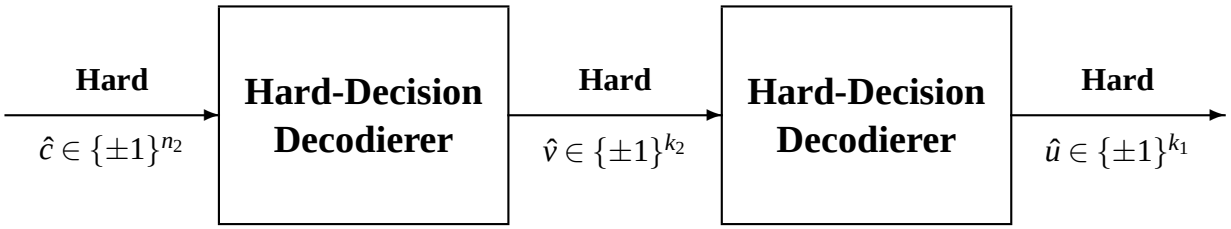

Abbildung 3.6: *Decodierung verketteter Codes: Hard → Hard → Hard*

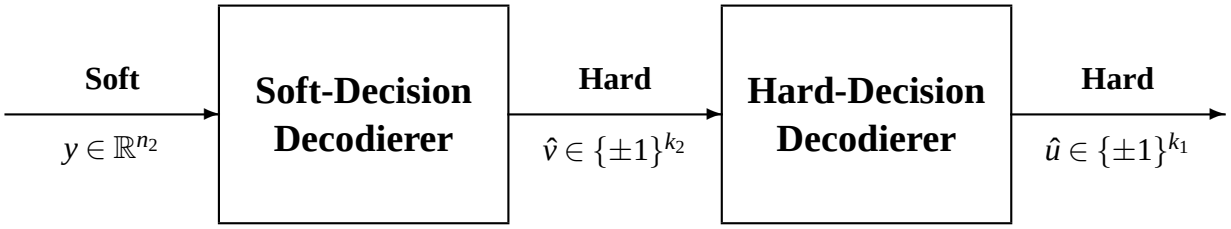

Abbildung 3.7: *Decodierung verketteter Codes: Soft → Hard → Hard*

Mit Hilfe der in den vorangegangenen Abschnitten bereitgestellten Werkzeuge läßt sich die Behandlung verketteter Codes jetzt einfach darstellen. Ohne Einschränkung der Allgemeinheit kann man sich auf die Behandlung von zwei Teilcodes beschränken, da sich die Vorgehensweisen sofort auf mehr als zwei Teilcodes erweitern lassen.

Im weiteren wird ein verketteter binärer linearer (n,k)-Blockcode $((n_1,k_1,\varphi_1),(n_2,k_2,\varphi_2))$ betrachtet mit

$$k = k_1 < n_1 = k_2 < n_2 = n.$$

Da der Ausgang des ersten Decodierers der Eingang für den zweiten Decodierer ist, schränkt die erste Decodierungsmethode in der Regel die Auswahl der zweiten Methode ein. Die drei prinzipiellen Methoden Hard-Decision, Soft-Decision und Soft-Output wurden durch drei Diagramme in Abbildung 3.3, Abbildung 3.4 und Abbildung 3.5 dargestellt, die nun in einfacher Weise verknüpft werden können. Die Schnittstelle der Verknüpfung wird dabei entweder durch Soft-Werte (Soft-Schnittstelle) oder durch Hard-Werte (Hard-Schnittstelle) realisiert.

Es lassen sich die folgenden Varianten bilden:

- Der (n_2,k_2)-Blockcode (n_2,k_2,φ_2) wird mit einer Hard-Decision Decodierungsmethode behandelt und der (n_1,k_1)-Blockcode (n_1,k_1,φ_1) wird ebenfalls mit einer Hard-Decision Decodierungsmethode decodiert, vergleiche Abbildung 3.6.

- Der (n_2,k_2)-Blockcode (n_2,k_2,φ_2) wird mit einer Soft-Decision Decodierungsmethode behandelt und der (n_1,k_1)-Blockcode (n_1,k_1,φ_1) wird mit einer Hard-Decision Decodierungsmethode decodiert, vergleiche Abbildung 3.7.

- Der (n_2,k_2)-Blockcode (n_2,k_2,φ_2) wird mit einer Soft-Output Decodierungsmethode behandelt und der (n_1,k_1)-Blockcode (n_1,k_1,φ_1) wird mit einer Soft-Decision Decodierungsmethode decodiert, vergleiche Abbildung 3.8.

- Der (n_2,k_2)-Blockcode (n_2,k_2,φ_2) wird mit einer Soft-Output Decodierungsmethode behandelt und der (n_1,k_1)-Blockcode (n_1,k_1,φ_1) wird ebenfalls mit einer Soft-Output Decodierungsmethode decodiert, vergleiche Abbildung 3.9.

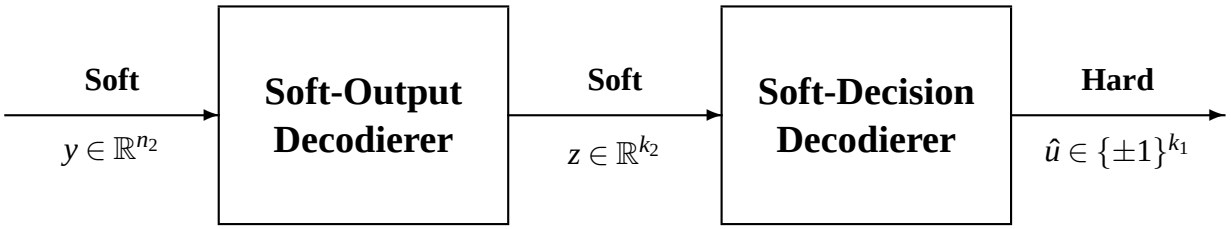

Abbildung 3.8: *Decodierung verketteter Codes: Soft → Soft → Hard*

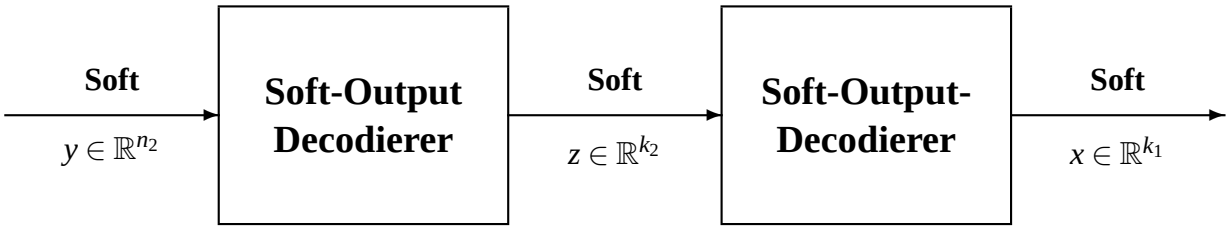

Abbildung 3.9: *Decodierung verketteter Codes: Soft → Soft → Soft*

Da verkettete Codes insbesondere auch „gewöhnliche“ (n,k)-Blockcodes sind (vergleiche Lemma 2.16), kann man einen verketteten Code alternativ auch direkt wie in den Abschnitten 3.3, 3.4 und 3.5 decodieren. Die drei Varianten sind in Abbildung 3.10 auf Seite 58 gemeinsam dargestellt. Einzelcodes werden jedoch gewöhnlich mit speziellen Eigenschaften entworfen, die sich in der Regel im verketteten Gesamtcode nicht wiederfinden. Bei praktischen Beispielen mit entsprechend hoher Komplexität hat sich die Einzeldecodierung der Teilcodes auch entsprechend als günstiger gegenüber der Gesamtdecodierung in einem Schritt herausgestellt.

Im einführenden Beispiel wurde eine Verkettung aus einem (systematischen) Blockcode und einem terminierten Faltungscode verwendet. Dies ist eine der wichtigsten Verkettungen und wird am häufigsten verwendet (bisweilen mit einem Interleaving-Schritt zwischen beiden Teilcodierung). Der Grund dafür liegt in der Anwendbarkeit zweier Standard-Verfahren zur Teildecodierung dieser Codes. Für die Decodierung des terminierten Faltungscodes wird üblicherweise der Viterbi-Algorithmus[9], ein Minimaldistanz Soft-Decision Verfahren, verwendet. Den Blockcode behandelt man dann mit einem Hard-Decision Verfahren, etwa einem Syndromkorrektur-Verfahren[10], welches nicht fehlerminimal ist. Somit entspricht diese Standard-Vorgehensweise Abbildung 3.7. Details zu diesen Methoden sind etwa in [VO79, Roh95, Fri95, Pro01] zu finden.

Hauptmanko dieser Vorgehensweise ist die Kopplung beider Decodierer durch „harte“ Werte, das heißt, der Soft-Decision Decodierer kann dem nächsten Decodierer keine Informationen darüber mitteilen, wie sicher die erste Decodierung durchgeführt werden konnte. Es wäre daher erheblicher

[9] Der Viterbi-Algorithmus nutzt eine spezielle Darstellung des Codebaums bei terminierten Faltungscodes aus, die als Trellis-Diagramm bekannt ist. In Kapitel 4 wird ebenfalls das Trellis-Diagramm zur Konstruktion einer L-Wert Soft-Output Decodierung genutzt. Das Viterbi-Verfahren gehört zu den Minimalfehler Soft-Decision Methoden, die fehlerminimale „harte“ Entscheidungen treffen. Für eine optimale Kopplung mit einem nachgeschalteten Decodierer werden jedoch auch Zuverlässigkeitsinformationen benötigt.

[10] Eine Matrix $H \in \mathbb{R}^{n-k,n}$ heißt Kontrollmatrix oder Prüfmatrix eines (n,k)-Blockcodes (n,k,φ), wenn die Spalten von H den Nullraum der injektiven Abbildung φ aufspannen. Somit ist $H \odot \varphi(u)$ das Nullelement in $\{\pm 1\}^{n-k}$ für jedes $u \in \{\pm 1\}^k$. Das Wort $H \odot \hat{c} \in \{\pm 1\}^{n-k}$ heißt dann das Syndrom von $\hat{c} \in \{\pm 1\}^n$. Ist nun $\hat{c} = c \oplus e$, wobei $c \in \varphi(\{\pm 1\}^k)$ ein gültiges Codewort ist und $e \in \{\pm 1\}^n$ eine Realisierung der Störung auf dem diskreten n-Kanal darstellt, so gilt $H \odot \hat{c} = H \odot e$ und folglich hängt das Syndrom nur von der Störung selbst ab. Man kann nun etwa die häufigsten Störungen mit ihren Syndromen tabellieren und anhand des Syndroms eines empfangenen Wortes $\hat{c}$ über diese Tabelle das ursprüngliche Wort $c = \hat{c} \oplus e$ zurückrechnen. Methoden dieser Art heißen Syndromkorrektur-Verfahren.

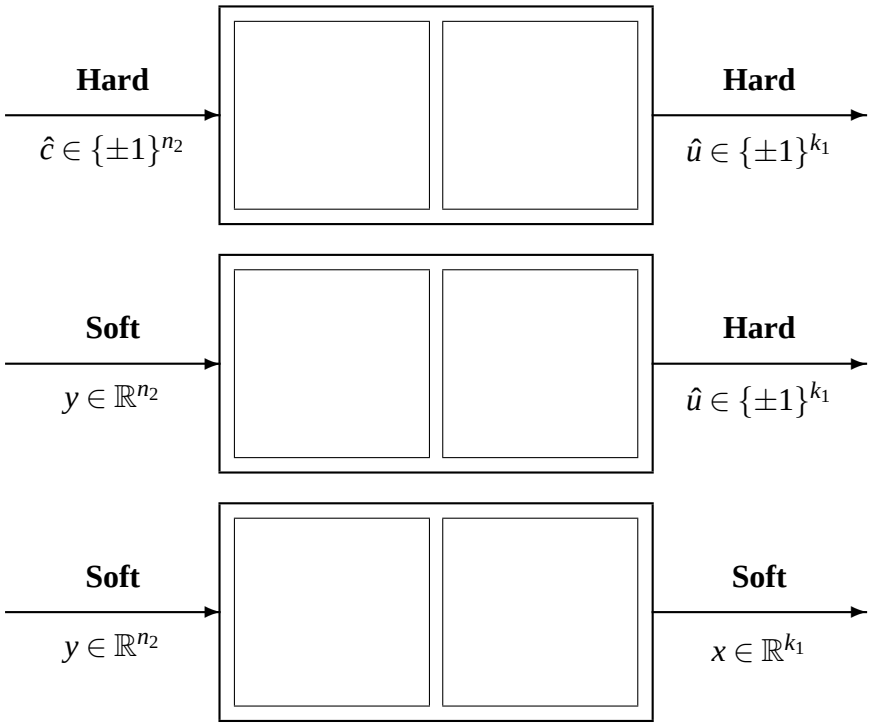

Abbildung 3.10: *Decodierung verketteter Codes: Betrachtung der Gesamtabbildung*

effizienter, eine Kopplung durch „softe“ Werte (Soft-Schnittstelle) zu verwenden wie in Abbildung 3.8, bei der diese fehlenden Informationen vorhanden sind. Bei gleichartigen Eingaben und Ausgaben des Gesamtdecodierers ist dadurch eine deutliche Fehlerreduktion zu erwarten.

An dieser Stelle soll noch einmal betont werden, dass hier eine *allgemeine* Verkettung von beliebigen binären linearen Blockcodes betrachtet wird und keine speziellen Anordnungen wie bei den Turbo-Codes [BM96, Rie97, BDMP98], d.h. auch die nachfolgend erarbeiteten Analysen und Verfahren gelten für alle binären linearen Blockcodes. In [HHFJ02] wird gezeigt, dass bei einer speziellen *idealen* Wahl des Codes Soft- und Hard-Decodierung äquivalent sind.

In der hier vorliegenden allgemeinen Situation bedeutet eine Änderung der Schnittstelle von Hard auf Soft zwischen beiden Decodierern zwangsläufig, daß **beide** Decodierungsverfahren geändert werden müssen. Man benötigt also die folgenden Methoden:

- Ein Soft-Output Decodierungsverfahren für terminierte Faltungscodes (siehe Kapitel 4).
- Ein Soft-Decision Decodierungsverfahren für allgemeine Blockcodes (siehe Kapitel 5).

In den folgenden Kapiteln werden die benötigten Verfahren entwickelt und erläutert.

Kapitel 4

Soft-Output Decodierung terminierter Faltungscodes

The hard and strong will fall.
The soft and weak will overcome.

(Lao-tse, „Tao Te Ching“)

4.1 Verfahrens-Verkettung

In Abschnitt 3.6 wurde dargelegt, daß zwei Decodierer über eine Soft-Schnittstelle (Soft-Werte) gekoppelt werden sollten, um eine maximale Informationsweitergabe zu gewährleisten.

In diesem Kapitel wird die **Soft-Output Decodierung** von terminierten Faltungscodes betrachtet, die in zahlreichen Anwendungen der Praxis eingesetzt werden, da zur **Soft-Decision Decodierung** dieser Codes der bekannte Viterbi-Algorithmus (siehe etwa [Fri95, Roh95, VO79, Vit95]) verwendet werden kann. Das im folgenden entwickelte Soft-Output Verfahren kann damit als optimale Decodierungsmethode bei verketteten Codes eingesetzt werden, siehe Abbildung 4.1.

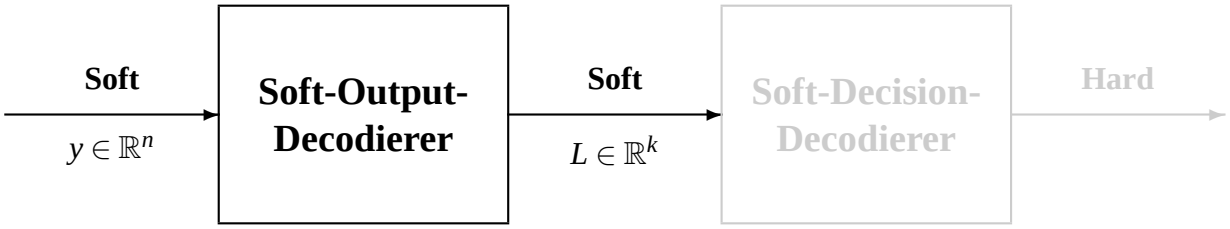

Abbildung 4.1: *Soft-Output Decodierung bei verketteten Codes*

In der weiteren Darstellung in diesem Kapitel kann das Verfahren isoliert betrachtet werden, da die Kopplung mit vorgeschalteten oder, wie im Bild dargestellt, nachgeschalteten Verfahren über die Schnittstelle der Soft-Werte erfolgt.

4.2 Terminierte Faltungscodes

Die spezielle Struktur einer Faltungscodierung wird meist in Form eines „Schieberegisters“ dargestellt.

Abbildung 4.2 zeigt das Prinzip einer Codierung mit Hilfe eines Schieberegisters[1].

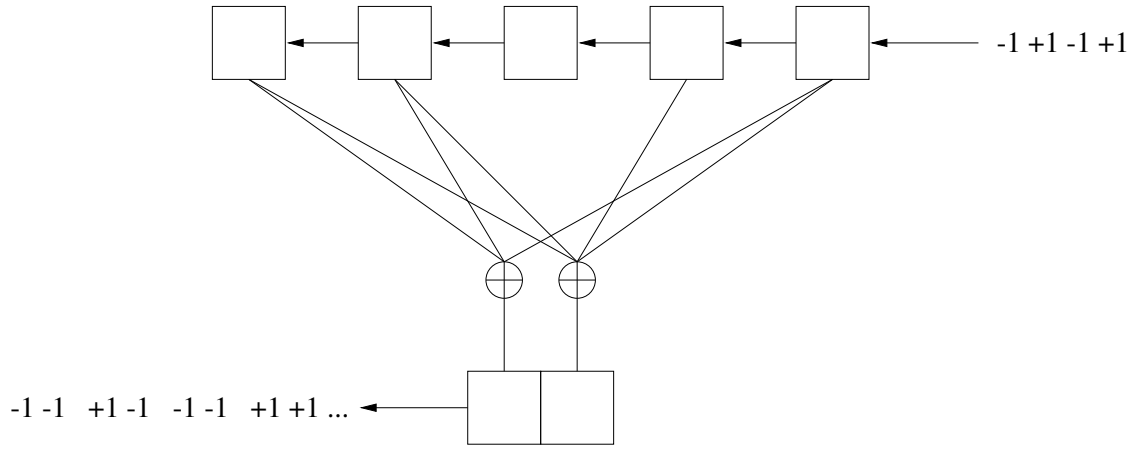

Abbildung 4.2: *Schieberegister-Darstellung einer terminierten Faltungscodierung*

Das Schieberegister besteht aus einer Reihe von Bitblöcken mit einem oder mehreren Bits pro Block, die zu Beginn der Codierung jeweils mit $+1$ vorbelegt sind. Sukzessive werden dann die Bits des uncodierten Wortes $u \in \{\pm 1\}^k$ blockweise in das Schieberegister geschoben. Bei jedem Takt werden dann die im Schieberegister befindlichen Bits über eine Codierungsvorschrift zu einem Ausgabeblock codiert, das heißt, jedes Bit eines Ausgabeblocks ist die binäre Summe einer Untermenge der Bits des Schieberegisters. Die Folge der Ausgabeblöcke stellt dann das Codewort $c \in \{\pm 1\}^n$ dar. Zur sogenannten Terminierung der Codierung werden nach dem letzten Eingabeblock des Wortes u noch solange „leere“ Blöcke (mit $+1$) in das Schieberegister geschoben (und selbiges codiert), bis der letzte Block von u aus dem Schieberegister verschwunden ist.

Den Inhalt des Schieberegisters bezeichnet man auch als Zustand des Schieberegisters. Die im weiteren Verlauf bedeutenden Eigenschaften der terminierten Faltungscodes resultieren im wesentlichen aus der in der Regel vergleichsweise geringen Anzahl von Zuständen des Schieberegisters, die eine drastische Kompaktifizierung des Codebaums erlauben wird.

Zunächst betrachten wir eine genaue Definition der terminierten (n,k)-Faltungscodes.

Definition 4.1 (termininierter (n,k)-Faltungscode)
Ein terminierter (n,k)-Faltungscode ist ein Tupel $(a,b,l,d,M_1,\ldots,M_d)$ mit den Eigenschaften:

(i) $a \in \mathbb{N}$ *heißt die Anzahl der Eingabeblöcke,*

(ii) $b \in \mathbb{N}$ *heißt die Anzahl der Eingabebits pro Eingabeblock,*

(iii) $l \in \mathbb{N}$, $l \geq 2$ *heißt die Blocklänge des Schieberegisters (Eindringtiefe),*

(iv) $d \in \mathbb{N}$ *heißt die Anzahl der Ausgabebits pro Ausgabeblock,*

[1] Die Abbildung 4.2 veranschaulicht das Schieberegister für eine Reihe von Kontrollkanälen des GSM-Mobilfunkstandards, vergleiche [GSM96b].

(v) $L := l \cdot b$ *heißt die Bitlänge des Schieberegisters,*

(vi) $M_1, \ldots, M_d \subseteq \{1, \ldots, L\}$ *heißen die definierenden Mengen,*

(vii) $S := \{\pm 1\}^L$ *heißt die Menge der Zustände des Schieberegisters,*

(viii) $V := \{\pm 1\}^b$ *heißt die Menge der Zustandsübergangszeichen,*

(ix) $Q := a + l - 1$ *heißt die Anzahl der Zustandsübergänge,*

(x) $k = a \cdot b$ *ist die Codedimension,*

(xi) $n := d \cdot Q$ *ist die Codelänge,*

(xii) *die Abbildung*

$$\begin{aligned} T:\ & S \times V \to S, \\ & (s, v) \mapsto (s^{b+1}, \ldots, s^L, v^1, \ldots, v^b)^\top \end{aligned} \tag{4.1}$$

heißt die Zustandsübergangsfunktion[2] *des Schieberegisters,*

(xiii) *die Abbildung*

$$\begin{aligned} \psi:\ & S \to \{\pm 1\}^d, \\ & s \mapsto \psi(s), \quad \text{mit } \psi_m(s) := \bigoplus_{i \in M_m} s^i, \quad \text{für } 1 \le m \le d. \end{aligned}$$

heißt die Codierungsabbildung des Schieberegisterinhaltes (dabei steht s^i für die i-te Komponente von s),

(xiv) $s_0 \in S$ *und* $v_0 \in V$ *sind die jeweiligen Nullelemente, d.h.*

$$s_0 = (+1, \ldots, +1)^\top, \quad v_0 = (+1, \ldots, +1)^\top,$$

(xv) *die Abbildung*

$$\begin{aligned} \varphi:\ & \{\pm 1\}^k \to \{\pm 1\}^n, \\ & u \mapsto \begin{pmatrix} \psi(s_1^u) \\ \vdots \\ \psi(s_Q^u) \end{pmatrix}, \end{aligned}$$

mit

$$\begin{aligned} u = \begin{pmatrix} v_1 \\ \vdots \\ v_a \end{pmatrix}, \quad & v_i \in V, \quad 1 \le i \le a, \\ & v_i := v_0, \quad a+1 \le i \le Q, \end{aligned}$$

und

$$\begin{aligned} s_1^u &:= T(s_0, v_1), \\ s_i^u &:= T(s_{i-1}^u, v_i), \quad 2 \le i \le Q, \end{aligned}$$

ist die Codierungsabbildung eines (n,k)-Blockcodes (n, k, φ).

[2] Die Zustandsübergangsfunktion des Schieberegisters ist in Abbildung 4.5 auf Seite 70 veranschaulicht.

Als Punkt (xv) wird also als definierende Eigenschaft gefordert, daß φ injektiv ist (φ ist linear nach Konstruktion). Damit sind implizit Eigenschaften für die Mengen $M_1,\ldots,M_d$ gefordert.

Anhand der obigen Definition sieht man sofort, daß

$$s^u_{Q+1} := T(s^u_Q, v_0) = s_0$$

und somit ist das Schieberegister bei Ende der Codierungsoperation wieder „leer“.

In Definition 4.1 wurden als Codedimension $k = a \cdot b$ nur die tatsächlichen informationstragenden Infobits gezählt. Bisweilen wird jedoch auch die Bitzahl der „nachgeschobenen leeren Blöcke“ zur Codedimension hinzugerechnet, so daß sich die Codedimension dann als $\tilde{k} = Q \cdot b$ ergibt.

Oft werden statt der definierenden Mengen M_m Polynome $p_m(x) \in \{0,1\}[x]$ mit $\deg(p_m(x)) \leq L-1$ zur Codedefinition verwendet, d.h.

$$p_m(x) = \sum_{i=0}^{L-1} \gamma_{i,m} x^i, \quad \text{mit } \gamma_{i,m} \in \{0,1\},\ i = 0,\ldots,L-1,\ m = 1,\ldots,d.$$

Es gelten dann für $m = 1,\ldots,d$ die Umformungen:

$$\begin{aligned} M_m &= \{i \in \{1,\ldots,L\};\ \gamma_{L-i,m} = 1\} \\ p_m(x) &= \sum_{i \in M_m} x^{L-i}. \end{aligned}$$

Da ein terminierter Faltungscode per definitionem ein Blockcode ist, lassen sich nach Lemma 2.10 die Codebits c_j, $1 \leq j \leq n$, aus den Nachrichtenbits u_i, $1 \leq i \leq k$, mit Indexmengen J_j auch alternativ wie folgt darstellen:

$$c_j := \bigoplus_{i \in J_j} u_i, \quad \text{für } 1 \leq j \leq n,$$

wobei

$$J_1,\ldots,J_n \subseteq \{1,\ldots,k\}.$$

Die Indexmengen J_j lassen sich direkt aus den definierenden Mengen M_m der Codedefinition berechnen.

Lemma 4.2 (Blockcodedarstellung eines terminierten (n,k)-Faltungscodes)
Sei $(a,b,l,d,M_1,\ldots,M_d)$ ein terminierter (n,k)-Faltungscode. Die charakterisierenden Mengen $J_1,\ldots,J_n \subseteq \{1,\ldots,k\}$ der Blockcodedarstellung sind dann wie folgt gegeben:

$$J_j = \left\{ i \in \{1,\ldots,k\};\ i + b\left(l - 1 - \frac{j-m}{d}\right) \in M_m \right\}, \quad \textit{für } m - 1 = (j-1) \bmod d,\ j = 1,\ldots,n.$$

Beweis. Betrachte für $j \in \{1,\ldots,n\}$

$$m := (j-1) \bmod d + 1.$$

Es ist

$$q := \frac{j-m}{d} + 1$$

der q-te Zustand des Schieberegisters und

$$j = d(q-1)+m.$$

Dann gilt

$$c_j = \psi_m(s_q^u) = \bigoplus_{i\in M_m} (s_q^u)^i = \bigoplus_{i\in M_m} u_{i+b(q-l)},$$

wobei $u_i := +1$ für $i \notin \{1,\ldots,k\}$.

Weiter gilt

$$c_j = \bigoplus_{i-b(q-l)\in M_m} u_i = \bigoplus_{i\in M_m+b(q-l)} u_i,$$

und somit folgt für $j = 1,\ldots,n$

$$\begin{aligned} J_j &= \{1,\ldots,k\} \cap (M_m + b(q-l)) = \{i \in \{1,\ldots,k\};\ i-b(q-l) \in M_m\} \\ &= \left\{ i \in \{1,\ldots,k\};\ i+b\left(l-1-\frac{j-m}{d}\right) \in M_m \right\}. \end{aligned}$$

□

Die Bestimmung der charakterisierenden Mengen läßt sich in einfacher Weise als Algorithmus 4.1 formulieren, wobei die Bezeichnungen wie in Definition 4.1 gewählt sind.

Blockcodedarstellung: **Eingang** Q, d, k, M_m, $1 \le m \le d$; **Ausgang** J_j, $1 \le j \le n$;

für $q = 1,\ldots,Q$:
  für $m = 1,\ldots,d$:
    $j := d(q-1)+m$;
    $J_j := \emptyset$;
    für $r \in M_m$:
      $i := r + b(q-l)$;
      falls $(i \ge 1) \wedge (i \le k)$:
        $J_j := J_j \cup \{i\}$;

Algorithmus 4.1: *Blockcodedarstellung eines terminierten (n,k)-Faltungscodes*

4.3 Beispiel SACCH-Faltungscode

Für eine Reihe von Kontrollkanälen des GSM-Mobilfunkstandards wird in [GSM96b] der terminierter Faltungscode definiert, welcher in Abbildung 4.2 auf Seite 60 dargestellt wurde. Diesen terminierten Faltungscode wollen wir im weiteren als SACCH-Faltungscode (**S**low **A**ccess **C**ontrol **Ch**annel) bezeichnen.

In der Terminologie von Definition 4.1 auf Seite 60 ist der SACCH-Faltungscode ein terminierter $(224,1,5,2,\{1,2,5\},\{1,2,4,5\})$-Faltungscode und besitzt die folgenden Eigenschaften:

$a = 224$	Anzahl der Eingabeblöcke
$b = 1$	Anzahl der Eingabebits pro Eingabeblock
$l = 5$	Blocklänge des Schieberegisters
$d = 2$	Anzahl der Ausgabebits pro Ausgabeblock
$L = 5$	Bitlänge des Schieberegisters
$M_1 = \{1,2,5\}$	definierende Menge; Polynom: $1+x^3+x^4$
$M_2 = \{1,2,4,5\}$	definierende Menge; Polynom: $1+x+x^3+x^4$
$S = \{\pm 1\}^5$	Menge der Schieberegisterzeichen
$V = \{\pm 1\}$	Menge der Zustandsübergangszeichen
$Q = 228$	Anzahl der Zustandsübergänge
$k = 224$	Codedimension
$n = 456$	Codelänge

Insbesondere stellt der SACCH-Faltungscode einen binären linearen $(456,224)$-Blockcode dar. Ab Seite 150 wird die Anwendung des im folgenden entwickelten numerischen Verfahrens auf diesen Faltungscode dargestellt.

4.4 Soft-Outputs bei terminierten Faltungscodes

In Abschnitt 3.5 wurde die Soft-Output Decodierung für allgemeine binäre lineare Blockcodes bei stetigen gedächtnislosen Kanälen untersucht und die spezielle L-Wert Soft-Output Decodierung (siehe Definition 3.28 auf Seite 52) hergeleitet. Im weiteren Verlauf betrachten wir nur noch diese Art der Soft-Output Decodierung unter der Annahme eines AWGN-Kanalmodells und der Verwendung von terminierten Faltungscodes.

Voraussetzung 4.3 (Decodierung terminierter Faltungscodes)
Es sei $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum, $n,k \in \mathbb{N}$, $n > k$. Weiter sei $(a,b,l,d,M_1,\ldots,M_d)$ ein terminierter (n,k)-Faltungscode mit Codierungsabbildung $\varphi\colon \{\pm 1\}^k \to \{\pm 1\}^n$, $\mathcal{K}$ sei ein AWGN-Kanal der Dimension n mit bitweiser Varianz $\sigma^2 > 0$ der Kanalstörung, $Y\colon \Omega \to \mathbb{R}^n$ sei die Zufallsvariable der Kanalausgabe und $U\colon \Omega \to \{\pm 1\}^k$ eine von $\mathcal{K}_c$ für alle $c \in \{\pm 1\}^n$ stochastisch unabhängige Zufallsvariable mit

$$Y(\omega) = \mathcal{K}(\varphi(U(\omega)),\omega), \quad \text{für alle } \omega \in \Omega,$$

d.h., $\varphi(U)$ ist die Zufallsvariable der Kanaleingabe. Weiter gelte, daß U gleichverteilt sei.

Man beachte, daß Voraussetzung 4.3 die Voraussetzung 3.12 auf Seite 40 umfaßt. Unter Voraussetzung 4.3 verwenden wir die in Definition 4.1 auf Seite 60 getroffenen Sprachregelungen.

In Korollar 3.30 auf Seite 55 wurde bereits eine konstruktive Vorschrift zur Berechnung der L-Wert Soft-Outputs $L(U_i|y)$, $1 \le i \le k$, bei gegebener Realisierung y von Y angegeben. Betrachtet man die folgende Funktion[3]

$$\begin{aligned} f:\ \{\pm 1\}^n &\to \mathbb{R}_0^+, \\ c &\mapsto \sum_{j=1}^{n} (y_j - c_j)^2, \end{aligned}$$

so gilt mit Korollar 3.30 unter Voraussetzung 4.3

$$L(U_i|y) = \ln\left(\frac{\sum\limits_{c\in\Gamma^i(+1)} \exp\left(-\frac{1}{2\sigma^2} f(c)\right)}{\sum\limits_{c\in\Gamma^i(-1)} \exp\left(-\frac{1}{2\sigma^2} f(c)\right)}\right), \quad \text{für } i = 1,\ldots,k, \tag{4.2}$$

mit

$$\Gamma^i(\alpha) = \left\{\varphi(u);\ u \in \{\pm 1\}^k,\ u_i = \alpha\right\}, \quad \text{für alle } \alpha \in \{\pm 1\}.$$

Bei Auswertung von (4.2) muß also k-mal der Funktionswert von f für alle 2^k Codewörter berechnet werden. Da 2^k im Beispiel des SACCH-Faltungscodes mit $k = 224$ bereits die Größenordnung 10^{67} besitzt, ist eine direkte numerische Auswertung bei realen Beispielen nicht durchführbar. Mit Hilfe der Faltungscodeeigenschaften kann aber im weiteren diese numerische Komplexität drastisch reduziert werden. Mit anderen Ansätzen, etwa über einen Markov-Prozeß, finden sich etwa in [BCJR74, HOP96, Rie97] vergleichbare Komplexitätsreduktionen, die aber numerisch aufwendiger sind als die hier hergeleiteten Rekursionen in Definition 4.9 auf Seite 70, Definition 4.11 auf Seite 73 und Satz 4.14 auf Seite 75, die zu Algorithmus 4.3 auf Seite 79 beziehungsweise Algorithmus 4.4 auf Seite 80 führen.

Da als Argumente von f nur zulässige Codeworte in (4.2) verwendet werden, läßt sich im folgenden die Funktion[4]

$$\begin{aligned} F:\ \{\pm 1\}^k &\to \mathbb{R}_0^+ \\ u &\mapsto f(\varphi(u)) \end{aligned} \tag{4.3}$$

betrachten, die uncodierte Worte als Argumente besitzt.

Zunächst läßt sich ein weiteres Korollar zu Satz 3.29 auf Seite 53 angeben, in dem die spezielle Struktur der Faltungscodierung zur Geltung kommt und stückweise Auswertungen $\Delta F_\bullet$ der Funktion F verwendet werden können.

Korollar 4.4 (L-Werte bei (n,k)-Faltungscodes)
Voraussetzung 4.3 auf Seite 64 sei erfüllt. Mit

$$\Delta F_q(s) := \sum_{j=1}^{d} \left(y_{d(q-1)+j} - \psi_j(s)\right)^2, \tag{4.4}$$

[3] Die Funktion f stellt auch eine Bewertungsfunktion (eine sogenannte Viterbi-Metrik) von Codewörtern dar, da mit ihrer Hilfe eine Minimaldistanz Soft-Decision Decodierung durchgeführt werden kann, vergleiche Satz 3.23 auf Seite 47.

[4] In Kapitel 5 spielt die Funktion F eine wichtige Rolle als sogenannte Soft-Decision Zielfunktion, vergleiche Definition 5.2 auf Seite 92.

für $s \in S$, $q = 1, \ldots, Q$, und

$$A^i_\alpha(y) := \sum_{u \in \{\pm 1\}^k,\ u_i = \alpha} \prod_{q=1}^{Q} \exp\left(-\frac{1}{2\sigma^2} \Delta F_q(s^u_q)\right), \tag{4.5}$$

für $i = 1, \ldots, k$ und $\alpha \in \{\pm 1\}$ gilt für die L-Wert Soft-Outputs

$$L(U_i|y) = \ln\left(\frac{A^i_{+1}(y)}{A^i_{-1}(y)}\right), \quad \text{für } i = 1, \ldots, k. \tag{4.6}$$

Beweis. Es gilt

$$F(u) = f(\varphi(u)) = f\begin{pmatrix} \psi(s^u_1) \\ \vdots \\ \psi(s^u_Q) \end{pmatrix} = \sum_{q=1}^{Q} \sum_{j=1}^{d} \left(y_{d(q-1)+j} - \psi_j(s^u_q)\right)^2 = \sum_{q=1}^{Q} \Delta F_q(s^u_q)$$

und somit gilt für $\alpha \in \{\pm 1\}$ und $i = 1, \ldots, k$

$$\begin{aligned} \sum_{c \in \Gamma^i(\alpha)} \exp\left(-\frac{1}{2\sigma^2} f(c)\right) &= \sum_{u \in \{\pm 1\}^k,\ u_i = \alpha} \exp\left(-\frac{1}{2\sigma^2} f(\varphi(u))\right) \\ &= \sum_{u \in \{\pm 1\}^k,\ u_i = \alpha} \exp\left(-\frac{1}{2\sigma^2} \sum_{q=1}^{Q} \Delta F_q(s^u_q)\right) \\ &= \sum_{u \in \{\pm 1\}^k,\ u_i = \alpha} \prod_{q=1}^{Q} \exp\left(-\frac{1}{2\sigma^2} \Delta F_q(s^u_q)\right) = A^i_\alpha(y). \end{aligned}$$

Mit Satz 3.29, Korollar 3.30 und (4.2) folgt dann die Aussage des Korollars.

□

Die Bedeutung des Korollars liegt in der Tatsache, daß die Berechnungsformel[5] (4.5) für die $A^i_\alpha(y)$ nur noch von $\Delta F_q(s)$ für alle Zustände $s \in S$ abhängt und die Größenordnung von S relativ gering ist im Vergleich zu 2^k (im Beispiel des SACCH-Codes ist $|S| = 2^5 = 32$).

Zur Reduktion der Berechnungskomplexität wird in den folgenden Abschnitten wie folgt vorgegangen:

[5] Wie man an (4.6) und den verschiedenen Darstellungen in Korollar 3.30 auf Seite 55 sofort sieht, können die $A^i_\alpha(y)$ in Zähler und Nenner mit einem beliebigen Faktor ungleich 0 erweitert werden. Da

$$\exp\left(-\frac{1}{2\sigma^2} \Delta F_q(s^u_q)\right) = \gamma \exp\left(\frac{1}{\sigma^2} \sum_{j=1}^{d} y_{d(q-1)+j} \psi_j(s^u_q)\right)$$

für ein $\gamma \in \mathbb{R}$, welches nicht von $\psi(s^u_q)$ abhängt, kann man alternativ auch etwa

$$\hat{A}^i_\alpha(y) = \sum_{u \in \{\pm 1\}^k,\ u_i = \alpha} \prod_{q=1}^{Q} \exp\left(\frac{1}{\sigma^2} \sum_{j=1}^{d} y_{d(q-1)+j} \psi_j(s^u_q)\right)$$

in Formel (4.6) verwenden. Allerdings hat sich die im weiteren verwendete Darstellung (4.5) als numerisch günstiger erwiesen.

- Verallgemeinerung von A_α^i durch Abbildungen $\tilde{A}_m$.
- Rekursive Darstellung von $\tilde{A}_m$ durch Abbildungen A_m, deren Werte mit einem „Von-Links-nach-Rechts“-Durchlauf eines Trellis-Diagramms[6] berechnet werden.
- Umkehrung der Rekursion durch Abbildungen B_m, deren Werte mit einem „Von-Rechts-nach-Links“-Durchlauf eines Trellis-Diagramms berechnet werden.
- Gemeinsame Berechnung aller A_α^i mittels eines weiteren Trellis-Diagramm-Durchlaufs unter Verwendung von A_m und B_m.

4.5 Komplexitätsreduktion

4.5.1 Trellis-Diagramm

Die Auswertung von (4.2) kann man als Durchlaufen eines kompletten binären Codebaums auffassen, wie er in Abbildung 4.3 dargestellt ist. Auf den Wegen zu den Blättern werden zwar viele Knoten mehrfach durchlaufen, aber die Anzahl der Blätter bestimmt die Komplexität des Gesamtbaums.

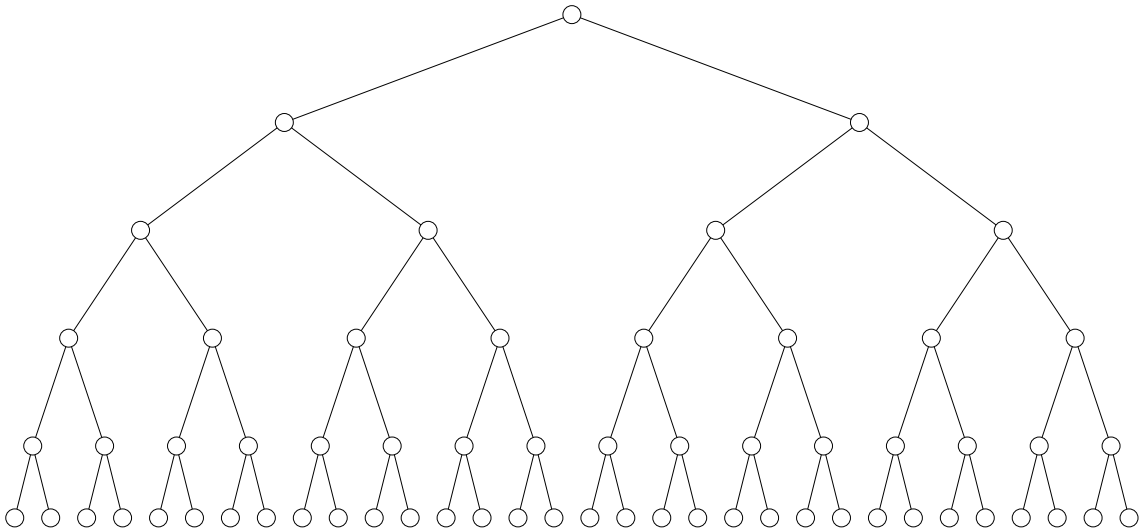

Abbildung 4.3: *Binärer Codebaum*

Nach Umformung zu (4.5) ist jetzt $\Delta F_q(s)$ für alle Zustände $s \in S$ und alle $q = 1, \ldots, Q$ auszuwerten und die Ergebnisse geeignet zu verknüpfen. Wie im weiteren zu sehen, entspricht diese Vorgehensweise dem Durchlaufen eines sogenannten Trellis-Diagramms, wie in Abbildung 4.4 zu sehen, dessen Knotenzahl im Vergleich zum gesamten Codebaum sehr gering ist. Für spezielle Anwendungen, insbesondere im Zusammenhang mit Turbo-Codes, lassen sich Trellis-Diagramme auch zu Tanner Graphen [Tan81, Wib96, Hub02] erweitern.

[6] Das Trellis-Diagramm wird in Definition 4.5 auf Seite 68 definiert.

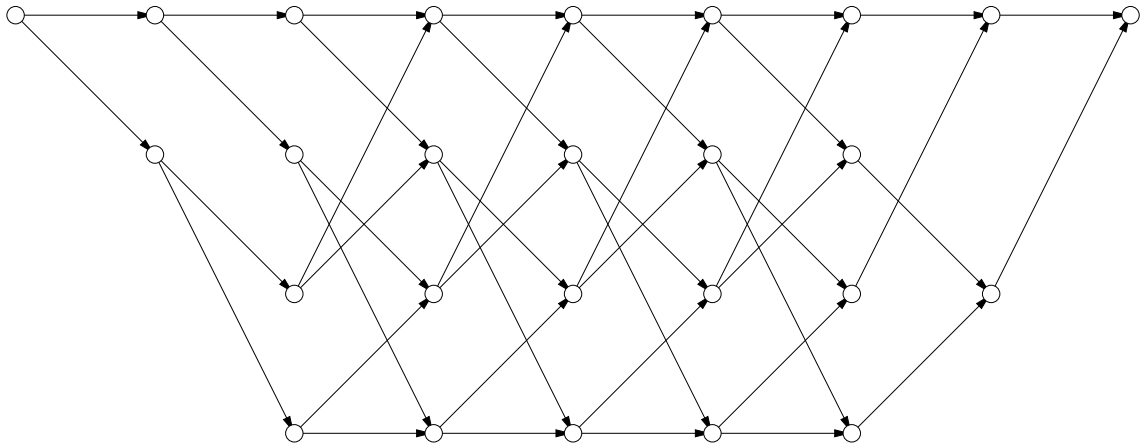

Abbildung 4.4: *Trellis-Diagramm*

Definition 4.5 (Trellis-Diagramm)
Voraussetzung 4.3 auf Seite 64 sei erfüllt. Dann heißt das Paar $(\mathcal{T}, T)$ *mit der Menge*

$$\mathcal{T} := \{(s,q);\ s \in S,\ q = 0,\ldots,Q+1\}$$

und der Zustandsübergangsfunktion[7] T *ein Trellis-Diagramm. Die Elemente* $(s,q) \in \mathcal{T}$ *werden als Knoten im Trellis-Diagramm bezeichnet. Für* $q \in \{0,\ldots,Q+1\}$ *heißt die Menge* $\{(s,q) \in \mathcal{T};\ s \in S\}$ *das q-te Trellis-Segment.*

Bei einem Knoten $(s,q) \in \mathcal{T}$ stellt s einen Zustand (des Schieberegisters) dar und q kann als dynamischer Wert angesehen werden, in manchen Arbeiten als *Zeit* bezeichnet. Mit Hilfe der Zustandsübergangsfunktion T erfolgt dann mit einem Zustandsübergangszeichen $v \in V$ ein „Weiterschalten“ vom Knoten (s,q) auf den Knoten $(T(s,v),q+1)$, dargestellt durch die Pfeile in Abbildung 4.4.

4.5.2 Allgemeine rekursive Darstellung

Zur Reduktion der Berechnungskomplexität werden zunächst die A_α^i in einer verallgemeinerten Form als Abbildungen $\tilde{A}_m$ dargestellt, die eine spätere Umformung erlaubt. Zunächst sind dazu einige Vereinbarungen nötig. Analog zu Definition 4.1 auf Seite 60 definiere rekursiv

$$\begin{aligned} s_1^u &:= T(s_0,u_1), \quad u \in V^m = V \times \ldots \times V,\ m \geq 1, \\ s_j^u &:= T(s_{j-1}^u,u_j) \quad u \in V^m,\ m \geq j > 2, \end{aligned}$$

d.h., s_j^u repräsentiert den Zustand des Schieberegisters nach j Shifts des Registers mit den Eingabezeichen $u_1,\ldots,u_j$.

Man betrachte weiterhin Mengen $V_j \subseteq V$, $j \in \mathbb{N}$, die die *zulässigen Zustandsübergangszeichen* im j-ten Schritt enthalten. Definiere die Produktmengen

$$U_m := V_1 \times \ldots \times V_m \subseteq V^m, \quad m \in \mathbb{N}, \tag{4.7}$$

d.h., U_m enthält die ersten m Komponenten der *zulässigen Eingabewörter*.

[7] Mit Voraussetzung 4.3 definiert als (4.1) auf Seite 61.

Weiter seien für $q \in \mathbb{N}$ Abbildungen

$$\mu_q : S \to \mathbb{R}$$

gegeben.

Definition 4.6 (Abbildungen $\tilde{A}_m$)
Voraussetzung 4.3 auf Seite 64 sei erfüllt. Für $m \in \mathbb{N}$ und Eingabewortmengen $U_m \subseteq V^m$ definiere Abbildungen

$$\begin{aligned} \tilde{A}_m &: \mathcal{P}(S) \to \mathbb{R}, \\ E &\mapsto \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u \in E)}} \prod_{j=1}^{m} \mu_j(s_j^u). \end{aligned} \tag{4.8}$$

Falls es keine $u \in U_m$ mit $s_m^u \in E$ gibt, so sei die Summe über eine leere Indexmenge als 0 definiert. ⌟

Somit wird in (4.8) über alle zulässigen Eingabewörter summiert, bei denen das Schieberegister nach m Shifts einen Zustand aus E erreicht. Bei geeigneter Wahl[8] von m, U_m, μ_q und E repräsentieren die Abbildungen $\tilde{A}_m$ die gesuchten Werte A_α^i.

Für eine rekursive Darstellung von $\tilde{A}_m$ werden weitere Definitionen benötigt.

Definition 4.7 (Hilfsabbildungen W, τ und $\hat{T}$)
Voraussetzung 4.3 auf Seite 64 sei erfüllt. Man definiere Abbildungen

$$\begin{aligned} W &: S \times \mathcal{P}(V) \to \mathcal{P}(S), \\ &(t, \hat{V}) \mapsto \left\{ s \in S;\ \exists \hat{v} \in \hat{V} \ni T(s, \hat{v}) = t \right\}, \end{aligned}$$

$$\begin{aligned} \tau &: S \to V, \\ s = (s^1, \ldots, s^L)^\top &\mapsto (s^{L-b+1}, \ldots, s^L)^\top, \end{aligned}$$

und

$$\begin{aligned} \hat{T} :\ & V \times S \to S, \\ & (v, s) \mapsto (v^1, \ldots, v^b, s^1, \ldots, s^{L-b})^\top. \end{aligned}$$

⌟

Die Abbildung W bildet $(t, \hat{V})$ in die Menge aller Zustände ab, die den Zustand t mit einem Übergangszeichen aus $\hat{V}$ erreichen können.

Wenn der Zustand s Ergebnis eines Zustandsübergangs ist, so war $\tau(s)$ das zugehörige Zustandsübergangszeichen, vergleiche Abbildung 4.5.

Die Abbildung $\hat{T}$ dreht die Richtung der Schieberegisteroperation um, vergleiche Abbildung 4.5.

Mit Hilfe der Abbildungen τ und $\hat{T}$ läßt sich die implizite Definition von W auch konstruktiv darstellen.

[8] Die Berechnung von A_α^i bei geeigneter Wahl von m, U_m, μ_q und E wird in Satz 4.14 auf Seite 75 dargelegt.

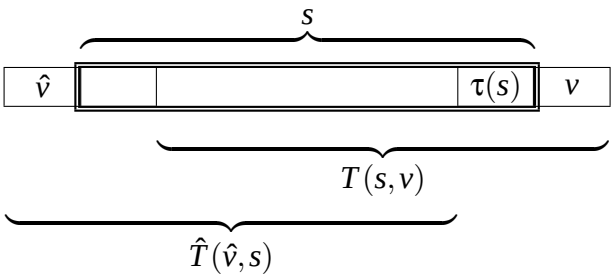

Abbildung 4.5: *Schieberegisteroperationen*

Lemma 4.8 (Konstruktive Darstellung der Abbildung W)
Voraussetzung 4.3 auf Seite 64 sei erfüllt.

(i) *Es gilt*

$$T\left(\hat{T}(v,s),\tau(s)\right) = s, \quad \text{für alle } s \in S,\ v \in V \tag{4.9}$$

(ii) *Für alle $t \in S$ und $\hat{V} \subseteq V$ gilt*

$$W(t,\hat{V}) = \begin{cases} \left\{\hat{T}(v,t);\ v \in V\right\}, & \text{falls } \tau(t) \in \hat{V}, \\ \emptyset, & \text{sonst.} \end{cases} \tag{4.10}$$

⌟

Beweis. Ad (i): Die Aussage folgt sofort über Definition 4.7.
Ad (ii): Nach Definition gilt

$$W(t,\hat{V}) = \left\{s \in S;\ \exists \hat{v} \in \hat{V} \ni T(s,\hat{v}) = t\right\}.$$

Falls $W(t,\hat{V}) \neq \emptyset$ und $T(s,\hat{v}) = t$, so gilt $\hat{v} = \tau(t)$ und $s \in \left\{\hat{T}(v,t);\ v \in V\right\}$. Damit folgt die Aussage.

□

Für die spätere Berechnung über ein Trellis-Diagramm werden jetzt Abbildungen an den Knoten dieses Diagramms definiert.

Definition 4.9 (Rekursive Abbildungen A_m)
Voraussetzung 4.3 auf Seite 64 sei erfüllt. Für $m \in \mathbb{N}_0$ definiere Abbildungen

$$A_m : S \to \mathbb{R},$$

über

$$A_0(s) := \begin{cases} 1, & \text{für } s = s_0, \\ 0, & \text{sonst} \end{cases}, \tag{4.11}$$

und für $m \in \mathbb{N}$ definiere

$$A_m(s) := \mu_m(s) \sum_{t \in W(s,V_m)} A_{m-1}(t), \tag{4.12}$$

$$= \begin{cases} \mu_m(s) \sum\limits_{v \in V} A_{m-1}\left(\hat{T}(v,s)\right), & \text{falls } \tau(s) \in V_m, \\ 0, & \text{sonst.} \end{cases} \tag{4.13}$$

⌟

Die Beziehung (4.12)=(4.13) folgt sofort mit (4.10).

Jetzt lassen sich die $\tilde{A}_m$ über A_m in einfacher Weise rekursiv auf den Knoten eines Trellis-Diagramms darstellen.

Lemma 4.10 (Vorwärtsrekursion)
Voraussetzung 4.3 auf Seite 64 sei erfüllt.

(i) *Es gilt für alle* $m \geq 2$ *und* $E \subseteq S$

$$\tilde{A}_m(E) = \sum_{s \in E} \mu_m(s) \tilde{A}_{m-1}\left(W(s, V_m)\right). \tag{4.14}$$

(ii) *Für alle* $m \in \mathbb{N}$ *gilt*

$$\tilde{A}_m(E) = \sum_{s \in E} A_m(s), \quad \textit{für } m \in \mathbb{N}. \tag{4.15}$$

Beweis. Ad (i): Es gilt für $m \geq 2$, $E \subseteq S$

$$\begin{aligned}
\tilde{A}_m(E) &= \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u \in E)}} \prod_{j=1}^{m} \mu_j(s_j^u) \\
&= \sum_{s \in E} \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u = s)}} \prod_{j=1}^{m} \mu_j(s_j^u) \\
&= \sum_{s \in E} \mu_m(s) \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u = s)}} \prod_{j=1}^{m-1} \mu_j(s_j^u) \\
&= \sum_{s \in E} \mu_m(s) \sum_{\substack{(u \in U_{m-1}) \\ \wedge (s_{m-1}^u \in W(s, V_m))}} \prod_{j=1}^{m-1} \mu_j(s_j^u) \\
&= \sum_{s \in E} \mu_m(s) \tilde{A}_{m-1}\left(W(s, V_m)\right).
\end{aligned}$$

Bei der Umformung im vorletzten Schritt ist zu beachten, daß es **genau ein** Übergangszeichen $v \in V_m$ gibt mit $T(s_{m-1}^u, v) = s$, wenn s_{m-1}^u in $W(s, V_m)$ liegt, d.h., es sind keine Vielfachheiten zu beachten.

Ad (ii): Der Beweis wird mit vollständiger Induktion nach m geführt.

$\underline{m=1}$: Unter Verwendung des Kronecker-Symbols[9] gilt

$$\begin{aligned}\tilde{A}_1(E) &= \sum_{\substack{(u\in U_1)\\ \wedge(s_1^u\in E)}} \mu_1(s_1^u) = \sum_{s\in E} \mu_1(s)\delta_{s_0\in W(s,V_1)} \\ &= \sum_{s\in E} \underbrace{\mu_1(s) \sum_{t\in W(s,V_1)} \underbrace{\delta_{t=s_0}}_{=A_0(t)}}_{=A_1(s)} = \sum_{s\in E} A_1(s)\end{aligned}$$

$\underline{m-1\mapsto m}$: Nach Induktionsannahme gilt insbesondere

$$\tilde{A}_{m-1}(W(s,V_m)) = \sum_{t\in W(s,V_m)} A_{m-1}(t)$$

und somit folgt mit (i) und (4.12)

$$\begin{aligned}\tilde{A}_m(E) &= \sum_{s\in E} \mu_m(s)\tilde{A}_{m-1}(W(s,V_m)) \\ &= \sum_{s\in E} \mu_m(s) \sum_{t\in W(s,V_m)} A_{m-1}(t) \\ &= \sum_{s\in E} A_m(s).\end{aligned}$$

□

Nun ist es möglich, mit Hilfe der hergeleiteten Rekursionsvorschrift die $\tilde{A}_m(E)$ mittels $A_m(s)$ auf den Knoten (s,m) eines Trellis-Diagramms zu berechnen.

Es ist noch zu bemerken, daß in diesem Abschnitt keinerlei Einschränkungen an die Menge V der Zustandsübergangszeichen oder an die Mengen $V_j \in \mathcal{P}(V)$ gemacht wurden.

4.5.3 Rekursionsumkehrung

Im vorangegangenen Abschnitt wurde die Berechnungsvorschrift für die $\tilde{A}_m(E)$ bereits in eine rekursive Darstellung auf den Knoten eines Trellis-Diagramms umgeformt. Dabei durchläuft die Rekursionsvorschrift (4.12) beziehungsweise (4.13) das Trellisdiagramm „von links nach rechts“, das heißt, mit inkrementellem dynamischen Parameter. Jetzt wird eine Rekursionsformel für die umgekehrte Richtung „von rechts nach links“ konstruiert, mit deren Hilfe die Berechnungskomplexität nochmals reduziert werden kann.

Im weiteren sei dazu

$$T(t,\hat{V}) := \left\{T(t,\hat{v});\ \hat{v}\in\hat{V}\right\}, \quad \text{für } t\in S,\ \hat{V}\subseteq V.$$

vereinbart.

[9] Die hier verwendeten Kronecker-Symbole bedeuten:

$$\delta_{t=s_0} = \begin{cases}1, & t=s_0,\\ 0, & t\neq s_0,\end{cases} \qquad \delta_{s_0\in W(s,V_1)} = \begin{cases}1, & s_0\in W(s,V_1),\\ 0, & s_0\notin W(s,V_1),\end{cases}$$

und es gilt sofort

$$\delta_{s_0\in W(s,V_1)} = \sum_{t\in W(s,V_1)} \delta_{t=s_0}.$$

Definition 4.11 (Rekursive Abbildungen B_m)
Voraussetzung 4.3 auf Seite 64 sei erfüllt. Für $0 \leq m \leq Q$ definiere Abbildungen

$$B_m : S \to \mathbb{R},$$

über

$$B_0(s) := \begin{cases} 1, & \text{für } s = s_0, \\ 0, & \text{sonst} \end{cases}, \tag{4.16}$$

und für $1 \leq m \leq Q$ definiere

$$B_m(s) := \mu_{Q-m+1}(s) \sum_{t \in T(s, V_{Q-m+2})} B_{m-1}(t). \tag{4.17}$$

Somit sind die B_m rekursiv auf dem Trellis-Diagramm mit dekrementellem dynamischen Parameter definiert. Nun können Beziehungen zwischen A_m und B_m angegeben werden, die die Berechnung der Werte von $\tilde{A}_Q$ „von zwei Seiten kommend“ gestattet.

Lemma 4.12 (Doppelrekursion)
Voraussetzung 4.3 auf Seite 64 sei erfüllt.

(i) *Für alle $m \in \{1, \ldots, Q\}$ und alle $j \in \{1, \ldots, m+1\}$ gilt*

$$\sum_{s \in S} A_m(s) \sum_{t \in T(s, V_{m+1})} B_{Q-m}(t) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s, V_j)} B_{Q-j+1}(t). \tag{4.18}$$

(ii) *Mit $V_{Q+1} := \{v_0\}$ gilt für alle $j \in \{1, \ldots, Q+1\}$*

$$\tilde{A}_Q(W(s_0, \{v_0\})) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s, V_j)} B_{Q-j+1}(t). \tag{4.19}$$

Beweis. Ad (i): Für alle $m \in \{1, \ldots, Q\}$ gilt mit (4.12) und (4.17)

$$\begin{aligned}
\sum_{s \in S} A_m(s) \sum_{t \in T(s, V_{m+1})} B_{Q-m}(t) &= \sum_{s \in S} \mu_m(s) \sum_{\hat{t} \in W(s, V_m)} A_{m-1}(\hat{t}) \sum_{t \in T(s, V_{m+1})} B_{Q-m}(t) \\
&= \sum_{\hat{t} \in S} \sum_{s \in T(\hat{t}, V_m)} \mu_m(s) A_{m-1}(\hat{t}) \sum_{t \in T(s, V_{m+1})} B_{Q-m}(t) \\
&= \sum_{\hat{t} \in S} A_{m-1}(\hat{t}) \sum_{s \in T(\hat{t}, V_m)} \mu_m(s) \sum_{t \in T(s, V_{m+1})} B_{Q-m}(t) \\
&= \sum_{\hat{t} \in S} A_{m-1}(\hat{t}) \sum_{s \in T(\hat{t}, V_m)} B_{Q-m+1}(s)
\end{aligned}$$

also

$$\sum_{s \in S} A_m(s) \sum_{t \in T(s, V_{m+1})} B_{Q-m}(t) = \sum_{s \in S} A_{m-1}(s) \sum_{t \in T(s, V_m)} B_{Q-m+1}(t). \tag{4.20}$$

Durch mehrfache Anwendung der Gleichung (4.20) ergibt sich für ein beliebiges $j \in \{1, \ldots, m+1\}$ die Aussage (4.18).

Ad (ii): Mit (4.15), (4.16) und (4.18) läßt sich $\tilde{A}_Q(W(s_0,\{v_0\}))$ für $V_{Q+1}=\{v_0\}$ und mit einem beliebigen $j\in\{1,\ldots,Q+1\}$ wie folgt darstellen

$$\begin{aligned}\tilde{A}_Q(W(s_0,\{v_0\})) &= \sum_{s\in W(s_0,\{v_0\})} A_Q(s)\\ &= \sum_{s\in S} A_Q(s)\delta_{T(s,v_0)=s_0}\\ &= \sum_{s\in S} A_Q(s) \sum_{t=T(s,v_0)} \delta_{t=s_0}\\ &= \sum_{s\in S} A_Q(s) \sum_{t\in T(s,\{v_0\})} B_0(t)\\ &= \sum_{s\in S} A_Q(s) \sum_{t\in T(s,V_{Q+1})} B_0(t)\\ &= \sum_{s\in S} A_{j-1}(s) \sum_{t\in T(s,V_j)} B_{Q-j+1}(t).\end{aligned}$$

□

Es ist wichtig zu bemerken, daß bei der Auswertung von (4.19) die Menge V_j nicht in die Berechnung der benötigten A_m und B_m eingeht.

4.5.4 Berechnung von A^i_α

Mit den Vorarbeiten aus den vorangegangenen Abschnitten läßt sich A^i_α nun auf einfache Weise berechnen. Dazu wird jetzt eine spezielle Wahl der Mengen V_m und der Funktionen μ_q getroffen.

Definition 4.13 (Spezielle Wahl von V_m und μ_q)
Voraussetzung 4.3 auf Seite 64 sei erfüllt und $y\in\mathbb{R}^n$ sei eine beliebige aber fest gewählte Realisierung der Zufallsvariable Y. Definiere

$$\begin{aligned}V_m &:= V, && \text{für } m\in\{1,\ldots,a\},\\ V_m &:= \{v_0\}, && \text{für } m\in\{a+1,\ldots,Q+1\},\\ U_m &:= V_1\times\ldots\times V_m, && \text{für } m\in\{1,\ldots,Q+1\},\end{aligned}$$

Man betrachte zu einer beliebigen aber festen Wahl von $i\in\{1,\ldots,k\}$ das eindeutige $j\in\{1,\ldots,a\}$ und das eindeutige $\hat{i}\in\{1,\ldots,b\}$ mit

$$i=(j-1)\cdot b+\hat{i}.$$

Man definiere für eine beliebige aber feste Wahl von $\alpha\in\{\pm1\}$

$$\begin{aligned}V^i_j(\alpha) &:= \{v\in V;\ v_{\hat{i}}=\alpha\}\\ U^i_Q(\alpha) &:= V_1\times\ldots\times V_{j-1}\times V^i_j(\alpha)\times V_{j+1}\times\ldots\times V_Q\subset U_Q,\end{aligned}\tag{4.21}$$

und definiere für $q\in\{1,\ldots,Q\}$ die Abbildungen

$$\begin{aligned}\mu_q &: S\to\mathbb{R},\\ &s\mapsto \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right).\end{aligned}\tag{4.22}$$

⌟

Mit dieser speziellen Wahl der Mengen und Funktionen aus den vorangegangenen Abschnitten läßt sich jetzt A_α^i einfach berechnen.

Satz 4.14 (Berechnung von A_α^i)
Voraussetzung 4.3 auf Seite 64 sei erfüllt und man verwende die Begriffe aus Definition 4.13, wobei $i \in \{1,\dots,k\}$ beliebig aber fest gewählt sei.

(i) *Es gilt*

$$\varphi\left(\{\pm 1\}^k\right) = \left\{ \begin{pmatrix} \psi(s_1^u) \\ \vdots \\ \psi(s_Q^u) \end{pmatrix} ;\ u \in U_Q \right\}, \quad \Gamma^i(\alpha) = \left\{ \begin{pmatrix} \psi(s_1^u) \\ \vdots \\ \psi(s_Q^u) \end{pmatrix} ;\ u \in U_Q^i(\alpha) \right\}. \tag{4.23}$$

(ii) *Für alle $u \in U_Q$ gilt*

$$s_Q^u \in W(s_0, \{v_0\}). \tag{4.24}$$

(iii) *Für alle $(s,q) \in S \times \{0,\dots,Q\}$ seien $A_q(s)$ und $B_q(s)$ mit den Mengen $V_1,\dots,V_Q$ nach Definition 4.9 auf Seite 70 und Definition 4.11 auf Seite 73 definiert. Dann gilt*

$$A_\alpha^i(y) = \sum_{s\in S} A_{j-1}(s) \sum_{t \in T(s, V_j^i(\alpha))} B_{Q-j+1}(t). \tag{4.25}$$

Beweis. Ad (i): Nach Definition 4.1 auf Seite 60 und der Definition von

$$U_Q = \underbrace{V \times \ldots \times V}_{a\text{-mal}} \times \underbrace{\{v_0\} \times \ldots \times \{v_0\}}_{(l-1)\text{-mal}}$$

folgt sofort

$$\varphi\left(\{\pm 1\}^k\right) = \left\{ \begin{pmatrix} \psi(s_1^u) \\ \vdots \\ \psi(s_Q^u) \end{pmatrix} ;\ u \in U_Q \right\}.$$

Weiter gilt

$$\begin{aligned} \Gamma^i(\alpha) &= \left\{ \varphi(u);\ u \in \{\pm 1\}^k,\ u_i = \alpha \right\} = \left\{ \begin{pmatrix} \psi(s_1^u) \\ \vdots \\ \psi(s_Q^u) \end{pmatrix} ;\ u \in U_Q,\ u_i = \alpha \right\} \\ &= \left\{ \begin{pmatrix} \psi(s_1^u) \\ \vdots \\ \psi(s_Q^u) \end{pmatrix} ;\ u \in U_Q^i(\alpha) \right\}. \end{aligned}$$

Ad (ii): Nach Definition 4.13 gilt für alle s_Q^u mit $u \in U_Q$

$$s_{Q+1}^u = T(s_Q^u, u_{Q+1}) = s_0, \quad u_{Q+1} \in V_{Q+1} = \{v_0\},$$

also

$$s_Q^u \in W(s_0, V_{Q+1}) = W(s_0, \{v_0\}).$$

Ad (iii): Unter Verwendung von (4.5), (4.8), (4.19), (4.21), (4.22), (4.23) und (4.24) gilt

$$\begin{aligned}
A^i_\alpha(y) &= \sum_{u\in\{\pm1\}^k,\ u_i=\alpha} \prod_{q=1}^{Q} \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s^u_q)\right) \\
&= \sum_{u\in U^i_Q(\alpha)} \prod_{q=1}^{Q} \mu_q(s^u_q) \\
&= \sum_{\substack{(u\in U^i_Q(\alpha))\\ \wedge(s^u_Q\in W(s_0,V_{Q+1}))}} \prod_{q=1}^{Q} \mu_q(s^u_q) \\
&= \tilde{A}_Q(W(s_0,V_{Q+1})) \\
&= \sum_{s\in S} A_{j-1}(s) \sum_{t\in T(s,V^i_j(\alpha))} B_{Q-j+1}(t)
\end{aligned}$$

Formal war $\tilde{A}_Q(W(s_0,V_{Q+1}))$ über das Hilfskonstrukt $U^i_Q(\alpha)$ bestimmt, welches in der resultierenden Darstellung aber nicht mehr benötigt wird, da dort der bestimmende Term $V^i_j(\alpha)$ explizit verwendet wird. Somit lassen sich $A_q(s)$ und $B_q(s)$ mit den Mengen $V_1,\ldots,V_Q$ unabhängig von i und α bestimmen.

□

Die Unabhängigkeit der $A_q(s)$ und $B_q(s)$ von i und α bedeutet, daß diese Werte nur *einmal* bestimmt werden müssen, um alle $A^i_\alpha(y)$, $1\le i\le k$, $\alpha\in\{\pm1\}$, zu berechnen.

Zusammenfassung der Vorgehensweise:

- Definiere

$$\begin{aligned}
V_j &:= V, && \text{für } j\in\{1,\ldots,a\},\\
V_j &:= \{v_0\}, && \text{für } j\in\{a+1,\ldots,Q+1\},\\
V^i_j(\alpha) &:= \{v\in V;\ v_{\hat{i}}=\alpha\}, && \text{für } i=(j-1)\cdot b+\hat{i},\ \hat{i}\in\{1,\ldots,b\},\ j\in\{1,\ldots,a\},\ \alpha\in\{\pm1\}.
\end{aligned}$$

- Zu einer beliebigen aber festen Wahl von $y\in\mathbb{R}^n$ definiere für $q\in\{1,\ldots,Q\}$

$$\begin{aligned}
\mu_q : S &\to \mathbb{R},\\
s &\mapsto \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right).
\end{aligned}$$

- Man berechne

$$\begin{aligned}
&A_m(s), && \text{für } s\in S,\ m\in\{1,\ldots,a-1\},\\
&B_m(s), && \text{für } s\in S,\ m\in\{1,\ldots,Q\},
\end{aligned}$$

  mit den Rekursionsformeln (4.13) und (4.17) und den Startwerten $A_0(s)$, $B_0(s)$ mit (4.11) und (4.16).

- Man berechne alle $A^i_\alpha(y)$, $i \in \{1,\dots,k\}$, $\alpha \in \{\pm 1\}$ über (4.25), also

$$A^i_\alpha(y) = \sum_{s\in S} A_{j-1}(s) \sum_{t\in T(s,V^i_j(\alpha))} B_{Q-j+1}(t).$$

  und bestimme die Soft-Outputs

$$L(U_i|y) = \ln\left(\frac{A^i_{+1}(y)}{A^i_{-1}(y)}\right), \quad i = 1,\dots,k.$$

Zusammen mit der Rekursionsformel aus dem vorangegangenen Abschnitt können alle $A^i_\alpha(y)$ jetzt gemeinsam mit $O(L\cdot Q)$ Operationen statt $O(k2^k)$ Operationen berechnet werden.

Erinnerung: $L = l\cdot b$, $Q = a+l-1$, $k = a\cdot b$, wobei k die Anzahl der Infobits ist.

Die numerische Komplexität zur Berechnung der Soft-Outputs ist also von exponentieller Ordnung auf lineare Ordnung verringert worden, wobei die Anzahl der Infobits k die entscheidende Größe ist.

4.5.5 Spezialfall: Binärer Zustandsübergang

Im wichtigen Spezialfall $b = 1$ besteht die Menge V der Zustandsübergangszeichen nur aus den beiden Elementen $+1, -1$. Die GSM-Codes gehören etwa zu diesem weit verbreiteten Spezialfall.

Da in der obigen Beschreibung jetzt $i = j$ und $V^i_j(\alpha) = \{\alpha\}$, vereinfacht sich die Vorgehensweise noch einmal wie folgt:

Vorgehensweise für $b = 1$:

- Definiere

$$\begin{aligned} V_m &:= \{\pm 1\}, \quad \text{für } m \in \{1,\dots,a\},\\ V_m &:= \{+1\}, \quad \text{für } m \in \{a+1,\dots,Q+1\} \end{aligned}$$

- Zu einer beliebigen aber festen Wahl von $y \in \mathbb{R}^n$ definiere für $q \in \{1,\dots,Q\}$

$$\begin{aligned} \mu_q : S &\to \mathbb{R},\\ s &\mapsto \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right). \end{aligned}$$

- Man berechne

$$\begin{aligned} &A_m(s), \quad \text{für } s\in S,\ m \in \{1,\dots,a-1\},\\ &B_m(s), \quad \text{für } s\in S,\ m \in \{1,\dots,Q\}, \end{aligned}$$

  mit den Rekursionsformeln (4.13) und (4.17) und den Startwerten $A_0(s)$, $B_0(s)$ mit (4.11) und (4.16).

- Man berechne alle $A^i_\alpha(y)$, $i \in \{1,\ldots,k\}$, $\alpha \in \{\pm 1\}$ über

$$A^i_\alpha(y) = \sum_{s\in S} A_{i-1}(s) B_{Q-i+1}(T(s,\alpha)). \tag{4.26}$$

  und bestimme die Soft-Outputs

$$L(U_i|y) = \ln\left(\frac{A^i_{+1}(y)}{A^i_{-1}(y)}\right), \quad i = 1,\ldots,k.$$

4.6 Algorithmische Umsetzung (TSO Verfahren)

4.6.1 Vorbereitung

Man betrachte für die algorithmische Umsetzung die Menge

$$\mathcal{T} = \{(s,q);\ s \in S,\ q = 0,\ldots,Q+1\}$$

des Trellis-Diagramms und die Abbildungen

$\mu:\ \mathcal{T} \to \mathbb{R}$,
$(s,q) \mapsto \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right)$ „Multiplikatoren im Zustand s des Trellis-Segments q“.
$A:\ \mathcal{T} \to \mathbb{R}$,
$(s,q) \mapsto A(s,q)$ „Teilsummen ’A’ im Zustand s des Trellis-Segments q“,
$B:\ \mathcal{T} \to \mathbb{R}$,
$(s,q) \mapsto B(s,q)$ „Teilsummen ’B’ im Zustand s des Trellis-Segments $Q-q+1$“.

Die Abbildungen werden nur auf den sinnvollen Teilmengen des Definitionsbereiches ausgewertet.

Berechnung von μ: **Eingang** σ^2, y; **Ausgang** μ;

für $q = 1,\ldots,Q$: *Fortschreiten im Trellis-Diagramm*
  für $s \in S$: *Betrachtung aller Zustände*
    $\mu(s,q) := \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right)$;

Algorithmus 4.2: *Berechnung der Multiplikatoren μ*

In einem vorbereitenden Schritt können die benötigten Multiplikatoren $\mu(s,q)$ mit Algorithmus 4.2 berechnet werden. Da der Wert von $\Delta F_q(s)$ nur mittelbar vom Zustand s abhängt und direkt mit $\psi(s)$ gebildet wird, gilt

$$\left|\{\Delta F_q(s);\ s \in S\}\right| \le \min\left\{2^L, 2^d\right\},$$

d.h., für $d < L$ haben viele der obigen $\mu(s,q)$ den gleichen Wert. Abhängig vom speziellen Code läßt sich $\mu(s,q)$ in der Implementierung also mit weitaus weniger Operationen bestimmen.

4.6.2 Allgemeiner Fall

Die Berechnung der L-Wert Soft-Outputs für den allgemeinen Fall von terminierten (n,k)-Faltungscodes läßt sich in kompakter Form als Algorithmus 4.3 formulieren. Diesen Algorithmus (beziehungsweise den binären Spezialfall Algorithmus 4.4) werden wir auch kurz als TSO Verfahren (**T**rellis-**S**oft-**O**utput Verfahren) zitieren.

TSO Allgemein: **Eingang** μ; **Ausgang** $L(U_\bullet|y)$;

für $s \in S$: *Vorbelegung*
  $A(s,0) := 0;\ B(s,0) := 0;$
$A(s_0,0) := 1;\ B(s_0,0) := 1;$ *Startzustand*
für $q = 1,\ldots,l$: *Terminierung*
  für $s \in S$: *Betrachtung aller Zustände*
    $s^+ := T(s,v_0);$ *Nachfolgerzustand*
    $B(s,q) := \mu(s,Q-(q-1))B(s^+,q-1);$ *Berechnung von B*
für $q = 1,\ldots,a-1$: *Fortschreiten im Trellis-Diagramm*
  für $s \in S$: *Betrachtung aller Zustände*
    $A(s,q) := A(\hat{T}(v_0,s),q-1);$ *Vorbelegung von A*
    $B(s,l+q) := B(T(s,v_0),l-1+q);$ *Vorbelegung von B*
    für $v \in V \setminus \{v_0\}$: *Betrachtung aller Übergänge*
      $A(s,q) := A(s,q) + A(\hat{T}(v,s),q-1);$ *Berechnung von A*
      $B(s,l+q) := B(s,l+q) + B(T(s,v_0),l-1+q);$ *Berechnung von B*
    $A(s,q) := \mu(s,q) \cdot A(s,q);$ *Berechnung von A*
    $B(s,l+q) := \mu(s,a-q) \cdot B(s,l+q);$ *Berechnung von B*
für $i = 1,\ldots,k$: *Fortschreiten im Trellis-Diagramm*
  $A^i_{+1} := 0;\ A^i_{-1} := 0;$ *Vorbelegung*
  $j = 1 + \lfloor (i-1)/b \rfloor;$
  für $s \in S$: *Betrachtung aller Zustände*
    für $v \in V^i_j(+1)$: *Übergänge*
      $A^i_{+1} := A^i_{+1} + A(s,j-1) \cdot B(T(s,v),Q-j+1);$ *Update von* A^i_{+1}
    für $v \in V^i_j(-1)$: *Übergänge*
      $A^i_{-1} := A^i_{-1} + A(s,j-1) \cdot B(T(s,v),Q-j+1);$ *Update von* A^i_{-1}
  $L(U_i|y) := \ln(A^i_{+1}/A^i_{-1});$ *i-ter Soft-Output*

Algorithmus 4.3: *Berechnung der L-Wert Soft-Outputs für allgemeine terminierte Faltungscodes*

Bei geeigneter Implementierungsdarstellung von V bzw. $V^i_j(\alpha)$, etwa als Teilmengen von $\mathbb{N}$, lassen sich die Iterationen $v \in V$ und $s \in S$ von Algorithmus 4.3 als gewöhnliche Programmschleifen implementieren. Vorkommende Indizes wie etwa $l-1+q$ werden bei der Implementierung natürlich

nur einmal berechnet und nicht bei jedem Auftreten, wie es hier zur besseren Übersicht aufgeschrieben ist.

4.6.3 Spezialfall: Binärer Zustandsübergang

Im wichtigen Spezialfall eines binären Zustandsübergangs, also $b = 1$ und $V = \{\pm 1\}$ vereinfacht sich die Implementierung zu Algorithmus 4.4.

TSO Binär: **Eingang** μ; **Ausgang** $L(U_\bullet|y)$;

für $s \in S$: *Vorbelegung*
  $A(s,0) := 0;\ B(s,0) := 0;$
$A(s_0,0) := 1;\ B(s_0,0) := 1;$ *Startzustand*
für $q = 1,\ldots,l$: *Terminierung*
  für $s \in S$: *Betrachtung aller Zustände*
    $s^+ := T(s,+1);$ *Nachfolgerzustand*
    $B(s,q) := \mu(s, Q-(q-1))B(s^+, q-1);$ *Berechnung von B*
für $q = 1,\ldots,a-1$: *Fortschreiten im Trellis-Diagramm*
  für $s \in S$: *Betrachtung aller Zustände*
    $t^+ := \hat{T}(+1,s);\ t^- := \hat{T}(-1,s);$ *Vorgängerzustände*
    $s^+ := T(s,+1);\ s^- := T(s,-1);$ *Nachfolgerzustände*
    $A(s,q) := \mu(s,q) \cdot (A(t^+, q-1) + A(t^-, q-1));$ *Berechnung von A*
    $B(s, l+q) := \mu(s, a-q) \cdot (B(s^+, l-1+q) + B(s^-, l-1+q));$ *Berechnung von B*
für $i = 1,\ldots,k$: *Fortschreiten im Trellis-Diagramm*
  $A^i_{+1} := 0;\ A^i_{-1} := 0;$ *Vorbelegung*
  für $s \in S$: *Betrachtung aller Zustände*
    $s^+ := T(s,+1);\ s^- := T(s,-1);$ *Nachfolgerzustände*
    $A^i_{+1} := A^i_{+1} + A(s,i-1) \cdot B(s^+, Q-i+1);$ *Update von* A^i_{+1}
    $A^i_{-1} := A^i_{-1} + A(s,i-1) \cdot B(s^-, Q-i+1);$ *Update von* A^i_{-1}
  $L(U_i|y) := \ln(A^i_{+1}/A^i_{-1});$ *i-ter Soft-Output*

Algorithmus 4.4: *Berechnung der L-Wert Soft-Outputs bei binärem Zustandsübergang*

4.7 Erwartungswerte der A^i_α und der einseitigen L-Werten

Abhängig von der verwendeten Codierung und der Kanalstörung können die mit (4.25) berechneten Werte $A^i_\alpha(y)$ numerisch sehr groß werden. Da bei der Bestimmung von $L(U_i|y)$ die Werte $A^i_{+1}(y)$ und $A^i_{-1}(y)$ mit einem beliebigen gemeinsamen Skalar multipliziert sein können, sucht man aus numerischen Gründen nach einem Skalierungsfaktor $\kappa(\varphi,\sigma^2)$, mit dem die Gewichte $\mu_q(s)$ in (4.22) so multipliziert werden, daß ein „numerisches Wohlverhalten“ der $A^i_\alpha(y)$ erwartet werden kann.

Daher bestimmen wir zunächst den Erwartungswert von $A^i_\alpha(Y)$, wobei Y wieder die Zufallsvariable der Kanalausgabe ist.

Lemma 4.15 (Flip eines uncodierten Bits)
Sei (n,k,φ) ein binärer linearer (n,k)-Blockcode mit charakterisierenden Mengen $J_1,\dots,J_n$. Man betrachte für $i\in\{1,\dots,k\}$ die Abbildungen

$$\begin{aligned}\gamma^i:\{\pm1\}^k &\to \{\pm1\}^k,\\ u &\mapsto \begin{cases}(u_1,\dots,u_{i-1},+1,u_{i+1},\dots,u_k)^\top, & \text{für } u_i=-1,\\ (u_1,\dots,u_{i-1},-1,u_{i+1},\dots,u_k)^\top, & \text{für } u_i=+1,\end{cases}\\ \delta^i:\mathbb{R}^n &\to \mathbb{R}^n,\\ x &\mapsto \delta^i(x),\ \text{mit } \begin{cases}\delta^i_j(x)=-x_j, & \text{für } i\in J_j,\\ \delta^i_j(x)=x_j, & \text{für } i\notin J_j.\end{cases}\end{aligned}$$

(i) *Es gilt für alle $u\in\{\pm1\}^k$ und alle $i\in\{1,\dots,k\}$, $j\in\{1,\dots,n\}$*

$$\begin{aligned}\varphi_j(u)&=-\varphi_j(\gamma^i(u)), && \text{für } i\in J_j,\\ \varphi_j(u)&=\varphi_j(\gamma^i(u)), && \text{für } i\notin J_j.\end{aligned}$$

also

$$\varphi(\gamma^i(u))=\delta^i(\varphi(u)),\quad \delta^i(\varphi(\gamma^i(u)))=\varphi(u).$$

(ii) *Es gilt für alle $u,v\in\{\pm1\}^k$ und alle $i\in\{1,\dots,k\}$*

$$\|\varphi(u)-\varphi(v)\|_2=\left\|\varphi(\gamma^i(u))-\varphi(\gamma^i(v))\right\|_2$$

Beweis. Ad (i): Aus

$$\varphi_j(u)=\prod_{i\in J_j}u_i$$

folgt die Aussage.

Ad (ii): Da nach (i)

$$\begin{aligned}\varphi_j(u)-\varphi_j(v)&=-\varphi_j(\gamma^i(u))+\varphi_j(\gamma^i(v)), && \text{für } i\in J_j,\\ \varphi_j(u)-\varphi_j(v)&=\varphi_j(\gamma^i(u))-\varphi_j(\gamma^i(v)), && \text{für } i\notin J_j.\end{aligned}$$

folgt auch diese Aussage sofort.

□

Lemma 4.16 (Erwartungswert von $A^i_\alpha(Y)$)
Voraussetzung 4.3 auf Seite 64 sei erfüllt und es sei[10]

$$A^i_\alpha(y) := \sum_{u\in\{\pm1\}^k,\ u_i=\alpha} \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-y\right\|_2^2\right) = \sum_{u\in\{\pm1\}^k,\ u_i=\alpha} \prod_{q=1}^{Q} \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s^u_q)\right),$$

für $i=1,\ldots,k$ und $\alpha\in\{\pm1\}$.

Dann gilt

(i)

$$\begin{aligned}
\mathbf{E}\left(A^i_\alpha(Y)\right) &= \frac{1}{2^{k+1}\sqrt{2^n}} \sum_{u,v\in\{\pm1\}^k} \exp\left(-\frac{1}{4\sigma^2}\left\|\varphi(u)-\varphi(v)\right\|_2^2\right) \\
&= 2^{-\left(\frac{n}{2}+k+1\right)} \sum_{u,v\in\{\pm1\}^k} \exp\left(-\frac{1}{\sigma^2} d_{\text{ham}}\left(\varphi(u),\varphi(v)\right)\right) \\
&= \frac{1}{2^{\frac{n}{2}+k+1}} \sum_{u,v\in\{\pm1\}^k} \prod_{q=1}^{Q} \exp\left(-\frac{1}{4\sigma^2}\left\|\psi(s^u_q)-\psi(s^v_q)\right\|_2^2\right) \\
&= \frac{1}{2^{\frac{n}{2}+k+1}} \sum_{u,v\in\{\pm1\}^k} \prod_{q=1}^{Q} \exp\left(-\frac{1}{\sigma^2} d_{\text{ham}}\left(\psi(s^u_q),\psi(s^v_q)\right)\right).
\end{aligned}$$

Insbesondere ist $\mathbf{E}\left(A^i_\alpha(Y)\right)$ unabhängig von i und α.

(ii) *Mit $c_0 := (+1,\ldots,+1)^\top \in \{\pm1\}^n$ und $d_0 := (+1,\ldots,+1)^\top \in \{\pm1\}^d$ gilt*

$$\begin{aligned}
\mathbf{E}\left(A^i_\alpha(Y)\right) &= 2^{-\left(\frac{n}{2}+1\right)} \sum_{u\in\{\pm1\}^k} \exp\left(-\frac{1}{\sigma^2} d_{\text{ham}}\left(\varphi(u),c_0\right)\right) \\
&= \frac{1}{2^{\frac{n}{2}+1}} \sum_{u\in\{\pm1\}^k} \prod_{q=1}^{Q} \exp\left(-\frac{1}{\sigma^2} d_{\text{ham}}\left(\psi(s^u_q),d_0\right)\right).
\end{aligned}$$

(iii)

$$2^{-\left(\frac{n}{2}+1\right)} \le \mathbf{E}\left(A^i_\alpha(Y)\right) \le 2^{-\left(\frac{n}{2}+1\right)} \cdot \left(1+(2^k-1)\exp\left(-\frac{1}{\sigma^2} d_{\text{ham}}\left(\varphi\right)\right)\right)$$

[10] siehe Korollar 4.4 auf Seite 65

Beweis. Ad (i): Nach Voraussetzung ist $Y = \varphi(U) + Z$, wobei U gleichverteilt ist und $Z \sim \mathcal{N}(0, \sigma^2 I_n)$. Damit folgt

$$\begin{aligned}
\mathbf{E}\left(A_\alpha^i(Y)\right) &= \mathbf{E}\left(\sum_{u\in\{\pm1\}^k,\ u_i=\alpha} \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-Y\right\|_2^2\right)\right) \\
&= \mathbf{E}\left(\sum_{u\in\{\pm1\}^k,\ u_i=\alpha} \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-\varphi(U)-Z\right\|_2^2\right)\right) \\
&= \sum_{v\in\{\pm1\}^k} \frac{1}{2^k} \int\limits_{\mathbb{R}^n} \sum_{u\in\{\pm1\}^k,\ u_i=\alpha} \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-\varphi(v)-z\right\|_2^2\right) \\
&\qquad \cdot \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp\left(-\frac{1}{2\sigma^2}\left\|z\right\|_2^2\right) dz \\
&= \frac{1}{2^k} \sum_{v\in\{\pm1\}^k} \sum_{u\in\{\pm1\}^k,\ u_i=\alpha} \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \int\limits_{\mathbb{R}^n} \exp\left(-\frac{1}{2\sigma^2}\left(\left\|z+\varphi(v)-\varphi(u)\right\|_2^2+\left\|z\right\|_2^2\right)\right) dz \\
&= \frac{1}{2^k\sqrt{2^n}} \sum_{v\in\{\pm1\}^k} \sum_{u\in\{\pm1\}^k,\ u_i=\alpha} \frac{1}{\sqrt{(2\pi\frac{\sigma^2}{2})^n}} \\
&\qquad \cdot \int\limits_{\mathbb{R}^n} \exp\left(-\frac{1}{2\frac{\sigma^2}{2}}\left(\left\|z\right\|^2+\left(\varphi(v)-\varphi(u)\right)^\top z+\frac{1}{2}\left\|\varphi(v)-\varphi(u)\right\|^2\right)\right) dz \\
&= \frac{1}{2^k\sqrt{2^n}} \sum_{v\in\{\pm1\}^k} \sum_{u\in\{\pm1\}^k,\ u_i=\alpha} \frac{1}{\sqrt{(2\pi\frac{\sigma^2}{2})^n}} \\
&\qquad \cdot \int\limits_{\mathbb{R}^n} \exp\left(-\frac{1}{2\frac{\sigma^2}{2}}\left(\left\|z-\frac{\varphi(u)-\varphi(v)}{2}\right\|^2+\frac{1}{4}\left\|\varphi(u)-\varphi(v)\right\|^2\right)\right) dz \\
&= \frac{1}{2^k\sqrt{2^n}} \sum_{v\in\{\pm1\}^k} \sum_{u\in\{\pm1\}^k,\ u_i=\alpha} \exp\left(-\frac{1}{4\sigma^2}\left\|\varphi(u)-\varphi(v)\right\|^2\right) \\
&= \frac{1}{2^k\sqrt{2^n}} \sum_{v\in\{\pm1\}^k} \sum_{u\in\{\pm1\}^k,\ u_i=\alpha} \exp\left(-\frac{1}{4\sigma^2}\left\|\varphi(\gamma^i(u))-\varphi(\gamma^i(v))\right\|^2\right) \\
&= \frac{1}{2^k\sqrt{2^n}} \sum_{v\in\{\pm1\}^k} \sum_{u\in\{\pm1\}^k,\ u_i=-\alpha} \exp\left(-\frac{1}{4\sigma^2}\left\|\varphi(u)-\varphi(v)\right\|^2\right) \\
&= \mathbf{E}\left(A_{(-\alpha)}^i(Y)\right).
\end{aligned}$$

Da außerdem gilt

$$\mathbf{E}\left(A_{+1}^i(Y)\right) + \mathbf{E}\left(A_{-1}^i(Y)\right) = \frac{1}{2^k\sqrt{2^n}} \sum_{v\in\{\pm1\}^k} \sum_{u\in\{\pm1\}^k} \exp\left(-\frac{1}{4\sigma^2}\left\|\varphi(u)-\varphi(v)\right\|^2\right)$$

folgt

$$\begin{aligned}
\mathbf{E}\left(A_\alpha^i(Y)\right) &= \frac{1}{2^{k+1}\sqrt{2^n}} \sum_{u,v\in\{\pm1\}^k} \exp\left(-\frac{1}{4\sigma^2}\left\|\varphi(u)-\varphi(v)\right\|_2^2\right) \\
&= 2^{-\left(\frac{n}{2}+k+1\right)} \sum_{u,v\in\{\pm1\}^k} \exp\left(-\frac{1}{\sigma^2} d_{\text{ham}}\left(\varphi(u),\varphi(v)\right)\right) \\
&= \frac{1}{2^{\frac{n}{2}+k+1}} \sum_{u,v\in\{\pm1\}^k} \prod_{q=1}^{Q} \exp\left(-\frac{1}{4\sigma^2}\left\|\psi(s_q^u)-\psi(s_q^v)\right\|_2^2\right) \\
&= \frac{1}{2^{\frac{n}{2}+k+1}} \sum_{u,v\in\{\pm1\}^k} \prod_{q=1}^{Q} \exp\left(-\frac{1}{\sigma^2} d_{\text{ham}}\left(\psi(s_q^u),\psi(s_q^v)\right)\right).
\end{aligned}$$

Ad (ii): Mit (i) folgt

$$\begin{aligned}
\mathbf{E}\left(A_\alpha^i(Y)\right) &= 2^{-\left(\frac{n}{2}+k\right)} \sum_{u\in\{\pm1\}^k} \sum_{v\in\{\pm1\}^k,\ v_1=+1} \exp\left(-\frac{1}{4\sigma^2}\left\|\varphi(u)-\varphi(v)\right\|^2\right) \\
&= 2^{-\left(\frac{n}{2}+k\right)} \sum_{u\in\{\pm1\}^k} \left[\sum_{v\in\{\pm1\}^k,\ v_1=+1,v_2=+1} \exp\left(-\frac{1}{4\sigma^2}\left\|\varphi(u)-\varphi(v)\right\|^2\right) + \right. \\
&\qquad \left. \sum_{v\in\{\pm1\}^k,\ v_1=+1,v_2=-1} \exp\left(-\frac{1}{4\sigma^2}\left\|\varphi(u)-\varphi(v)\right\|^2\right) \right] \\
&= 2^{-\left(\frac{n}{2}+k\right)} \sum_{u\in\{\pm1\}^k} \left[\sum_{v\in\{\pm1\}^k,\ v_1=+1,v_2=+1} \exp\left(-\frac{1}{4\sigma^2}\left\|\varphi(u)-\varphi(v)\right\|^2\right) + \right. \\
&\qquad \left. \sum_{v\in\{\pm1\}^k,\ v_1=+1,v_2=-1} \exp\left(-\frac{1}{4\sigma^2}\left\|\varphi(\gamma^2(u))-\varphi(\gamma^2(v))\right\|^2\right) \right] \\
&= 2\cdot 2^{-\left(\frac{n}{2}+k\right)} \sum_{u\in\{\pm1\}^k} \sum_{v\in\{\pm1\}^k,\ v_1=+1,v_2=+1} \exp\left(-\frac{1}{4\sigma^2}\left\|\varphi(u)-\varphi(v)\right\|^2\right) \\
&= 2^{-\left(\frac{n}{2}+1\right)} \sum_{u\in\{\pm1\}^k} \exp\left(-\frac{1}{4\sigma^2}\left\|\varphi(u)-\varphi(u_0)\right\|^2\right) \\
&= 2^{-\left(\frac{n}{2}+1\right)} \sum_{u\in\{\pm1\}^k} \exp\left(-\frac{1}{\sigma^2} d_{\text{ham}}\left(\varphi(u),c_0\right)\right) \\
&= \frac{1}{2^{\frac{n}{2}+1}} \sum_{u\in\{\pm1\}^k} \prod_{q=1}^{Q} \exp\left(-\frac{1}{\sigma^2} d_{\text{ham}}\left(\psi(s_q^u),d_0\right)\right).
\end{aligned}$$

Ad (iii): Es gilt

$$\mathbf{E}\left(A_\alpha^i(Y)\right) = 2^{-\left(\frac{n}{2}+1\right)} \left[1 + \sum_{u\in\{\pm1\}^k,\ u\neq u_0} \exp\left(-\frac{1}{\sigma^2} d_{\text{ham}}\left(\varphi(u),c_0\right)\right)\right]$$

und somit

$$\mathbf{E}\left(A_\alpha^i(Y)\right) \geq 2^{-\left(\frac{n}{2}+1\right)}$$

und

$$\mathbf{E}\left(A_\alpha^i(Y)\right) \leq 2^{-\left(\frac{n}{2}+1\right)} \cdot \left(1+(2^k-1)\exp\left(-\frac{1}{\sigma^2}d_{\text{ham}}(\varphi)\right)\right).$$

□

Mit (ii) läßt sich $\mathbf{E}\left(A_\alpha^i(Y)\right)$ mit einer Variante von Algorithmus 4.3 auf Seite 79 beziehungsweise Algorithmus 4.4 auf Seite 80 berechnen.

Vorgehensweise zur numerischen Berechnung:

- Definiere

$$\begin{aligned} V_j &:= V, && \text{für } j \in \{1,\ldots,a\}, \\ V_j &:= \{v_0\}, && \text{für } j \in \{a+1,\ldots,Q+1\}. \end{aligned}$$

- Definiere

$$\begin{aligned} &\mu : S \to \mathbb{R}, \\ &\quad s \mapsto \exp\left(-\frac{1}{\sigma^2}d_{\text{ham}}(\psi(s),d_0)\right), \end{aligned} \tag{4.27}$$

  und für $q \in \{1,\ldots,Q\}$

$$\begin{aligned} \mu_q : S &\to \mathbb{R}, \\ s &\mapsto \mu(s). \end{aligned}$$

- Man berechne

$$A_m(s), \quad \text{für } s \in S,\ m \in \{1,\ldots,Q\},$$

  mit der Rekursionsformeln (4.13) und den Startwerten $A_0(s)$ mit (4.11).

- Man berechne

$$\mathbf{E}\left(A_\alpha^i(Y)\right) = 2^{-\left(\frac{n}{2}+1\right)} \sum_{v \in V} A_Q(\hat{T}(v,s_0)).$$

Algorithmus 4.5 auf Seite 86 beschreibt für den Spezialfall des binären Zustandsübergangs die Berechnung von $\mathbf{E}\left(A_\alpha^i(Y)\right)$, wobei μ nach (4.27) berechnet wird. In Algorithmus 4.5 werden im Sinne einer kompakteren Aufschreibung zu viele Operationen für $q = a+1,\ldots,Q+1$ durchgeführt. Vergleicht man mit (4.13), so erkennt man, daß im $Q+1$-ten Trellis-Segment tatsächlich nur $A(s_0,Q+1)$ korrekt berechnet wird (der Rest ist aber uninteressant).

Da die Werte $A_\alpha^i(y)$ die bedingten Wahrscheinlichkeiten

$$P\left(\{\omega \in \Omega;\ U_i(\omega) = \alpha\} \,|\, \{\omega \in \Omega;\ Y(\omega) = y\}\right)$$

bis auf einen Vorfaktor darstellen, wäre auch aus numerischen Gründen $\mathbf{E}\left(A_\alpha^i(Y)\right) = \frac{1}{2}$ wünschenswert.

$\mathbf{E}\left(A_\alpha^i(Y)\right)$ Binär: **Eingang** μ; **Ausgang** $\mathbf{E}\left(A_\alpha^i(Y)\right)$;

für $s \in S$: *Vorbelegung*
    $A(s,0) := 0$;
$A(s_0,0) := 1$; *Startzustand*
für $q = 1,\ldots,Q$: *Fortschreiten im Trellis-Diagramm*
    für $s \in S$: *Betrachtung aller Zustände*
        $t^+ := \hat{T}(+1,s)$; $t^- := \hat{T}(-1,s)$; *Vorgängerzustände*
        $A(s,q) := \mu(s,q) \cdot (A(t^+,q-1) + A(t^-,q-1))$; *Berechnung von A*
$t^+ := \hat{T}(+1,s_0)$; $t^- := \hat{T}(-1,s_0)$; *Vorgängerzustände*
$\mathbf{E}\left(A_\alpha^i(Y)\right) = 2^{-\left(\frac{n}{2}+1\right)}(A(t^+,Q) + A(t^-,Q))$; *Berechnung von* $\mathbf{E}\left(A_\alpha^i(Y)\right)$

Algorithmus 4.5: *Berechnung von* $\mathbf{E}\left(A_\alpha^i(Y)\right)$ *bei binärem Zustandsübergang*

Bei der Berechnung der $\mu_q(s)$ in Abschnitt 4.5 kann daher ein Skalierungsfaktor

$$\kappa(\varphi,\sigma^2) := \left(\frac{1}{2^{\frac{n}{2}}} \sum_{u\in\{\pm1\}^k} \prod_{q=1}^{Q} \exp\left(-\frac{1}{\sigma^2} d_{\text{ham}}\left(\psi(s_q^u), d_0\right)\right)\right)^{-\frac{1}{Q}}$$
$$= 2^{\frac{d}{2}} \left(\sum_{u\in\{\pm1\}^k} \prod_{q=1}^{Q} \exp\left(-\frac{1}{\sigma^2} d_{\text{ham}}\left(\psi(s_q^u), d_0\right)\right)\right)^{-\frac{1}{Q}}.$$

hinzugefügt werden.

Für das Beispiel des Industriestandard-1/2-Codes ist $\mathbf{E}\left(A_\alpha^i(Y)\right)$ in Abbildung 4.6 auf Seite 87 und $\kappa(\varphi,\sigma^2)$ in Abbildung 4.7 auf Seite 87 dargestellt.

Abschliessend können im folgenden Lemma noch die Erwartungswerte der einseitigen L-Werte betrachtet werden, die als Maß der Verzerrung der Soft-Outputs im Vergleich zu $\{\pm1\}$ dienen. In Abschnitt 6.3 ist dargestellt, wie die einseitigen L-Werte für numerische Vergleich eingesetzt werden können.

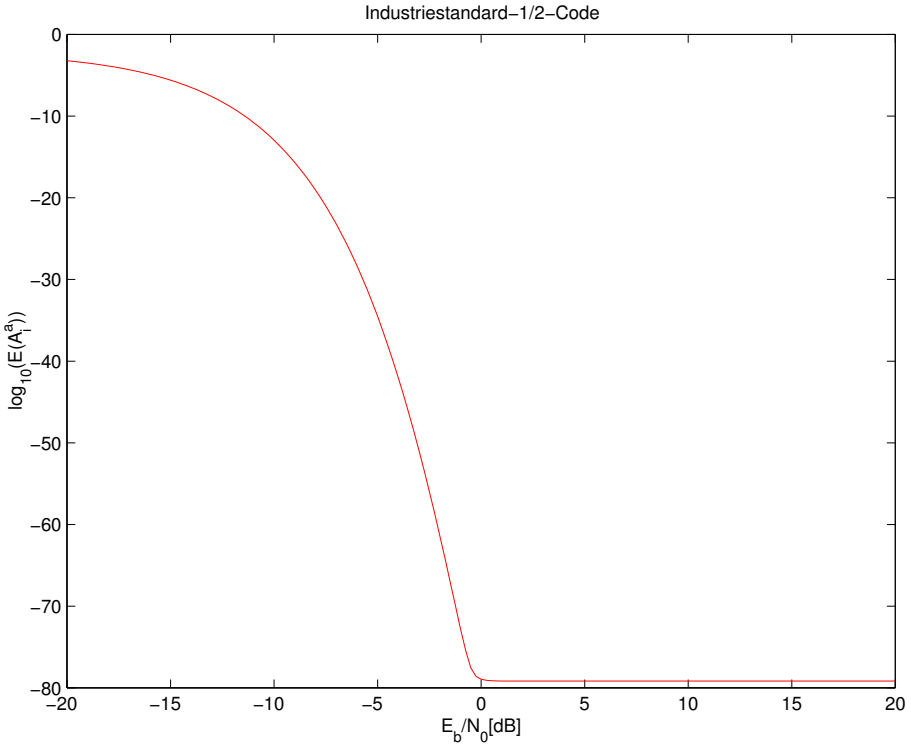

Abbildung 4.6: *Logarithmierte Erwartungswerte* $\mathbf{E}\left(A^i_\alpha(Y)\right)$ *für den Industriestandard-1/2-Code*

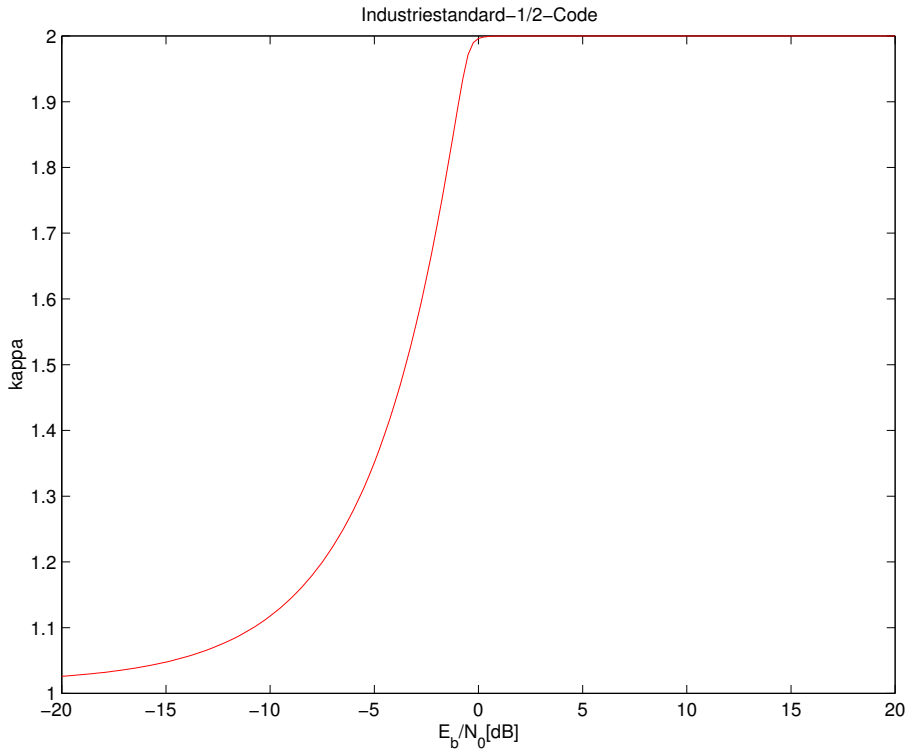

Abbildung 4.7: *Skalierungsfaktoren* $\kappa(\varphi,\sigma^2)$ *für den Industriestandard-1/2-Code*

Lemma 4.17 (Erwartungswert der einseitigen L-Werte)
Voraussetzung 4.3 auf Seite 64 sei erfüllt und es sei[11]

$$A^i_\alpha(y) := \sum_{u\in\{\pm1\}^k,\ u_i=\alpha} \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-y\right\|_2^2\right)$$

für $i=1,\ldots,k$ *und* $\alpha\in\{\pm1\}$.

Definiere die Zufallsvariable des i-ten einseitigen L-Werts als

$$\tilde{L}_i:\ \Omega\to\mathbb{R},$$
$$\omega\mapsto\tilde{L}_i(\omega) := U_i(\omega)\cdot L(U_i|Y(\omega) := U_i(\omega)\cdot\ln\left(\frac{A^i_{+1}(Y(\omega))}{A^i_{-1}(Y(\omega))}\right).$$

(i) *Für* $i=1,\ldots,k$ *und* $y\in\mathbb{R}^n$ *gilt*

$$A^i_\alpha(y) = A^i_{-\alpha}(\delta^i(y)),$$
$$A^i_\alpha(y) = A^i_\alpha(\delta^j(y)),\quad \text{für } j=1,\ldots,k;\ j\neq i.$$

(ii) *Für* $i=1,\ldots,k$ *gilt*

$$\mathbf{E}\left(\tilde{L}_i\right) = \frac{1}{2^{k-1}\sqrt{(2\pi\sigma^2)^n}}\int_{\mathbb{R}^n}\ln\left(\frac{A^i_{+1}(z)}{A^i_{-1}(z)}\right)A^i_{+1}(z)dz$$
$$= \frac{1}{2^{k-1}\sqrt{(2\pi\sigma^2)^n}}\int_{\mathbb{R}^n}\ln\left(\frac{A^i_{-1}(z)}{A^i_{+1}(z)}\right)A^i_{-1}(z)dz.$$

(iii) *Mit* $Z\sim\mathcal{N}(0,\sigma^2I_n)$ *und* $c_0 := (+1,\ldots,+1)^\top\in\{\pm1\}^n$ *gilt für* $i=1,\ldots,k$*:*

$$\mathbf{E}\left(\tilde{L}_i\right) = \frac{1}{\sqrt{(2\pi\sigma^2)^n}}\int_{\mathbb{R}^n}\ln\left(\frac{A^i_{+1}(z)}{A^i_{-1}(z)}\right)\exp\left(-\frac{1}{2\sigma^2}\left\|c_0-z\right\|_2^2\right)dz$$
$$= \mathbf{E}\left(\ln\left(\frac{A^i_{+1}(c_0+Z)}{A^i_{-1}(c_0+Z)}\right)\right)$$

Beweis. Ad (i):

$$A^i_\alpha(y) = \sum_{u\in\{\pm1\}^k,\ u_i=\alpha}\exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-y\right\|_2^2\right)$$
$$= \sum_{u\in\{\pm1\}^k,\ u_i=\alpha}\exp\left(-\frac{1}{2\sigma^2}\left\|\delta^i(\varphi(u))-\delta^i(y)\right\|_2^2\right)$$
$$= \sum_{u\in\{\pm1\}^k,\ u_i=\alpha}\exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(\gamma^i(u))-\delta^i(y)\right\|_2^2\right)$$
$$= \sum_{u\in\{\pm1\}^k,\ u_i=-\alpha}\exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-\delta^i(y)\right\|_2^2\right)$$
$$= A^i_{-\alpha}(\delta^i(y)).$$

[11] siehe Korollar 4.4 auf Seite 65

Sei $j \in \{1,\ldots,k\} \setminus \{i\}$. Dann gilt

$$\begin{aligned}
A^i_\alpha(y) &= \sum_{u\in\{\pm1\}^k,\ u_i=\alpha} \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-y\right\|_2^2\right) \\
&= \sum_{u\in\{\pm1\}^k,\ u_i=\alpha,\ u_j=+1} \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-y\right\|_2^2\right) \\
&\quad + \sum_{u\in\{\pm1\}^k,\ u_i=\alpha,\ u_j=-1} \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-y\right\|_2^2\right) \\
&= \sum_{u\in\{\pm1\}^k,\ u_i=\alpha,\ u_j=-1} \exp\left(-\frac{1}{2\sigma^2}\left\|\delta^j(\varphi(u))-y\right\|_2^2\right) \\
&\quad + \sum_{u\in\{\pm1\}^k,\ u_i=\alpha,\ u_j=+1} \exp\left(-\frac{1}{2\sigma^2}\left\|\delta^j(\varphi(u))-y\right\|_2^2\right) \\
&= A^i_\alpha(\delta^j(y)).
\end{aligned}$$

Ad (ii):

$$\begin{aligned}
\mathbf{E}\left(\tilde{L}_i\right) &= \mathbf{E}\left(U_i \cdot \ln\left(\frac{A^i_{+1}(Y)}{A^i_{-1}(Y)}\right)\right) \\
&= \sum_{u\in\{\pm1\}^k} \frac{1}{2^k} \int_{\mathbb{R}^n} u_i \cdot \ln\left(\frac{A^i_{+1}(\varphi(u)+z)}{A^i_{-1}(\varphi(u)+z)}\right) \cdot \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \exp\left(-\frac{1}{2\sigma^2}\left\|z\right\|_2^2\right) dz \\
&= \frac{1}{2^k\sqrt{(2\pi\sigma^2)^n}} \sum_{u\in\{\pm1\}^k} u_i \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{+1}(z)}{A^i_{-1}(z)}\right) \cdot \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-z\right\|_2^2\right) dz \\
&= \frac{1}{2^k\sqrt{(2\pi\sigma^2)^n}} \sum_{u\in\{\pm1\}^k,\ u_i=+1} \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{+1}(z)}{A^i_{-1}(z)}\right) \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-z\right\|_2^2\right) dz \\
&\quad - \frac{1}{2^k\sqrt{(2\pi\sigma^2)^n}} \sum_{u\in\{\pm1\}^k,\ u_i=-1} \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{+1}(z)}{A^i_{-1}(z)}\right) \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-z\right\|_2^2\right) dz \\
&= \frac{1}{2^k\sqrt{(2\pi\sigma^2)^n}} \sum_{u\in\{\pm1\}^k,\ u_i=+1} \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{+1}(z)}{A^i_{-1}(z)}\right) \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-z\right\|_2^2\right) dz \\
&\quad - \frac{1}{2^k\sqrt{(2\pi\sigma^2)^n}} \sum_{u\in\{\pm1\}^k,\ u_i=+1} \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{+1}(\delta^i(z))}{A^i_{-1}(\delta^i(z))}\right) \exp\left(-\frac{1}{2\sigma^2}\left\|\delta^i(\varphi(u)-z)\right\|_2^2\right) dz \\
&= \frac{1}{2^k\sqrt{(2\pi\sigma^2)^n}} \sum_{u\in\{\pm1\}^k,\ u_i=+1} \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{+1}(z)}{A^i_{-1}(z)}\right) \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-z\right\|_2^2\right) dz \\
&\quad - \frac{1}{2^k\sqrt{(2\pi\sigma^2)^n}} \sum_{u\in\{\pm1\}^k,\ u_i=+1} \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{-1}(z)}{A^i_{+1}(z)}\right) \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-z\right\|_2^2\right) dz
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2^{k-1}\sqrt{(2\pi\sigma^2)^n}} \sum_{u\in\{\pm1\}^k,\ u_i=+1} \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{+1}(z)}{A^i_{-1}(z)}\right) \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-z\right\|_2^2\right) dz \\
&= \frac{1}{2^{k-1}\sqrt{(2\pi\sigma^2)^n}} \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{+1}(z)}{A^i_{-1}(z)}\right) A^i_{+1}(z)\,dz \\
&= \frac{1}{2^{k-1}\sqrt{(2\pi\sigma^2)^n}} \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{-1}(\delta^i(z))}{A^i_{+1}(\delta^i(z))}\right) A^i_{-1}(\delta^i(z))\,dz \\
&= \frac{1}{2^{k-1}\sqrt{(2\pi\sigma^2)^n}} \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{-1}(z)}{A^i_{+1}(z)}\right) A^i_{-1}(z)\,dz.
\end{aligned}
$$

Ad (iii): Für jedes $j \in \{1,\ldots,k\}\setminus\{i\}$ gilt

$$
\begin{aligned}
\mathbf{E}\left(\tilde{L}_i\right) &= \frac{1}{2^{k-1}\sqrt{(2\pi\sigma^2)^n}} \sum_{u\in\{\pm1\}^k,\ u_i=+1,u_j=+1} \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{+1}(z)}{A^i_{-1}(z)}\right) \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-z\right\|_2^2\right) dz \\
&\quad + \frac{1}{2^{k-1}\sqrt{(2\pi\sigma^2)^n}} \sum_{u\in\{\pm1\}^k,\ u_i=+1,u_j=-1} \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{+1}(\delta^j(z))}{A^i_{-1}(\delta^j(z))}\right) \exp\left(-\frac{1}{2\sigma^2}\left\|\delta^j(\varphi(u))-\delta^j(z)\right\|_2^2\right) dz \\
&= 2\cdot\frac{1}{2^{k-1}\sqrt{(2\pi\sigma^2)^n}} \sum_{u\in\{\pm1\}^k,\ u_i=+1,u_j=+1} \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{+1}(z)}{A^i_{-1}(z)}\right) \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-z\right\|_2^2\right) dz.
\end{aligned}
$$

Somit gilt

$$
\begin{aligned}
\mathbf{E}\left(\tilde{L}_i\right) &= 2^{k-1}\cdot\frac{1}{2^{k-1}\sqrt{(2\pi\sigma^2)^n}} \sum_{u=(+1,\ldots,+1)^\top} \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{+1}(z)}{A^i_{-1}(z)}\right) \exp\left(-\frac{1}{2\sigma^2}\left\|\varphi(u)-z\right\|_2^2\right) dz \\
&= \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \int_{\mathbb{R}^n} \ln\left(\frac{A^i_{+1}(z)}{A^i_{-1}(z)}\right) \exp\left(-\frac{1}{2\sigma^2}\left\|c_0-z\right\|_2^2\right) dz \\
&= \mathbf{E}\left(\ln\left(\frac{A^i_{+1}(c_0+Z)}{A^i_{-1}(c_0+Z)}\right)\right).
\end{aligned}
$$

□

Kapitel 5

Soft-Decision Decodierung binärer linearer Blockcodes

Sometimes, you have to punch your way through.

(Kathryn Janeway)

5.1 Verfahrens-Verkettung

In Abschnitt 3.6 wurde dargelegt, daß zwei Decodierer über eine Soft-Schnittstelle (Soft-Werte) gekoppelt werden sollten, um eine maximale Informationsweitergabe zu gewährleisten.

In diesem Kapitel wird vorausgesetzt, daß Soft-Werte, also ein reeller Vektor, als Eingangsdaten für einen Decodierer vorliegen. Aufgrund dieser soften Werte soll nun für beliebige binäre lineare Blockcodes eine „harte“ Decodierungsentscheidung getroffen werden, das heißt, eine Methode zur **Soft-Decision Decodierung** wird entwickelt. Dabei ist es nicht von Bedeutung, ob die Soft-Werte von einem Demodulator oder einem vorgeschalteten Decodierer geliefert werden. Abbildung 5.1 zeigt den Einsatz bei der Decodierung von verketteten Codes, wobei der vorgeschaltete Soft-Output Decodierer etwa der aus Kapitel 4 sein kann[1].

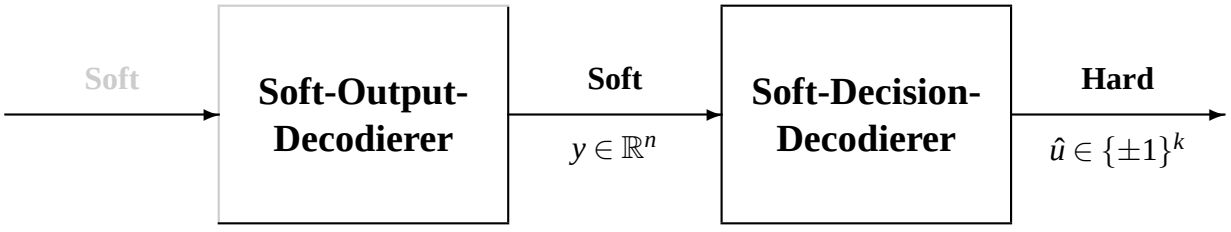

Abbildung 5.1: *Soft-Decision Decodierung bei verketteten Codes*

In der im weiteren betrachteten allgemeinen Darstellung kann das entwickelte Verfahren also wie im vorangegangenen Kapitel isoliert betrachtet werden, da die Verfahrenskopplung über die Soft-Schnittstelle erfolgt. Gemäß der Allgemeinheit der Darstellung wird der Eingang des Decodierers wieder als $y \in \mathbb{R}^n$ und der Ausgang als $\hat{u} \in \{\pm 1\}^k$ bezeichnet[2].

[1] Die in Kapitel 4 berechneten L-Werte haben zwar nicht Erwartungswerte aus $\{\pm 1\}$, wie man es von der Kanalausgabe erwartet, sondern bilden mit der Codierungsabbildung und dem Kanal einen verzerrten Superkanal, aber sie können dennoch ohne Skalierung verwendet werden, wie auf Seite 148 dargestellt.

[2] Eine allgemeine gemeinsame Darstellung der Decodierung verketteter Codes ist in Abschnitt 3.6 aufgeführt.

5.2 Zielfunktionsdefinition

In Abschnitt 3.4 wurde bereits gezeigt, daß unter geeigneten Voraussetzungen (AWGN-Kanal) eine Minimalfehler Soft-Decision Decodierung äquivalent zu einer Minimaldistanz Soft-Decision Decodierung bezüglich der euklidischen Norm ist. Somit muß zur fehlerminimalen Decodierung der Minimierer einer (Ziel-)Funktion gefunden werden.

Zunächst formulieren wir die benötigten Voraussetzungen gebündelt, wobei insbesondere ein AWGN-Kanalmodell angenommen wird.

Voraussetzung 5.1 (Soft-Decision Decodierung bei AWGN-Kanälen)
Es sei $(\Omega, \mathcal{S}, P)$ ein Wahrscheinlichkeitsraum und $n,k \in \mathbb{N}$, $n > k$. Weiter sei (n,k,φ) ein binärer linearer (n,k)-Blockcode, $\mathcal{K}$ sei ein AWGN-Kanal der Dimension n mit bitweiser Varianz $\sigma^2 > 0$ der Kanalstörung, $Y : \Omega \to \mathbb{R}^n$ sei die Zufallsvariable der Kanalausgabe und $U : \Omega \to \{\pm 1\}^k$ sei eine von $\mathcal{K}_c$ für alle $c \in \{\pm 1\}^n$ stochastisch unabhängige gleichverteilte Zufallsvariable mit

$$Y(\omega) = \mathcal{K}(\varphi(U(\omega)), \omega), \quad \text{für alle } \omega \in \Omega,$$

d.h., $\varphi(U)$ ist die Zufallsvariable der Kanaleingabe.

Man beachte, daß Voraussetzung 5.1 die Voraussetzung 3.12 auf Seite 40 umfaßt.

Als nächstes kann die Zielfunktion definiert werden.

Definition 5.2 (Soft-Decision Zielfunktion)
Voraussetzung 5.1 auf Seite 92 sei erfüllt. Dann heißt zu jeder Realisierung y von Y die Abbildung

$$\begin{aligned} F : \{\pm 1\}^k &\to \mathbb{R}_0^+ \\ u &\mapsto \sum_{j=1}^{n} \left(\bigoplus_{i \in J_j} u_i - y_j \right)^2 . \end{aligned} \tag{5.1}$$

Soft-Decision Zielfunktion.

Zur Vereinfachung der Darstellung[3] wurde hier (wie auch stets später) eine beliebige aber feste Realisierung y der Kanalausgabe Y angenommen.

Daß ein globaler Minimierer von F tatsächlich die fehlerminimale Decodierungsentscheidung darstellt, zeigt das folgende Korollar zu Satz 3.23 auf Seite 47.

[3] Da F von der Realisierung y abhängt, wäre eigentlich bei wechselndem y etwa eine Schreibweise F_y zu verwenden.

Korollar 5.3 (Soft-Decision Minimalfehler-Decodierung durch Zielfunktionsminimierung) *Voraussetzung 5.1 auf Seite 92 sei erfüllt. Es sei* $\delta_{SD}: \mathbb{R}^n \to \{\pm 1\}^k$ *die Soft-Decision Decodierungsabbildung, die jedem* $y \in Y$ *einen Minimierer*[4] *der Soft-Decision Zielfunktion* F *zuordnet, das heißt*

$$\begin{aligned}\delta_{SD}: \mathbb{R}^n &\to \{\pm 1\}^k \\ y &\mapsto \operatorname*{argmin}_{u\in\{\pm 1\}^k} \left\{ \sum_{j=1}^{n} \left(\bigoplus_{i\in J_j} u_i - y_j \right)^2 \right\}.\end{aligned}$$

Dann ist δ_{SD} *eine Minimalfehler Soft-Decision Decodierungsabbildung.*

Beweis. Für jedes beliebig aber fest gewählte $y \in \mathbb{R}^n$ gilt

$$\begin{aligned}\|\varphi(\delta_{SD}(y)) - y\|_2 = \sqrt{F(\delta_{SD}(y))} &= \sqrt{F\left(\operatorname*{argmin}_{\hat{u}\in\{\pm 1\}^k} \{F(\hat{u})\}\right)} \\ &\leq \sqrt{F(u)} = \|\varphi(u) - y\|_2\end{aligned}$$

für alle $u \in \{\pm 1\}^k$. Nach Definition 3.22 auf Seite 47 ist δ_{SD} somit eine Minimaldistanz-Decodierungabbildung bezüglich der euklidischen Norm $\|.\|_2$ und mit Satz 3.23 auf Seite 47 ist δ_{SD} folglich eine Minimalfehler Soft-Decision Decodierungsabbildung..

□

Unter den genannten Voraussetzungen ist das Problem der fehlerminimalen Soft-Decision Decodierung somit für jede Realisierung $y \in \mathbb{R}^n$ der Kanalausgabe Y auf die Lösung des Minimierungsproblems

$$\min_{u\in\{\pm 1\}^k} \{F(u)\} \tag{5.2}$$

reduziert.

Da die Menge $\{\pm 1\}^k$ aber 2^k Elemente enthält, kann man bei Anwendungen der Praxis mit großen k nicht einfach alle Wörter u in die Zielfunktion einsetzen, sondern muß den Codebaum geeignet behandeln.

5.3 Verfahrensschritte

Zu jeder empfangenen Nachricht y ist jeweils ein Wort $\hat{u} \in \{\pm 1\}^k$ gesucht mit der Eigenschaft

$$F(\hat{u}) \leq F(u) \quad \text{für alle } u \in \{\pm 1\}^k.$$

Um nicht alle 2^k Wörter untersuchen zu müssen, was bei großen k numerisch nicht möglich ist, wird nun ein Branch-and-Bound Algorithmus vorgestellt, der die Zahl der zu untersuchenden

[4] Bei beliebig aber fest gewähltem $y \in \mathbb{R}^n$ besitzt die Zielfunktion F ein globales Minimum auf $\{\pm 1\}^k$, welches aber nicht eindeutig sein muß. Somit können mehrere Minimierer, aber mindestens einer, der Zielfunktion F existieren. Der Term $\operatorname*{argmin}_{u\in\{\pm 1\}^k} \{F(u)\}$ soll nach einem beliebigen deterministischen Schema *einen* Minimierer $\hat{u} \in \{\pm 1\}^k$ von F darstellen („argmin“ steht für „Argument des Minimums“).

Wörter stark reduziert. Zudem kann der Algorithmus zu jeder Zeit ein bis dahin bestes Codewort als Ergebnis liefern. Der besondere Vorteil dieses speziellen Branch-and-Bound Verfahrens liegt darin, daß nicht auf dem Codebaum des gegebenen Codes operiert wird, sondern auf dem Codebaum eines adaptiv transformierten Codes, der numerisch besonders günstige Eigenschaft besitzt.

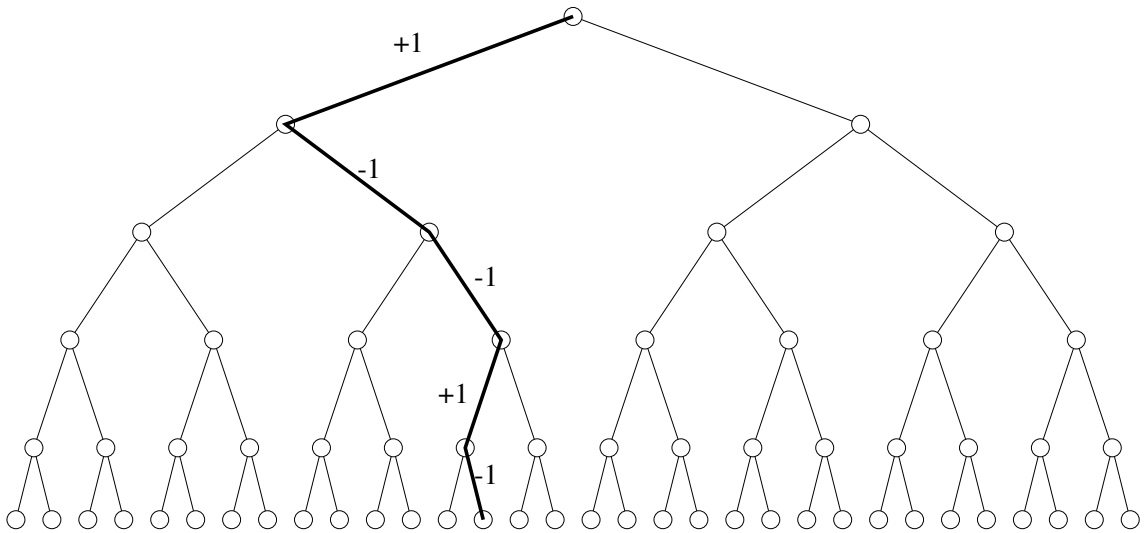

Abbildung 5.2: *Branch-and-Bound auf dem binären Codebaum*

Das Verfahren geht mit folgenden Schritten vor:

1. **„Code-Quasi-Systematisierung“**: In einem ersten Schritt wird der gegebene Blockcode derart in einen identischen quasi-systematischen Blockcode transformiert, daß die Infobits u_i mit den kleinsten Indizes den betragsgrößten Komponenten des Demodulationsergebnisses y zugeordnet werden.
   „Die sichersten Bits werden zuerst decodiert“

2. Im Codebaum des transformierten Codes werden sukzessive die Infobits $\tilde{u}_i$ festgelegt (**Branch**), vergleiche Abbildung 5.2. Durch Betrachtung von unteren Schranken für die noch unbekannten Bestandteile der Bewertungsfunktion F können ganze Unterbäume ohne explizite Auswertung verworfen werden (**Bound**).

3. **Rücktransformation** des Decodierungsergebnisses in den Originalcode.

Das Verfahren arbeitet somit wie in Abbildung 5.3 dargestellt: Der Code wird adaptiv, d.h. abhängig vom Demodulationsergebnis $y \in \mathbb{R}^n$, so transformiert, daß eine Decodierung im transformierten Bereich numerisch effizient durchführbar wird. Das Ergebnis dieser Decodierung wird dann in den „Normalbereich“ rücktransformiert.

Im folgenden wird das Verfahren mathematisch fundiert und algorithmisch ausformuliert. Alternative Verfahren, die die Zielfunktion mit Methoden der nichtlinearen Optimierung behandeln, sind in [Sch97, RSS98] zu finden.

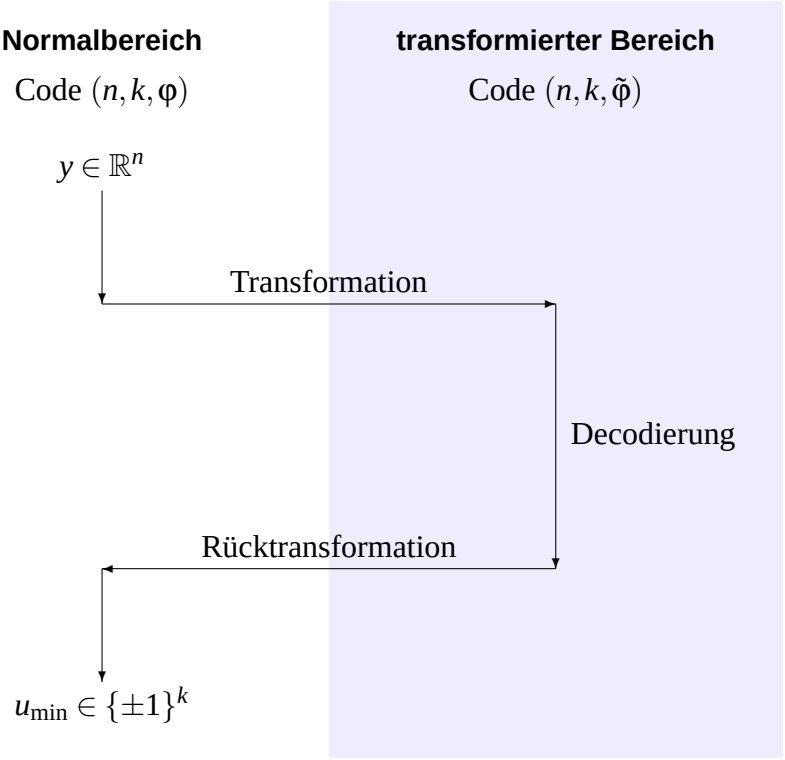

Abbildung 5.3: *Decodierung im transformierten Bereich*

5.4 Quasi-Systematisierung

5.4.1 Vorgehensweise

Zu einem beliebig aber fest gewählten kryptocodierten Codewort $u \in \{\pm 1\}^k$ mit zugehörigem kanalcodierten Codewort $c \in \{\pm 1\}^n$, $c^\top := u^\top G$, und einem beliebig aber fest gewählten Demodulationsergebnis $y \in \mathbb{R}^n$ soll der Code nun so quasi-systematisiert werden, daß $c_i = \tilde{u}_{\dots}$ für die Indizes i mit den betragsgrößten y_i gilt.

Es wird also eine folgende Darstellung der Generatormatrix G betrachtet:

$$G = A\tilde{G}, \quad A \in \{\pm 1\}^{k,k} \text{ regulär}, \ \tilde{G} \in \{\pm 1\}^{k,n}, \tag{5.3}$$

wobei k Spalten von $\tilde{G}$ aus den k Einheitsvektoren bestehen. Die Wahl der Spalten j soll möglichst den Indizes der betragsgrößten y_j entsprechen. Die Multiplikation von links mit A bedeutet auf dem Körper $\{\pm 1\}$ eine Abfolge von Zeilenadditionen der Matrix $\tilde{G}$.

$\tilde{G}$ ist dann die Generatormatrix eines speziell quasi-systematisierten Codes:

$$c^\top = u^\top G = \underbrace{u^\top A}_{=:\tilde{u}^\top}\tilde{G}. \tag{5.4}$$

Es gilt dann nach Konstruktion

$$\tilde{u}_i = c_{\tau(i)}, \quad i = 1,\dots,k, \tag{5.5}$$

wobei $\tau : \{1,\dots,k\} \to \{1,\dots,n\}$ möglichst auf die Indizes der betragsgrößten $y_{\tau(i)}$ abbildet.

Die Beziehung zwischen Original-Codierung und transformierter Codierung ist in Abbildung 5.4 dargestellt.

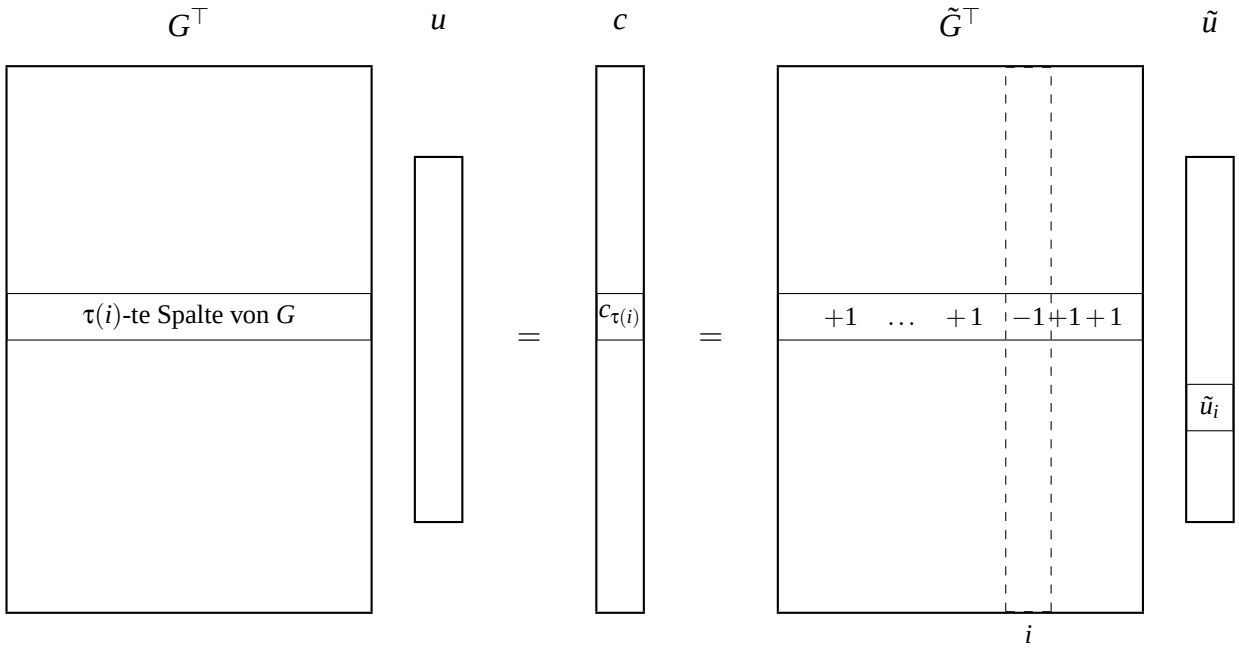

Abbildung 5.4: *Quasi-Systematisierung*

Im folgenden werden analog zu G die zu $\tilde{G}$ gehörigen charakterisierenden Mengen

$$\tilde{J}_1, \ldots, \tilde{J}_n \subseteq \{1, \ldots, k\}$$

betrachtet, also

$$\tilde{J}_j := \{i \in \{1, \ldots, k\};\ \tilde{G}_{ij} = -1\}, \quad \text{für } 1 \leq j \leq n.$$

Nach der Quasi-Systematisierung ist die weitere Vorgehensweise:

- Decodiere y über den durch $\tilde{G}$ quasi-systematisierten Code zu einem $\tilde{u} \in \{\pm 1\}^k$.
- Rücktransformation: Bestimme das Codewort $u \in \{\pm 1\}^k$ mit

$$\tilde{u}^\top = u^\top A, \quad u^\top = \tilde{u}^\top A^{-1}.$$

5.4.2 Algorithmus der Quasi-Systematisierung

Zur Konstruktion eines Quasi-Systematisierungsverfahrens betrachte eine bijektive Abbildung (Sortierung) $\mu : \{1,\dots,n\} \to \{1,\dots,n\}$, also $\mu \in \mathcal{S}_n$, mit

$$|y_{\mu(m)}| \geq |y_{\mu(m+1)}|, \quad 1 \leq m \leq n-1. \tag{5.6}$$

Wie oben beschrieben werden zur Transformation Zeilenadditionen benötigt, die durch Tupel

$$\alpha_m \in \{1,\dots,k\} \times \{1,\dots,k\}, \quad m = 1,\dots,a \tag{5.7}$$

protokolliert werden können, wobei $a \in \mathbb{N}_0$ die Anzahl der Zeilenadditionen ist und $(p,q) = \alpha_m$ bedeutet, daß die q-te Zeile zur p-ten Zeile addiert wurde. Für die Matrixschreibweise sei die Matrix $A_{\alpha_m} = A_{pq} \in \{\pm 1\}^{k,k}$ die $k \times k$-Einheitsmatrix mit Extra-Element -1 in der p-q-Position. Dann bewirkt[5] $A_{pq}G$ die Addition der q-ten Zeile von G zur p-ten Zeile von G.

Der Algorithmus definiert eine weitere bijektive Abbildung $\rho : \{1,\dots,k\} \to \{1,\dots,k\}$, also $\rho \in \mathcal{S}_k$, mit der Eigenschaft

$$|y_{\tau(\rho(s))}| \geq |y_{\tau(\rho(s+1))}|, \quad 1 \leq s \leq k-1. \tag{5.8}$$

Diese Eigenschaft wird beim Branch-and-Bound Algorithmus Verwendung finden[6].

Man betrachte weiterhin die Projektionsabbildung

$$\begin{aligned} &\pi : \mathbb{R} \to \{\pm 1\}, \\ &\qquad x \mapsto \begin{cases} +1, & \text{für } x \geq 0, \\ -1, & \text{für } x < 0. \end{cases} \end{aligned} \tag{5.9}$$

Der Algorithmus 5.1 generiert zusätzlich einen „Startpunkt“ $\tilde{u}^0$, der als erstes approximatives Decodierungsergebnis verwendet werden kann. $\tilde{u}^0$ wird so konstruiert, daß

$$\tilde{u}^0_q = \pi\left(y_{\tau(q)}\right), \quad \text{für } q = 1,\dots,k, \tag{5.10}$$

d.h., die möglichst „besten“ Komponenten von y werden zu $\tilde{u}^0$ gerundet. Im weiteren kann dann auf die Datenhaltung der Abbildung τ verzichtet werden.

Die Hilfsmenge Z im Algorithmus enthält jeweils die Indizes q der schon erzeugten Einheitsvektoren $e_q \in \{\pm 1\}^k$.

Lemma 5.4 (Algorithmus zur Quasi-Systematisierung)
Voraussetzung 5.1 auf Seite 92 sei erfüllt. Dann terminiert der Algorithmus 5.1 für jede Realisierung $y \in \mathbb{R}^n$ der Kanalausgabe. Der mittels Algorithmus 5.1 über die Generatormatrix $\tilde{G}$ definierte binäre lineare (n,k)-Blockcode $(n,k,\tilde{\varphi})$ ist ein zu (n,k,φ) identischer quasi-systematischer (n,k)-Blockcode, und es gilt

$$\tilde{G} = \underbrace{A_{\alpha_a} A_{\alpha_{a-1}} \dots A_{\alpha_2} A_{\alpha_1}}_{=:A^{-1}} G.$$

[5] Der Aufbau von Matrizen, die durch Matrixmultiplikation Zeilen- bzw. Spaltenoperationen an anderen Matrizen vollziehen, ist etwa in [Rei95] dargestellt.

[6] Die sukzessive zu entscheidenden Bits $\tilde{u}_{\dots}$ werden beim Branch-and-Bound Verfahren aufgrund dieser Eigenschaft in der Reihenfolge der Sortierung ρ gewählt, also $\tilde{u}_{\rho(1)}, \tilde{u}_{\rho(2)}, \tilde{u}_{\rho(3)}, \dots$

Die Spalten $\tau(1),\ldots,\tau(k)$ *von* $\tilde{G}$ *sind die* k *Einheitsvektoren des* $\{\pm 1\}^k$ *und weiter gilt*

$$|y_{\tau\circ\rho(1)}| \geq |y_{\tau\circ\rho(2)}| \geq \ldots \geq |y_{\tau\circ\rho(k)}|.$$

⌟

Beweis. Der Algorithmus 5.1 terminiert, da die Matrix G den vollen Rang k besitzt. Nach Konstruktion des Verfahrens gilt

$$\tilde{G} = A^{-1}G$$

und $A^{-1} = A_{\alpha_a}A_{\alpha_{a-1}}\ldots A_{\alpha_2}A_{\alpha_1}$ ist regulär, das heißt, $(n,k,\tilde{\varphi})$ ist ein zu (n,k,φ) identischer (n,k)-Blockcode. Weiter ist $(n,k,\tilde{\varphi})$ nach Konstruktion ein quasi-systematischer (n,k)-Blockcode.

Die restlichen Aussagen gelten nach Konstruktion des Verfahrens.

□

Quasi-Systematisierung (Variante 1): **Eingang** G, μ, y; **Ausgang** $\tilde{G}$, $\tilde{u}^0$, α, a, ρ, ρ^{-1};

```
Z := ∅;                                              Einheitsvektoren-Index
a := 0;   G̃ := G;   m := 1;
solange |Z| < k:
    j := μ(m);
    falls {i ∈ {1,...,k} \ Z; G̃_ij = −1} ≠ ∅:
        q := min({i ∈ {1,...,k} \ Z; G̃_ij = −1});
        Z := Z ∪ {q};
        τ(q) := j;                    zur Vollständigkeit; wird nicht implementiert
        ũ^0_q := π(y_j);                                  Startpunktgenerierung
        ρ(|Z|) := q;                                      für Branch-and-Bound
        ρ^{-1}(q) := |Z|;                                 für Branch-and-Bound
        für p ∈ {1,...,k} \ {q}:
            falls G̃_pj = −1:
                a := a + 1;
                α_a := (p,q);                             q-te Zeile zur p-ten addiert
                für r = 1,...,n:
                    G̃_pr := G̃_pr ⊕ G̃_qr;
    m := m + 1;
```

Algorithmus 5.1: *Spezielle Quasi-Systematisierung des* (n,k)*-Blockcodes (Variante 1)*

Inklusive der Sortierung μ des Demodulationsergebnisses beträgt der numerische Aufwand für Algorithmus 5.1

- $\mathrm{O}(n\log n)$ Fließkomma-Operationen (Sortierung μ) und
- $\mathrm{O}\left(k^2 n\right)$ Binär-Operationen.

5.4.3 Algorithmus der Quasi-Systematisierung (Variante 2)

In obigem Verfahren wurde eine vollständige Diagonalisierung der Matrix G beziehungsweise $\tilde{G}$ durchgeführt. Bei der im weiteren vorgestellten Implementierung eines Branch-and-Bound Verfahrens werden zugehörige Indexmengen $\tilde{J}_j$ anstelle der Matrix $\tilde{G}$ verwendet. Abhängig von der Art des verwendeten Codes können durch die folgende zweite Variante des Quasi-Systematisierungs-Algorithmus numerische Operationen eingespart werden. In Algorithmus 5.2 werden dabei die „Einheitsspalten“ nicht mehr tatsächlich zu Einheitsvektoren transformiert, sondern die nötigen Operationen lediglich protokolliert. Desweiteren muß keine Kopie der Matrix G angelegt werden (die Spalten werden sukzessive kopiert). Bei linearen Abhängigkeiten werden allerdings zusätzliche Operationen notwendig, da Indizes dann doppelt behandelt werden (festgehalten in der Hilfsmenge R).

Es werden $\mathrm{O}\left(k^2\right)$ numerische Operationen eingespart. Da auch die Datenhaltung günstiger ist, sind die Einspareffekte in der praktischen Anwendung aber deutlich höher.

Nach Konstruktion gelten alle Aussagen von Lemma 5.4 auf Seite 97 auch für Algorithmus 5.2, wobei $\tilde{G}$ implizit durch die charakterisierenden Mengen $\tilde{J}_j$, $j = 1, \ldots, n$, gegeben ist.

Quasi-Systematisierung (Variante 2): **Eingang** G, μ, y; **Ausgang** $\tilde{J}$, $\tilde{u}^0$, α, a, ρ, ρ^{-1};

```
Z := ∅;                                                  Einheitsvektoren-Index
R := ∅;                                                  Restindizes
a := 0;   m := 1;
solange |Z| < k:
    j := μ(m);
    für i = 1,...,k:                                     Kopie der j-ten Spalte
        g̃_i := G_ij;
    für b = 1,...,a:                                     Systematisiere die j-te Spalte
        (p,q) := α_b;                                    q-te Zeile zur p-ten addieren
        g̃_p := g̃_p ⊕ g̃_q;
    falls {i ∈ {1,...,k} \ Z; g̃_i = −1} ≠ ∅:
        q := min({i ∈ {1,...,k} \ Z; g̃_i = −1});
        Z := Z ∪ {q};
        ũ^0_q := π(y_j);                                 Startpunktgenerierung
        J̃_j := {q};                                      Einpunktige Indexmenge
        ρ(|Z|) := q;
        ρ^{-1}(q) := |Z|;
        für p ∈ {1,...,k} \ {q}:
            falls g̃_p = −1:
                a := a + 1;
                α_a := (p,q);                            q-te Zeile zur p-ten addieren
    sonst:
        R := R ∪ {j};
    m := m + 1;
für j ∈ R ∪ {μ(m),...,μ(n)}:                             Unsystematische Spalten
    für i = 1,...,k:                                     Kopie der j-ten Spalte
        g̃_i := G_ij;
    für b = 1,...,a:                                     Behandle die j-te Spalte
        (p,q) := α_b;                                    q-te Zeile zur p-ten addieren
        g̃_p := g̃_p ⊕ g̃_q;
    J̃_j := ∅;
    für p = 1,...,k:
        falls g̃_p = −1:
            J̃_j := J̃_j ∪ {p};
```

Algorithmus 5.2: *Spezielle Quasi-Systematisierung des (n,k)-Blockcodes (Variante 2)*

5.4.4 Quasi-Systematisierung systematischer Codes

Der Algorithmus 5.2 kann beliebige binäre lineare Blockcodes verarbeiten, da keine speziellen Annahmen getroffen wurden. Kennt man die spezielle Struktur eines Codes, so läßt sich das Verfahren „trimmen“, um numerische Operationen einzusparen.

Weit verbreitet sind systematische Blockcodes, das heißt, die ersten k Spalten der Generatormatrix bestehen bereits aus den k Einheitsvektoren.

Die Anpassung von Algorithmus 5.2 für diese wichtige Spezialisierung wollen wir nun als Algorithmus 5.3 formulieren. Im wesentlichen sind dabei zwei Aspekte zu berücksichtigen:

- Wird in Algorithmus 5.2 eine Spalte j mit $j \leq k$ bearbeitet, die unverändert den j-ten Einheitsvektor darstellt, so ist weder eine Spaltenkopie von G notwendig noch Zeilenadditionen. Die Indizes dieser Einheitsvektoren bilden jeweils die Menge $\{1,\ldots,k\} \setminus Z$.

- Wird eine Spalte j bearbeitet, die nicht (oder nicht mehr) einen Einheitsvektor darstellt, und wird q als Pivotelement gewählt (siehe Algorithmus 5.2), so wird durch die nachfolgenden Zeilenadditionen die q-te Spalte de-systematisiert. Statt

  $$q = \min(\{i \in \{1,\ldots,k\} \setminus Z;\ \tilde{g}_i = -1\})$$

  sollte daher

  $$q = \mu\left(\max\left(\mu^{-1}\left(\{i \in \{1,\ldots,k\} \setminus Z;\ \tilde{g}_i = -1\}\right)\right)\right)$$

  gewählt werden, damit die „schlechtest-mögliche“ Spalte de-systematisiert wird, welche man später unter geeigneten Umständen nicht mehr systematisieren muß.

Im Algorithmus 5.3 wird zur Einsparung numerischer Operationen ein Index *zmax* verwendet, der so verwaltet wird, daß $\mu(zmax+1),\ldots,\mu(n)$ unzulässige (das heißt $\mu(.) > k$) oder schon verwendete Pivot-Indizes sind.

Nach Konstruktion gelten alle Aussagen von Lemma 5.4 auf Seite 97 auch für Algorithmus 5.3, wobei $\tilde{G}$ implizit durch die charakterisierenden Mengen $\tilde{J}_j$, $j = 1,\ldots,n$, gegeben ist.

Quasi-Systematisierung (Variante 3): **Eingang** G, μ, y; **Ausgang** $\tilde{J}$, $\tilde{u}^0$, α, a, ρ, ρ^{-1};

```
Z := ∅; zmax := n;                                  Einheitsvektoren-Index und höchster freier Index
R := ∅;                                             Restindizes
a := 0;   m := 1;
solange |Z| < k:
    j := μ(m); gefunden := false;
    falls (j ≤ k) und (j ∉ Z):                      Die j-te Spalte ist der j-te Einheitsvektor
        q := j; gefunden := true;
    sonst:
        für i = 1,...,k:                            Kopie der j-ten Spalte
            g̃_i := G_ij;
        für b = 1,...,a:                            Systematisiere die j-te Spalte
            (p,q) := α_b;                           q-te Zeile zur p-ten addieren
            g̃_p := g̃_p ⊕ g̃_q;
        r := zmax;
        solange (gefunden = false) und (r ≥ 1):     Suche Pivot-Element
            q := μ(r);
            falls (q > k) oder (q ∈ Z):             q ist unzulässig oder verwendet
                falls (zmax > 1) und (zmax = r):    Anpassung von zmax
                    zmax := zmax − 1;
                    solange (zmax > 1) und ((μ(zmax) > k) oder (μ(zmax) ∈ Z)):
                        zmax := zmax − 1;
                r := min(r − 1, zmax);
            sonst:
                falls g̃_q = −1:
                    gefunden := true;
                    für p ∈ {1,...,k} \ {q}:
                        falls g̃_p = −1:
                            a := a + 1;
                            α_a := (p,q);           q-te Zeile zur p-ten addieren
                sonst:
                    r := r − 1;
```

Fortsetzung folgt...

Algorithmus 5.3: *Spezielle Quasi-Systematisierung eines systematischen (n,k)-Blockcodes*

```
...                                                          Fortsetzung
    falls gefunden = true:
        Z := Z ∪ {q};
        ũ_q^0 := π(y_j);                                     Startpunktgenerierung
        J̃_j := {q};                                          Einpunktige Indexmenge
        ρ(|Z|) := q;
        ρ^{-1}(q) := |Z|;
    sonst:                                                   Lineare Abhängigkeiten
        R := R ∪ {j};
    m := m + 1;
für j ∈ R ∪ {μ(m), ..., μ(n)}:                               Unsystematische Spalten
    für i = 1, ..., k:                                       Kopie der j-ten Spalte
        g̃_i := G_ij;
    für b = 1, ..., a:                                       Behandle die j-te Spalte
        (p, q) := α_b;                                       q-te Zeile zur p-ten addieren
        g̃_p := g̃_p ⊕ g̃_q;
    J̃_j := ∅;
    für p = 1, ..., k:
        falls g̃_p = −1:
            J̃_j := J̃_j ∪ {p};
```

Algorithmus 5.3: *Spezielle Quasi-Systematisierung eines systematischen (n,k)-Blockcodes — Fortsetzung*

5.4.5 Rücktransformation in den Originalcode

Nach Bestimmung eines Decodierungsergebnisses für den transformierten Code (siehe nächsten Abschnitt), muß dieses noch für den Originalcode rücktransformiert werden.

Sei also $\tilde{u} \in \{\pm 1\}^k$ ein kryptocodierte Codewort, für welches mit der Generatormatrix $\tilde{G}$ gilt:

$$c^\top = \tilde{u}^\top \tilde{G}.$$

Das mit Original-Generatormatrix G codierte u berechnet sich dann sehr einfach wie folgt:

$$\begin{aligned} u^\top &= \tilde{u}^\top A^{-1} \\ &= \tilde{u}^\top A_{\alpha_a} A_{\alpha_{a-1}} \ldots A_{\alpha_2} A_{\alpha_1}. \end{aligned}$$

Dabei bewirkt $u^\top A_{pq}$ die Addition des p-ten Elements von u zum q-ten Element von u und es ergibt sich Algorithmus 5.4.

Rücktransformation: **Eingang** $\tilde{u}$, α, a; **Ausgang** u;

$u := \tilde{u}$;
für $m = a$ **abwärts bis** 1:
    $(p,q) := \alpha_m$;
    $u_q := u_q \oplus u_p$;

Algorithmus 5.4: *Rücktransformation des Decodierungsergebnisses*

5.5 Branch-and-Bound Verfahren (BB)

5.5.1 Vorgehensweise

Zur Bewertung von Decodierungskandidaten $\tilde{u} \in \{\pm 1\}^k$ bezüglich des quasi-systematisierten Codes muß die Zielfunktion (5.1) entsprechend angepaßt werden. Mit $u^\top = \tilde{u}^\top A^{-1}$ könnte man eine angepaßte Zielfunktion so definieren, daß $\tilde{F}(\tilde{u}) = F(u)$ wäre. Die Funktionen $\tilde{F}(\tilde{u}) := \frac{1}{2}F(u) - const$ besitzen die identischen Minimierer und können bei einer geschickten Wahl von *const* mit einer kleineren Zahl von Operationen ausgewertet werden (wie im folgenden zu sehen).

Die Funktion $\tilde{F}$ wird so zerlegt, daß sich untere Schranken $\tilde{F}^b(\tilde{u})$ für den Funktionswert $\tilde{F}(\tilde{u})$ angeben lassen, wenn nur die ersten b Komponenten des Arguments $\tilde{u}$ festgelegt sind, wobei die Reihenfolge der Bits nach der Sortierung[7] ρ gewählt wird. Diese Schranken können dann im Branch-and-Bound Algorithmus zur Verwerfung von Unterbäumen eingesetzt werden.

[7] Die Sortierung ρ wurde vom Algorithmus 5.1 auf Seite 98 beziehungsweise Algorithmus 5.2 auf Seite 100 oder gegebenenfalls Algorithmus 5.3 auf Seite 102 erzeugt. Zu den Eigenschaften von ρ siehe Lemma 5.4 auf Seite 97.

Definition 5.5 (Transformierte Soft-Decision Zielfunktion)
Voraussetzung 5.1 auf Seite 92 sei erfüllt. $y \in \mathbb{R}^n$ sei eine Realisierung der Kanalausgabe und Algorithmus 5.2 auf Seite 100 sei durchgeführt worden.

(i) *Die transformierte Soft-Decision Zielfunktion ist definiert als*

$$\begin{aligned} \tilde{F} : \{\pm 1\}^k &\to \mathbb{R}, \\ \tilde{u} &\mapsto \frac{1}{2}\sum_{j=1}^{n}\left(\bigoplus_{i\in\tilde{J}_j}\tilde{u}_i - y_j\right)^2 - \frac{1}{2}\sum_{j=1}^{n}\left(|y_j|-1\right)^2 \end{aligned} \tag{5.11}$$

(ii) *Definiere*

$$\begin{aligned} \gamma &:= \sum_{j=1}^{n}|y_j|, \\ s_j(\tilde{u}) &:= y_j \bigoplus_{i\in\tilde{J}_j}\tilde{u}_i \quad \text{für alle } \tilde{u}\in\{\pm 1\}^k \text{ und } j = 1,\ldots,n. \end{aligned}$$

(iii) *Für $0 \le b \le k$, $1 \le j \le n$, $\tilde{u} \in \{\pm 1\}^k$ definiere*

$$\begin{aligned} \max(\rho^{-1}(\tilde{J}_j)) &:= \max\left(\{\rho^{-1}(i);\ i\in\tilde{J}_j\}\right), \\ s_j^b(\tilde{u}) &:= \begin{cases} y_j \bigoplus\limits_{i\in\tilde{J}_j}\tilde{u}_i, & \text{für } \max(\rho^{-1}(\tilde{J}_j)) \le b, \\ |y_j|, & \text{sonst.} \end{cases} \end{aligned}$$

(iv) *Definiere für $0 \le b \le k$, $\tilde{u} \in \{\pm 1\}^k$ „untere Schranken“*

$$\tilde{F}^b(\tilde{u}) := \gamma - \sum_{j=1}^{n} s_j^b(\tilde{u}).$$

(v) *Definiere für $0 \le b \le k$, $\tilde{u} \in \{\pm 1\}^k$*

$$\begin{aligned} K_b &:= \left\{ j\in\{1,\ldots,n\};\ \max(\rho^{-1}(\tilde{J}_j)) = b\right\}, \\ \hat{K}_b &:= \bigcup_{i=1}^{b} K_i, \\ L^b &:= \sum_{j\in\{1,\ldots,n\}\setminus\hat{K}_b}|y_j|, \\ \gamma^b &:= \gamma - L^b, \\ S^b(\tilde{u}) &:= \sum_{j\in\hat{K}_b} s_j(\tilde{u}), \\ \Delta^b(\tilde{u}) &:= \sum_{j\in K_b} s_j(\tilde{u}) = \sum_{j\in K_b} y_j \bigoplus_{i\in\tilde{J}_j}\tilde{u}_i. \end{aligned}$$

Mit den obigen Definitionen läßt sich nun der folgende Satz beweisen.

Satz 5.6 (Branch-and-Bound Schranken)
Voraussetzung 5.1 auf Seite 92 sei erfüllt. $y \in \mathbb{R}^n$ sei eine Realisierung der Kanalausgabe und Algorithmus 5.2 auf Seite 100 sei durchgeführt worden.

(i) *Für alle $\tilde{u} \in \{\pm 1\}^k$ gilt mit $u := A^{-\top}\tilde{u}$, daß*

$$\tilde{F}(\tilde{u}) = \frac{1}{2}F(u) - \frac{1}{2}\sum_{j=1}^{n}\big(|y_j|-1\big)^2.$$

(ii) *$\tilde{u} \in \{\pm 1\}^k$ ist genau dann ein Minimierer von $\tilde{F}$, wenn $u := A^{-\top}\tilde{u} \in \{\pm 1\}^k$ ein Minimierer von F ist.*

(iii) *Für alle $\tilde{u} \in \{\pm 1\}^k$ gilt*

$$\tilde{F}(\tilde{u}) = \gamma - \sum_{j=1}^{n} s_j(\tilde{u}).$$

(iv) *Für alle $\tilde{u} \in \{\pm 1\}^k$ und alle $j = 1,\ldots,n$ gilt*

$$|y_j| = s_j^0(\tilde{u}) \geq s_j^1(\tilde{u}) \geq \ldots \geq s_j^k(\tilde{u}) = s_j(\tilde{u}).$$

(v) *Für alle $\tilde{u} \in \{\pm 1\}^k$ gilt*

$$0 = \tilde{F}^0(\tilde{u}) \leq \tilde{F}^1(\tilde{u}) \leq \ldots \leq \tilde{F}^k(\tilde{u}) = \tilde{F}(\tilde{u}).$$

(vi) *Für alle $\tilde{u} \in \{\pm 1\}^k$ gilt rekursiv*

$$\begin{aligned} S^0(\tilde{u}) &= 0, \\ S^b(\tilde{u}) &= S^{b-1}(\tilde{u}) + \Delta^b(\tilde{u}), \quad \text{für } 1 \leq b \leq k, \\ \gamma^0 &= 0, \\ \gamma^b &= \gamma^{b-1} + \sum_{j \in K_b} |y_j|, \quad \text{für } 1 \leq b \leq k. \end{aligned}$$

(vii) *Für $\tilde{u} \in \{\pm 1\}^k$ und ein fest gewähltes $b \in \{1,\ldots,k\}$ betrachte man $\tilde{u}^+, \tilde{u}^- \in \{\pm 1\}^k$ mit $\tilde{u}_i^+ = \tilde{u}_i$ und $\tilde{u}_i^- = \tilde{u}_i$ für $i \neq \rho(b)$. Es sei $\tilde{u}_{\rho(b)}^+ := +1$ und $\tilde{u}_{\rho(b)}^- := -1$. Mit $\delta_S := \Delta^b(\tilde{u}^-)$ gilt*

$$\begin{aligned} S^b(\tilde{u}^-) &= S^{b-1}(\tilde{u}) + \delta_S, \\ S^b(\tilde{u}^+) &= S^{b-1}(\tilde{u}) - \delta_S. \end{aligned}$$

(viii) *Für alle $\tilde{u} \in \{\pm 1\}^k$ gilt*

$$\tilde{F}^b(\tilde{u}) = \gamma^b - S^b(\tilde{u}). \tag{5.12}$$

(ix) *Falls $\tilde{F}(\tilde{u}) = 0$ für ein $\tilde{u} \in \{\pm 1\}^k$, dann ist $\tilde{u}$ ein Minimierer von $\tilde{F}$.*

Beweis. Ad (i): Mit (5.1) und (5.11) folgt die Aussage unmittelbar.

Ad (ii): Über die Matrix A beziehungsweise $A^{-\top}$ wird ein Automorphismus auf $\{\pm 1\}^k$ definiert. Da F und $\tilde{F}$ bis auf diesen Automorphismus, eine additive Verschiebung und eine positive Skalierung identisch sind, sind auch die Mengen der Minimierer bis auf den Automorphismus identisch.

Ad (iii): Es gilt

$$\begin{aligned}\tilde{F}(\tilde{u}) &= \frac{1}{2}\sum_{j=1}^{n}\left(1+y_j^2-2y_j\bigoplus_{i\in\tilde{J}_j}\tilde{u}_i\right)-\frac{1}{2}\sum_{j=1}^{n}\left(y_j^2-2|y_j|+1\right)\\ &= \underbrace{\sum_{j=1}^{n}|y_j|}_{=\gamma}-\sum_{j=1}^{n}\underbrace{y_j\bigoplus_{i\in\tilde{J}_j}\tilde{u}_i}_{=s_j(\tilde{u})}=\gamma-\sum_{j=1}^{n}s_j(\tilde{u}).\end{aligned}$$

Ad (iv): Die Aussage folgt sofort mit Definition 5.5 (iii).

Ad (v): Mit

$$\tilde{F}^b(\tilde{u})=\gamma-\sum_{j=1}^{n}s_j^b(\tilde{u}),$$

und (iv) folgt

$$0=\gamma-\sum_{j=1}^{n}|y_j|=\tilde{F}^0(\tilde{u})\le\tilde{F}^1(\tilde{u})\le\ldots\le\tilde{F}^k(\tilde{u})=\gamma-\sum_{j=1}^{n}s_j(\tilde{u})=\tilde{F}(\tilde{u}).$$

Ad (vi): Es gilt für $1\le b\le k$

$$\begin{aligned}S^b(\tilde{u})&=\sum_{j\in\hat{K}_b}s_j(\tilde{u})=S^{b-1}(\tilde{u})+\sum_{j\in K_b}s_j(\tilde{u})=S^{b-1}(\tilde{u})+\Delta^b(\tilde{u}),\\ \gamma^b&=\gamma-L^b=\gamma-\sum_{j\in\{1,\ldots,n\}\setminus\hat{K}_b}|y_j|=\gamma-L^{b-1}+\sum_{j\in K_b}|y_j|=\gamma^{b-1}+\sum_{j\in K_b}|y_j|.\end{aligned}$$

Ad (vii): Nach Definition ist $\rho(b)\in\tilde{J}_j$ für alle $j\in K_b$. Somit gilt $s_j(\tilde{u}^-)=-s_j(\tilde{u}^+)$ für alle $j\in K_b$ und weiter

$$\Delta^b(\tilde{u}^-)=-\Delta^b(\tilde{u}^+).$$

Dann gilt mit $\delta_S=\Delta^b(\tilde{u}^-)$

$$\begin{aligned}S^b(\tilde{u}^-)&=S^{b-1}(\tilde{u})+\delta_S,\\ S^b(\tilde{u}^+)&=S^{b-1}(\tilde{u})-\delta_S.\end{aligned}$$

Ad (viii):

$$\begin{aligned}\tilde{F}^b(\tilde{u})&=\gamma-\sum_{j=1}^{n}s_j^b(\tilde{u})=\gamma-\sum_{j\in\{1,\ldots,n\}\setminus\hat{K}_b}s_j^b(\tilde{u})-\sum_{j\in\hat{K}_b}s_j^b(\tilde{u})\\ &=\gamma-\sum_{j\in\{1,\ldots,n\}\setminus\hat{K}_b}|y_j|-\sum_{j\in\hat{K}_b}s_j(\tilde{u})=\gamma-L^b-S^b(\tilde{u})\\ &=\gamma^b-S^b(\tilde{u}).\end{aligned}$$

Ad (ix): Mit (v) ist $\tilde{F}$ durch 0 nach unten beschränkt. Somit ist jedes $\tilde{u} \in \{\pm 1\}^k$ mit $\tilde{F}(\tilde{u}) = 0$ ein globaler Minimierer von $\tilde{F}$.

□

In Gleichung (5.12) summiert $S^b(\tilde{u})$ diejenigen $s_j(\tilde{u})$, die bei Kenntnis der Bits $\tilde{u}_{\rho(1)}, \ldots, \tilde{u}_{\rho(b)}$ berechnet werden können und γ^b summiert die restlichen Abschätzungen (unabhängig von $\tilde{u}$).

Mit (vi) lassen sich die benötigten Funktionsteile in einfacher Weise rekursiv berechnen und (vii) erlaubt eine weitere Einsparung von numerischen Operationen durch Verwendung gemeinsamer Formelteile.

5.5.2 Hilfsalgorithmen

Indexmengen-Berechnung: **Eingang** ρ^{-1}, $\tilde{J}$; **Ausgang** K;

für $b = 1, \ldots, k$:
    $K_b := \emptyset$;
für $j = 1, \ldots, n$:
    $b := 0$;
    für $i \in \tilde{J}_j$:
        $h := \rho^{-1}(i)$;
        falls $h > b$:
            $b := h$;
    $K_b := K_b \cup \{j\}$;

Algorithmus 5.5: *Berechnung der Indexmengen* K_b

Berechnung $\Delta^b(\tilde{u})$**:** **Eingang** b, K, $\tilde{J}$, $\tilde{u}$; **Ausgang** $\Delta^b(\tilde{u})$;

$\delta_S := 0$;
für $j \in K_b$:
    $h := +1$;
    für $i \in \tilde{J}_j$:
        $h := h \oplus \tilde{u}_i$;
    $\delta_S := \delta_S + y_j h$;
$\Delta^b(\tilde{u}) := \delta_S$;

Algorithmus 5.6: *Berechnung der Werte* $\Delta^b(\tilde{u})$

Vor der Formulierung des Branch-and-Bound Verfahrens betrachten wir zwei benötigte Hilfsalgorithmen. Algorithmus 5.5 berechnet die Indexmengen K_b, $b = 1, \ldots, k$, und Algorithmus 5.6 berechnet jeweils den Update $\Delta^b(\tilde{u})$ zu $\tilde{u} \in \{\pm 1\}^k$ und $b = 1, \ldots, k$.

5.5.3 Branch-and-Bound Algorithmus (Variante 1)

Nun kann der Branch-and-Bound Algorithmus formuliert werden. Die in Definition 5.5 auf Seite 105 definierten unteren Schranken $\tilde{F}^b(\tilde{u})$ von $\tilde{F}(\tilde{u})$ dienen dem Beschneiden des Binärbaums, da jene Unterbäume nicht betrachtet werden müssen, für die die untere Schranke bereits größer als der bislang beste Funktionswert ist. Die noch zu bearbeitenden Teile des Binärbaums werden mit Hilfe eines Kellerspeichers verwaltet.

Die Menge M mit Tupeln $(b,\tilde{u},F,S)$, $b \in \{1,\ldots,k-1\}$, $\tilde{u} \in \{\pm 1\}^k$, $F,S \in \mathbb{R}$, als Elementen repräsentiere einen FILO-Keller (**F**irst **I**n, **L**ast **O**ut).

Algorithmus 5.7 beschreibt dann ein Branch-and-Bound Verfahren zur Minimierung der transformierten Zielfunktion $\tilde{F}$.

Bemerkung: Aus numerischen Gründen sollte in der Implementierung statt des Vergleichs $F_{\min} = 0$ besser $F_{\min} < \textit{schranke}$ getestet werden.

Branch-and-Bound (Variante 1): **Eingang** y, $\tilde{u}^0$, K, ρ; **Ausgang** $\tilde{u}_{\min}$;

$\tilde{u}_{\min} = \tilde{u}^0 \in \{\pm 1\}^k$ sei der „Startpunkt“; *siehe Startpunktgenerierung*
$\gamma^0 := 0$;
für $b = 1,\ldots,k$:
    $h := 0$;
    für $j \in K_b$:
        $h := h + |y_j|$;
    $\gamma^b := \gamma^{b-1} + h$;
$S := 0$;
für $b = 1,\ldots,k$:
    $S := S + \Delta^b(\tilde{u}_{\min})$;
$F_{\min} := \gamma^k - S$; *Zielfunktionswert am Startpunkt*
falls $F_{\min} = 0$:
    Stop mit Ergebnis $\tilde{u}_{\min}$; *Minimierer ist gefunden*
$M := \emptyset$;
$(b,\tilde{u},F,S) := (0,\tilde{u},\gamma^0,0)$; *$\tilde{u}$ beliebig*

Fortsetzung folgt...

Algorithmus 5.7: *Branch-and-Bound Verfahren (Variante 1)*

```
...                                                              Fortsetzung
solange (b < k):                                                 Hauptiteration: Branch and Bound
    falls F < F_min:                                             Das Tupel (b, ũ, F, S) wird betrachtet
        b := b + 1;
        ũ_ρ(b) := −1;                                            ũ wird zu ũ⁻ erweitert
        δ_S := Δ^b(ũ);
        S⁻ := S + δ_S;   S⁺ := S − δ_S;
        F⁻ := γ^b − S⁻;   F⁺ := γ^b − S⁺;                        Gewichte der Nachfolgeknoten
        falls min(F⁻, F⁺) < F_min:
            falls F⁻ < F⁺:                                       also falls δ_S > 0
                falls b < k:                                     Zwischenknoten
                    falls F⁺ < F_min:
                        ũ_ρ(b) := +1;                            ũ wird zu ũ⁺
                        push(M, (b, ũ, F⁺, S⁺));                 Alternative in den Keller
                        ũ_ρ(b) := −1;                            ũ wird zu ũ⁻
                    F := F⁻;   S := S⁻;
                sonst:                                           Blatt
                    ũ_min := ũ;   F_min := F⁻;                   Bessere Decodierung gefunden
                    falls F_min = 0:
                        Stop mit Ergebnis ũ_min;                 Minimierer ist gefunden
            sonst:
                falls b < k:                                     Zwischenknoten
                    falls F⁻ < F_min:
                        push(M, (b, ũ, F⁻, S⁻));                 Alternative in den Keller
                    ũ_ρ(b) := +1;   F := F⁺;   S := S⁺;
                sonst:                                           Blatt
                    ũ_ρ(b) := +1;   ũ_min := ũ;   F_min := F⁺;   Bessere Decodierung gefunden
                    falls F_min = 0:
                        Stop mit Ergebnis ũ_min;                 Minimierer ist gefunden
        sonst:
            b := k;                                              Beende die Suche in diesem Zweig
    sonst:
        b := k;                                                  Beende die Suche in diesem Zweig
    falls (b = k) und M ≠ ∅:
        (b, ũ, F, S) := pop(M);                                  Alternative aus dem Keller
```

Algorithmus 5.7: *Branch-and-Bound Verfahren (Variante 1) — Fortsetzung*

Lemma 5.7 (Branch-and-Bound Algorithmus)
Voraussetzung 5.1 auf Seite 92 sei erfüllt. Sei $y \in \mathbb{R}^n$ eine Realisierung der Kanalausgabe und Algorithmus 5.2 auf Seite 100 sei durchgeführt worden[8]*. Dann terminiert der Algorithmus 5.7 und der berechnete Punkt $\tilde{u}_{\min}$ ist ein Minimierer der transformierten Zielfunktion $\tilde{F}$. Insbesondere ist $\hat{u} := A^{-\top}\tilde{u}_{\min}$ ein Minimierer der Soft-Decision Zielfunktion.*

Beweis. Die Terminierung des Verfahrens ergibt sich nach Konstruktion von Algorithmus 5.7, da im numerisch aufwendigsten Fall alle 2^k Codewörter betrachtet werden müssen und der Algorithmus danach terminiert. Ebenso ist $\tilde{u}_{\min}$ nach Konstruktion ein Minimierer von $\tilde{F}$. Mit Satz 5.6 (ii) ist $\hat{u} := A^{-\top}\tilde{u}_{\min}$ dann ein Minimierer der Soft-Decision Zielfunktion F.

□

Abhängig von der „Güte“ des Demodulationsergebnisses $y \in \mathbb{R}^n$ und des verwendeten binären linearen (n,k)-Blockcodes wird Algorithmus 5.7 mit mehr oder weniger Iterationen terminieren. Zu jedem Zeitpunkt des Verfahrens kann aber ein bislang bester Decodierungskandidat (jeweils $\tilde{u}_{\min}$) ausgegeben werden. Daher kann man den Algorithmus bei Erreichen von iterativen Schranken (maximale Iterationszahl) oder zeitlichen Schranken (Timer) mit dem bislang besten Ergebnis stoppen, wenn ein Echtzeitverhalten benötigt wird. In Abschnitt 5.7 wird sogar eine algorithmische Variante vorgestellt, die den Startpunkt als Decodierungsergebnis verwendet.

5.5.4 Branch-and-Bound Algorithmus (Variante 2)

Betrachtet man die Nutzung des Kellerspeichers in Algorithmus 5.7, so erkennt man, daß die Komponenten des aktuellen u's mit den jeweiligen Kellerelementen bis zum $\rho(b)$'ten Element übereinstimmen.

Daher bietet sich eine effektivere Implementierung des Kellerspeichers an, mit der sich viele Operationen einsparen lassen.

Man betrachte dazu die Menge M mit Tupeln (b,β,F,S), $b \in \{1,\ldots,k-1\}$, $\beta \in \{\pm 1\}$, $F,S \in \mathbb{R}$, als Elementen, die einen FILO-Keller (**F**irst **I**n, **L**ast **O**ut) repräsentiere.

Somit ergibt sich mit Algorithmus 5.8 ein zu Algorithmus 5.7 in Bezug auf Ein- und Ausgabe gleichwertiges Verfahren, welches weniger numerische Operationen benötigt.

Bemerkung: Aus numerischen Gründen sollte wiederum in der Implementierung statt des Vergleichs $F_{\min} = 0$ besser $F_{\min} < \mathit{schranke}$ getestet werden.

Die Aussagen von Lemma 5.7 auf Seite 111 sind auf Algorithmus 5.8 in gleicher Weise wie auf Algorithmus 5.7 anwendbar.

[8] Bei systematischen Codes kann auch Algorithmus 5.3 auf Seite 102 als Alternative durchgeführt worden sein.

Branch-and-Bound (Variante 2): **Eingang** y, $\tilde{u}^0$, K, ρ; **Ausgang** $\tilde{u}_{\min}$;

$\tilde{u}_{\min} = \tilde{u}^0 \in \{\pm 1\}^k$ sei der „Startpunkt“; *siehe Startpunktgenerierung*
$\gamma^0 := 0$;
für $b = 1, \ldots, k$:
    $h := 0$;
    für $j \in K_b$:
        $h := h + |y_j|$;
    $\gamma^b := \gamma^{b-1} + h$;
$S := 0$;
für $b = 1, \ldots, k$:
    $S := S + \Delta^b(\tilde{u}_{\min})$;
$F_{\min} := \gamma^k - S$; *Zielfunktionswert am Startpunkt*
falls $F_{\min} = 0$:
    Stop mit Ergebnis $\tilde{u}_{\min}$; *Minimierer ist gefunden*
$M := \emptyset$;
$(b, \beta, F, S) := (0, \beta, \gamma^0, 0)$; β *beliebig*
$b_{\text{last}} := 1$; *Pegelstand*
solange $(b < k)$: *Hauptiteration: Branch and Bound*
    falls $F < F_{\min}$: *Das Tupel* $(b, \tilde{u}, F, S)$ *wird betrachtet*
        $b := b + 1$;
        $\tilde{u}_{\rho(b)} := -1$; *$\tilde{u}$ wird zu $\tilde{u}^-$ erweitert*
        $\delta_S := \Delta^b(\tilde{u})$;
        $S^- := S + \delta_S$; $S^+ := S - \delta_S$;
        $F^- := \gamma^b - S^-$; $F^+ := \gamma^b - S^+$; *Gewichte der Nachfolgeknoten*
Fortsetzung folgt...

Algorithmus 5.8: *Branch-and-Bound Verfahren (Variante 2)*

```
...                                                              Fortsetzung
    falls min(F⁻,F⁺) < F_min:
        falls F⁻ < F⁺:                                           also falls δ_S > 0
            falls b < k:                                         Zwischenknoten
                falls F⁺ < F_min:
                    push(M,(b,+1,F⁺,S⁺));                        Alternative in den Keller
                F := F⁻;   S := S⁻;
            sonst:                                               Blatt
                F_min := F⁻;                                     Bessere Decodierung gefunden
                für i = b_last,...,k:
                    ũ_{min,ρ(i)} := ũ_{ρ(i)};
                falls F_min = 0:
                    Stop mit Ergebnis ũ_min;                     Minimierer ist gefunden
                b_last := k;
        sonst:
            falls b < k:                                         Zwischenknoten
                falls F⁻ < F_min:
                    push(M,(b,−1,F⁻,S⁻));                        Alternative in den Keller
                ũ_{ρ(b)} := +1;   F := F⁺;   S := S⁺;
            sonst:                                               Blatt
                ũ_{ρ(b)} := +1;   F_min := F⁺;                   Bessere Decodierung gefunden
                für i = b_last,...,k:
                    ũ_{min,ρ(i)} := ũ_{ρ(i)};
                falls F_min = 0:
                    Stop mit Ergebnis ũ_min;                     Minimierer ist gefunden
                b_last := k;
    sonst:
        b := k;                                                  Beende die Suche in diesem Zweig
sonst:
    b := k;                                                      Beende die Suche in diesem Zweig
falls (b = k) und M ≠ ∅:
    (b,β,F,S) :=pop(M);                                          Alternative aus dem Keller
    ũ_{ρ(b)} := β;
    falls b < b_last:
        b_last := b;
```

Algorithmus 5.8: *Branch-and-Bound Verfahren (Variante 2) — Fortsetzung*

5.6 Schematische Zusammenfassung

Das folgende Schema veranschaulicht zusammenfassend den Ablauf des vorgestellten Soft-Decision Decodierungsverfahrens und das Zusammenspiel der algorithmischen Komponenten.

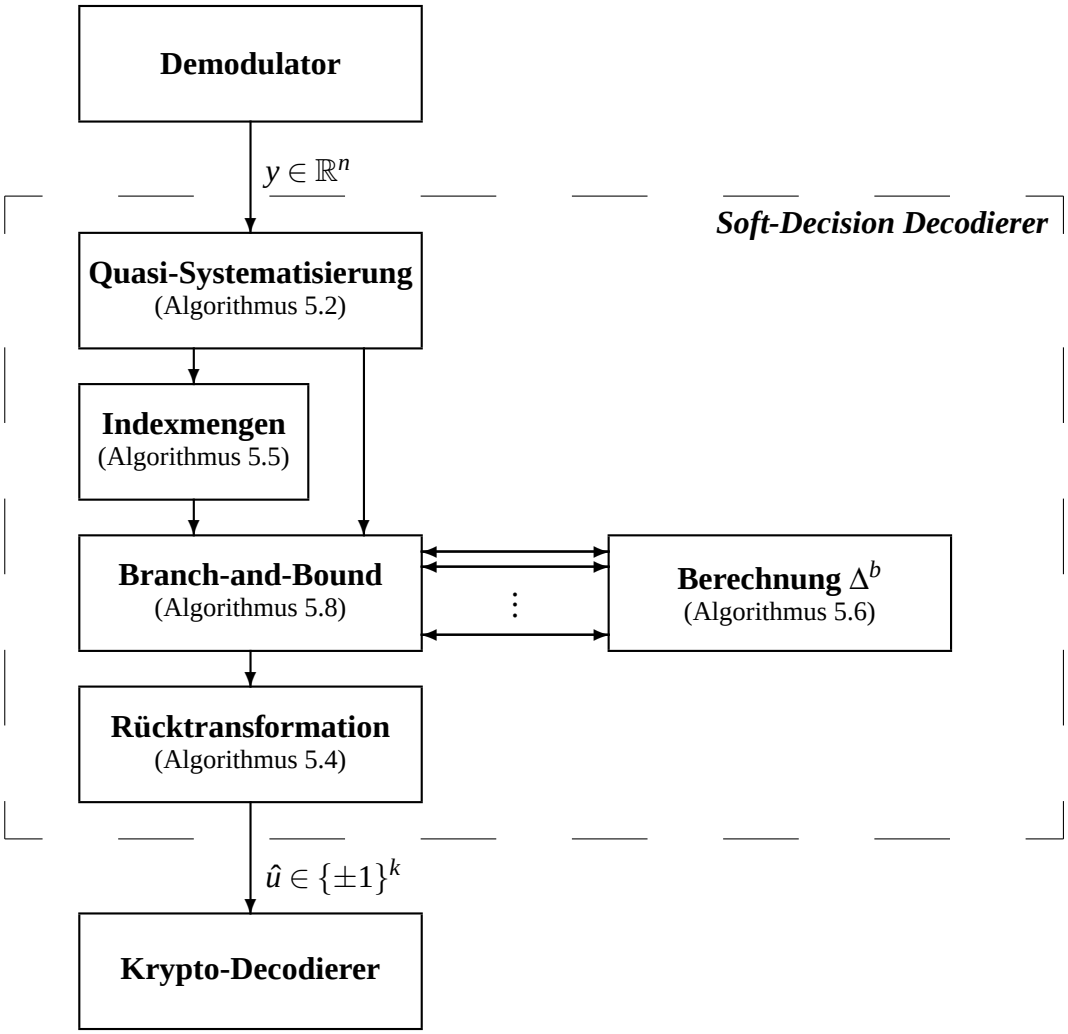

Abbildung 5.5: *Schematische Darstellung des Zusammenspiels der Algorithmen*

5.7 Startpunkt-Verfahren

In Algorithmus 5.2 auf Seite 100 beziehungsweise Algorithmus 5.1 auf Seite 98 wurde ein „Startpunkt“ $\tilde{u}^0$ berechnet, der als erster Decodierungskandidat im Branch-and-Bound Verfahren verwendet wird. In numerischen Tests[9] hat sich gezeigt, daß dieser „Startpunkt“ bereits sehr gute fehlerminimale Eigenschaften besitzt. Da sich ein „Startpunkt“ ohne anschließendes Branch-and-Bound Verfahren mit geringerem numerischen Aufwand berechnen läßt, eignet sich dieser als Decodierungsergebnis bei zeitkritischen Anwendungen.

Algorithmus 5.9 ist eine Zusammenfassung von Algorithmus 5.2 auf Seite 100 und Algorithmus 5.4 auf Seite 104, wobei auf alle Vorbereitungen für ein Branch-and-Bound Verfahren verzichtet

[9] siehe Kapitel 6

wird. Im Verfahren wird wieder die Projektionsabbildung

$$\pi : \mathbb{R} \to \{\pm 1\}, \qquad x \mapsto \begin{cases} +1, & \text{für } x \geq 0, \\ -1, & \text{für } x < 0, \end{cases}$$

verwendet.

„Startpunkt"-Verfahren: **Eingang** G, μ, y; **Ausgang** $\hat{u}$;

```
Z := ∅;                                              Einheitsvektoren-Index
a := 0;   m := 1;
solange |Z| < k:
    j := μ(m);
    für i = 1,...,k:                                 Kopie der j-ten Spalte
        g̃_i := G_ij;
    für b = 1,...,a:                                 Systematisiere die j-te Spalte
        (p,q) := α_b;                                q-te Zeile zur p-ten addieren
        g̃_p := g̃_p ⊕ g̃_q;
    falls {i ∈ {1,...,k} \ Z; g̃_i = −1} ≠ ∅:
        q := min({i ∈ {1,...,k} \ Z; g̃_i = −1});
        Z := Z ∪ {q};
        û_q := π(y_j);                               Startpunktgenerierung
        für p ∈ {1,...,k} \ {q}:
            falls g̃_p = −1:
                a := a + 1;
                α_a := (p,q);                        q-te Zeile zur p-ten addieren
    m := m + 1;
für m = a abwärts bis 1:                             Rückrechnung in den Originalcode
    (p,q) := α_m;
    û_q := û_q ⊕ û_p;
```

Algorithmus 5.9: *„Startpunkt"-Verfahren ohne Branch-and-Bound*

Mit Lemma 5.4 auf Seite 97 terminiert auch Algorithmus 5.9 als Spezialfall von Algorithmus 5.2 beziehungsweise Algorithmus 5.1.

Der numerische Aufwand zur Berechnung von $\hat{u}$ beträgt

- $\mathrm{O}(n \log n)$ Fließkomma-Operationen (Sortierung μ) und
- im worst case $\mathrm{O}\left(k^3\right)$ Binär-Operationen und im best case $\mathrm{O}\left(k^2\right)$ Binär-Operationen.

Der entsprechende Algorithmus 5.2 auf Seite 100 benötigt $\mathrm{O}\left(k^2 n\right)$ Binär-Operationen. Zudem ist der konstante Faktor in der Komplexitätsbetrachtung in der puren Startpunkt-Methode noch

deutlich geringer, da weder die am aufwendigsten zu behandelnden „unsystematischen“ Spalten von G beziehungsweise $\tilde{G}$ noch die Indexmengen $\tilde{J}_j$ betrachtet werden müssen.

„Startpunkt“-Verfahren (Variante 2): **Eingang** G, μ, y; **Ausgang** $\hat{u}$;

```
Z := ∅; zmax := n;                                  Einheitsvektoren-Index und höchster freier Index
a := 0;   m := 1;
solange |Z| < k:
    j := μ(m); gefunden := false;
    falls (j ≤ k) und (j ∉ Z):                      Die j-te Spalte ist der j-te Einheitsvektor
        q := j; gefunden := true;
    sonst:
        für i = 1,...,k:                            Kopie der j-ten Spalte
            g̃_i := G_ij;
        für b = 1,...,a:                            Systematisiere die j-te Spalte
            (p,q) := α_b;                           q-te Zeile zur p-ten addieren
            g̃_p := g̃_p ⊕ g̃_q;
        r := zmax;
        solange (gefunden = false) und (r ≥ 1):     Suche Pivot-Element
            q := μ(r);
            falls (q > k) oder (q ∈ Z):             q ist unzulässig oder verwendet
                falls (zmax > 1) und (zmax = r):    Anpassung von zmax
                    zmax := zmax − 1;
                    solange (zmax > 1) und ((μ(zmax) > k) oder (μ(zmax) ∈ Z)):
                        zmax := zmax − 1;
                r := min(r − 1, zmax);
            sonst:
                falls g̃_q = −1:
                    gefunden := true;
                    für p ∈ {1,...,k} \ {q}:
                        falls g̃_p = −1:
                            a := a + 1;
                            α_a := (p,q);           q-te Zeile zur p-ten addieren
                sonst:
                    r := r − 1;
```

Fortsetzung folgt...

Algorithmus 5.10: „*Startpunkt*“*-Verfahren für systematische* (n,k)*-Blockcodes*

```
...                                                          Fortsetzung
    falls gefunden = true:
        Z := Z ∪ {q};
        û_q := π(y_j);                                       Startpunktgenerierung
    m := m + 1;
für m = a abwärts bis 1:                                     Rückrechnung in den Originalcode
    (p, q) := α_m;
    û_q := û_q ⊕ û_p;
```

Algorithmus 5.10: *„Startpunkt"-Verfahren für systematische (n,k)-Blockcodes — Fortsetzung*

Für den wichtigen Spezialfall systematischer Codes läßt sich analog zur Quasi-Systematisierung auch eine Spezialisierung von Algorithmus 5.9 angeben. Algorithmus 5.10 berücksichtigt die Gegebenheiten bei systematischen Codes[10] und benötigt wiederum deutlich weniger numerische Operationen gegenüber der allgemeinen Variante. Algorithmus 5.10 kann auch als Spezialfall von Algorithmus 5.3 auf Seite 102 angesehen werden.

5.8 Fehlererkennung

5.8.1 Bewertung des Decodierergebnisses

Die Fragestellung dieser Arbeit dreht sich um die fehlerminimale Decodierung von binären linearen (n,k)-Blockcodes. Bei besonders fehlerkritischen Anwendungen ist aber auf Kosten der Decodierungsrate[11] bisweilen ein Verwerfen des Decodierungsergebnisses gewünscht, wenn die Fehlerfreiheit nicht in hohem Maße garantiert werden kann. Daher soll jetzt ein Ausblick auf die Fehlererkennung mit einem Soft-Decision Verfahren gegeben werden.

Die Soft-Decision Decodierungsabbildung δ_{SD} von Definition 3.11 auf Seite 39 soll also zu einer Abbildung

$$\begin{aligned}\hat{\delta}_{\mathrm{SD}}:\ \mathbb{R}^n &\to \{\pm 1\}^k \cup \{¿\},\\ y &\mapsto \hat{u} = \hat{\delta}_{\mathrm{SD}}(y),\end{aligned}$$

erweitert werden, wobei „¿" für die Verwerfung eines Decodierungsergebnisses steht, das heißt, es gibt in diesem Fall kein Decodierungsergebnis.

[10] Abschnitt 5.4.4 auf Seite 101 beschreibt die Behandlung systematischer Codes.

[11] Bei den hier üblicherweise betrachteten Verfahren trifft der Decodierer zu jedem Demodulationsergebnis y eine Decodierungsentscheidung $\hat{u}$. Die Verfahren wurden fehlerminimal konstruiert, das heißt, die Fehlerrate r_{err} wird minimal und die Decodierungsrate $r_{\mathrm{ok}} = 1 - r_{\mathrm{err}}$ maximal. Ein Fehlererkennungsverfahren verwirft einen gewissen Anteil $\hat{r}_{\mathrm{verw}}$, um die Fehlerrate $\hat{r}_{\mathrm{err}} < r_{\mathrm{err}}$ weiter zu senken. Da

$$\hat{r}_{\mathrm{ok}} + \hat{r}_{\mathrm{err}} + \hat{r}_{\mathrm{verw}} = 1$$

sinkt dadurch die Rate $\hat{r}_{\mathrm{ok}} < r_{\mathrm{ok}}$ der korrekt decodierten Codewörter (anderenfalls wäre das Verfahren nicht fehlerminimal gewesen). Im trivialen Extremfall $\hat{r}_{\mathrm{verw}} = 1$ senkt man die Fehlerrate auf 0, indem alle Ergebnisse verworfen werden (was aber natürlich auch die Decodierungsrate zu 0 werden läßt).

Die Branch-and-Bound Decodierung erzeugt ein Decodierungsergebnis $\hat{u}$ mit der Eigenschaft

$$F(\hat{u}) \leq F(u) \quad \text{für alle } u \in \{\pm 1\}^k \setminus \{\hat{u}\}.$$

Betrachte nun ein weiteres $\tilde{u} \in \{\pm 1\}^k$ mit $\tilde{u} \neq \hat{u}$ und der Eigenschaft

$$F(\hat{u}) \leq F(\tilde{u}) \leq F(u) \quad \text{für alle } u \in \{\pm 1\}^k \setminus \{\hat{u}, \tilde{u}\}.$$

Wenn $\hat{u}$ als ein „bestes Decodierungsergebnis“ bezeichnet wird, so kann man $\tilde{u}$ als ein „zweitbestes Decodierungsergebnis“ bezeichnen.

Im Extremfall kann $F(\hat{u}) = F(\tilde{u})$ gelten. In diesem Fall wird man das Decodierergebnis $\hat{u}$ als sehr unsicher ansehen und verwerfen.

Um weitestgehend sicher zu sein, daß $\hat{u}$ dem abgeschickten Codewort entspricht, wird man daher

$$w(\hat{u}, y) \gg w(\tilde{u}, y)$$

fordern[12] und bei Nichterfüllung einer geeigneten Bedingung das Decodierungsergebnis verwerfen.

Für die weitere Betrachtung führen wir die Berechnung der bedingten Wahrscheinlichkeiten auf die Auswertung der Soft-Decision Zielfunktion F zurück, wobei wir stets Voraussetzung 5.1 auf Seite 92 als erfüllt annehmen. Zunächst definieren wir

$$\begin{aligned} g:\ \{\pm 1\}^k &\to \mathbb{R}, \\ u &\mapsto \frac{1}{2\sigma^2} F(u). \end{aligned}$$

als die bezüglich der bitweisen Kanalstörung skalierte Zielfunktion.

Mit Satz 3.23 auf Seite 47 gilt für die bedingten Wahrscheinlichkeiten bei AWGN-Kanälen für alle $\check{u} \in \{\pm 1\}^k$ und alle $y \in \mathbb{R}^n$:

$$\begin{aligned} w(\check{u}, y) &= \frac{\exp\left(-\frac{(y-\varphi(\check{u}))^\top (y-\varphi(\check{u}))}{2\sigma^2}\right)}{\sum\limits_{u \in \{\pm 1\}^k} \exp\left(-\frac{(y-\varphi(u))^\top (y-\varphi(u))}{2\sigma^2}\right)} \\ &= \frac{\exp\left(-g(\check{u})\right)}{\sum\limits_{u \in \{\pm 1\}^k} \exp\left(-g(u)\right)} \\ &= \frac{1}{\sum\limits_{u \in \{\pm 1\}^k} \exp\left(g(\check{u}) - g(u)\right)} \\ &= \frac{1}{1 + \sum\limits_{u \in \{\pm 1\}^k \setminus \{\check{u}\}} \exp\left(g(\check{u}) - g(u)\right)}. \end{aligned}$$

Der nachfolgende Satz trifft eine Aussage über das Verhältnis der bedingten Wahrscheinlichkeiten für das „beste“ und das „zweitbeste“ Decodierungsergebnis.

[12] Da Voraussetzung 5.1 auf Seite 92, die in diesem Kapitel stets als erfüllt angenommen wird, die Voraussetzung 3.12 auf Seite 40 und die sonstigen Voraussetzungen von Definition 3.19 auf Seite 46 umfaßt, können wir mit den bedingten Wahrscheinlichkeiten

$$w(u, y) = P\left(\{\omega \in \Omega;\, U(\omega) = u\} \,|\, \{\omega \in \Omega;\, Y(\omega) = y\}\right), \quad \text{für alle } u \in \{\pm 1\}^k,\ y \in \mathbb{R}^n,$$

arbeiten.

Satz 5.8 (Bewertung des Decodierungsergebnisses)
Voraussetzung 5.1 auf Seite 92 sei erfüllt, $y \in \mathbb{R}^n$ sei eine Realisierung der Kanalausgabe und F sei die zugehörige Soft-Decision Zielfunktion. Es seien $\hat{u},\ \tilde{u} \in \{\pm 1\}^k$ mit $\tilde{u} \neq \hat{u}$ und es gelte

$$F(\hat{u}) \leq F(\tilde{u}) \leq F(u) \quad \text{für alle } u \in \{\pm 1\}^k \setminus \{\hat{u}, \tilde{u}\}.$$

Dann gilt für alle $d \geq 1 + \frac{e}{2^k}$:

$$\frac{w(\hat{u},y)}{w(\tilde{u},y)} \geq d, \quad \text{falls } F(\tilde{u}) - F(\hat{u}) \geq 2\sigma^2 \left(k\ln(2) + \ln(d-1)\right). \tag{5.13}$$

Weiter gilt

$$\frac{w(\hat{u},y)}{w(\tilde{u},y)} \geq 1 + \frac{1}{2^k} \exp\left(\frac{1}{2\sigma^2}\left(F(\tilde{u}) - F(\hat{u})\right)\right), \quad \text{falls } F(\tilde{u}) - F(\hat{u}) \geq 2\sigma^2. \tag{5.14}$$

Beweis. Definiere zur Abkürzung

$$\delta := g(\tilde{u}) - g(\hat{u}) \geq 0.$$

Es gilt dann

$$w(\hat{u},y) = \frac{1}{1 + \sum\limits_{u \in \{\pm 1\}^k \setminus \{\hat{u}\}} \exp\left(g(\hat{u}) - g(u)\right)} \geq \frac{1}{1 + \sum\limits_{u \in \{\pm 1\}^k \setminus \{\hat{u}\}} \exp\left(-\delta\right)} = \frac{1}{1 + (2^k - 1)\exp\left(-\delta\right)}.$$

Weiter gilt

$$\begin{aligned}
\frac{1}{w(\tilde{u},y)} - \frac{1}{w(\hat{u},y)} &= 1 + \sum_{u \in \{\pm 1\}^k \setminus \{\tilde{u}\}} \exp\left(g(\tilde{u}) - g(u)\right) - 1 - \sum_{u \in \{\pm 1\}^k \setminus \{\hat{u}\}} \exp\left(g(\hat{u}) - g(u)\right) \\
&= \exp\left(\delta\right) - \exp\left(-\delta\right) + \sum_{u \in \{\pm 1\}^k \setminus \{\hat{u},\tilde{u}\}} \left(\exp\left(g(\tilde{u}) - g(u)\right) - \exp\left(g(\hat{u}) - g(u)\right)\right) \\
&\geq \exp\left(\delta\right) - \exp\left(-\delta\right)
\end{aligned}$$

und folglich

$$\begin{aligned}
\frac{w(\hat{u},y)}{w(\tilde{u},y)} &\geq 1 + \left(\exp(\delta) - \exp(-\delta)\right) w(\hat{u},y) \\
&\geq 1 + \frac{\exp(\delta) - \exp(-\delta)}{1 + (2^k - 1)\exp(-\delta)}.
\end{aligned}$$

Betrachte nun $\delta \geq 1$, also

$$\begin{aligned}
\exp(\delta) \geq e \geq 1 + 2 \cdot \frac{1}{e} &\geq 1 + \frac{2^k}{2^k - 1} \cdot \frac{1}{\exp(\delta)}, \\
\frac{2^k - 1}{2^k} \exp(\delta) &\geq \frac{2^k - 1}{2^k} + \exp(-\delta), \\
\exp(\delta) - \exp(-\delta) &\geq \frac{1}{2^k}\exp(\delta) + \frac{2^k - 1}{2^k} = \frac{1}{2^k}\exp(\delta)\left(1 + (2^k - 1)\exp(-\delta)\right), \\
\frac{\exp(\delta) - \exp(-\delta)}{1 + (2^k - 1)\exp(-\delta)} &\geq \frac{1}{2^k}\exp(\delta).
\end{aligned}$$

Es gilt also für $\delta \geq 1$, daß

$$\frac{w(\hat{u},y)}{w(\tilde{u},y)} \geq 1 + \frac{1}{2^k}\exp(\delta).$$

Man betrachte nun ein $d \geq 1 + \frac{e}{2^k}$ und

$$F(\tilde{u}) - F(\hat{u}) \geq 2\sigma^2 \left(k\ln(2) + \ln(d-1)\right).$$

Dann gilt

$$\delta = g(\tilde{u}) - g(\hat{u}) \geq k\ln(2) + \ln(d-1) = \ln\left(2^k(d-1)\right) \geq \ln\left(2^k(1+\frac{e}{2^k}-1)\right) = 1$$

und somit

$$\frac{w(\hat{u},y)}{w(\tilde{u},y)} \geq 1 + \frac{1}{2^k}\exp(\delta) \geq 1 + \frac{1}{2^k}\exp\left(\ln\left(2^k(d-1)\right)\right) = d.$$

Für den Nachweis von (5.14) betrachte nun

$$F(\tilde{u}) - F(\hat{u}) \geq 2\sigma^2.$$

Somit gilt $\delta \geq 1$. Definiere weiter

$$d := 1 + \frac{1}{2^k}\exp(\delta) \geq 1 + \frac{e}{2^k}.$$

Es gilt dann

$$\delta = k\ln(2) + \ln(d-1),$$

also

$$F(\tilde{u}) - F(\hat{u}) = 2\sigma^2 \left(k\ln(2) + \ln(d-1)\right).$$

Mit (5.13) folgt

$$\frac{w(\hat{u},y)}{w(\tilde{u},y)} \geq d = 1 + \frac{1}{2^k}\exp(\delta) = 1 + \frac{1}{2^k}\exp\left(\frac{1}{2\sigma^2}\left(F(\tilde{u}) - F(\hat{u})\right)\right).$$

□

Die Aussage (5.13) läßt sich wie folgt interpretieren:

„Es ist d-mal wahrscheinlicher, daß das beste Decodierungsergebnis abgeschickt wurde als das zweitbeste Decodierungsergebnis, wenn y empfangen wurde."

Daraus ergibt sich eine Vorgehensweise zur Fehlererkennung:

- Wähle eine Schranke $d \geq 1 + \frac{e}{2^k}$.
- Bestimme ein bestes Decodierungsergebnis $\hat{u}$ und ein zweitbestes Decodierungsergebnis $\tilde{u}$ zu einem Empfangsvektor y.
- Akzeptiere $\hat{u}$, falls $F(\tilde{u}) - F(\hat{u}) \geq 2\sigma^2 \left(k\ln(2) + \ln(d-1)\right)$, und verwerfe $\hat{u}$ sonst („erkannter Fehler").

Je größer d ist, desto unwahrscheinlicher ist, daß ein Fehler in der Decodierung nicht erkannt wird. Andererseits steigt damit auch die Zahl der richtig decodierten Wörter, die trotzdem als fehlerhaft angesehen werden.

Satz 5.8 trifft zwar keine Aussage über das Fehlerverhältnis, falls $F(\tilde{u}) - F(\hat{u}) < 2\sigma^2$, aber im Grenzfall $F(\tilde{u}) - F(\hat{u}) = 2\sigma^2$ gilt mit (5.14), daß

$$\frac{w(\hat{u},y)}{w(\tilde{u},y)} \geq 1 + \frac{1}{2^k} \exp\left(\frac{1}{2\sigma^2}\left(F(\tilde{u}) - F(\hat{u})\right)\right) = 1 + \frac{e}{2^k}.$$

Für große k ist $1 + \frac{e}{2^k} \approx 1$ und somit würde man das Decodierungsergebnis $\hat{u}$ im Grenzfall $F(\tilde{u}) - F(\hat{u}) = 2\sigma^2$ verwerfen und ebenso im Fall $F(\tilde{u}) - F(\hat{u}) < 2\sigma^2$.

Das folgende Lemma betrachtet die Auswirkungen von extremen Demodulationsergebnissen auf die Werte der Soft-Decision Zielfunktion.

Lemma 5.9 (Abstand zum zweitbesten Decodierungsergebnis)
Voraussetzung 5.1 auf Seite 92 sei erfüllt, $y \in \mathbb{R}^n$ sei eine Realisierung der Kanalausgabe und F sei die zugehörige Soft-Decision Zielfunktion. Es seien $\hat{u}$, $\tilde{u} \in \{\pm 1\}^k$ mit $\tilde{u} \neq \hat{u}$ und es gelte

$$F(\hat{u}) \leq F(\tilde{u}) \leq F(u) \quad \text{für alle } u \in \{\pm 1\}^k \setminus \{\hat{u}, \tilde{u}\}.$$

(i) *Falls $y = 0$, so gilt*

$$F(\hat{u}) = F(\tilde{u}) = F(u) = n \quad \text{für alle } u \in \{\pm 1\}^k \setminus \{\hat{u}, \tilde{u}\}.$$

(ii) *Falls $y = \varphi(\hat{u})$, so gilt*

$$F(\hat{u}) = 0, \quad F(\tilde{u}) = 4d_{\text{ham}}(\varphi(\tilde{u}), \varphi(\hat{u})) \geq 4d_{\text{ham}}(\varphi).$$

Beweis. Ad (i): Für alle $u \in \{\pm 1\}^k$ gilt

$$F(u) = \sum_{j=1}^{n} \left(\bigoplus_{i \in J_j} u_i - y_j \right)^2 = \sum_{j=1}^{n} \left(\bigoplus_{i \in J_j} u_i \right)^2 = \sum_{j=1}^{n} 1 = n$$

Ad (ii): Es gilt trivialerweise

$$F(\hat{u}) = \sum_{j=1}^{n} \left(\bigoplus_{i \in J_j} \hat{u}_i - y_j \right)^2 = \sum_{j=1}^{n} \left(\bigoplus_{i \in J_j} \hat{u}_i - \bigoplus_{i \in J_j} \hat{u}_i \right)^2 = 0$$

und weiter

$$\begin{aligned} F(\tilde{u}) &= \sum_{j=1}^{n} \left(\bigoplus_{i \in J_j} \tilde{u}_i - y_j \right)^2 = \sum_{j=1}^{n} \left(\bigoplus_{i \in J_j} \tilde{u}_i - \bigoplus_{i \in J_j} \hat{u}_i \right)^2 = \sum_{j=1}^{n} 2 \left| \bigoplus_{i \in J_j} \tilde{u}_i - \bigoplus_{i \in J_j} \hat{u}_i \right| \\ &= 4d_{\text{ham}}(\varphi(\tilde{u}), \varphi(\hat{u})) \geq 4d_{\text{ham}}(\varphi). \end{aligned}$$

□

Wenn das Demodulationsergebnis aussagelos ist, also $y = 0$, so läßt sich keine sinnvolle Decodierungsentscheidung treffen. Im besonders günstigen Fall $y = \varphi(\hat{u})$ ist $F(\tilde{u}) - F(\hat{u}) \geq 4d_{\text{ham}}(\varphi)$ und somit kann die Decodierungsentscheidung um so sicherer getroffen werden, je größer die Hamming-Distanz des verwendeten binären linearen (n,k)-Blockcodes ist.

5.8.2 Algorithmus

Nun wird eine Erweiterung von Algorithmus 5.8 auf Seite 112 betrachtet, die neben dem besten Decodierungsergebnis $\hat{u}$ (beziehungsweise $\tilde{u}_{\min}$ vor der Rücktransformation) die Funktionswerte $F_{\min}$ und F_{submin} der Soft-Decision Zielfunktion für das „beste“ und das „zweitbeste“ Decodierungsergebnis liefert. Die transformierte Soft-Decision Zielfunktion $\tilde{F}$ wird daher nun so gewählt, daß $\tilde{F}(\tilde{u}) = F(u)$ mit $u = A^{-\top}\tilde{u}$ gilt, das heißt, im nachfolgenden Algorithmus unterscheiden sich die berechneten Funktionswerte um den Skalierungsfaktor 2 und eine additive Konstante[13] von den Funktionswerten in Algorithmus 5.8.

Analog zur Darstellung in Abbildung 5.5 auf Seite 114 wird der Code anhand des Demodulationsergebnisses $y \in \mathbb{R}^n$ mit Algorithmus 5.2 auf Seite 100 quasi-systematisiert und die Hilfsalgorithmen Algorithmus 5.5 (Indexmengen-Berechnung) und Algorithmus 5.6 (Berechnung von Δ^b) stehen zur Verfügung.

Anstelle des Branch-and-Bound Algorithmus 5.8 berechnet Algorithmus 5.11 auf Seite 123 zunächst zusätzlich einen suboptimalen Funktionswert, indem das letzte[14] Bit des Startpunktes gedreht wird. Dann wird der angepaßte Branch-and-Bound Algorithmus 5.12 auf Seite 124 durchgeführt, wobei das „Bounding“ der zu verwerfenden Unterbäume jetzt davon abhängt, ob die unteren Schranken der Zielfunktion für die Unterbäume größer als F_{submin} sind. Da in der Regel $F_{\text{submin}} > F_{\min}$, sind in Algorithmus 5.12 mehr Knotendurchläufe zu erwarten als in Algorithmus 5.8.

Nach Durchführung von Algorithmus 5.12 ist das decodierte Codewort $\tilde{u}_{\min}$ identisch zum Ergebnis von Algorithmus 5.8. Zusätzlich ist $F_{\min}$ der Zielfunktionswert für dieses „beste“ Codewort und F_{submin} ist der Zielfunktionswert für das „zweitbeste“ Codewort.

Falls für eine gewählte Konstante

$$F_{\text{submin}} - F_{\min} < \text{const},$$

so wird das Decodierungsergebnis $\tilde{u}_{\min}$ beziehungsweise[15] $\hat{u} = A^{-\top}\tilde{u}_{\min}$ verworfen, da das beste und das zweitbeste Decodierungsergebnis in der Bewertung zu dicht zusammenliegen.

[13] Wie in Abschnitt 5.5.1 dargelegt wurde, bleibt die Menge der Minimierer bei positiver Skalierung und konstanter additiver Verschiebung der Zielfunktion unverändert.

[14] Aufgrund der speziellen Quasi-Systematisierung wird $\tilde{u}^0_{\rho(k)}$ als das „unwichtigste“ Bit des Startpunkts angesehen, dessen Änderung den geringsten Einfluß auf den Zielfunktionswert nimmt und folglich eine gute erste Abschätzung des besten suboptimalen Funktionswerts liefert.

[15] Die Matrix A wurde zur Quasi-Systematisierung verwendet, siehe Seite 95.

Vorbereitungen: **Eingang** y, $\tilde{u}^0$, K, ρ; **Ausgang** γ, $F_{\min}$, F_{submin}, $\tilde{u}_{\min}$, M, (b,β,F,S);

$\tilde{u}_{\min} = \tilde{u}^0 \in \{\pm 1\}^k$ sei ein Startpunkt; *siehe Startpunktgenerierung*
$\gamma^0 := 0$;
für $j = 1,\ldots,n$:
    $\gamma^0 := \gamma^0 + (|y_j| - 1)^2$;
für $b = 1,\ldots,k$:
    $h := 0$;
    für $j \in K_b$:
        $h := h + |y_j|$;
    $\gamma^b := \gamma^{b-1} + 2h$;
$M := \emptyset$;
$(b,\beta,F,S) := (0,\beta,\gamma^0,0)$; β *beliebig*
für $b = 1,\ldots,k-1$:
    $\tilde{u}_{\rho(b)} := \tilde{u}_{\min,\rho(b)}$; *$\tilde{u}$ wird erweitert*
    $\delta_S := \Delta^b(\tilde{u})$;
    $S^{alt} := S - \delta_S$; $S := S + \delta_S$;
    $F^{alt} := \gamma^b - 2S^{alt}$; *Gewicht eines Nachfolgeknotens*
    push$(M,(b,\tilde{u}_{\min,\rho(b)} \oplus -1, F^{alt}, S^{alt}))$; *Alternative in den Keller*
$\tilde{u}_{\rho(k)} := \tilde{u}_{\min,\rho(k)}$; *$\tilde{u}$ wird erweitert*
$\delta_S := \Delta^k(\tilde{u})$;
$S^{alt} := S - \delta_S$; $S := S + \delta_S$;
$F^{alt} := \gamma^k - 2S^{alt}$; $F := \gamma^k - 2S$; *Gewichte der Nachfolgeknoten*
falls $F \le F^{alt}$: *Startpunkt ist besser*
    $F_{\min} := F$; $F_{\text{submin}} := F^{alt}$;
sonst: *Alternative ist besser*
    $\tilde{u}_{\rho(k)} := \tilde{u}_{\min,\rho(b)} \oplus -1$;
    $\tilde{u}_{\min,\rho(b)} := \tilde{u}_{\rho(k)}$;
    $F_{\min} := F^{alt}$; $F_{\text{submin}} := F$;
$(b,\beta,F,S) :=$**pop**(M); *Nächste Alternative aus dem Keller*
$\tilde{u}_{\rho(b)} := \beta$;
$b_{\text{last}} := b$;

Algorithmus 5.11: *Fehlererkennung: Vorbereitungen für die beste und zweitbeste Lösung*

BB-Fehlererkennung: **Eingang** $\gamma, F_{\min}, F_{\text{submin}}, \tilde{u}_{\min}, M, (b,\beta,F,S), K, \rho$; **Ausgang** $\tilde{u}_{\min}$, $F_{\min}$, F_{submin};

solange $(b < k)$: *Hauptiteration: Branch and Bound*
  falls $F < F_{\text{submin}}$: *Das Tupel $(b,\tilde{u},F,S)$ wird betrachtet*
    $b := b+1$;
    $\tilde{u}_{\rho(b)} := -1$; *$\tilde{u}$ wird zu $\tilde{u}^-$ erweitert*
    $\delta_S := \Delta^b(\tilde{u})$;
    $S^- := S + \delta_S$; $S^+ := S - \delta_S$;
    $F^- := \gamma^b - 2S^-$; $F^+ := \gamma^b - 2S^+$; *Gewichte der Nachfolgeknoten*
    falls $\min(F^-,F^+) < F_{\text{submin}}$:
      falls $F^- < F^+$: *also falls $\delta_S > 0$*
        falls $b < k$: *Zwischenknoten*
          falls $F^+ < F_{\text{submin}}$:
            push$(M,(b,+1,F^+,S^+))$; *Alternative in den Keller*
          $F := F^-$; $S := S^-$;
        sonst: *Blatt*
          falls $F^- < F_{\min}$:
            $F_{\text{submin}} := F_{\min}$;
            $F_{\min} := F^-$; *Bessere Decodierung gefunden*
            für $i = b_{\text{last}},\ldots,k$:
              $\tilde{u}_{\min,\rho(i)} := \tilde{u}_{\rho(i)}$;
            $b_{\text{last}} := k$;
          sonst:
            $F_{\text{submin}} := F^-$;

Fortsetzung folgt...

Algorithmus 5.12: *Branch-and-Bound Verfahren mit Fehlererkennung*

```
...                                                              Fortsetzung
        sonst:
            falls b < k:                                         Zwischenknoten
                falls F⁻ < F_submin:
                    push(M, (b, −1, F⁻, S⁻));                    Alternative in den Keller
                ũ_ρ(b) := +1;   F := F⁺;   S := S⁺;
            sonst:                                               Blatt
                falls F⁺ < F_min:
                    ũ_ρ(b) := +1; F_submin := F_min;
                    F_min := F⁺;                                 Bessere Decodierung gefunden
                    für i = b_last, ..., k:
                        ũ_min,ρ(i) := ũ_ρ(i);
                    b_last := k;
                sonst:
                    F_submin := F⁺;
        sonst:
            b := k;                                              Beende die Suche in diesem Zweig
    sonst:
        b := k;                                                  Beende die Suche in diesem Zweig
    falls (b = k) und M ≠ ∅:
        (b, β, F, S) := pop(M);                                  Alternative aus dem Keller
        ũ_ρ(b) := β;
        falls b < b_last:
            b_last := b;
```

Algorithmus 5.12: *Branch-and-Bound Verfahren mit Fehlererkennung — Fortsetzung*

Kapitel 6

Beispiele und numerische Ergebnisse

I am fully functional.

(Data)

6.1 Numerischer Vergleich von Codierungen und Decodierungsverfahren

Beim numerischen Vergleich von Codes und Decodierungsverfahren genügt es nicht, für einen vorgegebenen n-Kanal $\mathcal{K}$ die empirischen Fehlerwahrscheinlichkeiten für verschiedene Codierungen und Decodierungsverfahren zu erheben. Eine Codierung A, die $k = n$ Infobits auf n Codebits abbildet, wird eine erheblich höhere Fehlerrate bei einem Decodierungsverfahren aufweisen als eine Codierung B, die $k = 1$ Infobits auf n Codebits abbildet. Dafür wurden mit Codierung A insgesamt n Infobits pro Codewort gesendet, aber bei Codierung B lediglich 1 Infobit. Um einen sinnvollen Vergleich zu erstellen, muß die Coderate $R = \frac{k}{n}$ also in der Kanalstörung berücksichtigt werden.

Dazu führen wir zunächst die gebräuchlichen nachrichtentechnischen Begriffe zur Beschreibung der Kanalstörung ein, vergleiche [Fri95, Pro01]:

- $N_0 > 0$ heißt die einseitige Rauschleistungsdichte und bezieht sich auf die tatsächlichen physikalischen Eigenschaften des physikalischen Kanals.
- $E_c > 0$ bezeichnet die (Sende-)Energie pro Codebit. Je höher die Energie ist, desto geringer wird der Einfluß der Rauschleistungsdichte auf die tatsächliche Störung der Übertragung. Wenn $\mathcal{K}$ ein AWGN-Kanal ist, so ist die bitweise Varianz σ^2 der Kanalstörung definiert über
$$\sigma^2 := \frac{N_0}{2E_c}.$$
Der Faktor 2 ergibt sich aus der Einseitigkeit der Rauschleistungsdichte.
- Für einen gegebenen (n,k)-Blockcode bezeichnet $E_b > 0$ die (mittlere) Energie pro Infobit und ist definiert als
$$E_b := \frac{n}{k} E_c.$$
Vergleicht man etwa diesen Code mit einem (n,k')-Blockcode, so führt man diesen Vergleich bei identischer Energie pro Infobit durch. Somit folgt aus
$$\frac{n}{k} E_c = E_b = E_b' = \frac{n}{k'} E_c',$$

daß $\frac{E_c'}{E_c} = \frac{k'}{k}$. Je mehr Infobits pro fixer Codewortlänge n „transportiert“ werden, desto höher ist die betrachtete Energie, um einen gerechten Codevergleich durchführen zu können.

- Das Verhältnis E_b/N_0 heißt Signal-to-Noise Ratio (SNR).
- E_b/N_0[dB] bezeichnet die Signal-to-Noise Ratio (SNR[dB]) in Dezibel und ist definiert als

$$E_b/N_0[\text{dB}] = 10 \cdot \log_{10}(E_b/N_0).$$

  Bei nachrichtentechnischen Angaben wird oft E_b/N_0[dB] als Größe beim Vergleich von Codierungen und Decodierungsverfahren verwendet.

- Zusammenfassend berechnet sich die bitweise Varianz σ^2 der Kanalstörung aus E_b/N_0[dB] wie folgt

$$\sigma^2 = \frac{n}{2 \cdot k \cdot 10^{(E_b/N_0[\text{dB}])/10}}.$$

Zum numerischen Vergleich betrachten wir die empirische Wortfehlerwahrscheinlichkeit

$$P_w := \frac{\text{Anzahl fehlerhaft decodierter Codewörter}}{\text{Anzahl decodierter Codewörter}}$$

und die empirische Bitfehlerwahrscheinlichkeit

$$P_b := \frac{\text{Anzahl fehlerhaft decodierter Infobits}}{\text{Anzahl decodierter Infobits}}.$$

6.2 Soft-Decision Decodierung

Die zum numerischen Vergleich betrachteten BCH[1]-Codes [Hoc59, BRC60b, BRC60a] sind spezielle polynomerzeugte systematische binäre lineare (n,k)-Blockcodes (n,k,φ), die wir hier als (n,k)-BCH-Codes bezeichnen.

Für diese Codierungen ist die sogenannte BM-Methode (Begrenzte Minimaldistanz) anwendbar. Jedes Codewort $c \in \varphi\left(\{\pm 1\}^k\right)$ wird dabei als Mittelpunkt einer Kugel im $\mathbb{R}^n$ bezüglich der Summennorm mit Radius $t \leq d_{\text{ham}}(\varphi) - 1$ angesehen[2], das heißt, diese Kugeln sind damit untereinander disjunkt. Befindet sich das Demodulationsergebnis in einer solchen Kugel, so wird es zum Kugelmittelpunkt decodiert, anderenfalls findet keine Decodierung statt.

Die nachfolgenden Fehlerkurvendarstellungen für den Branch-and-Bound Algorithmus und seinen Startpunkt aus Kapitel 5, siehe dazu die schematische Darstellung in Abbildung 5.5 auf Seite 114, sind zum Vergleich durch die Fehlerkurven der BM-Methode[3] ergänzt.

[1] BCH steht für Bose, Chaudhuri und Hocqzenghem. Eine ausführliche Darstellung dieser Codes findet sich z.B, in [Fri95, Jun95, Bos98, Bos99].

[2] Die Kugel $K_t(c)$ mit Mittelpunkt c und Radius t bezüglich der Summennorm ist also

$$K_t(c) = \left\{ x \in \mathbb{R}^n;\ \sum_{j=1}^{n} |x_j - c_j| \leq t \right\}.$$

[3] Wie in [Fri95] dargestellt, lassen sich die Fehlerkurven für die verwendeten (n,k)-BCH-Codes bei Verwendung der BM-Methode leicht approximativ berechnen. Für jeden dieser BCH-Codes ist die (Entwurfs-)Anzahl t korrigierbarer Fehler bekannt. Zunächst berechnet sich bei einem AWGN-Kanal die Wahrscheinlichkeit p_e, daß ein Bit bei der

Zur Berechnung der empirischen Fehlerwahrscheinlichkeiten wurden für jeden verwendeten Code je für eine festgewählte Signal-to-Noise Ratio E_b/N_0 solange Codewörter decodiert, bis mindestens 100 Wörter falsch decodiert wurden. Die uncodierten Wörter $u \in \{\pm 1\}^k$ wurden dabei durch gleichverteilte Zufallszahlen und die additive Kanalstörung durch normalverteilte Zufallszahlen simuliert.

Das Branch-and-Bound Verfahren wurde jeweils bei 10 000 000 Knotendurchläufen gestoppt, falls diese Anzahl erreicht wurde.

Die Branch-and-Bound Methode wurde als wortfehlerminimales Verfahren entworfen, aber ergänzend werden neben den Wortfehlerkurven auch die jeweiligen Bitfehlerkurven dargestellt.

Übertragung das Vorzeichen wechselt, zu

$$p_e = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{0} \exp\left(-\frac{(x-1)^2}{2\sigma^2}\right)\, dx = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-\frac{1}{\sigma}} \exp\left(-\frac{x^2}{2}\right)\, dx = \frac{1}{2}\operatorname{erfc}\left(\frac{1}{\sqrt{2\sigma^2}}\right) = \frac{1}{2}\operatorname{erfc}\left(\sqrt{\frac{E_c}{N_0}}\right),$$

wobei erfc die komplementäre Gaußsche Fehlerfunktion ist. Dann gilt für die Fehlerwahrscheinlichkeiten

$$P_{w,\mathrm{BM}} \approx \sum_{r=t+1}^{n} \binom{n}{k} p_e^r (1-p_e)^{n-r}, \quad P_{b,\mathrm{BM}} \approx \sum_{r=t+1}^{n} \min\left\{1, \frac{r+t}{k}\right\} \binom{n}{k} p_e^r (1-p_e)^{n-r}.$$

(7,4)-BCH-Code	
Codelänge $n =$	7
Codedimension $k =$	4
Generatorpolynom $g(x) =$	$x^3 + x^1 + 1$

Quellenangabe: [Fri95, Pro01]

Dieser triviale Code wurde als einführendes Beispiel unter der Bezeichnung $(7,4,\varphi_{\text{bsp}})$ in Kapitel 2 ab Seite 23 verwendet. Die numerische Untersuchung dieses aus lediglich 16 Elementen bestehenden Codes schließt das Beispiel ab.

(7,4)-BCH-Code
Decodierung mit dem Branch-and-Bound Verfahren

E_b/N_0[dB]	P_w	$\log_{10}(P_w)$	P_b	$\log_{10}(P_b)$
2.5	0.0459800000	**-1.3374**	0.0207400000	**-1.6832**
3.0	0.0301100000	**-1.5213**	0.0134775000	**-1.8704**
3.5	0.0190300000	**-1.7206**	0.0084750000	**-2.0719**
4.0	0.0120000000	**-1.9208**	0.0053275000	**-2.2735**
4.5	0.0067500000	**-2.1707**	0.0029250000	**-2.5339**
5.0	0.0037300000	**-2.4283**	0.0016950000	**-2.7708**
5.5	0.0018700000	**-2.7282**	0.0008325000	**-3.0796**
6.0	0.0007868687	**-3.1041**	0.0003658940	**-3.4366**
6.5	0.0003847056	**-3.4149**	0.0001836969	**-3.7359**
7.0	0.0001010983	**-3.9953**	0.0000462525	**-4.3349**
7.5	0.0000401533	**-4.3963**	0.0000180690	**-4.7431**
8.0	0.0000132863	**-4.8766**	0.0000063110	**-5.1999**

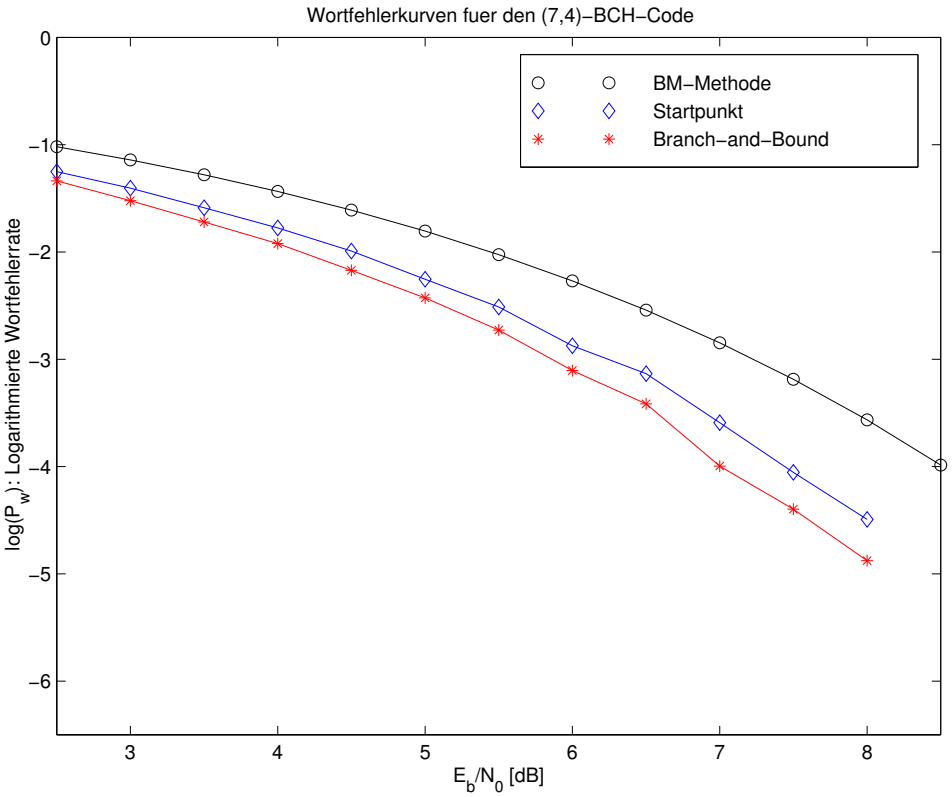

Abbildung 6.1: *Wortfehlerkurven für den (7,4)-BCH-Code*

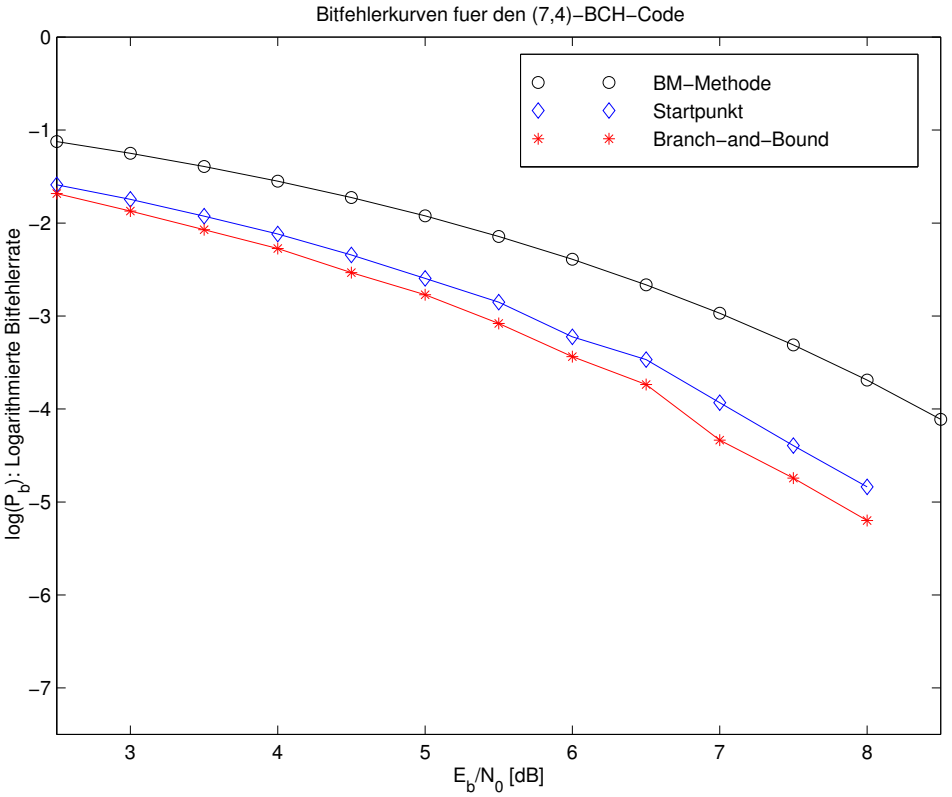

Abbildung 6.2: *Bitfehlerkurven für den (7,4)-BCH-Code*

(31,16)-BCH-Code	
Codelänge $n =$	31
Codedimension $k =$	16
Generatorpolynom $g(x) =$	$x^{15} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^5 + x^3 + x^2 + x^1 + 1$

Quellenangabe: [Fri95, Pro01]

(31,16)-BCH-Code
Decodierung mit dem Branch-and-Bound Verfahren

E_b/N_0[dB]	P_w	$\log_{10}(P_w)$	P_b	$\log_{10}(P_b)$
2.5	0.0312000000	**-1.5058**	0.0078937500	**-2.1027**
3.0	0.0150000000	**-1.8239**	0.0038000000	**-2.4202**
3.5	0.0062912866	**-2.2013**	0.0016750550	**-2.7760**
4.0	0.0022422530	**-2.6493**	0.0005185210	**-3.2852**
4.5	0.0007456566	**-3.1275**	0.0001840840	**-3.7350**
5.0	0.0002097623	**-3.6783**	0.0000474587	**-4.3237**
5.5	0.0000397038	**-4.4012**	0.0000092063	**-5.0359**
6.0	0.0000088294	**-5.0541**	0.0000020308	**-5.6923**
6.5	0.0000011889	**-5.9248**	0.0000002705	**-6.5679**

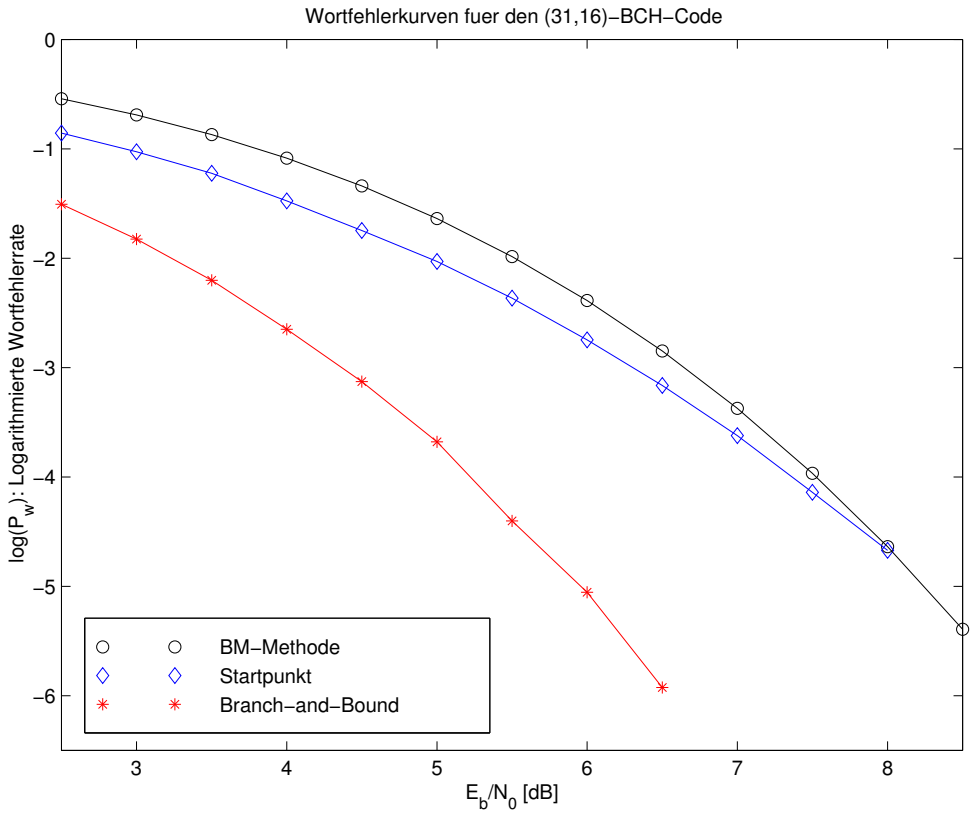

Abbildung 6.3: *Wortfehlerkurven für den (31,16)-BCH-Code*

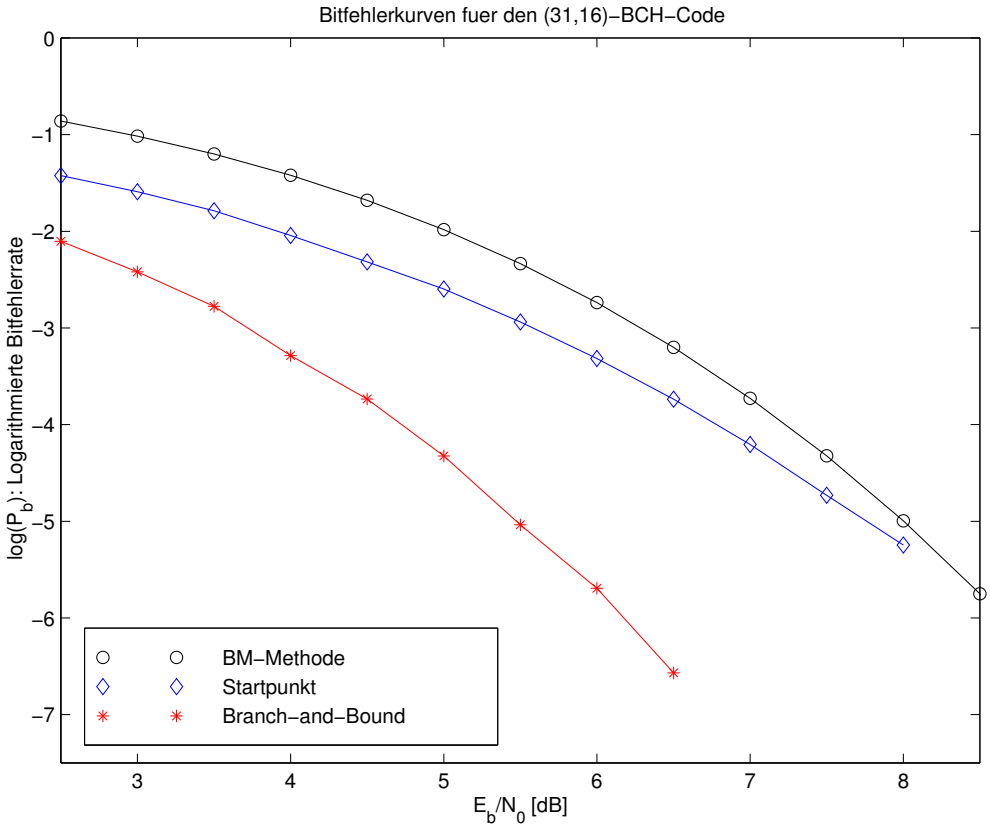

Abbildung 6.4: *Bitfehlerkurven für den (31,16)-BCH-Code*

(31,21)-BCH-Code	
Codelänge $n =$	31
Codedimension $k =$	21
Generatorpolynom $g(x) =$	$x^{10} + x^9 + x^8 + x^6 + x^5 + x^3 + 1$

Quellenangabe: [Fri95, Pro01]

(31,21)-BCH-Code
Decodierung mit dem Branch-and-Bound Verfahren

E_b/N_0[dB]	P_w	$\log_{10}(P_w)$	P_b	$\log_{10}(P_b)$
2.5	0.0620000000	**-1.2076**	0.0120047619	**-1.9206**
3.0	0.0306000000	**-1.5143**	0.0058047619	**-2.2362**
3.5	0.0145000000	**-1.8386**	0.0027238095	**-2.5648**
4.0	0.0047463097	**-2.3236**	0.0008678966	**-3.0615**
4.5	0.0016358312	**-2.7863**	0.0002960076	**-3.5287**
5.0	0.0005421729	**-3.2659**	0.0000913949	**-4.0391**
5.5	0.0001490880	**-3.8266**	0.0000268358	**-4.5713**
6.0	0.0000245629	**-4.6097**	0.0000043160	**-5.3649**
6.5	0.0000047302	**-5.3251**	0.0000007929	**-6.1008**
7.0	0.0000005799	**-6.2367**	0.0000001016	**-6.9931**

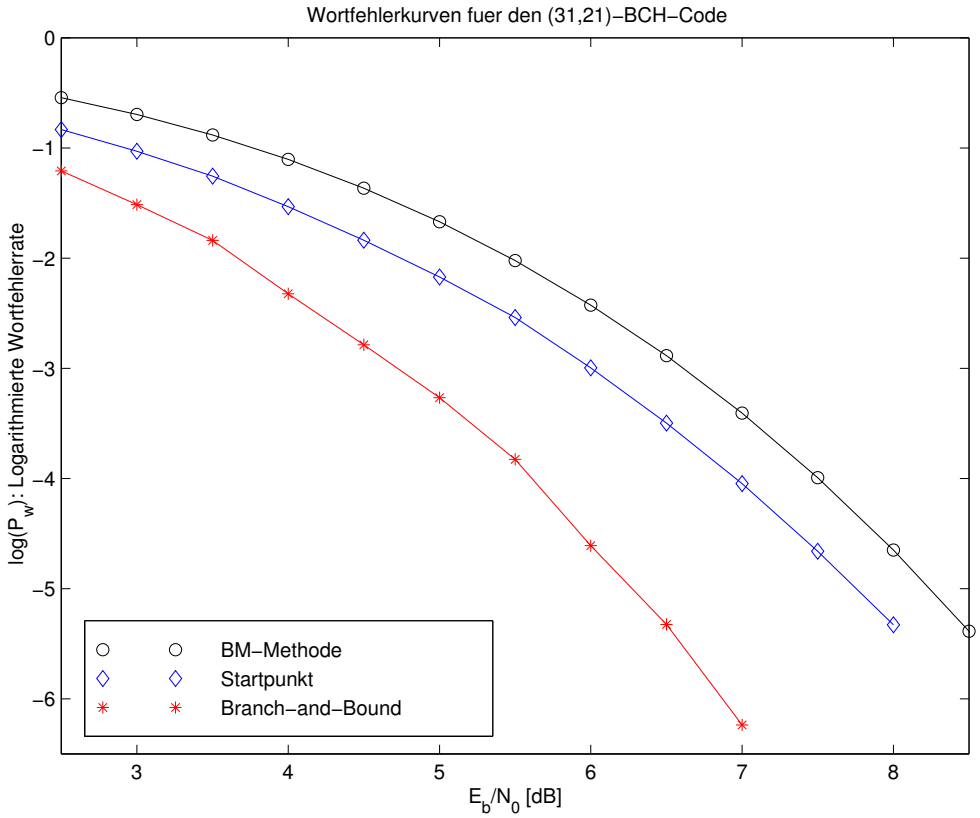

Abbildung 6.5: *Wortfehlerkurven für den (31,21)-BCH-Code*

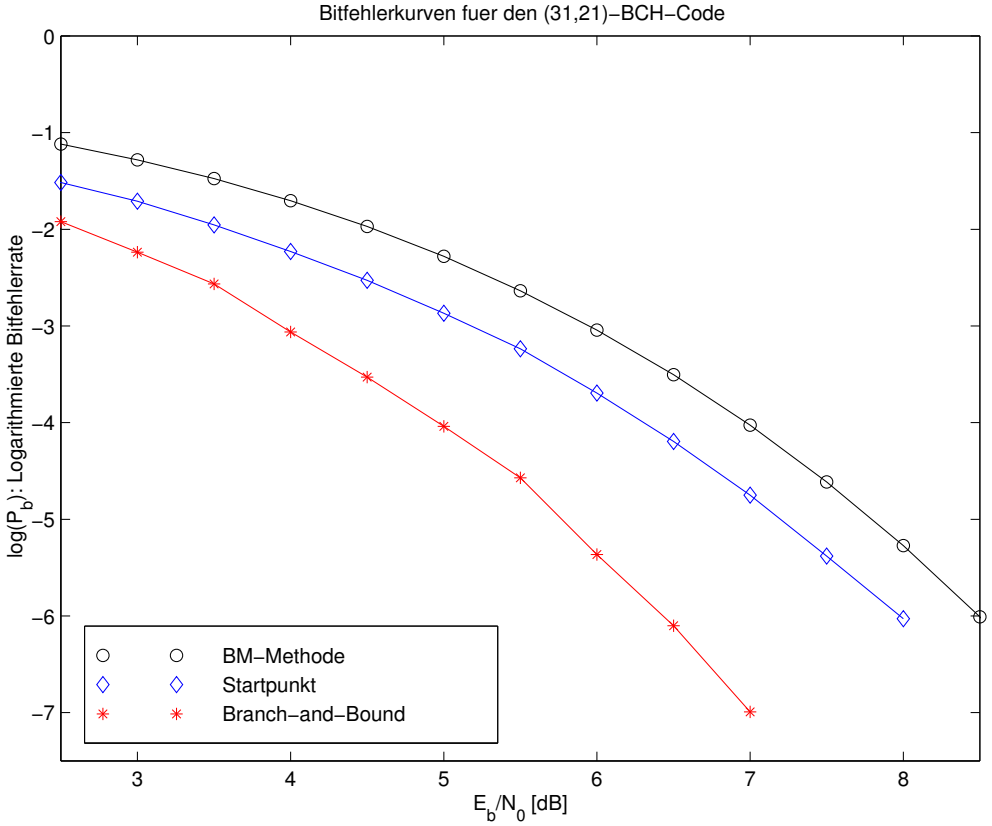

Abbildung 6.6: *Bitfehlerkurven für den (31,21)-BCH-Code*

(63,30)-BCH-Code	
Codelänge $n =$	63
Codedimension $k =$	30
Generatorpolynom $g(x) =$	$x^{33} + x^{32} + x^{30} + x^{29} + x^{28} + x^{27} + x^{26} + x^{23} + x^{22} + x^{20} + x^{15} + x^{14} + x^{13} + x^{11} + x^{9} + x^{8} + x^{6} + x^{5} + x^{2} + x^{1} + 1$

Quellenangabe: [Fri95, Pro01]

(63,30)-BCH-Code
Decodierung mit dem Branch-and-Bound Verfahren

E_b/N_0[dB]	P_w	$\log_{10}(P_w)$	P_b	$\log_{10}(P_b)$
2.5	0.0080224629	**-2.0957**	0.0020243348	**-2.6937**
3.0	0.0020130851	**-2.6961**	0.0004972320	**-3.3034**
3.5	0.0003501744	**-3.4557**	0.0000849757	**-4.0707**
4.0	0.0000829445	**-4.0812**	0.0000198790	**-4.7016**
4.5	0.0000096389	**-5.0160**	0.0000023005	**-5.6382**
5.0	0.0000010363	**-5.9845**	0.0000002332	**-6.6323**

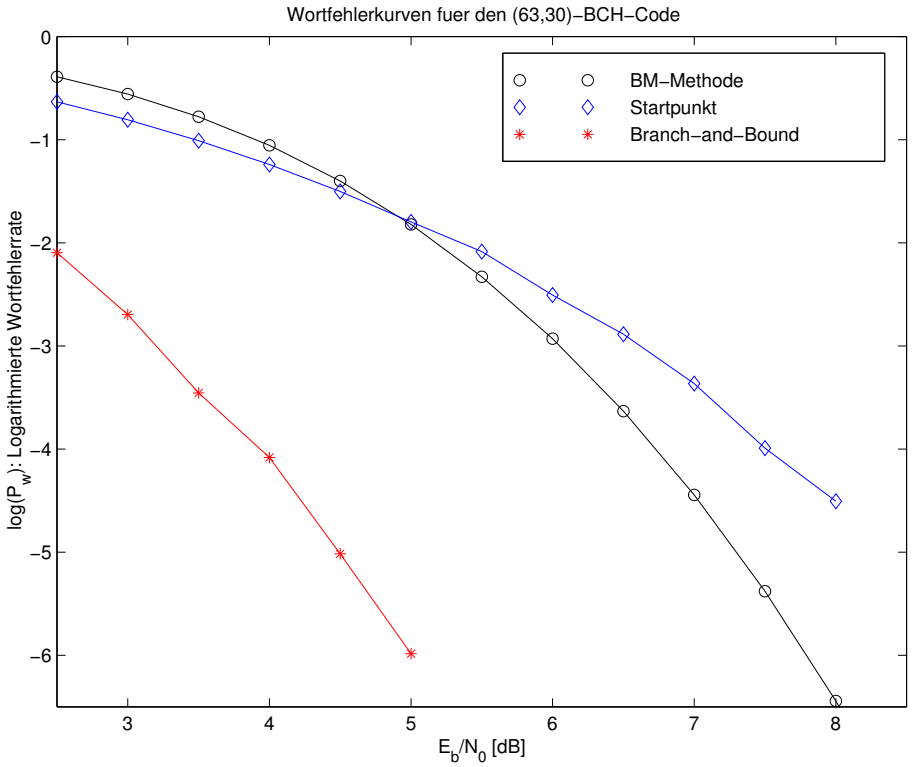

Abbildung 6.7: *Wortfehlerkurven für den (63,30)-BCH-Code*

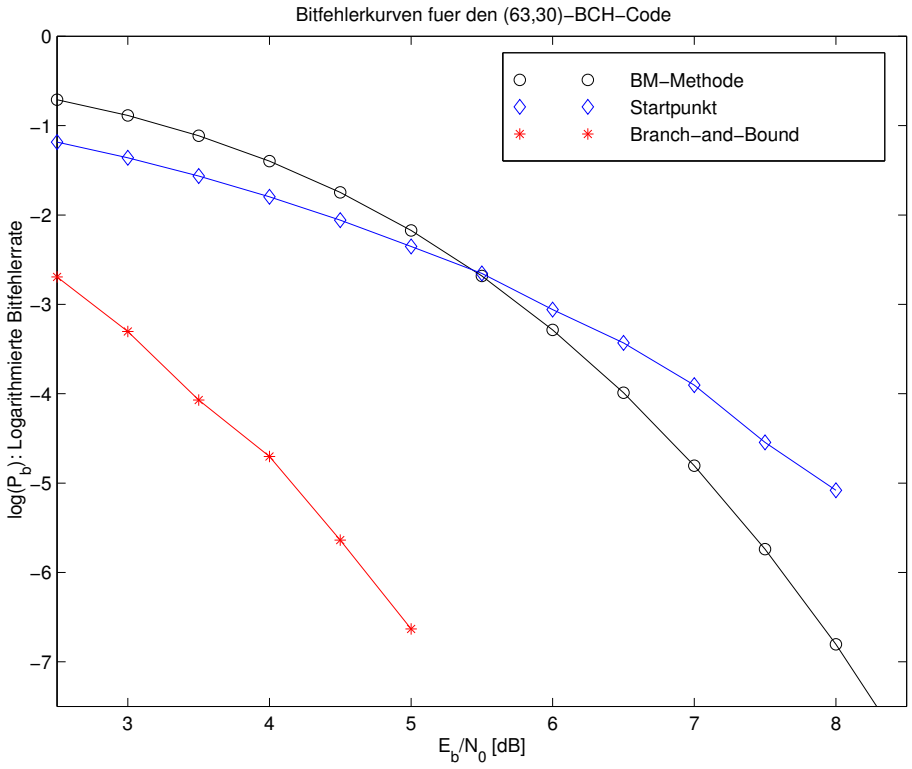

Abbildung 6.8: *Bitfehlerkurven für den (63,30)-BCH-Code*

(63,45)-BCH-Code	
Codelänge $n =$	63
Codedimension $k =$	45
Generatorpolynom $g(x) =$	$x^{18} + x^{17} + x^{16} + x^{15} + x^9 + x^7 + x^6 + x^3 + x^2 + x^1 + 1$

Quellenangabe: [Fri95, Pro01]

(63,45)-BCH-Code *Decodierung mit dem Branch-and-Bound Verfahren*				
E_b/N_0[dB]	P_w	$\log_{10}(P_w)$	P_b	$\log_{10}(P_b)$
2.5	0.0681000000	**-1.1669**	0.0093733333	**-2.0281**
3.0	0.0282000000	**-1.5498**	0.0037822222	**-2.4223**
3.5	0.0079516539	**-2.0995**	0.0011397371	**-2.9432**
4.0	0.0018590471	**-2.7307**	0.0002292825	**-3.6396**
4.5	0.0003987543	**-3.3993**	0.0000521925	**-4.2824**
5.0	0.0000564100	**-4.2486**	0.0000072832	**-5.1377**
5.5	0.0000067518	**-5.1706**	0.0000008237	**-6.0842**
6.0	0.0000005352	**-6.2715**	0.0000000676	**-7.1700**

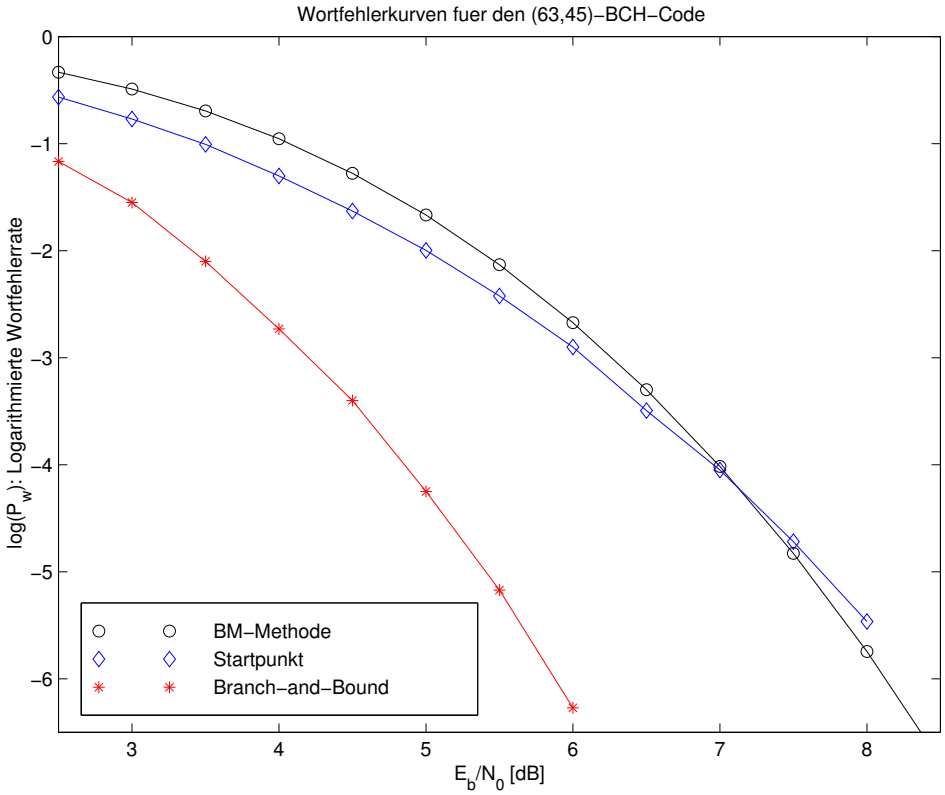

Abbildung 6.9: *Wortfehlerkurven für den (63,45)-BCH-Code*

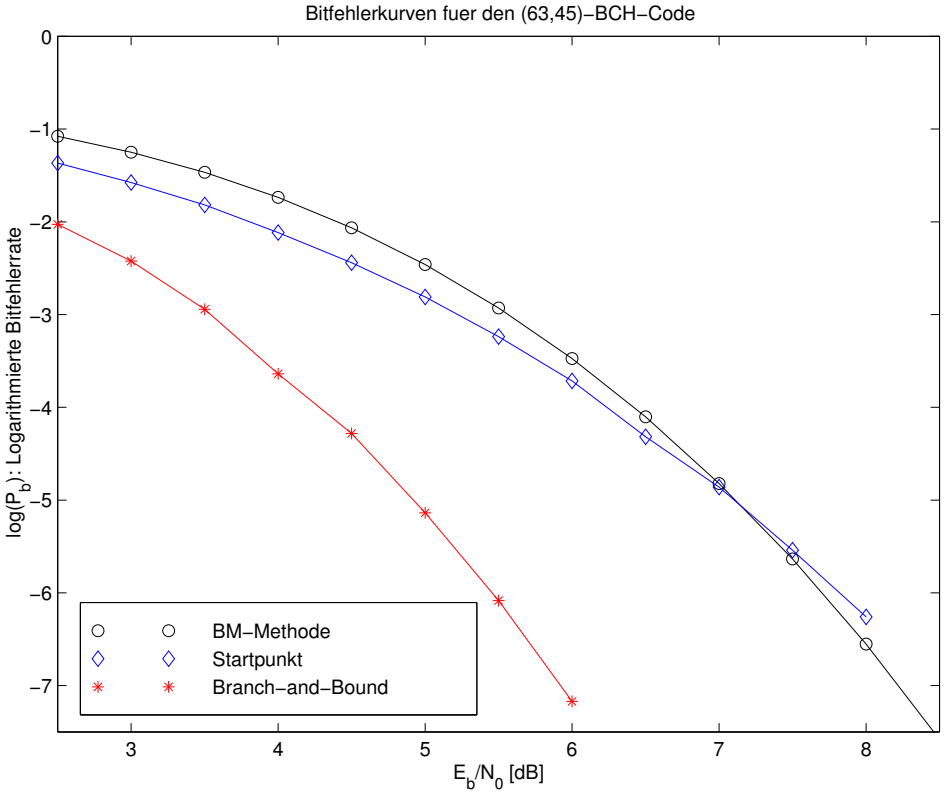

Abbildung 6.10: *Bitfehlerkurven für den (63,45)-BCH-Code*

(127,99)-BCH-Code	
Codelänge $n =$	127
Codedimension $k =$	99
Generatorpolynom $g(x) =$	$x^{28} + x^{27} + x^{26} + x^{23} + x^{20} + x^{19} + x^{18} + x^{13} + x^{10} + x^9 + x^7 + x^5 + x^4 + x^3 + 1$

Quellenangabe: [Fri95, Pro01]

(127,99)-BCH-Code
Decodierung mit dem Branch-and-Bound Verfahren

E_b/N_0[dB]	P_w	$\log_{10}(P_w)$	P_b	$\log_{10}(P_b)$
2.5	0.1088000000	**-0.9634**	0.0114244444	**-1.9422**
3.0	0.0317300000	**-1.4985**	0.0032303030	**-2.4908**
3.5	0.0058500000	**-2.2328**	0.0005675758	**-3.2460**
4.0	0.0008158870	**-3.0884**	0.0000759846	**-4.1193**
4.5	0.0000608402	**-4.2158**	0.0000055494	**-5.2558**
5.0	0.0000035305	**-5.4522**	0.0000003006	**-6.5220**

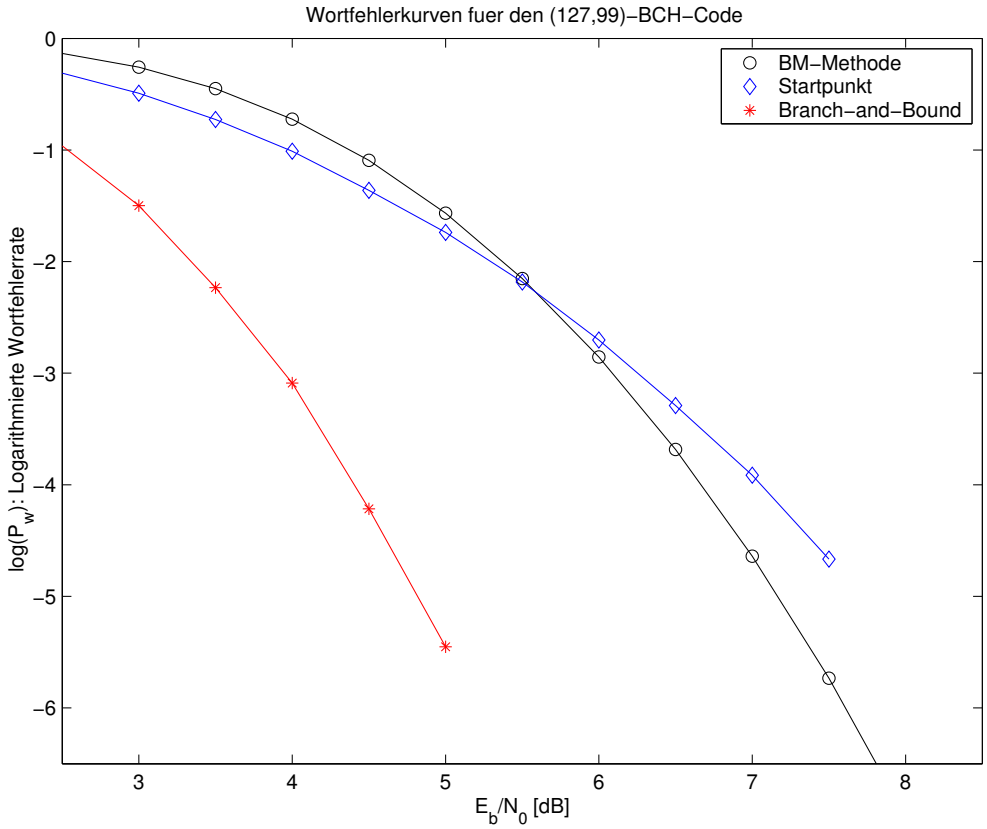

Abbildung 6.11: *Wortfehlerkurven für den (127,99)-BCH-Code*

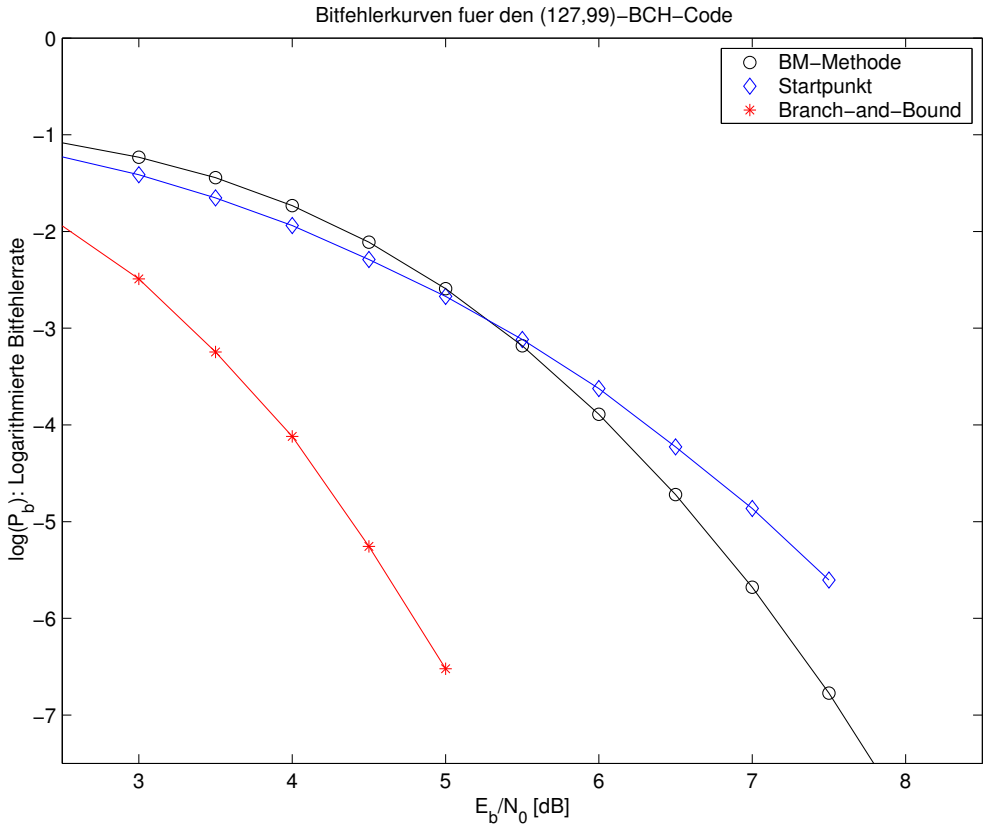

Abbildung 6.12: *Bitfehlerkurven für den (127,99)-BCH-Code*

(255,191)-BCH-Code	
Codelänge $n =$	255
Codedimension $k =$	191
Generatorpolynom $g(x) =$	$x^{64} + x^{62} + x^{61} + x^{59} + x^{58} + x^{55} + x^{54} + x^{53} + x^{50} + x^{49} + x^{48} + x^{42} + x^{41} + x^{40} + x^{39} + x^{38} + x^{37} + x^{33} + x^{30} + x^{29} + x^{27} + x^{25} + x^{24} + x^{22} + x^{21} + x^{19} + x^{18} + x^{17} + x^{16} + x^{15} + x^{12} + x^{11} + x^{8} + x^{6} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

Quellenangabe: [Fri95, Pro01]

(255,191)-BCH-Code *Decodierung mit dem Branch-and-Bound Verfahren*				
E_b/N_0[dB]	P_w	$\log_{10}(P_w)$	P_b	$\log_{10}(P_b)$
2.5	0.0456000000	**-1.3410**	0.0049293194	**-2.3072**
3.0	0.0070841598	**-2.1497**	0.0007325244	**-3.1352**
3.5	0.0006895981	**-3.1614**	0.0000682016	**-4.1662**
4.0	0.0000132979	**-4.8762**	0.0000014760	**-5.8309**

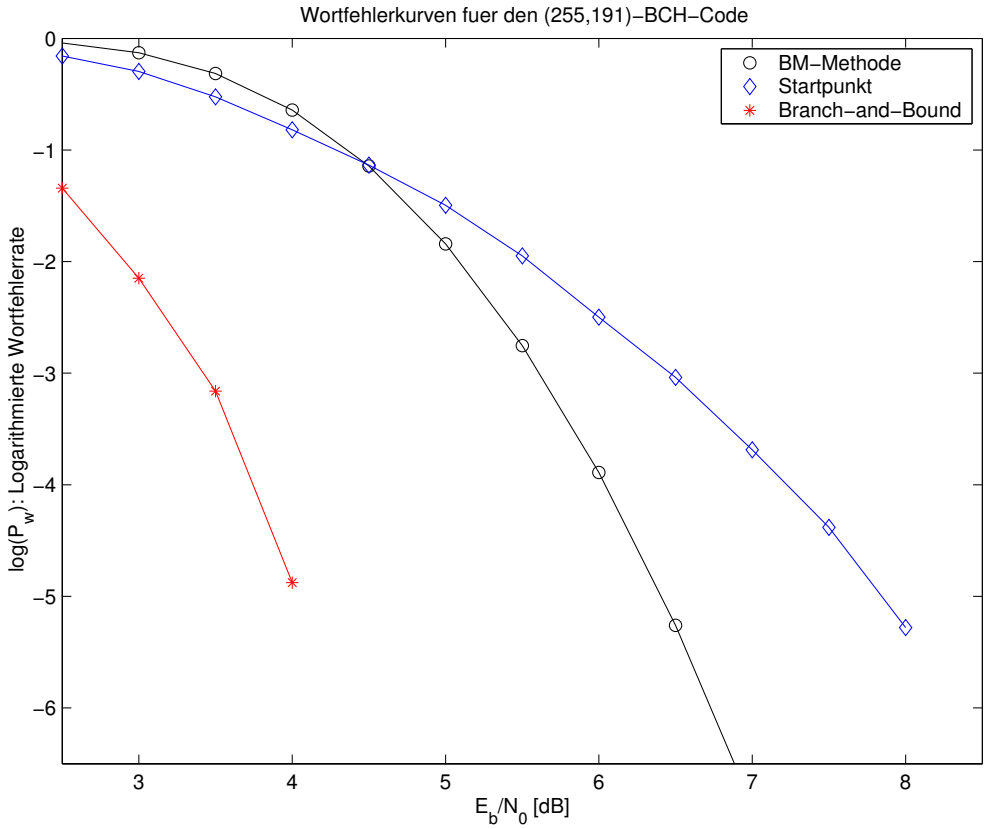

Abbildung 6.13: *Wortfehlerkurven für den (255,191)-BCH-Code*

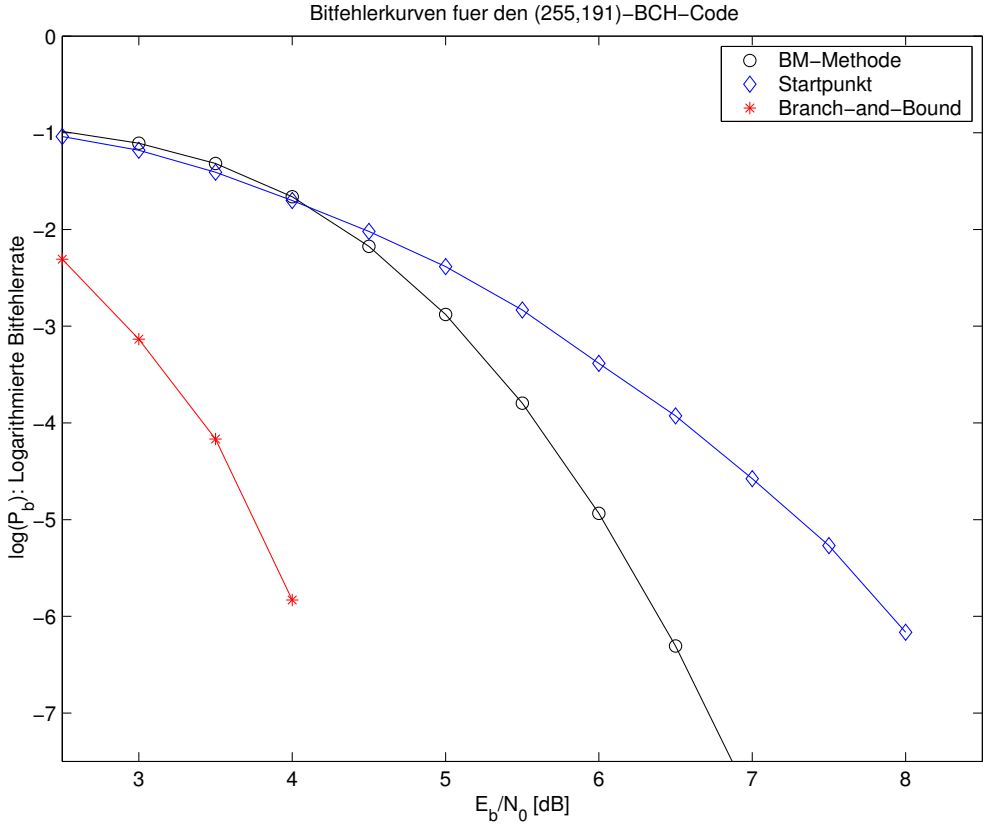

Abbildung 6.14: *Bitfehlerkurven für den (255,191)-BCH-Code*

(255,223)-BCH-Code	
Codelänge $n =$	255
Codedimension $k =$	223
Generatorpolynom $g(x) =$	$x^{32} + x^{31} + x^{30} + x^{29} + x^{27} + x^{26} + x^{25} + x^{22} + x^{20} + x^{19} + x^{17} + x^{16} + x^{14} + x^{9} + x^{7} + x^{6} + x^{5} + x^{4} + x^{3} + x^{2} + 1$

Quellenangabe: [Fri95, Pro01]

In diesem Beispiel ist ab 4.5 Dezibel für E_b/N_0 eine Verschiebung der Fehlerkurven für das Branch-and-Bound Verfahren zu erkennen, die auf Erreichen der Iterationsschranken zurückzuführen ist.

(255,223)-BCH-Code
Decodierung mit dem Branch-and-Bound Verfahren

E_b/N_0[dB]	P_w	$\log_{10}(P_w)$	P_b	$\log_{10}(P_b)$
2.5	0.5431000000	**-0.2651**	0.0337878924	**-1.4712**
3.0	0.2448000000	**-0.6112**	0.0144421525	**-1.8404**
3.5	0.0671000000	**-1.1733**	0.0038717489	**-2.4121**
4.0	0.0093624192	**-2.0286**	0.0004840749	**-3.3151**
4.5	0.0027590012	**-2.5592**	0.0001281760	**-3.8922**
5.0	0.0007449511	**-3.1279**	0.0000314349	**-4.5026**
5.5	0.0003784138	**-3.4220**	0.0000151366	**-4.8200**
6.0	0.0001578841	**-3.8017**	0.0000064286	**-5.1919**
6.5	0.0000338681	**-4.4702**	0.0000013107	**-5.8825**
7.0	0.0000077772	**-5.1092**	0.0000002999	**-6.5230**
7.5	0.0000009967	**-6.0014**	0.0000000477	**-7.3217**

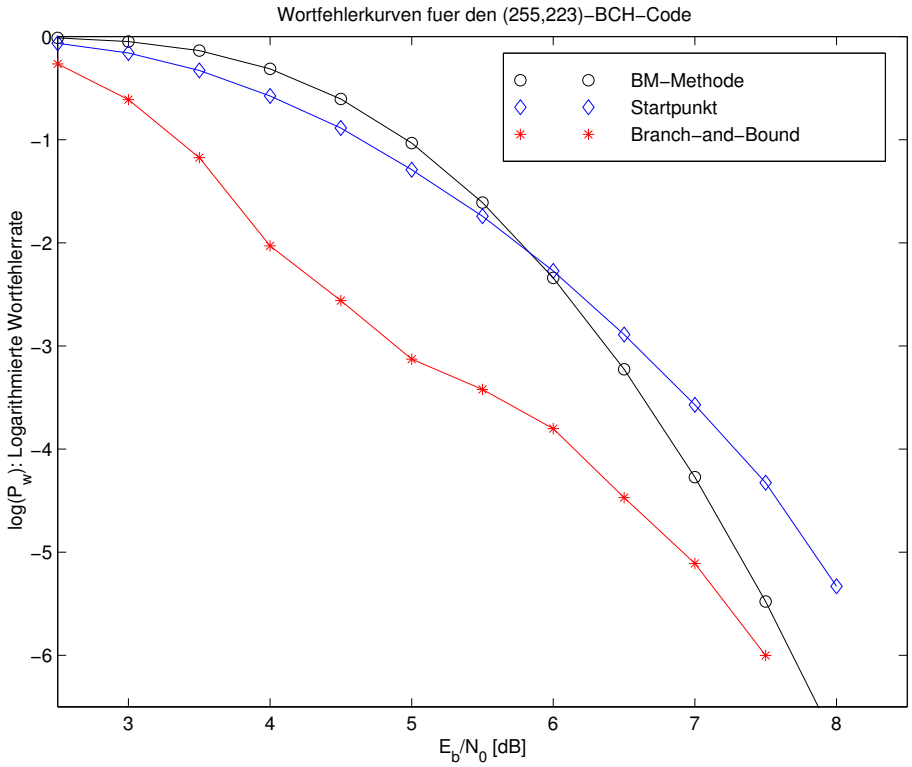

Abbildung 6.15: *Wortfehlerkurven für den (255,223)-BCH-Code*

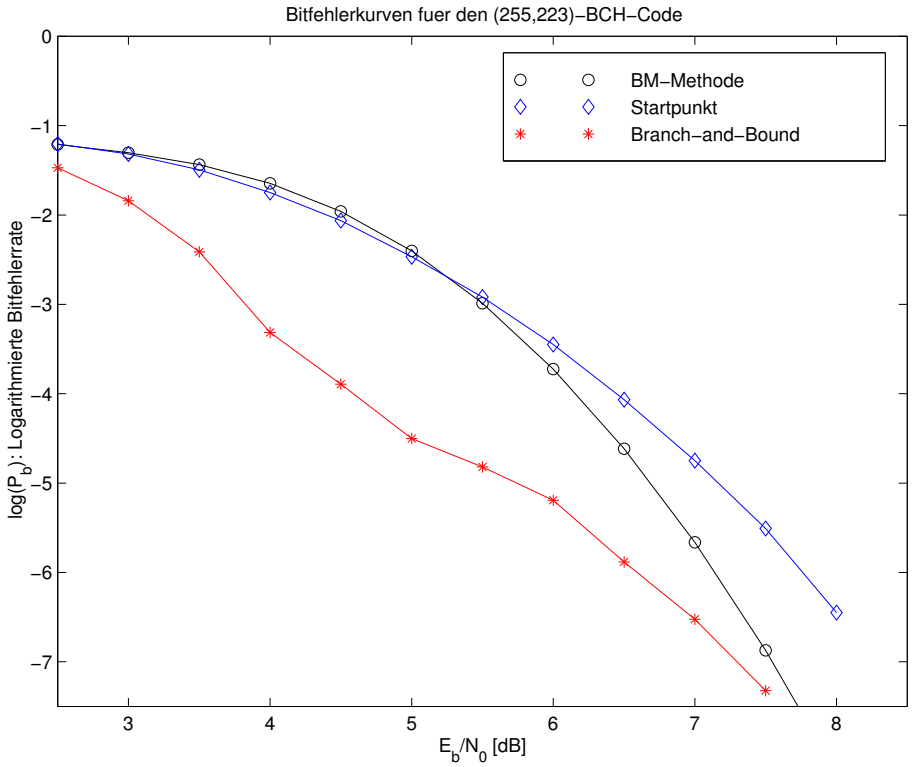

Abbildung 6.16: *Bitfehlerkurven für den (255,223)-BCH-Code*

(224,184)-Fire-Code	
Codelänge $n =$	224
Codedimension $k =$	184
Generatorpolynom $g(x) =$	$x^{40} + x^{26} + x^{23} + x^{17} + x^3 + 1$

Quellenangabe: [GSM96b]

Dieser sogenannte Fire-Code[4] findet im GSM-Mobilfunkstandard bei den Kontrollkanälen Verwendung. Die Codes dieser Kanäle sind verkettete Codes, die den Fire-Code mit einem nachgeschalteten Faltungscode verketten. Ein Beispiel für einen solchen Gesamt-Code ist der SACCH-Code, siehe Seite 158.

Hier betrachten wir den Fire-Code für sich und vergleichen die Decodierungsergebnisse mit den Ergebnissen eines klassischen Syndromkorrektur[5]-Verfahrens.

(224,184)-Fire-Code
Decodierung mit dem Branch-and-Bound Verfahren

E_b/N_0[dB]	P_w	$\log_{10}(P_w)$	P_b	$\log_{10}(P_b)$
2.5	0.3793000000	**-0.4210**	0.0236141304	**-1.6268**
3.0	0.1442000000	**-0.8410**	0.0083570652	**-2.0779**
3.5	0.0379000000	**-1.4214**	0.0020445652	**-2.6894**
4.0	0.0056905480	**-2.2448**	0.0002832903	**-3.5478**
4.5	0.0008597565	**-3.0656**	0.0000344370	**-4.4630**
5.0	0.0001003697	**-3.9984**	0.0000036002	**-5.4437**
5.5	0.0000106646	**-4.9721**	0.0000003460	**-6.4609**
6.0	0.0000015957	**-5.7970**	0.0000000530	**-7.2757**

[4] Details zur Definition von Firecodes sind etwa in [Fri95] zu finden
[5] siehe auch Seite 57

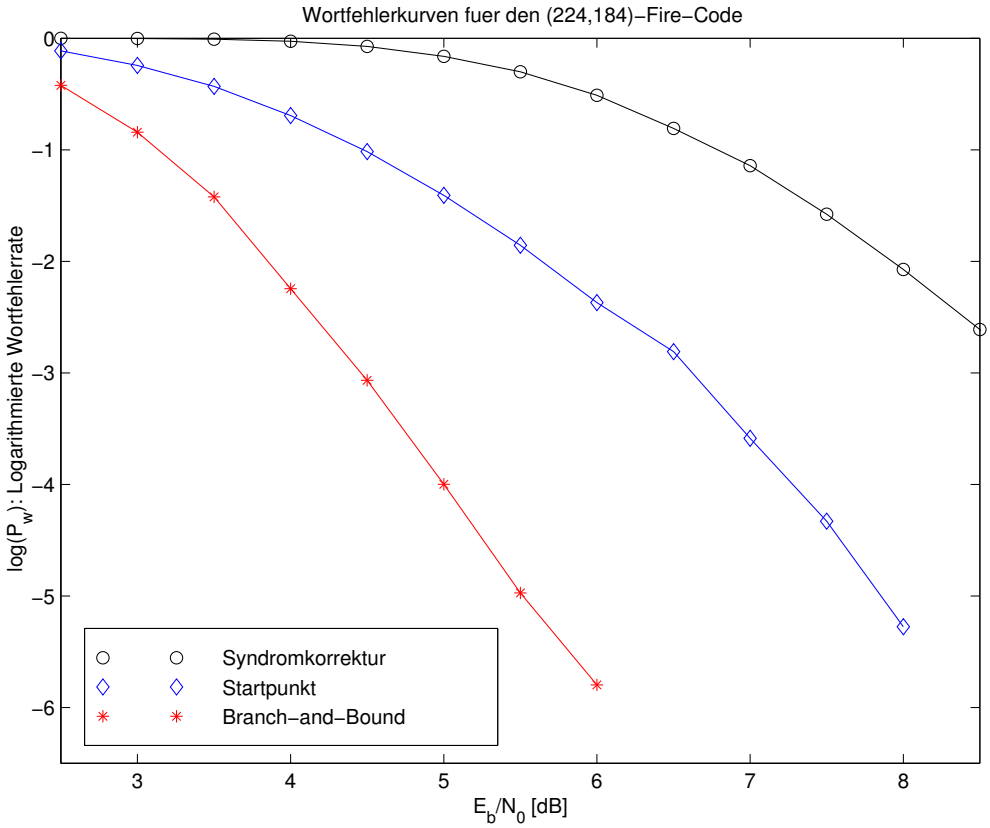

Abbildung 6.17: *Wortfehlerkurven für den (224,184)-Fire-Code*

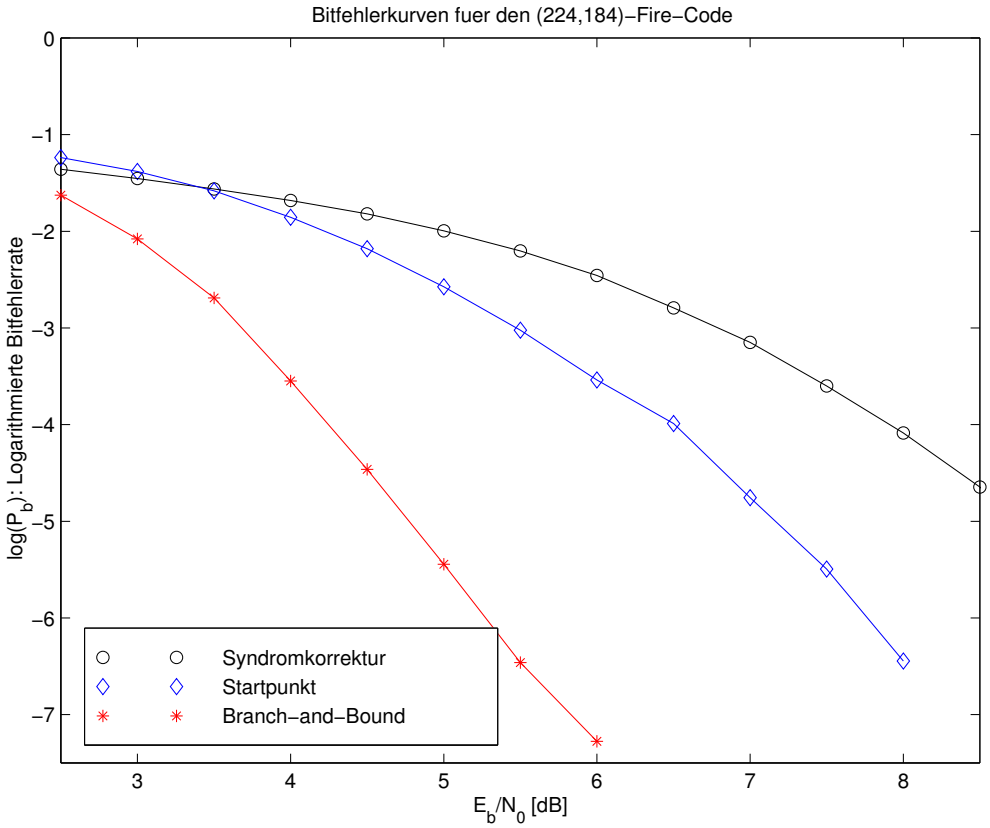

Abbildung 6.18: *Bitfehlerkurven für den (224,184)-Fire-Code*

6.3 Soft-Output Decodierung

Wie in Lemma 3.25 auf Seite 50 dargelegt, läßt sich die Kombination der Codierungsabbildung φ eines binären linearen (n,k)-Blockcodes (n,k,φ) mit einem stetigen n-Kanal $\mathcal{K}$ und einer Soft-Output Decodierungsabbildung δ_{SO} als Superkanal $\hat{\mathcal{K}}$ bezüglich $(\varphi,\mathcal{K},\delta_{SO})$ interpretieren. Abbildung 2.3 auf Seite 29 veranschaulicht den Superkanal bildhaft.

Bei Verwendung der in Kapitel 4 definierten L-Wert Soft-Output Decodierung erwarten wir, daß der Superkanal $\hat{\mathcal{K}}$ im allgemeinen „besser“ als der n-Kanal $\mathcal{K}$ ist. Ist $\mathrm{SNR}_{\mathrm{in}}$ die Signal-to-Noise Ratio auf dem Kanal $\mathcal{K}$ für den Code (n,k,φ), so berechnen wir $\mathrm{SNR}_{\mathrm{out}}$ als approximative Signal-to-Noise Ratio auf dem Superkanal $\hat{\mathcal{K}}$. In nachfolgenden Testbeispielen wird für verschiedene Codes $\mathrm{SNR}_{\mathrm{in}}$ mit $\mathrm{SNR}_{\mathrm{out}}$ verglichen.

Zur Berechnung wird der Superkanal approximativ als (verzerrter) AWGN-Kanal aufgefaßt, das heißt, für ein $\mu_L > 0$ und ein $\sigma_L^2 > 0$ wird angenommen, daß

$$\hat{\mathcal{K}}_u \sim \mathcal{N}\left(\mu_L \cdot u, \sigma_L^2 \cdot I_k\right), \quad \text{für alle } u \in \{\pm 1\}^k.$$

Somit ist dann der k-Kanal

$$\begin{aligned}\tilde{\mathcal{K}}:\ \{\pm 1\}^k \times \Omega &\to \mathbb{R}^k,\\ (u,\omega) &\mapsto \frac{1}{\mu_L}\hat{\mathcal{K}}(u,\omega),\end{aligned}$$

ein AWGN-Kanal mit bitweiser Varianz

$$\hat{\sigma}^2 = \frac{\sigma_L^2}{\mu_L^2}$$

der Kanalstörung und

$$\tilde{\mathcal{K}}_u \sim \mathcal{N}\left(u, \hat{\sigma}^2 \cdot I_k\right), \quad \text{für alle } u \in \{\pm 1\}^k.$$

$\tilde{\mathcal{K}}$ wird als unverzerrter Superkanal bezeichnet.

Bei Verwendung der L-Werte als Eingabe einer nachgeschalteten Soft-Decision Decodierung, die auf Grundlage der Soft-Decision Zielfunktion (5.1) von Seite 92 konstruiert wurde, ist die Verzerrung des AWGN-Kanals irrelevant, weil für die Soft-Decision Zielfunktion für alle $u \in \{\pm 1\}^k$ gilt

$$F(u) = \sum_{j=1}^{n}\left(\bigoplus_{i\in J_j} u_i - y_j\right)^2 = \left(n + \sum_{j=1}^{n} y_j^2\right) - 2\sum_{j=1}^{n} y_j \bigoplus_{i\in J_j} u_i$$

und entsprechend gilt für eine Zielfunktion mit obiger Verzerrung

$$\hat{F}(u) := \sum_{j=1}^{n}\left(\bigoplus_{i\in J_j} u_i - \mu_L y_j\right)^2 = \left(n + \mu_L^2 \sum_{j=1}^{n} y_j^2\right) - 2\mu_L \sum_{j=1}^{n} y_j \bigoplus_{i\in J_j} u_i.$$

Da F und $\hat{F}$ bis auf additive Konstanten und positive Skalierung identisch sind, ist die Menge der Minimierer beider Funktionen identisch.

Zur Berechnung von $\hat{\sigma}^2$ betrachten[6] wir mit gleichverteiltem U für $i = 1,\ldots,k$ Zufallsvariablen $\tilde{L}_i : \Omega \to \mathbb{R}$ (einseitige L-Werte), wobei für alle $\omega \in \Omega$ gelte

$$\begin{aligned}\tilde{L}_i(\omega) &:= U_i(\omega) \cdot L(U_i|Y(\omega)) = U_i(\omega) \cdot \tilde{\mathcal{K}}(U_i(\omega),\omega) \\ &= U_i(\omega) \cdot (\mu_L U_i(\omega) + \sigma_L Z(\omega)) = \mu_L + \sigma_L U_i(\omega) Z(\omega)\end{aligned}$$

mit $Z \sim \mathcal{N}(0,1)$ stochastisch unabhängig von U_i und folglich

$$\mathbf{E}\left(\tilde{L}_i\right) = \mu_L, \quad \mathbf{Var}\left(\tilde{L}_i\right) = \mathbf{E}\left(\sigma_L^2 U_i^2 Z^2\right) = \sigma_L^2 \mathbf{E}\left(Z^2\right) = \sigma_L^2.$$

Die theoretische Berechnung der Erwartungswerte wurde in Lemma 4.17 auf Seite 88 beschrieben. Nun wird zur empirischen[7] Berechnung wie folgt vorgegangen:

- Codewörter $c = \varphi(u) \in \{\pm 1\}^n$ mit gleichverteilt erzeugten $u \in \{\pm 1\}^k$ werden additiv durch normalverteilt erzeugte Zufallszahlen mit Varianz

  $$\sigma^2 = \frac{n}{2 \cdot k \cdot 10^{(\mathrm{SNR_{in}[dB]})/10}}.$$

  der Normalverteilung gestört.

- Mit dem resultierenden Demodulationsergebnis y werden die L-Wert Soft-Outputs $L(U_i|y)$, $i = 1,\ldots,k$, mit dem TSO Algorithmus von Kapitel 4 berechnet.

- Zur Analyse werden jeweils für $i = 1,\ldots,k$

  $$\tilde{L}_i := u_i \cdot L(U_i|y) = \begin{cases} +L(U_i|y), & \text{für } u_i = +1, \\ -L(U_i|y), & \text{für } u_i = -1. \end{cases}$$

  als einseitige L-Werte berechnet.

- Für je 100000 Codewörter werden der gemeinsame empirische Erwartungswert μ_L und die gemeinsame empirische Varianz σ_L^2 der $\tilde{L}_i$, $i = 1,\ldots,k$ berechnet.

- Als empirische Varianz der bitweisen Störung auf dem unverzerrten Superkanal wird

  $$\hat{\sigma}^2 = \frac{\sigma_L^2}{\mu_L^2}$$

  berechnet.

- Die approximative Signal-to-Noise Ratio auf dem unverzerrtem Superkanal wird zu

  $$\mathrm{SNR_{out}[dB]} = 10 \cdot \log_{10}\left(\frac{n}{2k\hat{\sigma}^2}\right)$$

  berechnet.

[6] Die numerische Berechnung der empirischen Erwartungswerte und Varianzen erfolgt über die Betrachtung von gleichverteilt erzeugten Codewörtern, weil der Superkanal nur approximativ als verzerrter AWGN-Kanal aufgefaßt werden kann und Seiteneffekte durch die Wahl eines speziellen fixen $u \in \{\pm 1\}^k$ als Kanaleingabe vermieden werden sollen.

[7] Zur Erhöhung der Übersichtlichkeit werden die empirischen Erwartungswerte und Varianzen genauso wie die Erwartungswerte und Varianzen bezeichnet.

Faltungscode des SACCH-Codes	
Codelänge $n=$	456
Codedimension $k=$	224
Eingabeblockzahl $a=$	224
Bits pro Eingabeblock $b=$	1
Eindringtiefe $l=$	5
Ausgabebits $d=$	2
Generatorpolynom $g_1(x)=$	x^4+x^3+1
Generatorpolynom $g_2(x)=$	$x^4+x^3+x^1+1$

Quellenangabe: [GSM96b]

Dieser Faltungscode findet im GSM-Mobilfunkstandard bei den Kontrollkanälen Verwendung. Die Codes dieser Kanäle sind verkettete Codes, die den Faltungscode mit einem vorgeschalteten Blockcode verketten. Ein Beispiel für einen solchen Gesamt-Code ist der SACCH-Code, siehe Seite 158.

Faltungscode des SACCH-Codes
Decodierung mit dem TSO Verfahren

SNR_{in}[dB]	σ^2	μ_L	σ_L^2	$\hat{\sigma}^2$	SNR_{out}[dB]
0.00	1.01785714	3.60485068	9.52890856	0.73327791	**1.4242**
0.50	0.90716613	4.99604500	12.97546507	0.51984067	**2.9182**
1.00	0.80851267	6.74105451	16.77950899	0.36925261	**4.4036**
1.50	0.72058767	8.83050205	20.57364745	0.26383992	**5.8635**
2.00	0.64222444	11.24675472	24.25738046	0.19177388	**7.2490**
2.50	0.57238313	13.95740274	27.74204297	0.14240631	**8.5416**
3.00	0.51013701	16.95873237	31.25579799	0.10867855	**9.7154**
3.50	0.45466008	20.29645097	35.26652284	0.08560959	**10.7516**
4.00	0.40521623	23.97912683	39.61585096	0.06889731	**11.6948**
4.50	0.36114934	28.07268980	44.62883318	0.05663012	**12.5464**
5.00	0.32187469	32.63749686	50.33584936	0.04725456	**13.3324**
5.50	0.28687112	37.72662557	56.92807899	0.03999729	**14.0566**
6.00	0.25567415	43.42695732	64.46283622	0.03418146	**14.7390**
6.50	0.22786983	49.83576754	72.93560012	0.02936684	**15.3983**
7.00	0.20308920	57.00064046	82.51486818	0.02539643	**16.0291**
7.50	0.18100344	65.05057166	93.58582070	0.02211606	**16.6298**
8.00	0.16131949	74.08067377	106.02653449	0.01931989	**17.2168**

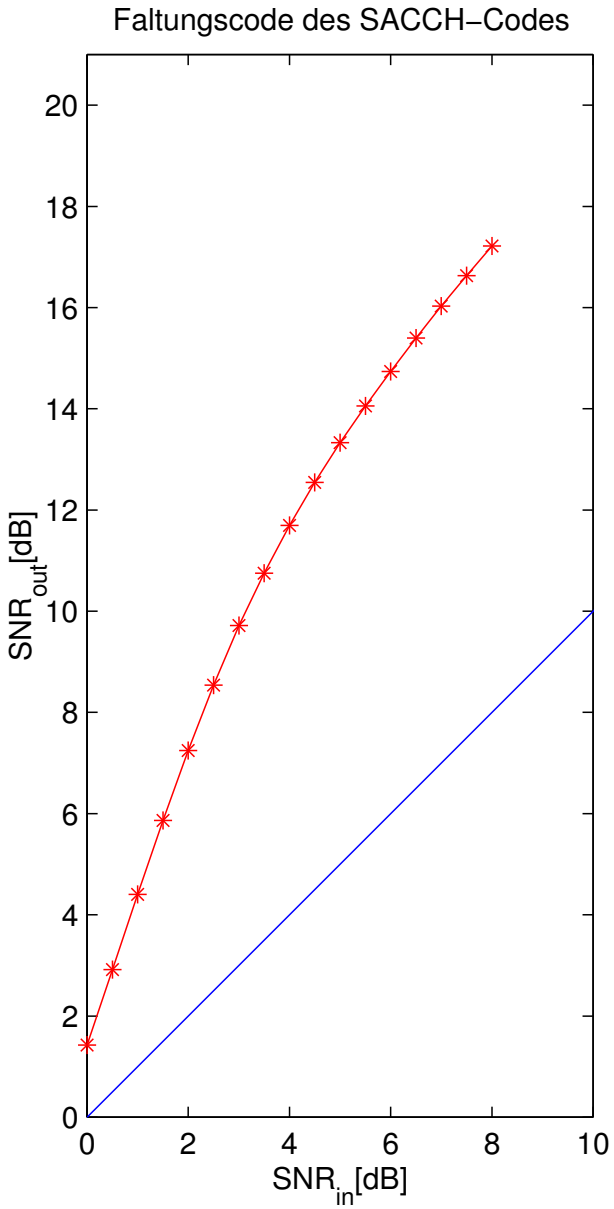

Abbildung 6.19: *$SNR_{in}[dB]$ zu $SNR_{out}[dB]$ für den Faltungscode des SACCH-Codes*

Industriestandard-1/2-Code	
Codelänge $n=$	524
Codedimension $k=$	256
Eingabeblockzahl $a=$	256
Bits pro Eingabeblock $b=$	1
Eindringtiefe $l=$	7
Ausgabebits $d=$	2
Generatorpolynom $g_1(x)=$	$x^6+x^5+x^3+x^2+1$
Generatorpolynom $g_2(x)=$	$x^6+x^3+x^2+x+1$

Quellenangabe: [Fri95]

Die sogenannten Industriestandard-Codes sind auf Codierer-/Decodierer-Chips verschiedener Firmen verfügbar. Exemplarisch werden hier $k=256$ Infobits codiert.

Industriestandard-1/2-Code
Decodierung mit dem TSO Verfahren

SNR_{in}[dB]	σ^2	μ_L	σ_L^2	$\hat{\sigma}^2$	SNR_{out}[dB]
0.00	1.02343750	3.10388512	10.29438834	1.06853627	**-0.1873**
0.50	0.91213963	4.95246080	15.63195658	0.63734010	**2.0569**
1.00	0.81294530	7.39790243	21.09608072	0.38546467	**4.2408**
1.50	0.72453826	10.34017789	25.62922922	0.23970634	**6.3038**
2.00	0.64574541	13.62942231	29.16292240	0.15699150	**8.1419**
2.50	0.57552120	17.20643976	32.36529084	0.10931946	**9.7136**
3.00	0.51293381	21.11957883	35.62989518	0.07988110	**11.0762**
3.50	0.45715274	25.39266496	39.55720114	0.06134921	**12.2225**
4.00	0.40743781	30.13627485	44.24988291	0.04872288	**13.2233**
4.50	0.36312933	35.40618106	49.58086341	0.03955086	**14.1291**
5.00	0.32363935	41.30895532	55.79251334	0.03269546	**14.9557**
5.50	0.28844388	47.92150180	62.89359603	0.02738707	**15.7252**
6.00	0.25707588	55.35453682	70.88848492	0.02313499	**16.4579**
6.50	0.22911912	63.75786161	79.95803029	0.01966956	**17.1627**
7.00	0.20420263	73.20667682	90.17796102	0.01682670	**17.8406**
7.50	0.18199578	83.82725929	101.70801294	0.01447387	**18.4948**
8.00	0.16220391	95.83363784	114.77386084	0.01249704	**19.1325**

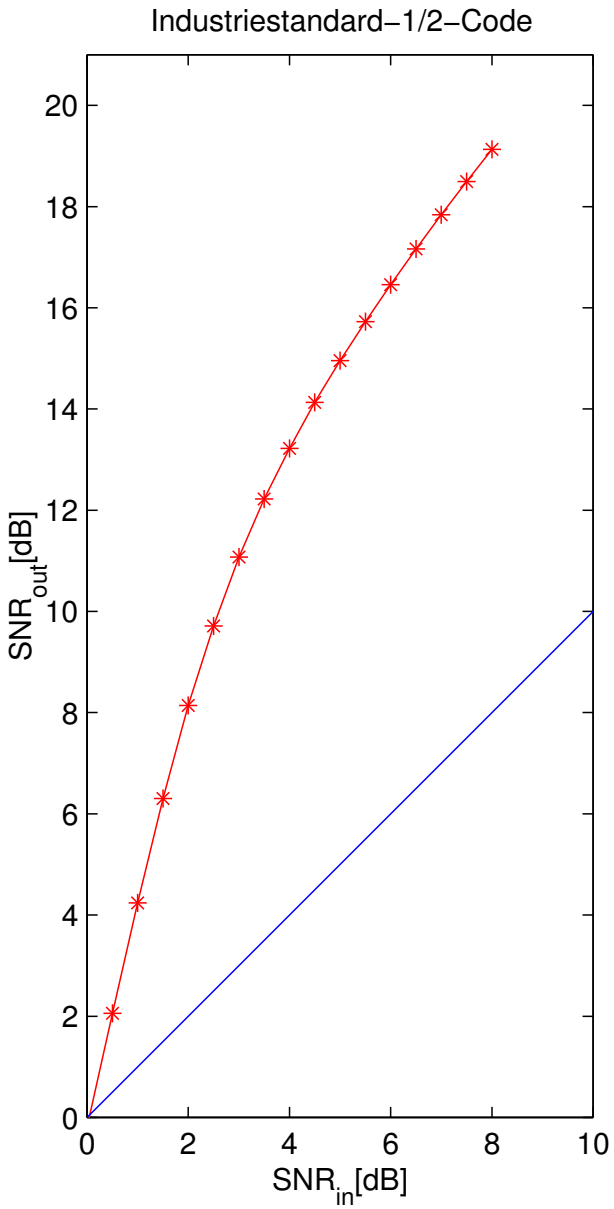

Abbildung 6.20: *$SNR_{in}[dB]$ zu $SNR_{out}[dB]$ für den Industriestandard-1/2-Code*

Industriestandard-1/3-Code	
Codelänge $n=$	786
Codedimension $k=$	256
Eingabeblockzahl $a=$	256
Bits pro Eingabeblock $b=$	1
Eindringtiefe $l=$	7
Ausgabebits $d=$	3
Generatorpolynom $g_1(x)=$	$x^6+x^5+x^3+x^2+1$
Generatorpolynom $g_2(x)=$	x^6+x^4+x+1
Generatorpolynom $g_3(x)=$	$x^6+x^4+x^3+x^2+x+1$

Quellenangabe: [Fri95]

Die sogenannten Industriestandard-Codes sind auf Codierer-/Decodierer-Chips verschiedener Firmen verfügbar. Exemplarisch werden hier $k=256$ Infobits codiert.

Industriestandard-1/3-Code
Decodierung mit dem TSO Verfahren

SNR_{in}[dB]	σ^2	μ_L	σ_L^2	$\hat{\sigma}^2$	SNR_{out}[dB]
0.00	1.53515625	5.17368399	15.96093484	0.59629138	**4.1069**
0.50	1.36820945	7.40299264	20.95112992	0.38228989	**6.0376**
1.00	1.21941795	10.02952045	25.32010191	0.25171269	**7.8525**
1.50	1.08680740	12.99287430	28.98111031	0.17167401	**9.5145**
2.00	0.96861811	16.24968366	32.21120199	0.12198800	**10.9984**
2.50	0.86328180	19.79778616	35.58098898	0.09077887	**12.2817**
3.00	0.76940071	23.69433132	39.28666343	0.06997715	**13.4120**
3.50	0.68572911	28.02345871	43.58469177	0.05549968	**14.4186**
4.00	0.61115671	32.83625238	48.71628825	0.04518216	**15.3119**
4.50	0.54469399	38.21434821	54.42203767	0.03726678	**16.1483**
5.00	0.48545903	44.23693693	60.97198143	0.03115733	**16.9259**
5.50	0.43266582	51.00969299	68.32902915	0.02626031	**17.6685**
6.00	0.38561382	58.67096891	76.65238392	0.02226789	**18.3847**
6.50	0.34367867	67.29751299	86.10596289	0.01901232	**19.0712**
7.00	0.30630394	76.97620379	96.55581475	0.01629542	**19.7409**
7.50	0.27299368	87.91541658	108.33888943	0.01401698	**20.3950**
8.00	0.24330587	100.24299347	121.90650391	0.01213162	**21.0223**

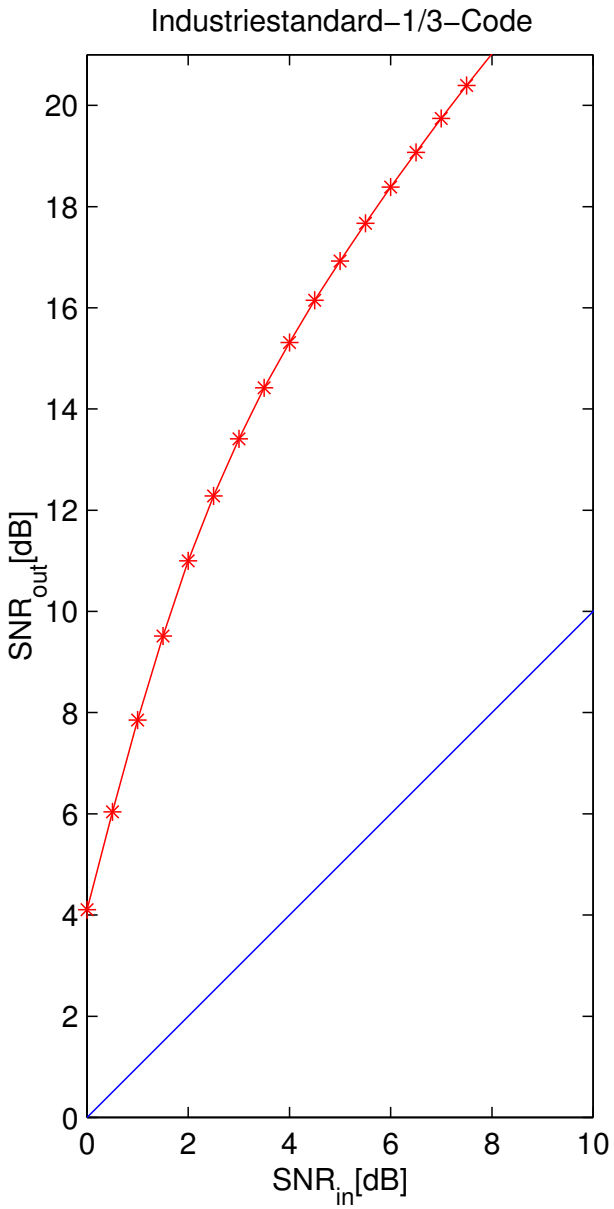

Abbildung 6.21: *$SNR_{in}[dB]$ zu $SNR_{out}[dB]$ für den Industriestandard-1/3-Code*

Telemetrie-Faltungscode	
Codelänge $n =$	4092
Codedimension $k =$	2040
Eingabeblockzahl $a =$	2040
Bits pro Eingabeblock $b =$	1
Eindringtiefe $l =$	7
Ausgabebits $d =$	2
Generatorpolynom $g_1(x) =$	$x^6+x^5+x^4+x^3+1$
Generatorpolynom $g_2(x) =$	$x^6+x^4+x^3+x^1+1$

Quellenangabe: [CCS92]

Dieser Code wurde vom CCSDS[8] als Faltungscode für das Telemetry Channel Coding als gemeinsamer Standard der verschiedenen Raumfahrtagenturen definiert (optional in Verkettung mit einem Reed-Solomon-Code). Die zugehörige Recommendation [CCS92] ist auch als ISO 11754:1994 genormt.

Telemetrie-Faltungscode
Decodierung mit dem TSO Verfahren

SNR_{in}[dB]	σ^2	μ_L	σ_L^2	$\hat{\sigma}^2$	SNR_{out}[dB]
0.00	1.00294118	3.04470767	9.72471621	1.04902475	**-0.1951**
0.50	0.89387226	5.01078321	15.32432911	0.61033777	**2.1571**
1.00	0.79666449	7.57550770	20.91806536	0.36450042	**4.3958**
1.50	0.71002798	10.61722384	25.45461132	0.22581070	**6.4753**
2.00	0.63281310	14.00672775	28.94714154	0.14754765	**8.3234**
2.50	0.56399527	17.67570859	32.12053376	0.10280852	**9.8925**
3.00	0.50266131	21.67180668	35.49843477	0.07558210	**11.2286**
3.50	0.44799737	26.03600814	39.46010971	0.05821159	**12.3627**
4.00	0.39927807	30.86801211	44.15391087	0.04633955	**13.3532**
4.50	0.35585696	36.24220161	49.60194747	0.03776327	**14.2421**
5.00	0.31715785	42.26408323	55.86105258	0.03127276	**15.0611**
5.50	0.28266723	49.01750101	63.00732310	0.02622338	**15.8259**
6.00	0.25192743	56.61414927	71.11170143	0.02218662	**16.5518**
6.50	0.22453056	65.16843067	80.29001754	0.01890545	**17.2469**
7.00	0.20011307	74.80165983	90.62240524	0.01619620	**17.9186**
7.50	0.17835096	85.66437484	102.25450240	0.01393420	**18.5719**
8.00	0.15895546	97.90117516	115.24728564	0.01202416	**19.2122**

[8] CCSDS = Consultative Committee for Space Data Systems

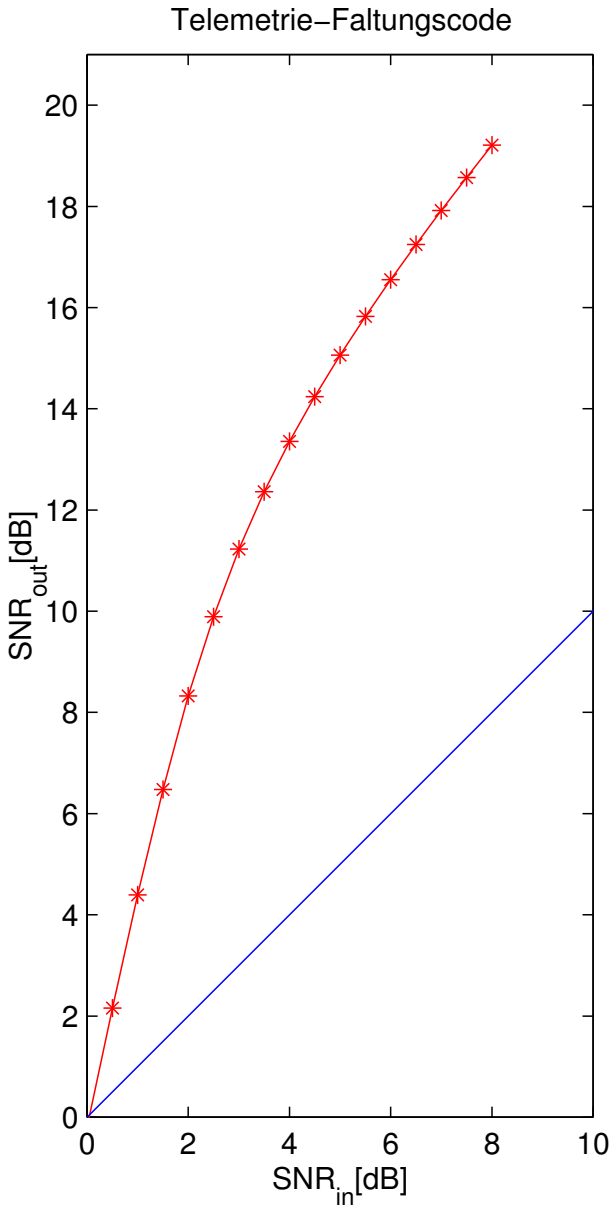

Abbildung 6.22: *$SNR_{in}[dB]$ zu $SNR_{out}[dB]$ für den Telemetrie-Faltungscode*

6.4 Decodierung des verketteten SACCH-Codes

SACCH-Code	
Codelänge $n =$	456
Codedimension $k =$	184
1. Teilcodelänge $n_1 =$	224
1. Teilcodedimension $k_1 =$	184
1. Teilcodierungsabb. $\varphi_1 =$	(224,184)-Fire-Code, siehe Seite 146
2. Teilcodelänge $n_2 =$	456
2. Teilcodedimension $k_2 =$	224
2. Teilcodierungsabb. $\varphi_2 =$	Faltungscode des SACCH-Codes, siehe Seite 150

Quellenangabe: [GSM96b, DB96]

Im Mobilfunk wird der SACCH[9]-Code für einen GSM[10]-Kontrollkanal eingesetzt. Es handelt sich um einen verketteten binären linearen $(456, 184)$-Blockcode $((224, 184, \varphi_1), (456, 224, \varphi_2))$, der aus einem Fire-Code und einem terminierten Faltungscode zusammengesetzt ist[11].

Hier wird dieser Code mit dem TSOBB[12] Verfahren behandelt und mit einer klassischen Kombination aus Viterbi-Decodierer und Syndromkorrektur verglichen.

SACCH-Code
Decodierung mit dem TSOBB-Verfahren

E_b/N_0[dB]	P_w	$\log_{10}(P_w)$	P_b	$\log_{10}(P_b)$
1.5	0.4902000000	**-0.3096**	0.0490826087	**-1.3091**
2.0	0.2066000000	**-0.6849**	0.0180282609	**-1.7440**
2.5	0.0554000000	**-1.2565**	0.0043206522	**-2.3645**
3.0	0.0099000000	**-2.0044**	0.0007108696	**-3.1482**
3.5	0.0010136642	**-2.9941**	0.0000622522	**-4.2058**
4.0	0.0000600194	**-4.2217**	0.0000033514	**-5.4748**
4.5	0.0000031125	**-5.5069**	0.0000001402	**-6.8534**

[9] SACCH = Slow Associated Control Channel

[10] GSM = Global System for Mobile communications

[11] In der Spezifikation [GSM96b] besteht die Ausgabe des ersten Codierers / die Eingabe des zweiten Codierers aus 228 Bits. Der Grund ist aber nur eine andere Zuordnung der Terminierungsbits des Faltungscodes als in der hier vorliegenden Terminologie.

[12] TSOBB = TSO (Kapitel 4) + Branch-and-Bound (Kapitel 5)

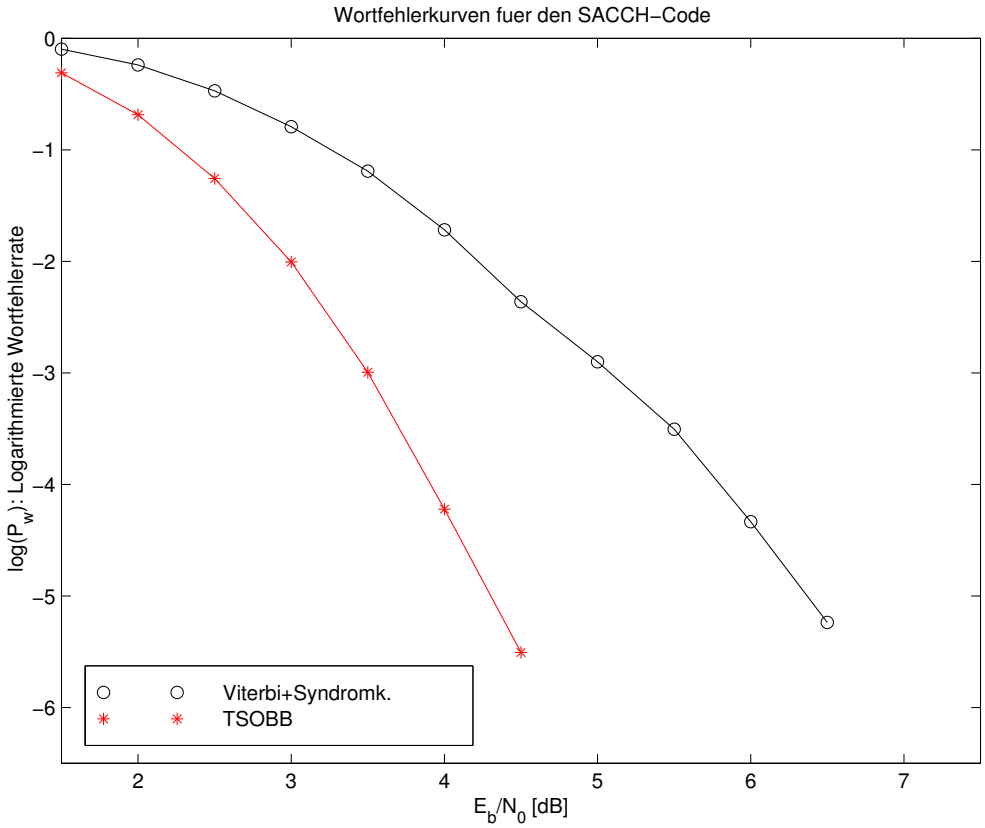

Abbildung 6.23: *Wortfehlerkurven für den SACCH-Code*

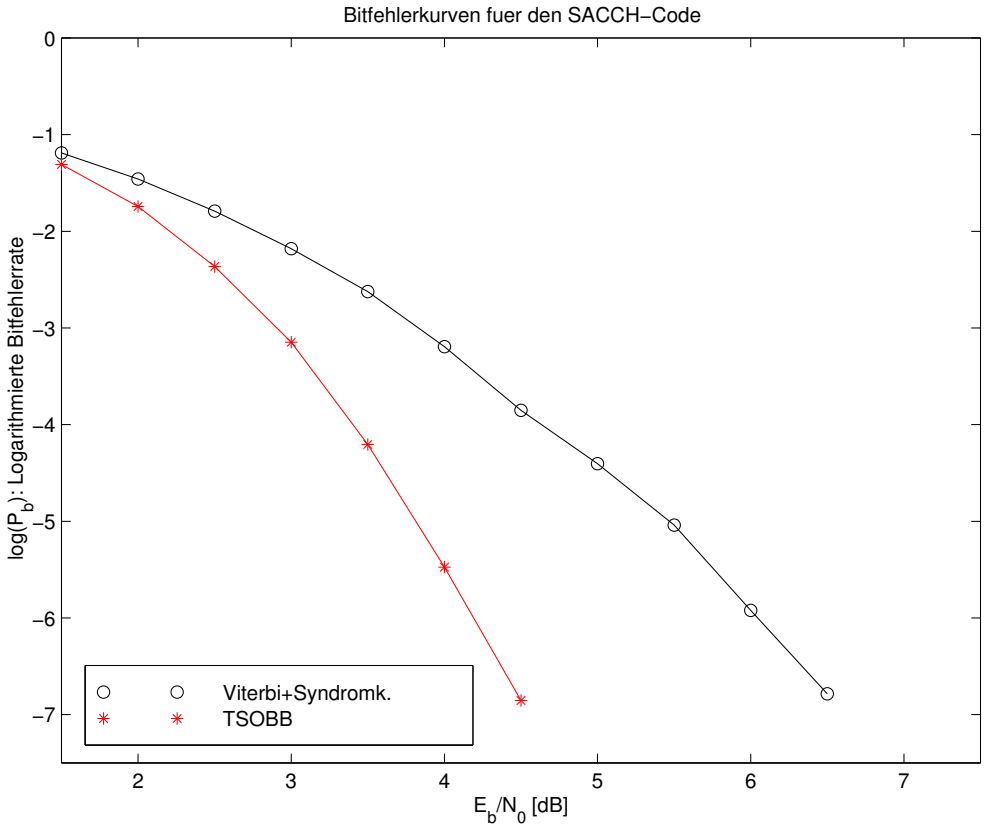

Abbildung 6.24: *Bitfehlerkurven für den SACCH-Code*

Kapitel 7

Zusammenfassung und Ausblick

Für die digitale Zeichenübertragung über analoge Kanäle mit Störung werden in der Nachrichtentechnik Fehlerschutzcodes verwendet, die aufgrund redundant übertragener Informationen die Rekonstruktion der übertragenen Nachrichten bis auf einen Restfehler erlauben. Eine der wesentlichen Fragestellungen der Nachrichtentechnik besteht in der Suche nach Decodierungsverfahren, die diesen Restfehler unter geeigneten Voraussetzungen minimieren.

In der vorliegenden Arbeit wurde zunächst eine exakte stochastische Modellierung des abstrakten Kanalmodells (*n*-Kanal) für beliebige binäre lineare Blockcodes vorgestellt, die im wesentlichen auf einer Kanalbeschreibung über Wahrscheinlichkeitsdichtefunktionen beruht[1]. Auf Basis dieses Modells wurde eine zweifelsfreie mathematische Behandlung der abgeleiteten Fragestellungen mit stochastischen Werkzeugen möglich. Insbesondere wurden die prinzipiellen Decodierungsmethodiken klassifiziert und Kriterien zur Konstruktion *bester* Decodierungsverfahren in allgemeiner Form hergeleitet. Verglichen mit den Darstellungen der Literatur [Fri95, HQ95, Jun95, Roh95, Bos98, Bos99, Pro01, CCR01] konnten verallgemeinerte Aussagen mit beliebigen stetigen Dichten formuliert und bewiesen werden, siehe etwa Satz 3.14, Satz 3.17, Lemma 3.10, Lemma 3.25, Lemma 3.27 und Satz 3.29.

Für die Klasse der terminierten Faltungscodes wurde ein optimales Soft-Output Verfahren hergeleitet und analysiert, welches funktional vergleichbar mit dem MAP-Verfahren [BCJR74] ist, und mit dem die auch für Turbo-Codes wichtigen L-Wert Soft-Outputs berechnet werden können. Die vorgestellte Methode wurde möglichst operationsoptimiert mit einem Rekursionsansatz entwickelt und durch Berechnung der Erwartungswerte von Teilformeln numerisch stabil formuliert. Die Verbesserung des Signal-Rausch-Verhältnisses auf dem durch die Soft-Outputs definierten Superkanal wurde in numerischen Beispielen für Standard-Faltungscodes berechnet und dargestellt.

Die Soft-Decision Decodierung beliebiger binärer linearer Blockcodes wurde durch ein neues fehleroptimales Verfahren behandelt. Dabei wurde mit Hilfe einer spezifischen adaptiven Transformation der zu betrachtende Code empfängerseitig in einen anderen Code transformiert, der besonders operationsgünstig für ein Branch-and-Bound Verfahren konstruiert wurde. Durch die fehleroptimale Decodierung im transformierten Bereich waren deutlich weniger numerische Schritte notwendig als bei einem gewöhnlichen Branch-and-Bound Verfahren auf dem originalen Codebaum. Bei den vorgestellten numerischen Beispielen konnten daher Codes mit Codelängen bis 255 mit diesem Verfahren noch wortfehleroptimal decodiert werden.

Schließlich wurde am Beispiel des SACCH-Codes auch die Verbesserung für die Decodierung

[1] In diesem Zusammenhang sei darauf hingewiesen, daß der informationstheoretischen Begriff der Entropie [Sha48] in seiner allgemeinen Form ausschließlich über Wahrscheinlichkeitsdichtefunktionen definiert ist.

verketteter Codes durch den Einsatz der beiden Verfahren im Vergleich zur Viterbi-Decodierung mit Syndromkorrektur numerisch nachgewiesen.

Insgesamt wurden in der vorliegenden Arbeit die Fragestellungen zur fehleroptimalen Decodierung von beliebigen binären linearen Blockcodes in allgemeiner Form stochastisch modelliert und analysiert. Für AWGN-Kanäle ohne Einschränkung auf Codeklassen wurde ein optimales Soft-Decision Verfahren vollständig entwickelt. Für die Codeklasse der terminierten Faltungscodes wurde die Soft-Output Decodierung ebenfalls vollständig durch ein optimales Verfahren beschrieben.

Ausgehend von den Ergebnissen dieser Arbeit lassen sich weitere Forschungsschwerpunkte für die Zukunft identifizieren:

- Die Entwicklung von Soft Decodierungsverfahren für andere Kanäle als AGWN auf Basis der allgemeinen Herleitung von Kapitel 3. Eine naheliegende Erweiterung ist z.B. die Verwendung von Kanaldichten

$$f_c(x) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} \exp\left(-\frac{(x-c)^\top \Sigma^{-1}(x-c)}{2}\right), \quad \text{für alle } x \in \mathbb{R}^n,$$

  die für positiv definites $\Sigma \in \mathbb{R}^{n,n}$ Kanäle mit Gedächtnis beschreiben.

- Die Entwicklung von suboptimalen Soft-Decision Verfahren. Da das Branch-and-Bound Verfahren naturgemäß als Suchverfahren formuliert ist, läßt sich das Suchgebiet durch Zusatzanforderungen einschränken, die das Verfahren somit beschleunigen auf Kosten der Optimalität. Die Einschränkung der Hamming-Distanz zur Startlösung etwa erzeugt eine solche Verfahrensklasse.

- Die Anwendung der Entwicklungsschritte der Verfahren für spezielle Codeklassen, um daraus neue spezialisierte Verfahren abzuleiten.

Anhang A

Wahrscheinlichkeitstheoretische Begriffe und Grundlagen

And twenty-five miles away to the northwest in Las Vegas, every pair of dice on every Craps table had come up snakeeyes in the instant of Snayheever's death, and every roulette ball rocked to a solid halt in the 00 slot, and every car in town that had its key turned in the ignition at that moment started up instantly.

(Tim Powers, „Last Call")

Im folgenden werden grundlegende Begriffe der Wahrscheinlichkeitstheorie eingeführt, die in dieser Arbeit Verwendung finden. Dieser Überblick folgt der Darstellung in [SS94, SS95, Sch97].

Wir betrachten eine beliebige nichtleere Basismenge Ω. Eine Menge $\mathcal{F} \subseteq \mathcal{P}(\Omega)$ wird als Mengensystem (über Ω) bezeichnet, wobei $\mathcal{P}$ die Potenzmenge (Menge aller Teilmengen) von Ω darstellt. Mit $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$ wird eine Erweiterung der Menge aller reellen Zahlen definiert. Die algebraische Struktur von $\mathbb{R}$ wird folgendermaßen auf $\overline{\mathbb{R}}$ erweitert: Für alle $a \in \mathbb{R}$ gilt:

$$a + (\pm\infty) = (\pm\infty) + a = (\pm\infty) + (\pm\infty) = (\pm\infty), \quad +\infty - (-\infty) = +\infty,$$

$$a \cdot (\pm\infty) = (\pm\infty) \cdot a = \begin{cases} (\pm\infty), & \text{für } a > 0, \\ 0, & \text{für } a = 0, \\ (\mp\infty), & \text{für } a < 0, \end{cases}$$

$$(\pm\infty) \cdot (\pm\infty) = +\infty, \quad (\pm\infty) \cdot (\mp\infty) = -\infty, \quad \frac{a}{\pm\infty} = 0.$$

Somit ist $\overline{\mathbb{R}}$ kein Körper. Die Vorzeichen bei $\pm\infty$ dürfen bei den obigen Formeln nicht kombiniert werden, denn der Ausdruck $+\infty - (+\infty)$ ist nicht definiert. Die Bedeutung der Festlegung $0 \cdot (\pm\infty) = (\pm\infty) \cdot 0 = 0$ wird später deutlich. Vorsicht ist allerdings bei den Grenzwertsätzen geboten:

$$\lim_{x \to +\infty} \left(x \cdot \frac{1}{x} \right) \neq (+\infty) \cdot 0 = 0.$$

Ergänzt man die Ordnungsstruktur von $\mathbb{R}$ durch $-\infty < a$, $a < +\infty$ für alle $a \in \mathbb{R}$ und $-\infty < +\infty$, so ist $(\overline{\mathbb{R}}, \leq)$ eine geordnete Menge. Aufgrund topologischer Überlegungen können wir unter Verzicht auf die entsprechenden Grenzwertsätze vereinbaren, daß die Folge $\{n\}_{n \in \mathbb{N}}$ den Grenzwert $+\infty \in \overline{\mathbb{R}}$

besitzt. Für „$+\infty$“ schreiben wir oft „∞“. In Analogie zur Berechnung von Volumina in der Geometrie versucht man, Mengen aus einem Mengensystem $\mathcal{F}$ über Ω Maße (Volumina) zuzuordnen. Zu diesem Zweck zeichnet man spezielle Funktionen aus.

Definition A.1 (σ-endliches) Maß)
Sei $\mathcal{F} \subseteq \mathcal{P}(\Omega)$, $\emptyset \in \mathcal{F}$. Eine Funktion $\mu\colon \mathcal{F} \to \bar{\mathbb{R}}$ heißt Maß auf $\mathcal{F}$, falls die folgenden Bedingungen erfüllt sind:

(M1) $\mu(A) \geq 0$ *für alle* $A \in \mathcal{F}$,

(M2) $\mu(\emptyset) = 0$,

(M3) *Für jede Folge* $\{A_i\}_{i\in\mathbb{N}}$ *paarweise disjunkter Mengen mit* $A_i \in \mathcal{F}$, $i \in \mathbb{N}$, *und* $\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}$ *gilt:*

$$\mu\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mu(A_i) \quad (\sigma\text{-Additivität}).$$

Besitzen für eine Folge $\{B_i\}_{i\in\mathbb{N}}$ *mit* $B_i \subseteq B_{i+1}$, $B_i \in \mathcal{F}$ *und* $\bigcup_{i=1}^{\infty} B_i = \Omega$ *alle Mengen* B_i, $i \in \mathbb{N}$, *ein endliches Maß, so wird* μ *als* σ*-endlich bezeichnet.*

Es wäre naheliegend, Maße auf der Potenzmenge von Ω zu betrachten. Allerdings ist diese Vorgehensweise problematisch, da es zum Beispiel nicht möglich ist, ein translationsinvariantes Maß μ auf der Potenzmenge des $\mathbb{R}^3$ mit $\mu(\mathbb{R}^3) = 1$ zu finden. Daher hat man sich im allgemeinen mit speziellen Mengensystemen über Ω (Teilmengen der Potenzmenge) zu begnügen. Dies führt auf den Begriff der σ-Algebra.

Definition A.2 (σ-Algebra)
Ein Mengensystem $\mathcal{S} \subseteq \mathcal{P}(\Omega)$ heißt σ-Algebra über Ω, falls die folgenden Axiome erfüllt sind:

(S1) $\Omega \in \mathcal{S}$,

(S2) *Aus* $A \in \mathcal{S}$ *folgt* $A^c := \Omega \setminus A \in \mathcal{S}$,

(S3) *Aus* $A_i \in \mathcal{S}$, $i \in \mathbb{N}$, *folgt* $\bigcup_{i=1}^{\infty} A_i \in \mathcal{S}$.

Die folgende Eigenschaft von σ-Algebren ist wichtig.

Satz A.3 (Durchschnittsstabilität von σ-Algebren)
Sei I eine beliebige nichtleere Menge und $\mathcal{S}_i$ für jedes $i \in I$ eine σ-Algebra über Ω, so ist auch $\bigcap_{i\in I} \mathcal{S}_i$ eine σ-Algebra über Ω. Diese Eigenschaft wird Durchschnittsstabilität von σ-Algebren genannt.

Wir können also von erzeugten σ-Algebren sprechen.

Definition A.4 (erzeugte σ-Algebra)
Sei $\mathcal{F} \subseteq \mathcal{P}(\Omega)$ und sei Σ die Menge aller σ-Algebren über Ω, die $\mathcal{F}$ enthalten, dann wird die σ-Algebra $\sigma(\mathcal{F}) := \bigcap_{\mathcal{S}\in\Sigma} \mathcal{S}$ als die von $\mathcal{F}$ erzeugte σ-Algebra bezeichnet.

Für $\Omega = \mathbb{R}^n$, $n \in \mathbb{N}$, betrachten wir die σ-Algebra

$$\mathcal{B}^n = \sigma(\{([a_1,b_1[\times\ldots\times[a_n,b_n[)\cap\mathbb{R}^n;\ -\infty \le a_i \le b_i \le \infty, i = 1,\ldots,n\}),$$

wobei $[a_1,b_1[\times\ldots\times[a_n,b_n[:= \emptyset$, falls $a_j \ge b_j$ für mindestens ein $j \in \{1,\ldots,n\}$. Auf dieser σ-Algebra läßt sich nun ein eindeutiges Maß λ^n durch

$$\lambda^n(([a_1,b_1[\times\ldots\times[a_n,b_n[)\cap\mathbb{R}^n) = \begin{cases} \prod_{i=1}^{n}(b_i-a_i), & \text{falls } b_i > a_i,\ i = 1,\ldots,n \\ 0, & \text{sonst} \end{cases}$$

festlegen. Dieses Maß heißt Lebesgue-Borel-Maß. Die σ-Algebra $\mathcal{B}^n$ wird als Borelsche σ-Algebra bezeichnet. Alle für die Praxis wichtigen Teilmengen des $\mathbb{R}^n$ (etwa alle offenen, abgeschlossenen und kompakten Teilmengen) sind in $\mathcal{B}^n$ enthalten. Das Maß λ^n ist unter allen translationsinvarianten Maßen μ auf $\mathcal{B}^n$ das einzige Maß mit $\mu([0,1[\times\ldots\times[0,1[) = 1$. Sei nun μ ein Maß auf einer σ-Algebra $\mathcal{S}$ über Ω, so heißt jede Menge $A \in \mathcal{S}$ mit $\mu(A) = 0$ eine μ-Nullmenge. Es ist nun naheliegend, jeder Teilmenge $B \subseteq A$ einer μ-Nullmenge ebenfalls das Maß $\mu(B) = 0$ zuzuordnen. Allerdings ist nicht gewährleistet, daß für jedes $B \subseteq A$ auch $B \in \mathcal{S}$ gilt. Das führt zum Begriff der Vervollständigung und des vollständigen Maßes.

Definition A.5 (vollständiges Maß, Vervollständigung)
Ein Maß μ auf einer σ-Algebra $\mathcal{S}$ über Ω heißt vollständig, falls jede Teilmenge einer μ-Nullmenge zu $\mathcal{S}$ gehört und damit eine μ-Nullmenge ist. Ist μ nicht vollständig, so heißt die σ-Algebra

$$\mathcal{S}_0 := \{A\cup N;\ A \in \mathcal{S},\ N \text{ Teilmenge einer } \mu\text{-Nullmenge}\}$$

μ-Vervollständigung von $\mathcal{S}$. Mit $\mu_0(A\cup N) := \mu(A)$ ist μ_0 ein vollständiges Maß auf $\mathcal{S}_0$.

Die Mengen der σ-Algebra $\mathcal{B}_0^n$ heißen Lebesgue-meßbare Mengen. Das Maß λ_0^n auf $\mathcal{B}_0^n$ heißt Lebesgue-Maß. Die zugehörigen Nullmengen heißen Lebesguesche Nullmengen.

Betrachtet man eine Funktion $F : \mathbb{R} \to \mathbb{R}$ mit folgenden Eigenschaften:

- F ist monoton steigend,
- F ist stetig von links,

so ist durch

$$\mu^F([a,b[\cap\mathbb{R}) := \begin{cases} F(b)-F(a), & \text{falls } -\infty < a < b < \infty \\ \lim_{b\to\infty} F(b)-F(a), & \text{falls } -\infty < a < b = \infty \\ F(b)-\lim_{a\to-\infty} F(a), & \text{falls } -\infty = a < b < \infty \\ \lim_{b\to\infty} F(b)-\lim_{a\to-\infty} F(a), & \text{falls } -\infty = a,\ b = \infty \\ 0, & \text{falls } a \ge b \end{cases}$$

ein eindeutiges Maß μ^F auf $\mathcal{B}$ definiert. Dieser Sachverhalt führt zu folgender Definition.

Definition A.6 (maßerzeugende Funktion)
Eine monoton steigende Funktion $F : \mathbb{R} \to \mathbb{R}$, die stetig von links ist, heißt maßerzeugende Funktion.

Das Maß μ^F heißt Lebesgue-Borel-Stieltjes-Maß. Das vollständige Maß μ_0^F auf der μ^F-Vervollständigung $\mathcal{B}_0^F$ von $\mathcal{B}$ heißt Lebesgue-Stieltjes-Maß. Die Mengen $A \in \mathcal{B}_0^F$ heißen Lebesgue-Stieltjes-meßbar. Durch analoge Vorgehensweise lassen sich maßerzeugende Funktionen auf $\Omega = \mathbb{R}^n$ definieren. Wir wollen darauf aber nicht näher eingehen.

Definition A.7 (Meßraum, Maßraum)
Ist $\mathcal{S}$ eine σ-Algebra über Ω, so heißt das Paar $(\Omega, \mathcal{S})$ Meßraum. Ist μ ein Maß auf $\mathcal{S}$, so heißt das Tripel $(\Omega, \mathcal{S}, \mu)$ Maßraum.

Nun untersuchen wir spezielle Funktionen zwischen zwei Grundmengen $\Omega_1, \Omega_2 \neq \emptyset$.

Definition A.8 (meßbare Abbildung)
Seien $(\Omega_1, \mathcal{S}_1)$ und $(\Omega_2, \mathcal{S}_2)$ zwei Meßräume.
Eine Abbildung $T : \Omega_1 \to \Omega_2$ mit $T^{-1}(A') := \{x \in \Omega_1;\ T(x) \in A'\} \in \mathcal{S}_1$ für alle $A' \in \mathcal{S}_2$ heißt $\mathcal{S}_1$-$\mathcal{S}_2$-meßbar.

Meßbare Abbildungen spielen in der Wahrscheinlichkeitstheorie bei der Definition von Zufallsvariablen eine wichtige Rolle. Der folgende Satz zeigt, daß für den Nachweis der Meßbarkeit einer Abbildung nicht immer das Urbild $T^{-1}(A')$ für alle Mengen $A' \in \mathcal{S}_2$ untersucht werden muß.

Satz A.9 (Meßbarkeit bei einer erzeugten σ-Algebra)
Seien $(\Omega_1, \mathcal{S}_1)$ und $(\Omega_2, \mathcal{S}_2)$ zwei Meßräume, wobei $\mathcal{S}_2 = \sigma(\mathcal{F})$ von einem Mengensystem $\mathcal{F}$ erzeugt ist. Die Abbildung $T :\ \Omega_1 \to \Omega_2$ ist genau dann $\mathcal{S}_1$-$\mathcal{S}_2$-meßbar, falls $T^{-1}(A') \in \mathcal{S}_1$ für alle $A' \in \mathcal{F}$.

Sind drei Meßräume $(\Omega_1, \mathcal{S}_1)$, $(\Omega_2, \mathcal{S}_2)$, $(\Omega_3, \mathcal{S}_3)$ und zwei Abbildungen
$T_1 :\ \Omega_1 \to \Omega_2$, T_1 $\mathcal{S}_1$-$\mathcal{S}_2$-meßbar,
$T_2 :\ \Omega_2 \to \Omega_3$, T_2 $\mathcal{S}_2$-$\mathcal{S}_3$-meßbar,
gegeben, so ist die Abbildung
$T_2 \circ T_1 :\ \Omega_1 \to \Omega_3$, $\omega \mapsto T_2(T_1(\omega))$, $\mathcal{S}_1$-$\mathcal{S}_3$-meßbar.

Satz A.10 (Bildmaß)
Seien $(\Omega_1, \mathcal{S}_1, \mu_1)$ ein Maßraum, $(\Omega_2, \mathcal{S}_2)$ ein Meßraum und $T :\ \Omega_1 \to \Omega_2$ $\mathcal{S}_1$-$\mathcal{S}_2$-meßbar, so ist durch

$$\mu_2(A') := \mu_1\left(T^{-1}(A')\right), \quad A' \in \mathcal{S}_2,$$

ein Maß μ_2 auf $\mathcal{S}_2$ definiert.
Das Maß μ_2 wird als Bildmaß von μ_1 bezeichnet mit der Schreibweise $\mu_2 = T(\mu_1)$.

Um Zufallsgrößen analysieren zu können, benötigt man einen Integralbegriff. Daher soll im folgenden kurz die Integrationstheorie für meßbare Abbildungen zusammengefaßt werden. Zunächst betrachten wir die Integration einer speziellen Klasse von Funktionen.

Definition A.11 (elementare Funktion)
Sei $(\Omega,\mathcal{S})$ ein Meßraum. Eine $\mathcal{S}$-$\mathcal{B}$-meßbare Funktion $e:\ \Omega\to\mathbb{R}$ heißt elementare Funktion, falls sie nur endlich viele verschiedene Funktionswerte annimmt.

Eine spezielle elementare Funktion ist die Indikatorfunktion

$$I_A:\Omega\to\mathbb{R},\quad \omega\mapsto\begin{cases}1, & \text{falls } \omega\in A\\ 0, & \text{sonst}\end{cases},$$

die anzeigt, ob ω Element einer Menge $A\in\mathcal{S}$ ist. Mit Hilfe von Indikatorfunktionen lassen sich die elementaren Funktionen darstellen.

Satz A.12 (Darstellung elementarer Funktionen)
Sei $(\Omega,\mathcal{S})$ ein Meßraum. Ist $e:\Omega\to\mathbb{R}$ eine elementare Funktion, so existieren eine natürliche Zahl n, paarweise disjunkte Mengen $A_1,\ldots,A_n\in\mathcal{S}$ und reelle Zahlen $\alpha_1,\ldots,\alpha_n$ mit:

$$e=\sum_{i=1}^{n}\alpha_i I_{A_i},\quad \sum_{i=1}^{n}A_i=\Omega.$$

Die eben betrachtete Darstellung von e heißt eine Normaldarstellung von e. Sind alle α_i paarweise verschieden, so spricht man von einer kürzesten Normaldarstellung von e. Kürzeste Normaldarstellungen sind eindeutig. Aus der Normaldarstellung elementarer Funktionen folgt sofort: Summe, Differenz und Produkt elementarer Funktionen sind elementare Funktionen. Für alle $c\in\mathbb{R}$ ist auch $c\cdot e$ eine elementare Funktion, wenn e eine elementare Funktion ist.
Nun betrachten wir nichtnegative elementare Funktionen auf einem Maßraum $(\Omega,\mathcal{S},\mu)$ und definieren das (μ-)Integral dieser Funktionen.

Definition A.13 ($(\mu$-$)$Integral nichtnegativer elementarer Funktionen)
Sei $(\Omega,\mathcal{S},\mu)$ ein Maßraum und $e:\Omega\to\mathbb{R}_0^+$, $e=\sum_{i=1}^{n}\alpha_i I_{A_i}$, $\alpha_i\geq 0$, $i=1,\ldots,n$, eine nichtnegative elementare Funktion in Normaldarstellung, so wird

$$\int e\,d\mu:=\int_{\Omega}e\,d\mu:=\sum_{i=1}^{n}\alpha_i\cdot\mu(A_i)$$

als (μ-)Integral von e über Ω bezeichnet.

Damit $\int e\,d\mu$ wohldefiniert ist, ist natürlich zu zeigen, daß $\int e\,d\mu$ unabhängig von der Wahl der Normaldarstellung für e ist.

Sei nun E die Menge aller nichtnegativen elementaren Funktionen auf $(\Omega,\mathcal{S},\mu)$, so erhalten wir eine Abbildung

$$\mathrm{Int}:\ E\to\bar{\mathbb{R}}_0^+,\ e\mapsto\int e\,d\mu.$$

Die folgenden Eigenschaften von Int lassen sich leicht nachweisen:

- $\int I_A\,d\mu=\mu(A)$ für alle $A\in\mathcal{S}$.

- $\int(\alpha e)d\mu = \alpha\int e\,d\mu$ für alle $e \in E$, $\alpha \in \mathbb{R}_0^+$.
- $\int(u+v)d\mu = \int u\,d\mu + \int v\,d\mu$ für alle $u, v \in E$.
- Ist $u(\omega) \leq v(\omega)$ für alle $\omega \in \Omega$, so ist $\int u\,d\mu \leq \int v\,d\mu$ für alle $u, v \in E$.

Wählen wir $\Omega = \mathbb{R}^n$, $\mathcal{S} = \mathcal{B}^n$, $\mu = \lambda^n$ und $f : \Omega \to \mathbb{R}_0^+$, $x \mapsto 0$, so erhalten wir

$$\int f\,d\lambda^n = \int 0\,d\lambda^n = 0 \cdot \lambda^n(\mathbb{R}^n) = 0 \cdot \infty = 0.$$

Unsere Vereinbarung $0 \cdot \infty = 0$ erlaubt uns somit, das (λ^n-)Integral über die Nullfunktion zu berechnen.

Betrachtet man die Menge $\bar{\mathbb{R}}$ der um $\{\pm\infty\}$ erweiterten reellen Zahlen, so bildet die Menge

$$\bar{\mathcal{B}} := \left\{A \in \mathcal{P}(\bar{\mathbb{R}});\, A \cap \mathbb{R} \in \mathcal{B}\right\}$$

eine σ-Algebra über $\bar{\mathbb{R}}$. Um nun den Integralbegriff auf eine größere Klasse von Funktionen fortzusetzen, benötigen wir die folgende Definition.

Definition A.14 (numerische Funktion)
Eine auf einer nichtleeren Menge $A \subseteq \Omega$ definierte Funktion $f : A \to \bar{\mathbb{R}}$ heißt numerische Funktion.

Nun betrachten wir nichtnegative numerische Funktionen, die als Grenzwert einer Folge elementarer Funktionen gegeben sind.

Satz A.15 (Grenzwerte spezieller Folgen elementarer Funktionen)
Seien $(\Omega, \mathcal{S})$ ein Meßraum und $f : \Omega \to \bar{\mathbb{R}}_0^+$ eine nichtnegative, $\mathcal{S}$-$\bar{\mathcal{B}}$-meßbare numerische Funktion, so gibt es eine monoton steigende Folge $\{e_n\}_{n\in\mathbb{N}}$ von nichtnegativen elementaren Funktionen $e_n : \Omega \to \mathbb{R}_0^+$, $n \in \mathbb{N}$, die punktweise gegen f konvergiert. Wir schreiben dafür: $e_n \uparrow f$.

Nach diesen Vorbereitungen sind wir in der Lage, die (μ-)Integration auf eine spezielle Klasse von Funktionen in naheliegender Weise fortzusetzen.

Definition A.16 ((μ-)Integral für meßbare, nichtnegative numerische Funktionen)
Seien $(\Omega, \mathcal{S}, \mu)$ ein Maßraum und $f : \Omega \to \bar{\mathbb{R}}_0^+$ eine $\mathcal{S}$-$\bar{\mathcal{B}}$-meßbare, nichtnegative numerische Funktion. Sei ferner $\{e_n\}_{n\in\mathbb{N}}$ eine monoton steigende Folge nichtnegativer elementarer Funktionen $e_n : \Omega \to \mathbb{R}_0^+$, $n \in \mathbb{N}$, mit $e_n \uparrow f$, so definieren wir durch

$$\int f\,d\mu := \int_\Omega f\,d\mu := \lim_{n\to\infty} \int e_n\,d\mu$$

das (μ-)Integral von f über Ω.

Da die approximierende Folge elementarer Funktionen für f nicht eindeutig ist, muß natürlich erwähnt werden, daß das eben definierte Integral wohldefiniert ist. Wir werden nun in einem letzten Schritt die Klasse der integrierbaren Funktionen erweitern. Dazu dient die folgende Definition.

Definition A.17 (Positivteil, Negativteil einer numerischen Funktion)
Seien $(\Omega,\mathcal{S})$ ein Meßraum und $f:\Omega\to\overline{\mathbb{R}}$ eine $\mathcal{S}$-$\overline{\mathcal{B}}$-meßbare numerische Funktion, so wird die Funktion

$$f^+:\Omega\to\overline{\mathbb{R}}_0^+,\omega\mapsto\begin{cases} f(\omega), & \text{falls } f(\omega)\geq 0\\ 0, & \text{sonst}\end{cases}$$

Positivteil von f und die Funktion

$$f^-:\Omega\to\overline{\mathbb{R}}_0^+,\omega\mapsto\begin{cases} -f(\omega), & \text{falls } f(\omega)\leq 0\\ 0, & \text{sonst}\end{cases}$$

Negativteil von f genannt.

Die folgenden Eigenschaften von f^+ und f^- sind unmittelbar einzusehen:

- $f^+(\omega)\geq 0$, $f^-(\omega)\geq 0$ für alle $\omega\in\Omega$.
- f^+ und f^- sind $\mathcal{S}$-$\overline{\mathcal{B}}$-meßbare numerische Funktionen.
- $f=f^+-f^-$.

Mit Hilfe des Positiv- und Negativteils einer meßbaren numerischen Funktion $f:\Omega\to\overline{\mathbb{R}}$ können wir das (μ-)Integral auf meßbare numerische Funktionen erweitern.

Definition A.18 ((μ-)integrierbar, (μ-)quasiintegrierbar, (μ-)Integral)
Seien $(\Omega,\mathcal{S},\mu)$ ein Maßraum und $f:\Omega\to\overline{\mathbb{R}}$ eine $\mathcal{S}$-$\overline{\mathcal{B}}$-meßbare numerische Funktion.
f heißt (μ-)integrierbar, falls $\int f^+\,d\mu<\infty$ und $\int f^-\,d\mu<\infty$.
f heißt (μ-)quasiintegrierbar, falls $\int f^+\,d\mu<\infty$ oder $\int f^-\,d\mu<\infty$.
Ist f (μ-)quasiintegrierbar, so ist durch

$$\int f\,d\mu:=\int_\Omega f\,d\mu:=\int f^+\,d\mu-\int f^-\,d\mu$$

das (μ-)Integral von f über Ω definiert.

Als (μ-)Integral über einer Menge $A\in\mathcal{S}$ definieren wir für (μ-)quasiintegrierbares $f\cdot I_A$:

$$\int_A f\,d\mu:=\int f\cdot I_A\,d\mu.$$

Betrachtet man speziell den Maßraum $(\mathbb{R}^n,\mathcal{B}^n,\lambda^n)$, so wird das ($\lambda^n$-)Integral als Lebesgue-Integral bezeichnet. Ist f (λ^n-)integrierbar, so heißt f Lebesgue-integrierbar. Ist ein Maß μ^F durch eine maßerzeugende Funktion $F:\mathbb{R}^n\to\mathbb{R}$ gegeben, so wird das (μ^F-)Integral als Lebesgue-Stieltjes-Integral bezeichnet und in der Form

$$\int f\,dF:=\int f\,d\mu^F$$

geschrieben. Lebesgue-Stieltjes-Integrale besitzen die wichtige Eigenschaft, daß sie häufig durch Riemann-Integrale berechnet werden können.

In der Wahrscheinlichkeitstheorie werden Methoden zur Beschreibung und Analyse von Zufallsexperimenten (Experimente mit nicht vorhersehbarem Ausgang) bereitgestellt (für Details sei auf [Bau02, Bau92, SS95] verwiesen). Der umgangssprachliche Begriff „Zufallsexperiment“ wird durch einen Maßraum $(\Omega, \mathcal{S}, P)$ mit der Eigenschaft $P(\Omega) = 1$ mathematisch präzisiert. Wir definieren daher:

Definition A.19 (Wahrscheinlichkeitsraum, Wahrscheinlichkeitsmaß, Ergebnis, Ereignis)
Ein Maßraum $(\Omega, \mathcal{S}, P)$ mit $P(\Omega) = 1$ wird als Wahrscheinlichkeitsraum bezeichnet. Die Punkte $\omega \in \Omega$ heißen Ergebnisse, die Mengen $A \in \mathcal{S}$ Ereignisse. Das Maß P wird als Wahrscheinlichkeitsmaß bezeichnet. Für alle Ereignisse A wird $P(A)$ die Wahrscheinlichkeit von A genannt.

Wir werden im folgenden davon ausgehen, daß ein Zufallsexperiment durch einen Wahrscheinlichkeitsraum $(\Omega, \mathcal{S}, P)$ gegeben ist. Es ist in der Praxis oft nicht leicht, ein verbal formuliertes Zufallsexperiment durch einen Wahrscheinlichkeitsraum zu modellieren - insbesondere dann, wenn das Experiment ungenau formuliert ist. Die Elemente der Menge Ω stellen die möglichen Ergebnisse des Zufallsexperimentes dar.

Beispiel „Scheibenschießen“:
Wir betrachten das Schießen mit einem Gewehr auf eine kreisförmige Schießscheibe mit dem Radius $r = \frac{1}{\sqrt{\pi}}$ und dem Mittelpunkt $m = (0,0)^\top$. Wir nehmen an, daß bei jedem Schuß die Scheibe getroffen wird. Als Ergebnis eines Schußes erhalten wir einen Punkt

$$\omega = (\omega^1, \omega^2)^\top \in \Omega := K_{\frac{1}{\sqrt{\pi}},0} := \{x \in \mathbb{R}^2;\ \|x\|_2 \leq \frac{1}{\sqrt{\pi}}\}.$$

Wir wählen $\mathcal{S} := \{A \cap K_{\frac{1}{\sqrt{\pi}},0};\ A \in \mathcal{B}^2\}$ als σ-Algebra und $P = \lambda^2|_{\mathcal{S}}$ als Wahrscheinlichkeitsmaß auf $\mathcal{S}$. Da der Schütze bei jedem Schuß umso mehr Punkte (Ringe) erhält, je kleiner der Abstand seines Schußes zum Mittelpunkt der Schießscheibe ist, interessiert als Ergebnis in erster Linie dieser Abstand zum Mittelpunkt.
Man betrachtet also eine Funktion

$$d : \Omega \to [0, \frac{1}{\sqrt{\pi}}] =: \Omega',\ \omega \mapsto \|\omega\|_2.$$

Kann man nun mit Hilfe der Funktion d und des Wahrscheinlichkeitsraumes $(\Omega, \mathcal{S}, P)$ jeder Menge $A \in \mathcal{S}' := \{B \cap [0, \frac{1}{\sqrt{\pi}}];\ B \in \mathcal{B}\}$ eine Wahrscheinlichkeit zuordnen? Dies ist genau dann möglich, wenn d $\mathcal{S}$-$\mathcal{S}'$-meßbar ist. Daher definieren wir:

Definition A.20 ((n-dimensionale reelle, numerische) Zufallsvariable)
Seien $(\Omega, \mathcal{S}, P)$ ein Wahrscheinlichkeitsraum und $(\Omega', \mathcal{S}')$ ein Meßraum, dann heißt eine $\mathcal{S}$-$\mathcal{S}'$-meßbare Funktion $X : \Omega \to \Omega'$ Zufallsvariable.
Ist $\Omega' = \mathbb{R}^n$, $n \in \mathbb{N}$, und $\mathcal{S}' = \mathcal{B}^n$, so wird X als n-dimensionale reelle Zufallsvariable bezeichnet. Ist $\Omega' = \overline{\mathbb{R}}$ und $\mathcal{S}' = \overline{\mathcal{B}}$, so wird X als numerische Zufallsvariable bezeichnet. Eine eindimensionale reelle Zufallsvariable wird reelle Zufallsvariable genannt.

Als geeignetes Wahrscheinlichkeitsmaß P' auf $\mathcal{S}'$ ergibt sich das Bildmaß von X. Somit erhalten wir für unser obiges Beispiel $P'(A') = P(d^{-1}(A'))$ für alle $A' \in \mathcal{S}'$. Die Tatsache, daß $\lambda^2(\{\omega\}) = 0$ für alle $\omega \in K_{\frac{1}{\sqrt{\pi}},0}$ verdeutlicht den Sinn der Verwendung von Ereignissen $A \in \mathcal{S}$.

Definition A.21 (Verteilung einer Zufallsvariablen, Bildmaß)
Seien $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum, $(\Omega',\mathcal{S}')$ ein Meßraum und $X:\Omega\to\Omega'$ eine Zufallsvariable, dann wird das Bildmaß P_X von X Verteilung von X genannt.

Nach unserer Interpretation von Wahrscheinlichkeitsräumen ist der Wert $X(\omega)$ einer Zufallsvariablen an der Stelle ω vom Ergebnis eines Zufallsexperimentes abhängig. Wir fragen danach, welcher Wert von X „zu erwarten“ ist.

Definition A.22 (Erwartungswert einer numerischen Zufallsvariablen)
Seien $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum und X eine $(P$-$)$quasiintegrierbare numerische Zufallsvariable $X:\Omega\to\overline{\mathbb{R}}$, dann wird durch

$$\mathbf{E}(X) := \int X\, dP$$

der Erwartungswert von X definiert.

Den Erwartungswert einer n-dimensionalen reellen Zufallsvariablen definiert man durch komponentenweise Bildung des Erwartungswertes.

Um eine Vorstellung vom Begriff des Erwartungswertes zu bekommen, betrachten wir die folgende reelle Zufallsvariable auf $(\Omega,\mathcal{S},P)$: Seien $A_1,\ldots,A_n$ paarweise disjunkte Mengen aus $\mathcal{S}$ mit $\sum_{i=1}^{n} A_i = \Omega$ und $\alpha_1,\ldots,\alpha_n$ nichtnegative reelle Zahlen, dann ist

$$X:\Omega\to\mathbb{R},\ \omega\mapsto\sum_{i=1}^{n}\alpha_i I_{A_i}(\omega)$$

eine reelle Zufallsvariable. Für den Erwartungswert von X erhalten wir

$$\mathbf{E}(X)=\sum_{i=1}^{n}\alpha_i P(A_i).$$

Der Erwartungswert ist in diesem Fall also eine gewichtete Summe der möglichen Werte von X, wobei die Gewichte gerade die Wahrscheinlichkeiten für das Auftreten dieser Werte sind. Gilt $P(A_i)=\frac{1}{n}$ für alle $i=1,\ldots,n$, so erhalten wir als Erwartungswert das arithmetische Mittel der Werte von X.

Ist eine reelle Zufallsvariable $(P$-$)$integrierbar, so läßt sich der Erwartungswert von X auch mit Hilfe des Bildmaßes P_X berechnen:

$$\mathbf{E}(X)=\int x\,dP_X(x) := \int f\,dP_X \quad \text{mit } f:\mathbb{R}\to\mathbb{R},\ x\mapsto x.$$

Da wir auch an Erwartungswerten von speziellen Funktionen von X interessiert sind, benötigen wir den folgenden Satz.

Satz A.23 (Meßbarkeit stetiger Funktionen reeller Zufallsvariablen)
Seien X eine reelle Zufallsvariable definiert auf dem Wahrscheinlichkeitsraum $(\Omega,\mathcal{S},P)$ und $g:\mathbb{R}\to\mathbb{R}$ eine stetige Funktion, dann ist $g\circ X:\Omega\to\mathbb{R},\ \omega\to g(X(\omega))$ eine reelle Zufallsvariable auf $(\Omega,\mathcal{S},P)$.

Somit folgt sofort, daß für eine reelle Zufallsvariable X auf $(\Omega,\mathcal{S},P)$ und für jedes $k\in\mathbb{N}$ und jedes $\alpha\in\mathbb{R}$ auch $(X-\alpha)^k$ und $|X-\alpha|^k$ reelle Zufallsvariablen auf $(\Omega,\mathcal{S},P)$ sind. Dies ermöglicht die folgende Definition.

Definition A.24 (zentrierte (absolute) Momente k-ter Ordnung)
Sei X eine auf dem Wahrscheinlichkeitsraum $(\Omega,\mathcal{S},P)$ definierte reelle Zufallsvariable, dann heißt $\mathbf{E}\left(|X-\alpha|^k\right)$, $k\in\mathbb{N}$, das in α zentrierte absolute Moment k-ter Ordnung von X. Ist $(X-\alpha)^k$ (P-)quasiintegrierbar, so heißt $\mathbf{E}\left((X-\alpha)^k\right)$ das in α zentrierte Moment k-ter Ordnung. Ist $\alpha=0$, so spricht man nur von absoluten Momenten bzw. Momenten k-ter Ordnung.

Besonders interessant ist der Fall $k=2$.

Definition A.25 (Varianz einer reellen Zufallsvariablen)
Sei X eine auf dem Wahrscheinlichkeitsraum $(\Omega,\mathcal{S},P)$ definierte, (P-)integrierbare reelle Zufallsvariable, dann heißt

$$\mathbf{Var}(X) := \int (X-\mathbf{E}(X))^2\,dP$$

die Varianz von X.
Die Zahl $\sigma=\sqrt{\mathbf{Var}(X)}$ wird als Streuung oder Standardabweichung von X bezeichnet.

Oft schreibt man σ^2 für $\mathbf{Var}(X)$. Die Varianz ist ein Maß für die zu erwartende Abweichung von X und $\mathbf{E}(X)$.

Lemma A.26 (Standardisierung)
Sei X eine auf dem Wahrscheinlichkeitsraum $(\Omega,\mathcal{S},P)$ definierte, (P-)integrierbare reelle Zufallsvariable mit Streuung $0<\sigma<\infty$. Dann ist

$$Y := \frac{X-\mathbf{E}(X)}{\sigma}$$

eine Zufallsvariable mit Erwartungswert $\mathbf{E}(Y)=0$ und Varianz $\mathbf{Var}(Y)=1$.

Den Übergang von X zu Y bezeichnet man als „Standardisierung“ von X.

Im folgenden betrachten wir einige wichtige Begriffe der elementaren Wahrscheinlichkeitstheorie. Ausgangspunkt ist der Wahrscheinlichkeitsraum $(\Omega,\mathcal{S},P)$ und zwei Mengen $A,B\in\mathcal{S}$ mit $P(B)>0$. Auf $\mathcal{S}$ definieren wir nun ein Wahrscheinlichkeitsmaß $P^B:\mathcal{S}\to[0,1]$ durch $A\mapsto\frac{P(A\cap B)}{P(B)}$. Durch den Übergang von P zu P^B erhält die Menge B das Wahrscheinlichkeitsmaß 1. Wir interpretieren $P^B(A)$ als die Wahrscheinlichkeit von A unter der Bedingung, daß das Ereignis B (P^B-)fast sicher eintrifft. Dies führt zur Definition der bedingten Wahrscheinlichkeit.

Definition A.27 (bedingte Wahrscheinlichkeit)
Sei $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum und $A,B\in\mathcal{S}$ mit $P(B)>0$. Dann heißt

$$P(A|B) := \frac{P(A\cap B)}{P(B)}$$

die (bedingte) Wahrscheinlichkeit von A unter der Bedingung B.

Satz A.28 (Formel von der totalen Wahrscheinlichkeit, Satz von Bayes)
Sei $(\Omega, \mathcal{S}, P)$ ein Wahrscheinlichkeitsraum und $\{D_i \subset \Omega; i \in \mathbb{N}\}$ eine Partition[1] von Ω, so daß $D_i \in \mathcal{S}$ und $P(D_i) > 0$ für alle $i \in \mathbb{N}$.

(i) *Es gilt die „Formel von der totalen Wahrscheinlichkeit“:*

$$P(A) = \sum_{i=1}^{\infty} P(D_i) \cdot P(A|D_i), \quad \text{für alle } A \in \mathcal{S}.$$

(ii) *Ist $A \in \mathcal{S}$ mit $P(A) > 0$, so gilt*

$$P(D_i|A) = \frac{P(A|D_i) \cdot P(D_i)}{P(A)}, \quad \text{für alle } i \in \mathbb{N}.$$

(iii) *Es gilt der „Satz von Bayes“:*

$$P(D_i|A) = \frac{P(A|D_i) \cdot P(D_i)}{\sum_{j=1}^{\infty} P(D_j) \cdot P(A|D_j)}, \quad \text{für alle } i \in \mathbb{N}.$$

⌟

Analoge Formeln ergeben sich natürlich für eine endliche Partition $\{D_i \subset \Omega; i = 1, \ldots, n\}$ von Ω.

Es soll nun die Frage untersucht werden, unter welchen Voraussetzungen ein Wahrscheinlichkeitsmaß P in der folgenden Art und Weise durch ein Maß μ dargestellt werden kann.

Definition A.29 (Dichte)
Sei $(\Omega, \mathcal{S}, P)$ ein Wahrscheinlichkeitsraum und μ ein Maß auf $\mathcal{S}$. Wenn eine nichtnegative, $\mathcal{S}$-$\bar{\mathcal{B}}$-meßbare numerische Funktion $f : \Omega \to \bar{\mathbb{R}}$ existiert mit

$$P(A) = \int_A f \, d\mu, \quad \text{für alle } A \in \mathcal{S},$$

so heißt f eine Dichte(funktion) des Wahrscheinlichkeitsmaßes P bezüglich μ. Man sagt auch, daß P bezüglich μ eine Dichte f besitzt.

⌟

Satz A.30 (Beziehung zwischen $(P$-$)$ und $(\mu$-$)$Nullmengen)
Seien $(\Omega, \mathcal{S}, P)$ ein Wahrscheinlichkeitsraum, μ ein Maß auf $\mathcal{S}$ und f eine Dichte von P bezüglich μ, dann gilt für alle $A \in \mathcal{S}$ mit $\mu(A) = 0$: $P(A) = 0$.

⌟

Die folgende Definition resultiert aus dem eben betrachteten Satz.

Definition A.31 (absolute Stetigkeit von P bezüglich μ)
Seien $(\Omega, \mathcal{S}, P)$ ein Wahrscheinlichkeitsraum und μ ein Maß auf $\mathcal{S}$. P heißt absolutstetig bezüglich μ, falls für alle $A \in \mathcal{S}$ mit $\mu(A) = 0$ gilt: $P(A) = 0$.

⌟

[1] $\{D_i \subset \Omega; i \in \mathbb{N}\}$ heißt eine Partition von Ω, falls die Mengen D_i paarweise disjunkt sind und $\Omega = \bigcup_{i=1}^{\infty} D_i$.

Wie der folgende Satz zeigt, ist die absolute Stetigkeit bezüglich eines σ-endlichen Maßes μ das entscheidende Kriterium für die Existenz einer Dichte.

Satz A.32 (Radon-Nikodym)
Seien $(\Omega, \mathcal{S}, P)$ ein Wahrscheinlichkeitsraum und μ ein σ-endliches Maß auf $\mathcal{S}$, dann besitzt P genau dann eine Dichte bezüglich μ, wenn P absolutstetig bezüglich μ ist.

Nun betrachten wir eine spezielle Klasse von Wahrscheinlichkeitsmaßen. Mit $|A|$ wird die Anzahl der Elemente (Mächtigkeit) von A bezeichnet.

Definition A.33 (diskretes Wahrscheinlichkeitsmaß, diskrete Zufallsvariable)
Sei $(\mathbb{R}^n, \mathcal{B}^n, P)$, $n \in \mathbb{N}$, ein Wahrscheinlichkeitsraum. Das Wahrscheinlichkeitsmaß P heißt diskret, falls eine Menge $B \in \mathcal{B}^n$ mit $|B| \leq |\mathbb{N}|$ und $P(B) = 1$ existiert. Eine m-dimensionale reelle Zufallsvariable X definiert auf $(\mathbb{R}^n, \mathcal{B}^n, P)$, $m \in \mathbb{N}$, heißt diskret, falls das Bildmaß P_X von X ein diskretes Wahrscheinlichkeitsmaß auf $(\mathbb{R}^m, \mathcal{B}^m)$ ist.

Da für $m = n$ das Bildmaß P_X der Zufallsvariablen $X : \mathbb{R}^n \to \mathbb{R}^n$, $x \mapsto x$, gleich P ist, wird oft der Begriff Verteilung statt Wahrscheinlichkeitsmaß verwendet. Um nun mit Hilfe des Satzes von Radon-Nikodym diskrete Verteilungen (Wahrscheinlichkeitsmaße) durch Dichtefunktionen darstellen zu können, benötigen wir ein spezielles Maß.

Definition A.34 (Zählmaß)
Das auf einer σ-Algebra $\mathcal{S}$ über Ω definierte Maß

$$\zeta : \mathcal{S} \to \overline{\mathbb{R}}, A \mapsto \begin{cases} |A|, & \text{falls } |A| \text{ endlich ist} \\ \infty, & \text{sonst} \end{cases}$$

wird als das Zählmaß auf $\mathcal{S}$ bezeichnet.

Sei nun $(\mathbb{R}^n, \mathcal{B}^n, P)$ ein Wahrscheinlichkeitsraum und P eine diskrete Verteilung auf $\mathcal{B}^n$ mit $P(B) = 1$ für ein $B \in \mathcal{B}^n$ und $|B| \leq |\mathbb{N}|$, dann gilt für alle $C \in \mathcal{B}^n$:

$$P(C) = P(C \cap B) + P(C \cap B^c) = P(C \cap B).$$

Somit genügt es, den Wahrscheinlichkeitsraum $(B, \mathcal{B}_B^n, P)$ mit $\mathcal{B}_B^n := \{C \cap B;\ C \in \mathcal{B}^n\} = \mathcal{P}(B)$ zu betrachten. Da ζ ein σ-endliches Maß auf $\mathcal{P}(B)$ ist und $\zeta(A) = 0$ genau dann gilt, wenn $A = \emptyset$, ist jedes Wahrscheinlichkeitsmaß auf $\mathcal{P}(B)$ absolutstetig bezüglich ζ. Somit existiert zu jedem Wahrscheinlichkeitsmaß P auf $\mathcal{P}(B)$ eine Dichte $f : B \to \mathbb{R}_0^+$ mit

$$P(A) = \int_A f \, d\zeta = \sum_{\omega \in A} f(\omega) = \sum_{\omega \in A} P(\{\omega\}) \text{ für alle } A \in \mathcal{P}(B).$$

Es läßt sich also jede diskrete Verteilung auf $\mathcal{B}^n$ durch eine Folge $\{p_j\}_{j \in \mathbb{N}_0}$ nichtnegativer reeller Zahlen mit $\sum_{j=0}^{\infty} p_j = 1$ darstellen.

Definition A.35 (spezielle diskrete Verteilungen)
Sei $(\mathbb{R},\mathcal{B},P)$ ein Wahrscheinlichkeitsraum mit einem diskreten Wahrscheinlichkeitsmaß P und $P(\mathbb{N}_0)=1$.

(i) ***Poisson-Verteilung:***
Ist

$$P(\{j\}) = p_j = e^{-\lambda}\frac{\lambda^j}{j!}, \quad j \in \mathbb{N}_0, \lambda > 0,$$

so spricht man von einer Poisson[2]-Verteilung mit Parameter λ.

(ii) ***Gleichverteilung und Laplace-Experiment:***
Ist

$$P(\{j\}) = p_j = \frac{1}{k+1}, \quad \text{für } j = 0,\ldots,k, \text{ und } p_j = 0 \text{ für } j > k, k \in \mathbb{N}_0,$$

so wird diese Verteilung Gleichverteilung genannt. Ein Zufallsexperiment, das durch einen Wahrscheinlichkeitsraum mit Gleichverteilung repräsentiert wird, heißt Laplace[3]-Experiment.

(iii) ***Binomial-Verteilung und Bernoulli-Experiment:***
Wählt man $p \in \mathbb{R}$, $0 < p < 1$, und $B = \{0,1,2,\ldots,s\}$, $s \in \mathbb{N}$, so wird (mit $\binom{s}{j} := \frac{s!}{(s-j)!\,j!}$) die durch

$$P(\{j\}) = p_j = \binom{s}{j} p^j (1-p)^{s-j} \quad \text{für } j = 0,\ldots,s, \text{ und } p_j = 0 \text{ für } j > s,$$

gegebene Verteilung Binomial-Verteilung $B(s,p)$ mit Parameter s,p genannt. Ein Zufallsexperiment, das durch einen Wahrscheinlichkeitsraum mit Binomial-Verteilung mit Parameter s,p repräsentiert wird, heißt Bernoulli[4]-Experiment mit Parameter s,p.

(iv) *Eine auf einem Wahrscheinlichkeitsraum $(\Omega,\mathcal{S},\hat{P})$ definierte reelle Zufallsvariable X heißt poissonverteilt / gleichverteilt / binomialverteilt, wenn das Bildmaß $P = \hat{P}_X$ poissonverteilt / gleichverteilt / binomialverteilt ist.*

⌟

Ein Bernoulli-Experiment kann folgendermaßen interpretiert werden: Man betrachtet ein Zufallsexperiment, bei dem es nur zwei mögliche Ergebnisse gibt, nämlich mit Wahrscheinlichkeit p das Ergebnis 'T' (Treffer) und mit Wahrscheinlichkeit $(1-p)$ das Ergebnis 'N' (Niete). Dieses Experiment führen wir s-mal durch, ohne daß sich die Ergebnisse gegenseitig beeinflussen. Die Wahrscheinlichkeit, daß nach diesen s Versuchen genau j Treffer auftreten, ist gegeben durch $\binom{s}{j} p^j (1-p)^{s-j}$, $0 \le j \le s$, $s \in \mathbb{N}$. Somit wird die s-malige Durchführung unseres Experimentes durch ein Bernoulli-Experiment beschrieben, falls die Ergebnisse sich nicht gegenseitig beeinflussen. Für sehr große s und sehr kleine p ist es möglich, eine Binomial-Verteilung durch die wesentlich einfacher zu berechnende Poisson-Verteilung mit Parameter $\lambda = s \cdot p$ zu approximieren.

Nun betrachten wir die folgende naheliegende Definition.

[2] nach D. Poisson (1781-1840)
[3] nach P. S. de Laplace (1749-1827)
[4] nach J. Bernoulli (1654-1705)

Definition A.36 (absolutstetige Zufallsvariable)
Sei $(\mathbb{R}^n, \mathcal{B}^n, P)$, $n \in \mathbb{N}$, ein Wahrscheinlichkeitsraum. Eine m-dimensionale reelle Zufallsvariable X definiert auf $(\mathbb{R}^n, \mathcal{B}^n, P)$, $m \in \mathbb{N}$, heißt absolutstetig, falls das Bildmaß P_X von X ein absolutstetiges Wahrscheinlichkeitsmaß auf $\mathcal{B}^m$ bezüglich λ^m ist.

Nach dem Satz von Radon-Nikodym ist P_X genau dann absolutstetig bezüglich λ^m, wenn P_X eine Dichte bezüglich λ^m besitzt. Mit Hilfe der beiden nächsten Definitionen ist es möglich, alle Wahrscheinlichkeitsmaße auf $\mathcal{B}$ zu klassifizieren.

Definition A.37 (Verteilungsfunktion)
Sei $(\mathbb{R}^n, \mathcal{B}^n, P)$, $n \in \mathbb{N}$, ein Wahrscheinlichkeitsraum. Die Funktion

$$F : \mathbb{R}^n \to [0,1], (x_1, \ldots, x_n)^\top \mapsto P(]-\infty, x_1[\times \ldots \times]-\infty, x_n[)$$

wird als Verteilungsfunktion von P bezeichnet. Die Verteilungsfunktion des Bildmaßes P_X einer m-dimensionalen reellen Zufallsvariable $X : \mathbb{R}^n \to \mathbb{R}^m$, $m \in \mathbb{N}$, wird auch Verteilungsfunktion von X genannt.

Definition A.38 (stetiges Wahrscheinlichkeitsmaß, stetige Zufallsvariable)
Sei $(\mathbb{R}^n, \mathcal{B}^n, P)$, $n \in \mathbb{N}$, ein Wahrscheinlichkeitsraum. Das Wahrscheinlichkeitsmaß (die Verteilung) P heißt stetig, falls die Verteilungsfunktion von P stetig ist. Eine m-dimensionale reelle Zufallsvariable X definiert auf $(\mathbb{R}^n, \mathcal{B}^n, P)$, $n \in \mathbb{N}$, $m \in \mathbb{N}$, heißt stetig, falls die Verteilungsfunktion von X stetig ist.

Die Verteilungsfunktion einer diskreten Verteilung ist eine Treppenfunktion und damit nicht stetig. Die Verteilungsfunktion einer bezüglich λ^n absolutstetigen Verteilung auf $\mathcal{B}^n$ ist stetig (für $n = 1$ sogar absolut stetig im topologischen Sinne). Die Umkehrung gilt aber nicht, da es stetige Wahrscheinlichkeitsmaße P gibt, die nicht absolutstetig bezüglich λ^n sind. Diese Wahrscheinlichkeitsmaße werden singulär genannt.

Definition A.39 (singuläres Wahrscheinlichkeitsmaß)
Sei $(\mathbb{R}^n, \mathcal{B}^n, P)$, $n \in \mathbb{N}$, ein Wahrscheinlichkeitsraum. Das Wahrscheinlichkeitsmaß (die Verteilung) P heißt singulär bezüglich λ^n, wenn eine Menge $N \in \mathcal{B}^n$ existiert mit $\lambda^n(N) = 0$ und $P(N) = 1$.

Nun sind wir in der Lage, die Verteilungsfunktion einer reellen Zufallsvariable in drei Komponenten zu zerlegen.

Satz A.40 (Zerlegungssatz von Lebesgue)
Sei X eine reelle Zufallsvariable mit Verteilungsfunktion F, die auf einem Wahrscheinlichkeitsraum $(\mathbb{R}, \mathcal{B}, P)$ definiert ist, dann gibt es nichtnegative reelle Zahlen a_1, a_2, a_3 mit $a_1 + a_2 + a_3 = 1$ und drei Funktionen $F_i : \mathbb{R} \to \mathbb{R}$, $i = 1, 2, 3$, mit:

- $F = a_1 F_1 + a_2 F_2 + a_3 F_3$.
- *F_1 ist Verteilungsfunktion einer diskreten Zufallsvariable auf $(\mathbb{R}, \mathcal{B}, P)$, F_2 ist Verteilungsfunktion einer bezüglich λ absolutstetigen reellen Zufallsvariable auf $(\mathbb{R}, \mathcal{B}, P)$ und F_3 ist Verteilungsfunktion einer stetigen reellen Zufallsvariable auf $(\mathbb{R}, \mathcal{B}, P)$, deren Bildmaß singulär bezüglich λ ist.*

Riemann-Integration
Ist P_1 ein bezüglich λ absolutstetiges Wahrscheinlichkeitsmaß auf $(\mathbb{R},\mathcal{B})$, so existiert eine Dichte f mit

$$P_1(A) = \int_A f\, d\lambda, A \in \mathcal{B}.$$

Die Funktion f ist in einem Intervall $[a,b]$, $a < b$, Riemann-integrierbar, falls sie auf diesem Intervall beschränkt ist und die Menge der Unstetigkeitsstellen von f auf $[a,b]$ das Lebesgue-Maß Null hat. Sind diese Voraussetzungen an f erfüllt, so können wir für jede (P_1-)integrierbare Funktion $g:\ [a,b] \to \mathbb{R}$ das (P_1-)Integral von g über dem Intervall $[a,b]$ durch ein Riemann-Integral berechnen, falls $g \cdot f$ Riemann-integrierbar über $[a,b]$ ist:

$$\int_{[a,b]} g\, dP_1 = \int_{[a,b]} g \cdot f\, d\lambda = \int_a^b g(x) \cdot f(x)\, dx.$$

Definition A.41 (Dichtefunktion)
Sei $d : \mathbb{R}^n \to \mathbb{R}$, $n \in \mathbb{N}$, eine stetige Funktion mit folgenden Eigenschaften:

- $d(x) \geq 0$ *für alle* $x \in \mathbb{R}^n$,
- $\int_{\mathbb{R}^n} d(x)\, dx = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} d(x)\, dx_1 \dots dx_n = 1$,

dann ist auch $\int d\, d\lambda^n = 1$ und wir können die Funktion d als Dichte eines Wahrscheinlichkeitsmaßes bezüglich λ^n auffassen.

Nun betrachten wir für jeden Vektor $\mu \in \mathbb{R}^n$ und für jede positiv definite Matrix $\Sigma \in \mathbb{R}^{n,n}$ die Funktion

$$\nu_{\mu,\Sigma} : \mathbb{R}^n \to \mathbb{R}, x \mapsto \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} \cdot \exp\left(-\frac{(x-\mu)^T \Sigma^{-1} (x-\mu)}{2}\right).$$

Offensichtlich ist $\nu_{\mu,\Sigma}(x) > 0$ für alle $\mu, x \in \mathbb{R}^n$, $\Sigma \in \mathbb{R}^{n,n}$, Σ positiv definit.
Aus der Analysis (Substitutionsregel, Satz von Fubini) ist das Folgende bekannt:

$$\int_{\mathbb{R}^n} \exp\left(-\frac{(x-\mu)^T \Sigma^{-1} (x-\mu)}{2}\right) dx = \sqrt{(2\pi)^n \det(\Sigma)}$$

für alle $\mu \in \mathbb{R}^n$, $\Sigma \in \mathbb{R}^{n,n}$, Σ positiv definit. Somit können wir $\nu_{\mu,\Sigma}$ als Dichte eines Wahrscheinlichkeitsmaßes bezüglich λ^n auffassen.

Definition A.42 (Normalverteilung)
Seien $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum, $\mu \in \mathbb{R}^n$, $n \in \mathbb{N}$, und $\Sigma \in \mathbb{R}^{n,n}$, Σ positiv definit. Die Zufallsvariable $X_{\mu,\Sigma} : \Omega \to \mathbb{R}^n$ heißt $\mathcal{N}(\mu,\Sigma)$ normalverteilt, falls ihr Bildmaß $P_{X_{\mu,\Sigma}}$ bezüglich λ^n die folgende Dichte besitzt:

$$\nu_{\mu,\Sigma} : \mathbb{R}^n \to \mathbb{R}, x \mapsto \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} \cdot \exp\left(-\frac{(x-\mu)^T \Sigma^{-1} (x-\mu)}{2}\right).$$

Um die Parameter μ und Σ einer Normalverteilung interpretieren zu können, benötigen wir die folgende Definition.

Definition A.43 (Covarianz, unkorreliert, Korrelationskoeffizient)
Seien $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum und $X:\Omega\to\mathbb{R}$, $Y:\Omega\to\mathbb{R}$ zwei reelle, (P-)integrierbare Zufallsvariable mit (P-)integrierbarem Produkt $X\cdot Y$.

(i) *Dann heißt*

$$\mathbf{Cov}(X,Y) := \mathbf{E}\left((X-\mathbf{E}(X))\cdot(Y-\mathbf{E}(Y))\right) = \mathbf{E}(X\cdot Y)-\mathbf{E}(X)\cdot\mathbf{E}(Y)$$

die Covarianz von X und Y. X und Y heißen unkorreliert, falls $\mathbf{Cov}(X,Y)=0$.

(ii) *Besitzen die Zufallsvariablen X bzw. Y zudem endliche Varianzen $\mathbf{Var}(X)>0$ bzw. $\mathbf{Var}(Y)>0$, so wird die Größe*

$$\rho(X,Y) := \frac{\mathbf{Cov}(X,Y)}{\sqrt{\mathbf{Var}(X)\cdot\mathbf{Var}(Y)}}$$

Korrelationskoffizient von X und Y genannt.

Normalverteilte Zufallsvariablen spielen in der Wahrscheinlichkeitstheorie eine bedeutende Rolle, auf die wir im Zusammenhang mit dem zentralen Grenzwertsatz[5] noch zu sprechen kommen. Zunächst fassen wir einige Eigenschaften einer $\mathcal{N}(\mu,\Sigma)$ normalverteilten Zufallsvariablen $X_{\mu,\Sigma}$ zusammen. Dazu fassen wir die Funktion $X_{\mu,\Sigma}:\ \Omega\to\mathbb{R}^n$ als Abbildung

$$\omega\mapsto\left(X^1_{\mu,\Sigma}(\omega),\ldots,X^n_{\mu,\Sigma}(\omega)\right)^\top$$

auf. Jede Funktion $X^i_{\mu,\Sigma}:\ \Omega\to\mathbb{R}$, $i=1,\ldots,n$, ist eine reelle Zufallsvariable. Definiert man

$$\mathbf{E}\left(X_{\mu,\Sigma}\right) := \left(\mathbf{E}\left(X^1_{\mu,\Sigma}\right),\ldots,\mathbf{E}\left(X^n_{\mu,\Sigma}\right)\right)^\top,$$

so erhält man

$$\mathbf{E}\left(X_{\mu,\Sigma}\right)=\mu.$$

Ferner gilt mit $\Sigma=(\sigma_{i,j})_{i,j=1,\ldots,n}$:

$$\mathbf{Cov}\left(X^i_{\mu,\Sigma},X^j_{\mu,\Sigma}\right)=\sigma_{i,j},\quad i,j=1,\ldots,n.$$

Daher heißt Σ die Covarianzmatrix von $X_{\mu,\Sigma}$.

Auf der Basis eines Wahrscheinlichkeitsraumes $(\Omega,\mathcal{S},P)$ haben wir für $A,B\in\mathcal{S}$ und $P(B)>0$ durch $P^B(A)=\frac{P(A\cap B)}{P(B)}$ ein Wahrscheinlichkeitsmaß auf $\mathcal{S}$ eingeführt. Wir interpretierten $P^B(A)$ als die Wahrscheinlichkeit von A unter der Bedingung, daß B (P^B-)fast sicher eintrifft. Nun stellt sich die Frage, wann diese Bedingung die Wahrscheinlichkeit für A nicht ändert, wann also $P^B(A)=P(A|B)=P(A)$ gilt. Wir erhalten:

$$P^B(A)=P(A|B)=P(A)\iff P(A\cap B)=P(A)\cdot P(B).$$

[5] siehe dazu Definition A.50 auf Seite 181 und Satz A.51 auf Seite 181

Definition A.44 (stochastisch unabhängige Ereignisse)
Seien $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum und $A_1,\ldots,A_n \in \mathcal{S}$, $n \in \mathbb{N}$, dann heißen die Ereignisse $A_1,\ldots,A_n$ stochastisch unabhängig, falls für alle $k \in \mathbb{N}$, $k \leq n$, und für alle $i_j \in \mathbb{N}$, $1 \leq j \leq k$, mit $1 \leq i_1 < \ldots < i_k \leq n$ gilt:

$$P\left(\bigcap_{j=1}^{k} A_{i_j}\right) = \prod_{j=1}^{k} P(A_{i_j}).$$

⌟

Die stochastische Unabhängigkeit einer Menge $\{A_i \in \mathcal{S};\, i \in I\}$, $I \neq \emptyset$, von Ereignissen führt man auf die stochastische Unabhängigkeit ihrer endlichen Teilmengen zurück.

Definition A.45 (stochastische Unabhängigkeit einer Menge von Ereignissen)
Seien $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum und $\{A_i \in \mathcal{S};\, i \in I\}$, $I \neq \emptyset$, eine Menge von Ereignissen, dann heißen diese Ereignisse stochastisch unabhängig, falls $A_{i_1},\ldots,A_{i_n}$ für jedes $n \in \mathbb{N}$ mit $n \leq |I|$ und für jede Menge $\{i_1,\ldots,i_n\} \subseteq I$ stochastisch unabhängig sind.

⌟

Um stochastisch unabhängige Zufallsvariable definieren zu können, wird zunächst die stochastische Unabhängigkeit von Mengensystemen betrachtet.

Definition A.46 (stochastische Unabhängigkeit von Mengensystemen)
Seien $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum und $\{\mathcal{F}_i \subseteq \mathcal{S};\, i \in I\}$, $I \neq \emptyset$, eine Menge von Mengensystemen über Ω, dann heißen diese Mengensysteme stochastisch unabhängig, falls für jedes $n \in \mathbb{N}$ mit $n \leq |I|$ und für jedes $\{i_1,\ldots,i_n\} \subseteq I$ die n Ereignisse $A_{i_1},\ldots,A_{i_n}$ für beliebige $A_{i_k} \in \mathcal{F}_{i_k}$, $i = 1,\ldots,n$, stochastisch unabhängig sind.

⌟

Definition A.47 (von einer Zufallsvariablen erzeugte σ-Algebra)
Sei $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum, $(\Omega',\mathcal{S}')$ ein Meßraum und $X : \Omega \to \Omega'$ eine Zufallsvariable. Weiter sei $\mathcal{F}$ die Menge aller σ-Algebren über Ω, für die gilt: X ist $\mathcal{C}$-$\mathcal{S}'$-meßbar genau dann, wenn $\mathcal{C} \in \mathcal{F}$. Die Menge $\sigma(X) := \sigma(\mathcal{F}) = \bigcap_{\mathcal{C}\in\mathcal{F}} \mathcal{C}$ ist ebenfalls eine σ-Algebra und wird die von X erzeugte σ-Algebra genannt.

⌟

Unter allen σ-Algebren $\mathcal{A}$ über Ω ist $\sigma(X)$ die kleinste, für die X $\mathcal{A}$-$\mathcal{S}'$-meßbar ist. Somit sind wir in der Lage, die stochastische Unabhängigkeit von Zufallsvariablen in naheliegender Weise durch die stochastische Unabhängigkeit von speziellen Mengensystemen zu definieren.

Definition A.48 (stochastische Unabhängigkeit von Zufallsvariablen)
Seien $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum, $(\Omega',\mathcal{S}')$ ein Meßraum und $\{X_i : \Omega \to \Omega';\, i \in I\}$, $I \neq \emptyset$, eine Menge von Zufallsvariablen, dann heißen diese Zufallsvariablen stochastisch unabhängig, falls die Mengensysteme $\{\sigma(X_i);\, i \in I\}$ stochastisch unabhängig sind.

⌟

Die stochastische Unabhängigkeit von Zufallsvariablen ist ein zentraler Begriff der Wahrscheinlichkeitstheorie und im wesentlichen Bestandteil der Modellierung zu untersuchender Vorgänge.

Da eine Folge von reellen Zufallsvariablen eine Funktionenfolge ist, betrachtet man - wie in der Analysis (z.B. gleichmäßige- und punktweise Konvergenz) - auch in der Wahrscheinlichkeitstheorie verschiedene Konvergenzbegriffe.

Definition A.49 (verschiedene Konvergenzbegriffe für Folgen reeller Zufallsvariable)
Seien $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum, $\{X_i\}_{i\in\mathbb{N}}$ eine Folge reeller Zufallsvariable $X_i:\Omega\to\mathbb{R}$, $i\in\mathbb{N}$, und $X:\Omega\to\mathbb{R}$ ebenfalls eine reelle Zufallsvariable, dann konvergiert $\{X_i\}_{i\in\mathbb{N}}$ definitionsgemäß

(i) *im r-ten Mittel $(r\in\mathbb{R}^+)$ gegen X genau dann, wenn*

$$\int |X_i|^r\,dP<\infty \text{ für alle } i\in\mathbb{N},\ \int |X|^r\,dP<\infty \text{ und } \lim_{i\to\infty}\int |X_i-X|^r\,dP=0,$$

(ii) *stochastisch gegen X genau dann, wenn für alle $\varepsilon>0$*

$$\lim_{i\to\infty} P\left(\{\omega\in\Omega;\,|X_i(\omega)-X(\omega)|<\varepsilon\}\right)=1,$$

(iii) *mit Wahrscheinlichkeit 1 gegen X genau dann, wenn*

$$P\left(\left\{\omega\in\Omega;\,\lim_{i\to\infty}X_i(\omega)=X(\omega)\right\}\right)=1,$$

(iv) *in Verteilung gegen X genau dann, wenn*

$$\lim_{i\to\infty}\int f\,dP_{X_i}=\int f\,dP_X$$

für alle beliebig oft differenzierbaren Funktionen $f:\mathbb{R}\to\mathbb{R}$ mit kompaktem Träger.

Die stochastische Konvergenz von $\{X_i\}_{i\in\mathbb{N}}$ gegen X wird oft durch

$$(P\text{-})\lim_{i\to\infty}X_i=X,\ st\text{-}\lim_{i\to\infty}X_i=X \text{ oder } X_i\to X \text{ nach Wahrscheinlichkeit}$$

dargestellt. Die Konvergenz mit Wahrscheinlichkeit 1 von $\{X_i\}_{i\in\mathbb{N}}$ gegen X heißt auch $(P$-$)$fast sichere Konvergenz und wird durch $X_i\to X$ $(P$-$)$f.s. dargestellt. Die Konvergenz nach Verteilung wird auch als schwache Konvergenz bezeichnet. Die folgenden Implikationen lassen sich leicht nachweisen.

schwache Konvergenz	$\Leftarrow$	stochastische Konvergenz	$\Leftarrow$	Konvergenz im r-ten Mittel
		$\Uparrow$		
		Konvergenz mit Wahrscheinlichkeit 1		

Ausgehend von einem Wahrscheinlichkeitsraum $(\Omega,\mathcal{S},P)$ betrachten wir spezielle Folgen $\{X_i\}_{i\in\mathbb{N}}$, von reellen Zufallsvariablen $X_i:\Omega\to\mathbb{R}$, $i\in\mathbb{N}$, deren Quadrate $X_i^2:\ \Omega\to\mathbb{R}$, $\omega\mapsto X_i^2(\omega)$ für alle $i\in\mathbb{N}$ $(P$-$)$integrierbar sind. Wegen

$$\begin{aligned}\int_\Omega |X_i|\,dP &= \int_{\{\omega\in\Omega;\,|X_i(\omega)|\le 1\}} |X_i|\,dP+\int_{\{\omega\in\Omega;\,|X_i(\omega)|>1\}} |X_i|\,dP\\ &\le 1+\int_{\{\omega\in\Omega;\,|X_i(\omega)|>1\}} |X_i|\,dP\ \le\ 1+\int_\Omega X_i^2\,dP \quad \text{für alle } i\in\mathbb{N}\end{aligned}$$

besitzen die Zufallsvariablen X_i, $i\in\mathbb{N}$, endliche Erwartungswerte. Dies erlaubt die folgende Definition.

Definition A.50 (Der zentrale Grenzwertsatz)
Sei $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum und $\{X_i\}_{i\in\mathbb{N}}$ eine Folge von reellen Zufallsvariablen $X_i:\Omega\to\mathbb{R}$, $i\in\mathbb{N}$, deren Quadrate $X_i^2:\ \Omega\to\mathbb{R}$, $\omega\mapsto X_i^2(\omega)$ für alle $i\in\mathbb{N}$ (P-)integrierbar sind mit Varianzen $\mathbf{Var}(X_i)>0$ für alle $i\in\mathbb{N}$. Wir vereinbaren, daß für die Folge $\{X_i\}_{i\in\mathbb{N}}$ genau dann der zentrale Grenzwertsatz gilt, wenn die Folge $\{T_i\}_{i\in\mathbb{N}}$ standardisierter reeller Zufallsvariablen

$$T_i:\Omega\to\mathbb{R},\ \omega\mapsto\frac{\sum_{j=1}^{i}\left(X_j-\mathbf{E}\left(X_j\right)\right)}{\sqrt{\mathbf{Var}\left(\sum_{j=1}^{i}X_j\right)}},\quad i\in\mathbb{N},$$

in Verteilung gegen eine $\mathcal{N}(0,1)$ normalverteilte Zufallsvariable konvergiert.

⌟

Satz A.51 (Der zentrale Grenzwertsatz für stoch. unabh., identisch vert. Zufallsvariable)
Seien $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum und $\{X_i\}_{i\in\mathbb{N}}$ eine Folge stochastisch unabhängiger, identisch verteilter (d.h. $P_{X_i}=P_{X_j}$ für alle $i,j\in\mathbb{N}$) reeller Zufallsvariablen $X_i:\Omega\to\mathbb{R}$ mit $0<\mathbf{Var}(X_i)<\infty$ für alle $i\in\mathbb{N}$, dann gilt für $\{X_i\}_{i\in\mathbb{N}}$ der zentrale Grenzwertsatz.

⌟

Satz von de Moivre-Laplace
Besteht im obigen Satz die Folge $\{X_i\}_{i\in\mathbb{N}}$ aus stochastisch unabhängigen, $\mathrm{B}(1,p)$ binomialverteilten Zufallsvariablen, so wird die Gültigkeit des zentralen Grenzwertsatzes für $\{X_i\}_{i\in\mathbb{N}}$ als Satz von de Moivre-Laplace bezeichnet. In diesem Fall ist $X_1+\ldots+X_n$, $n\in\mathbb{N}$, $\mathrm{B}(n,p)$ binomialverteilt und die für große n aufwendig zu berechnende Binomial-Verteilung läßt sich somit durch die häufig tabellierte $\mathcal{N}(0,1)$ Normalverteilung approximieren.

Abschließend betrachten wir ein sehr hilfreiches Resultat.

Satz A.52 (Ungleichung von Chebyschev-Markov)
Seien $(\Omega,\mathcal{S},P)$ ein Wahrscheinlichkeitsraum und $X:\ \Omega\to\bar{\mathbb{R}}$ eine numerische Zufallsvariable, dann gilt für jedes Paar reeller Zahlen $\alpha>0$, $\kappa>0$ die folgende Ungleichung von Chebyschev-Markov

$$P\left(\{\omega\in\Omega;\,|X(\omega)|\geq\alpha\}\right)\leq\frac{1}{\alpha^{\kappa}}\int|X|^{\kappa}\,dP.$$

⌟

Literaturverzeichnis

[BB90] Sergio Benedetto & Ezio Biglieri. *Principles of Digital Transmission With Wireless Applications*. Information Technology: Transmission, Processing, and Storage. Kluwer Academic Publishers, Dordrecht, Boston, London, 1990.

[BCJR74] L. R. Bahl, J. Cocke, F. Jelinek & J. Raviv. *Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate*. IEEE Transactions on Information Theory, **20**, 284–287, März 1974.

[BDMP98] Sergio Benedetto, Dariush Divsalar, Guido Montorsi & Fabrizio Pollara. *Serial Concatenation of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding*. IEEE Transactions on Information Theory, **44**(3), 909–926, Mai 1998.

[BGH+00] Anton Buchmeier, Michael Greiner, Claus Hillermeier, Stefan Schäffler & Thomas F. Sturm. On the Construction of a Super-Channel Using Soft-Output Decoding of Systematic Linear Block Codes. In Michael Greiner & Manfred Jobmann, editors, *Stochastic Modeling of High-Speed Networks*, pages 149–154. CS Press, München, 2000.

[BM96] Sergio Benedetto & Guido Montorsi. *Unveiling turbo codes: Some results in parallel concatenated coding schemes*. IEEE Transactions on Information Theory, **42**, 409–429, März 1996.

[BRC60a] R. C. Bose & D. K. Ray-Chaudhuri. *Further results on error correcting binary group codes*. Inf. and Control, **3**, 279–290, September 1960.

[BRC60b] R. C. Bose & D. K. Ray-Chaudhuri. *On a class of error correcting binary group codes*. Inf. and Control, **3**, 68–79, März 1960.

[Bau92] Heinz Bauer. *Maß- und Integrationstheorie*. de Gruyter Verlag, Berlin, New York, Zweite Auflage, 1992.

[Bau02] Heinz Bauer. *Wahrscheinlichkeitstheorie*. de Gruyter Verlag, Berlin, New York, 5te Auflage, 2002.

[Beu94] Albrecht Beutelspacher. *Kryptologie*. Vieweg Verlag, Braunschweig, Wiesbaden, 4te Auflage, 1994.

[Bos98] Martin Bossert. *Kanalcodierung*. Teubner Verlag, Stuttgart, Zweite Auflage, 1998.

[Bos99] Martin Bossert. *Channel Coding for Telecommunications*. Wiley, Chichester, 1999.

[CCR01] A. Bruce Carlson, Paul B. Crilly & Janet C. Rutledge. *Communication Systems*. Electrical and Computer Engineering. McGraw-Hill, Inc., New York, 4te Auflage, 2001.

[CCS92] CCSDS 101.0-B-3. *Telemetry Channel Coding*. Recommendation for Space Data System Standards, Consultative Committee for Space Data Systems, CCSDS Secretariat, Communications and Data Systems Division, NASA, Washington, DC 20546, USA, Mai 1992.

[DB96] Klaus David & Thorsten Benkner. *Digitale Mobilfunksysteme*. Informationstechnik. Teubner Verlag, Stuttgart, 1996.

[DR87] Wilbur B. Davenport, Jr. & William L. Root. *An Introduction to the Theory of Random Sigals and Noise*. IEEE Communications Society, New York, 1987.

[For66] G. D. Forney, Jr. *Concatenated Codes*. MIT Press, Cambridge, Massachusetts, 1966.

[Fri95] Bernd Friedrichs. *Kanalcodierung*. Information und Kommunikation. Springer Verlag, Berlin, Heidelberg, New York, 1995.

[GSM96a] GSM 01.02. *Digital cellular telecommunications system (Phase 2+); General description of a GSM Public Land Mobile Network (PLMN)*. Technical Report, European Telecommunications Standard Institute, F-06921 Sophia Antipolis, France, März 1996. Version 5.0.0.

[GSM96b] GSM 05.03. *Digital cellular telecommunications system (Phase 2+); Channel coding*. European Telecommunication Standard ETS 300 565, European Telecommunications Standard Institute, F-06921 Sophia Antipolis, France, August 1996. Version 5.2.0.

[Gal62] R. G. Gallager. *Low density parity check codes*. IRE Trans. Info. Theory, **IT-8**, 21–28, Januar 1962.

[HHFJ02] Simon Hüttinger, Johannes Huber, Robert Fischer & Rolf Johannesson. Soft-Output-Decoding: Some Aspects From Information Theory. In *4. ITG Conference Source and Channel Coding*, pages 81–89, Berlin, Januar 2002.

[HOP96] Joachim Hagenauer, Elke Offer & Lutz Papke. *Iterative Decoding of Binary Block and Convolutional Codes*. IEEE Transactions on Information Theory, **42**(2), 429–445, März 1996.

[HQ95] Werner Heise & Pasquale Quattrocchi. *Informations- und Codierungstheorie*. Springer Verlag, Berlin, Heidelberg, New York, Dritte Auflage, 1995.

[Hag02] Joachim Hagenauer. The Turbo Principal in Mobile Communications. In *International Symposium on Information Theory and its Applications*, Xi'an, China, Oktober 2002.

[Hoc59] A. Hocquenghem. *Codes correcteurs d'erreurs*. Chiffres, **2**, 147–156, 1959.

[Hub02] Johannes Huber. *Grundlagen der Wahrscheinlichkeitsrechnung für iterative Decodierverfahren*. Springer e&i (Elektrotechnik und Informationstechnik), **119**(11), 386ff, November 2002.

[Jun95] Dieter Jungnickel. *Codierungstheorie*. Spektrum Akademischer Verlag, Heidelberg, Berlin, Oxford, 1995.

[LC83] Shi Lin & Daniel J. Costello, Jr. *Error Control Coding: Fundamentals and Applications*. Computer Applications in Electrical Engineering. Prentice-Hall, Englewood Cliffs, New Jersey, Erste Auflage, 1983.

[Lük99] Hans Dieter Lüke. *Signalübertragung*. Springer Verlag, Berlin, Heidelberg, New York, 7te Auflage, 1999.

[MN97] D. J. C. MacKay & R. M. Neal. *Near Shannon limit performance of low density parity check codes*. Electronics Letters, **33**(6), 457–458, März 1997.

[PW72] W. Wesley Peterson & E. J. Weldon, Jr. *Error-Correcting Codes*. MIT Press, Cambridge, Massachusetts, Zweite Auflage, 1972.

[Pät99] Matthias Pätzold. *Mobilfunkkanäle*. Nachrichtentechnik. Vieweg Verlag, Braunschweig, Wiesbaden, Erste Auflage, 1999.

[Pro01] John G. Proakis. *Digital Communications*. Electrical and Computer Engineering. McGraw-Hill, Inc., New York, 4te Auflage, 2001.

[RSS98] Klaus Ritter, Stefan Schäffler & Thomas F. Sturm. *Soft Decision Decoding of Binary Linear Block Codes Using the BFGS-Method*. Yugoslav Journal of Operations Research, **8**(1), 169–175, 1998.

[Rei95] Christian Reinsch. *Numerische Mathematik 1 und 2*. Vorlesungs-Skriptum TUM-MATH-11-95-00-250/3.-FMI, Technische Universität München, München, November 1995.

[Rie97] Sven Riedel. *Iterative Decodierung parallel verketteter binärer Faltungscodes*. Informatik/Kommunikationstechnik, Band 498. VDI Verlag, Düsseldorf, 1997.

[Roh95] Hermann Rohling. *Einführung in die Informations- und Codierungstheorie*. Teubner Verlag, Stuttgart, 1995.

[SS94] Stefan Schäffler & Thomas F. Sturm. *Wahrscheinlichkeitstheorie und Statistik I*. Vorlesungs-Skriptum IAMS Nr. 5, Technische Universität München, München, 1994.

[SS95] Stefan Schäffler & Thomas F. Sturm. *Wahrscheinlichkeitstheorie und Statistik II*. Vorlesungs-Skriptum IAMS Nr. 6, Technische Universität München, München, 1995.

[SW76] Claude E. Shannon & Warren Weaver. *Mathematische Grundlagen der Informationstheorie*. Scientia Nova. R. Oldenburg Verlag, München, Wien, Erste Auflage, 1976.

[Sch97] Stefan Schäffler. *Decodierung binärer linearer Blockcodes durch globale Optimierung*. Theorie und Forschung, Band 485. Roderer Verlag, Regensburg, 1997.

[Sha48] Claude E. Shannon. *A Mathematical Theory of Communication*. The Bell System Technical Journal, **27**, 379–423, 623–656, 1948.

[Tan81] R. M. Tanner. *A recursive approach to low complexity codes*. IEEE Transactions on Information Theory, **27**(5), 533–547, 1981.

[VO79] Andrew J. Viterbi & Jim K. Omura. *Principles of Digital Communication and Coding*. Electrical Engineering. McGraw-Hill, Inc., New York, Erste Auflage, 1979.

[Vit95] Andrew J. Viterbi. *CDMA: Principles of Spread Spectrum Communication*. Wireless Communications. Addison-Wesley, Reading, Massachusetts, Erste Auflage, 1995.

[Wib96] Niclas Wiberg. *Codes and Decoding on General Graphs*. Linköping Studies in Science and Technology. Dissertations No. 440, Linköping University, S-581 83 Linköping, Sweden, 1996.

Stichwortverzeichnis

D