Vorlesungsskript Kanalcodierung I

SoSe 2016

von

DR.-ING. VOLKER KÜHN

aktualisiert von

DR.-ING. DIRK WÜBBEN

Fachbereich Physik/Elektrotechnik (FB 1)
Arbeitsbereich Nachrichtentechnik
Postfach 33 04 40
D-28334 Bremen

Bearbeitungsstand: 21.03.2016

Inhaltsverzeichnis

Kapitel 1

Einführende Vorbetrachtungen

1.1 Aufbau und Ziel der Vorlesung

Die Vorlesung "Kanalcodierung I und II" soll die grundlegenden Prinzipien der Kanalcodierung und ihren Einsatz in praktischen Systemen verdeutlichen. Aufgrund der ungeheuren Vielzahl von Codes kann keine vollständige Beschreibung aller Verfahren, sondern nur eine kleine Auswahl der wichtigsten Codefamilien vorgestellt werden. Aus diesem Grund beschränkt sich die Vorlesung auf die in praktischen Systemen häufig eingesetzten Codes.

Die Vorlesung selbst gliedert sich in zwei Semester. Die "Kanalcodierung I" beschäftigt sich zunächst mit den Grundlagen, wie z.B. dem Aufbau digitaler Übertragungssysteme, der Informationstheorie und den zwei wichtigsten Vertretern aus der Klasse der Kanalcodes, nämlich den Blockcodes und den Faltungscodes. Im zweiten Semester werden dann aufbauend auf den erworbenen Kenntnissen komplexere Techniken und Systeme, wie z.B. die trelliscodierte Modulation, verknüpfte Codes, hybride FEC-/ARQ-Systeme und einige Beispiele wie das GSM-System, Satellitenübertragung und Modems betrachtet.

Die Kanalcodierung hat ihren Durchbruch mit der Einführung digitaler Übertragungssysteme errungen. Diese verdrängen nach und nach ihre analogen Vorgänger, da sie eine Reihe von Vorteilen besitzen. Hier sind effiziente Algorithmen zur Signalverarbeitung (wie z.B. Datenkompression, Entzerrungsverfahren usw.), aber auch die Möglichkeit leistungsfähiger Kanalcodierungs- und Modulationsverfahren zu nennen. Die Grundidee der Kanalcodierung und die ihr zugrundeliegende Informationstheorie wurden allerdings schon deutlich vor der Realisierung der ersten digitalen Systeme formuliert, nämlich 1948 von **C.E. Shannon**. Kapitel 2 beschäftigt sich mit den für diese Vorlesung wichtigen Aussagen der Theoreme von Shannon, so z.B. der Bedeutung der Kanalkapazität.

1.2 Grundlegendes zur Kanalcodierung

1.2.1 Der Begriff der Codierung

Bevor eine erste grobe Beschreibung der prinzipiellen Funktionsweise von Kanalcodierungsverfahren erfolgt, soll an dieser Stelle zunächst eine Abgrenzung zwischen verschiedenen Formen von Codierung gegeben werden. Dazu sind auch die Begriffe Nachricht, Information, Redundanz und Irrelevanz zu erläutern.

Nachricht:	Die Menge aller von der Quelle gesendeten Daten oder Zeichen
Information:	Derjenige Anteil der Nachricht, welcher für die Senke neu ist
Redundanz:	Die Differenz zwischen Nachricht und Information, die der Senke schon bekannt ist
	Nachricht = Information + Redundanz
Irrelevanz:	Information, die für die Senke **nicht** von Bedeutung ist, andernfalls ist sie relevant In der Literatur auch häufig: Fehlinformation, d.h. Information, die nicht von der gewünschten Quelle stammt
Fehlinformation:	Nicht von der gewünschten Quelle stammende Informationen

In der Praxis ist eine Nachricht immer in einer bestimmten Zeit zu übertragen. Damit ergeben sich die folgenden wichtigen Größen:

Nachrichtenfluß:	Nachrichtenmenge pro Zeit
Informationsfluß:	Informationsmenge pro Zeit
Transinformationsfluß:	Menge der fehlerfrei von der Quelle zur Senke übertragenen Information pro Zeit (vgl. Abschnitt 2.2).

Häufig werden drei große Klassen von Codierverfahren unterschieden, nämlich die Quellencodierung, die Kanalcodierung und die Kryptographie.

Quellencodierung (Entropiecodierung):

- Codierung des abgetasteten und quantisierten Signals mit minimaler Wortlänge

$\rightarrow$ Minimale Nachrichtenmenge (kleinstmöglichen Anzahl von Binärstellen), um Signal ohne Informationsverlust wieder zu rekonstruieren

$\rightarrow$ Vollständige Entfernung der enthaltenen Redundanz (Datenkompression)

- Entropie stellt untere Grenze der erforderlichen Nachrichtenmenge dar
  (Ohne Informationsverlust lässt sich kein Signal mit weniger Binärstellen als der Entropie darstellen.)
- Weiterführende Ansätze lassen gewissen Informationsverlust zu, um effektiver komprimieren zu können. (Verzerrung des Signals)

Kanalcodierung:

- Schutz des Datensignals vor Übertragungsfehlern durch Hinzufügen von Redundanz
- Im Vergleich zur Quellencodierung genau entgegengesetztes Konzept
- Erkennung bzw. Korrektur von Fehlern anhand der Redundanz

Kryptographie:

- Verschlüsselung des Datensignals zum Schutz vor unberechtigtem Abhören
- Entschlüsselung nur bei Kenntnis des Codeschlüssels möglich

Seit der bahnbrechenden Arbeit von Shannon im Jahr 1948 hat man versucht, jedes dieser drei Codierverfahren getrennt voneinander zu optimieren. Es existieren für jeden Bereich umfangreiche Theorien, die jede für sich genommen nur schwer in einem Buch umfassend dargestellt werden können. In den letzten Jahren zeichnet sich jedoch der Trend ab, beispielsweise Quellencodierung und Kanalcodierung nicht getrennt voneinander zu betrachten, sondern beide Verfahren miteinander zu kombinieren. Offensichtlich scheint es von Vorteil zu sein, eine 'optimale' Kombination beider Techniken zu nutzen, anstatt zunächst möglichst viel Redundanz aus dem Datensignal zu entfernen, um sie dann bei der Kanalcodierung wieder hinzuzufügen. Dies soll aber nicht Gegenstand dieser Vorlesung sein.

1.2.2 Verfahren der Kanalcodierung

Bezüglich der Kanalcodierung unterscheidet man zwei große Klassen:

- **ARQ** (*Automatic Repeat Request*)-Verfahren
- **FEC** (*Forward Error Correction*)-Verfahren

ARQ-Verfahren

- Signal wird durch einen ausschließlich fehler**erkennenden** Code geschützt.
- Bei Fehlerdetektion veranlasst Empfänger Wiederholung des fehlerhaften Blocks
- Existenz eines Rückkanals erforderlich (kann ebenfalls fehlerbehaftet sein)
- **Vorteil für gute Übertragungsbedingungen:**
  - Geringe Redundanz durch 'nur' fehler**erkennenden** Code
  - Mehr Redundanz (Wiederholung) nur dann, wenn nötig (Wiederholung im Fehlerfall).

  $\rightarrow$ Adaptive Fehlerkontrolle
- **Nachteil:** Stark reduzierter Datendurchsatz bei schlechten Übertragungsbedingungen (häufiges Wiederholen)
- Reine ARQ-Verfahren nur bei wenig gestörten Übertragungskanälen

FEC-Verfahren

- Einsatz fehler**korrigierender** Codes (höhere Redundanz)
- Kanal beeinflusst direkt die Qualität der Datenübertragung (unter ungünstigen Bedingungen wird Korrekturfähigkeit des Codes überschritten $\longrightarrow$ Restfehler)
- Höhere Redundanz auch bei guten Übertragungsbedingungen, hier geringerer Durchsatz als ARQ
- Kein Rückkanal für reine FEC-Verfahren (bei Restfehlern entweder Detektion durch fehlererkennenden Code und *Verschleierung* oder gar keine Erkennung)
- **FEC-Verfahren haben konstanten Datendurchsatz unabhängig vom aktuellen Kanalzustand**
- Einsatz bei stärker gestörten Kanälen

Hybride Verfahren:

- Kombination von FEC- und ARQ-Verfahren, um Vorteile zu vereinen und Nachteile zu vermeiden (siehe Kanalcodierung II)

1.2.3 Grundgedanke der Kanalcodierung

Wie bereits erwähnt wurde, fügen alle Kanalcodierungsverfahren dem Informationssignal Redundanz hinzu, anhand derer im Empfänger Fehler erkannt und/oder korrigiert werden können. Der Kanalcodierer erzeugt aus einem Eingangsvektor $\mathbf{u}$ der Länge k einen Vektor $\mathbf{x}$ der Länge $n > k$. Im binären Fall ließen sich damit insgesamt 2^n verschiedene Vektoren darstellen. Aufgrund der eindeutigen bijektiven Zuordnung des Codierers werden aber nur $2^k < 2^n$ Vektoren genutzt, d.h. nur eine Teilmenge aller möglichen Worte wird tatsächlich zur Übertragung verwendet. Eine wichtige Größe zur Charakterisierung eines solchen Codes ist die **Coderate**

$$R_c = \frac{k}{n}, \tag{1.1}$$

die das Verhältnis zwischen uncodierter und codierter Sequenzlänge beschreibt und für den uncodierten Fall ($k = n$) den Wert Eins annimmt. Die Coderate stellt auch ein Maß für die Erhöhung der erforderlichen Signalbandbreite dar, da nach der Codierung mehr Symbole ($n > k$) in der gleichen Zeit übertragen werden müssen.

Beispiele für das Ausnutzen von Redundanz zur Fehlerkorrektur:

- Sprache enthält Redundanz
- Lehre: Mehrfache Wiederholung des Stoffes (Beispiele)
- **Veranschaulichung der Kanalcodierung am Beispiel des Codewürfels:**

  Jede der acht Ecken eines Würfels stellt mögliches Wort dar und lässt sich mit drei Bit adressieren.

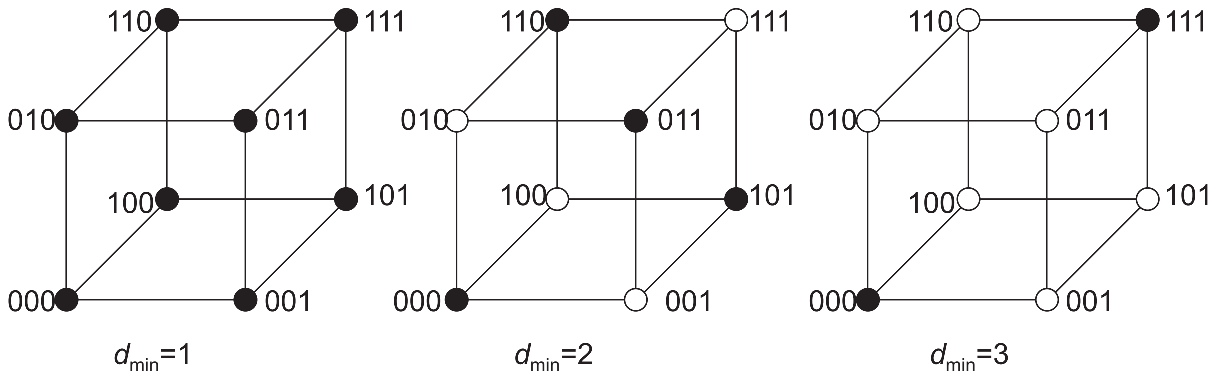


Bild 1.1: Veranschaulichung der Fehlerkorrektur/ -erkennung anhand des Codewürfels

<u>Linker Würfel:</u>

- Verwendung aller acht Ecken als Codeworte (uncodiert, Coderate $R_c = 1$)
- Kleinstmögliche Distanz zu einem anderen Codewort beträgt somit $d_{min} = 1$.
- Bei Übertragungsfehler wird wieder gültiges Codewort empfangen
- **KEINE Fehlererkennung und KEINE Fehlerkorrektur möglich**

<u>Mittlerer Würfel:</u>

- Verwendung jeder zweiten Ecke (•, Coderate $R_c = 2/3$)
- Kleinste vorkommende Distanz zwischen benachbarten Codeworten ist $d_{min} = 2$

- Bei Einzelfehler (1 Bit falsch) wird ungültiges Wort (○) empfangen
- 1-Fehler-Erkennung, KEINE Fehlerkorrektur möglich
  (Zuordnung des empfangenen Wortes zu einem direkten Nachbarn wäre rein zufällig)

Rechter Würfel:

- Verwendung von zwei der acht möglichen Ecken (•, Coderate $R_c = 1/3$)
- Erhöhung der minimalen Distanz auf $d_{min} = 3$
- 1-Fehler-Korrektur und 2-Fehler-Erkennung möglich
  Bsp.: Empfang von Wort '001', geringste Distanz zu '000' $\longrightarrow$ '000' als korrektes Wort annehmen

→ Fehlerkorrektur bzw. -detektion auf Kosten einer verringerten Datenrate möglich, d.h. pro Codewort werden statt 3 Bit nur zwei (mittlerer Würfel) bzw. 1 Bit (rechter Würfel) übertragen

→ Unveränderte Datenrate hat erhöhten Bandbreitenbedarf zur Folge

Zusammenfassung:

- Geschickte Bildung von Teilmengen eines Vektorraums erlaubt Fehlerdetektion und -korrektur
- Vorschriften oder Strategien zur Bildung der Teilmengen sehr schwierig
- Rein zufällige Auswahl nicht geeignet
- Bei Blockcodes Verwendung von algebraischen Methoden zur Codierungsvorschrift

→ Je leistungsfähiger die Codes, desto aufwendiger die mathematische Beschreibung

Aber: Hierdurch wahrscheinlich nur *schlechte* Codes zu finden (noch großer Abstand zur Kanalkapazität, siehe Kapitel 2)

- *Vielzahl heute bekannter Codes schlechter als die von Shannon verwendeten zufälligen Codes*

1.2.4 Praktische Bedeutung der Kanalcodierung

Ohne die Kanalcodierung wären viele der heute für uns selbstverständlichen Systeme überhaupt nicht oder aber nicht in der bekannten Form realisierbar. Codes zur Fehlererkennung und -korrektur werden nicht nur in wissenschaftlichen Anwendungen, wie z.B. bei Weltraummissionen (Viking (Mars), Voyager (Jupiter, Saturn), Galileo (Jupiter), Cassini) eingesetzt, sondern auch in Systemen des täglichen Lebens. Nahezu alle digitalen Speichermedien wie die CD (*Compact Disc*), die DVD (*Digitale Versatile Disc*), das Dat-Tape oder die Festplatte eines PC's schützen ihre Daten durch extrem leistungsfähige Codierungsverfahren. Aber nicht nur die Speicherung digitaler Daten, auch die Datenübertragung selbst ist vor Fehlern zu schützen. Das naheliegendste Beispiel sind die in den letzten Jahren stark verbreiteten digitalen Mobilfunksysteme (z.B. GSM, UMTS, LTE, LTE-Advanced) und WLAN-Systeme (Wirelss Local Area Network. Insbesondere die reine Datenübertragung (keine Sprache) erfordert eine hohe Übertragungsqualität (sehr niedrige Fehlerraten), die ohne Kanalcodierung nicht zu erreichen ist. Dies gilt beispielsweise auch für Datenverbindungen über Telefonleitungen (Modem oder DSL) für Anwendungen wie Internet, WWW und andere Dienste. Selbstverständlich verwenden auch in den vergangenen Jahren eingeführten neuen Medien wie digitaler Rundfunk (DAB, *Digital Audio Broadcasting*) und digitales Fernsehen (DVB, *Digital Video Broadcasting*) Verfahren zur Fehlerbehandlung.

Aufgrund der immer stärker dominierenden Stellung digitaler Systeme und ihrer Bedürftigkeit nach fehlerschützenden Maßnahmen nimmt die Kanalcodierung einen wichtigen Platz in der digitalen Datenübertragung ein. Dabei kommen je nach Einsatzgebiet sehr unterschiedliche Verfahren zum Einsatz, da z.B. bei Weltraummissionen die Signalbandbreite nur eine untergeordnete Rolle spielt, während sie bei der Datenübertragung über Telefonleitungen der leistungsbegrenzende Faktor ist. Anwendungsbeispiele werden eingehender gegen Ende des Semesters betrachtet.

1.3 Prinzipielle Struktur digitaler Übertragungssysteme

Um eine einheitliche Festlegung der Signalbezeichnungen zu gewährleisten, soll im Folgenden zunächst die prinzipielle Struktur digitaler Datenübertragungssysteme vorgestellt werden. Dazu zeigt Bild 1.2 das Blockschaltbild einer typischen Übertragungsstrecke.

Quelle: Sendet ein analoges Signal $d(t)$, z.B. Sprachsignal aus

Quellencodierer: Tastet analoge Signale ab, quantisiert und komprimiert sie

Digitale Quelle: Zusammenfassung von Quelle und Quellencodierer zu einem Block; liefert werte- und zeitdiskrete Datenvektoren $\mathbf{u} = [u_0\ u_1\ \dots\ u_{k-1}]$ der Länge k

Kanalcodierer: Erweitert $\mathbf{u}$ zu einem Vektor $\mathbf{x} = [x_0\ x_1\ \dots\ x_{n-1}]$ derart, dass Fehler erkannt oder sogar korrigiert werden können

Kanalcodierer kann sich aus mehreren Teilcodes zusammensetzen

Coderate: $R_c = \frac{k}{n}$

Modulator: Setzt die diskreten Vektoren $\mathbf{x}$ in analoge Signalformen um und verschiebt Signal ins Übertragungsband

Übertragungskanal Stellt das Übertragungsmedium dar

Eigenschaften bestimmen die Wahl der Codier- und Modulationsverfahren

Typische Übertragungsbedingungen im Bereich des Mobilfunks:

- Mehrwegeausbreitung $\longrightarrow$ Intersymbolinterferenzen
- Zeitvariantes Fading, d.h. Einbrüche der komplexen Einhüllenden,
- Additive Überlagerung von Rauschen, wegen zentralem Grenzwertsatz der Stochastik in guter Näherung gaußverteilt und weiß

Demodulation: Im wesentlichen Verschiebung zurück ins Basisband und Tiefpaßfilterung, evtl. auch Quantisierung des Signals

Diskreter Kanal: auch **Digitaler Kanal** genannt

Zusammenfassung von analogem Anteil des Modulators, Kanal und analogem Anteil des Demodulators

$\mathcal{A}_{in}$: Eingangsalphabet des diskreten Kanals

$\mathcal{A}_{out}$: Ausgangsalphabet des diskreten Kanals

Kanaldecodierer: Schätzung des Infovektors $\mathbf{u}$ aus dem empfangenen Vektor $\mathbf{y}$ liefert $\hat{\mathbf{u}}$

$\mathbf{y}$ muss nicht zwangsläufig aus hart entschiedenen Werten bestehen

Analog zum Codierer kann auch Decodierer ebenfalls aus einzelnen Modulen bestehen

Quellendecodierer: Gegenstück vom Quellencodierer, bereitet $\hat{\mathbf{u}}$ für Senke auf

Diskrete Senke: Zusammenfassung von Quellendecodierer und Senke

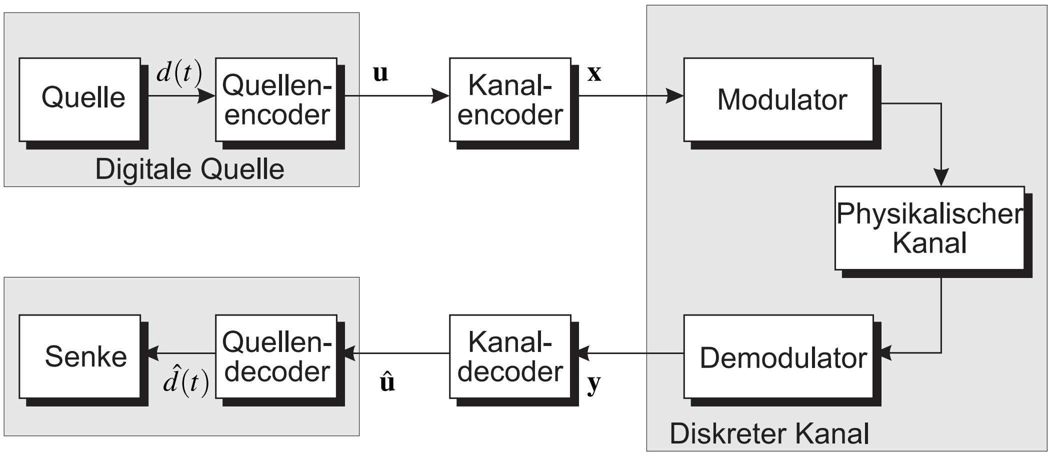


Bild 1.2: Blockschaltbild eines digitalen Übertragungssystems

1.4 Der diskrete Kanal

1.4.1 Einführung

Die Trennung zwischen Kanalcodierer und Modulator in Bild 1.2 ist nicht immer eindeutig. Insbesondere bei der *Codierten Modulation*, die im nächsten Semester vorgestellt wird, verschmelzen Codierer und Modulator zu einer Einheit. Daher wollen wir im Folgenden digitale und analoge Komponenten im Modulator und auch im Demodulator trennen und folgende Vereinbarung treffen:

Kanalcodierer ist für reine Kanalcodierung und auch Signalraumcodierung zuständig. Letztere bildet logische Symbole 0 und 1 auf diskrete Modulationssymbole ab

Modulator enthält alle analogen Anteile der Modulation (Impulsformung und Verschiebung ins Übertragungsband)

Demodulator führt die 'analoge' Demodulation durch (Verschiebung zurück ins Basisband, *matched*-Filterung und Abtastung im Symboltakt)
Gegebenenfalls wird hier auch noch eine Quantisierung des Signals vorgenommen.

Es gilt also:

Diskreter Kanal = Analoger Modulator + physikalischer Kanal + analoger Demodulator.

Hierzu passt die **Fundamentale Aussage von Massey (ETH Zürich):**

> *Die Aufgabe von Modulator und Demodulator besteht darin, zusammen mit dem physikalischen Medium einen möglichst guten diskreten Kanal zu bilden, während Codierer und Decodierer dafür verantwortlich sind, eine zuverlässige Übertragung über eben diesen diskreten Kanal zu gewährleisten.*

Aus den obigen Vereinbarungen folgt, dass die Eingangs- und Ausgangssignale aus zeitdiskreten Abtastwerten bestehen. Die Amplituden am Kanalausgang können dagegen kontinuierlich verteilt sein.

1.4.2 Eingangs- und Ausgangsalphabete des diskreten Kanals

Bei der hier betrachteten digitalen Übertragung enthält das Eingangsalphabet $\mathcal{A}_{in}$ eine endliche Anzahl diskreter Elemente X_ν. Der Einfachheit halber wollen wir an dieser Stelle lediglich den binären Fall einer antipodalen Übertragung (BPSK, *Binary Phase Shift Keying*) berücksichtigen. Dann gilt für den Eingangsvektor $\mathbf{x}$ des diskreten Kanals

$$\mathbf{x} = [x_0\, x_1\, \ldots\, x_{n-1}] \quad \text{mit} \quad x_i \in \mathcal{A}_{in} = \{-1, +1\}\ .$$

In der Regel besitzen beide X_ν die gleiche Auftrittswahrscheinlichkeit $P(X_\nu) = 1/2$. Wie schon erwähnt, beinhaltet der Kanalcodierer aus Bild 1.2 auch den Signalraumcodierer, d.h. er setzt die aus der eigentlichen Codierung hervorgegangenen logischen Werte 0 und 1 in antipodale Symbole $+1$ und -1 um. Sie bilden die Eingangssignale des diskreten Kanals (s. Bild 1.3). Die Energie E_s der Sendesymbole wird demnach auf Eins normiert und mit der Rauschleistung auf dem Kanal (s. AWGN-Kanal in Abschnitt 1.4.4) verrechnet. Eine Verallgemeinerung auf höherstufige Modulationsverfahren erfolgt dann in der Vorlesung Kanalcodierung II im Kapitel *Codierte Modulation*.

Bild 1.3: Diskreter Kanal

Die Form der Ausgangssignale des diskreten Kanals bei gegebenem Eingangsalphabet hängt nun von dem jeweiligen Demodulationsverfahren ab. Bei einer **Hard-Decision** sind das Eingangsalphabet $\mathcal{A}_{in}$ und das Ausgangsalphabet $\mathcal{A}_{out}$ des Kanals identisch $\mathcal{A}_{in} = \mathcal{A}_{out}$. Allerdings geht in diesem Fall Information verloren, da in dem hart entschiedenen Symbol keine Information über die Zuverlässigkeit des empfangenen Wertes mehr enthalten ist.

Besser ist eine q-Bit-**Soft Decision**, die im Extremfall $q \to \infty$ kontinuierlich verteilte Ausgangswerte ($\mathcal{A}_{out} = \mathbb{R}$) liefert. In der Praxis können auf einem digitalen System natürlich nur endliche viele Quantisierungsstufen realisiert werden. Bei weichen Entscheidungen enthält $\mathcal{A}_{out}$ mehr Elemente als $\mathcal{A}_{in}$. Im weiteren Verlauf gehen wir davon aus, dass am Eingang des diskreten Kanals ein Vektor $\mathbf{x}$ der Länge n mit den Elementen $x_i \in \mathcal{A}_{in}$ liegt. An seinem Ausgang erhalten wir den Vektor $\mathbf{y}$ gleicher Länge mit den Elementen $y_i \in \mathcal{A}_{out}$ (s. Bild 1.3).

1.4.3 Statistische Beschreibung des diskreten Kanals

Entscheidend für die Übertragungseigenschaften des diskreten Kanals ist die Wahrscheinlichkeit $P(y_i = Y_\mu | x_i = X_\nu)$, mit welcher bei einem gesendeten Symbol $x_i = X_\nu$ ein bestimmtes Symbol $y_i = Y_\mu$ angenommen wird. Diese werden Übergangswahrscheinlichkeiten genannt und charakterisieren das Fehlerverhalten des Kanals. Im Folgenden wird nur noch die vereinfachte Schreibweise $P(Y_\mu|X_\nu)$ verwendet. Wir wollen nun kurz die grundlegenden Zusammenhänge zwischen Wahrscheinlichkeiten, bedingten Wahrscheinlichkeiten und Verbundwahrscheinlichkeiten erläutern. Dabei beschränken wir uns zunächst auf diskrete Alphabete, bevor wir abschließend äquivalente Aussagen über kontinuierlich verteilte Signale mit Hilfe der Wahrscheinlichkeitsdichten treffen.

Zunächst Beschränkung auf diskrete Ausgangsalphabete

- **Auftrittswahrscheinlichkeit** eines diskreten Zeichens X_ν : $\qquad P(X_\nu)$
- **Verbundwahrscheinlichkeit** zweier Ereignisse X_ν und Y_μ : $\qquad P(X_\nu, Y_\mu)$
  gibt die Wahrscheinlichkeit des gleichzeitigen Auftretens von X_ν und Y_μ an

- Generell gilt:

$$\sum_{X_\nu \in \mathcal{A}_{in}} P(X_\nu) = \sum_{Y_\mu \in \mathcal{A}_{out}} P(Y_\mu) = 1 \tag{1.2}$$

$$P(Y_\mu) = \sum_{X_\nu \in \mathcal{A}_{in}} P(X_\nu, Y_\mu) \qquad \text{und} \qquad P(X_\nu) = \sum_{Y_\mu \in \mathcal{A}_{out}} P(X_\nu, Y_\mu) \tag{1.3}$$

$$\Longrightarrow \quad \sum_{Y_\mu \in \mathcal{A}_{out}} \sum_{X_\nu \in \mathcal{A}_{in}} P(X_\nu, Y_\mu) = 1 \tag{1.4}$$

Sind X_ν und Y_μ **statistisch unabhängig** voneinander, gilt:

$$P(X_\nu, Y_\mu) = P(X_\nu) \cdot P(Y_\mu) \tag{1.5}$$

In der Nachrichtentechnik ist bei Kenntnis des empfangenen Wertes Y_μ die Wahrscheinlichkeit, dass ein bestimmtes Zeichen X_ν gesendet wurde, von Interesse. Diese wird **a-posteriori-Wahrscheinlichkeit** genannt und lautet

$$P(X_\nu | Y_\mu) = \frac{P(X_\nu, Y_\mu)}{P(Y_\mu)} \tag{1.6}$$

Für eine möglichst zuverlässige Übertragung ist es dabei erstrebenswert, dass die a-posteriori-Wahrscheinlichkeit aus Gl. (1.6) bei jedem Y_μ für ein anderes X_ν groß ist, aber gleichzeitig für alle anderen $X_{i \neq \nu}$ sehr klein bleibt. In diesem Fall kann nämlich mit großer Sicherheit vom Empfangswert Y_μ auf den gesendeten Wert X_ν geschlossen werden. Für statistisch unabhängige Ereignisse X_ν und Y_μ besteht keinerlei Verbindung zwischen beiden Größen, so dass auch keine Rückschlüsse vom empfangenen Wert Y_μ auf einen möglichen gesendeten Wert X_ν gezogen werden können. Dann ist eine fehlerfreie Übertragung nicht möglich und für die bedingte Wahrscheinlichkeit gilt:

$$P(X_\nu | Y_\mu) = \frac{P(X_\nu, Y_\mu)}{P(Y_\mu)} = \frac{P(X_\nu) \cdot P(Y_\mu)}{P(Y_\mu)} = P(X_\nu). \tag{1.7}$$

Bayes-Regel: Umrechnung der a-posteriori-Wahrscheinlichkeit in die Übergangswahrscheinlichkeit

$$P(b|a) = \frac{P(a,b)}{P(a)} = \frac{P(a|b) \cdot P(b)}{P(a)} \tag{1.8}$$

d.h.

$$P(Y_\mu | X_\nu) = P(X_\nu | Y_\mu) \cdot \frac{P(Y_\mu)}{P(X_\nu)} \tag{1.9}$$

In der Praxis ist die bedingte Wahrscheinlichkeit $P(Y_\mu | X_\nu)$ unter bestimmten Voraussetzungen leichter zu berechnen.

Vorsicht: Es gilt

$$\sum_{X_\nu \in \mathcal{A}_{in}} P(X_\nu | Y_\mu) = \sum_{X_\nu \in \mathcal{A}_{in}} \frac{P(X_\nu, Y_\mu)}{P(Y_\mu)} = \frac{P(Y_\mu)}{P(Y_\mu)} = 1 \qquad \text{für alle} \qquad Y_\mu \in \mathcal{A}_{out}, \tag{1.10}$$

d.h. unter der Annahme eines empfangenen Symbols Y_μ wurde mit absoluter Sicherheit irgendein Symbol $X_\nu \in \mathcal{A}_{in}$ gesendet. Äquivalent gilt auch, dass für ein gesendetes Symbol X_ν am Empfänger mit absoluter Sicherheit irgendein Symbol aus $\mathcal{A}_{out}$ auftritt.

$$\sum_{Y_\mu \in \mathcal{A}_{out}} P(Y_\mu | X_\nu) = 1 \qquad \text{für alle} \qquad X_\nu \in \mathcal{A}_{in}$$

Aber:

$$\sum_{Y_\mu \in \mathcal{A}_{out}} P(X_\nu | Y_\mu) = \sum_{Y_\mu \in \mathcal{A}_{out}} \frac{P(X_\nu, Y_\mu)}{P(Y_\mu)} \neq 1 \tag{1.11}$$

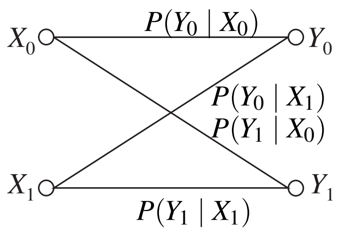
Beispiel:

Diskreter Kanal mit Ein- und Ausgangsalphabeten:
$\mathcal{A}_{in} = \mathcal{A}_{out} = \{0,\ 1\}$

Für Signale gilt:
$X_0 = Y_0 = 0$ und $X_1 = Y_1 = 1$

mit den Auftrittswahrscheinlichkeiten
$P(X_0) = P(X_1) = 0.5$

Übergangswahrscheinlichkeiten des Kanals:
$P(Y_0|X_0) = 0.9$ bzw. $P(Y_1|X_1) = 0.8$.



Aufgrund von $\sum_\nu P(X_\nu) = 1$ lassen sich die Übergangswahrscheinlichkeiten vervollständigen. Wir erhalten folgende Tabelle.

$P(Y\|X)$	X_0	X_1	
Y_0	0.9	0.2	$\longrightarrow \quad \sum_X P(Y\|X) = 1.1$
Y_1	0.1	0.8	$\longrightarrow \quad \sum_X P(Y\|X) = 0.9$
	$\downarrow$	$\downarrow$	
	$\sum_Y P(Y\|X) = 1$	$\sum_Y P(Y\|X) = 1$	

Mit Hilfe von Tabelle 1.4.3 und den Eingangswahrscheinlichkeiten $P(X_\nu)$ kann entsprechend auch eine Tabelle der Verbundwahrscheinlichkeiten angegeben werden.

$P(Y,X)$	X_0	X_1	
Y_0	0.45	0.1	$\longrightarrow \quad \sum_X = P(Y_0) = 0.55$
Y_1	0.05	0.4	$\longrightarrow \quad \sum_X = P(Y_1) = 0.45$
	$\downarrow$	$\downarrow$	$\downarrow$
	$\sum_Y = P(X_0) = 0.5$	$\sum_Y = P(X_1) = 0.5$	$\longrightarrow \quad \sum = 1$

Zuletzt erhalten wir noch die Tabelle der a-posteriori-Wahrscheinlichkeiten.

$P(X\|Y)$	X_0	X_1	
Y_0	0.818	0.182	$\longrightarrow \quad \sum_X P(Y\|X) = 1$
Y_1	0.111	0.889	$\longrightarrow \quad \sum_X P(Y\|X) = 1$

Aufgrund der Unsymmetrie des Kanals, d.h. das Eingangssymbol X_0 wird anders beeinflusst als X_1, ergeben sich trotz einer Gleichverteilung am Eingang unterschiedliche Werte für $P(X|Y)$ und $P(Y|X)$. Dies wäre bei einem symmetrischen Kanal (s. BSC, Abschnitt 1.4.7) nicht der Fall.

Für einen idealen Kanal würden die Tabellen folgende Form annehmen.

$P(Y\|X)$	X_0	X_1
Y_0	1	0
Y_1	0	1

$P(X,Y)$	X_0	X_1
Y_0	0.5	0
Y_1	0	0.5

$P(X\|Y)$	X_0	X_1
Y_0	1	0
Y_1	0	1

Übergang auf kontinuierlich verteilte Ausgangsalphabete

- Diskrete Wahrscheinlichkeiten $P(Y_\mu) \quad \longrightarrow \quad$ Wahrscheinlichkeitsdichtefunktionen $p_y(\xi)$
- Kontinuierlich verteilte Amplituden, wenn am Empfänger keine Quantisierung stattfindet
- Mit digitalen Techniken in der Praxis nicht umzusetzen
- Aber vom Standpunkt der Informationstheorie durchaus interessant, den durch Quantisierung entstehenden Verlust zu bestimmen

Alle obigen Beziehungen behalten weiterhin ihre Gültigkeit. Allerdings gehen die Summen für kontinuierlich verteilte Größen in Integrale über.

Beispiele:

$$P(X_\nu) = \int\limits_{\mathcal{A}_{out}} p_y(\xi, X_\nu)\, d\xi. \tag{1.12}$$

$$\int\limits_{\mathcal{A}_{out}} p_y(\xi|X_\nu)\, d\xi = 1. \tag{1.13}$$

Bei Quantisierung: Auftrittswahrscheinlichkeit der diskreten Werte Y_μ:

$$P(Y_\mu) = \int_{Y_\mu^-}^{Y_\mu^+} p_y(\xi)\, d\xi, \tag{1.14}$$

Y_μ^- und Y_μ^+ stellen untere bzw. obere Quantisierungsgrenze für den Wert Y_μ dar.

1.4.4 AWGN-Kanal

Nachdem im vorangegangenen Abschnitt die formale Beschreibung des diskreten Kanals abgehandelt wurde, sollen im Folgenden einige wichtige Beispiele für klassische Kanalmodelle vorgestellt werden. Dabei beschränken wir uns auf sehr einfache Modelle, die aber sehr wohl zur Beurteilung der Güte von Kanalcodierungsverfahren geeignet sind und auch in der Praxis eine wichtige Rolle spielen. Auf komplexere Modelle wie z.B. Mobilfunkkanäle wird an dieser Stelle verzichtet.

Einer der wichtigsten Kanäle in der Nachrichtentechnik ist der AWGN (*Additive White Gaussian Noise*)-Kanal, welcher den Eingangswerten weißes, gaußverteiltes Rauschen überlagert. Dabei charakterisiert das Attribut ’weiß’ eine konstante spektrale Leistungsdichte, d.h. aufeinanderfolgende Rauschwerte sind statistisch unabhängig, was für die Leistungsfähigkeit der Kanalcodierungsverfahren von großer Bedeutung ist. Eine klassische Anwendung ist die Satellitenkommunikation, deren Übertragung primär durch den AWGN-Kanal beeinflusst wird. Die Annahme gaußverteilter additiver Störsignale beruht auf der Tatsache, dass in der Praxis eine Vielzahl von Rauschprozessen die Übertragung stören (thermisches Rauschen von Bauelementen, Sonnenstrahlung, ...), deren Überlagerung durch den zentralen Grenzwertsatz der Stochastik in guter Näherung als gaußverteilt und weiß angesehen werden kann. Die Gaußverteilung lautet

$$p_n(\eta) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{\eta^2}{2\sigma^2}}. \tag{1.15}$$

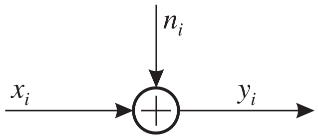


Bild 1.4: AWGN-Kanal

Das Ausgangssignal des AWGN-Kanals besitzt dementsprechend eine kontinuierlich verteilte Amplitude. Das Modell des AWGN-Kanals zeigt Bild 1.4.

Charakteristisch für den AWGN-Kanal ist das **Signal-Rausch-Verhältnis** S/N, welches das Verhältnis der Signalleistung S zur Rauschleistung N beschreibt. Bei idealer *matched*-Filterung und Abtastung des *matched*-Filter-Signals im Symboltakt T_s kann dieses wie folgt umgerechnet werden

$$\frac{S}{N} = \frac{E_s/T_s}{N_0/2/T_s} = \frac{E_s}{N_0/2}, \tag{1.16}$$

wobei E_s die Symbolenergie und $N_0/2$ die spektrale Rauschleistungsdichte beschreiben. Für symmetrische Eingangsalphabete ist die Wahrscheinlichkeit für eine Fehlentscheidung beim AWGN-Kanal aufgrund der aus Bild 1.5 ersichtlichen Symmetrie unabhängig vom gesendeten Symbol. Sie hängt vielmehr vom Signal-Rausch-Verhältnis E_s/N_0 ab, so dass für eine antipodale Modulation ($+\sqrt{E_s/T_s}$, $-\sqrt{E_s/T_s}$) gilt

$$\begin{aligned} P_e &= P_{err}(x_i=-1) &&= \int_0^\infty p_{y|x}(\vartheta|\xi=-\sqrt{E_s/T_s})\,d\vartheta = \int_0^\infty p_n\left(\eta+\sqrt{E_s/T_s}\right)\,d\eta \\ &= P_{err}(x_i=+1) &&= \int_{-\infty}^0 p_{y|x}(\vartheta|\xi=+\sqrt{E_s/T_s})\,d\vartheta = \int_{-\infty}^0 p_n\left(\eta-\sqrt{E_s/T_s}\right)\,d\eta \\ & &&= \frac{1}{\sqrt{2\pi\sigma^2}}\cdot\int_0^\infty e^{-\frac{(\eta+\sqrt{E_s/T_s})^2}{2\sigma^2}}\,d\eta \end{aligned} \tag{1.17}$$

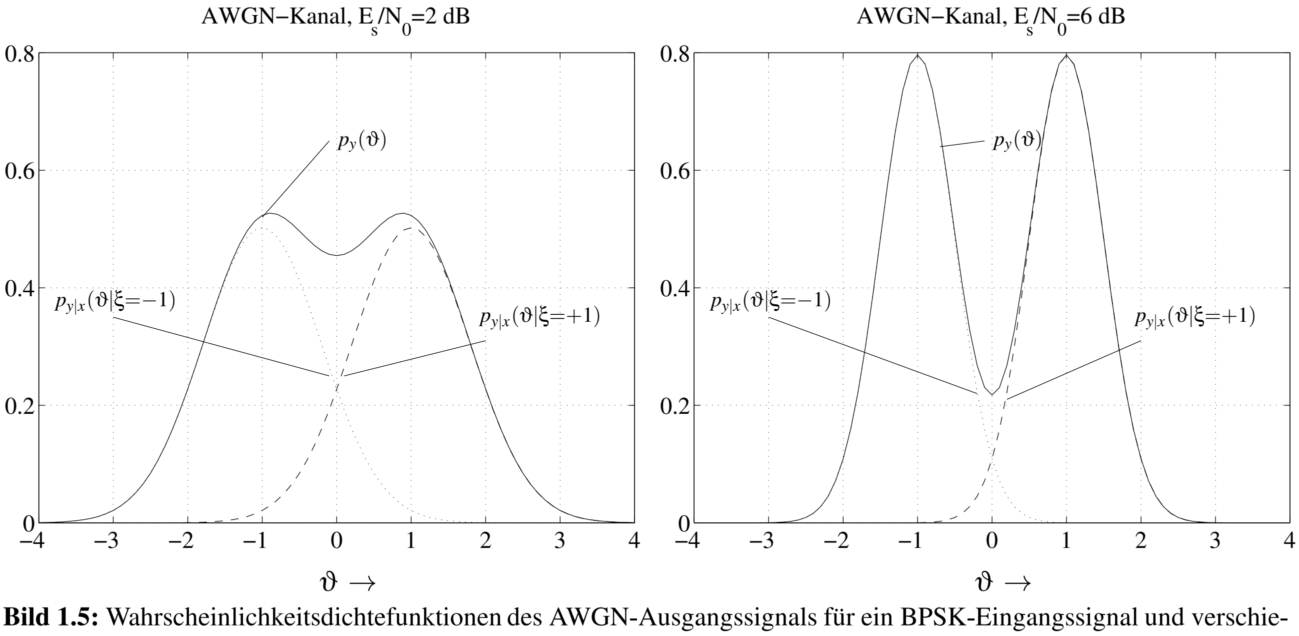


Bild 1.5: Wahrscheinlichkeitsdichtefunktionen des AWGN-Ausgangssignals für ein BPSK-Eingangssignal und verschiedene Signal-Rausch-Abstände

Setzt man für die Rauschleistung die Beziehung $N = \sigma^2 = N_0/2/T_s$ ein, so ergibt sich mit Hilfe der Substitutionen $\xi = (\eta + \sqrt{E_s/T_s})/\sqrt{N_0/T_s} \to \mathrm{erfc}()$ bzw. $\xi = (\eta + \sqrt{E_s/T_s})/\sqrt{N_0/2/T_s} \to Q()$

$$\begin{aligned} P_e &= \frac{1}{\sqrt{\pi N_0/T_s}} \cdot \int_0^\infty e^{-\frac{(\eta+\sqrt{E_s/T_s})^2}{N_0/T_s}} d\eta \\ &= \frac{1}{\sqrt{\pi}} \cdot \int\limits_{\sqrt{E_s/N_0}}^{\infty} e^{-\xi^2} d\xi &&= \frac{1}{\sqrt{2\pi}} \cdot \int\limits_{\sqrt{E_s/N_0/2}}^{\infty} e^{-\frac{\xi^2}{2}} d\xi \\ &= \frac{1}{2} \cdot \mathrm{erfc}\left(\sqrt{\frac{E_s}{N_0}}\right) &&= Q\left(\sqrt{\frac{2E_s}{N_0}}\right). \end{aligned} \tag{1.18}$$

Dabei stellen

$$Q(\alpha) = \frac{1}{\sqrt{2\pi}} \cdot \int_\alpha^\infty e^{-\frac{\eta^2}{2}} d\eta \tag{1.19}$$

und

$$\mathrm{erfc}(\alpha) = 1 - \mathrm{erf}(\alpha) = 1 - \frac{2}{\sqrt{\pi}} \cdot \int_0^\alpha e^{-\eta^2} d\eta = \frac{2}{\sqrt{\pi}} \cdot \int_\alpha^\infty e^{-\eta^2} d\eta \tag{1.20}$$

die komplementäre Gauß'sche Fehlerfunktion dar (vgl. Bild 1.6). Gl. (1.18) gilt für den Fall, dass für die Entscheidungsschwelle $w = 0$ gilt, d.h. sie liegt aufgrund der Symmetrie im Schnittpunkt der Kurven ($p_{y|x}(\vartheta|\xi = -1) = p_{y|x}(\vartheta|\xi = +1)$).

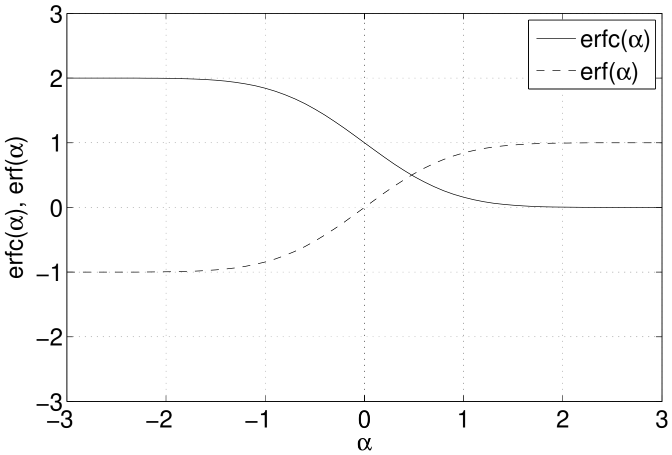


Bild 1.6: erfc(α) und erfc(α)-Funktion

1.4.5 Schwundkanal

Im Bereich der Mobilfunkübertragung stellen die Schwundkanäle eine wichtige Klasse von Übertragungskanälen dar. Mobilfunkkanäle sind geprägt durch starke zeitliche Schwankungen der komplexen Einhüllenden des Empfangssignals, die durch verschiedene Effekte hervorgerufen werden. Zum einen sorgen vorübergehende Abschattung des Signals durch Bäume oder Hochhäuser zum Einbruch des Empfangspegels. Außerdem treffen am Empfänger aufgrund von Reflexionen, Streuungen und Beugungen unterschiedlich verzögerte Signalanteile ein (Mehrwegeausbreitung), die sich konstruktiv, aber auch destruktiv überlagern können. Die auf diesem Effekt beruhenden Schwunderscheinungen werden auch als *Fading* bezeichnet. Ohne eine vollständige Beschreibung der Mobilfunkkanäle zu geben, soll an dieser Stelle stellvertretend der nicht-frequenzselektive Schwundkanal, der sogenannte 1-Pfad-Rayleigh-Kanal behandelt werden. Bild 1.7 zeigt das Blockschaltbild.

Das Nutzsignal x_i wird zunächst mit einem im allgemeinen komplexwertigen Faktor α_i multipliziert, dessen Real- und Imaginärteil jeweils gaußverteilt und statistisch unabhängig voneinander sind. Hieraus ergibt sich für den Betrag von α_i eine Rayleigh-Verteilung

$$p_{|\alpha|}(\xi) = \begin{cases} 2\xi/\sigma_s^2 \cdot \exp(-\xi^2/\sigma_s^2) & \text{für } \xi \geq 0 \\ 0 & \text{sonst .} \end{cases} \tag{1.21}$$

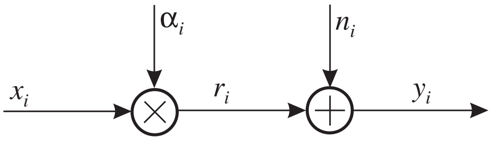


Bild 1.7: Übertragung beim 1-Pfad-Rayleigh-Kanal

Die entsprechenden Wahrscheinlichkeitsdichtefunktionen sind in Bild 1.8 dargestellt. Es ist zu erkennen, dass

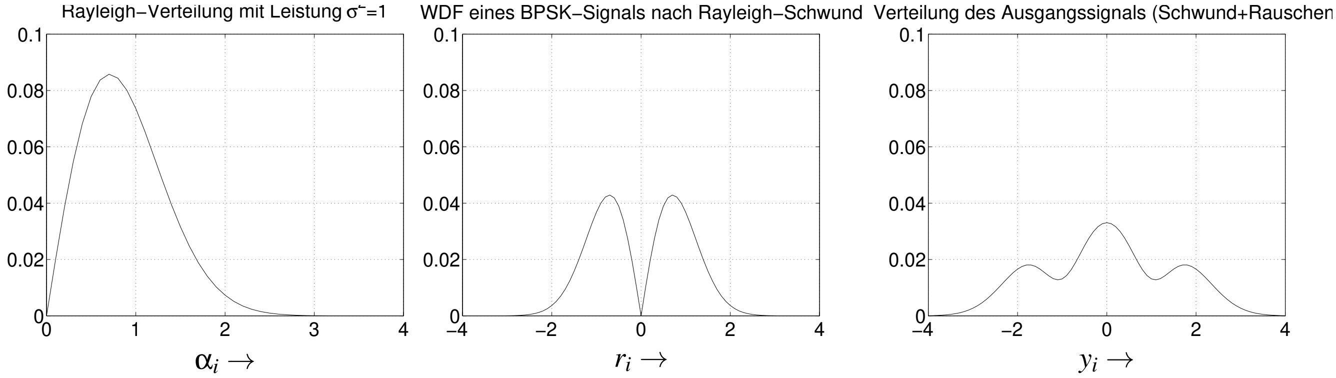


Bild 1.8: Wahrscheinlichkeitsdichtefunktionen beim 1-Pfad-Rayleigh-Kanal

im Vergleich zum AWGN-Kanal ein wesentlich größerer Anteil der Empfangswerte in der Nähe des Nullpunkts, also der Entscheidungsschwelle, liegt. Diese Werte sind dann sehr unsicher zu entscheiden, was sich in einer deutlich erhöhten Bitfehlerrate bemerkbar macht. Diese lautet (ohne Herleitung) für den 1-Pfad-Rayleigh-Kanal

$$P_b = \frac{1}{2} \cdot \left[1 - \sqrt{\frac{E_s/N_0}{1+E_s/N_0}} \right] . \tag{1.22}$$

Die Bitfehlerwahrscheinlichkeiten für eine uncodierte Übertragung über den AWGN- und den 1-Pfad-Rayleigh-

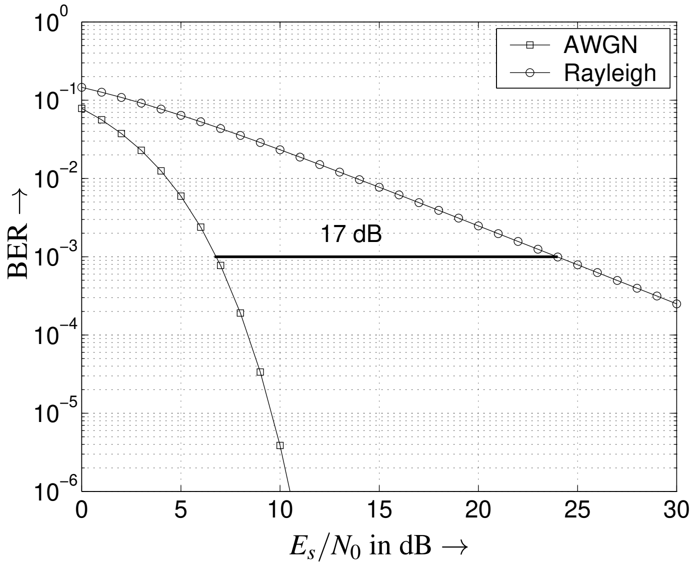


Bild 1.9: Bitfehlerwahrscheinlichkeiten beim AWGN- und 1-Pfad-Rayleigh-Kanal

Kanal zeigt Bild 1.9. Die schlechtere Qualität des Schwundkanals ist deutlich zu erkennen. Bei einer Fehlerrate von beispielsweise $P_b = 10^{-3}$, die häufig als Referenz für die Sprachübertragung im GSM-Netz verwendet wird, weist der Schwundkanal einen Verlust von etwa 17 dB auf. Dies bedeutet, dass man für den 1-Pfad-Rayleigh-Kanal die Sendeleistung um den 50 Faktor erhöhen muss, um die gleiche Übertragungsqualität wie beim AWGN-Kanal zu erhalten. Ausschlaggebend ist also häufig der horizontale Abstand zweier Kurven im Bitfehlerratendiagramm.

Neben der statistischen Betragsverteilung des Kanalkoeffizienten ist außerdem sein zeitliches Verhalten von entscheidender Bedeutung. Während der AWGN-Kanal statistisch unabhängige Störungen verursacht, sind beim Schwundkanal zeitlich aufeinander folgende Koeffizienten in der Regel nicht statistisch unabhängig voneinander. Dies führt dann häufig zu bündelartigen Störungen, so genannten *bursty errors*, gegenüber denen viele Kanalcodierungsverfahren sehr empfindlich sind. Hier kommt es nun darauf an, entweder speziell für Bündelfehler geeignete Codes zu konstruieren oder aber mit Hilfe von Interleavern, die die Reihenfolge der Daten verwürfeln, Bündelfehler in Einzelfehler aufzubrechen.

1.4.6 Diskreter gedächtnisloser Kanal (DMC)

Wie erwähnt können auf Digitalrechnern nur diskrete Werte verarbeitet werden. Daher ist im Demodulator digitaler Systeme immer eine Quantisierung des empfangenen Signals durchzuführen. In Abhängigkeit dieser Quantisierung kann $\mathcal{A}_{out}$ mehr Elemente enthalten als $\mathcal{A}_{in}$. Da ferner beim AWGN-Kanal aufeinanderfolgende Werte statistisch unabhängig sind, spricht man auch von gedächtnislosen Kanälen. Auch der 1-Pfad-Rayleigh-Kanal ist ein gedächtnisloser Kanal. Zeitlich aufeinanderfolgende Koeffizienten sind zwar häufig korreliert, aber ein Ausgangswert y_i wird nur von dem zugehörigen Eingangswert x_i beeinflusst und nicht von x_{i-m}.

Im allgemeinen Fall entsteht dann durch die Quantisierung ein diskreter, gedächtnisloser Kanal, welcher in der Literatur auch oft DMC (*Discrete Memoryless Channel*) genannt wird. Die Gedächtnislosigkeit drückt sich dadurch aus, dass die Übergangswahrscheinlichkeiten zu einem Zeitpunkt k nicht von vorangegangenen Zeitpunkten abhängen. Hierdurch lässt sich die Übergangswahrscheinlichkeit $P(\mathbf{y}|\mathbf{x})$ zwischen einem Eingangsvektor $\mathbf{x}$ und einem Ausgangsvektor $\mathbf{y}$ aus dem Produkt der Übergangswahrscheinlichkeiten der einzelnen Vektorelemente berechnen

$$P(\mathbf{y} \mid \mathbf{x}) = P(y_0, y_1, \dots, y_{n-1}|x_0, x_1, \dots, x_{n-1}) = \prod_{i=0}^{n-1} P(y_i|x_i). \tag{1.23}$$

Im Folgenden soll stets der binäre Fall, also eine BPSK-Modulation betrachtet werden. Die Erweiterung auf nicht-binäre Eingangsalphabete kann einfach hergeleitet werden. Wird das Kanalausgangssignal mit 2 Bit quantisiert, so entsteht ein 4-stufiges Signal, für welches das in Bild 1.10 dargestellte Übergangsdiagramm gilt. Die dort aufgeführten Übergangswahrscheinlichkeiten hängen von der verwendeten Quantisierungskennlinie ebenso wie von dem jeweiligen Signal-Rausch-Abstand ab.

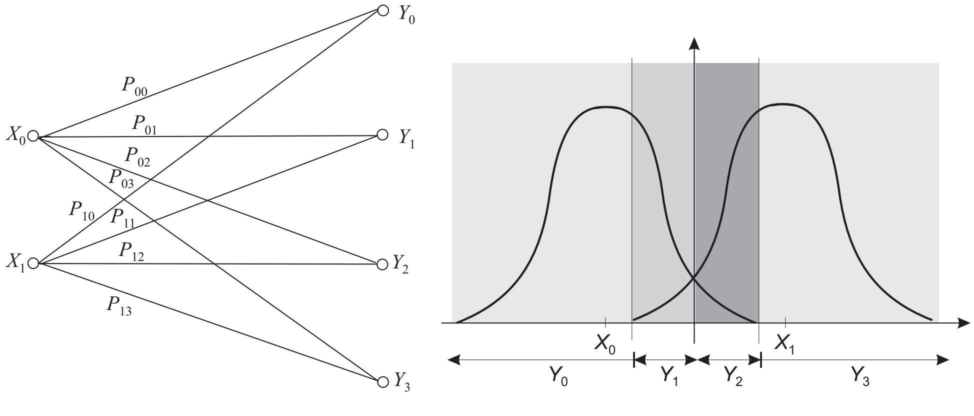


Bild 1.10: Übergangswahrscheinlichkeiten für BPSK und 2-Bit-Quantisierung

1.4.7 Binärer symmetrischer Kanal (BSC und BSEC)

Im Fall einer rein binären Übertragung mit Hard-Decision am Demodulator erhält man einen binären Kanal. Besitzt dieser zusätzlich gleiche Übergangswahrscheinlichkeiten P_e unabhängig vom gesendeten Symbol entsprechend Bild 1.11, so spricht man vom binären symmetrischen Kanal. Er stellt nicht nur ein theoretisches Hilfsmittel dar, sondern hat praktische Bedeutung, da er z.B. aus der Zusammenfassung von BPSK-Modulation, AWGN-Kanal und Hard-Decision-Demodulation hervorgeht.

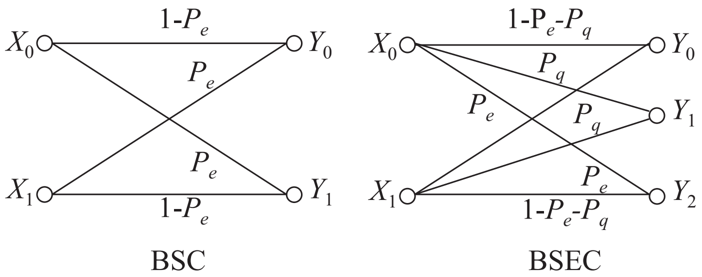


Bild 1.11: Veranschaulichung der Übergangswahrscheinlichkeiten beim BSC und BSEC

In diesem Fall hängen die Fehlerwahrscheinlichkeiten für ein gesendetes Symbol X_0 bzw. ein Symbol X_1 nur vom Signal-Rausch-Abstand E_s/N_0 ab. Sie sind beim BSC identisch und lauten

$$P(Y_0|X_1) = P(Y_1|X_0) = P_e = \frac{1}{2} \cdot \operatorname{erfc}\sqrt{\frac{E_s}{N_0}}\,. \tag{1.24}$$

Aufgrund der Gedächtnislosigkeit lassen sich die Fehlerwahrscheinlichkeiten für aus mehreren Symbolen bestehende Blöcke sehr einfach angeben. Die Wahrscheinlichkeit, dass eine Sequenz der Länge n korrekt empfangen wurde, lautet nämlich

$$\begin{aligned} P(\mathbf{x}=\mathbf{y}) &= P(x_0=y_0, x_1=y_1, \ldots, x_{n-1}=y_{n-1}) \\ &= \prod_{i=0}^{n-1} P(x_i=y_i) \\ &= (1-P_e)^n. \end{aligned} \tag{1.25}$$

Dagegen ist die Wahrscheinlichkeit für den fehlerhaften Empfang einer Sequenz, d.h. es tritt mindestens ein Fehler auf, zu

$$\begin{aligned} P(\mathbf{x}\neq\mathbf{y}) &= 1-P(x_0=y_0, x_1=y_1, \ldots, x_{n-1}=y_{n-1}) \\ &= 1-(1-P_e)^n \approx n P_e \quad \text{für} \quad n P_e \ll 1. \end{aligned} \tag{1.26}$$

gegeben. Die Wahrscheinlichkeit, dass genau m Fehler an <u>bestimmten</u> Stellen in einer Sequenz der Länge n auftreten, lautet

$$P(m \text{ Bit von } n \text{ falsch}) = P_e^m \cdot (1-P_e)^{n-m}. \tag{1.27}$$

Berücksichtigt man ferner die Anzahl der Kombinationen, m Bit in einer Sequenz der Länge n zu vertauschen, so erhält man die Wahrscheinlichkeit, dass m Fehler auftreten.

$$P(m \text{ Fehler in Sequenz der Länge } n) = \binom{n}{m} \cdot P_e^m \cdot (1-P_e)^{n-m} \tag{1.28}$$

Oftmals ist es von Vorteil, sehr unsichere Binärstellen auszublenden anstatt sie mit großer Wahrscheinlichkeit falsch zu entscheiden und dann mit fehlerhaften Werten weiterzurechnen. Für das Ausblenden wird ein drittes

Ausgangssymbol benötigt, welches praktischerweise den Wert Null annimmt. $\mathcal{A}_{out}$ enthält dann die Elemente $\{-1, 0, +1\}$, für die Übergangswahrscheinlichkeiten gilt dann:

$$P(x|y) = \begin{cases} 1 - P_e - P_q & \text{für } y = x \\ P_q & \text{für } y = ? \\ P_e & \text{für } y \neq x. \end{cases} \tag{1.29}$$

Den zugehörigen Übergangsgraphen zeigt Bild 1.11. Ein derartiger Kanal wird als BSEC (*Binary Symmetric Erasure Channel*) bezeichnet. Die Wahrscheinlichkeit für eine Fehlentscheidung lautet weiterhin P_e, während die neue Wahrscheinlichkeit für eine Auslöschung mit P_q bezeichnet wird. Wie diese Information dann konkret ausgenutzt wird, hängt vom jeweiligen Decodierverfahren ab und wird erst später behandelt.

Nachdem im nächsten Abschnitt einige Grundlagen der Informationstheorie erläutert wurden, folgt dann die Vorstellung der schon angesprochenen Blockcodes und der Faltungscodes.

Kapitel 2

Streifzug durch die Informationstheorie

2.1 Definitionen

Nachdem im vorangegangenen ersten Kapitel die grundlegende Struktur digitaler Übertragungssysteme, einige häufig verwendete Kanalmodelle sowie die prinzipielle Idee der Kanalcodierung vorgestellt wurden, folgt in diesem Kapitel eine kurze Einführung in die Informationstheorie. Sie bildet die Basis für die Konzeption aller modernen Kommunikationssysteme und wurde schon 1948 von C.E. SHANNON formuliert. Für diese Vorlesung ist das Kanalcodiertheorem von besonderer Bedeutung. Es beantwortet die wichtige Frage, mit welcher Rate über einen vorgegebenen Kanal noch fehlerfrei übertragen werden kann. Die maximal mögliche Rate wird dann als Kanalkapazität bezeichnet. Um die Frage beantworten zu können, sind zunächst einige grundlegende Begriffe zu klären.

Informationsgehalt, mittlerer Informationsgehalt, Entropie

Zunächst ist eine sinnvolle Größe zur Beschreibung des Informationsgehaltes $I(X_\nu)$ eines Symbols X_ν erforderlich. Das Informationsmaß soll dabei die folgenden Eigenschaften aufweisen:

- das Informationsmaß nimmt nur positive, reelle Werte an: $I(X_\nu) \in \mathbb{R}, I(X_\nu) \geq 0$
- das Informationsmaß ist eine Funktion der Symbolwahrscheinlichkeit: $I(X_\nu) = f(P(X_\nu))$
- für unabhängige Symbole $P(X_\nu, Y_\mu) = P(X_\nu) \cdot P(Y_\mu)$ soll gelten $I(X_\nu, Y_\mu) = I(X_\nu) + I(Y_\mu)$

Die einzige reelle Funktion $f(\cdot)$, die das Produkt reeller Zahlen auf die Summe der Funktionswerte abbildet ist der Logarithmus. Damit ergibt sich die nachfolgende Definition zu dem Informationsmaß.

Als Maß für den **Informationsgehalt** eines Zeichens oder Symbols wird der Logarithmus dualis vom Kehrwert der zugehörigen Auftrittswahrscheinlichkeit des entsprechenden Symbols

$$I(X_\nu) = \log_2 \frac{1}{P(X_\nu)} = -\log_2 P(X_\nu) \tag{2.1}$$

verwendet. Diese Definition erscheint sinnvoll, da mit ihr Symbole, die sehr selten auftreten, einen hohen Informationsgehalt besitzen, während er bei sehr oft vorkommenden Symbolen eher gering ist. Der Informationsgehalt ist stets positiv, da $0 \leq P(X_\nu) \leq 1$ gilt und der Kehrwert somit immer größer oder gleich Eins ist. Bei Verwendung des Logarithmus zur Basis 2 hat der Informationsgehalt die Dimension *bit*. Logarithmen mit einer Basis ungleich 2 sind zwar prinzipiell auch geeignet, allerdings basieren nahezu alle digitalen Systeme auf der binären Darstellung, wodurch sich die Dimension *bit* direkt anbietet. Häufig ist auch der **mittlere Informationsgehalt** von Interesse, welcher zu

$$H(X_\nu) = -P(X_\nu) \cdot \log_2 P(X_\nu) \tag{2.2}$$

definiert. Durch die Gewichtung des Informationsgehalts $I(X_\nu)$ mit der zugehörigen Auftrittswahrscheinlichkeit $P(X_\nu)$ wird berücksichtigt, dass sehr seltene Ereignisse zwar einen hohen Informationsgehalt besitzen, aufgrund ihres seltenen Auftretens aber auch nur einen geringen Beitrag zum Gesamtinformationsgehalt eines Zeichenvorrats leisten. Dieser wird **Entropie** genannt und setzt sich für statistisch unabhängige Ereignisse aus der Summe der mittleren Informationsgehalte der einzelnen Elemente

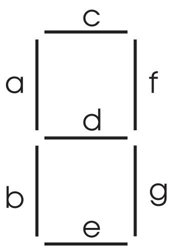
$$H(X) = \mathrm{E}\{-\log_2 P(X)\} = -\sum_\nu P(X_\nu) \cdot \log_2 P(X_\nu) \tag{2.3}$$

zusammen. Die Entropie einer Ereignismenge wird genau dann maximal, wenn eine Gleichverteilung aller Elemente vorliegt, d.h. alle Elemente mit der gleichen Wahrscheinlichkeit auftreten. Der mittlere Informationsgehalt nimmt dann für eine Menge mit 2^k Elementen den Wert

$$\max_{P(X)} H(X) = H_{gleich}(X) = \sum_\nu \frac{1}{2^k} \cdot \log_2 2^k = 2^k \cdot 2^{-k} \cdot k = k \text{ bit} \tag{2.4}$$

an. Entsprechend dem Quellencodiertheorem von SHANNON lassen sich bei optimaler Codierung die Elemente einer Quelle im Mittel mit genau $H(X)$ Bit darstellen. Ziel der Quellencodierung ist es, eine Codiervorschrift zu finden, deren mittlere Wortlänge der Entropie möglichst nahe kommt, wodurch sich die in den Codeworten enthaltene Redundanz minimiert.

Beispiel: Sieben-Segment-Anzeige zur Darstellung der Ziffern 0-9



Ziffer	a	b	c	d	e	f	g
0	1	1	1	0	1	1	1
1	0	0	0	0	0	1	1
2	0	1	1	1	1	1	0
3	0	0	1	1	1	1	1
4	1	0	0	1	0	1	1
5	1	0	1	1	1	0	1
6	1	1	1	1	1	0	1
7	0	0	1	0	0	1	1
8	1	1	1	1	1	1	1
9	1	0	1	1	1	1	1

Die 10 Ziffern treten alle mit der gleichen Wahrscheinlichkeit auf, so dass gilt:

$$P(X_\nu) = 0.1 \quad \forall \quad \nu\,.$$

Damit besitzen alle Ziffern von 0 bis 9 den gleichen Informationsgehalt

$$I(X_\nu) = \log_2 \frac{1}{0.1} = \log_2 10 = 3.32 \text{ bit}\,,$$

der mittlere Informationsgehalt lautet entsprechend

$$H(X_\nu) = P(X_\nu) \cdot I(X_\nu) = 0.332 \text{ bit}\,.$$

Die Entropie des Zeichenvorrats X ergibt sich dann zu

$$H(X) = \sum_{\nu} H(X_\nu) = 10 \cdot 0.332 \text{ bit} = 3.32 \text{ bit} = I(X_\nu)\,.$$

Der Entropie von 3.32 bit steht eine mittlere Wortlänge (Nachrichtenmenge) von $\bar{m} = 7$ bit gegenüber, da zur Darstellung aller Ziffern genau 7 Segmente, die die Zustände '0=aus' und '1=an' annehmen können, erforderlich sind. Die absolute Redundanz dieser Codierung beträgt somit

$$R = \bar{m} - H(X) = 7 \text{ bit} - 3.32 \text{ bit} = 3.68 \text{ bit}\,.$$

also mehr als die Hälfte des Informationsgehalts, die relative Redundanz beträgt

$$r = \frac{R}{\bar{m}} = \frac{\bar{m} - H(X)}{\bar{m}} = \frac{3.68 \text{ bit}}{7 \text{ bit}} = 0.5254$$

Verbundentropie, Äquivokation, Transinformation

Treten zwei Ereignisse X_ν und Y_μ nicht unabhängig voneinander auf, so sind sie mit Hilfe der Verbundwahrscheinlichkeit zu beschreiben. Die Information eines Ereignispaares lautet dann

$$I(X_\nu, Y_\mu) = \log_2 \frac{1}{P(X_\nu, Y_\mu)} = -\log_2 P(X_\nu, Y_\mu). \tag{2.5}$$

Entsprechend gilt für den mittleren Informationsgehalt des Paares

$$H(X_\nu, Y_\mu) = -P(X_\nu, Y_\mu) \cdot \log_2 P(X_\nu, Y_\mu) \tag{2.6}$$

und für die **Verbundentropie** des Alphabets

$$H(X,Y) = \mathrm{E}\{-\log_2 P(X,Y)\} = -\sum_{\nu}\sum_{\mu} P(X_\nu, Y_\mu) \cdot \log_2 P(X_\nu, Y_\mu). \tag{2.7}$$

Bild 2.1 illustriert die Zusammenhänge der verschiedenen Entropien auf sehr anschauliche Art und Weise. Seien X und Y zwei Zeichenvorräte mit den mittleren Informationsgehalten $H(X)$ bzw. $H(Y)$. Vor dem Hintergrund einer Datenübertragung stellt X das Signalraumalphabet des Sendesignals und Y das des Empfangssignals dar. Ist ein Teil der Information aus X nicht in Y enthalten, so ist 'unterwegs' Information verloren gegangen. Diese verlorene Information wird mit $H(X|Y)$ bezeichnet, d.h. als die bedingte mittlere Information, die X bei Kenntnis von Y noch liefern könnte. Sie heißt auch **Äquivokation**.

$$\begin{aligned}
H(X|Y) &= H(X,Y) - H(Y) && (2.8)\\
&= -\sum_{\nu}\sum_{\mu} P(X_\nu, Y_\mu) \cdot \log_2 P(X_\nu, Y_\mu) + \sum_{\mu} P(Y_\mu) \cdot \log_2 P(Y_\mu) \\
&= -\sum_{\nu}\sum_{\mu} P(X_\nu, Y_\mu) \cdot \log_2 P(X_\nu, Y_\mu) + \sum_{\mu}\sum_{\nu} P(X_\nu, Y_\mu) \cdot \log_2 P(Y_\mu) \\
&= -\sum_{\nu}\sum_{\mu} P(X_\nu, Y_\mu) \cdot \log_2 \frac{P(X_\nu, Y_\mu)}{P(Y_\mu)} \\
&= -\sum_{\nu}\sum_{\mu} P(X_\nu, Y_\mu) \cdot \log_2 P(X_\nu | Y_\mu) && (2.9)
\end{aligned}$$

Genau umgekehrt ist die Definition von $H(Y|X)$ zu verstehen. Enthält Y noch Information $H(Y|X)$, die nicht in X vorhanden ist, so kann diese nicht von der Quelle, also aus X, stammen und muss daher **Fehlinformation** sein. In der Literatur wird $H(Y|X)$ auch häufig **Irrelevanz** genannt, was eigentlich nicht ganz zutreffend ist, da irrelevante Information auch durchaus in X enthalten sein kann.

$$\begin{aligned} H(Y|X) &= H(X,Y) - H(X) \\ &= -\sum_{\nu}\sum_{\mu} P(X_\nu, Y_\mu) \cdot \log_2 P(Y_\mu|X_\nu) \end{aligned} \tag{2.10}$$

Die gesamte, im System enthaltene Information ist die **Verbundentropie** $H(X,Y)$. Die obigen Ableitungen können in der **Kettenregel der Entropie** zusammengefasst werden

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y) \leq H(X) + H(Y) \tag{2.11}$$

Derjenige Informationsanteil, welcher ungestört von der Quelle zur Senke gelangt, wird als **Transinformation** $I(X;Y)$ bezeichnet. Sie gilt es zu maximieren, wobei das Maximum als **Kanalkapazität** bezeichnet und im nächsten Abschnitt hergeleitet wird.

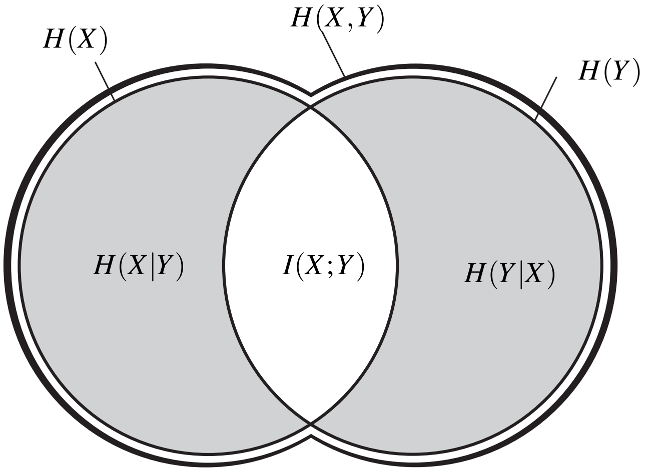


Bild 2.1: Veranschaulichung der Entropien

2.2 Kanalkapazität nach SHANNON

2.2.1 Kanäle mit diskretem Ausgangsalphabet $\mathcal{A}_{out}$

Die Aussagen des letzten Abschnitts sollen nun weiter konkretisiert werden. Dabei unterscheiden wir zwischen Kanälen mit diskretem Ausgangsalphabet und solchen, deren Ausgangswerte kontinuierlich verteilt sind ($\mathcal{A}_{out} = Y = \mathbb{R}$). In diesem Abschnitt werden zunächst nur diskrete Ausgangsalphabete betrachtet, welche sich durch Quantisierung des Kanalausgangssignals ergeben. Ausschlaggebend für das Verhalten eines Kanals ist die Kanalstatistik, die durch die Übergangswahrscheinlichkeiten $P(Y_\mu|X_\nu)$ bestimmt ist. Die Verteilung der $P(Y_\mu)$ Ausgangssymbole des Kanals wird sowohl von ihr als auch von der Quellenstatistik $P(X_\nu)$ beeinflusst, denn es gilt:

$$P(Y_\mu) = \sum_{\nu} P(X_\nu, Y_\mu) = \sum_{\nu} P(Y_\mu|X_\nu) \cdot P(X_\nu). \tag{2.12}$$

Wie im vorigen Abschnitt erwähnt wurde, besteht das Ziel darin, über einen vorgegebenen Kanal möglichst viel Information fehlerfrei zu übertragen. Mit anderen Worten, die Transinformation $I(X;Y)$ soll maximiert werden. Um die Einflüsse des Kanals aus der Sicht der Informationstheorie zu beleuchten, betrachten wir Bild 2.2.

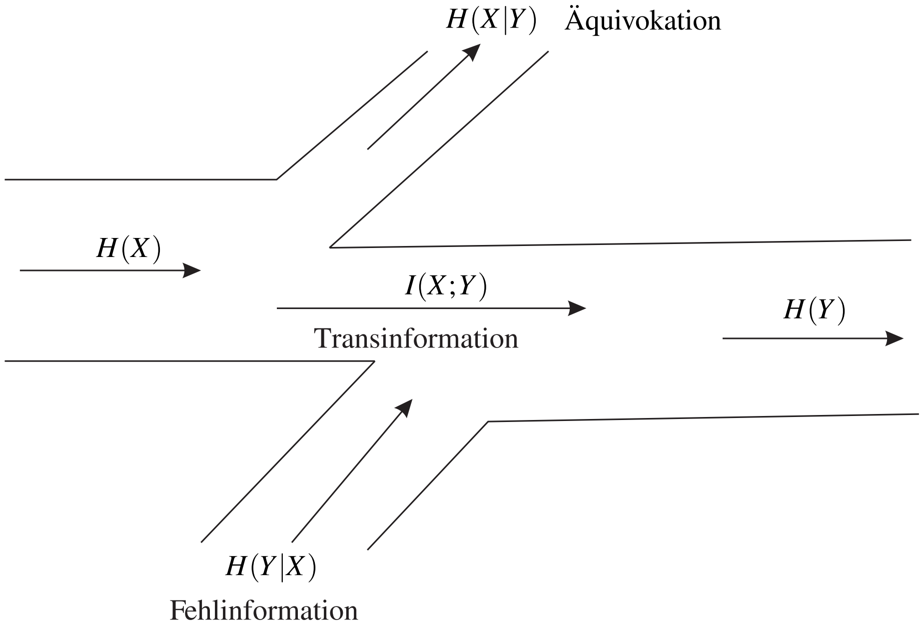


Bild 2.2: Bilanz der Entropien

Demnach setzt sich die Transinformation $I(X;Y)$ aus der Differenz der Quellenentropie $H(X)$ und der Äquivokation $H(X|Y)$ zusammen. Äquivalent kann sie auch als Differenz von $H(Y)$ und $H(Y|X)$ betrachtet werden.

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{2.13}$$

Entsprechend Bild 2.1 berechnen sich die bedingten Entropien aus Gl. (2.13) aus der Differenz der Verbundentropie $H(X,Y)$ und den Entropien von Quelle bzw. Senke. Wir erhalten

$$\begin{aligned} I(X;Y) &= H(X) - [H(X,Y) - H(Y)] \\ &= H(Y) - [H(X,Y) - H(X)] \\ &= H(X) + H(Y) - H(X,Y). \end{aligned} \tag{2.14}$$

Setzen wir in Gl. (2.14) die Gleichungen (2.3) und (2.7) ein, so ergibt sich

$$\begin{aligned} I(X;Y) = &-\sum_{\nu} P(X_\nu) \cdot \log_2 P(X_\nu) - \sum_{\mu} P(Y_\mu) \cdot \log_2 P(Y_\mu) \\ &+ \sum_{\nu}\sum_{\mu} P(X_\nu, Y_\mu) \cdot \log_2 P(X_\nu, Y_\mu). \end{aligned} \tag{2.15}$$

Mit Hilfe der Beziehung

$$P(a) = \sum_i P(a, b_i) \tag{2.16}$$

können die ersten beiden Summen in Gl. (2.16) erweitert werden. Wir erhalten damit

$$\begin{aligned} I(X;Y) &= -\sum_{\nu}\sum_{\mu} P(X_\nu, Y_\mu) \cdot \log_2 P(X_\nu) - \sum_{\nu}\sum_{\mu} P(X_\nu, Y_\mu) \cdot \log_2 P(Y_\mu) \\ &\quad + \sum_{\nu}\sum_{\mu} P(X_\nu, Y_\mu) \cdot \log_2 P(X_\nu, Y_\mu) \\ &= \sum_{\nu}\sum_{\mu} P(X_\nu, Y_\mu) \cdot [\log_2 P(X_\nu, Y_\mu) - \log_2 P(X_\nu) - \log_2 P(Y_\mu)] \\ &= \sum_{\nu}\sum_{\mu} P(X_\nu, Y_\mu) \cdot \log_2 \frac{P(X_\nu, Y_\mu)}{P(X_\nu) \cdot P(Y_\mu)} \end{aligned} \tag{2.17}$$

Nutzen wir ferner die Beziehung $P(a,b) = P(b|a) \cdot P(a)$ und nochmals Gl. (2.16) aus, so erhalten wir einen Ausdruck für die Transinformation, der nur noch von den Übergangswahrscheinlichkeiten $P(Y_\mu|X_\nu)$ des Kanals und der Statistik $P(X_\nu)$ des Eingangssignals abhängig ist.

$$\begin{aligned} I(X;Y) &= \sum_\nu \sum_\mu P(Y_\mu|X_\nu) \cdot P(X_\nu) \cdot \log_2 \frac{P(Y_\mu|X_\nu)}{P(Y_\mu)} && (2.18) \\ &= \sum_\nu \sum_\mu P(Y_\mu|X_\nu) \cdot P(X_\nu) \cdot \log_2 \frac{P(Y_\mu|X_\nu)}{\sum_l P(X_l, Y_\mu)} && (2.19) \\ &= \sum_\nu \sum_\mu P(Y_\mu|X_\nu) \cdot P(X_\nu) \cdot \log_2 \frac{P(Y_\mu|X_\nu)}{\sum_l P(Y_\mu|X_l) \cdot P(X_l)} && (2.20) \end{aligned}$$

Die **Kanalkapazität** C nach SHANNON ist nun als Maximum der Transinformation über alle möglichen Quellenstatistiken $P(X_\nu)$ definiert. Es gilt[1]:

$$C = \sup_{P(X)} \sum_\nu \sum_\mu P(Y_\mu|X_\nu) \cdot P(X_\nu) \cdot \log_2 \frac{P(Y_\mu|X_\nu)}{\sum_l P(Y_\mu|X_l) \cdot P(X_l)}. \tag{2.21}$$

Gl. (2.21) bedarf einiger zusätzlicher Erläuterungen. Das **Kanaltheorem von Shannon** besagt, dass durch Verwendung eines Codes, dessen Coderate R_c kleiner als die Kanalkapazität C ist, mit beliebig langer Codewortlänge eine beliebig kleine Fehlerwahrscheinlichkeit erreicht werden kann. Im Extremfall lässt sich somit bei unendlich langen Codeworten immer eine fehlerfreie Übertragung sicherstellen, solange die Coderate die Kanalkapazität nicht überschreitet. Kehrt man diese Formulierung um, so erhält man die Aussage, dass für $R_c > C$ auch mit noch so großem Aufwand keine fehlerfreie Übertragung erreicht werden kann.

Leider enthält das Kanaltheorem von Shannon keine Aussage darüber, mit welchem konkreten Code die Kapazitätsgrenze erreicht werden kann. In der Praxis gebräuchliche Codierungsverfahren sind in der Regel deutlich vom theoretischen Grenzwert entfernt, was selbstverständlich auch an dem begrenzten Realisierungsaufwand liegt. Ferner ist es in der Praxis nur selten möglich, die Statistik der Eingangsalphabete exakt den Bedürfnissen des Kanals anzupassen, insbesondere dann, wenn dieser zeitvariant ist. Für ein festes Eingangsalphabet mit gleichwahrscheinlichen Elementen gilt $P(X_\nu) \equiv 2^{-k}$, so dass sich die Kanalkapazität zu

$$C = 2^{-k} \cdot \sum_\nu \sum_\mu P(Y_\mu|X_\nu) \cdot \log_2 \frac{P(Y_\mu|X_\nu)}{2^{-k} \cdot \sum_l \cdot P(Y_\mu|X_l)} \tag{2.22}$$

ergibt. Gl. (2.22) verdeutlicht, dass die Kanalkapazität bei gleichwahrscheinlichen Eingangswerten des diskreten Kanals nur von den Übergangswahrscheinlichkeiten des Kanals abhängt. Sind diese bekannt, so lässt sich relativ einfach die maximale Datenrate angeben, die noch fehlerfrei über diesen Kanal übertragen werden kann. Allerdings enthält Gl. (2.22) keinerlei Aussage darüber, mit welchen Übertragungsverfahren, d.h. mit welchen Modulations- bzw. Codierungsverfahren die Kanalkapazität erreicht werden kann. Praktische Systeme erreichen die Kanalkapazität oft bei weitem nicht.

Binärer symmetrischer Kanal (BSC)

Im Folgenden sollen stellvertretend für die vielen möglichen Kanäle der BSC und der BSEC hinsichtlich ihrer Kanalkapazität untersucht werden. Eine ergänzende Betrachtung weiterer Kanalmodelle sowie eine Optimierung der Quellenstatistik bei vorgegebenen Kanal ist für die Übung vorgesehen. Die Kanalkapazität des binären symmetrischen Kanals kann direkt aus Gl. (2.22) hergeleitet werden, indem Gl. (1.24) eingesetzt wird. Wir erhalten

$$C^{\mathrm{BSC}} = 1 + P_e \cdot \log_2(P_e) + (1 - P_e) \cdot \log_2(1 - P_e)\ . \tag{2.23}$$

Bild 2.3 zeigt dazu die Kapazität des binären symmetrischen Kanals für verschiedene Eingangsstatistiken $P(X_\nu)$ und unterschiedliche Übergangswahrscheinlichkeiten P_e.

[1] Das Supremum gibt die kleinste obere Schranke einer Funktion an.

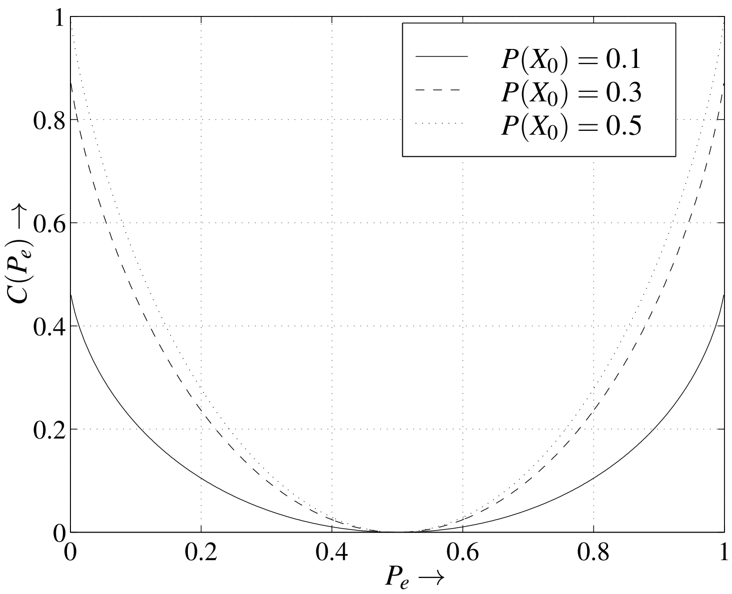


Bild 2.3: Kapazität des BSC für verschiedene Eingangsverteilungen

In Bild 2.3 ist zu erkennen, dass die Kanalkapazität ihr Maximum $C_{\max}^{\text{BSC}} = 1$ bit/s/Hz für die Grenzfälle $P_e = 0$ und $P_e = 1$ erreicht. Dann kann über den BSC fehlerfrei und uncodiert übertragen werden. Mit zunehmender Übergangswahrscheinlichkeit P_e nimmt die Kapazität ab, bis sie schließlich für $P_e = 0.5$ ihr Minimum $C_{\min}^{\text{BSC}} = 0$ bit/s/Hz erreicht. Dieses Ergebnis lässt sich anschaulich erklären, denn für $P_e = 0.5$ sind die fehlerfreie und die fehlerbehaftete Übertragung gleichwahrscheinlich, so dass die empfangenen Werte Y_μ rein zufällig sind und keine Korrelation mehr zu den Eingangswerten X_ν besitzen. Somit lässt sich keine Information mehr über den Kanal übertragen, seine Kapazität nimmt den Wert Null an. Für $P_e \approx 0.1$ besitzt die Kanalkapazität den Wert $C^{\text{BSC}}(0.1) \approx 0.5$ bit/s/Hz, d.h. bei einer optimalen Kanalcodierung mit der Coderate $R_c < 1/2$ kann theoretisch eine fehlerfreie Übertragung sichergestellt werden.

Ein Vergleich der Kurven für unterschiedliche Eingangsverteilungen (unterschiedliche $P(X_\nu)$) zeigt, dass die maximale Information bei gleichverteilten Zeichen $P(X_0) = P(X_1) = 0.5$ übertragen wird. Dies liegt an der symmetrischen Struktur des BSC. Für unsymmetrische Kanäle gelten entsprechend andere Bedingungen (s. Übung). Das Minimum der Kanalkapazität bei $P_e = 0.5$ ist unabhängig von der Verteilung des Eingangsalphabets.

Binärer symmetrischer Auslöschungskanal (BSEC)

Während der BSC das aus der Zusammenfassung von BPSK-Modulator, Kanal und Hard-Decision-Demodulator resultierende diskrete Kanalmodell darstellt, geht der BSEC (vgl. Abschnitt 1.4.7) aus dem Einsatz einer dreistufigen Demodulation hervor. Hier werden die unsichersten Empfangswerte, die in einem bestimmten Bereich um die Entscheidungsschwelle herum liegen, nicht hart entschieden, sondern als unbestimmt deklariert. Dies kann unter Umständen für die weitere Signalverarbeitung von Vorteil sein.

Als Beispiel für den BSEC betrachten wir nun einen AWGN-Kanal, dessen Ausgangssignal entsprechend Bild 2.5 dreistufig quantisiert wird. Alle Empfangswerte im Bereich $-a\sqrt{E_s/T_s} < y_i < +a\sqrt{E_s/T_s}$ werden dem Auslöschungssymbol Y_1 zugeordnet, wobei $a > 0$ eine geeignet zu wählende Konstante darstellt. Die Integration der Wahrscheinlichkeitsdichtefunktionen über den entsprechenden Teilbereichen liefert dann die Wahrscheinlichkeiten P_e und P_q in Abhängigkeit von E_s/N_0 und a. Mit ihnen kann dann durch Einsetzen in Gl. (2.22) die Kanalkapazität für den BSEC zu

$$C^{\text{BSEC}} = 1 - P_q + P_e \cdot \log_2(P_e) + (1 - P_e - P_q) \cdot \log_2(1 - P_e - P_q) - (1 - P_q) \cdot \log_2(1 - P_q)\ . \tag{2.24}$$

berechnet werden. Dabei ist zu beachten, dass P_e und P_q nicht unabhängig voneinander sind.

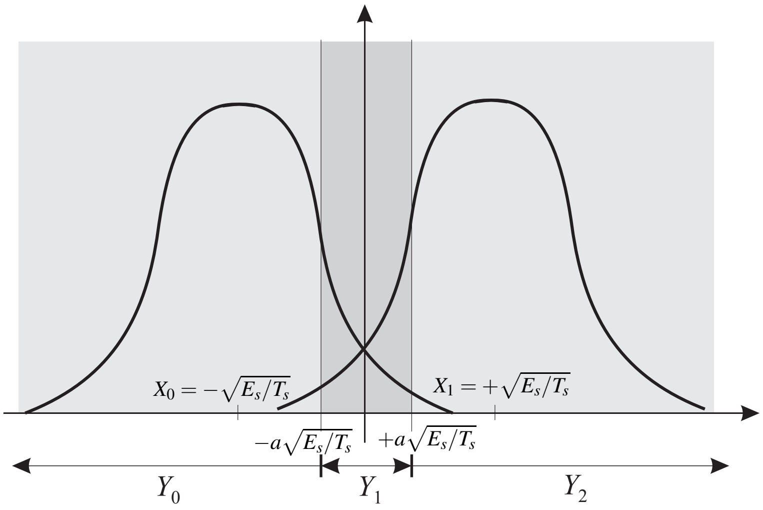


Bild 2.4: Dreistufige Quantisierung des Ausgangssignals eines AWGN-Kanals

Bild 2.5 zeigt die erzielten Ergebnisse. Links ist in der dreidimensionalen Darstellung die Abhängigkeit der Kanalkapazität vom Signal-Rausch-Abstand und der Konstanten a illustriert. Es ist zu erkennen, dass die optimale Wahl von a für verschiedene E_s/N_0 durchaus unterschiedliche Resultate liefern kann. Im Bereich großer Signal-Rausch-Abstände darf a nicht zu groß gewählt werden, da sonst zu viele 'gute' Empfangswerte als Auslöschung (unzuverlässig) deklariert werden. Für extrem kleine Signal-Rausch-Abstände scheint die Wahl von a dagegen von untergeordneter Bedeutung zu sein.

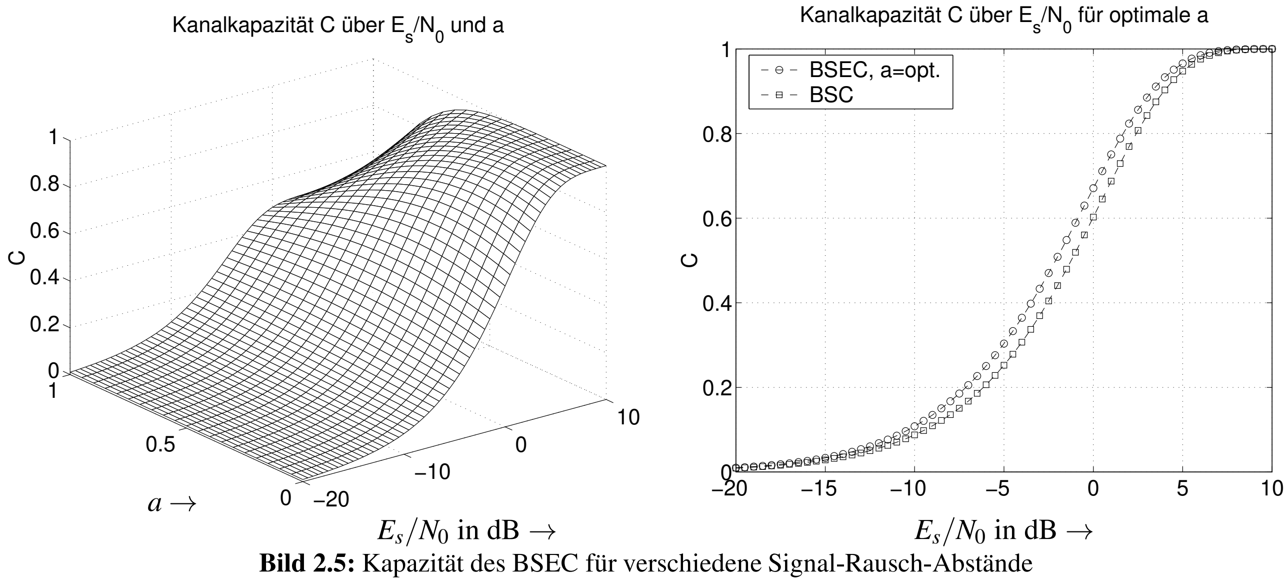


Bild 2.5: Kapazität des BSEC für verschiedene Signal-Rausch-Abstände

Im rechten Bild wurde für jedes E_s/N_0 der optimale Faktor a gewählt und die zugehörige Kanalkapazität des BSEC der des BSC ($a = 0$) gegenübergestellt. Der BSEC besitzt für alle Signal-Rausch-Abstände die größere Kapazität, d.h. über ihn kann bei gleichem E_s/N_0 eine höhere Datenrate fehlerfrei übertragen werden. Anders formuliert kann die gleiche Datenrate schon bei geringerem E_s/N_0 fehlerfrei gesendet werden. Dies liegt daran, dass der BSEC durch das dritte Symbol, die Auslöschung, die einfachste Form der *Soft-Decision* realisiert und somit mehr Information aus dem System nutzt als der einfache BSC. Da praktische Übertragungssysteme in der Regel deutlich von der Kanalkapazität entfernt arbeiten, ist es möglich, dass dort der Vorteil des BSEC noch größer ausfällt.

2.2.2 Kanäle mit kontinuierlichem Ausgangsalphabet $\mathcal{A}_{out}$

Ein häufig verwendetes Modell ist der schon vorgestellte AWGN-Kanal, welcher dem Sendesignal weißes gaußverteiltes Rauschen additiv überlagert. Hierdurch gilt $\mathcal{A}_{out} = \mathbb{R}$. Aufgrund der kontinuierlichen Verteilung der Amplituden von y geht die Summe in Gl. (2.22) in ein Integral über. Außerdem sind die diskreten Auftrittswahrscheinlichkeiten durch kontinuierliche Wahrscheinlichkeitsdichtefunktionen zu ersetzen. Die Kanalkapazität für einen AWGN-Kanal lautet somit bei diskretem Eingangssignal

$$C = 2^{-k} \cdot \int_{\mathcal{A}_{out}} \sum_{\nu} p_{y|x}(\vartheta|x = X_\nu) \cdot \log_2 \frac{p_{y|x}(\vartheta|x = X_\nu)}{2^{-k} \cdot \sum_l \cdot p_{y|x}(\vartheta|x = X_l)} \, d\vartheta. \tag{2.25}$$

Die Auswertung von Gl. (2.25) zeigt Bild 2.6. Dargestellt sind die Verläufe der Kanalkapazität über dem Signal-Rausch-Abstand für einen AWGN-Kanal bei BPSK-Modulation für unterschiedliche Quantisierungen. Eine 1-bit-Quantisierung ($q = 1$) entspricht einer Hard-Decision, eine 2-bit-Quantisierung ($q = 2$) zerlegt den Wertebereich in vier Teilräume (vgl. Bild 1.10), eine 3-bit-Quantisierung ($q = 3$) entsprechend in acht usw. Dabei ist anzumerken, dass durch die Quantisierung im Empfänger wiederum diskrete Ausgangsalphabete entstehen und somit die Zusammenhänge aus Abschnitt 2.2.1 gelten. Prinzipiell ist zu erkennen, dass die Quantisierung zu einem Informationsverlust führt, denn die Kanalkapazität verringert sich für alle Signal-Rausch-Abstände mit abnehmender Stufigkeit der Quantisierung. Allerdings scheint eine Quantisierung mit $q = 3$ Bit nur noch einen geringen Verlust zu verursachen.

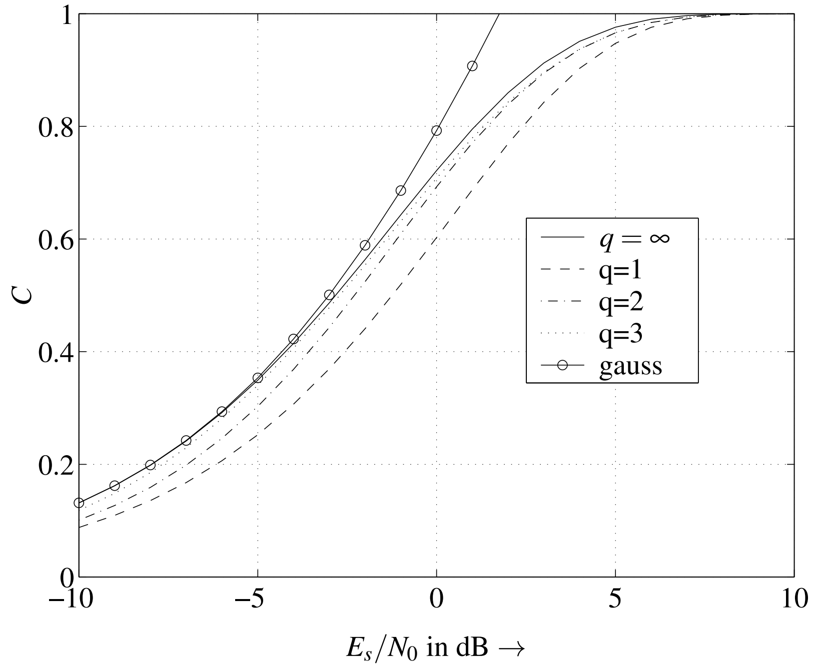


Bild 2.6: Kapazität für den AWGN-Kanal und BPSK-Modulation bei verschiedenen Quantisierungen q

Eine weiterführende Interpretation der Kanalkapazität hinsichtlich mehrstufiger Modulationsverfahren wie M-PSK oder M-QAM ist für den Vorlesungsteil 'Trelliscodierte Modulation' im nächsten Semester von Bedeutung. Hier wird dann ausführlich auf Verfahren zur bandbreiteneffizienten Übertragung eingegangen, die in der digitalen Modemtechnik eine entscheidende Rolle spielen.

2.2.3 Kapazität für bandbegrenzten Gaußkanal mit normalverteiltem Eingang

Bisher wurden nur Kanalmodelle mit diskretem Eingangsalphabet $\mathcal{A}_{in}$ behandelt. Dabei konnte $\mathcal{A}_{out}$ sowohl diskrete Symbole oder aber auch ein Kontinuum an Werten enthalten. Der Vollständigkeit halber sollen in diesem Abschnitt nun auch kontinuierlich verteilte Eingangssignale betrachtet werden. Dabei stellt die additive Überlagerung von gaußverteiltem Rauschen die schlimmste Störung von allen additiven Rauschstörungen dar. Hieraus folgt, dass der AWGN-Kanal für kontinuierliche, gaußverteilte Eingangsalphabete die geringste Kanalkapazität unter diesen Kanälen besitzt. Wir beschränken uns hier auf den eindimensionalen (reellen) Fall.

Die zu berechnenden Entropien gehen aus den Gleichungen 2.3 und 2.9 durch Ersetzen der Summen durch Integrale hervor. Dabei ist zu beachten, dass die sich ergebenden Ausdrücke nicht in der gleichen Art und Weise interpretiert werden können wie im diskreten Fall. Hier werden theoretisch unendlich viele Binärstellen zur Darstellung des Wertebereichs benötigt, wodurch sich eine unendlich große Entropie ergeben müsste. Tatsächlich kann der Ausdruck

$$H(y) = -\int_{-\infty}^{\infty} p_y(\vartheta) \log_2 p_y(\vartheta)\, d\vartheta \tag{2.26}$$

physikalisch nicht interpretiert werden und sogar negative Werte annehmen. $H(y)$ aus Gl. (2.26) wird als *differentielle Entropie* bezeichnet. Mit ihr kann trotzdem eine sinnvolle Definition der Kanalkapazität gefunden werden, welche

$$C = \sup_{p_x(\xi)} \left[H(y) - H(y|x)\right] \tag{2.27}$$

lautet. Für den einfachen Fall eines AWGN-Kanals mit einer konstanten spektralen Leistungsdichte von $N_0/2$ kann die Kanalkapazität leicht berechnet werden. Auf die Lösung der einzelnen Integrale soll an dieser Stelle auf die Übung verwiesen werden. Als Ergebnis erhält man schließlich

$$C = \frac{1}{2} \cdot \log_2 \left(1 + 2\frac{E_s}{N_0}\right) , \tag{2.28}$$

d.h. mit wachsendem Signal-Rausch-Abstand und somit steigender Übertragungsqualität steigt die Kanalkapazität.

Wie schon im letzten Kapitel erwähnt, fügt der Codierer dem Datenstrom Redundanz hinzu, d.h. der codierte Vektor $\mathbf{x}$ enthält mehr Elemente als der Informationsvektor $\mathbf{u}$ ($n > k$). Da hierbei die Energie nicht erhöht wird, besitzt jedes codierte Bit zwangsläufig weniger Energie als ein Informationsbit. Bei der Beurteilung und dem Vergleich von Systemen ist es daher häufig von Interesse, nicht die Energie E_s pro Kanalsymbol zu betrachten, sondern die für jedes Informationsbit aufgebrachte Energie E_b. Beide Größen sind über die Coderate R_c miteinander verbunden, denn es gilt:

$$k \cdot E_b = n \cdot E_s \qquad \Longrightarrow \qquad E_s = \frac{k}{n} \cdot E_b = R_c \cdot E_b \, . \tag{2.29}$$

Anschaulich kann Gl. (2.29) derart interpretiert werden, dass im Mittel pro Informationsbit genau $1/R_c$ codierte Bit übertragen werden, so dass sich die pro Informationsbit übertragene Energie um den Faktor $1/R_c$ erhöht. Mit Gl. (2.29) nimmt der Ausdruck für die Kanalkapazität die Form

$$C = \frac{1}{2} \cdot \log_2 \left(1 + 2R_c\frac{E_b}{N_0}\right) \tag{2.30}$$

an. In Hinblick auf die Bandbreiteneffizienz wird das Ziel einer möglichst effizienten Übertragung genau dann erreicht, wenn die Coderate R_c gleich der Kanalkapazität C ist. Mit Hilfe der Gl. (2.30) kann dann für den AWGN-Kanal der für eine bestimmte Coderate erforderliche Signal-Rausch-Abstand angegeben werden. Die Umstellung von Gl. (2.30) nach E_b/N_0 liefert die Beziehung

$$\frac{E_b}{N_0} = \frac{2^{2R_c} - 1}{2R_c}. \tag{2.31}$$

Bild 2.7 zeigt die graphische Auswertung von Gl. (2.31). Es ist ein annähernd linearer Verlauf zu erkennen. Interessant ist der Grenzwert für $R_c \to 0$. Er kann einfach mit Hilfe der Regel von L'Hospital berechnet werden. Es gilt:

$$\lim_{R_c \to 0} \frac{E_b}{N_0} = \lim_{R_c \to 0} \frac{2^{2R_c} \cdot \ln 2 \cdot 2}{2} = \ln 2 \,\hat{=}\, -1.59\,\text{dB}. \tag{2.32}$$

Dieses Signal-Rausch-Verhältnis stellt die untere Grenze für den AWGN-Kanal dar, bis zu der eine fehlerfreie Übertragung zumindest theoretisch noch möglich ist. Für kleinere Signal-Rausch-Abstände kann eine fehlerfreie Übertragung auch mit noch so großem Aufwand nicht mehr realisiert werden, da die Coderate gegen Null strebt und somit keine Information mehr übertragen wird.

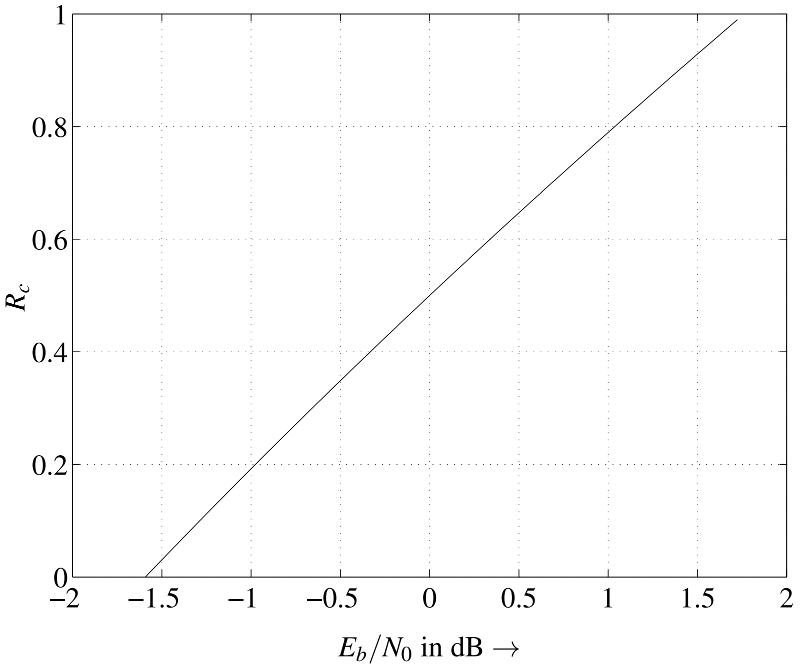


Bild 2.7: Maximale Coderate über E_b/N_0 für den AWGN-Kanal bei kontinuierlichem Eingangssignal

Bandbegrenzung

Abschließend wird nun der auf die Bandbreite B begrenzte zeitkontinuierliche Gaußkanal betrachtet. Bei dieser Bandbreite können entsprechend dem Abtasttheorem nach Shannon bei Einhaltung der ersten Nyquist-Bedingung in einem Zeitraum der Dauer T insgesamt $2BT$ Symbole übertragen werden. Mit der Rauschleistung $N = 2B \cdot N_0/2$ und der Signalleistung $S = R_b \cdot E_b$ (R_b die Datenrate der Informationsbit) lässt sich die Kanalkapazität in die Form

$$\tilde{C} = B \cdot \log_2 \left(1 + \frac{R_b \cdot E_b}{B \cdot N_0} \right) , \tag{2.33}$$

überführen [Fri96]. Die Einheit von $\tilde{C}$ beträgt [Infobit/s]. Es ist zu erkennen, dass die Kapazität $\tilde{C}$ sowohl von der Bandbreite B als auch von dem Signal-Rausch-Verhältnis E_b/N_0 abhängt. In gewissen Grenzen ist also ein Austausch von Bandbreite und Signal-Rausch-Verhältnis möglich. Für den Extremfall $R_b = \tilde{C}$ ergibt sich beim Grenzübergang $\tilde{C}/B \to 0$ für das Signal-Rausch-Verhältnis E_b/N_0 der Grenzwert aus Gl. (2.32).

2.3 Fehlerexponent nach Gallager und R_0-Theorem

Ein gravierender Nachteil der Kanalkapazität nach Shannon besteht darin, dass das Theorem weder eine Aussage über die Struktur des Codes noch über seine Länge enthält. Ferner beschreibt sie nur das asymptotische Verhalten für sehr lange Codes, so dass sie nicht zur Abschätzung einer Wortfehlerrate bei gegebener Wortlänge geeignet ist. Sie stellt lediglich einen theoretischen Grenzwert dar, von dem praktische Verfahren sehr weit entfernt sind.

Mehr Aussagekraft besitzt hingegen der **Fehlerexponent nach Gallager**. An dieser Stelle soll nun keine exakte Herleitung erfolgen, sehr wohl aber der grundsätzliche Weg zur Berechnung des Fehlerexponenten aufgezeigt werden. Wir gehen von einem beliebigen Code der Rate $R_c = k/n$ aus. Die Codierfunktion sei $\mathbf{x} = g(\mathbf{u})$, die zugehörige Decodierfunktion wird mit $\hat{\mathbf{u}} = g^{-1}(\mathbf{y})$ bezeichnet. Als Decodierbereich

$$\mathcal{D}_i = \left\{\mathbf{y} | g^{-1}(\mathbf{y}) = \mathbf{u}^{(i)}\right\} \tag{2.34}$$

verstehen wir die Menge aller Empfangsvektoren $\mathbf{y}$, deren Decodierung das Ergebnis $\hat{\mathbf{u}} = \mathbf{u}^{(i)}$ besitzen. Die Fehlerwahrscheinlichkeit für ein bestimmtes Informationswort $\mathbf{u}^{(i)}$ lautet dann

$$\begin{aligned} P_{w,i} &= P\left(\hat{\mathbf{u}} \neq \mathbf{u}^{(i)} | \mathbf{u} = \mathbf{u}^{(i)}\right) \\ &= P\left(\mathbf{y} \notin \mathcal{D}_i | \mathbf{u} = \mathbf{u}^{(i)}\right) \\ &= \sum_{\mathbf{y} \notin \mathcal{D}_i} P_{\mathbf{Y}|\mathbf{X}}\left(\mathbf{y}|\mathbf{x}^{(i)}\right) \qquad \text{mit} \qquad \mathbf{x}^{(i)} = g\left(\mathbf{u}^{(i)}\right) \end{aligned} \tag{2.35}$$

Es ist nun unter Umständen sehr aufwendig, die Summe über alle Elemente von $\mathcal{D}_i$ zu berechnen, da die Mengen $\mathcal{D}_i$ in n-dimensionalen Räumen sehr komplex sein können und a-priori nicht bekannt sind. Einfacher ist es, die Summe über alle möglichen Empfangsworte $\mathbf{y}$ zu berechnen. Dazu veranschaulichen wir die Strategie der Abschätzung zunächst für den Fall eines Codes mit 2 Codeworten ($k = 1$). Hier gilt

$$P_{w,1} = \sum_{\mathbf{y} \in \mathcal{D}_2} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(1)}) \; . \tag{2.36}$$

Wir führen nun eine Skalierung der Summanden mit dem Faktor

$$\sqrt{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(2)}) / P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(1)})} = \begin{cases} >1 & \mathbf{y} \in \mathcal{D}_2 \\ <1 & \mathbf{y} \in \mathcal{D}_1 \end{cases} \tag{2.37}$$

ein und erhalten aus Gl. (2.36)

$$\begin{aligned} P_{w,1} &\leq \sum_{\mathbf{y} \in \mathcal{D}_2} P_{\mathbf{Y}|\mathbf{X}}\left(\mathbf{y}|\mathbf{x}^{(1)}\right) \cdot \sqrt{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(2)}) / P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(1)})} \\ &= \sum_{\mathbf{y} \in \mathcal{D}_2} \sqrt{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(1)}) \cdot P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(2)})} \\ &\leq \sum_{\mathbf{y}} \sqrt{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(1)}) \cdot P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(2)})} \; . \end{aligned} \tag{2.38}$$

Die Abschätzung für $P_{w,2}$ erfolgt vollkommen äquivalent. Unter Ausnutzung von $P_{\mathbf{X}}(\mathbf{x}^{(1)}) + P_{\mathbf{X}}(\mathbf{x}^{(2)}) = 1$ lässt sich dann die mittlere Fehlerwahrscheinlichkeit wie folgt schreiben

$$\begin{aligned} P_w &= P_{w,1} \cdot P_{\mathbf{X}}(\mathbf{x}^{(1)}) + P_{w,2} \cdot P_{\mathbf{X}}(\mathbf{x}^{(2)}) \\ &\leq \sum_{\mathbf{y}} \sqrt{P_{\mathbf{Y}|\mathbf{X}}\left(\mathbf{y}|\mathbf{x}^{(1)}\right) \cdot P_{\mathbf{Y}|\mathbf{X}}\left(\mathbf{y}|\mathbf{x}^{(2)}\right)} \cdot \left(P_{\mathbf{X}}(\mathbf{x}^{(1)}) + P_{\mathbf{X}}(\mathbf{x}^{(2)})\right) \\ &= \sum_{\mathbf{y}} \sqrt{P_{\mathbf{Y}|\mathbf{X}}\left(\mathbf{y}|\mathbf{x}^{(1)}\right) \cdot P_{\mathbf{Y}|\mathbf{X}}\left(\mathbf{y}|\mathbf{x}^{(2)}\right)} \; . \end{aligned} \tag{2.39}$$

Für den diskreten gedächtnislosen Kanal (DMC) setzt sich die bedingte Wahrscheinlichkeit zweier Vektoren bekanntlich aus dem Produkt der bedingten Wahrscheinlichkeiten der Vektorelemente zusammen ($P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(i)}) = \prod_{j=0}^{n-1} P_{Y|X}(y_j|x_j^{(i)})$, vgl. Gl. (1.23)). Wir erhalten schließlich

$$\begin{aligned} P_w &\leq \sum_{\mathbf{y}} \sqrt{\prod_{j=0}^{n-1} P_{Y|X}\left(y_j|x_j^{(1)}\right) \cdot \prod_{j=0}^{n-1} P_{Y|X}\left(y_j|x_j^{(2)}\right)} \\ &= \sum_{y_1} \cdots \sum_{y_n} \prod_{j=0}^{n-1} \sqrt{P_{Y|X}\left(y_j|x_j^{(1)}\right) \cdot P_{Y|X}\left(y_j|x_j^{(2)}\right)} \\ &= \prod_{j=0}^{n-1} \sum_{y} \sqrt{P_{Y|X}\left(y|x_j^{(1)}\right) \cdot P_{Y|X}\left(y|x_j^{(2)}\right)} \; . \end{aligned} \tag{2.40}$$

Gl. (2.40) wird **Bhattacharyya-Schranke** genannt und gibt die Wortfehlerwahrscheinlichkeit eines Codes bestehend aus 2 Codeworten ($k = 1$) an. Eine Verallgemeinerung auf Codes mit 2^k Codeworten ergibt sich, wenn für alle Codewortpaare ($\mathbf{x}^{(i)}$, $\mathbf{x}^{(l \neq i)}$) ein Gl. (2.37) entsprechender Gewichtungsfaktor eingeführt wird. Diese allgemeine Bhattacharyya-Schranke für einen Code mit 2^k Worten lautet dann

$$P_{w,i} \leq \sum_{\substack{l=1 \\ l \neq i}}^{2^k} \prod_{j=0}^{n-1} \sum_{y} \sqrt{P_{Y|X}(y|x_j^{(i)}) \cdot P_{Y|X}(y|x_j^{(l)})} \, . \tag{2.41}$$

Die Abschätzung der Wortfehlerwahrscheinlichkeit in Gl. (2.41) kann für große Werte von k sehr ungenau werden. Eine Verbesserung wurde von Gallager eingeführt, indem er die Gewichtung der Summanden in Gl. (2.35) verfeinerte. Wird ein bestimmtes Codewort $\mathbf{x}^{(i)}$ gesendet, so gilt im Fall einer Fehlentscheidung ($\mathbf{y} \notin \mathcal{D}_i$) für mindestens ein $l \neq i$

$$P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(l)}) \geq P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(i)}) \qquad \Leftrightarrow \qquad \frac{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(l)})}{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(i)})} \geq 1 \, . \tag{2.42}$$

Da der Quotient in Gl. (2.42) für beliebige l stets größer oder gleich Null ist, ist die Summe über alle Quotienten größer oder gleich Eins. Wir können daher die Ungleichung

$$\left[\sum_{\substack{l=1 \\ l \neq i}}^{2^k} \left(\frac{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(l)})}{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(i)})} \right)^s \right]^\rho \geq 1 \qquad ; \quad s, \rho \geq 0 \tag{2.43}$$

aufstellen. Das Ziel ist es nun, die Parameter s und ρ derart zu wählen, dass die Summe die Ungleichung möglichst knapp erfüllt, also ein Minimum annimmt und somit dicht bei Eins liegt. Dabei sorgt $0 \leq s < 1$ dafür, dass einzelne Quotienten, die deutlich größer Eins sind, verkleinert werden. Der Parameter ρ hat dagegen die Aufgabe, für große Codewortalphabete ($k \gg 1$), wo viele Terme aufaddiert werden, die Gesamtsumme zu reduzieren. Zunächst setzen wir jedoch Gl. (2.43) in Gl. (2.35) ein und erhalten

$$\begin{aligned} P_{w,i} &\leq \sum_{\mathbf{y} \notin \mathcal{D}_i} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(i)}) \cdot \left[\sum_{\substack{l=1 \\ l \neq i}}^{2^k} \left(\frac{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(l)})}{P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(i)})} \right)^s \right]^\rho \\ &= \sum_{\mathbf{y} \notin \mathcal{D}_i} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(i)})^{1-\rho s} \cdot \left[\sum_{\substack{l=1 \\ l \neq i}}^{2^k} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(l)})^s \right]^\rho \end{aligned} \tag{2.44}$$

Im weiteren Verlauf wählte Gallager den Parameter s zu $s = \frac{1}{1+\rho}$. Übernehmen wir diese Wahl und erweitern die Summe in Gl. (2.44) auf alle Vektoren $\mathbf{y}$, so lautet die neue Abschätzung

$$P_{w,i} \leq \sum_{\mathbf{y}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(i)})^{\frac{1}{1+\rho}} \cdot \left[\sum_{\substack{l=1 \\ l \neq i}}^{2^k} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}^{(l)})^{\frac{1}{1+\rho}} \right]^\rho \tag{2.45}$$

Mit der entsprechenden Faktorisierung für den DMC erhalten wir

$$P_{w,i} \leq \prod_{j=0}^{n-1} \sum_{y} P_{Y|X}(y|x_j^{(i)})^{\frac{1}{1+\rho}} \cdot \left[\sum_{\substack{l=1 \\ l \neq i}}^{2^k} P_{Y|X}(y|x_j^{(l)})^{\frac{1}{1+\rho}} \right]^\rho \tag{2.46}$$

Eine direkte Auswertung von Gl. (2.46) für konkrete Codes und Kanäle ist in der Praxis aus Aufwandsgründen nicht möglich. Statt dessen hat man versucht, eine Abschätzung der Fehlerwahrscheinlichkeit für den Mittelwert über alle Codes C zu berechnen. Dabei erwies es sich als vorteilhaft, den Erwartungswert

$$E_C[P_{w,i}] = \sum_{\mathbf{x}'} E_{C \backslash \{\mathbf{x}^{(i)}\}} \left[P_{w,i} | \mathbf{x}^{(i)} = \mathbf{x}' \right] \cdot P_{\mathbf{X}}(\mathbf{x}') \tag{2.47}$$

der bedingten Wahrscheinlichkeit, dass $\mathbf{x}^{(i)}$ ein bestimmtes Codewort $\mathbf{x}'$ ist, zu berechnen. Ohne detailliert auf die weitere Herleitung eingehen zu wollen, soll an dieser Stelle jetzt das Ergebnis vorgestellt werden. Der Erwartungswert ist unabhängig von dem gesendeten Codewort $\mathbf{x}^{(i)}$ und lautet

$$E_C[P_w] = \left(2^k - 1\right)^{\rho} \cdot \sum_{\mathbf{y}} \left(\sum_{\mathbf{x}} P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})^{\frac{1}{1+\rho}} \cdot P_{\mathbf{X}}(\mathbf{x}) \right)^{1+\rho} \tag{2.48}$$

Für den gedächtnislosen, diskreten Kanal setzen wir voraus, dass die Codesymbole x_j statistisch unabhängig voneinander sind und die gleiche Wahrscheinlichkeitsdichtefunktion $P_X(x)$ besitzen, so dass

$$P_{\mathbf{X}}(\mathbf{x}) = \prod_{j=0}^{n-1} P_X(x_j)$$

gilt. Dann geht Gl. (2.48) entsprechend in

$$\begin{aligned} \bar{P}_w = E_c[P_w] &= \underbrace{\left(2^k - 1\right)^{\rho}}_{\approx 2^{k\rho}} \cdot \underbrace{\left[\sum_{y} \left(\sum_{x} P_{Y|X}(y|x)^{\frac{1}{1+\rho}} \cdot P_X(x) \right)^{1+\rho} \right]^n}_{2^{-E_0(\rho,P_X)}} \\ &\approx 2^{-[E_0(\rho,P_X) - \rho k/n] \cdot n} \end{aligned} \tag{2.49}$$

über. Wir definieren nun die **Gallager-Funktion** zu

$$E_0(\rho, P_X) := -\log_2 \sum_{y} \left(\sum_{x} P_{Y|X}(y|x)^{\frac{1}{1+\rho}} \cdot P_X(x) \right)^{1+\rho} \tag{2.50}$$

und erinnern uns daran, dass wir durch die Abschätzung in Gl. (2.44) nur eine obere Schranke für die Wortfehlerrate erhalten. Soll diese mit einer akzeptablen Genauigkeit berechnet werden, muss die Ungleichung (2.44) so knapp wie möglich erfüllt sein. Hieraus folgt direkt die Minimierung der linken Seite von Gl. (2.43) bzw. die Maximierung von Gl. (2.50) bzgl. der Parameter ρ und P_X. Die Maximierung ergibt den sogenannten **Gallager-Exponenten** oder auch **Fehlerexponenten nach Gallager**

$$E_G(R_c) := \max_{P_X} \max_{\rho} \left(E_0(\rho, P_X) - \rho \cdot R_c \right) . \tag{2.51}$$

Gl. (2.51) besagt, dass der Gallager-Exponent nach erfolgter Maximierung nur noch linear von der Coderate R_c abhängt. Wir erhalten also für jedes Paar aus P_X und ρ Geraden mit der Steigung $-\rho$, über denen dann für jede Coderate das Maximum den Fehlerexponenten liefert.

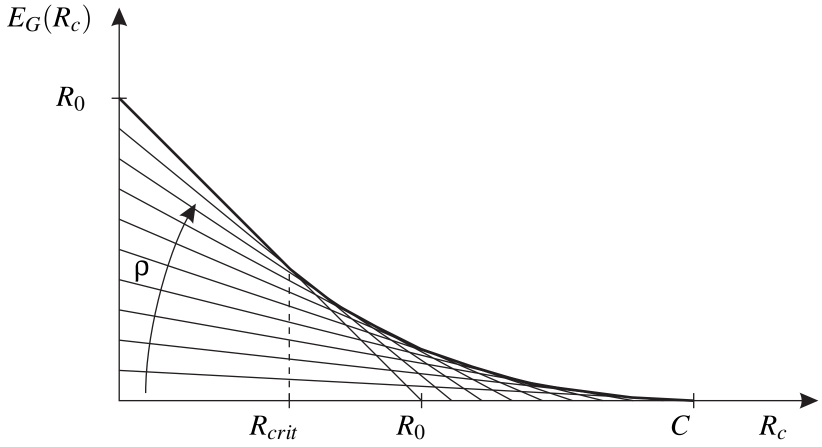


Bild 2.8: Prinzipieller Verlauf des Gallager-Fehlerexponenten $E_G(R_c)$

Dieser Zusammenhang ist in Bild 2.8 illustriert. Bis zur kritischen Grenze R_{crit} wird das Maximum stets für $\rho = 1$ erreicht, welches bei $R_c = 0$ in den sogenannten R_0-Wert

$$R_0 = E_G(0) = \max_{P_X} E_0(1, P_X) \tag{2.52}$$

übergeht. Er wird auch *computational cut off rate* genannt und spielt für die sequentielle Decodierung eine entscheidende Rolle. Für Coderaten $R_c > R_0$ wird diese spezielle Art der Decodierung nämlich sehr aufwändig.

Interessant ist nun die Frage, wie groß die Coderate R_c maximal sein darf, damit die Nebenbedingung $E_G(R_c) \geq 0$ erfüllt ist. Entsprechend Bild 2.8 wird das Maximum offensichtlich für $\rho = 0$ erreicht. Da gleichzeitig auch der Gallager-Exponent gegen Null geht, erhalten wir aus Gl. (2.51) mit Hilfe des Satzes von l'Hospital

$$R_{c\max} = \lim_{\rho \to 0} \frac{E_0(\rho, P_X)}{\rho} = \frac{\partial}{\partial \rho} E_0(\rho, P_X)\bigg|_{\rho=0} = \sum_x \sum_y P_{Y|X}(y|x) \cdot P_X(x) \cdot \log_2 \frac{P_{Y|X}(y|x)}{P_Y(y)} = I(X;Y) \ , \tag{2.53}$$

also gerade die Transinformation. Daraus folgt direkt

$$R_{c\max} = \max_{P_X} I(X;Y) = C \ , \tag{2.54}$$

d.h. die maximale Coderate bei $E_G(R_c) \geq 0$ ergibt gerade die Kanalkapazität C. Mit Hilfe von $E_G(R_c)$ lässt sich weiterhin auch die Wortfehlerwahrscheinlichkeit in Abhängigkeit von der Coderate, der Blocklänge und den Kanaleigenschaften abschätzen.

Definition:

> Es existiert immer ein (n,k)-Blockcode der Rate $R_c = k/n < C$, so dass die Wortfehlerwahrscheinlichkeit durch
>
> $$P_w \leq 2^{-nE_G(R_c)} \tag{2.55}$$
>
> abgegrenzt werden kann. Somit enthält der Fehlerexponent nicht nur eine Aussage über den Kanal (durch die Übergangswahrscheinlichkeiten), sondern auch über die zu erreichende Fehlerrate P_w bei gegebener Blocklänge n.

Wie Bild 2.8 zeigt, nimmt der Fehlerexponent einen Wert größer Null an, solange die Coderate R_c kleiner als die Kanalkapazität C gewählt wird und ist ansonsten Null, d.h. es gilt

$$\begin{aligned} E_G(R_c) &> 0 \quad \text{für} \quad R_c < C \\ E_G(R_c) &= 0 \quad \text{für} \quad R_c \geq C \ . \end{aligned} \tag{2.56}$$

Aus den Gleichungen (2.55) und (2.56) folgt, dass die Wortfehlerwahrscheinlichkeit durch vergrößern der Codewortlänge n für $R_c < C$ beliebig klein gemacht werden kann. Für $R_c \to C$ geht der Gallager-Exponent gegen Null und n muss gegen Unendlich streben, um eine endliche Fehlerwahrscheinlichkeit zu erhalten. Für $\rho = 1$ ergibt sich mit dem R_0-Wert folgende eine Abschätzung für den Fehlerexponenten

$$P_w < 2^{-n(R_0 - R_c)}. \tag{2.57}$$

Aus Gl. (2.57) kann interpretiert werden, dass die Länge n des Blockcodes umso größer sein muss, je dichter R_c an R_0 liegt, damit sich die Fehlerrate P_w nicht verschlechtert. Weiterhin lassen sich folgende drei Bereiche unterteilen.

$0 < R_c < R_0$	:	Die Wortfehlerwahrscheinlichkeit P_w ist begrenzt durch n und die Differenz $R_0 - R_c$. Die Grenze ist berechenbar.
$R_0 < R_c < C$	:	Die Wortfehlerwahrscheinlichkeit P_w ist begrenzt durch n und den Fehlerexponenten $E_G(R_c)$. Die Grenze ist kaum zu berechnen.
$R_c > C$	:	Die Wortfehlerwahrscheinlichkeit P_w kann nicht beliebig klein werden.

Bild 2.9 zeigt noch einmal den Vergleich zwischen dem R_0-Wert und der Kanalkapazität für den BSC. Es ist ersichtlich, dass das R_0-Kriterium schwächer als die Kanalkapazität ist.

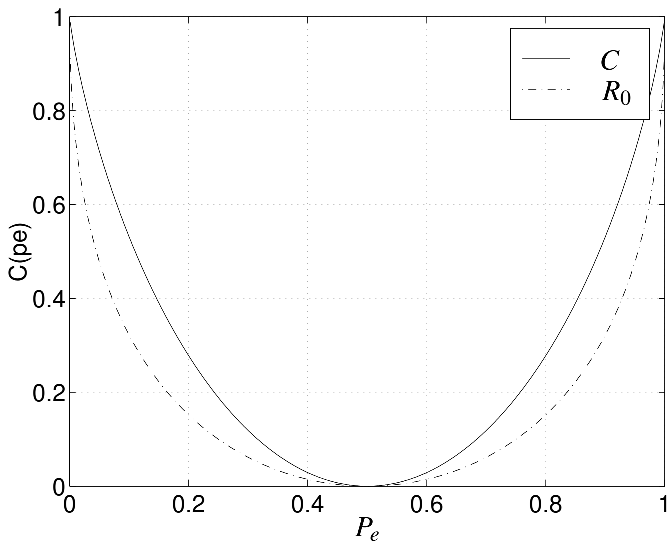


Bild 2.9: Vergleich von Kanalkapazität und Cut-Off-Rate für BSC mit $P(X_0) = P(X_1) = 0.5$

Kapitel 3

Lineare Blockcodes

3.1 Allgemeines, Definitionen

Wie bereits in Abschnitt 1.2.3 erläutert wurde, kann bei Verwendung aller 2^n möglichen Codeworte der Länge n weder ein Fehler erkannt noch korrigiert werden, da jeder Fehler automatisch wieder ein zugelassenes Codewort ergibt. Um eine Fehlerkorrektur bzw. -erkennung durchführen zu können, ist demnach eine Einschränkung des zur Verfügung stehenden Alphabets, d.h. die geschickte Bildung einer Teilmenge, erforderlich. Wie die konkrete Auswahl der Teilmenge zu erfolgen hat, damit der Code eine möglichst grosse Leistungsfähigkeit besitzt, ist kein triviales Problem. Ein Ziel ist mit Sicherheit, die Distanzeigenschaften eines Codes zu maximieren. Ob dabei die kleinste Distanz maximiert werden muss oder aber die Anzahl von Codeworten mit kleiner Distanz zu minimieren ist, scheint nicht von vorn herein klar und hängt von den konkreten Anforderungen an einen Code ab. In der Regel sind keine optimalen Lösungen bekannt. Hinzu kommt die Forderung nach einer möglichst effizienten Form für Codierung und Decodierung, weshalb bei der Konstruktion eines Codes in der Regel auf algebraische Methoden zurückgegriffen wird. In Kapitel 2 wurde allerdings schon erläutert, dass hierdurch nicht unbedingt die besten Codes gefunden werden, wahrscheinlich lassen sich hierdurch sogar nur schlechte Codes finden.

Prinzip der Codierung
Prinzipiell wollen wir zwischen dem **Code** und dem **Codierer** unterscheiden. Unter einem **Blockcode** Γ verstehen wir die Menge aller Vektoren $\mathbf{x} = [x_0\, x_1\, \ldots\, x_{n-1}]$ der Länge n. Dabei kann die Erzeugung der Vektoren durch unterschiedliche **Codierer** erfolgen. Diese Unterscheidung hat weiterreichende Folgen, z.B. hängen die Distanzeigenschaften (Hamming-Distanzen zwischen den Codeworten) und damit auch die Wortfehlerwahrscheinlichkeiten nur vom Code, nicht aber vom Codierer ab. Dies gilt nicht für die Bitfehlerraten, wie später noch gezeigt wird.

Die Klasse der linearen Blockcodes umfasst einen sehr grossen Bereich von Codierungsverfahren. Allgemein ordnet ein $(n,k)_q$-Codierer einem Informationswort $\mathbf{u}$ bestehend aus k Stellen ein Codewort $\mathbf{x}$ mit $n > k$ Stellen zu, wobei jede Stelle q verschiedene Werte annehmen kann. Die $n-k$ zugefügten Stellen werden als Prüfsymbole bezeichnet. Sie enthalten keine neue Information und dienen ausschließlich der Fehlererkennung bzw. -korrektur und stellen deshalb **Redundanz** dar. Der Code ist nun die Gesamtheit aller Codeworte und beschreibt einen Vektorraum Γ der Mächtigkeit q^k, welcher als Unterraum von $\mathrm{GF}(q)^n$ interpretiert werden kann. Da also nicht alle q^n möglichen Elemente von $\mathrm{GF}(q)^n$, sondern nur $q^k < q^n$ Elemente verwendet werden, wurde durch das Hinzufügen der Redundanz eine Untermenge gebildet. Treten während der Übertragung Fehler auf, die zu einem nicht zugelassenen Codewort führen, so können diese Fehler erkannt, vielleicht sogar korrigiert werden.

Allgemeine Codeeigenschaften

- **Systematische Codes:**
  Informationsbit $\mathbf{u}$ explizit im Codewort $\mathbf{x}$ enthalten $\mathbf{x} = [\mathbf{u}\ \mathbf{p}]$
  $\mathbf{p}$ besteht aus $n-k$ angehängten **Prüfsymbolen** und enthält keine neue Information

- **Nicht-systematische Codes:**
  Keine Trennung von Informations- und Prüfbit möglich

- Blockcodes lassen sich immer als systematische Codes darstellen

- **Linearität:**
  Gültigkeit des Superpositionsprinzips, d.h. Linearkombination zweier Codeworte eines linearen Blockcodes ergibt wiederum ein Codewort
  Beachte: Addition und Multiplikation finden innerhalb eines Vektorraums $\mathrm{GF}(q)^n$ der Basis q und der Dimension n statt

3.2 Restklassenarithmetik

Wie im letzten Abschnitt erwähnt wurde, können die einzelnen Stellen u_i des Informationswortes $\mathbf{u}$ und auch die Stellen x_i des Codewortes $\mathbf{x}$ q verschiedene Werte annehmen. Aus diesem Grund sprechen wir hier noch nicht von Binärstellen, sondern allgemein von Symbolen. Später beschränken wir uns dann auf den binären Fall mit $q = 2$. Alle mathematischen Operationen wie Additionen oder Multiplikationen werden modulo-q ausgeführt, d.h. sie finden in einem endlichen Feld $\mathrm{GF}(q)$ zur Basis q statt. Um Verwechslungen zu vermeiden, definiert dieser Abschnitt eindeutig die Begriffe Gruppe, Ring, Körper, Galoisfeld und Vektorraum.

3.2.1 Gruppen, Ringe, Körper, Galoisfelder und Vektorräume

Definition einer Gruppe

Eine Gruppe $(\mathcal{G},*)$ besteht aus einer Menge $\mathcal{G}$ und einer zwischen den Elementen von $\mathcal{G}$ definierten Operation $*$ und erfüllt folgende Bedingungen:

1. *Für alle $a,b \in \mathcal{G}$ gilt $a*b \in \mathcal{G}$ (Abgeschlossenheit).*
2. *Für alle $a,b,c \in \mathcal{G}$ gilt $(a*b)*c = a*(b*c)$ (Assoziativgesetz).*
3. *Es existiert ein* neutrales Element *e, für das gilt: $a*e = a$.*
4. *Für jedes $a \in \mathcal{G}$ existiert ein inverses Element a^{-1}, so dass gilt: $a*a^{-1} = e$.*

Eine Gruppe wird als abelsche oder kommutative Gruppe bezeichnet, wenn zusätzlich das Kommutativgesetz gilt:

5. *Für alle $a,b \in \mathcal{G}$ gilt: $a*b = b*a$.*

Die Operation $*$ in der obigen Definition kann unter anderem die Addition oder auch die Multiplikation darstellen. Beispiele für Gruppen sind $(\mathbb{Z},+)$, $(\mathbb{R},+)$, $(\mathbb{R}\setminus\{0\},\cdot)$.

Definition eines Rings

Ein Ring $(\mathcal{R},+,\cdot)$ besteht aus einer Menge $\mathcal{R}$, für deren Elemente die Operationen Addition und Multiplikation definiert sind und die folgende Bedingungen erfüllt:

1. *$(\mathcal{R},+)$ ist eine kommutative Gruppe*
2. *Abgeschlossenheit der Multiplikation $\forall_{a,b\in\mathcal{R}} \quad a\cdot b\in\mathcal{R}$*
3. *Für alle $a,b,c \in \mathcal{R}$ gilt: $a\cdot(b+c) = a\cdot b + a\cdot c$ (Distributivgesetz)*
4. *Für alle $a,b,c \in \mathcal{R}$ gilt: $a\cdot(b\cdot c) = (a\cdot b)\cdot c$ (Assoziativgesetz)*

Das Rechnen modulo-q genügt den Anforderungen eines Ringes. Es ist zu beachten, dass nicht notwendigerweise ein inverses Element $a^{-1}\in\mathcal{R}$ zu jedem $a\in\mathcal{R}$ mit $a\cdot a^{-1}=1$ existieren muss.

Definition eines Körpers (*field*)

Ein Körper $(\mathcal{K},+,\cdot)$ besteht aus einer Menge $\mathcal{K}$, für deren Elemente die Operationen Addition und Multiplikation definiert sind und die folgende Bedingungen erfüllt:

1. *$(\mathcal{K},+)$ ist eine kommutative Gruppe*
2. *$(\mathcal{K}\setminus\{0\},\cdot)$ ist eine kommutative Gruppe (multiplikative Gruppe)*
3. *Für alle $a,b,c \in \mathcal{K}$ gilt: $a\cdot(b+c) = a\cdot b + a\cdot c$ (Distributivgesetz)*

Definition eines Galoisfelds (*finite field*)

*Enthält die Menge $\mathcal{K}$ nur endlich viele (q) Elemente, so sprechen wir von einem endlichen Körper (*finite field*) oder auch von einem Galoisfeld $\mathrm{GF}(q)$ zur Basis q. Galois-Felder existieren nur für $q=p^m$, wobei p eine Primzahl und m eine natürliche Zahl ist. Für ein Galoisfeld gelten ansonsten die gleichen Regeln wie für Körper. Ist $q=p$ eine Primzahl, sprechen wir auch von **Primkörpern**!*

Gruppen oder Körper werden als zyklisch bezeichnet, wenn sich alle Elemente außer dem Nullelement durch Potenzen eines Elementes z bilden lassen. Dieses eine Element heißt **primitives Element**. Für jedes Galoisfeld existiert ein primitives Element z und es gilt:

$$\mathrm{GF}(p) = \{z^i \,|\, z\in\mathrm{GF}(p) \ \wedge \ i=1,\ldots,p-1\} \tag{3.1}$$

Alle Potenzen von z sind automatisch ebenfalls primitive Elemente des $\mathrm{GF}(p)$.

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

$\cdot$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Tabelle 3.1: Verknüpfungstabellen für das Galois-Feld GF(5)

Als Beispiel enthält Tabelle 3.1 die additive und die multiplikative Verknüpfung für ein Galois-Feld zur Basis $q=p=5$. Bezüglich der Addition stellt '0' das neutrale Element dar, denn es gilt $a+0=a$. Für die inversen Elemente gilt $a+(-a)=0$, also z.B. $2+3=0$. Für die Multiplikation gilt unterdessen hinsichtlich des neutralen Elementes $a\cdot 1=a$ und hinsichtlich der inversen Elemente $a\cdot a^{-1}=1$ (siehe auch Tabelle 3.1). Das primitive Element des GF(5) ist $z=2$, denn es gilt

$$\mathrm{GF}(5)\setminus\{0\} = \{1\ 2\ 3\ 4\} = \{2^4\ 2^1\ 2^3\ 2^2\}$$

Ein Codewort $\mathbf{x}$ setzt sich nun aus n Symbolen zusammen, die jeweils Elemente eines Galoisfeldes der Basis q sind $x_i \in \mathrm{GF}(q)$. Die Menge aller n-Tupel bildet somit einen Vektorraum $\mathrm{GF}(q)^n$, der wie folgt definiert ist:

Definition eines Vektorraums (Linearer Raum)

Ein Vektorraum $\mathcal{V}$ über einem Körper $\mathcal{K}$ beschreibt eine Menge von Vektoren, für die eine Addition und eine Skalarmultiplikation definiert sind. $\mathcal{V}$ ist bzgl. dieser beiden Operationen abgeschlossen, d.h. für $\mathbf{a},\mathbf{b} \in \mathcal{V}$ und $\alpha \in \mathcal{K}$ gilt $\mathbf{a}+\mathbf{b} \in \mathcal{V}$ und $\alpha \cdot \mathbf{a} \in \mathcal{V}$. Ferner müssen folgende Gesetze erfüllt sein:

1. $\mathbf{a}+\mathbf{b} = \mathbf{b}+\mathbf{a}$
2. $(\mathbf{a}+\mathbf{b})+\mathbf{c} = \mathbf{a}+(\mathbf{b}+\mathbf{c})$
3. $\mathbf{a}+\mathbf{0} = \mathbf{a}$
4. $\mathbf{a}+(-\mathbf{a}) = \mathbf{0}$
5. $\alpha \cdot (\mathbf{a}+\mathbf{b}) = \alpha\mathbf{a}+\alpha\mathbf{b}$
6. $(\alpha+\beta)\cdot\mathbf{a} = \alpha\mathbf{a}+\beta\mathbf{a}$
7. $(\alpha\cdot\beta)\cdot\mathbf{a} = \alpha\cdot(\beta\cdot\mathbf{a})$
8. $1\cdot\mathbf{a} = \mathbf{a}$

Mathematische Operationen zwischen den Vektoren $\mathbf{a},\mathbf{b} \in \mathcal{V}$ sind komponentenweise definiert

$$\mathbf{x}+\mathbf{y} = [x_0+y_0 \;\; x_1+y_1 \;\; \ldots \;\; x_{n-1}+y_{n-1}] \, .$$

3.2.2 Erweiterungskörper und Polynomdarstellung

Für viele Anwendungen ist es vorteilhaft, nicht nur Felder zur Basis einer Primzahl p zu betrachten. So sind in der heutigen Digitaltechnik in der Regel Dualzahlensysteme anzutreffen, die sich mit Primzahlen schlecht darstellen lassen. Zur Veranschaulichung der Problematik untersuchen wir das Galois-Feld mit $q = 4 = 2^{m=2}$ in Tabelle 3.2.

$+$	0	1	z	$1+z$
0	0	1	z	$1+z$
1	1	0	$1+z$	z
z	z	$1+z$	0	1
$1+z$	$1+z$	z	1	0

$\cdot$	0	1	z	$1+z$
0	0	0	0	0
1	0	1	z	$1+z$
z	0	z	$1+z$	1
$1+z$	0	$1+z$	1	z

Tabelle 3.2: Verknüpfungstabellen für das Galois-Feld GF(4)

- Für $q = p^{m=1}$ besteht $\mathrm{GF}(q)$ aus den natürlichen Zahlen von 0 bis $q-1$
- Dies gilt für $m \geq 2$ nicht mehr

$\rightarrow$ Abstrakte Größe z einführen

$\rightarrow$ $\mathrm{GF}(4) = \{0,\, 1,\, z,\, 1+z\}$, mit $z^2+z+1 = 0$
(Für $z^2+1 = 0$ würde zu $1+z$ kein multiplikatives Inverses existieren)

Durch die Verwendung der abstrakten Größe z bietet sich die Polynomdarstellung an. Daher betrachten wir im Folgenden Polynome $p(D) = p_0 + p_1 D + \cdots + p_m D^m$ vom Grad m mit Koeffizienten aus $\mathrm{GF}(p)$.

Definition irreduzibles Polynom:

> *Ein Polynom $p(D)$ vom Grad m mit Koeffizienten $p_i \in \mathrm{GF}(p)$ heißt irreduzibel, wenn es sich nicht in Polynome vom Grad $< m$ (mit Koeffizienten aus $\mathrm{GF}(p)$) faktorisieren lässt. Folglich besitzt $p(D)$ keine Nullstellen im $\mathrm{GF}(p)$ (**das Fehlen von Nullstellen allein garantiert nicht die Irreduzibilität eines Polynoms!**). Der Begriff irreduzibel bezieht sich immer auf einen bestimmten Körper.*

Beispiel:
Polynom $p(D) = D^2 + D + 1$ hat keine Nullstellen im $\mathrm{GF}(2)$, da $p(0) \neq 0$ und $p(1) \neq 0$. Allerdings aber in $\mathrm{GF}(4)$ des obigen Beispiels existiert die abstrakte Nullstelle z

Beispiel:
Das Polynom $p(D) = D^4 + D^2 + 1$ ist nicht irreduzibel (d.h. es ist reduzibel), da $D^4 + D^2 + 1 = (D^2 + D + 1)^2$ gilt. Es existiert jedoch keine Nullstelle im $\mathrm{GF}(2)$, da $p(0) \neq 0$ und $p(1) \neq 0$ gilt.

Satz:

> *Für jede Primzahl p und jede natürliche Zahl m existiert ein irreduzibles Polynom $p(D)$ vom Grad m mit Koeffizienten aus $\mathrm{GF}(p)$.*

Die Nullstelle bzw. Wurzel z eines Polynoms $p(D)$ hat die Eigenschaft

$$p(z) = 0$$

und korrespondiert immer mit einem konkreten Körper.

Definition primitives Polynom

> *Zu jeder Primzahl p und jedem $m \in \mathbb{N}$ existiert ein irreduzibles Polynom $p(D)$ über $\mathrm{GF}(p)$ mit folgender Eigenschaft:*
> *Für jede Nullstelle $p(z) = 0$ sind $z^1, \cdots, z^n$ verschieden voneinander, wobei $z^0 = z^n = 1$ mit $n = p^m - 1$ gilt. z heißt **primitives Element** von $\mathrm{GF}(p^m)$ und $p(D)$ **primitives Polynom**.*

Damit lässt sich der Erweiterungskörper $\mathrm{GF}(p^m)$ folgendermaßen definieren:

Definition Erweiterungskörper (*extension field*)

> *Es sei $p(D)$ ein primitives Polynom vom Grad m mit Koeffizienten $p_i \in \mathrm{GF}(p)$ und $z \notin \mathrm{GF}(p)$ das primitive Element von $p(D)$. Dann wird der Erweiterungskörper $\mathrm{GF}(p^m)$ durch alle Linearkombinationen der Potenzen $z^0, \ldots, z^{m-1}$ aufgespannt (zyklischer Körper). Außerdem lassen sich alle Elemente von $\mathrm{GF}(p^m)$ als Linearkombinationen der Potenzen von z darstellen. Es gilt:*

$$\begin{aligned}\mathrm{GF}(p^m) &= \left\{ \sum_{i=0}^{m-1} p_i \cdot z^i \mid p_0, \ldots p_{m-1} \in \mathrm{GF}(p) \right\} \\ &= \left\{ 0, z^1, z^2, \ldots, z^{n-1}, z^n = z^0 = 1 \right\}\end{aligned}$$

> *Somit ist $\mathrm{GF}(q) = \mathrm{GF}(p^m)$ ein Erweiterungskörper von $\mathrm{GF}(p)$ mit $q = p^m$ Elementen. Man nennt $\mathrm{GF}(p)$ auch Primkörper von $\mathrm{GF}(q)$.*

Anschaulich kann man sich den Erweiterungskörper zum einen als Menge aller Potenzen des primitiven Elementes von $p(D)$ vorstellen (**Exponentendarstellung**). Zum anderen kann GF(q) auch als Menge aller Polynome $p(z)$ vom Grad $\leq m-1$ interpretiert werden (**Komponentendarstellung**). Erweiterungskörper sind insbesondere für die Reed-Solomon-Codes von großer Bedeutung, da diese nicht-binäre Codes sind, die in der Regel auf einem Galoisfeld zur Basis 2^m basieren.

3.3 Distanzeigenschaften von Blockcodes

3.3.1 Minimaldistanz

Ein wichtiges Maß zur Beurteilung der Leistungsfähigkeit eines Codes ist die **Hamming-Distanz**. Die Hamming-Distanz $d_{\mathrm{H}}(\mathbf{a},\mathbf{b})$ gibt die Anzahl unterschiedlicher Symbole zwischen zwei Codeworten **a** und **b** an. Aufgrund der Linearität der hier betrachteten Codierungsvorschriften werden häufig alle Codeworte mit dem Nullwort verglichen. Dann ist das **Hamming-Gewicht** $w_{\mathrm{H}}(\mathbf{a})$ eines Codewortes **a** von Interesse, welches die Anzahl der von Null verschiedenen Elemente in **a** angibt. Die kleinste vorkommende Hamming-Distanz

$$d_{\min} = \min_{\mathbf{a},\mathbf{b}\in\Gamma,\, \mathbf{a}\neq\mathbf{b}} d_{\mathrm{H}}(\mathbf{a},\mathbf{b}) \tag{3.2}$$

bestimmt im wesentlichen die Leistungsfähigkeit des Codes. Je größer sie ist, desto mehr Fehler können erkannt bzw. korrigiert werden. Bild 3.1 veranschaulicht den Zusammenhang von $d_{\min}$ mit der Anzahl korrigierbarer bzw. erkennbarer Fehler.

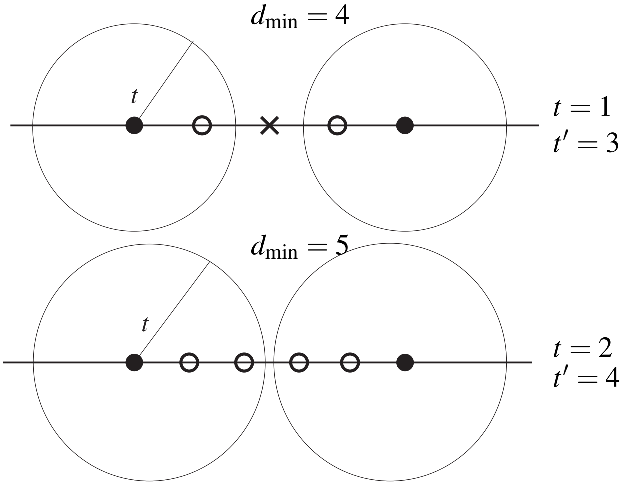


Bild 3.1: Veranschaulichung der Korrekturfähigkeit in Abhängigkeit der Hamming-Distanzen

Fehlerkorrektur und Fehlererkennung

- Massive Punkte stellten gültige Codeworte dar
- Transparenten Kreise repräsentieren korrigierbare Empfangsworte
- **x** beschreibt ein nicht korrigierbares Wort
  (es hat gleichen Abstand zu beiden Codeworten $\rightarrow$ eine Entscheidung wäre rein zufällig

$\rightarrow$ **Anzahl korrigierbarer Fehler:**

$$t = \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor \tag{3.3}$$

$\rightarrow$ **Anzahl erkennbarer Fehler:**

$$t' = d_{\min} - 1 \tag{3.4}$$

- Code kann unter Umständen auch noch darüberhinaus Fehler erkennen oder korrigieren, dies kann aber nicht garantiert werden.

- Gleichzeitige Korrektur von t Fehlern und Erkennung von $t' > t$ Fehlern

$$t + t' + 1 \leq d_{\min}. \tag{3.5}$$

Aufgrund der Wichtigkeit der minimalen Hamming-Distanz $d_{\min}$ wird sie in der Literatur auch häufig bei der Benennung eines konkreten Blockcodes angegeben, z.B. $(n,k,d_{\min})_q$-Code.

Die bisherigen Erläuterungen verdeutlichen die Relevanz der Minimaldistanz zur Beurteilung der Leistungsfähigkeit von Codes zur Kanalcodierung. Um eine Vorstellung zu bekommen, welche Minimaldistanz für eine bestimmte Parameterwahl von k, n und q überhaupt erreicht werden kann, sollen im Folgenden einige Schranken vorgestellt werden (**ohne Beweis**).

Singleton-Schranke

- Für einen $(n,k,d_{\min})_q$-Code gilt stets

$$d_{\min} \leq n - k + 1 \tag{3.6}$$

- Erklärung: Alle Codewörter unterscheiden sich an mindestens $d_{\min}$ Stellen.
  Werden bei allen Codewörtern die erste $d_{\min} - 1$ Stellen gestrichen, so sind die gekürzten Codewörter der Länge $n - d_{\min} + 1$ immer noch alle verschieden. Es gibt daher q^k verschiedene gekürzte Codewörter im Raum der $q^{n-d_{\min}+1}$ gekürzten Wörter. Dies ist nur möglich, wenn $k \leq n - d_{\min} + 1$ gilt.

- Minimaldistanz ist immer kleiner gleich der Anzahl der Prüfstellen plus Eins

- Bei Gleichheit ergibt sich **MDS (*Maximum Distance Separable*)-Code**

- Für binäre Codes sind nur einfache $(n,1,n)_2$-Wiederholungscodes (*Repetition Codes*) MDS-Codes.

- Die in Abschnitt 3.7.3 definierten nicht-binären Reed-Solomon-Codes sind MDS-Codes!

Hamming-Schranke, *sphere-packing bound*

- Für einen $(n,k,d_{\min})_q$-Code, der t Fehler korrigieren soll, gilt stets:

$$q^{n-k} \geq \sum_{r=0}^{t} \binom{n}{r} \cdot (q-1)^r \tag{3.7}$$

- Erklärung von Gl. (3.7):
  - Mit $n-k$ Prüfsymbolen lassen sich genau q^{n-k} verschiedene Vektoren (**Syndrome**) erzeugen
  - Zur Korrektur jedes Fehlerwortes $\mathbf{e}$ mit Gewicht $w_{\mathrm{H}}(\mathbf{e}) \leq t$ (maximal t Fehler) muss jedem $\mathbf{e}$ mindestens eins der q^{n-k} Syndrome zugeordnet werden
  - Anzahl möglicher Fehlerworte: rechte Seite von Gl. (3.7).

  $\rightarrow$ Gl. (3.7) besagt: Zur Korrektur von t Fehlern muss Anzahl Syndrome immer größer als Zahl der Fehlermuster sein

- Bei Gleichheit: **perfekter Code**

- Zu jedem Fehlermuster existiert genau ein Syndrom, d.h. es gibt exakt nur so viel Syndrome wie zur Korrektur von t Fehlern erforderlich
- Anschaulich: Verteilung aller q^n Elemente des Vektorraums $\mathrm{GF}(q)^n$ auf **disjunkte** Decodierkugeln, d.h. Kugeln sind so dicht gepackt, dass sie den gesamten Vektorraum enthalten.
- Beispiele für perfekte Codes: $(7,4,3)_2$-Hamming-Code, (23,12,7)-Golay-Code

Plotkin-Schranke

- Für einen $(n,k,d_{\min})_q$-Blockcode gilt stets:

$$d_{\min} \leq \frac{n(q-1)q^{k-1}}{q^k-1} \approx \frac{n(q-1)}{q} \tag{3.8}$$

- Herleitung:
  - Mittleres Gewicht eines Symbols im $\mathrm{GF}(q)$ beträgt $(q-1)/q$
  - Für ganzes Codewort der Länge n gilt für mittleres Gewicht $n(q-1)/q$
  - Ausschluss des Nullworts: Erhöhung des mittleren Gewicht um Faktor $q^k/(q^k-1)$
  - Insgesamt ergibt sich rechter Ausdruck in Gl. (3.8)
  - Mittleres Gewicht immer größer oder mindestens gleich minimalem Gewicht, qed.

Gilbert-Varshamov-Schranke

- Bisherige Schranken erlauben Abschätzung der Minimaldistanz
- Sie geben keine Garantie für die Existenz eines realen Codes
- Gilbert-Varshamov-Schranke beweist Existenz eines realen Codes (ohne Konstruktionsvorschrift)

- Es existiert immer ein $(n,k,d_{\min})_q$-Code, wenn gilt:

$$q^{n-k} > \sum_{r=0}^{d_{\min}-2} \binom{n-1}{r} (q-1)^r \tag{3.9}$$

3.3.2 Distanzspektrum und IOWEF

Um die Leistungsfähigkeit eines konkreten Codes beurteilen zu können, wird in der Regel die zu erreichende Wortfehlerrate, also die Wahrscheinlichkeit für das Auftreten eines Decodierfehlers, in Abhängigkeit bestimmter Kanaleigenschaften (z.B. Übergangswahrscheinlichkeit beim DMC, Signal-Rausch-Verhältnis beim AWGN-Kanal) abgeschätzt. Häufig reicht dazu nicht mehr alleine die Minimaldistanz des Codes aus, sondern es muss sein gesamtes Distanzspektrum, d.h. seine Gewichtsverteilung bekannt sein. Aufgrund der oben erläuterten Linearität ist es erlaubt, anstatt der Hamming-Distanzen zwischen allen möglichen Codeworten nur die Distanzen zum Nullwort zu betrachten, so dass direkt die Hamming-Gewichte $w_\mathrm{H}(\mathbf{a})$ der Codeworte ausschlaggebend ist. Diese Tatsache verringert den Aufwand bei der Bestimmung der Distanzeigenschaften enorm. Das Distanzspektrum beschreibt also die Anzahl von Codeworten mit einem bestimmten Gewicht. Es kann mathematisch in der Form

$$A(D) = \sum_{d=0}^{n} A_d D^d = \sum_{\mathbf{x}\in\Gamma} D^{w_\mathrm{H}(\mathbf{x})} = 1 + \sum_{d=d_{\min}}^{n} A_d D^d \tag{3.10}$$

angegeben werden, wobei D nur ein Platzhalter ist. Die Koeffizienten A_d in Gl. (3.10) geben die Anzahl der Codeworte $\mathbf{a}$ mit dem Gewicht $w_\mathrm{H}(\mathbf{a}) = d$ an. Für sie gilt der Zusammenhang

$$\sum_{d=0}^{n} A_d = q^k,$$

d.h. die Summe aller Koeffizienten entspricht logischerweise der Gesamtzahl aller Codeworte.

Das Distanzspektrums hängt ausschließlich vom Code, nicht aber vom Codierer (Abbildung Infowort auf Codewort), ab und erlaubt die Abschätzung der Auftrittswahrscheinlichkeit eines Decodierfehlers.

Zur Berechnung der **Bitfehlerrate**, d.h. dem Verhältnis der fehlerhaften Informationsbit zur Gesamtzahl übertragener Informationsbit, ist die Kenntnis des Distanzspektrums unzureichend. Vielmehr muss bekannt sein, wieviele Informationsbit beim Verwechseln zweier Codeworte mit der Hamming-Distanz $d_\mathrm{H}(\mathbf{a},\mathbf{b})$ verfälscht werden. Dazu dient die sogenannte **IOWEF** (*Input Output Weight Enumerating Function*), die die Verbindung zwischen Eingangsvektoren $\mathbf{u}$ und Ausgangsvektoren $\mathbf{x}$ eines konkreten Codierers berücksichtigt. Sie definiert sich als

$$A(W,D) = \sum_{w=0}^{k} \sum_{d=0}^{n} A_{w,d} \cdot W^w D^d \ , \tag{3.11}$$

wobei $A_{w,d}$ die Anzahl der Codeworte mit einem Eingangsgewicht von w und einem Ausgangsgewicht von d beschreibt. Anders als die Wortfehlerwahrscheinlichkeit beeinflusst der Codierer sehr wohl die Bitfehlerwahrscheinlichkeit. Distanzspektrum und IOWEF werden nun anhand eines kleinen Beispiels noch einmal veranschaulicht.

Beispiel: Systematischer $(7,4)_2$-Hamming-Code (Definition in Abschnitt 3.5.8)

- $(7,4)_2$-Hamming-Code setzt sich aus $2^4 = 16$ Codeworten zusammen (s. Tabelle)
- Zweite Spalte enthält Gewicht der Infoworte $\mathbf{u}$
- Dritte Spalte enthält Gewicht der zugehörigen Codeworte $\mathbf{x}$

Codeworte $\mathbf{x}$	Gewicht $w_\mathrm{H}(\mathbf{u})$	Gewicht $w_\mathrm{H}(\mathbf{x})$
0 0 0 0 0 0 0	0	0
0 0 0 1 1 1 1	1	4
0 0 1 0 1 1 0	1	3
0 0 1 1 0 0 1	2	3
0 1 0 0 1 0 1	1	3
0 1 0 1 0 1 0	2	3
0 1 1 0 0 1 1	2	4
0 1 1 1 1 0 0	3	4
1 0 0 0 0 1 1	1	3
1 0 0 1 1 0 0	2	3
1 0 1 0 1 0 1	2	4
1 0 1 1 0 1 0	3	4
1 1 0 0 1 1 0	2	4
1 1 0 1 0 0 1	3	4
1 1 1 0 0 0 0	3	3
1 1 1 1 1 1 1	4	7

Aus obiger Tabelle wird ersichtlich, dass je 1 Codewort mit den Gewichten $w_\mathrm{H}(\mathbf{x}) = 0$ und $w_\mathrm{H}(\mathbf{x}) = 7$ und je 7 Codeworte mit den Gewichten $w_\mathrm{H}(\mathbf{x}) = 3$ und $w_\mathrm{H}(\mathbf{x}) = 4$ existieren.

Gewicht d	A_d
0	1
1	0
2	0
3	7
4	7
5	0
6	0
7	1

IOWEF	$d=0$	$d=1$	$d=2$	$d=3$	$d=4$	$d=5$	$d=6$	$d=7$
$w=0$	1	0	0	0	0	0	0	0
$w=1$	0	0	0	3	1	0	0	0
$w=2$	0	0	0	3	3	0	0	0
$w=3$	0	0	0	1	3	0	0	0
$w=4$	0	0	0	0	0	0	0	1

Das Distanzspektrum hat demnach die Form

$$A(D) = 1 + 7 \cdot D^3 + 7 \cdot D^4 + D^7 \ ,$$

die Minimaldistanz nimmt den Wert $d_{\min} = 3$ an und die IOWEF lautet

$$A(W,D) = 1 + 3 \cdot WD^3 + 3 \cdot W^2D^3 + W^3D^3 + WD^4 + 3 \cdot W^2D^4 + 3 \cdot W^3D^4 + W^4D^7 \ .$$

3.4 Decodierprinzipien und Wortfehlerwahrscheinlichkeit

3.4.1 Grundprinzipien der Decodierung

Bezüglich der prinzipiellen Decodierung von Blockcodes sind im wesentlichen drei Decodierprinzipien zu nennen, die in Bild 3.2 graphisch dargestellt und im Folgenden kurz erläutert werden. MAP- und Maximum Likelihood-Decodierung wurden zu einer Kategorie zusammengefasst.

Maximum-a-posteriori-Kriterium (MAP)
Die optimale Decodierung garantiert das MAP-Kriterium (*maximum a posteriori probability*). Hierbei wird das Codewort **a** bestimmt, welches die a-posteriori-Wahrscheinlichkeit $P(\mathbf{a}|\mathbf{y})$ maximiert. Es gilt

$$\begin{aligned} \hat{\mathbf{x}} &= \arg\max_{\mathbf{a}\in\Gamma} P(\mathbf{a}|\mathbf{y}) = \arg\max_{\mathbf{a}\in\Gamma} P(\mathbf{y}|\mathbf{a}) \cdot \frac{P(\mathbf{a})}{P(\mathbf{y})} \\ &= \arg\max_{\mathbf{a}\in\Gamma} P(\mathbf{y}|\mathbf{a}) \cdot P(\mathbf{a}) \ . \end{aligned} \tag{3.12}$$

Gleichung (3.12) zeigt, dass neben den Übergangswahrscheinlichkeiten $P(\mathbf{y}|\mathbf{a})$ des Kanals auch die a-priori-Wahrscheinlichkeiten $P(\mathbf{a})$ in die Decodierentscheidung eingehen. Steht dem Decodierer also die Statistik des Codealphabets a-priori zur Verfügung, kann er sie gewinnbringend nutzen. Die MAP-Decodierung ist dann die optimale Form der Decodierung. Stellt $\mathbf{x} = f(\mathbf{u})$ die Abbildung des Codierers dar, erhalten wir die geschätzten Informationsbit durch $\hat{\mathbf{u}} = f^{-1}(\hat{\mathbf{x}})$.

Maximum-Likelihood-Decodierung (MLD)
Ist die Verteilung der Codeworte gleichverteilt oder aber nicht von vornherein bekannt, entfällt der Term $P(\mathbf{a})$ in Gl. (3.12). Wir erhalten dann die sogenannte *Maximum-Likelihood*-Decodierung

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{a}\in\Gamma} P(\mathbf{y}|\mathbf{a}) \ , \tag{3.13}$$

die von allen Codeworten dasjenige auswählt, das die geringste Distanz zum empfangenen Wort hat. Hierdurch wird bei gleicher a-priori-Wahrscheinlichkeit der Codeworte eine minimale Wortfehlerwahrscheinlichkeit erzielt. Allerdings nimmt der Aufwand wie auch beim MAP-Kriterium mit wachsender Anzahl k exponentiell zu, so dass ein einfacher Vergleich aller Codeworte mit dem empfangenen Wort schnell unpraktikabel wird. Dies gilt umso mehr, da die Leistungsfähigkeit der Codes mit steigendem k und n zunimmt und daher eine hohe Leistungsfähigkeit auch mit einem hohen Decodieraufwand verbunden ist. Da bei diesen beiden Verfahren stets der nächste Nachbar zum empfangenen Wort gesucht wird, können unter Umständen auch mehr als $\lfloor (d_{\min} - 1)/2 \rfloor$ Fehler korrigiert werden. Dies gilt für die Folgenden Verfahren nicht.

Begrenzte Distanz-Decodierung (BDD)
Beim BDD wird um jedes Codewort eine Kugel vom Radius t gelegt. Alle Empfangsworte, die ausschließlich innerhalb einer einzigen Kugel liegen, werden decodiert, d.h. dem mit ihrer Kugel korrespondierendem Codewort zugeordnet. Empfangsworte, die in mehreren oder aber gar keiner Kugel liegen, werden nicht decodiert, sondern als fehlerhaft gekennzeichnet. Der BDD ist nur unwesentlich schlechter als der MLD.

Begrenzte Minimaldistanz-Decodierung (BMD)
Um jedes Codewort wird eine Kugel mit dem Radius $t = \lfloor (d_{\min} - 1)/2 \rfloor$ gelegt, wodurch alle Kugeln disjunkt sind, also keine gemeinsamen Bereiche besitzen. Alle Empfangsworte innerhalb einer Kugel werden dann Richtung Kugelmittelpunkt decodiert; es ergibt sich also eine Decodierung bis zur halben Minimaldistanz. Die BMD besitzt von den drei hier vorgestellten Prinzipien die geringste Leistungsfähigkeit, für *perfekte Codes* ist sie allerdings mit der *Maximum Likelihood*-Decodierung identisch.

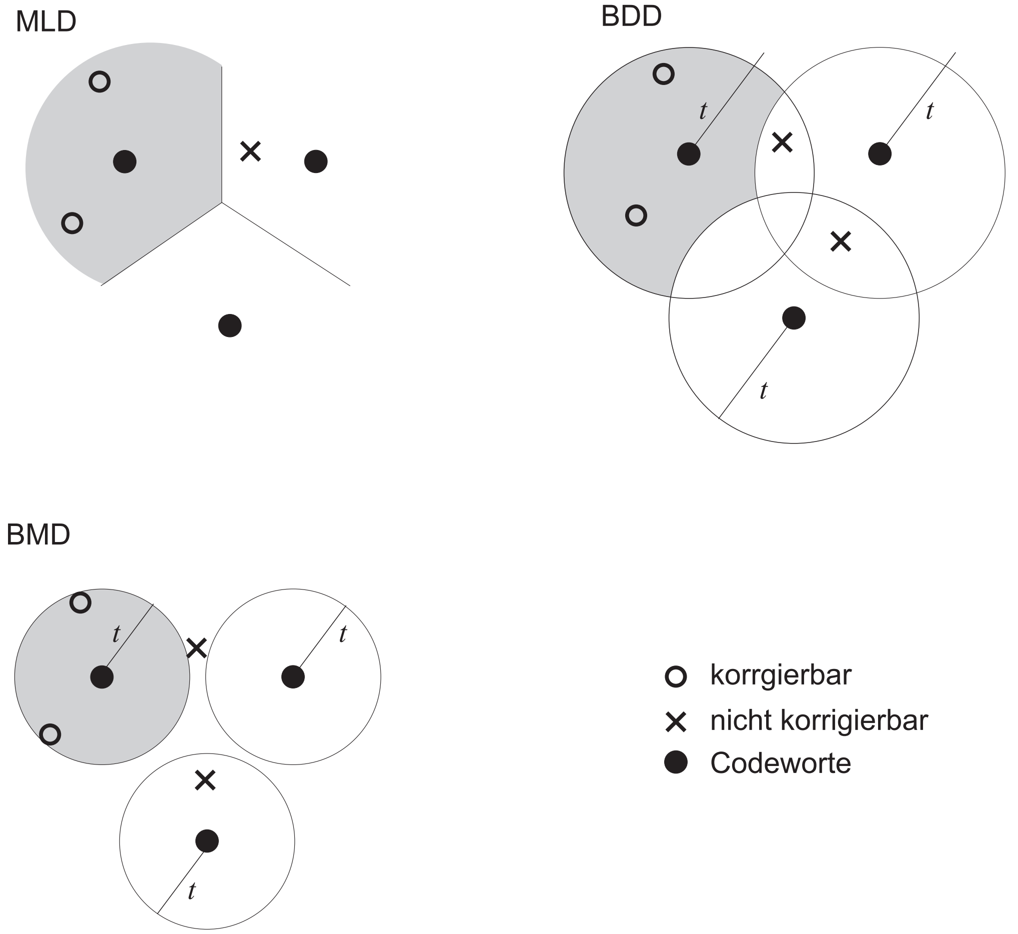


Bild 3.2: Veranschaulichung der Prinzipien von MLD, BDD und BMD

Bei perfekten Codes existiert zu jedem Syndrom genau ein korrigierbares Fehlermuster bzw. jedes korrigierbare Fehlermuster korrespondiert genau mit einem Syndrom. Damit sind die Empfangsworte komplett auf Decodierkugeln mit Radius $t = \left\lfloor \frac{d_{\min}-1}{2} \right\rfloor$ aufgeteilt.

Die Abschätzung der Leistungsfähigkeit von Blockcodes wird im Folgenden in drei Bereiche unterteilt, der Wahrscheinlichkeit für unerkannte Fehler bei Fehlererkennung, der Wahrscheinlichkeit für falsch korrigierte oder nicht korrigierbare Fehler im Fall einer Hard-Decision und einer Soft-Decision. Es wird stets die Wortfehlerwahrscheinlichkeit, nicht die Bitfehlerwahrscheinlichkeit betrachtet.

3.4.2 Fehlererkennung beim diskreten symmetrischen Kanal

Allgemein kann das Empfangswort $\mathbf{y}$ bei einer Hard-Decision im Empfänger durch die Überlagerung des gesendeten Codewortes $\mathbf{x}$ und einem Fehlerwort $\mathbf{e}$

$$\mathbf{y} = \mathbf{x} + \mathbf{e} \tag{3.14}$$

dargestellt werden. Dabei enthält $\mathbf{e}$ genau an den Stellen ein Symbol ungleich Null, an denen ein Fehler aufgetreten ist. Das Gewicht von $\mathbf{e}$ gibt somit die Anzahl der Fehler im empfangenen Wort $\mathbf{y}$ an. Ein Fehler kann immer dann nicht erkannt werden, wenn durch die Addition von $\mathbf{x}$ und $\mathbf{e}$ wieder ein gültiges Codewort entstanden ist. Aufgrund der Eigenschaften des den Code beschreibenden Vektorraums ist dies aber nur der Fall, wenn $\mathbf{e}$ selbst ein Codewort ist ($\mathbf{e} \in \Gamma$) und damit mindestens das Gewicht $d_{\min}$, also die Minimaldistanz besitzt.

Fehlerwahrscheinlichkeit

- Wahrscheinlichkeit für Fehler an Stelle i des Codewortes ($e_i \neq 0$): P_e
- Wahrscheinlichkeit für Fehlerfreiheit: $1 - P_e$
- Mittlere Wahrscheinlichkeit für bestimmten Wert an Stelle i ($e_i = 1, 2, \ldots q-1$): $\frac{P_e}{q-1}$
- Auftrittswahrscheinlichkeit eines bestimmten Fehlerwortes $\mathbf{e}$ mit Gewicht $w_{\mathrm{H}}(\mathbf{e})$

$$(1 - P_e)^{n - w_{\mathrm{H}}(\mathbf{e})} \cdot \left(\frac{P_e}{q-1}\right)^{w_{\mathrm{H}}(\mathbf{e})}$$

Wahrscheinlichkeit P_{ue} (ue: *undetected error*) eines unerkannten Fehlers

- Unerkannter Fehler nur für $\mathbf{e} \in \Gamma \quad \longrightarrow \quad$ alle q^k Codewörter erfassen
- Es gilt:

$$\begin{aligned} P_{\mathrm{ue}} &= \sum_{\mathbf{x} \in \Gamma \setminus \{0\}} (1 - P_e)^{n - w_{\mathrm{H}}(\mathbf{x})} \cdot \left(\frac{P_e}{q-1}\right)^{w_{\mathrm{H}}(\mathbf{x})} && (3.15) \\ &= \sum_{d=d_{\min}}^{n} A_d (1 - P_e)^{n-d} \cdot \left(\frac{P_e}{q-1}\right)^{d}. && (3.16) \end{aligned}$$

- Gl. (3.16) durch Zusammenfassen aller Codeworte $\mathbf{x}$ mit gleichem Hamming-Gewicht $w_{\mathrm{H}}(\mathbf{x})$ (siehe Distanzspektrum)
- Summe über alle Gewichte, also von $d_{\min}$ bis maximal n
- Zur Auswertung von Gl. (3.16) gesamtes Distanzspektrum erforderlich
- **Spezialfall:** $q = 2$ (BSC)

$$P_{ue} = \sum_{d=d_{\min}}^{n} A_d (1 - P_e)^{n-d} \cdot P_e^d \tag{3.17}$$

3.4.3 Fehlerkorrektur beim BSC

Die Wahrscheinlichkeit P_{ue} für unerkannte Fehler kann auch bei der Fehlerkorrektur aus Gl. (3.16) übernommen werden, denn Fehler, die erst gar nicht als solche zu erkennen sind, können auch nicht korrigiert werden. Hinsichtlich der Wahrscheinlichkeit P_w für das Auftreten eines nicht korrigierbaren oder aber nur falsch korrigierbaren Fehlers spielt mit Sicherheit auch die Art der Decodierung (s. Abschnitt 3.3) eine Rolle. Eine sehr einfache Bestimmung von P_w ist für die begrenzte Minimaldistanz-Decodierung (BMD) möglich. Da sie von den drei in Abschnitt 3.3 vorgestellten Verfahren die geringste Leistungsfähigkeit besitzt, stellt das Ergebnis gleichzeitig eine obere Schranke für die beiden übrigen Verfahren, die *Maximum Likelihood*-Decodierung (MLD) und die begrenzte Distanz-Decodierung (BDD) dar. Bei perfekten Codes ist die Abschätzung für die beiden zuletzt genannten Verfahren ebenfalls exakt.

Voraussetzung bei BMD: $(n,k,d_{\min})$-Code kann maximal $t = \lfloor \frac{d_{\min}-1}{2} \rfloor$ Fehler korrigieren

- Wortfehlerwahrscheinlichkeit:
$$P_w = 1 - P_{\mathrm{korrekt}}$$
- t Fehler immer korrigierbar $\longrightarrow$ Wahrscheinlichkeit P_{korrekt} für korrigierbares Fehlermuster:
  - Alle Fehlermuster mit Gewicht $w_{\mathrm{H}}(\mathbf{e}) \leq t$ betrachten
  - Binomialkoeffizient $\binom{n}{d}$ beschreibt Anzahl möglicher Anordnungen von d Symbolen ungleich Null in Codewort der Länge n

$$P_{\mathrm{korrekt}} = \sum_{d=0}^{t} \binom{n}{d} \cdot P_e^d \cdot (1-P_e)^{n-d} . \tag{3.18}$$

- Wortfehlerwahrscheinlichkeit

$$P_w = 1 - \sum_{d=0}^{t} \binom{n}{d} \cdot P_e^d \cdot (1-P_e)^{n-d} \tag{3.19}$$

$$= \sum_{d=t+1}^{n} \binom{n}{d} \cdot P_e^d \cdot (1-P_e)^{n-d} \tag{3.20}$$

Gl. (3.20) gilt, da die Summe der Auftrittswahrscheinlichkeiten über alle Codeworte $\mathbf{x}$ Eins ergeben muss und somit die Differenz in Gl. (3.19) auch durch die Summe über die restlichen Distanzen von t bis n ersetzt werden kann. Da t nur von der Minimaldistanz $d_{\min}$ abhängt, geht in Gl. (3.20) nicht die gesamte Gewichtsverteilung ein. Entsprechend den obigen Erläuterungen stellt Gl. (3.20) für die Decodierverfahren BDD und MLD eine obere Schranke dar; hier gilt:

$$P_w \leq \sum_{d=t+1}^{n} \binom{n}{d} \cdot P_e^d \cdot (1-P_e)^{n-d} . \tag{3.21}$$

Für perfekte Codes gilt Gleichheitszeichen in Gl. (3.21).

3.4.4 Fehlerkorrektur bei Soft-Decision

Im Gegensatz zur Hard-Decision spielt hier wieder das gesamte Distanzspektrum des Codes eine Rolle. Wir gehen im Folgenden zunächst davon aus, dass ein beliebiges, aber festes Codewort $\mathbf{x}^{(i)}$ gesendet wurde. Die Wahrscheinlichkeit für einen Decodierfehler wird dann mit $P_e(\mathbf{x}^{(i)})$ bezeichnet und lautet

$$P_e(\mathbf{x}^{(i)}) = P(\text{Decodierfehler} \mid \mathbf{x}^{(i)}) \tag{3.22}$$

$$= P(\mathbf{y} \notin \mathcal{M}_i \mid \mathbf{x}^{(i)}), \tag{3.23}$$

wobei die Menge $\mathcal{M}_i$ zu

$$\mathcal{M}_i = \{\mathbf{y} \mid P(\mathbf{y} \mid \mathbf{x}^{(i)}) \geq P(\mathbf{y} \mid \mathbf{x}^{(j)}), \forall \mathbf{x}^{(j)} \in \Gamma,\ j \neq i\} \tag{3.24}$$

definiert ist und die Menge aller $\mathbf{y}$ beschreibt, die eine geringere Distanz (größere Ähnlichkeit) zu $\mathbf{x}^{(i)}$ besitzen als zu irgendeinem anderen Codewort $\mathbf{x}^{(j\neq i)}$. Dieses Kriterium entspricht exakt der Maximum-Likelihood-Entscheidung, so dass alle $\mathbf{y} \in \mathcal{M}_i$ zu $\mathbf{x}^{(i)}$ decodiert würden. Dabei ist zu beachten, dass bei 'weich' entschiedenen Elementen y_i nicht die Hamming-Distanz, sondern ein geometrisches Maß wie z.B. die quadratische euklidische Distanz verwendet werden muss. Die zu $\mathcal{M}_i$ komplementäre Menge $\bar{\mathcal{M}}_i$ lässt sich zu

$$\begin{aligned}
\bar{\mathcal{M}}_i &= \mathcal{A}_{out} \setminus \mathcal{M}_i && (3.25)\\
&= \{\mathbf{y} \mid P(\mathbf{y} \mid \mathbf{x}^{(j)}) > P(\mathbf{y} \mid \mathbf{x}^{(i)}), \forall j \neq i\} && (3.26)\\
&= \bigcup_{\substack{j=1\\ j\neq i}}^{2^k} \underbrace{\{\mathbf{y} \mid P(\mathbf{y} \mid \mathbf{x}^{(j)}) > P(\mathbf{y} \mid \mathbf{x}^{(i)})\}}_{M_{i,j}} && (3.27)
\end{aligned}$$

formulieren. Damit kann die Wahrscheinlichkeit für eine falsche Detektion von $\mathbf{x}^{(i)}$ durch die *Union Bound* folgendermaßen abgeschätzt werden:

$$\begin{aligned}
P_e(\mathbf{x}^{(i)}) &= P(\mathbf{y} \in \bigcup_{\substack{j=1\\ j\neq i}}^{2^k} \mathcal{M}_{i,j} \mid \mathbf{x}^{(i)})\\
&\leq \sum_{\substack{j=1\\ j\neq i}}^{2^k} P(\mathbf{y} \in \mathcal{M}_{i,j} \mid \mathbf{x}^{(i)}). && (3.28)
\end{aligned}$$

Veranschaulichung der Ungleichung im zweiten Teil von Gl. (3.28):

- Empfangswort $\mathbf{y}$ kann auch in mehr als einer Menge $\mathcal{M}_{i,j}$ enthalten sein

$\rightarrow$ Summe über Einzelmengen größer als Vereinigungsmenge

- Gleichheit, wenn alle $\mathcal{M}_{i,j}$ disjunkt sind, d.h. jedes $\mathbf{y}$ kommt nur in einer einzigen Menge $\mathcal{M}_{i,j}$ vor

AWGN-Kanal mit binärer Übertragung ($q = 2$)
Wir setzen im Folgenden wieder eine antipodale Übertragung mit $x_\nu = \pm\sqrt{E_s/T_s}$ voraus. Bei Verwendung der *Maximum-Likelihood*-Decodierung gilt

$$P\left(\mathbf{y} \in \mathcal{M}_{i,j} \mid \mathbf{x}^{(i)}\right) = P\left(\|\mathbf{y} - \mathbf{x}^{(j)}\|^2 < \|\mathbf{y} - \mathbf{x}^{(i)}\|^2 \mid \mathbf{x}^{(i)}\right) . \tag{3.29}$$

Ein Fehler tritt also genau dann auf, wenn ein anderes Codewort $\mathbf{x}^{(j)} \neq \mathbf{x}^{(i)}$ näher am empfangenen Wort $\mathbf{y}$ liegt. Setzt man die Beziehung $\mathbf{y} = \mathbf{x}^{(i)} + \mathbf{n}$ in Gl. (3.29) ein, so ergibt sich

$$\begin{aligned}
P\left(\mathbf{y} \in \mathcal{M}_{i,j} \mid \mathbf{x}^{(i)}\right) &= P\left(\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)} + \mathbf{n}\|^2 < \|\mathbf{n}\|^2\right)\\
&= P\left(\sum_{r=0}^{n-1}\left(x_r^{(i)} - x_r^{(j)} + n_r\right)^2 < \sum_{r=0}^{n-1} n_r^2\right)\\
&= P\left(\underbrace{\sum_{r=0}^{n-1} n_r\left(x_r^{(i)} - x_r^{(j)}\right)}_{\xi} < -\frac{1}{2}\sum_{r=0}^{n-1}\left(x_r^{(i)} - x_r^{(j)}\right)^2\right).
\end{aligned}$$

- Interpretation von ξ:
  - Nur unterschiedliche Stellen von $\mathbf{x}^{(i)}$ und $\mathbf{x}^{(j)}$ von Bedeutung
  - An allen übrigen Stellen nimmt Differenz den Wert Null an

  $\rightarrow$ Insgesamt nur $d_{\mathrm{H}}(\mathbf{x}^{(i)},\mathbf{x}^{(j)})$ Stellen wichtig
  - Variable ξ ist gaußverteilte Zufallsvariable mit Mittelwert $\mu = 0$ und Varianz $\sigma_\xi^2 = N_0/2/T_s \cdot d_{\mathrm{H}}(\mathbf{x}^{(i)},\mathbf{x}^{(j)}) \cdot (2\sqrt{E_s/T_s})^2 = 2 \cdot d_{\mathrm{H}}(\mathbf{x}^{(i)},\mathbf{x}^{(j)}) \cdot E_s N_0/T_s^2$
- Rechte Seite nimmt konstanten Wert $-1/2 d_{\mathrm{H}}(\mathbf{x}^{(i)},\mathbf{x}^{(j)})(2\sqrt{E_s/T_s})^2 = -2 d_{\mathrm{H}}(\mathbf{x}^{(i)},\mathbf{x}^{(j)}) E_s/T_s$ an

Für die Fehlerwahrscheinlichkeit $P(\mathbf{y} \in \mathcal{M}_{i,j} \mid \mathbf{x}^{(i)})$ gilt dann

$$\begin{aligned}
P(\mathbf{y} \in \mathcal{M}_{i,j} \mid \mathbf{x}^{(i)}) &= \int_{-\infty}^{-2d_{\mathrm{H}}(\mathbf{x}^{(i)},\mathbf{x}^{(j)})E_s/T_s} \frac{1}{\sqrt{2\pi 2 d_{\mathrm{H}}(\mathbf{x}^{(i)},\mathbf{x}^{(j)}) E_s N_0/T_s^2}} e^{-\frac{\xi^2}{4 d_{\mathrm{H}}(\mathbf{x}^{(i)},\mathbf{x}^{(j)}) E_s N_0/T_s^2}} \, d\xi \\
&= \frac{1}{\sqrt{\pi}} \int_{-\infty}^{-\sqrt{d_{\mathrm{H}}(\mathbf{x}^{(i)},\mathbf{x}^{(j)})E_s/N_0}} e^{-\xi^2} \, d\xi = \frac{1}{\sqrt{\pi}} \int_{\sqrt{d_{\mathrm{H}}(\mathbf{x}^{(i)},\mathbf{x}^{(j)})E_s/N_0}}^{\infty} e^{-\xi^2} \, d\xi \\
&= \frac{1}{2}\mathrm{erfc}\left(\sqrt{d_{\mathrm{H}}(\mathbf{x}^{(i)},\mathbf{x}^{(j)})\frac{E_s}{N_0}}\right) \qquad (3.30) \\
&= Q\left(\sqrt{2 d_{\mathrm{H}}(\mathbf{x}^{(i)},\mathbf{x}^{(j)})\frac{E_s}{N_0}}\right).
\end{aligned}$$

Die Wortfehlerwahrscheinlichkeit für $\mathbf{x}^{(i)}$ kann jetzt zu

$$\begin{aligned}
P_e(\mathbf{x}^{(i)}) &\leq \frac{1}{2} \sum_{\substack{j=1 \\ j \neq i}}^{2^k} \mathrm{erfc}\left(\sqrt{d_{\mathrm{H}}(\mathbf{x}^{(i)},\mathbf{x}^{(j)})\frac{E_s}{N_0}}\right) \qquad (3.31) \\
&= \frac{1}{2} \sum_{d=d_{\min}}^{n} A_d \cdot \mathrm{erfc}\left(\sqrt{d\frac{E_s}{N_0}}\right) \\
&= \sum_{d=d_{\min}}^{n} A_d \cdot Q\left(\sqrt{2d\frac{E_s}{N_0}}\right)
\end{aligned}$$

angegeben werden. Da $P_e(\mathbf{x}^{(i)})$ unabhängig von einem speziell gewählten Codewort $\mathbf{x}^{(i)}$ ist, stellt $P_e(\mathbf{x}^{(i)})$ auch gleichzeitig die allgemeine Wortfehlerwahrscheinlichkeit P_w des gesamten Codes dar. Die Ergebnisse für die Abschätzung der Wortfehlerwahrscheinlichkeit bei Hard- und Soft-Decision für einen AWGN-Kanal zeigt Bild 3.3. Der Unterschied beträgt bei sehr kleinen Fehlerraten etwa 1.7 dB, was verdeutlicht, dass bei einer harten Entscheidung vor der Decodierung Information unwiderruflich verlorengeht. Obwohl die Abschätzung von P_w bei Hard- bzw. Soft-Decision für unterschiedliche Decodierverfahren (BMD bzw. MLD) hergeleitet wurde, ist der Vergleich hier erlaubt, da der $(7,4)_2$-Hamming-Code perfekt ist und somit das Gleichheitszeichen in Gl. (3.21) gilt.

3.5 Matrixbeschreibung von Blockcodes

3.5.1 Generatormatrix

Lineare Blockcodes können vollständig durch eine sogenannte Generatormatrix $\mathbf{G}$ beschrieben werden. Für einen allgemeinen (n,k)-Blockcode besteht diese aus k Zeilen und n Spalten. Aus jedem Informationswort $\mathbf{u}$

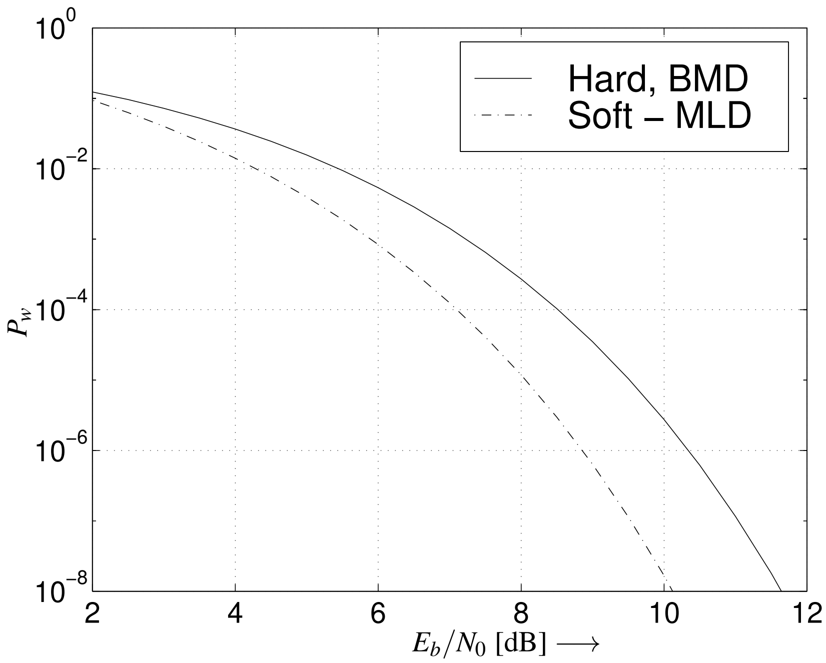


Bild 3.3: Wortfehlerwahrscheinlichkeit für $(7,4)_2$-Hamming-Code beim AWGN-Kanal

kann dann mit der Beziehung

$$\mathbf{x} = \mathbf{u} \cdot \mathbf{G} \qquad \text{mit} \qquad \mathbf{G} = \begin{bmatrix} g_{0,0} & \cdots & g_{0,n-1} \\ & \vdots & \\ g_{k-1,0} & \cdots & g_{k-1,n-1} \end{bmatrix} \tag{3.32}$$

ein Codewort gebildet werden. Der Vektorraum Γ aller Codeworte besteht aus 2^k Elementen und es gilt:

$$\Gamma = \left\{ \mathbf{u} \cdot \mathbf{G} \mid \mathbf{u} \in \mathrm{GF}(q)^k \right\} \tag{3.33}$$

Das Codewort $\mathbf{x}$ kann als Linearkombination der Zeilen von $\mathbf{G}$ aufgefasst werden, wobei die Koeffizienten die Symbole des Informationswortes sind. Aufgrund der geforderten Linearität und der Abgeschlossenheit des Vektorraums folgt hieraus, dass die Zeilen der Generatormatrix gültige Codeworte repräsentieren. Sie sind linear unabhängig und bilden die Basis des Coderaums, d.h. sie spannen diesen auf.

Erlaubte (elementare) Matrixoperationen

- Vertauschen von Spalten (**Äquivalente Codes**)
  - Vertauschen zweier oder mehrerer Spalten in $\mathbf{G}$ führt zu Wechsel der Reihenfolge der Symbole eines Codewortes und liefert **äquivalenten Code**
  - Spalten von $\mathbf{G}$ können beliebig angeordnet werden
  - Äquivalente Codes besitzen identische Distanzeigenschaften, nur die Zuordnung von $\mathbf{u}$ auf $\mathbf{x}$ ist unterschiedlich
  - Trotzdem Unterschiede in Leistungsfähigkeit z.B. bei Erkennung bzw. Korrektur von Bündelfehlern
- Zeilenoperationen (Code ändert sich nicht)
  - Vertauschen zweier Zeilen
  - Multiplikation einer Zeile mit einem Skalar gemäß den Regeln des GF(q)

– Linearkombinationen zweier Zeilen, d.h. Addition zweier mit Skalaren multiplizierten Zeilen.

Durch elementare Operationen lässt sich jede Generatormatrix in die *gaußsche Normalform* überführen.

$$\mathbf{G} = [\mathbf{I}_{k,k} \mid \mathbf{P}_{k,n-k}] \tag{3.34}$$

- $\mathbf{I}_{k,k}$ repräsentiert die Einheitsmatrix mit k Zeilen und Spalten
- $\mathbf{P}_{k,n-k}$ eine Prüfmatrix mit entsprechend k Zeilen und $n-k$ Spalten
- Generatormatrix in Gl. (3.34) beschreibt systematischen Code (zunächst Wiederholung der Infosymbole durch $\mathbf{I}_{k,k}$, dann Anhängen der Prüfsymbole mit $P_{k,n-k}$)

→ Alle linearen Blockcodes immer als systematische Codes darstellbar

3.5.2 Prüfmatrix

Eine zum vorigen Abschnitt äquivalente Beschreibung ist mit der sogenannten Prüfmatrix **H** möglich. Sie wird auch zur Überprüfung der Richtigkeit eines Empfangsvektors herangezogen. Unter Verwendung der gaußschen Normalform aus Gl. (3.34) kann die Prüfmatrix zu

$$\mathbf{H} = \left[-\mathbf{P}_{k,n-k}^T \mid \mathbf{I}_{n-k,n-k}\right] \tag{3.35}$$

dargestellt werden. Demnach besteht sie aus $n-k$ Zeilen und n Spalten. Es gilt stets

$$\mathbf{x} \cdot \mathbf{H}^T = \mathbf{0}, \tag{3.36}$$

d.h. die Zeilen in **H** (Spalten in $\mathbf{H}^T$) sind orthogonal zu allen Codeworten. Dementsprechend stellt der Coderaum Γ den Nullraum bezüglich **H** dar und kann somit auch durch

$$\Gamma = \left\{\mathbf{x} \in \mathrm{GF}(q)^n \mid \mathbf{x}\mathbf{H}^T = \mathbf{0}\right\} \tag{3.37}$$

definiert werden. Automatisch gilt auch der Zusammenhang

$$\mathbf{G}\mathbf{H}^T = \mathbf{0}. \tag{3.38}$$

Überprüfung des korrekten Empfangs mit Prüfmatrix
Die Prüfmatrix wird (wie in Abschnitt 3.6.6 noch gezeigt wird) auch häufig zur Decodierung eingesetzt (daher auch die Namensgebung). Mit Hilfe der Beziehung 3.36 kann schnell nachgeprüft werden, ob es sich bei dem empfangenen Wort **y** um ein Codewort handelt oder nicht. Das Ergebnis wird auch Syndrom **s** genannt und es gilt

$$\mathbf{s} = \mathbf{y} \cdot \mathbf{H}^T = (\mathbf{x}+\mathbf{e}) \cdot \mathbf{H}^T = \underbrace{\mathbf{x}\mathbf{H}^T}_{=\mathbf{0}} + \mathbf{e}\mathbf{H}^T = \mathbf{e}\mathbf{H}^T. \tag{3.39}$$

Nimmt das Syndrom den Wert Null an, so ist entweder kein Fehler ($\mathbf{e} = \mathbf{0}$) aufgetreten oder durch einen Fehler ($\mathbf{e} \in \Gamma$) ist ein neues Codewort $\mathbf{x}'$, aber nicht das gesendete **x** entstanden. In diesem Fall kann der Fehler nicht erkannt und somit auch nicht korrigiert werden.

Ohne Beweis soll an dieser Stelle noch angemerkt werden, dass mit Hilfe der Prüfmatrix die Minimaldistanz eines Codes berechnet werden kann. Sie stellt nämlich die minimale Anzahl linear abhängiger Spalten in **H** dar. Eine derartige Berechnung ist mit der Generatormatrix nicht möglich.

3.5.3 Duale Codes

Verwendet man zur Codeerzeugung statt der Generatormatrix die Prüfmatrix, so ergibt sich ein zum Original orthogonaler Code $\Gamma^{\perp}$. Er wird auch als *dualer Code* bezeichnet und ist wie folgt definiert:

$$\begin{aligned} \Gamma^{\perp} &= \{\mathbf{b} \in \mathrm{GF}(q)^n \mid \mathbf{b} \perp \mathbf{a} \quad \forall \quad \mathbf{a} \in \Gamma\} = \left\{\mathbf{b} \in \mathrm{GF}(q)^n \mid \mathbf{b} \cdot \mathbf{G}^T = \mathbf{0}\right\} \\ &= \left\{\mathbf{v} \cdot \mathbf{H} \mid \mathbf{v} \in \mathrm{GF}(q)^{n-k}\right\}, \end{aligned} \tag{3.40}$$

wobei mit $\mathbf{b} = \mathbf{v} \cdot \mathbf{H}$ die Codeworte des dualen Codes gebildet werden können. Aufgrund der Dimension der Prüfmatrix $\mathbf{H}$ bestehen die Codeworte $\mathbf{b}$ ebenfalls aus n Symbolen, allerdings enthält der Coderaum jetzt q^{n-k} Elemente. Diese Tatsache wird teilweise auch bei der Decodierung ausgenutzt. Ist $n-k$ deutlich kleiner als k, kann es von Vorteil sein, die Decodierung mit Hilfe des dualen Codes zu realisieren. Darauf soll hier allerdings nicht näher eingegangen werden.

3.5.4 Nebenklassenzerlegung

Bekanntes Syndrom:

$$\mathbf{s} = \mathbf{y} \cdot \mathbf{H}^T = (\mathbf{x} + \mathbf{e}) \cdot \mathbf{H}^T = \underbrace{\mathbf{x} \cdot \mathbf{H}^T}_{=\mathbf{0}} + \mathbf{e} \cdot \mathbf{H}^T \tag{3.41}$$

Somit ist das Syndrom unabhängig vom gesendeten Codewort $\mathbf{x}$. Da es q^k Codeworte gibt, existieren im $\mathrm{GF}(q)^n$ insgesamt $q^n - q^k$ Fehlermuster. Ferner gibt es aber nur q^{n-k} Syndrome, so dass das Fehlermuster $\mathbf{e}$ nicht eindeutig durch das von ihm erzeugte Syndrom $\mathbf{s} = \mathbf{e} \cdot \mathbf{H}^T$ bestimmt ist. Mit anderen Worten, mehrere Fehlervektoren $\mathbf{e}$ erzeugen das gleiche Syndrom $\mathbf{s}$.

Aufteilung in Nebenklassen
Im Folgenden numerieren wir alle Syndrome mit $\mathbf{s}_\mu$ durch $(0 \le \mu \le q^{n-k} - 1)$. Weiterhin sei das Syndrom $\mathbf{s}_0 = \mathbf{0}$. Dann können alle Empfangsvektoren $\mathbf{y}$, die ein bestimmtes Syndrom $\mathbf{s}_\mu$ erzeugen, in einer Menge $\mathcal{M}_\mu$ (*Cosets*) zusammengefasst werden. Es gilt:

$$\mathcal{M}_\mu = \left\{\mathbf{e} \in \mathrm{GF}(q)^n \mid \mathbf{e} \cdot \mathbf{H}^T = \mathbf{s}_\mu\right\}. \tag{3.42}$$

Die Mengen $\mathcal{M}_\mu$ werden als *Nebenklassen* (*Cosets*) bezeichnet. Speziell gilt für die Nebenklasse mit dem Nullsyndrom $\mathcal{M}_0 = \Gamma$, da $\mathbf{s} = \mathbf{0}$ ein gültiges Codewort anzeigt. Die übrigen Nebenklassen $\mathcal{M}_{\mu \neq 0}$ enthalten keine Codeworte. Es wird nun zu jeder Nebenklasse ein Anführer $\mathbf{e}_\mu$ bestimmt, welcher im Allgemeinen das geringste Gewicht aller Elemente dieser Nebenklasse besitzt. Die Wahl eines Anführers ist nicht zwingend eindeutig.

Decodierung mittels Nebenklassenzerlegung (*Standard Array Decoding*)

- In Nebenklassen wird nach empfangenem Wort $\mathbf{y}$ gesucht
- Annahme: Fehler mit geringstem Gewicht am wahrscheinlichsten
- $\mathbf{y} \in \mathcal{M}_\mu \;\rightarrow\;$ Fehler am wahrscheinlichsten durch Überlagerung des Anführers dieser Klasse hervorgerufen
- Decodierung durch

$$\hat{\mathbf{x}} = \mathbf{y} - \mathbf{e}_\mu \qquad \text{und} \qquad \hat{\mathbf{u}} = g^{-1}(\hat{\mathbf{x}}) \tag{3.43}$$

- Decodiervorschrift realisiert Maximum-Likelihood-Decodierung
- **Nachteil:** Suche nach richtiger Nebenklasse selbst für einfache Blockcodes viel zu aufwendig.

Die im Folgenden behandelte Syndromdecodierung stellt eine gewisse Vereinfachung dar.

3.5.5 Syndromdecodierung

- Ablegen aller q^{n-k} Syndrome $\mathbf{s}_\mu$ mit jeweiligen Nebenklassenanführern in Tabelle
- Nach Empfang von $\mathbf{y}$ Berechnung des Syndroms $\mathbf{s} = \mathbf{y} \cdot \mathbf{H}^T$
- Suche des Syndroms in Tabelle
- Abschließend Subtraktion des zum Syndrom gehörenden Nebenklassenanführers vom Empfangswort
- Suche nach passendem Syndrom schränkt Suche im Vergleich zur reinen Nebenklassenzerlegung ein

Beispiel: $(7,4)$**-Hamming-Code**

- Es gibt $2^{7-4} = 2^3 = 8$ Syndrome einschließlich dem Nullsyndrom
- Es existieren $2^7 - 2^4 = 128 - 16 = 112$ Fehlervektoren
- Wegen $d_{min} = 3$ nur $t = 1$ Einzelfehler korrigierbar

$\rightarrow$ Nebenklassenanführer bestehen aus 7 möglichen Einzelfehlermustern

Syndrom	Nebenklassenanführer
0 0 1	0 0 0 0 0 0 1
0 1 0	0 0 0 0 0 1 0
0 1 1	1 0 0 0 0 0 0
1 0 0	0 0 0 0 1 0 0
1 0 1	0 1 0 0 0 0 0
1 1 0	0 0 1 0 0 0 0
1 1 1	0 0 0 1 0 0 0

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

3.5.6 Modifikation linearer Codes

Die Modifikation von Codes dient in der Praxis dazu, den Blockcode an die speziellen, für eine bestimmte Anwendung geltenden Randbedingungen anzupassen. Das kann beispielsweise die vorgegebene Länge eines Datenrahmens sein. Für die Theorie der Codierung spielen diese Modifikationen nur eine untergeordnete Rolle. Prinzipiell werden folgende Formen der Modifikation unterschieden:

1. **Expandieren**: es werden zusätzliche Prüfsymbole angehängt;
$$n' > n, \quad k' = k, \quad R'_c < R_c; \quad d'_{\min} \geq d_{\min}$$
2. **Punktieren**: es werden Prüfsymbole ausgeblendet;
$$n' < n, \quad k' = k, \quad R'_c > R_c; \quad d'_{\min} \leq d_{\min}$$
3. **Verlängern**: es werden zusätzliche Infosymbole angehängt;
$$n' > n, \quad k' > k, \quad n' - k' = n - k, \quad R'_c > R_c; \quad d'_{\min} \leq d_{\min}$$
4. **Verkürzen**: es werden Infosymbole ausgeblendet;
$$n' < n, \quad k' < k, \quad n' - k' = n - k, \quad R'_c < R_c; \quad d'_{\min} \geq d_{\min}$$

Beispiel: **Expansion** Aus einem (n,k)-Code wird durch Anhängen eines zusätzlichen Prüfbits s ein $(n+1,k)$-Code. Die neue Generatormatrix lautet dann

$$\mathbf{G}' = \left[\begin{array}{c|c} & s_0 \\ \mathbf{G} & \vdots \\ & s_{k-1} \end{array}\right]. \tag{3.44}$$

Die Beziehungen der Prüfsymbole behalten weiterhin ihre Gültigkeit, d.h. die Fehlerkorrekturfähigkeit des Codes bleibt unverändert. Mit dem zusätzlichen Prüfsymbol kann jetzt gleichzeitig noch ein Fehler erkannt werden. Die zugehörige Prüfmatrix hat nun die Form

$$\mathbf{H}' = \left[\begin{array}{ccc|c} & & & 0 \\ & \mathbf{H} & & \vdots \\ & & & 0 \\ \hline 1 & \dots & 1 & 1 \end{array}\right]. \tag{3.45}$$

3.5.7 Einfache Parity-Check- und Wiederholungscodes

$(n,1,n)_q$-**Wiederholungscode (*Repetition Code*)**

- Codeworte bestehen aus 1 Informationssymbol $\mathbf{u} = u_0$ und $n-1$ Prüfsymbolen $p_i = u_0$
- Informationssymbole wird also $n-1$ mal wiederholt
- Coderate $R_c = 1/n$
- Coderaum besteht aus nur zwei Elementen, nämlich dem Nullwort und dem Einswort

$$\Gamma = \{\underbrace{0\dots0}_{n}, \underbrace{1\dots1}_{n}\}$$

- Codeworte haben Distanz n zueinander (sehr gering im Vergleich zur Coderate)
- Geringe Leistungsfähigkeit aufgrund ihrer einfachen Struktur
- Generatormatrix und Prüfmatrix:

$$\mathbf{G} = \underbrace{[1 \mid 1 \dots 1]}_{n} \qquad \mathbf{H} = \left.\left[\begin{array}{c|ccccc} 1 & 1 & & & & \\ 1 & & 1 & & 0 & \\ \vdots & & & \ddots & & \\ 1 & & 0 & & 1 & \\ 1 & & & & & 1 \end{array}\right]\right\} n-1. \tag{3.46}$$

$(n,n-1,2)_q$-***Single Parity Check Code*** **(SPC-Code)**

- SPC-Codes hängen den Informationssymbolen ein Prüfsymbol an
- Prüfsymbol berechnet sich aus der Quersumme aller Infosymbole (im GF(q))
- Minimaldistanz von SPC-Codes: $d_{\min} = 2$
  $\rightarrow$ Keine Fehlerkorrektur möglich
  $\rightarrow$ Einzelfehler können erkannt werden

- Generator- und Prüfmatrix:

$$\mathbf{G} = \left.\left[\begin{array}{ccccc|c} 1 & & & & & 1 \\ & 1 & & 0 & & 1 \\ & & \ddots & & & \vdots \\ & 0 & & 1 & & 1 \\ & & & & 1 & 1 \end{array}\right]\right\} n-1 \qquad \mathbf{H} = \underbrace{[1\,1\,\ldots\,1]}_{n} \tag{3.47}$$

Wie den beiden Gleichungen (3.46) und (3.47) zu entnehmen ist, können Parity-Check-Code und Wiederholungscode durch einfaches Vertauschen von Prüfmatrix und Generatormatrix ineinander überführt werden. Sie sind daher zueinander duale, also orthogonale Codes.

3.5.8 Hamming-Codes, Simplex-Codes

Hamming-Codes
Hamming-Codes stellen die wahrscheinlich bekannteste Klasse der 1-Fehler korrigierenden und 2-Fehler erkennenden Codes dar. Sie besitzen die Eigenschaft, dass mit zunehmender Blocklänge n die Coderate R_c gegen den Wert 1 strebt.

Definition:

Ein $(n,k,d_{min})_q = (n,n-r,3)_q$-Hamming-Code der Ordnung r ist durch

$$n = \frac{q^r-1}{q-1}$$

definiert. Hamming-Codes sind perfekte Codes, d.h. die Anzahl der darstellbaren Syndrome entspricht exakt der Zahl korrigierbarer Fehlermuster. Für den Spezialfall $q=2$ ergeben sich $(2^r-1, 2^r-r-1, 3)$-Hamming-Codes, es kommen also folgende Kombinationen vor: (3,1), (7,4), (15,11), (31,26), (63,57), (127,120) ...

Hier stellen die Spalten der Prüfmatrix die 2^r-1 binären Worte der Länge r dar (ohne Nullwort).

Beispiel: **(7,4)-Hamming-Code**

$$\mathbf{G} = \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right] \qquad \mathbf{H} = \left[\begin{array}{cccc|ccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right]$$

Hamming-Codes sind Perfekte Codes

- Prüfmatrix **H** enthält bis auf Nullsyndrom alle $2^{n-k}-1$ Syndrome als Spalten
- Somit gilt $n = 2^{n-k}-1$, d.h. die Anzahl korrigierbarer Fehlermuster für $t=1$ entspricht der Anzahl der Syndrome minus Eins
- Vergleich der Spalten von **H** mit dem Nebenklassenanführern (s. Tabelle Syndromdecodierung):
  - Position des Syndroms in **H** ist gleich der Position des Fehlers in **y**
  - Bei Fehler an der 4. Stelle von **x** ($\mathbf{e}_4 = (0001000)$) lautet Syndrom $\mathbf{s}_4 = (111)$
  - $\mathbf{s}_4$ steht in vierter Spalte von **H**
  - → Fehlerkorrektur beschränkt sich nach Syndromberechnung auf Suche der entsprechenden Spalte in Prüfmatrix

- Durch Umsortieren der Spalten in $\mathbf{H}$ ließe sich diese Suche vermeiden, Syndrom – als Dezimalzahl interpretiert – könnte direkt als Adresse für fehlerhafte Stelle dienen;
  Man erhält dann einen äquivalenten Code!

- $\mathbf{H}$ hätte dann die Form (nicht mehr systematisch):

$$\mathbf{H} = \left[\begin{array}{cccc|ccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right]$$

Simplex-Codes
Der mit der Prüfmatrix $\mathbf{H}$ erzeugte duale Code wird als *Simplex-Code* bezeichnet. Er zeichnet sich durch die Eigenschaft aus, dass alle Zeilen von $\mathbf{H}$ und somit auch alle Codeworte das konstante Gewicht 2^{r-1} haben (abgesehen vom Nullwort). Dies geht aus der Tatsache hervor, dass die Spalten von $\mathbf{H}$ alle Zahlenkombinationen von 1 bis $2^r - 1$ darstellen und somit Einsen und Nullen gleichverteilt sind. Außerdem besitzen alle Codeworte die konstante Distanz 2^{r-1} zueinander, was in der Geometrie als Simplex bezeichnet wird.

3.6 Zyklische Codes

3.6.1 Definition zyklischer Codes

Zyklische Codes stellen eine Teilmenge der linearen Codes dar. Sie enthalten noch mehr zusätzliche algebraische Struktur, wodurch sie sich kompakt beschreiben lassen und sehr einfache schaltungstechnische Realisierung des Decodierers erlauben. Prinzipiell zeichnen sie sich dadurch aus, dass die zyklische Verschiebung eines Codewortes $\mathbf{x}$ der Länge n wiederum ein Codewort ergibt.

$$[x_0\, x_1\, \ldots\, x_{n-1}] \in \Gamma \quad \Longrightarrow \quad [x_{n-1}\, x_0\, \ldots\, x_{n-2}] \in \Gamma \tag{3.48}$$

Gl. (3.48) impliziert auch, dass mehrfache zyklische Verschiebungen ebenfalls gültige Codeworte erzeugen. Eine elegante Möglichkeit der Beschreibung zyklischer Codes stellt die Nutzung von Polynomen dar. Sie können einfach aus der Zuordnung

$$\begin{gathered} [x_0\, x_1\, \ldots\, x_{n-1}] \in \mathrm{GF}(q)^n \\ \updownarrow \\ x(D) = \sum_{i=0}^{n-1} x_i \cdot D^i \in \mathrm{GF}_q[D]_{n-1} \end{gathered} \tag{3.49}$$

$\mathrm{GF}_q[D]$: Menge aller Polynome beliebigen Grades, Koeffizienten aus $\mathrm{GF}(q)$
$\mathrm{GF}_q[D]_r$: Menge aller Polynome mit maximalem Grad r, Koeffizienten aus $\mathrm{GF}(q)$

gewonnen werden. Dabei repräsentiert D lediglich einen Platzhalter. Bei einem *normierten Polynom* erfüllt der Koeffizient der höchsten Potenz die Bedingung $x_{n-1} = 1$. Die zyklische Verschiebung des Codewortes $\mathbf{x}$ um m Stellen kann durch eine Multiplikation von $x(D)$ mit D^m und anschließender Division modulo $D^n - 1$ realisiert werden.

$$\begin{array}{ccc} [x_0\, x_1\, \ldots\, x_{n-1}] & \Rightarrow & [x_{n-m}\, \ldots\, x_{n-1}\, x_0\, \ldots\, x_{n-m-1}] \\ \updownarrow & & \\ x(D) = \displaystyle\sum_{i=0}^{n-1} x_i \cdot D^i & \Rightarrow & R_{D^n-1}[D^m \cdot x(D)] \end{array}$$

3.6.2 Beschreibung mit Generatorpolynom

Zu einem zyklischen (n,k)-Code existiert genau ein *Generatorpolynom* $g(D)$ vom Grad $n-k$, mit dem der gesamte Coderaum erzeugt werden kann. Stellt $u(D)$ das Polynom des Informationswortes dar, so kann über die Beziehung

$$x(D) = u(D) \cdot g(D) \tag{3.50}$$

das zugehörige Codewort ermittelt werden. Demnach wird der Coderaum aus allen möglichen Linearkombinationen von zyklisch verschobenen Versionen des Generatorpolynoms bebildet. Jedes Codewort ist daher durch $g(D)$ ohne Rest teilbar. Für ein normiertes Generatorpolynom gilt $g_{n-k} = 1$, so dass es die Form

$$g(D) = g_0 + g_1 D + \ldots + g_{n-k-1} D^{n-k-1} + D^{n-k}$$

besitzt. Für beliebiges Polynom $b(D)$ vom Grad $< n$ gilt allgemein:

$$b(D) = u(D) \cdot g(D) + r(D) \qquad \text{mit} \qquad \text{Grad } r(D) < \text{Grad } g(D) . \tag{3.51}$$

Das Polynom $r(D)$ stellt den Rest der Polynomdivision von $b(D)$ mit $g(D)$ dar. Wir verwenden im Folgenden eine Abkürzung für Reste von Polynomdivisionen:

$$r(D) = b(D) \,\text{modulo}\, g(D) = R_{g(D)}\left[b(D)\right] \tag{3.52}$$

Allgemein gilt, dass die Division modulo $g(D)$ mit dem Gleichsetzen von $g(D) = 0$ in $b(D)$ übereinstimmt. Mit

$$r(D) = R_{g(D)}\left[b(D)\right] = b(d)|_{g(D)=0} \tag{3.53}$$

lässt sich dann oftmals die Modulo-Berechnung vereinfachen.

Der Coderaum Γ lässt sich mit Hilfe von $g(D)$ nun folgendermaßen beschreiben:

$$\begin{aligned} \Gamma &= \left\{u(D) \cdot g(D) \;|\; u(D) \in \mathrm{GF}_q[D]_{k-1}\right\} & (3.54)\\ &= \left\{x(D) \in \mathrm{GF}_q[D]_{n-1} \;|\; R_{g(D)}\left[x(D)\right] = 0\right\} . & (3.55) \end{aligned}$$

Bezüglich der Polynomdivision gelten folgende Regeln:

- $R_{g(D)}\left[a(D) + b(D)\right] = R_{g(D)}\left[a(D)\right] + R_{g(D)}\left[b(D)\right]$
- $R_{g(D)}\left[a(D) \cdot b(D)\right] = R_{g(D)}\left[R_{g(D)}\left[a(D)\right] \cdot R_{g(D)}\left[b(D)\right]\right]$
- $R_{g(D)}\left[a(D) \cdot g(D)\right] = 0$
- Grad $a(D) <$ Grad $g(D) \Rightarrow R_{g(D)}\left[a(D)\right] = a(D)$

Ein Generatorpolynom $g(D)$ beschreibt einen Code nicht eindeutig. Bisher wurde nämlich nur die Abhängigkeit zwischen $g(D)$ und der Anzahl der Prüfstellen $n-k$ (Grad $g(D) = n-k$) betrachtet. Zur eindeutigen Beschreibung des Codes ist jedoch noch eine Beziehung zur Codewortlänge n nötig.

Satz: *Sei $g(D)$ ein Generatorpolynom vom Grad $n-k$. Falls Γ zyklisch ist, wird $R_{g(D)}[D^n - 1] = 0$ erfüllt, d.h. $g(D)$ teilt $D^n - 1$ ohne Rest.*

Somit kann ein Generatorpolynom durchaus verschiedene Codes beschreiben, da für mehrere Codewortlängen n gelten kann: $R_{g(D)}\left[D^n - 1\right] = 0$. Die Anzahl $n-k$ der Prüfstellen, d.h. der Grad von $g(D)$ ist dabei konstant, die Codewortlänge n und somit die Anzahl k der Informationsstellen je Codewort allerdings nicht.

Da zyklische Codes eine Untermenge der linearen Blockcodes sind, gelten selbstverständlich alle bisherigen Aussagen auch hier. So führt z.B. die Addition zweier Codeworte wiederum zu einem gültigen Codewort. Bei bekanntem n enthält $g(D)$ alle einen Code bestimmenden Informationen und beschreibt ihn damit vollständig.

Berechnung der Generatormatrix G aus $g(D)$

$$\begin{bmatrix} g_0 & \cdots & g_{n-k} & & \\ & g_0 & \cdots & g_{n-k} & \\ & & g_0 & \cdots & g_{n-k} \end{bmatrix} \qquad \longleftrightarrow \qquad \begin{bmatrix} g(D) \\ D\,g(D) \\ \vdots \\ D^{k-1}\,g(D) \end{bmatrix} \tag{3.56}$$

Diese Generierung von **G** führt nicht automatisch zu einem systematischem Code. Allerdings kann durch elementare Zeilenoperationen immer die gaußsche Normalform erzielt und somit eine systematische Beschreibung des Codes erreicht werden.

Beispiel: Zyklischer $(7,4)$ Hamming-Code mit $g(D) = 1 + D^2 + D^3$, $k = 4$, $n = 7$

$$\begin{matrix} \\ \\ \\ D^3\,g(D) \end{matrix} \quad \left[\begin{array}{cccc|ccc} & & & & & & \\ & & & & & & \\ & & & & & & \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

$$\begin{matrix} \\ \\ D^2\,g(D) \\ D^3\,g(D) \end{matrix} \quad \left[\begin{array}{cccc|ccc} & & & & & & \\ & & & & & & \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

$$\begin{matrix} \\ D\,g(D) + D^3\,g(D) \\ D^2\,g(D) \\ D^3\,g(D) \end{matrix} \quad \left[\begin{array}{cccc|ccc} & & & & & & \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

$$\begin{matrix} g(D) + D^2\,g(D)) + D^3\,g(D) \\ D\,g(D) + D^3\,g(D) \\ D^2\,g(D) \\ D^3\,g(D) \end{matrix} \quad \left[\begin{array}{cccc|ccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

3.6.3 Beschreibung mit Prüfpolynom

Analog zu Abschnitt 3.5.2 kann auch bei den zyklischen Codes ein Prüfpolynom zur Codeerzeugung verwendet werden. Zur Erinnerung:

$$\Gamma = \left\{ \mathbf{uG} \mid \mathbf{u} \in \mathrm{GF}(q)^k \right\} = \{ \mathbf{x} \in \mathrm{GF}(q)^n \mid \mathbf{xH}^T = 0 \}$$

Im Folgenden sei $g(D)$ ein Generatorpolynom vom Grad $n-k$ und es gilt $R_{g(D)}[D^n - 1] = 0$. Das *Prüfpolynom* $h(D)$ ist dann zu

$$h(D) \cdot g(D) = D^n - 1\,; \qquad \text{Grad}\; h(D) = k \tag{3.57}$$

definiert. Mit Hilfe des Prüfpolynoms kann nun auch der gesamte Coderaum durch

$$\Gamma = \left\{ x(D) \in \mathrm{GF}_q[D]_{n-1} \mid R_{D^n - 1}\left[x(D) h(D) \right] = 0 \right\} \tag{3.58}$$

generiert werden, denn es gilt

$$x(D) = u(D)g(D) \qquad \Rightarrow \qquad x(D)\cdot h(D) = u(D)\cdot g(D)\cdot h(D) = u(D)\cdot (D^n - 1) \tag{3.59}$$

Aus dem Prüfpolynom kann eine Prüfmatrix erzeugt werden. Sie lautet:

$$\mathbf{H} = \begin{bmatrix} h_k & \dots & h_0 & & \\ & h_k & \dots & h_0 & \\ & & h_k & \dots & h_0 \end{bmatrix} \longleftrightarrow \begin{bmatrix} \bar{h}(D) \\ D\bar{h}(D) \\ \vdots \\ D^{n-k-1}\bar{h}(D) \end{bmatrix} \tag{3.60}$$

wobei $\bar{h}(D) = D^k h(D^{-1})$ das zu $h(D)$ reziproke Polynom repräsentiert. Durch die Multiplikation mit D^k enthält $\bar{h}(D)$ nur positive Exponenten.

Der zu einem durch $g(D)$ beschriebenen (n,k)-Code gehörende duale Code $\Gamma^\perp$ wird durch das reziproke Prüfpolynom $\bar{h}(D)$ generiert. Für den $(n,n-k)$-Code gilt:

$$g^\perp(D) = \bar{h}(D) = D^k h(D^{-1}) \tag{3.61}$$

$$h^\perp(D) = D^{n-k} g(D^{-1}) = \bar{g}(D) \tag{3.62}$$

3.6.4 Systematische Codierung mit Schieberegistern über das Generatorpolynom

Codierung mit Generatorpolynom
Ein großer Vorteil zyklischer Codes besteht unter anderem in der einfachen Codierung und Decodierung mit rückgekoppelten Schieberegistern. Hierdurch werden aufwendige Matrixmultiplikationen vermieden. Um einen systematischen Code zu erhalten, sind allerdings noch einige Ergänzungen erforderlich. Systematische Codes zeichnen sich dadurch aus, dass ein Codewort $\mathbf{x}$ sich in zwei Anteile, nämlich den Informationsteil $u(D)$ und den Prüfteil $p(D)$ zerlegen lässt.

$$\mathbf{x} = [p_0 \, \dots \, p_{n-k-1} \; u_0 \, \dots \, u_{k-1}] \Rightarrow x(D) = p(D) + D^{n-k}\cdot u(D) \tag{3.63}$$

Das Prüfpolynom $p(D)$ mit Grad $p(D) < n-k$ ist dann derart zu wählen, dass $x(D)$ sich ohne Rest durch $g(D)$ dividieren lässt. Wir erhalten

$$\begin{aligned} R_{g(D)}[x(D)] &= R_{g(D)}\left[p(D) + D^{n-k}\cdot u(D)\right] = R_{g(D)}[p(D)] + R_{g(D)}\left[D^{n-k}\cdot u(D)\right] \overset{!}{=} 0 \\ &\Rightarrow \quad p(D) = R_{g(D)}\left[-D^{n-k}\cdot u(D)\right] \end{aligned} \tag{3.64}$$

Die Division modulo $g(D)$ ist gleichbedeutend mit Substitution von D^{n-k} durch $-\sum_{i=0}^{n-k-1} g_i D^i$, weil für ein normiertes Generatorpolynom ($g_{n-k} = 1$) gilt:

$$\begin{aligned} x(D) &= u(D)\cdot g(D) = u(D)\cdot \sum_{i=0}^{n-k} g_i D^i \\ &= u(D)\cdot D^{n-k} + u(D)\cdot \sum_{i=0}^{n-k-1} g_i D^i \overset{!}{=} u(D)\cdot D^{n-k} + R_{g(D)}\left[-D^{n-k}\cdot u(D)\right] \\ \Longrightarrow R_{g(D)}\left[D^{n-k}\cdot u(D)\right] &= -u(D)\cdot \sum_{i=0}^{n-k-1} g_i D^i \end{aligned} \tag{3.65}$$

Realisierung mit Schieberegistern

Um zu zeigen, dass das in Bild 3.4 dargestellte rückgekoppelte Schieberegister tatsächlich auch die Generierung der Prüfstellen leistet, ist das Prüfpolynom $p(D)$ nach dem bekannten Horner-Schema zu zerlegen. Es gilt:

$$D^{n-k} \cdot u(D) = D^{n-k} \cdot u_0 + D\left(D^{n-k} \cdot u_1 + D\left(D^{n-k} \cdot u_2 + \ldots \quad \right) \quad \ldots \quad \right).$$

Mit der Abkürzung $R = R_{g(D)}$ lautet $p(D)$ dann

$$p(D) = \underbrace{R\left[-u_0 D^{n-k} + D \cdot \underbrace{R\left[-u_1 D^{n-k} + \ldots + D \cdot \underbrace{R\left[-u_{k-2} D^{n-k} + D \cdot \underbrace{R\left[-u_{k-1} D^{n-k}\right]}_{r^{(1)}(D)}\right]}_{r^{(2)}(D)} \ldots\right]}_{r^{(k-1)}(D)}\right]}_{r^{(k)}(D)}$$

Für Teilpolynome $r^{(i)}(D)$ gilt

$$\begin{aligned} r^{(0)}(D) &= 0 \\ r^{(i)}(D) &= R\left[-u_{k-i} D^{n-k} + D \cdot r^{(i-1)}(D)\right] \\ r^{(k)}(D) &= p(D)\,. \end{aligned}$$

Die Ausnutzung von Gl. (3.65) für i-tes Polynom $r^{(i)}(D)$ liefert schließlich

$$\begin{aligned} r^{(i)}(D) &= \sum_{j=0}^{n-k-1} r_j^{(i)} \cdot D^j \\ &= R\left[-u_{k-i} \cdot D^{n-k} + D \sum_{j=0}^{n-k-1} r_j^{(i-1)} \cdot D^j\right] \\ &= R\left[-u_{k-i} \cdot D^{n-k} + \sum_{j=1}^{n-k-1} r_{j-1}^{(i-1)} \cdot D^j + r_{n-k-1}^{(i-1)} \cdot D^{n-k}\right] \\ &= \sum_{j=1}^{n-k-1} r_{j-1}^{(i-1)} \cdot D^j + R\left[\left(-u_{k-i} + r_{n-k-1}^{(i-1)}\right) \cdot D^{n-k}\right] \\ &= \sum_{j=1}^{n-k-1} r_{j-1}^{(i-1)} \cdot D^j + \left(-u_{k-i} + r_{n-k-1}^{(i-1)}\right) \cdot \left(-\sum_{j=0}^{n-k-1} g_j \cdot D^j\right) \\ &= \sum_{j=0}^{n-k-1} \left[r_{j-1}^{(i-1)} - g_j \cdot \left(-u_{k-i} + r_{n-k-1}^{(i-1)}\right)\right] \cdot D^j \quad ; \quad r_{-1}^{(i-1)} = 0\,. \end{aligned}$$

Vergleich von obiger Gleichung und Bild 3.4 zeigt, dass Polynome $r^{(i)}(D)$ die Registerinhalte zum i-ten Takt beschreiben, d.h. $r_j^{(i)}$ ist Inhalt der j-ten Speicherstelle zum i-ten Takt

Interpretation von obiger Gleichung:

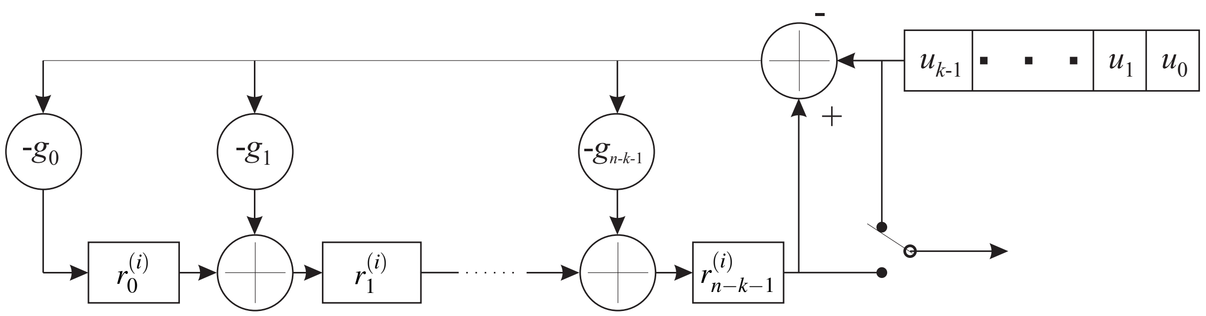


Bild 3.4: Allgemeine Schieberegisterrealisierung mit $g(D)$

$i=1$ Informationssymbol u_{k-1} negieren und in einzelnen Zweigen des Schieberegisters mit negativen Koeffizienten $-g_j$ des Generatorpolynoms multiplizieren (Initialisierung mit $r_{n-k-1}^{(i-1)}=0$)
Potenz des Platzhalters D gibt den jeweiligen Zweig des Registers an
In Speicherstelle $r_j^{(1)}$ des Registers steht $u_{k-1}\cdot g_j$

$i=2$ Differenz zwischen nächstem Informationssymbol u_{k-2} und altem Inhalt der $n-k$-ten Speicherstelle $r_{n-k-1}^{(1)}$ bilden
Ergebnis in einzelnen Zweigen mit negativen Koeffizienten des Generatorpolynoms multiplizieren
Addition der Produkte zu alten Registerinhalten $r_{j-1}^{(1)}$

$i>2$ Vorgang wiederholen bis alle k Informationssymbole abgearbeitet

$i=k$ Schieberegister enthält Koeffizienten des Prüfpolynoms $p(D)$
Koeffizienten auslesen und dem systematischen Informationsteil voranstellen

Beispiel: (7,4)-Hamming-Codes (s. Bild 3.5)
Der schon bekannte (7,4)-Hamming-Codes lässt sich auch mit einem zyklischen Codierer realisieren. Bild 3.5 zeigt die zugehörige Schieberegisterstruktur. Da es sich hier um einen binären Code (GF(2)) handelt, sind Addition und Subtraktion identisch, und die Minuszeichen können entfallen.

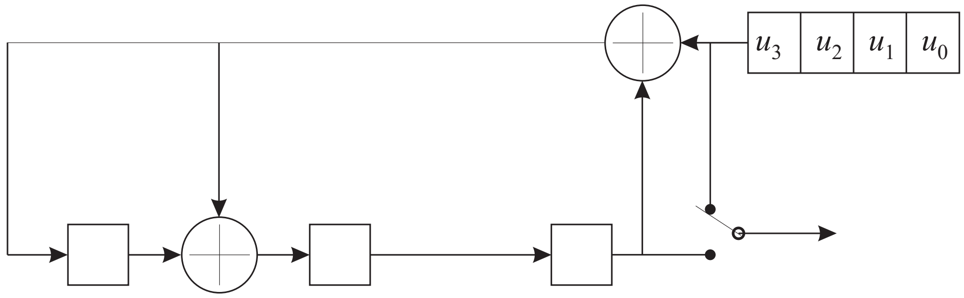


Bild 3.5: Schieberegisterrealisierung für $(7,4)_2$-Hamming-Code mit $g(D)=1+D+D^3$

3.6.5 Systematische Codierung über Prüfpolynom

Entsprechend der Beschreibung zyklischer Codes durch ihr Prüfpolynom $h(D)$ lässt sich auch eine Schieberegisterstruktur angeben, die auf $h(D)$ basiert. Sie ist in ihrer allgemeinen Form in Bild 3.6 dargestellt. Wenn wir ein Codewort $\mathbf{x}$ in der Form

$$\mathbf{x}=[\underbrace{x_0\,x_1\,\ldots\,x_{n-k-1}}_{n-k\text{ Prüfstellen}}\ \underbrace{x_{n-k}\,\ldots\,x_{n-1}}_{k\text{ Infostellen}}] \tag{3.66}$$

erzeugen wollen, dann berechnen sich die $n-k$ Prüfsymbole entsprechend der Vorschrift ($m = n-k-1, \ldots, 0$)

$$x_m = -\sum_{i=0}^{k-1} h_i \cdot x_{m-i+k} = f\left(x_{m+1}, \ldots, x_{m+k}\right). \tag{3.67}$$

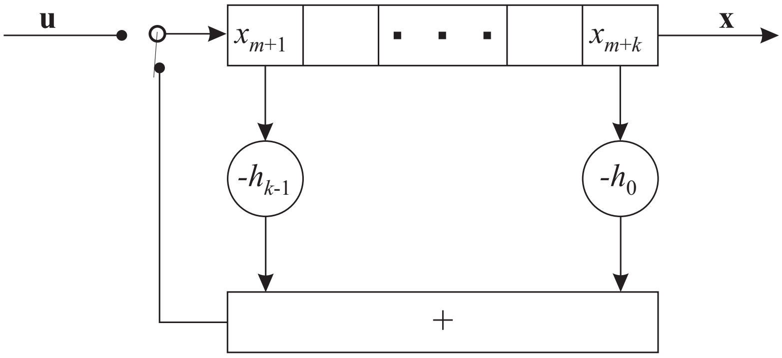


Bild 3.6: Allgemeine Schieberegisterrealisierung mit $h(D)$

Beispiel: (7,4)-Hamming-Code (s. Bild 3.7)
Für den schon oben benutzten (7,4)-Hamming-Code nimmt das Schieberegister die in Bild 3.7 gezeigte Form an. Das Prüfpolynom lautet hier $h(D) = D^4 + D^2 + D + 1$.

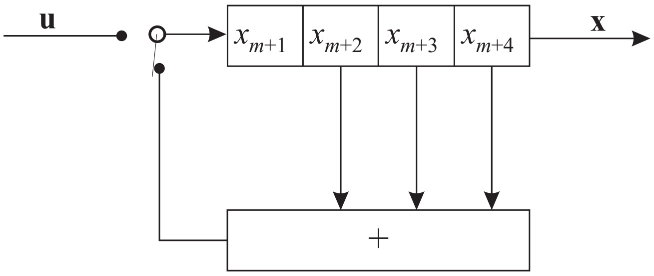


Bild 3.7: Schieberegisterrealisierung für $(7,4)_2$-Hamming-Code mit $h(D) = D^4 + D^2 + D + 1$

3.6.6 Bestimmung des Syndroms

In Abschnitt 3.5 wurde bei der Matrixbeschreibung von Blockcodes das Syndrom $\mathbf{s}$ definiert (vgl. Gleichung 3.39). Mit Hilfe von $\mathbf{s}$ ist die Erkennung und sogar die Korrektur von Übertragungsfehlern möglich.

Anstelle des Syndromvektors wird bei den zyklischen Codes das so genannte Syndrompolynom $s(D) = s_0 + s_1 D + \ldots + s_{n-k-1} D^{n-k-1}$ definiert, so dass auch dieses mit Hilfe eines Schieberegisters zu erzeugen ist.

Allgemein gilt: $\mathbf{y} = \mathbf{x} + \mathbf{e}$ bzw. $y(D) = x(D) + e(D)$

Berechnung des Syndroms aus der Division des Polynoms $y(D)$ modulo dem Generatorpolynom $g(D)$

$$R_{g(D)}\left[y(D)\right] = \underbrace{R_{g(D)}\left[x(D)\right]}_{=0} + \underbrace{R_{g(D)}\left[e(D)\right]}_{=s(D)} \tag{3.68}$$

Bild 3.8 zeigt die zugehörige Struktur des Schieberegisters. Das empfangene Wort $\mathbf{y}$ wird symbolweise, beginnend mit y_{n-1}, in das Register geschoben. Nach n Takten enthalten die Speicherelemente des Registers die Koeffizienten des Syndroms.

Es ist zu beachten, dass das Syndrompolynom $s(D)$ nicht notwendigerweise mit dem Syndromvektor übereinstimmen muss, da die Prüfmatrix $\mathbf{H}$ nicht eindeutig bestimmt ist.

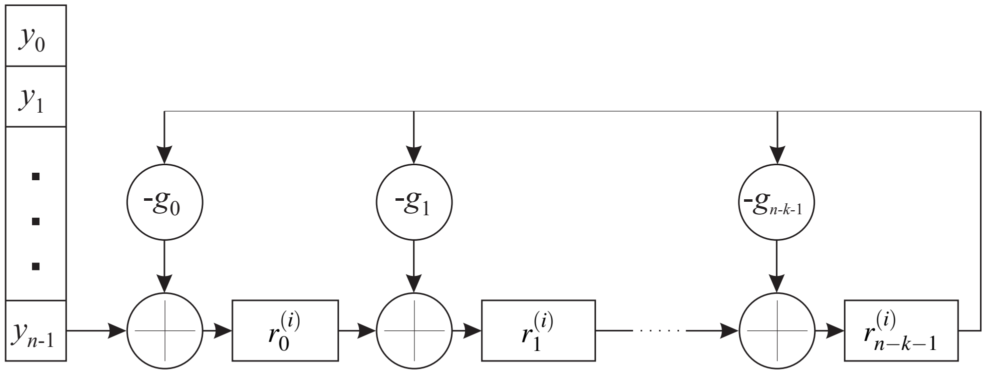


Bild 3.8: Schieberegisterrealisierung zur Berechnung des Syndroms

3.6.7 Erkennung von Einzel- und Bündelfehlern (CRC-Codes)

Voraussetzung: Hard-Decision am Kanalausgang
Bei reiner Fehlererkennung unterscheiden wir zwei Fälle:

1. $s(D) = 0$:
   - Es liegt kein Fehler vor.
   - Der Fehler ist nicht erkennbar (Durch den Fehler ist wieder ein gültiges Codewort entstanden.)
2. $s(D) \neq 0$: Es liegt ein Fehler vor.

Die Leistungsfähigkeit eines Codes, Fehler zu erkennen, hängt zwar unmittelbar von dessen Distanzeigenschaften ab, allerdings hat auch die Struktur der Fehler selbst entscheidenden Einfluss. Prinzipiell unterscheiden wir zwei Arten von Fehlerereignissen: Einzelfehler und Bündelfehler. Letztere werden charakterisiert durch ihre Länge t und die Anzahl der fehlerhaften Symbole.

Bündelfehler:

Ein Bündelfehler der Länge t zeichnet sich dadurch aus, dass t aufeinanderfolgende Symbole mit hoher Wahrscheinlichkeit falsch sind. Dies bedeutet aber nicht, dass nicht auch korrekte Symbole unter ihnen sein können. Somit gilt für das Gewicht des Fehlervektors $\mathbf{e}$ *bei einem Bündelfehler der Länge t:* $w_{\mathrm{H}}(\mathbf{e}) \leq t$.
Es existieren auch zyklische Bündelfehler, d.h. ein Fehler am Wortende kann sich am Wortanfang fortsetzen.

Fehlererkennung

Jeder zyklische $(n,k)_q$*-Code erkennt alle Bündelfehler bis zur Länge* $t' \leq n-k$. *Von den Bündelfehlern größerer Länge erkennt er nur eine Quote von*

$$P_{\mathrm{ue}} = \begin{cases} \frac{q^{-(n-k-1)}}{q-1} & \text{für} \quad t' = n-k+1 \\ q^{-(n-k)} & \text{für} \quad t' \geq n-k+2 \end{cases} \tag{3.69}$$

nicht!

- Gute Eignung zyklischer Codes zur Erkennung von Bündelfehlern

- Codes werden speziell an typische Fehlerstrukturen des Kanals angepasst

→ Zyklische Codes sind sehr leistungsfähig Codes bei der Erkennung von Bündelfehlern, aber schlecht bei der Erkennung von Einzelfehlern

- Andere Codes sehr sensitiv gegenüber bündelartigen Störungen, aber gut bei Einzelfehlern (z.B. Faltungscodes)

Äquivalente Codes

- Äquivalente Codes (Vertauschen von Spalten der Generatormatrix) besitzen identische Distanzeigenschaften
- Aber: nicht unbedingt gleiche Leistungsfähigkeit bei Behandlung von Bündelfehlern
- Grund:
  - Zyklische Codes erkennen selbst Bündelfehler, deren Länge größer als die Minimaldistanz ist
  - Durch Spaltentausch kann äquivalenter, aber nicht-zyklischer Code entstehen

  → Keine Erkennung langer Bündelfehler mehr
- Bei Einzelfehlern verhalten sich äquivalente Codes nahezu identisch.

Definition: CRC-Codes (*Cyclic Redundancy Check*)

CRC-Codes sind zyklische $(2^r-1, 2^r-r-2, 4)_2$*-Codes, deren Generatorpolynom die Form*

$$g(D) = (1+D)\cdot p(D) \tag{3.70}$$

besitzt, wobei $p(D)$ *ein primitives Polynom vom Grad* r *ist.*

Eigenschaften von CRC-Codes:

- Alle Fehlermuster mit $w_{\mathrm{H}}(\mathbf{e}) \leq 3$ werden erkannt.
- Alle Fehler mit ungeradem Gewicht werden erkannt.
- Alle Bündelfehler bis zur Länge $r+1$ werden erkannt.
- Von den Bündelfehlern mit einer Länge von $r+2$ wird nur eine Quote von 2^{-r} nicht erkannt
- Von den Bündelfehlern mit einer Länge von $\geq r+3$ wird nur eine Quote von $2^{-(r+1)}$ nicht erkannt.

Beispiel: CRC-Code mit 16 Prüfbits
Ein CRC-Code mit $n-k=16$ Prüfbits besitzt ein primitives Polynom vom Grad $r=n-k-1=15$ und erkennt die folgenden Fehlerereignisse

- 100 % aller Bündelfehler der Länge ≤ 16,
- 99,9969 % der Bündelfehler der Länge 17,
- 99,9985 % der Bündelfehlern mit einer Länge ≥ 18.

Somit sind CRC-Codes prädestiniert für die Erkennung von Bündelfehlern. Die durch das Generatorpolynom erzeugten Prüfbit werden in der Literatur auch häufig als *Frame Check Sequence* (FCS) bezeichnet. Anwendung finden CRC-Codes beispielsweise in den Mobilfunknetzen nach dem GSM-Standard, auf die im Laufe der Vorlesung noch genauer eingegangen wird.

3.6.8 Korrektur von Einzel- und Bündelfehlern (Fire-Codes)

- Korrektur von Bündelfehlern erfordert, dass jedes zu korrigierende Fehlermuster eindeutig einem Syndrom zugeordnet ist
- D.h. für zwei beliebige Codeworte $\mathbf{x}$ und $\mathbf{x}'$ und zwei beliebige Fehlervektoren $\mathbf{e}$ und $\mathbf{e}'$ gilt stets $\mathbf{x}+\mathbf{e} \neq \mathbf{x}'+\mathbf{e}'$
- Ein Empfangswort darf nie in mehrfacher Weise aus $\mathbf{y} = \mathbf{x}+\mathbf{e}$ hervorgehen
- Entwurf zyklischer Blockcodes:
  - Anstelle der Maximierung der Minimaldistanz vielmehr Erkennung bestimmter Fehlermuster wichtig
  - $\rightarrow$ Bei Nebenklassenzerlegung nicht Fehlerworte mit minimalem Gewicht Anführer der Nebenklassen, sondern die Fehlerworte mit typischen Bündelfehlermustern
  - $\rightarrow$ Automatisch Korrektur von Bündelfehlern

Reiger-Schranke

Soll ein zyklischer Code Bündelfehler bis zur Länge t korrigieren, so muss gelten:

$$t \leq \lfloor \frac{n-k}{2} \rfloor \qquad \textit{oder} \qquad 2t \leq n-k\,. \tag{3.71}$$

Diese Forderung ist gleichbedeutend mit der Aussage, dass die Anzahl der Syndrome (q^{n-k} darstellbare Fehlermuster) größer gleich der Anzahl zu korrigierender Fehlermuster sein muss. Letztere lautet

$$L_t = \begin{cases} \sum_{r=0}^{t} \binom{n}{r} (q-1)^r & \leq t \text{ Einzelfehler} \\ \\ 1+n\,(q-1)\,q^{t-1} & \text{zykl. Bündelfehler, Länge } \leq t \end{cases} \tag{3.72}$$

Ein klassisches Beispiel für sehr gute Korrektureigenschaften von Bündelfehlern sind die Fire-Codes. Sie lassen sich analytisch konstruieren, wodurch eine aufwendige Suche entfällt.

Definition Fire-Codes

Ist $p(D)$ ein primitives Polynom vom Grad m und $t < m$ eine Zahl, so dass $D^{2t-1}-1$ kein Vielfaches von $p(D)$ ist, so stellt

$$g(D) = \left(D^{2t-1}-1\right) \cdot p(D) \tag{3.73}$$

das Generatorpolynom des $(n,k)_q$-Fire-Codes dar, wobei $n = (2t-1)(q^m-1)$ gilt. Der Fire-Code korrigiert Bündelfehler bis zur Länge t.

3.6.9 Algebraische und nicht-algebraische Decodierung

Die Decodierung zyklischer Codes läuft prinzipiell immer nach dem gleichen Schema ab.

1. Berechnung des Syndroms $s(D) = R_{g(D)}[y(D)]$ mit Hilfe von rückgekoppelten Schieberegistern

2. Bestimmung des Fehlermusters $e(D)$ aus dem Syndrom $s(D)$. Dieser Teil ist am aufwändigsten und lässt sich in einer trivialsten Form durch das Speichern aller möglichen Syndrome und der dazugehörigen Fehlermuster realisieren. Dies erfordert aber einen enormen Speicherbedarf und ist daher nur für die einfachsten Codes realisierbar.

3. Fehlerkorrektur durch $\hat{x}(D) = y(D) - e(D)$.

Der zweite Punkt ist unter Umständen für zyklische Codes sehr einfach ohne Tabelle zu lösen. So ermöglichen die im nächsten Abschnitt beschriebenen Reed-Solomon- und BCH-Codes eine sehr effiziente, wenn auch mathematisch anspruchsvolle Decodierung.

Algebraische Decodierung

- In der Regel mathematisch sehr anspruchsvolle Verfahren
- Bei Verwendung von BMD anstelle von MLD ergeben sich drastische Vereinfachungen (s. Reed- Solomon-Codes, BCH-Codes)

Nicht-algebraische Decodierung

- Bestimmung des Syndroms unter Ausnutzung der Codestrukturen
- Anschauliche Bestimmung des Fehlervektors aus dem Syndrom
- Bekannte Verfahren: Permutationsdecodierer, Meggit-Decodierer, Majoritätsdecodierer, Schwellwertdecodierer, u.a.
- Aufwand steigt mit der Anzahl zu korrigierender Fehler stark an

Prinzipielle Vorgehensweise:

- Durch zyklische Verschiebung endet jedes Fehlermuster bei der höchsten Potenz D^{n-1}
- Bei L_{end} solcher Fehlermuster nur L_{end} Fehlermuster abspeichern

$\rightarrow$ Enorme Einsparung an Speicherbedarf

- Iterative Berechnung des Syndroms aus den zyklisch verschobenen Empfangsworten:
  - Bestimmung des Syndroms vom Empfangswort $y(D)$
  - Ist dieses nicht in der Liste von L_{end} Syndromen enthalten, $y(D)$ um eine Stelle zyklisch verschieben
  - Syndrom von $R_{D^n-1}[Dy(D)]$ berechnet
  - Prozedur solange wiederholen, bis entweder passendes Syndrom gefunden und Fehler korrigiert oder n-te Verschiebung erreicht
  - Im letzten Fall kein erkennbares Fehlermuster

3.7 Reed-Solomon- und BCH-Codes

3.7.1 Einführung

Bei der Konstruktion der bisher vorgestellten Codes kann nur bedingt eine bestimmte Korrekturfähigkeit, d.h. ein bestimmtes Distanzspektrum, vorgegeben werden. Dies ist aber für die Entwicklung eines leistungsfähigen Kanalcodes wünschenswert. Die in diesem Abschnitt beschriebenen Reed-Solomon- und BCH-Codes wurden ca. im Jahr 1960 entwickelt. Sie besitzen gegenüber den bisher behandelten Verfahren eine ganze Reihe von Vorteilen, von denen hier die wichtigsten aufgezählt werden sollen.

- Es existiert eine analytisch geschlossene Konstruktionsvorschrift.
- Bei der Konstruktion kann die Minimaldistanz und somit auch die Korrekturfähigkeit vorgegeben werden, bei den RS-Codes sogar das gesamte Gewichtsspektrum.
- Reed-Solomon (RS)-Codes sind MDS-Codes, d.h. die Gleichheit der Singleton-Schranke ist erfüllt und die tatsächliche Mindestdistanz ist somit gleich der Entwurfsdistanz
- Beide Codefamilien sind für nicht zu große Blocklängen n sehr leistungsfähig.
- Es ist eine gute Anpassung an den Kanal möglich:
  - RS-Codes eignen sich hervorragend zur Korrektur von Bündelfehlern.
  - Binäre BCH-Codes eignen sich besser zur Korrektur von Einzelfehlern.
- Für beide Codefamilien ist eine effiziente Decodierung nach dem BMD-Prinzip möglich.
- Außerdem erlauben die Decodieralgorithmen eine einfache Ausnutzung von Ausfallinformation, d.h. die einfachste Form von *Soft*-Information (s. BSEC).

Da sich sowohl RS- als auch BCH-Codes sehr anschaulich im 'Spektralbereich' beschreiben lassen, soll im Folgenden die Spektraltransformation vorgestellt werden.

3.7.2 Spektraltransformation auf Galoisfeldern

Die Spektraltransformation auf Galoisfeldern ist äquivalent zur Fouriertransformation komplexer Signale. Sie ist weder bei der Codierung noch bei der Decodierung explizit auszuführen sondern dient lediglich der anschaulichen Beschreibung der Codes.

Definition:
Gegeben seien zwei Polynome mit Koeffizienten aus GF(p^m).

$$\begin{aligned} a(D) &= \sum_{i=0}^{n-1} a_i \cdot D^i \quad \leftrightarrow \quad (a_0,\, a_1,\, \ldots\, a_{n-1}) \\ A(D) &= \sum_{i=0}^{n-1} A_i \cdot D^i \quad \leftrightarrow \quad (A_0,\, A_1,\, \ldots\, A_{n-1}) \end{aligned}$$

Dann gilt für die spektrale Hin- und Rücktransformation für $p = 2$:

$$A(D) = DFT\,(a(D)) \qquad \rightarrow \qquad A_i = a\left(z^{-i}\right) = \sum_{\mu=0}^{n-1} a_\mu \cdot z^{-i\mu} \tag{3.74}$$

$$a(D) = IDFT\,(A(D)) \qquad \rightarrow \qquad a_i = A\left(z^{i}\right) = \sum_{\mu=0}^{n-1} A_\mu \cdot z^{i\mu} \tag{3.75}$$

Wichtig: Korrespondenz zwischen Vektoren und Polynomen

- z^{-i} ist Nullstelle von $a(D)$ $\iff$ i-te Komponente von $\mathbf{A}$ ist Null ($A_i = 0$)
- z^{i} ist Nullstelle von $A(D)$ $\iff$ i-te Komponente von $\mathbf{a}$ ist Null ($a_i = 0$)

Zyklische Faltung: geht über in komponentenweise Multiplikation

$$c(D) = R_{D^n-1}[a(D) \cdot b(D)] \quad \Longleftrightarrow \quad C_i = -A_i \cdot B_i \tag{3.76}$$

$$C(D) = R_{D^n-1}[A(D) \cdot B(D)] \quad \longleftrightarrow \quad c_i = a_i \cdot b_i \tag{3.77}$$

Zyklische Verschiebung um b Stellen:

$$c(D) = R_{D^n-1}\left[D^b \cdot a(D)\right] \quad \Longleftrightarrow \quad C_i = z^{-ib} \cdot A_i \tag{3.78}$$

$$C(D) = R_{D^n-1}\left[D^b \cdot A(D)\right] \quad \longleftrightarrow \quad c_i = z^{ib} \cdot a_i \tag{3.79}$$

3.7.3 Definition von Reed-Solomon-Codes

Das Ziel bei der Konstruktion von Codes ist es, für eine vorgegebene Blocklänge n und eine bestimmte Coderate $R_c = k/n$ einen Code Γ zu finden, der z.B. t Fehler korrigieren kann. Dazu muss Γ eine Mindestdistanz von $d_{\min} = 2t+1$ besitzen. Um den Weg zur Konstruktion eines solchen Codes zu veranschaulichen, betrachten wir ein Polynom

$$X(D) = X_0 + X_1 D + \ldots + X_{k-1} D^{k-1} \tag{3.80}$$

vom Grad $\leq k-1$ mit Koeffizienten $X_i \in \mathrm{GF}(q) = \mathrm{GF}(p^m)$. Aus der Algebra wissen wir, dass $X(D)$ maximal $k-1$ Nullstellen haben kann (Faktorisierung). Wir nehmen nun n verschiedene Elemente $z_0, \ldots, z_{n-1} \in \mathrm{GF}(q)$ ungleich Null und bilden ein Codewort $\mathbf{x}$ der Länge n durch

$$\mathbf{x} = [x_0\, x_1\, \ldots\, x_{n-1}] \qquad \text{mit} \qquad x_i = X(z_i)\,. \tag{3.81}$$

Dann hat $\mathbf{x}$ mindestens das Gewicht $w_{\mathrm{H}}(\mathbf{x}) \geq d = n-k+1$. Diese Behauptung lässt sich einfach beweisen. $X(D)$ hat maximal $k-1$ Nullstellen, dann besitzt $x(D)$ nach Abschnitt 3.7.2 maximal ebenso viele Koeffizienten x_i gleich Null. Da $x(D)$ aber insgesamt aus n Koeffizienten besteht, müssen zwangsläufig $n-(k-1)$ Koeffizienten ungleich Null sein, womit die Behauptung bewiesen ist.

Bilden wir nun einen Code Γ, in dem wir alle $q^k = p^{mk}$ Polynome $X(D)$ und Gl. (3.81) verwenden, so besitzt Γ aufgrund seiner Linearität die Mindestdistanz $d_{\min} = n-k+1$. Wir haben somit eine Möglichkeit gefunden, Codes mit bestimmten Mindestdistanzen zu entwerfen.

Definition RS-Code

> *Gegeben sei ein Element $z \in \mathrm{GF}(q)$ der Ordnung n, d.h. n ist die kleinste Zahl mit $z^n = 1$. Ein $(n,k,d_{\min})$-Reed-Solomon-Code Γ ist definiert als die Menge aller Polynome $X(D)$ mit $\mathrm{grad}\,X(D) \leq k-1$ und $X_i \in \mathrm{GF}(q)$. Die Codeworte $\mathbf{x} \in \Gamma$ werden durch die Beziehung $x_i = X(z^i)$ gebildet. Die Mindestdistanz des Codes beträgt $d_{\min} = n-k+1$.*

In der Regel ist z ein primitives Element von $\mathrm{GF}(q)$ und hat damit die Ordnung $n = q-1 = p^m-1$. RS-Codes besitzen dann folgende Parameter:

- Codewortlänge: $n = p^m - 1 \rightarrow$ die Wahl des GF bestimmt die Codewortlänge!
- Dimension des Codes beträgt $|\Gamma| = q^k = p^{mk}$ mit $k = p^m - d_{\min}$ (folgt aus $d = n-k+1 = p^m - k$)
- Coderate: $R_c = \frac{k}{n} = \frac{p^m - d_{\min}}{p^m - 1}$

Aus der Beziehung $z^n = 1$ folgt sofort $z^n - 1 = 0$. Damit erfüllen alle Potenzen von z die Gleichung $D^n - 1 = 0$. Da die n Potenzen von z alle verschieden voneinander sind, gilt

$$D^n - 1 = \prod_{i=0}^{n-1} (D - z^i) \tag{3.82}$$

Generator- und Prüfpolynom für RS-Codes
Bekanntlich gilt für zyklische Codes

$$x(D) = u(D) \cdot g(D) \qquad \text{mit} \qquad \text{grad}\, g(D) \leq n-k\ .$$

Die Spektraltransformation bzgl. $x(D)$ liefert uns $X(D)$, das laut Definition der RS-Codes nur den Grad $k-1$ besitzt. Damit müssen die übrigen $n-k$ Koeffizienten Null sein, d.h. es gilt

$$X_i = 0 \quad \text{für} \quad k \leq i \leq n-1\ . \tag{3.83}$$

RS-Codes zeichnen sich also dadurch aus, das die Codeworte im Spektralbereich $n-k = d-1$ aufeinanderfolgende Nullstellen besitzen, welche **Paritätsfrequenzen** genannt werden. Für den Originalbereich gilt wegen der Beziehung $X_i = x(z^{-i})$, dass $n-k$ aufeinanderfolgende Potenzen von z Nullstellen des Polynoms $x(D)$ sind.

$$x(z^{-k}) = \ldots = x(z^{-(n-1)}) = 0 \tag{3.84}$$

Da Gl. (3.84) für alle Codewortpolynome $x(D)$ unabhängig von $u(D)$ gelten muss, muss das Generatorpolynom $g(D)$ Gl. (3.84) erfüllen. Somit lässt es sich in $n-k$ Terme faktorisieren

$$g(D) = \prod_{i=k}^{n-1}(D - z^{-i}) = \prod_{i=k}^{n-1}(D - z^{-i} \cdot z^n) = \prod_{i=1}^{n-k}(D - z^i) \tag{3.85}$$

Für das Prüfpolynom folgt mit $g(D)h(D) = D^n - 1$ und Gl. (3.82)

$$h(D) = \prod_{i=0}^{k-1}(D - z^{-i}) = \prod_{i=0}^{k-1}(D - z^{-i} \cdot z^n) = \prod_{i=n-k+1}^{n}(D - z^i)\ . \tag{3.86}$$

Verallgemeinerung der Definition:
Die Nullstellen des Generatorpolynoms $g(D)$ bzw. die Nullsymbole von $X(D)$ müssen nicht an den oben erläuterten Stellen liegen, sondern können an beliebigen Positionen auftreten, solange sie zusammenhängend bleiben (siehe [Fri96]). Der Coderaum Γ kann dann wie folgt definiert werden.

$$\Gamma = \left\{ x(D) \mid X(D) = R_{D^n-1}\left[D^{n-k+\ell} \cdot B(D) \right] \right\} \quad \text{mit Grad } B(D) \leq k-1 \tag{3.87}$$

- **Interpretation:** $X_\ell = X_{\ell+1} = \ldots = X_{\ell+d-2} = 0$ bzw. z^{-l}, $z^{-(\ell+1)}$, $\ldots$ $z^{-(\ell+d-2)}$ sind Nullstellen des Polynoms $x(D)$
- Parameter ℓ beschreibt Lage der ersten von $n-k$ aufeinanderfolgenden Nullstellen
(für Distanzeigenschaften der RS-Codes eher von untergeordneter Bedeutung)
  - $\ell = 1: \quad \Gamma = \left\{ x(D) \mid x(z^{-1}) = x(z^{-2}) = \cdots x(z^{-(d-1)}) = 0 \right\}, X_1 = X_2 = \ldots = X_{d-1} = 0$
  - $\ell = k: \quad \Gamma = \{ x(D) \mid \ \text{grad}\, X(D) \leq k-1 \}$

Reed-Solomon-Codes gehören zu den wenigen Codes, deren komplette Gewichtsverteilung geschlossen angegeben werden kann. Es gilt

$$A_d = \binom{n}{d}(q-1) \sum_{j=0}^{d-d_{\min}} (-1)^j \binom{d-1}{j} q^{d-d_{\min}-j} \tag{3.88}$$

RS-Codes sind MDS-Codes $\longrightarrow$ Entwurfsdistanz d ist gleich der Minimaldistanz $d_{\min}$. Es gilt der Zusammenhang

$$d_{\min} = d = n-k+1 = p^m - k\ . \tag{3.89}$$

3.7.4 Beispiele für RS-Codes

Wir wollen im Folgenden RS-Codes über dem Erweiterungskörper GF(8) mit $p = 2$ und $m = 3$ betrachten. Damit beträgt die Codewortlänge $n = q - 1 = 2^3 - 1 = 7$ Symbole, die jeweils $q = 8$ Werte annehmen können. Für den Erweiterungskörper GF(8) existieren zwei primitive Polynome, von denen wir das Polynom $p(D) = D^3 + D + 1$ verwenden wollen. Damit gilt für das primitive Element z die Nebenbedingung $z^3 = z + 1$ und folgende Zusammenhänge:

$$\begin{array}{lllllll}
z^1 & = & & & & = & z \\
z^2 & = & & & & = & z^2 \\
z^3 & = & z+1 & & & = & z+1 \\
z^4 & = & z^2+z & & & = & z^2+z \\
z^5 & = & z^3+z^2 & = & z+1+z^2 & = & z^2+z+1 \\
z^6 & = & z^4+z^3 & = & z^2+z+z+1 & = & z^2+1 \\
z^7 & = & z^5+z^4 & = & z^2+z+1+z^2+z & = & 1 = z^0
\end{array} \tag{3.90}$$

Die Elemente im obigen Erweiterungskörper lauten entsprechend

$$\mathrm{GF}(8) = \{0,\, 1,\, z,\, 1+z,\, z^2,\, 1+z^2,\, z+z^2,\, 1+z+z^2\} = \{0,\, 1,\, z,\, z^2,\, z^3,\, z^4,\, z^5,\, z^6\}\,, \tag{3.91}$$

stellen also alle Polynome vom maximalen Grad $m - 1 = 2$ dar und können auch als Tripel bestehend aus 3 Binärstellen beschrieben werden.

Beispiel 1

Es soll für den obigen Erweiterungskörper ein Reed-Solomon-Code konstruiert werden, der $t = 1$ Fehler korrigieren kann. Daraus ergeben sich folgende Codeparameter:

- Die Mindestdistanz des RS-Codes beträgt $d_{\min} = 3$ (wegen $t = 1$ und $d_{\min} = 2t + 1$)
- Länge des Codes: $n = p^m - 1 = 2^3 - 1 = 7$
- Dimension des Codes: $k = p^m - d = 2^3 - 3 = 5$

$\rightarrow$ Coderate: $R_c = \frac{k}{n} = \frac{5}{7} \approx 0.714$

- Wir erhalten einen $(7,5,3)_8$-Code mit $8^5 = 32.768$ Codeworten
- Zwei beliebige Codeworte unterscheiden sich um mindestens 3 Symbole
- Bei binärer Repräsentation der 3-Bit-Codesymbole ergibt sich $(21,15,3)_2$-Code
- Minimaldistanz bleibt bei dieser Betrachtung unverändert, da zwei 3-Bit-Symbole auch dann schon verschieden sind, wenn sie sich nur durch 1 Bit unterscheiden
- Nullstellen des Generatorpolynoms besitzen $d - 1 = 2$ aufeinanderfolgende Exponenten:

$$g(D) = (D - z^1) \cdot (D - z^2) = D^2 + \underbrace{(z + z^2)}_{z^4} D + z^3 = D^2 + z^4 D + z^3$$

- Prüfpolynom:

$$h(D) = (D - z^3) \cdot (D - z^4) \cdot (D - z^5) \cdot (D - z^6) \cdot (D - z^7) = D^5 + z^4 D^4 + D^3 + z^5 D^2 + z^5 D + z^4$$

Beispiel 2 für RS-Codes:

Mit dem gleichen Erweiterungskörper sollen nun $t = 2$ Fehler korrigiert werden $\rightarrow d_{\min} = 5$. Hieraus ergeben sich die Codeparameter:

- Länge des Codes: $n = 2^3 - 1 = 7$
- Dimension des Codes: $k = 2^m - d = 2^3 - 5 = 3$
  $\longrightarrow$ Der Code besteht aus $q^k = (2^3)^3 = 512$ Codeworten

$\rightarrow$ Coderate: $R_c = \frac{k}{n} = \frac{3}{7} \approx 0.429$

- Nullstellen des Generatorpolynoms besitzen $d-1 = 4$ aufeinanderfolgende Exponenten:

$$g(D) = (D - z^1) \cdot (D - z^2) \cdot (D - z^3) \cdot (D - z^4) = D^4 + z^3 D^3 + D^2 + zD + z^3$$

- Prüfpolynom:

$$h(D) = (D - z^5) \cdot (D - z^6) \cdot (D - z^7) = D^3 + z^3 D^2 + z^5 D + z^4$$

3.7.5 Definition von BCH-Codes

Der große Unterschied zwischen den im letzten Abschnitt behandelten RS-Codes und den Bose-Chaudhuri-Hocquenghem- (BCH)-Codes besteht darin, dass RS-Codes in der Regel als nicht-binäre Codes mit $x_i \in \mathrm{GF}(p^m)$ eingesetzt werden, während für BCH-Codes normalerweise $x_i \in \mathrm{GF}(p)$ gilt. Trotzdem sei darauf hingewiesen, dass BCH-Codes durchaus auch nicht-binär sein können und RS-Codes wiederum auch im $\mathrm{GF}(p)$ definiert sind. Wir wollen im Folgenden allerdings den Spezialfall $p = 2$ betrachten, für den BCH-Codes rein binäre Codes sind. Zuvor sind jedoch noch die *Kreisteilungsklassen* zu definieren.

Definition Kreisteilungsklassen (*Splitting Fields*)

Die Zahlenmenge $\{0, 1, \ldots, n-1\}$ kann in disjunkte Teilmengen aufgeteilt werden, sogenannte Kreisteilungsklassen. Es gilt für $n = q^\ell - 1$ mit $q = p^m$

$$K_i = \left\{ i \cdot q^j \bmod n\,,\ j = 0, 1, \ldots, \ell - 1 \right\}\ , \tag{3.92}$$

wobei i das kleinste Element in K_i ist.

Die Kreisteilungsklassen K_i haben folgende Eigenschaften:

- $|K_i| \leq \ell$ — maximal ℓ Elemente je Klasse
- $K_i \bigcap K_{j \neq i} = \emptyset$ — also ist kein Element in mehreren Klassen
- $K_0 = \{0\}$
- $\bigcup_i K_i = \{0, 1, \ldots, n-1\}$ — die Vereinigung aller Klassen enthält alle Elemente $0, \ldots, n-1$

Für die Konstruktion von BCH-Codes ist eine weitere wesentliche Eigenschaft der Kreisteilungsklassen von Interesse. Verwendet man die Elemente von K_i als Exponenten eines primitiven Elements $z \in \mathrm{GF}(p^m)$ und setzt die z^i in die faktorisierte Form eines Polynoms $m_i(D)$ ein, so besitzt das Polynom nur Koeffizienten im Galoisfeld $\mathrm{GF}(p)$ und ist zudem irreduzibel bzgl. $\mathrm{GF}(p)$.

Beispiel: $p = 2, m = 1 \quad \rightarrow \quad q = 2^1 = 2, \ell = 3 \quad \rightarrow \quad n = 2^3 - 1 = 7$

Die Kreisteilungsklassen lauten:

$$\begin{aligned} K_0 &= \{0\} \\ K_1 &= \{1, 2, 4\} \\ K_3 &= \{3, 5, 6\}\ . \end{aligned}$$

Wir können mit den Klassen K_1 und K_3 zwei Polynome $m_1(D)$ und $m_3(D)$ bilden, deren Koeffizienten in GF(2) liegen und irreduzibel sind. Es gilt:

$$\begin{aligned} m_1(D) &= (D-z)(D-z^2)(D-z^4) = (D^2 + (\underbrace{z+z^2}_{z^4})D + z^3)(D-z^4) = D^3 + D + 1 \\ m_3(D) &= (D-z^3)(D-z^5)(D-z^6) = (D^2 + (\underbrace{z^5+z^3}_{z^2})D + z)(D-z^6) = D^3 + D^2 + 1\ . \end{aligned}$$

Vergleich: Konjugiert Komplexe Zahlen

Gegeben sind die beiden komplexen Zahlen a und a^*. Dann enthält das Polynom

$$(D-a)(D-a^*) = D^2 - aD - a^*D + aa^* = D^2 - 2Re\{a\}D + |a|^2 \tag{3.93}$$

ausschließlich reelle Koeffizienten!

Definition von primitiven BCH-Codes:

Es seien K_i die Kreisteilungsklassen für $n = p^\ell - 1$ und z das primitive Element des Erweiterungskörpers $\mathrm{GF}(p^m)$. Ferner stellt $\mathcal{M} = \bigcup_i K_i$ die Vereinigungsmenge beliebig vieler Kreisteilungsklassen dar. Dann wird ein BCH-Code der primitiven Blocklänge n durch das Generatorpolynom

$$g(D) = \prod_{\nu \in \mathcal{M}} (D - z^\nu) = \prod_i m_i(D) \tag{3.94}$$

beschrieben, wobei der Index i über die an $\mathcal{M}$ beteiligten Kreisteilungsklassen läuft. Durch die Wahl von $\nu \in \mathcal{M}$ wird $g_i \in \mathrm{GF}(p)$ gewährleistet. Die Entwurfsdistanz d wird erreicht, wenn $d-1$ aufeinanderfolgende Zahlen in $\mathcal{M}$ enthalten sind. Für die tatsächliche Mindestdistanz gilt $\delta \geq d \geq n-k+1$. Der Code hat die Dimension $|\Gamma| = 2^k$ mit $k = n - |\mathcal{M}|$.

Aufgrund der Forderung nach $d-1$ aufeinanderfolgenden Elementen in $\mathcal{M}$ enthält Γ alle Codeworte $\mathbf{x}$, für die die Nullstellen des Polynoms $x(D)$ genau $d-1$ aufeinanderfolgende Potenzen des primitiven Elementes z darstellen.

$$\Gamma = \left\{ x(D) \in \mathrm{GF}_p[D] \mid x(z^l) = x(z^{l+1}) = \ldots x(z^{l+d-2}) = 0 \right\} \tag{3.95}$$

Dabei beeinflusst die Lage der Nullstellen im Gegensatz zu den RS-Codes sehr wohl die Minimaldistanz. Entsprechend gilt im Frequenzbereich, dass $X(D)$ an mindestens $d-1$ zyklisch aufeinanderfolgenden Stellen Nullkoeffizienten ($X_\nu = 0$) besitzt.

$$\Gamma = \left\{ x(D) \mid x(D) \circ\!\!-\!\!\bullet\ X(D), X_{n-l} = \ldots = X_{n-d-l+2} = 0, X_{2i} = X_i^2 \right\} \tag{3.96}$$

Die Bedingung $X_{2i} = X_i^2$ garantiert, dass für die Koeffizienten x_i tatsächlich $x_i \in \mathrm{GF}(p)$ gilt. Allgemein gilt:

- Wahl der Vereinigung beliebig vieler Kreisteilungsklassen beeinflusst die Dimension des Codes und damit die Coderate (größere Flexibilität im Vergleich zu RS-Codes)
- BCH-Codes sind nicht zwingend MDS-Codes, d.h. tatsächliche Minimaldistanz kann wesentlich größer als Entwurfsdistanz sein

$$d_{\min} \geq d \quad \longrightarrow \quad k \leq n - d_{\min} + 1 \leq n - d + 1$$

- Minimaldistanz größer als Entwurfsdistanz kann bei Decodierung nach BMD nicht ausgenutzt werden (andere Decodierverfahren zu aufwendig)
- Bei Vorgabe von d nicht automatisch k, R_c und $d_{\min}$ bekannt (sie müssen noch berechnet werden)

- Es gibt auch **nicht primitive** BCH-Codes, für die $n < 2^l - 1$ gilt. Bekanntes Beispiel ist der $(23,12)$-Golay-Code

Beispiel: $p = 2$, $n = 7$, Entwurfsdistanz $d = 3 \quad \rightarrow \quad t = 1$ Fehler korrigierbar

Kreisteilungsklassen K_i siehe letztes Beispiel
Es müssen $d-1 = 2$ aufeinanderfolgende Zahlen in $\mathcal{M}$ enthalten sein

$$\begin{aligned} m_1(D) &= (D-z)(D-z^2)(D-z^4) = (D^2 + \underbrace{zD + z^2D}_{z^4D} + z^3)(D+z^4) \\ &= D^3 + z^4D + z^3D + z^4D + zD + 1 = D^3 + D + 1 \quad \rightarrow \quad m_{1,i} \in \mathrm{GF}(2) \end{aligned}$$

$$\begin{aligned} m_3(D) &= (D-z^3)(D-z^5)(D-z^6) = (D^2 + \underbrace{z^5D + z^3D}_{z^2D} + z)(D+z^6) \\ &= D^3 + z^2D^2 + zD + z^6D^2 + zD + 1 = D^3 + D^2 + 1 \quad \rightarrow \quad m_{3,i} \in \mathrm{GF}(2) \end{aligned}$$

$k = n - 3 = 4 \qquad \Rightarrow \qquad 2^4 = 16$ Codeworte mit einer Coderate $R_c = 4/7$
Wir erhalten einen zyklischen $(7,4,3)_2$-Hamming-Code.

3.7.6 Vergleich von RS- und BCH-Codes

BCH- und RS-Codes können ineinander überführt werden und stellen daher Spezialfälle der jeweils anderen Codefamilie dar. Beispielsweise entstehen aus RS-Codes durch die Beschränkung $x_i \in \mathrm{GF}(2)$ BCH-Codes, d.h. sie sind eine binäre Teilmenge aller RS-Codes und folglich gilt

$$d_{\min}^{BCH} \geq d_{\min}^{RS} = d\ .$$

Andererseits können BCH-Codes auch auf Körpern der Form $\mathrm{GF}(p^{mr})$ gebildet werden. Sie eignen sich dann besonders gut zur Korrektur von Bündelfehlern bis zu einer Länge von m und besitzen die primitive Länge $n = p^{mr} - 1$. Dann enthalten diese BCH-Codes für $r = 1$ als Spezialfall die RS-Codes. Allerdings finden in der Praxis fast ausschließlich binäre BCH-Codes Verwendung. Ferner kann gezeigt werden, dass alle 1-Fehler-korrigierenden BCH-Codes zyklische Hamming-Codes sind.

Wir wollen nun anhand eines Beispiels die Korrekturfähigkeit von RS-Codes und binären BCH-Codes erläutern.

(15,11,5)$_{16}$-RS-Code

- Code kann aufgrund von $d_{\min} = d = 5$ immer 2 fehlerhafte Codesymbole korrigieren
- Binäre Betrachtung: Jedes Codesymbol besteht aus 4 Bit

$\rightarrow$ Es können Bündelfehler bis zur maximalen Länge $t = 8$ korrigiert werden

- Beispiel:

$$\begin{aligned} \mathbf{e} &= \cdots 0000\,1111\,1111\,0000\ \cdots \quad && \text{korrigierbar} \\ \mathbf{e} &= \cdots 0001\,1111\,1000\,0000\ \cdots && \text{nicht korrigierbar} \\ \mathbf{e} &= \cdots 0001\,0010\,1000\,0000\ \cdots && \text{nicht korrigierbar} \end{aligned}$$

  - Erste Zeile zeigt einen Bündelfehler der Länge 8
  - Insgesamt sind nur **zwei** 4-Bit-Codesymbole betroffen $\longrightarrow$ Fehler korrigierbar
  - Zweite Zeile illustriert Bündelfehler mit 6 Bitfehlern über insgesamt **3** Symbole

→ Wegen $d_{\min} = 5$ kann Fehler nicht korrigiert werden (trotz weniger Einzelfehlern)
- Letzte Zeile: 3 Symbole durch 3 Bitfehler verfälscht $\longrightarrow$ nicht-korrigierbarer Fehler

→ Beispiel verdeutlicht spezielle Eignung von RS-Codes zur Korrektur von Bündelfehlern (Bei Bündelfehlern mehr einzelne Binärstellen korrigierbar als bei voneinander unabhängigen Einzelfehlern)

- Keine Eignung zur Korrektur von statistisch unabhängigen Einzelfehlern

(63,45,7)$_2$-BCH-Code

- BCH-Code besitzt ungefähr vergleichbare Wortlänge n und Coderate R_c
- Eignung zur Korrektur von Einzelfehlern, da wegen $d_{\min} = 7$ immer 3 Einzelfehler korrigierbar
  (Dies schafft der obige RS-Code nicht)

Fazit:

- RS-Codes besser zur Korrektur von Bündelfehlern geeignet
- Binäre BCH-Codes bei Einzelfehlern zu bevorzugen

→ Wahl eines konkreten Codierungsverfahrens abhängig von Fehlereigenschaften des Übertragungskanals

Wortlänge n, Anzahl Infobit k und Anzahl korrigierbarer Fehler t für die wichtigsten BCH-Codes
Tabelle 3.3 zeigt für die wichtigsten BCH-Codes die zugehörigen Parameter n, k und t. Dabei ist zu beachten, dass die BCH-Codes unter Umständen mehr Fehler korrigieren könnten, als durch die Entwurfsdistanz vorgegeben. Allerdings kann dies bei der BMD-Decodierung nicht ausgenutzt werden.

Leistungsfähigkeit von BCH-Codes

Bilder 3.9 zeigt Bitfehlerkurven für verschiedene BCH-Codes der Länge $n = 255$ über dem Signal-Rausch-Abstand E_b/N_0 aufgetragen (AWGN-Kanal). Es ist ersichtlich, dass zunächst mit abnehmender Coderate R_c die Bitfehlerrate stetig abnimmt (oder der für eine bestimmte Bitfehlerrate erforderliche Signal-Rausch-Abstand wird kleiner), was auf eine wachsende Mindestdistanz und damit auch eine steigende Korrekturfähigkeit t zurückzuführen ist. Ab $t \geq 25$ geht die Leistungsfähigkeit aber wieder zurück, was auf das asymptotisch schlechte Verhalten von BCH-Codes zurückzuführen ist. Hier ist zu beachten, dass über E_b/N_0, und nicht über E_s/N_0 aufgetragen wurde. Zwar lassen sich bei wachsendem t immer mehr Fehler korrigieren, für $t \geq 25$ sinkt die Coderate R_c allerdings schneller als die Entwurfsdistanz d und somit auch t steigt (**asymptotisch schlecht**). Da R_c in das Maß E_b/N_0 eingeht, erklärt sich das beobachtete Verhalten. Die übrigen Bilder 3.10 bis 3.12 zeigen die Bitfehlerkurven für jeweils konstante Coderaten und unterschiedliche Blocklängen n. Hier wird die *erwartete* Verbesserung bei wachsender Blocklänge deutlich.

n	k	t	n	k	t	n	k	t	n	k	t
7	4	1	127	85	6	255	123	19	511	403	12
				78	7		115	21		394	13
15	11	1		71	9		107	22		385	14
	7	2		64	10		99	23		.	.
	5	3		57	11		91	25		259	30
				50	13		87	26		.	.
31	26	1		43	14		79	27		130	55
	21	2		36	15		71	29		.	.
	16	3		29	21		63	30		.	.
	11	5		22	23		55	31	1023	1013	1
	6	7		15	27		47	42		1003	2
				8	31		45	43		993	3
63	57	1					37	45		983	4
	51	2	255	247	1		29	47		973	5
	45	3		239	2		21	55		963	6
	39	4		231	3		13	59		953	7
	36	5		223	4		9	63		943	8
	30	6		215	5					933	9
	24	7		207	6	511	502	1		923	10
	18	10		199	7		493	2		913	11
	16	11		191	8		484	3		903	12
	10	13		187	9		475	4		.	.
	7	15		179	10		466	5		768	26
				171	11		457	6		.	.
127	120	1		163	12		448	7		513	57
	113	2		155	13		439	8		.	.
	106	3		147	14		430	9		258	106
	99	4		139	15		421	10		.	.
	92	5		131	18		412	11		.	.

Tabelle 3.3: Beispiele für Parameter von BCH-Codes

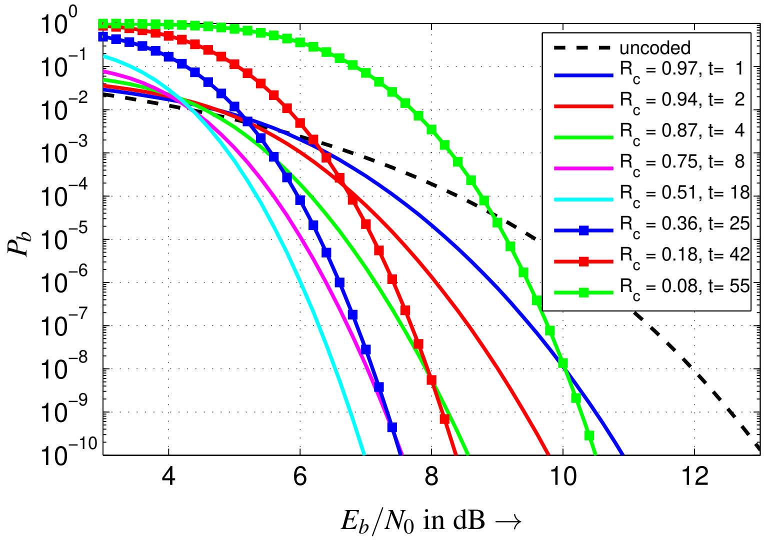


Bild 3.9: Bitfehlerkurven für BCH-Codes der Länge $n = 255$

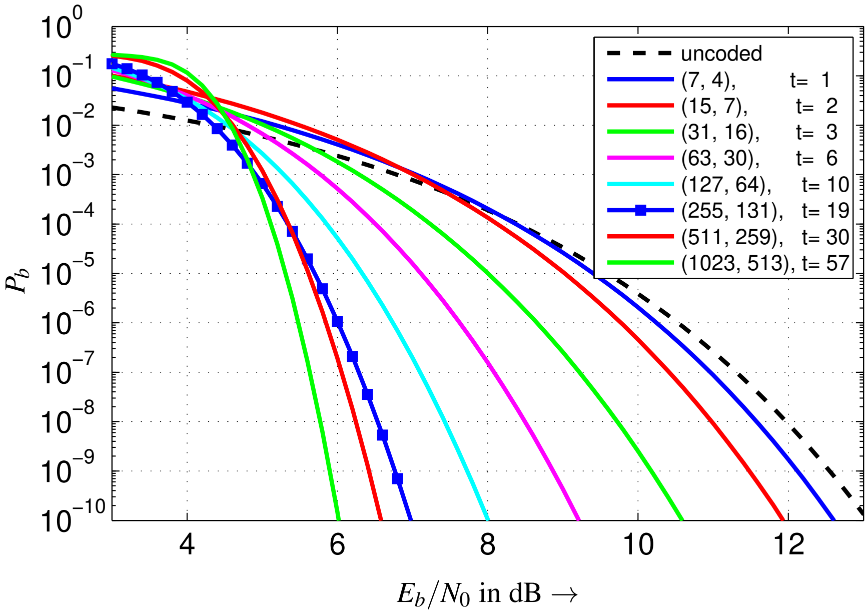


Bild 3.10: Bitfehlerkurven für BCH-Codes der Coderate $R_c = 1/2$

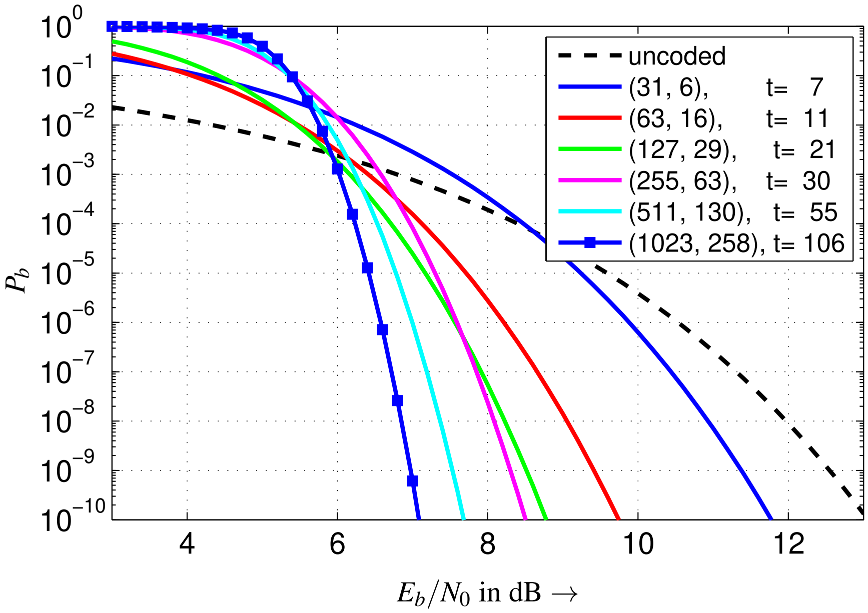


Bild 3.11: Bitfehlerkurven für BCH-Codes der Coderate $R_c = 1/4$

3.7.7 Decodierung von BCH- und RS-Codes

Auch für die beiden sehr wichtigen Codefamilien, die BCH- und die RS-Codes, ist eine optimale Maximum-Likelihood-Decodierung viel zu aufwendig, um in der Praxis umgesetzt zu werden. Allerdings können mit der einfacheren begrenzten Minimaldistanz-Decodierung (BMD) sehr effiziente Algorithmen entwickelt werden, die auf anspruchsvollen algebraischen Verfahren basieren und stets eine Hard-Decision am Kanalausgang voraussetzen. In Sonderfällen kann auch eine sehr einfache Form der Soft-Decision, die Ausfallinformation (vgl. BSEC-Kanal) ausgenutzt werden. Den Ausgangspunkt aller Verfahren bildet somit die folgende Darstellung des Empfangsvektors

$$\mathbf{y} = \mathbf{x} + \mathbf{e} \quad \circ\!\!-\!\!\bullet \quad \mathbf{Y} = \mathbf{X} + \mathbf{E} \tag{3.97}$$

Voraussetzung:

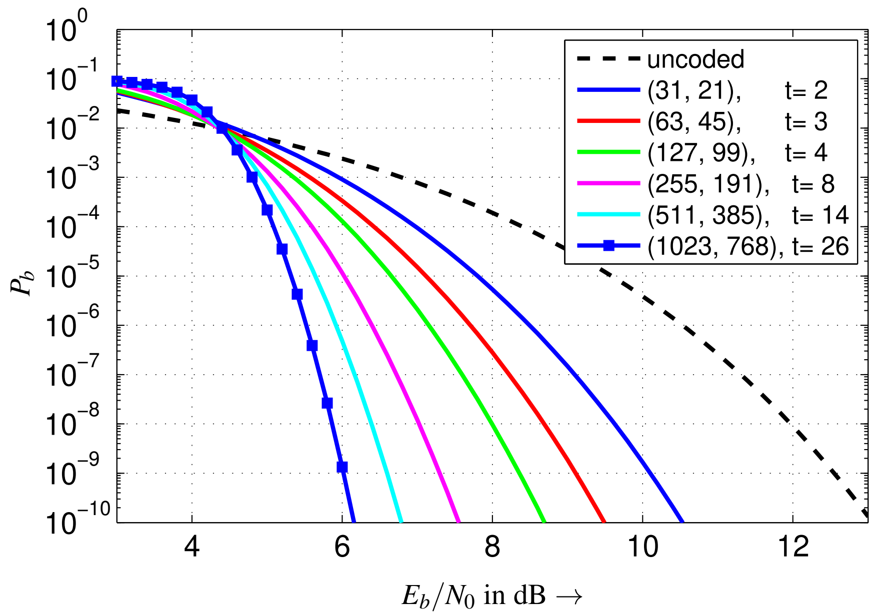


Bild 3.12: Bitfehlerkurven für BCH-Codes der Coderate $R_c = 3/4$

- Symbole von $\mathbf{x}$ und $\mathbf{e}$ bei RS-Codes stets q-stufig
- Symbole von $\mathbf{x}$ und $\mathbf{e}$ bei BCH-Codes p-stufig
- Symbole von $\mathbf{X}$ und $\mathbf{E}$ immer q-stufig
- Entwurfsdistanz: $d = 2t+1$
- Es treten nie mehr als $\tau \leq t$ Fehler auf (andernfalls versagt Decodierung)

Wir nehmen an, dass die Nullsymbole im Frequenzbereich an den Stellen 0 bis $d-2$ liegen

$$\mathbf{X} = [\underbrace{0 \cdots 0}_{2t=d-1} X_{2t} \cdots X_{n-1}]. \tag{3.98}$$

Für das Empfangswort $\mathbf{Y}$ im Spektralbereich bedeutet dies, dass die ersten $2t$ Stellen direkt die Symbole des Fehlerwortes $\mathbf{E}$ enthalten. Dieser Teilvektor wird als Syndrom $\mathbf{S}$ bezeichnet, d.h. es gilt

$$\mathbf{Y} = [\underbrace{E_0 \cdots E_{2t-1}}_{\mathbf{S}} X_{2t} \oplus E_{2t} \cdots X_{n-1} \oplus E_{n-1}] \,. \tag{3.99}$$

Das Syndrom ist somit direkt durch die Paritätsfrequenzen des Empfangswortes bestimmt.

$$S(D) = \sum_{i=0}^{2t-1} S_i \cdot D^i \quad \text{mit} \quad S_i = E_i = Y_i = \sum_{\mu=0}^{n-1} y_\mu \cdot z^{-i\mu} \tag{3.100}$$

Die Berechnung der S_i folgt direkt aus der Definition der Spektraltransformation in Gl. (3.74). Selbstverständlich gilt auch hier weiterhin für Codeworte $\mathbf{S} = \mathbf{0}$.

Grober Ablauf der Decodierung

- Decodierung von RS- und BCH-Codes läuft prinzipiell immer nach gleichem Schema ab
- Bei BCH-Codes enorme Vereinfachungen aufgrund ihrer binären Struktur (nicht weiter behandelt)

1. Mit Hilfe der DFT aus Empfangswort $\mathbf{y}$ das zugehörige Frequenzwort $\mathbf{Y}$ bestimmen (Syndrom $\mathbf{S}$ geht nicht aus DFT von $\mathbf{s}$ hervor, sondern aus Teil-DFT des kompletten Empfangswortes $\rightarrow$ Berechnung von $\mathbf{S}$ erfordert größten Anteil des Decodieraufwandes)

2. An Paritätsfrequenzen direkt das Syndrom ablesen

3a. Bei reiner Fehlererkennung endet Decodierung mit der Prüfung auf $\mathbf{S} = 0$

3b. Bei Fehlerkorrektur aus Syndrom Fehlerstellenpolynom $C(D)$ und Fehlerpolynom $E(D)$ berechnen

4. Rücktransformation von $E(D)$ in den Zeitbereich ($e(D)$)

5. Berechnung des geschätzten Codewortes mit $\hat{x}(D) = y(D) - e(D)$

Bestimmung des *Fehlerstellenpolynoms* $C(D)$

- Zu bekanntem Syndrom wird Fehlervektor $\mathbf{e}$ gesucht, der minimales Hamming-Gewicht $w_{\mathrm{H}}(\mathbf{e})$ besitzt (reine Syndromdecodierung ist viel zu aufwendig, Bsp. $2^{63} = 9.2 \cdot 10^{18}$ Codeworte prüfen)

$\rightarrow$ Suche nach geeignetem Polynom minimalen Grades

- Definition der Fehlerstellenmenge I, die alle fehlerhaften Stellen von $\mathbf{y}$ enthält (*Error Location Set*)

$$I = \{i \mid e_i \neq 0 \quad \wedge \quad 0 \leq i \leq n-1\} \tag{3.101}$$

- Mit Hilfe von Gl. (3.101) kann zugehöriges Fehlerstellenpolynom (*Error Location Polynomial*)

$$C(D) = \prod_{i \in I} \left(D - z^i\right) = \tilde{C}_0 + \tilde{C}_1 \cdot D + \cdots + \tilde{C}_\tau \cdot D^\tau \tag{3.102}$$

  definiert werden, welches bei der Normierung auf $C_0 = 1$ die Form

$$C(D) = \prod_{i \in I} \left(1 - Dz^{-i}\right) = 1 + C_1 \cdot D + \cdots + C_\tau \cdot D^\tau \tag{3.103}$$

  annimmt

- Im fehlerfreien Fall gilt $I = \emptyset$ und $C(D) = 1$

- Nullstellen des **Fehlerstellenpolynoms** lauten z^i mit $i \in I$

$\rightarrow$ Berechnung der Nullstellen von $C(D)$ ermöglicht Bestimmung der fehlerhaften Stellen i des empfangenen Wortes $\mathbf{y}$

$$I = \left\{i \mid C\left(z^i\right) = 0\right\} \tag{3.104}$$

Chien-Suche: Durchprobieren aller möglichen negativen Potenzen von z und Überprüfen auf $C(z^i) = 0$.

Beispiel für $\mathrm{GF}(8)$**:**

$$\begin{aligned}
e(D) &= 1 + D^3; \qquad \mathbf{e} = (1\,0\,0\,1\,0\cdots) \\
C(D) &= (1-D)(1-Dz^{-3}) \\
C(z^0) &= \underbrace{(1-z^0)}_{=0}(1-z^{-3}) = 0 \\
C(z^1) &= (1-z)(1-z^{-2}) = (1-z)(1-z^{-2}z^7) = 1+z+z^5+z^6 = 1 \neq 0 \\
C(z^2) &= (1-z^2)(1-z^{-1}) = (1-z^2)(1-z^{-1}z^7) = 1+z^2+z^6+z = z \neq 0 \\
C(z^3) &= (1-z^3)(1-1) = 0
\end{aligned}$$

- Es gilt $c(D) \circ\!\!-\!\!\bullet\, C(D)$
- Zur Erinnerung:

$$\begin{aligned} z^{-i} \text{ ist Nullstelle von } a(D) \quad &\Longleftrightarrow \quad A_i = 0 \\ z^{i} \text{ ist Nullstelle von } A(D) \quad &\Longleftrightarrow \quad a_i = 0 \end{aligned}$$

→ Hieraus folgt direkt die Beziehung

$$e_i \cdot c_i = 0, \tag{3.105}$$

denn es gilt:

  - $i \in I \;\Rightarrow$ fehlerhafte Stelle $\Rightarrow z^i$ ist Nullstelle von $C(D) \;\Rightarrow\; c_i = 0$
  - $i \notin I \;\Rightarrow$ fehlerfreie Stelle $\Rightarrow e_i = 0$

- Gl. (3.105) lautet im Frequenzbereich

$$e_i \cdot c_i = 0 \qquad \circ\!\!-\!\!\bullet \qquad R_{D^n-1}\left[C(D) \cdot E(D)\right] = 0 \tag{3.106}$$

  Interpretation:

  - Alle Koeffizienten von $R_{D^n-1}\left[C(D) \cdot E(D)\right]$ müssen gleich Null sein
  - Aufgrund der modulo $D^n - 1$-Operation treten nur Potenzen zwischen 0 und $n-1$ auf
  - Größere Potenzen von $C(D) \cdot E(D)$ werden durch $(i+j) \bmod n$ wieder auf Bereich 0 bis $n-1$ abgebildet

  → Folgende Terme tragen zum Koeffizienten der i-ten Potenz bei:

$$\sum_{\mu=0}^{\tau} C_\mu \cdot E_{(i-\mu) \; mod \; n} = 0$$

- Auswertung nur an den Stellen $\tau \leq i \leq 2\tau - 1$
  → es kommen nur Werte des Fehlervektors von E_0 bis $E_{2\tau-1}$ zum Einsatz
- Diese sind bekannt (Paritätsfrequenzen) und repräsentieren das Syndrom $\mathbf{S}$
- Wegen $E_i = S_i$ für $\tau \leq i \leq 2\tau - 1$ erhalten wir

$$\sum_{\mu=0}^{\tau} C_\mu \cdot S_{i-\mu} = 0 \qquad \text{für} \qquad i = \tau, \ldots, 2\tau - 1 \tag{3.107}$$

- **Schlüsselgleichung** oder **Newton-Gleichung** ($C_0 = 1$)

$$\boxed{S_i + \sum_{\mu=1}^{\tau} C_\mu \cdot S_{i-\mu} = 0 \qquad \text{für} \qquad i = \tau, \ldots, 2\tau - 1} \tag{3.108}$$

- Gl. (3.108) für jedes i (insgesamt τ) lösen → Gleichungssystem aus τ Gleichungen mit τ Unbekannten

→ Zumindest prinzipiell eindeutig lösbar

- Matrixdarstellung der Schlüsselgleichung

$$\begin{bmatrix} -S_\tau \\ -S_{\tau+1} \\ \vdots \\ -S_{2\tau-1} \end{bmatrix} = \underbrace{\begin{bmatrix} S_0 & S_1 & \cdots & S_{\tau-1} \\ S_1 & S_2 & \cdots & S_\tau \\ \vdots & \vdots & & \vdots \\ S_{\tau-1} & S_\tau & \cdots & S_{2\tau-2} \end{bmatrix}}_{\mathbf{S}_{\tau,\tau}} \cdot \begin{bmatrix} C_\tau \\ C_{\tau-1} \\ \vdots \\ C_1 \end{bmatrix} \tag{3.109}$$

Lösen der Schlüsselgleichung:

- Problem: Dimension der Matrix $\mathbf{S}_{\tau,\tau}$ ist nicht bekannt, da Empfänger nicht von vorn herein Anzahl der Fehler im empfangenen Wort kennt

→ Dimension der Matrix **S** unbekannt

- Lösung:
  - Annahme: es können nicht mehr als t Fehler auftreten
    (wegen minimaler Distanz $d_{\min} = 2t + 1$ können sowieso nicht mehr Fehler korrigiert werden)
  - Matrix $\mathbf{S}_{t,t}$ aufstellen und auf Singularität prüfen
    - $\mathbf{S}_{t,t}$ singulär $\longrightarrow$ weniger als t Fehler aufgetreten

    → Dimension der Matrix durch Streichen der untersten Zeile und der rechten Spalte verringern und erneute Prüfung auf Singularität
    - . Wiederholung bis **S** nicht mehr singulär

    → Dimension von **S** entspricht der Anzahl an Fehlern

Ist tatsächliche Fehlerzahl τ durch oben beschriebene Prozedur bestimmt worden, muss die Schlüsselgleichung gelöst werden. Aufgrund des hohen Rechenaufwandes sind in der Praxis aufwandsgünstige Algorithmen erforderlich. Ein geeigneter Algorithmus ist der sogenannte Berlekamp-Massey-Algorithmus.

Berlekamp-Massey-Algorithmus

- Berechnung mit Hilfe eines rückgekoppelten Schieberegisters der Länge t (s. Bild 3.13)

→ Iterativer Prozess, da die i-te Stelle S_i des Syndroms aus Linearkombination τ vorangegangener Stellen $S_{i-\mu}$ berechenbar

→ Aufgabe: Bestimmen der Koeffizienten eines rückgekoppelten Schieberegisters **minimaler Länge** (τ), welches das Syndrom **S** nachbildet

- Lösung:
  - $d - 1 = 2t$ Symbole des Syndroms sind bekannt (Paritätsfrequenzen)
  - Schieberegister der Länge t wird mit den ersten t Symbolen des Syndroms S_0 bis S_{t-1} initialisiert
  - In t zyklischen Verschiebungen werden die Koeffizienten C_i derart bestimmt, dass sie die übrigen t Symbole S_t bis S_{2t-1} des Syndroms nachbilden

→ Koeffizienten der Linearkombination bzw. des Schieberegisters sind die gesuchten Koeffizienten C_i des Fehlerstellenpolynoms

→ Mit Lösung der Schlüsselgleichung ist automatisch das Fehlerstellenpolynom $C(D)$ bekannt

→ Mit Suche nach Chien kann dann aus $C(D)$ die Fehlerstellenmenge I bestimmt werden

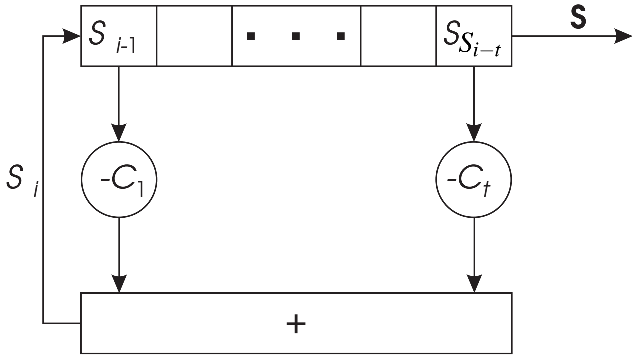


Bild 3.13: Schieberegisterrealisierung zur Lösung der Schlüsselgleichung

$\rightarrow$ Fehlerhafte Stellen im Empfangsvektor $\mathbf{y}$ bekannt

Bestimmung des Fehlerpolynoms $E(D)$ durch Prinzip der rekursiven Ergänzung

Da RS-Codes nicht-binäre Codes sind, reicht ein einfaches Invertieren der fehlerhaften Symbole nicht aus.
$\longrightarrow$ Es muss auch der Wert des Fehlers bekannt sein.

Es existieren Verfahren sowohl im Zeit- als auch im Frequenzbereich (hier Darstellung im Frequenzbereich wegen einfacherer Darstellung):

- Aus $\sum_{\mu=0}^{\tau} C_\mu \cdot E_{(i-\mu)\ mod\ n} = 0$ folgt wegen $C_0 = 1$

$$E_i = -\sum_{\mu=1}^{\tau} C_\mu \cdot E_{(i-\mu)\ mod\ n}.$$

- Demnach gilt $E_i = f(E_{i-1}, \ldots, E_{i-\tau})$
- Wegen $E_0 = S_0, \cdots, E_{2t-1} = S_{2t-1}$ können restliche $n-2t$ Fehlersymbole E_i sukzessive ermittelt werden
- Effiziente Realisierung mit rückgekoppeltem Schieberegister nach Bild 3.14
  Abschließend Rücktransformation von $E(D)$ in den Zeitbereich
  Das geschätzte Codewort lautet dann

$$\hat{x}(D) = y(D) - e(D) \tag{3.110}$$

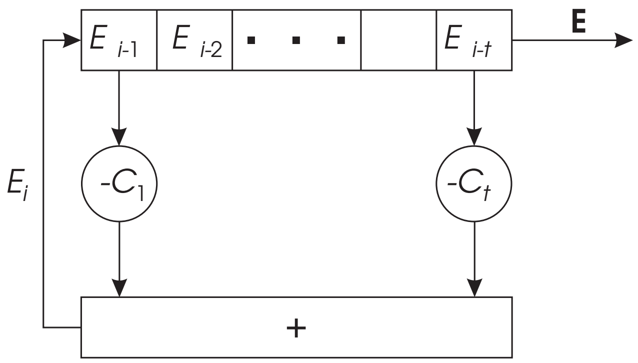


Bild 3.14: Schieberegisterrealisierung für rekursive Ergänzung

Da die Rücktransformation auch einen erheblichen Rechenaufwand bedingt, werden in der Praxis nur diejenigen Stellen des Fehlervektors **e** bestimmt, die ungleich Null sind. Bild 3.15 zeigt noch einmal den Ablauf der Decodierung im Überblick.

Die Decodierung der binären BCH-Codes erfolgt prinzipiell nach der gleichen Vorgehensweise. Allerdings ergeben sich hier teilweise drastische Vereinfachungen, auf die im Rahmen dieser Vorlesung allerdings nicht weiter eingegangen werden kann. Hierfür wären weitere, vertiefende Kenntnisse aus der Polynomarithmetik erforderlich. Aufgrund ihrer sehr effizienten Decodiermöglichkeit besitzen BCH- als auch Reed-Solomon-Codes eine sehr große Bedeutung in praktischen Systemen.

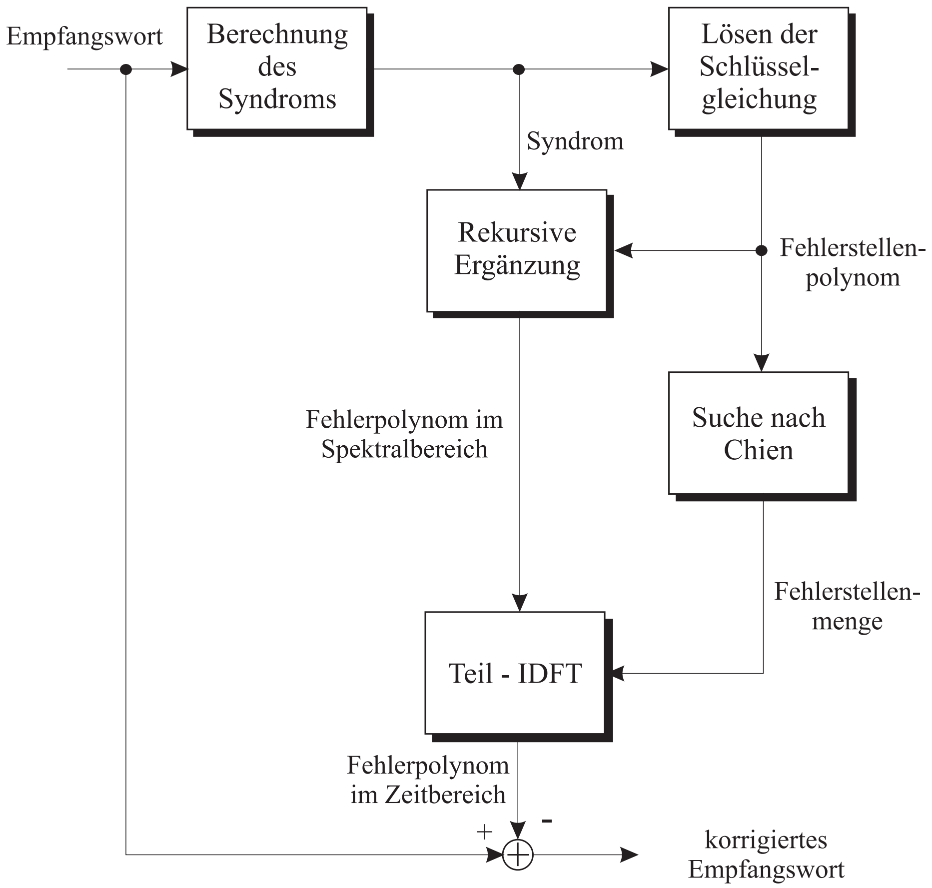


Bild 3.15: Ablauf der Decodierung von RS- und BCH-Codes

Kapitel 4

Faltungscodes

4.1 Grundlagen

4.1.1 Aufbau des Codierers

Faltungscodierer lassen sich wie auch zyklische Blockcodierer mit Hilfe von Schieberegistern effizient realisieren. Die allgemeine Struktur von Schieberegistern zur Faltungscodierung zeigt Bild 4.1. Die Funktionsweise kann wie folgt zusammengefaßt werden:

- Schieberegister mit i.a. $L_c \cdot k$ Elementen
- In jedem Takt werden k Bit weitergeleitet

$\rightarrow$ Jedes Bit beeinflußt L_c Mal das Ausgangswort $\rightarrow$ L_c heißt Einflusslänge oder auch *Constraint Length*

- Berechnung der codierten Symbole durch Verknüpfung der Registerinhalte über n modulo-2-Addierer

$\rightarrow$ Ausgangswort besteht aus n Bit (Codewort)

$\rightarrow$ Coderate $R_c = k/n$

- Beschreibung der Verbindungen über sogenannte Generatorpolynome oder Generatoren
- Geeignete Struktur nicht trivial und nicht analytisch zu bestimmen

$\rightarrow$ Aufwendige Rechnersuche erforderlich

- $\boxed{\text{Im folgenden Beschränkung auf Codes der Rate } R_c = 1/n}$

Ein wichtiger Parameter der Faltungscodes ist die oben erwähnte **Einflusslänge** oder auch ***Constraint Length*** L_c. Sie beschreibt die Anzahl der Takte, die ein Eingangsbit direkt an der Bildung eines Ausgangswortes beteiligt ist und entspricht für $k = 1$ der Speicherlänge m des Registers plus Eins ($L_c = m + 1$). Je größer L_c ist, desto mehr Kombinationsmöglichkeiten bestehen zur Bildung der codierten Symbole und desto leistungsfähiger ist der Faltungscode. Die Einflusslänge L_c übernimmt also die Rolle der Blocklänge n der linearen Blockcodes.

Beispiel: $R_c = 1/2$, $L_c = 3$ $\quad\rightarrow\quad$ Speicherlänge $m = 2$

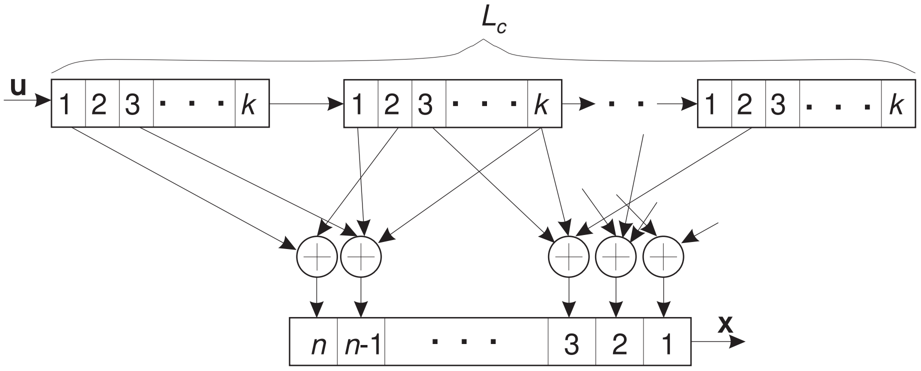


Bild 4.1: Allgemeine Schieberegisterstruktur von Faltungscodes

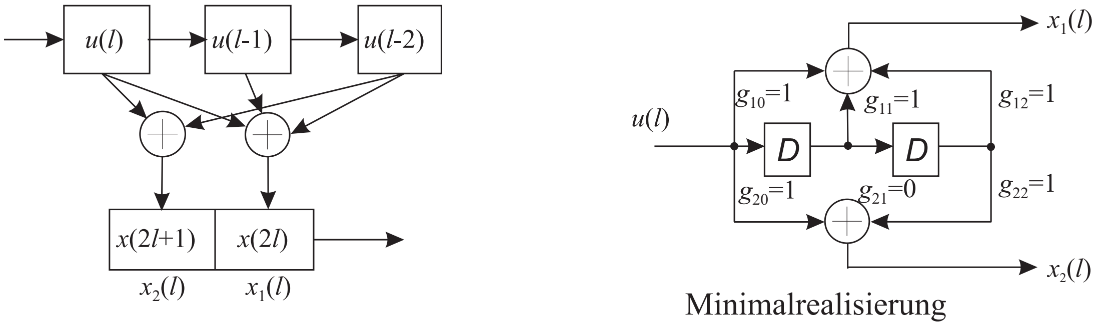


Bild 4.2: Schieberegisterstruktur für Faltungscode mit Generatoren $g_1 = 7_8$ und $g_2 = 5_8$

4.1.2 Äquivalenz von Blockcodes und Faltungscodes

- Faltungscode bildet k Informationsbit auf Codewort $\mathbf{x}$ mit n Bit ab
- Codeworte $\mathbf{x}$ sind voneinander abhängig (Faltungscode hat Gedächtnis)
- Blockcodes erzeugen voneinander unabhängige Codeworte

$\rightarrow$ Blockcodes sind spezielle Faltungscodes ohne Gedächtnis

- In der Praxis Betrachtung von faltungscodierten Sequenzen endlicher Länge
- Interpretation der codierten Sequenz als Codewort eines Blockcodes

$\rightarrow$ Faltungscodes sind Spezialfall von Blockcodes

Blockcodes und Faltungscodes sind ineinander überführbar, aber Beschreibung der Faltungscodes viel einfacher

Eigenschaften von Faltungscodes

- Nur wenige einfache Faltungscodes in der Praxis relevant
- Faltungscodes bieten sehr einfache Möglichkeit der *Soft-Decision*-Decodierung (Bei Blockcodes bisher nur *Hard-Decision*-Decodierung)
- Keine analytische Konstruktion guter Faltungscodes möglich $\longrightarrow$ Aufwendige Rechnersuche (dafür aber einfache mathematische Beschreibung)
- Wie bei Blockcodes Unterscheidung zwischen systematischen und nicht-systematischen Codes (in der Praxis fast ausschließlich nicht-systematische Codes)

4.1.3 Algebraische Beschreibung

Faltungscodes lassen sich über ihre Generatoren $\mathbf{g}_j$ beschreiben, welche in der Regel in oktaler Form dargestellt werden. Der Faltungscode mit $L_c = 3$ aus dem obigen Beispiel besitzt wegen $R_c = 1/2$ genau 2 Generatoren.

$$\mathbf{g}_1 \quad = \quad [g_{1,0}\, g_{1,1}\, g_{1,2}] = [1\,1\,1] \hat{=} 7_8 \tag{4.1}$$

$$\mathbf{g}_2 \quad = \quad [g_{2,0}\, g_{2,1}\, g_{2,2}] = [1\,0\,1] \hat{=} 5_8 \tag{4.2}$$

Soll die oktale Schreibweise auch für $L_c \neq 3\kappa$ beibehalten werden, sind den Vektoren $\mathbf{g}_j$ entsprechend viele Nullen links voran zu stellen, so dass sie insgesamt als Länge ein Vielfaches von drei besitzen. Die Codierung kann dann durch diskrete Faltung der Eingangssequenz $\mathbf{u}$ mit den Generatoren erfolgen

$$\mathbf{x}_1 = \mathbf{u} * \mathbf{g}_1 \qquad \text{und} \qquad \mathbf{x}_2 = \mathbf{u} * \mathbf{g}_2\ . \tag{4.3}$$

und es gilt allgemein

$$x_\nu(\ell) = \sum_{i=0}^{m} g_{\nu,i} \cdot u_{l-i} \bmod 2\ . \tag{4.4}$$

Mit Hilfe der z-Transformation können die Signalfolgen wie auch die Generatoren im Spektralbereich dargestellt werden. Allgemein gilt der Zusammenhang $\mathcal{Z}(\mathbf{x}) = \sum_{i=0}^{\infty} x_i \cdot z^{-i}$. In der Codierungstheorie hat sich dabei die Vereinbarung durchgesetzt, den Verzögerungsoperator z^{-1} durch D zu ersetzen. Wir erhalten somit $X(D) = \sum_{i=0}^{\infty} x_i \cdot D^i$ und die Generatoren lassen sich folgendermaßen beschreiben:

$$G_1(D) \quad = \quad g_{10} + g_{11}D + g_{12}D^2 = 1 + D + D^2 \tag{4.5}$$

$$G_2(D) \quad = \quad g_{20} + g_{21}D + g_{22}D^2 = 1 + D^2\ . \tag{4.6}$$

Allgemein gilt für das ν-te Polynom und die zugehörige Ausgangsfolge $\mathbf{X}_\nu(D)$:

$$G_\nu(D) = \sum_{i=0}^{m} g_{\nu\, i} \cdot D^i \qquad \text{und} \qquad X_\nu(D) = U(D) \cdot G_\nu(D) \tag{4.7}$$

Die gesamte codierte Sequenz kann in der Polynomdarstellung durch

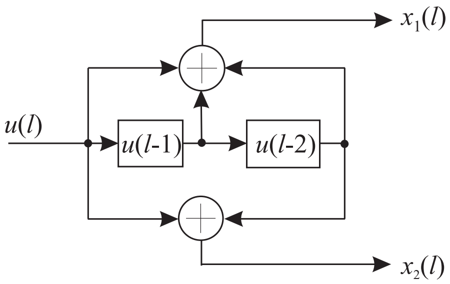
$$\mathbf{X}(D) = [X_1(D)\; X_2(D)\; \ldots\; X_n(D)] = U(D) \cdot \mathbf{G}(D) \tag{4.8}$$

mit der Generatormatrix $\mathbf{G}(D) = [G_1(D)\; G_2(D)\; \ldots\; G_n(D)]$ dargestellt werden. Für den Coderaum gilt entsprechend

$$\Gamma = \{U(D) \cdot \mathbf{G}(D) \mid U(D), U_i \in GF_2\}\ . \tag{4.9}$$

Beispiel: $\mathbf{u} = (1\,0\,0\,1\,1\,\ldots)$

u_l	Zustand	Folgezustand	Ausgang
1	0 0	1 0	1 1
0	1 0	0 1	1 0
0	0 1	0 0	1 1
1	0 0	1 0	1 1
1	1 0	1 1	0 1
0	1 1	0 1	0 1
0	0 1	0 0	1 1



$$
\begin{aligned}
U(D) &= 1+D^3+D^4 \\
\mathbf{G}(D) &= \begin{bmatrix}1+D+D^2 & 1+D^2\end{bmatrix} \\
\mathbf{X}(D) &= \begin{bmatrix}[1+D^3+D^4]\cdot[1+D+D^2] & [1+D^3+D^4]\cdot[1+D^2]\end{bmatrix} \\
&= \begin{bmatrix}1+D^3+D^4+D+D^4+D^5+D^2+D^5+D^6 & 1+D^3+D^4+D^2+D^5+D^6\end{bmatrix} \\
&= \begin{bmatrix}1+D+D^2+D^3+D^6 & 1+D^2+D^3+D^4+D^5+D^6\end{bmatrix} \\
&\triangleq [11\,10\,11\,11\,01\,01\,11]
\end{aligned}
$$

4.1.4 Graphische Beschreibung durch Zustandsdiagramm

- Faltungscodierer kann als Mealy-Automat dargestellt werden
  - $\rightarrow$ Ausgangssignal ist abhängig vom aktuellen Zustand <u>und</u> vom Eingangssignal: $\mathbf{x}(\ell) = f_x(u(\ell), S(\ell))$
  - $\rightarrow$ Folgezustand resultiert aus altem Zustand <u>und</u> Eingangssignal: $S(\ell+1) = f_S(u(\ell), S(\ell))$
- Zustandsdiagramm illustriert mögliche Zustandsübergänge und dazugehörige Ausgangswerte
- Zustandsdiagramm enthält **keine** Information über den zeitlichen Ablauf der Codierung

Beispiel:
Faltungscode mit $g_1 = 7_8$, $g_2 = 5_8$ $\quad\longrightarrow\quad$ $R_c = 1/2$, $L_c = 3 \rightarrow m = L_c - 1 = 2 \Rightarrow 2^m = 4$ Zustände

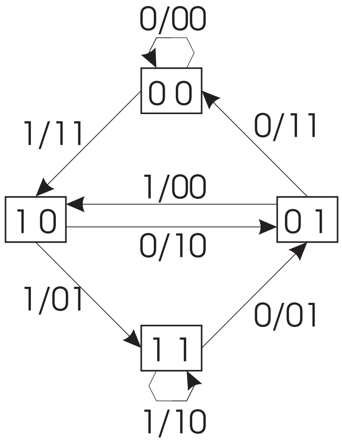


Bild 4.3: Zustandsdiagramm für Faltungscode mit Generatoren $g_1 = 7_8$ und $g_2 = 5_8$

4.1.5 Graphische Beschreibung durch Trellisdiagramm

Das Trellisdiagramm geht aus dem Zustandsdiagramm durch eine zusätzliche zeitliche Komponente hervor. Jeder Knoten stellt einen Zustand und ein Pfad einen Übergang zwischen zwei Zuständen dar. In der Regel beginnt das Trellisdiagramm im Nullzustand. Dann ist es nach $\ell = L_c$ Schritten voll entwickelt, d.h. alle Zustände bzw. Knoten wurden mindestens einmal erreicht. Ab dann wird das Trellis periodische fortgesetzt.

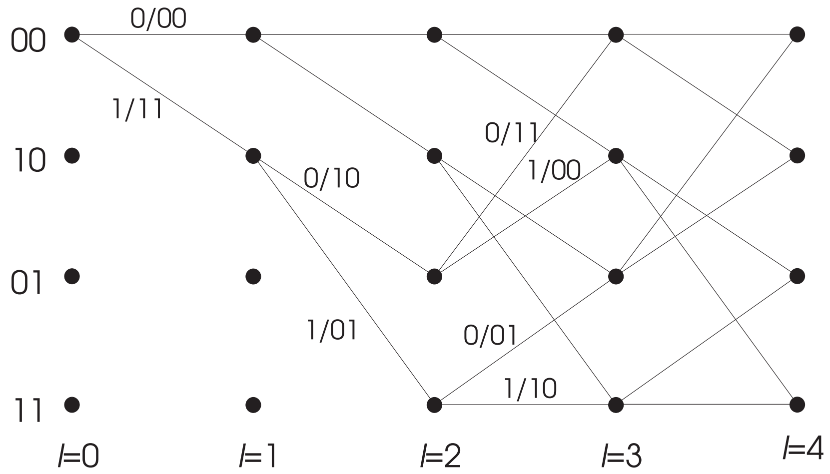


Bild 4.4: Trellisdiagramm für Faltungscode mit Generatoren $g_1 = 7_8$ und $g_2 = 5_8$

4.2 Charakterisierung von Faltungscodes

4.2.1 Systematische, nicht-systematische und rekursive Faltungscodes

Wie schon die in Kapitel 3 behandelten Blockcodes sind auch Faltungscodes linear (s. auch Distanzspektrum von Faltungscodes). Dabei wollen wir im Folgenden systematische und nicht-systematische Faltungscodes unterscheiden, wobei den rekursiven Codes eine besondere Bedeutung zukommt.

Nicht-systematische Faltungscodes: (siehe auch Bild 4.2)
NSC-Codes: *Nonrecursive Nonsystematic Convolutional Codes*

- Keine Trennung zwischen Informations- und Prüfbit im Codewort möglich
- Wesentlich höhere Leistungsfähigkeit als systematische Faltungscodes
- Werden in der Praxis fast ausschließlich eingesetzt

Systematische Faltungscodes:

- Informationsbit ist explizit im Codewort enthalten
- In der Praxis fast bedeutungslos, da geringe Leistungsfähigkeit
- **Ausnahme:** rekursive Faltungscodes (s. Turbo-Codes, TCM in Kanalcodierung II)

Rekursive systematische Faltungscodes (RSC-Codes)
RSC-Codes: *Recursive Systematic Convolutional Codes*

Bei rekursiven Faltungscodes hängt der Folgezustand vom aktuellen Zustand, dem Eingangswert **und der Rückkopplungsstruktur des Codierers ab**. In der Praxis gebräuchliche rekursive Codes sind i.a. systematisch und lassen sich aus nicht-systematischen, nicht-rekursiven (NSC)-Codes ableiten. Den Ausgangspunkt bildet ein NSC-(*nonrecursive nonsystematic convolutional*) Code, dessen Generatorpolynome so umgeformt werden, dass sie einen systematischen, aber rekursiven Code (RSC-Code) beschreiben.

$$
\begin{aligned}
G_1(D) \quad &\longrightarrow \quad \tilde{G}_1(D) = 1 \\
G_2(D) \quad &\longrightarrow \quad \tilde{G}_2(D) = \frac{G_2(D)}{G_1(D)}
\end{aligned}
$$

Die Codeworte des neuen systematischen RSC-Codes berechnen sich wie folgt:

$$\begin{aligned}\tilde{X}_1(D) &= U(D)\cdot\tilde{G}_1(D) = U(D) \quad\longrightarrow\quad \text{systematischer Code}\\ \tilde{X}_2(D) &= U(D)\cdot\tilde{G}_2(D) = \frac{U(D)}{G_1(D)}\cdot G_2(D) = A(D)\cdot G_2(D)\end{aligned}$$

mit

$$A(D) = \frac{U(D)}{G_1(D)} \quad\Longleftrightarrow\quad A(D)\cdot\sum_{i=0}^{m} g_{1,i}\cdot D^i = U(D)\,. \tag{4.10}$$

Mit D als Verzögerungsoperator lauten die Gl. (4.10) im Zeitbereich ($g_{1,0} \stackrel{!}{=} 1$)

$$a(\ell)+\sum_{i=1}^{m} g_{1,i}\cdot a(\ell-i) = u(\ell) \quad\Longrightarrow\quad a(\ell) = u(\ell)+\sum_{i=1}^{m} g_{1,i}\cdot a(l-i)\,.$$

Der Wert $a(\ell)$ kann als Inhalt der aktuellen Registerzelle interpretiert werden und hängt vom aktuellen Eingang $u(\ell)$ und den alten Registerinhalten $a(\ell-i)$ ab. Hierdurch wird die rekursive Struktur des Codierers deutlich.

Beispiel: Faltungscode aus Bild 4.2

$$\begin{aligned}G_1(D) &= 1+D+D^2 \quad\longrightarrow\quad \tilde{G}_1(D) = 1\\ G_2(D) &= 1+D^2 \quad\longrightarrow\quad \tilde{G}_2(D) = \frac{G_2(D)}{G_1(D)} = \frac{1+D^2}{1+D+D^2}\\ \Longrightarrow A(D) &= \frac{U(D)}{G_1(D)} = \frac{U(D)}{1+D+D^2}\\ &\Longleftrightarrow\quad A(D)\cdot[1+D+D^2] = U(D)\\ \Longrightarrow a(\ell) &= u(\ell)+a(\ell-1)+a(\ell-2)\end{aligned}$$

Die neuen Codeworte berechnen sich wie folgt und führen zu dem in Bild 4.5 skizzierten Codierer.

$$\begin{aligned}\tilde{x}_1(\ell) &= u(\ell)\\ \tilde{x}_2(\ell) &= a(\ell)+a(l-2)\end{aligned}$$

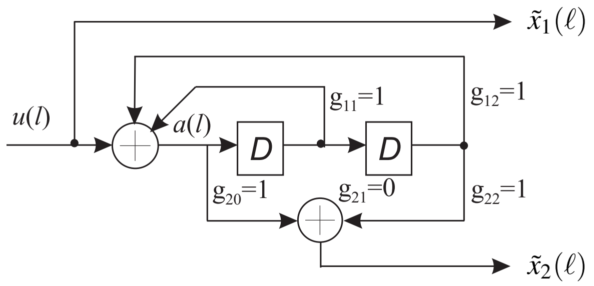


Bild 4.5: Schieberegisterstruktur für rekursiven, systematischen Faltungscode

Eigenschaften rekursiver Codes:

- NSC- und abgeleiteter RSC-Code besitzen die gleichen Distanzeigenschaften (s. später), aber unterschiedliches Ein- / Ausgangsverhalten
- Unendlich lang abklingende Impulsantwort (IIR-Filter)

→ Mindestens ein Gewicht von 2 in Eingangssequenz für endliches Ausgangsgewicht erforderlich

4.2.2 Katastrophale Codes

- Katastrophale Codes können unendlich lange Sequenz mit endlichem Gewicht erzeugen, die nicht auf den Nullpfad zurückkehrt

→ Katastrophale Codes erzeugen bei endlich vielen Übertragungsfehlern u.U. unendlich viele Fehler nach der Decodierung

→ Katastrophale Codes sind nicht zur Übertragung geeignet und daher zu vermeiden

- Merkmale katastrophaler Codes:
  - Alle Generatorpolynome besitzen gemeinsamen Faktor
  - Im Zustandsdiagramm existiert geschlossene Schleife mit Gewicht Null (außer Nullzustand)
  - Alle Addierer haben gerade Anzahl von Verbindungen
    $\longrightarrow$ Selbstschleife im Zustand $1 \cdots 1$ hat Gewicht Null

Beispiel für katastrophalen Code: $g_1 = 5$, $g_2 = 3$

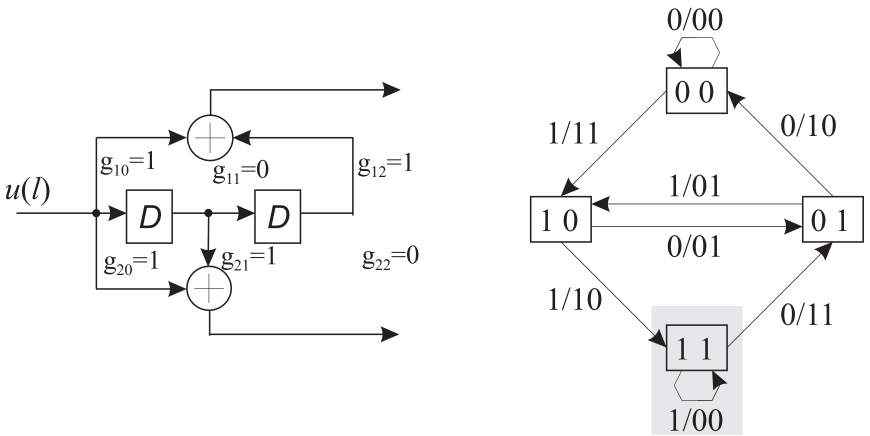


Bild 4.6: Schieberegisterstruktur und Zustandsdiagramm für katastrophalen Faltungscode

4.2.3 *Truncated Convolutional Codes*

Die Decodierung von Faltungscodes erfordert die Betrachtung der gesamten Sequenz bzw. eines ausreichend langen Ausschnitts (s. Viterbi-Algorithmus). Dabei sind in der Praxis verständlicherweise nur Sequenzen mit endlicher Länge (N Codeworte) von Bedeutung. Bei einem willkürlichen Ende der Informationsfolge $\mathbf{u}$ kann das Trellisdiagramm in jedem beliebigen Zustand enden, d.h. der Endzustand ist dem Decodierer unbekannt. Dies wirkt sich auf die Leistungsfähigkeit der Decodierung aus, da sich die letzten Bit in $\mathbf{u}$ nur sehr unsicher schätzen lassen. Bei der Betrachtung endlicher Sequenzen sind Faltungscodes auch als Blockcode interpretierbar und lassen sich dementsprechend durch eine Generatormatrix beschreiben:

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \cdots & \mathbf{G}_m & & & & \\ & \mathbf{G}_0 & \mathbf{G}_1 & \cdots & \mathbf{G}_m & & & \\ & & \ddots & \ddots & & \ddots & & \\ & & & \mathbf{G}_0 & \mathbf{G}_1 & \cdots & \mathbf{G}_m & \\ & & & & \ddots & \ddots & \vdots & \\ & & & & & \mathbf{G}_0 & \mathbf{G}_1 & \\ & & & & & & \mathbf{G}_0 & \end{bmatrix} \qquad \text{mit} \qquad \mathbf{G}_i = [g_{1,i}\; g_{2,i}\; \cdots\; g_{n,i}] \;.$$

4.2.4 Terminierte Faltungscodes

Durch das Anhängen von sogenannten *Tailbit* kann der Codierer am Ende der Informationsfolge wieder in einen bekannten Endzustand (in der Regel der Nullzustand) überführt werden. Damit lassen sich auch die letzten Bit zuverlässig schätzen. Die Anzahl der Tailbit entspricht exakt dem Gedächtnis des Codierers, die Wertigkeit der Tailbit hängt dagegen von dessen Struktur ab. Hier gilt

- NSC-Codes: Anhängen von m Nullen
- RSC-Codes: m Tailbit hängen vom letzten Zustand vor dem Anhängen der Tailbit ab (in Tabelle ablegen)

Das Anhängen von Tailbit reduziert allerdings die Coderate, d.h. für eine Sequenz $\mathbf{u}$ mit N Informationsbit gilt bei einem $1/n$-ratigen Faltungscode

$$R_c^{Tail} = \frac{N}{n \cdot (N+m)} = R_c \cdot \frac{N}{N+m} \, . \tag{4.11}$$

Je größer N im Vergleich zur Einflusslänge bzw. dem Gedächtnis des Codierers ist, desto weniger fallen die angehängten Tailbit bzgl. der Coderate ins Gewicht. Die Beschreibung mittels Generatormatrix lautet nun:

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \cdots & \mathbf{G}_m & & & \\ & \mathbf{G}_0 & \mathbf{G}_1 & \cdots & \ddots & & \\ & & \ddots & \ddots & \cdots & & \\ & & & \mathbf{G}_0 & \mathbf{G}_1 & \cdots & \mathbf{G}_m \end{bmatrix} \qquad \text{mit} \qquad \mathbf{G}_i = [g_{1,i}\, g_{2,i}\, \cdots\, g_{n,i}] \tag{4.12}$$

4.2.5 *Tailbiting convolutional codes*

Sollen Datenpakete mit sehr niedriger Länge übertragen werden, wie es beispielsweise für moderne paketorientierte Übertragungsprotokolle vorgesehen ist, so führt das Anhängen von Tailbit allerdings doch zu einem merklichen Verlust der zur Verfügung stehenden Datenrate. Hier bieten sich die *Tailbiting Convolutional Codes* an, bei denen der Endzustand des Codierers mit dem Startzustand identisch ist. Daher sind keine Tailbits erforderlich. Für nicht-rekursive Codes kann der gesamte Informationsblock codiert und der Endzustand des Codierers bestimmt werden. Dann wird der Endzustand als Anfangszustand gewählt und erneut codiert. Man erhält einen quasi-zyklischen Faltungscode. Die Generatormatrix lautet

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_0 & \mathbf{G}_1 & \cdots & \mathbf{G}_m & & & & \\ & \mathbf{G}_0 & \mathbf{G}_1 & \cdots & \mathbf{G}_m & & & \\ & & \ddots & \ddots & & \ddots & & \\ & & & \mathbf{G}_0 & \mathbf{G}_1 & \cdots & \mathbf{G}_m \\ \mathbf{G}_m & & & & \ddots & \ddots & \vdots \\ \vdots & \ddots & & & & \mathbf{G}_0 & \mathbf{G}_1 \\ \mathbf{G}_1 & \cdots & \mathbf{G}_m & & & & \mathbf{G}_0 \end{bmatrix} .$$

4.3 Optimale Decodierung mit Viterbi-Algorithmus

Wir wollen zunächst einige Vereinbarungen bzgl. der Nomenklatur treffen. Die Informationssequenz $\mathbf{u}$ besteht aus k Informationsbit und wird durch den Faltungscodierer in die codierte Sequenz

$$\mathbf{x} = (\underbrace{x_1(0) \cdots x_n(0)}_{\mathbf{x}(0)} \cdots \underbrace{x_1(N-1) \cdots x_n(N-1)}_{\mathbf{x}(N-1)})$$

bestehend aus n Symbolen umgesetzt. Ferner bezeichnet $\mathbf{y}$ die empfangene Symbolfolge, $\hat{\mathbf{x}}$ die durch den Decodierer geschätzte codierte Folge und $\mathbf{a} \in \Gamma$ eine beliebige codierte Sequenz.

Bei der Decodierung unterscheidet man prinzipiell zwischen Verfahren, die komplette Codeworte (Blockcodes) bzw. Codesequenzen (Faltungscodes) schätzen und solchen, die symbolweise arbeiten. Letztere bestimmen Schätzwerte für jedes Codebit, wobei ein Codierer-Inverses dann die Abbildung auf das zugehörige Informationswort realisiert. Da symbolweise entschieden wird, kann es vorkommen, dass die Codesequenz / das Codewort nicht Element des Coderaums ist und der Decodiervorgang versagt. Wir betrachten in diesem Abschnitt die sequenzweise Decodierung, wobei wiederum zwischen 2 Kriterien unterschieden werden muss.

MAP-Kriterium

Das MAP-Kriterium (Maximum A-posteriori Probability) ist bereits von den Blockcodes bekannt und stellt die optimale Decodierung dar. Dabei wird die Sequenz $\hat{\mathbf{x}}$ bestimmt, die die a-posteriori-Wahrscheinlichkeit $P(\hat{\mathbf{x}} \mid \mathbf{y})$ maximiert.

$$\begin{aligned} P(\hat{\mathbf{x}} \mid \mathbf{y}) &\geq P(\mathbf{a} \mid \mathbf{y}) && (4.13) \\ p(\mathbf{y} \mid \hat{\mathbf{x}}) \cdot \frac{P(\hat{\mathbf{x}})}{p(\mathbf{y})} &\geq p(\mathbf{y} \mid \mathbf{a}) \cdot \frac{P(\mathbf{a})}{p(\mathbf{y})} \\ p(\mathbf{y} \mid \hat{\mathbf{x}}) \cdot P(\hat{\mathbf{x}}) &\geq p(\mathbf{y} \mid \mathbf{a}) \cdot P(\mathbf{a}) && (4.14) \end{aligned}$$

Die Decodierung erfolgt über $\hat{\mathbf{x}} = \arg\max_{\mathbf{a}}(p(\mathbf{y}|\mathbf{a})P(\mathbf{a}))$ und berücksichtigt durch $P(\hat{\mathbf{x}})$ eine a-priori-Information der Quellenstatistik.

Maximum Likelihood-Kriterium

Treten alle codierten Sequenzen mit der gleichen Wahrscheinlichkeit $P(\hat{\mathbf{x}}) = P(\mathbf{a}) = 2^{-k}$ auf bzw. ist die Quellenstatistik dem Empfänger nicht bekannt, so kann keine a-priori-Information bei der Decodierung berücksichtigt werden. Dann gilt

$$p(\mathbf{y} \mid \hat{\mathbf{x}}) \geq p(\mathbf{y} \mid \mathbf{a}) \; . \tag{4.15}$$

Die Decodierung erfolgt über $\hat{\mathbf{x}} = arg \max_{\mathbf{a}} p(\mathbf{y}|\mathbf{a})$. Für gleichverteilte Eingangssequenzen liefern MAP- und *Maximum-Likelihood*-Kriterium identische (optimale) Ergebnisse. Sind die Eingangssequenzen nicht gleichverteilt und $P(\mathbf{a})$ dem Empfänger nicht bekannt, so ist das *Maximum-Likelihood*-Kriterium suboptimal.

Im Folgenden werden wir die *Maximum Likelihood*-Decodierung weiter verfolgen, eine Erweiterung auf die Decodierung mit dem MAP-Kriterium ist dann einfach möglich. Für diskreten gedächtnislosen Kanal (DMC) können die Verbundwahrscheinlichkeiten faktorisiert werden und es gilt

$$p(\mathbf{y} \mid \mathbf{a}) = \prod_{\ell=0}^{N-1} p\left(\mathbf{y}(\ell) \mid \mathbf{a}(\ell)\right) = \prod_{\ell=0}^{N-1} \prod_{i=1}^{n} p\left(y_i(\ell) \mid a_i(\ell)\right) \; . \tag{4.16}$$

Da ferner der ln eine streng monoton steigende Funktion ist, gilt auch

$$\begin{aligned} \ln p(\mathbf{y} \mid \mathbf{a}) &= \ln \prod_{\ell=0}^{N-1} \prod_{i=1}^{n} p\left(y_i(\ell) \mid a_i(\ell)\right) \\ &= \sum_{\ell=0}^{N-1} \sum_{i=1}^{n} \ln p\left(y_i(\ell) \mid a_i(\ell)\right) \\ &= \sum_{\ell=0}^{N-1} \sum_{i=1}^{n} \gamma\left(y_i(\ell) \mid a_i(\ell)\right) \; . && (4.17) \end{aligned}$$

Der Ausdruck $\gamma'(y_i(\ell) \mid a_i(\ell))$ wird auch **Viterbi-Metrik** benannt und beschreibt die Übergangswahrscheinlichkeiten des Kanals. Für den Spezialfall des AWGN-Kanals sind die Verhältnisse in Bild 4.7 dargestellt.

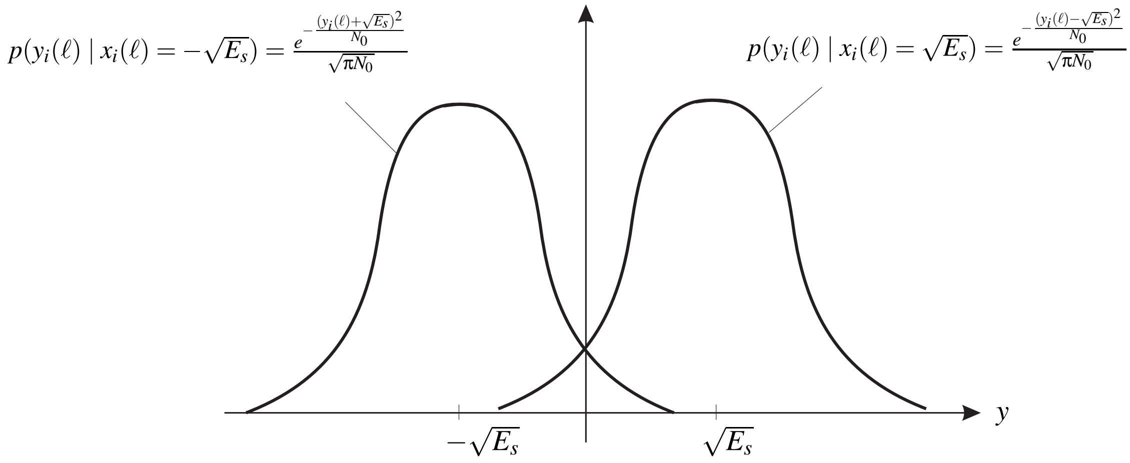


Bild 4.7: Bedingte Wahrscheinlichkeitsdichtefunktionen beim AWGN-Kanal mit binärem Eingang

Hier heben sich die Exponentialfunktion der Gaußverteilung und der ln gegenseitig auf, so dass besonders einfache Verhältnisse vorliegen. In der Praxis sind 2 äquivalente Metriken von Bedeutung:

1. Quadratische euklidische Distanz:

$$\begin{aligned}
p(y_i(\ell) \mid a_i(\ell)) &= \frac{1}{\sqrt{\pi N_0}} \cdot e^{-\frac{(y_i(\ell)-a_i(\ell))^2}{N_0}} \\
\Longrightarrow \gamma'(y_i(\ell) \mid a_i(\ell)) &= \ln p(y_i(\ell) \mid a_i(\ell)) = C - \frac{(y_i(\ell)-a_i(\ell))^2}{N_0} \\
\Longrightarrow \gamma(y_i(\ell) \mid a_i(\ell)) &= \frac{(y_i(\ell)-a_i(\ell))^2}{N_0} \qquad (4.18)
\end{aligned}$$

2. Korrelationsmetrik:

$$\gamma'(y_i(\ell) \mid a_i(\ell)) = C - \frac{y_i(\ell)^2}{N_0} + 2\frac{a_i(\ell)y_i(\ell)}{N_0} - \frac{a_i(\ell)^2}{N_0} = \underbrace{C - \frac{y_i(\ell)^2}{N_0}}_{\text{unabhg. von } a_i(\ell)} + \frac{a_i(\ell)y_i(\ell)}{N_0/2} \underbrace{-\frac{E_s}{N_0}}_{\text{konstant}}$$

$$\gamma(y_i(\ell) \mid a_i(\ell)) = \frac{2}{N_0} \cdot a_i(\ell) \cdot y_i(\ell) \qquad (4.19)$$

Eine direkte Umsetzung der *Maximum-Likelihood*-Decodierung kann nun beispielsweise so erfolgen, dass die $\gamma(y_i(\ell) \mid a_i(\ell))$ für alle möglichen Codesequenzen $\mathbf{a}$ aufsummiert werden. Als Lösung erhalten wir dann die Sequenz $\hat{\mathbf{x}}$ mit der geringsten euklidischen Distanz oder aber der maximalen Korrelationsmetrik zur empfangenen Folge $\mathbf{y}$. Diese maximiert die bedingte Wahrscheinlichkeit $p(\mathbf{y}|\hat{\mathbf{x}})$. Diese direkte Umsetzung hat den Nachteil, dass sie viel zu aufwendig ist (Anzahl der Sequenzen wächst exponentiell mit ihrer Länge) und daher in der Praxis nicht realisiert werden kann.

Dieses Problems kann durch Ausnutzung der Markov-Eigenschaft von Faltungscodes gelöst werden. Die Markov-Eigenschaft besagt nämlich, dass der aktuelle Zustand nur vom Vorzustand und dem aktuellen Eingangswert abhängt. Damit ist es möglich, die Pfadmetriken sukzessive zu berechnen. Ein effizienter Verfahren stellt der **Viterbi-Algorithmus** dar, welcher im Folgenden genauer beschrieben wird.

Viterbi-Algorithmus

1. Beginne Trellis im Nullzustand zum Zeitpunkt $\ell = 0$
2. Berechne $\gamma(\mathbf{y}(\ell) \mid \mathbf{a}(\ell))$ zwischen empfangenen Codewort $\mathbf{y}(\ell)$ und allen möglichen Codeworten $\mathbf{a}(\ell)$
3. Addiere unter 2) berechnete Pfadmetriken zu alten Zustandsmetriken $M_j(\ell-1)$, $j = 0 \dots 2^m - 1$
4. An jedem Zustand Auswahl desjenigen Pfades mit kleinster euklidischer Distanz (größter Korrelationsmetrik) und Verwerfung (keine weitere Berücksichtigung) der anderen Pfade
   $\Longrightarrow$ Aufwand wächst **nur linear** mit Pfadlänge (nicht exponentiell)
5. Wiederholung ab 2), bis alle N empfangenen Worte abgearbeitet wurden
6. Ende des Trellisdiagramms:
   - Terminierte Codes (Trellis endet im Nullzustand):
     $\Longrightarrow$ Bestimmung des Pfades mit der besten Metrik $M_0(N)$ im Nullzustand
   - Nicht-terminierte Codes:
     $\Longrightarrow$ Bestimmung des Pfades mit der global besten Zustandsmetrik $M_j(N)$, $j = 0 \cdots 2^m - 1$
7. Zurückverfolgen des in 6) bestimmten Pfades (*Survivor*) und Ausgabe der zugehörigen Informationsbit

Beispiel: 1/2-ratiger Faltungscode mit $L_c = 3$, $g_1 = 7_8$, $g_2 = 5_8$ und Sequenzlänge $N = 6$ ($K = 4$)

- Informationssequenz: $\mathbf{u} = (1\,0\,0\,1)$, Tailbit: $(0\,0)$
- Codierte Folge: $\mathbf{x} = (+1\ +1\ \ +1\ -1\ \ +1\ +1\ \ +1\ +1\ \ \underbrace{+1\ -1\ \ +1\ +1}_{\text{für Tailbit}})$
- Hard-Decision-Decodierung
- Empfangene Folge: $\mathbf{y} = (\underbrace{-1\ -1}_{f}\ \ +1\ -1\ \ +1\ +1\ \ +1\ +1\ \ +1\ -1\ \ \underbrace{-1}_{f}\ +1)$

  Die Empfangsfolge $\mathbf{y}$ (*Hard-Decision*) enthält 3 Fehler, die i.a. wegen $d_f = 5$ nicht korrekt decodiert werden können. Allerdings spielt bei Faltungscodes auch die Verteilung der Fehler in der Sequenz eine wichtige Rolle. Einzelne, weit auseinander liegende Fehler können unter Umständen noch korrigiert werden, auch wenn die Gesamtfehlerzahl die halbe Mindestdistanz übersteigt. Bild 4.8 illustriert die Arbeitsweise des Viterbi-Algorithmus für das obige Beispiel im Trellisdiagramm.

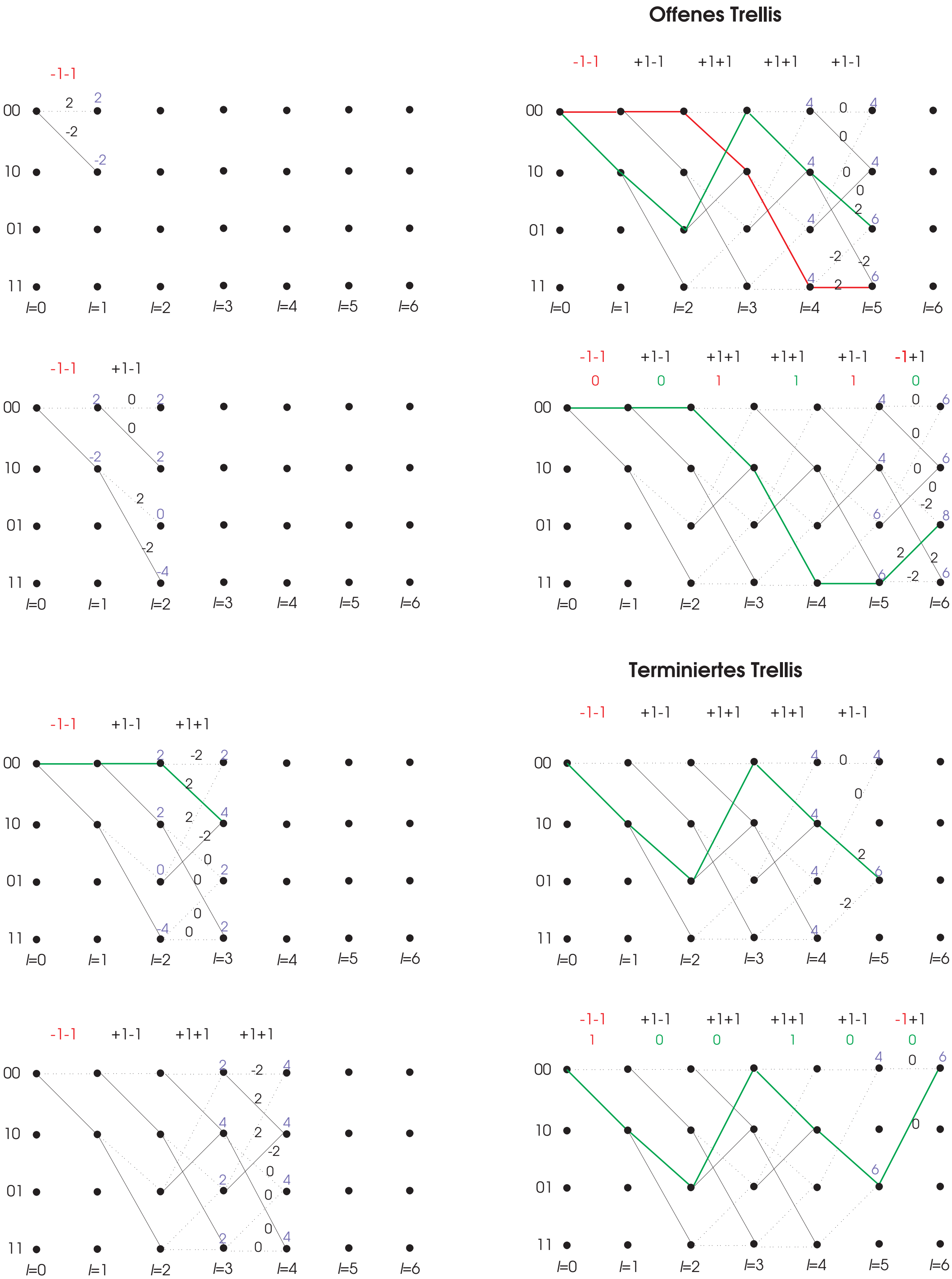


Bild 4.8: Trellisdiagramm-Abarbeitung zur Veranschaulichung des Viterbi-Algorithmus

Es ist zu erkennen, dass die Decodierung bei einem offenem Trellis eine falsche Folge ausgeben würde (wäre das vorletzte Bit korrekt gewesen, hätte die Decodierung das richtige Ergebnis geliefert). Bei Verwendung von Tailbit (terminiertes Trellis) sorgt die Kenntnis des Endzustands dafür, dass sich der letzte Fehler nicht mehr störend auswirkt und korrekte Folge decodiert wird. Dies verdeutlicht, dass bei einem offenem Trellisdiagramm über die letzten Bit nur sehr unsicher entschieden werden kann.

Faustregel:
Bei kontinuierlicher Datenübertragung (oder auch sehr langen Blöcken) wird das Trellis mit einer Entscheidungstiefe (Rückgrifftiefe) von ca. $5 \cdot L_c$ abgearbeitet, bis über ein Bit entschieden wird.

Grund:
Ist die Entscheidungstiefe groß genug, verschmelzen die Anfänge der verschiedenen Pfade miteinander und die Entscheidung ist in diesem Bereich eindeutig.

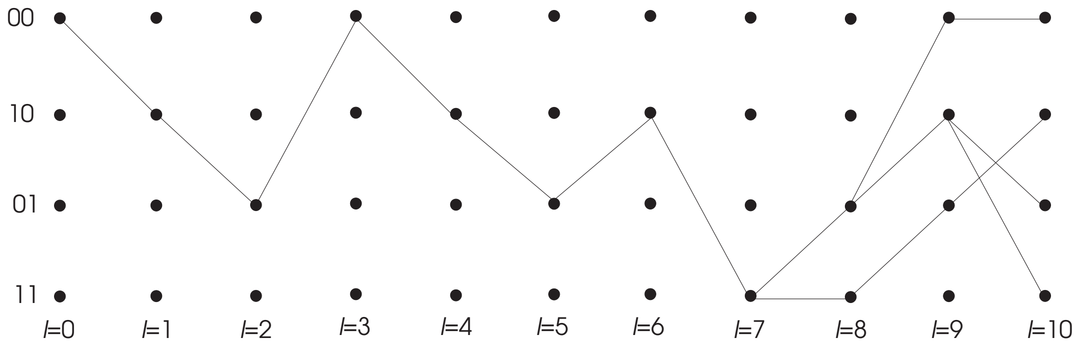


Bild 4.9: Veranschaulichung der Bildung eines *Survivors*

4.4 Punktieren von $1/n$-ratigen Faltungscodes

Bezüglich der Modifikation von Blockcodes sind schon die Methoden 'Expandieren', 'Punktieren', 'Verkürzen' und 'Verlängern' bekannt. Bei den Faltungscodes spielt dagegen nur das Punktieren eine praktische Rolle. Die Einstellung der Coderate erfolgt durch Ausblenden (nicht Übertragen) bestimmter Codesymbole, wobei kein Unterschied zwischen Informationsbit und Prüfbit gemacht wird. Dem Empfänger muss das Punktierungsschema selbstverständlich bekannt sein.

Vorteile der Punktierung:

- Kein mehrfacher Hardware-Aufwand bei flexibler Coderate erforderlich
(einfache Anpassung der Coderate an die aktuellen Übertragungsbedingungen)
- Unter Umständen geringerer Aufwand bei der Decodierung (s. später)
- Punktierung verringert zwar Leistungsfähigkeit des Originalcodes, der punktierte Code ist i. a. aber genauso gut wie unpunktierte Codes **gleicher** Rate

Die Punktierung erfolgt normalerweise periodisch mit der Periode L_P und läßt sich dann mit Hilfe einer Punktierungsmatrix $\mathbf{P}$ beschreiben.

$$\mathbf{P} = \begin{bmatrix} p_{1,0} & p_{1,1} & \cdots & p_{1,L_P-1} \\ p_{2,0} & p_{2,1} & \cdots & p_{2,L_P-1} \\ \vdots & \vdots & & \vdots \\ p_{n,0} & p_{n,1} & \cdots & p_{n,L_P-1} \end{bmatrix} \tag{4.20}$$

$$= \begin{bmatrix} \mathbf{p}_0 & \mathbf{p}_1 & \cdots & \mathbf{p}_{L_P-1} \end{bmatrix} \tag{4.21}$$

Jede Spalte $\mathbf{p}_i$ von $\mathbf{P}$ enthält das Punktierungsschema für ein Codewort und besteht somit aus n Elementen. Statt der ursprünglich $n \cdot L_P$ Bit werden jetzt durch die Punktierung nur noch $\ell + L_P$ Bit übertragen, wobei der Parameter ℓ im Bereich $1 \leq \ell \leq (n-1) \cdot L_P$ liegt. Dadurch können Coderaten im Bereich von

$$\ell = 1 \qquad \longrightarrow \quad R_c' = \frac{L_p}{L_P+1}$$

bis

$$\ell = (n-1) \cdot L_P \quad \longrightarrow \quad R_c' = \frac{L_p}{L_P \cdot n} = \frac{1}{n}$$

eingestellt werden. Die Struktur von $\mathbf{P}$ ist nur in einfachsten Fällen trivial. So kann beispielsweise durch die Punktierung ein katastrophaler Code entstehen. In jedem Fall ist die Punktierung auf die jeweiligen Generatoren abzustimmen. Ferner ist zu beachten, dass durch die Punktierung ein **zeitvarianter Faltungscode** entsteht, der eine erweiterte Beschreibung erfordert, auf die an dieser Stelle aber nicht weiter eingegangen werden soll.

Beispiel: Punktierungsperiode $L_P = 2$, Codewortlänge $n = 2$, Originalcoderate $R_c = 1/2$

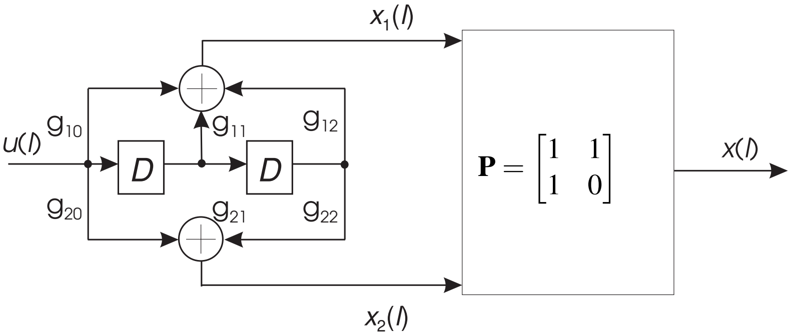


Bild 4.10: Punktierung eines halbratigen Faltungscodes mit $\ell = 1$ zur Rate $R_c' = 2/3$

Punktierungsmatrix: $\mathbf{P} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = [\mathbf{p}_0 \, \mathbf{p}_1]$

Punktierung mit $\ell = 1 \quad \longrightarrow \quad R_c' = 2/3$

$$x_1(0)\, x_2(0)\, x_1(1)\, x_2(1)\, x_1(2)\, x_2(2)\, x_1(3)\, x_2(3) \;\longrightarrow\; x_1(0)\, x_2(0)\, x_1(1) \quad x_1(2)\, x_2(2)\, x_1(3) \quad x_1(4) \ldots$$

Anwendung:
Rate-compatible punctured convolutional codes (RCPC-Codes) erlauben eine adaptive Anpassung der Coderate an aktuelle Übertragungsbedingungen. Unter der Voraussetzung, dass Sender und Empfänger die Übertragungsbedingungen und das Punktierungsschema austauschen können, steht somit eine einfache Möglichkeit der adaptiven Systemoptimierung zur Verfügung.

Decodierung punktierter Codes:
Vor der Decodierung sind zunächst Platzhalter für die punktierten Bit einzufügen (z.B. Nullen bei einer antipodalen Übertragung). Da sich die Distanzeigenschaften des Codes durch die Punktierung verschlechtern, muss die Entscheidungstiefe entsprechend verlängert werden, um weiterhin sichere Entscheidung zu ermöglichen.

4.5 Distanzeigenschaften von Faltungscodes

Von den Blockcodes ist schon bekannt, dass die Distanzeigenschaften die Leistungsfähigkeit eines Codes, d.h. seine Fähigkeit, Fehler zu erkennen oder zu korrigieren, bestimmen. Dies gilt in gleichem Maße für Faltungscodes: Bei ihnen ist allerdings nicht die Distanz zwischen einzelnen Codeworten, sondern die Distanz zwischen ganzen Codeesequenzen entscheidend. Als Äquivalent zur Mindestdistanz der Blockcodes wird hier die **freie Distanz** d_f benutzt, welche die minimale Hamming-Distanz zwischen zwei Sequenzen (siehe Trellisdiagramm in Bild 4.12) angibt.

Die freie Distanz bestimmt die asymptotische ($E_b/N_0 \to \infty$) Leistungsfähigkeit des Faltungscodes. Für mittlere und kleine Signal-Rausch-Abstände wirken sich die übrigen Distanzen ebenfalls aus, so dass hier das gesamte Distanzspektrum zu betrachten ist.

Distanzspektrum

- Faltungscodes sind linear $\longrightarrow$ Vergleich aller Sequenzen mit dem Nullpfad ausreichend (anstatt alle Sequenzen untereinander zu vergleichen)

$\rightarrow$ Hamming-Gewicht aller Sequenzen muss bestimmt werden

- Zur Bestimmung des Distanzspektrums modifiziertes Zustandsdiagramm:
  - Auftrennen der Selbstschleife im Nullzustand
  - Nullzustand als Startzustand S_a und Endzustand S_e anordnen
  - An Pfaden (Zustandsübergänge) stehen Platzhalter für:
    - Sequenzlänge: L
    - Gewicht der uncodierten Eingangssequenz: W
    - Gewicht der codierten Ausgangssequenz: D
  - Bestimmen aller Übergangsmöglichkeiten zwischen Start- und Endzustand

Beispiel: Faltungscode mit $R_c = 1/2$, $L_c = 3$, $\mathbf{g}_1 = 7_8$, $\mathbf{g}_2 = 5_8$

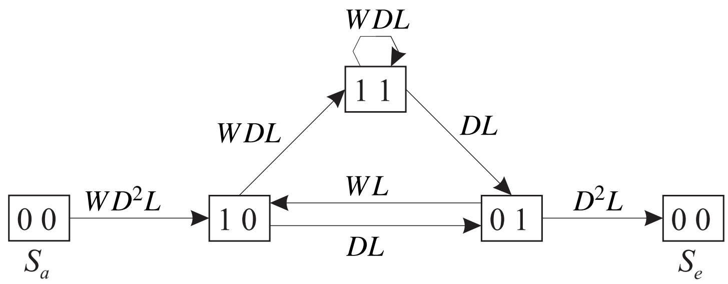


Bild 4.11: Modifiziertes Zustandsdiagramm zur Bestimmung des Distanzspektrums

Lineares Gleichungssystem:

$$\begin{aligned}
S_{10} &= WD^2L \cdot S_a + WL \cdot S_{01} \\
S_{01} &= DL \cdot S_{10} + DL \cdot S_{11} \\
S_{11} &= WDL \cdot S_{11} + WDL \cdot S_{10} \\
S_e &= D^2L \cdot S_{01}
\end{aligned}$$

Lösung:

$$\frac{S_e}{S_a} = \frac{WD^5L^3}{1 - WDL - WDL^2} =: T\,(W,D,L)$$

Reihenentwicklung:

$$\begin{aligned}
T\,(W,D,L) &= WD^5L^3 + \\
&\quad W^2D^6L^4 + W^2D^6L^5 + \\
&\quad W^3D^7L^5 + 2W^3D^7L^6 + W^3D^7L^7 + \ldots \\
&= \sum_w \sum_d \sum_\ell T_{w,d,\ell} \cdot W^w \cdot D^d \cdot L^\ell
\end{aligned} \tag{4.22}$$

Interpretation:

- 1 Sequenz der Länge $\ell = 3$ (Codeworte) mit Eingangsgewicht $w = 1$ und Ausgangsgewicht $d = 5$
- 2 Sequenzen mit Eingangsgewicht $w = 2$ und Ausgangsgewicht $d = 6$ und den Längen $\ell = 4$ bzw. $\ell = 5$
- Je 1 Sequenz mit Eingangsgewicht $w = 3$ und Ausgangsgewicht $d = 7$ und den Längen $\ell = 5$ bzw. $\ell = 7$ und 2 Sequenzen der Länge $\ell = 6$ mit Eingangsgewicht $w = 3$ und Ausgangsgewicht $d = 7$
- Bild 4.12 zeigt für den im Beispiel betrachteten Faltungscode das Trellisdiagramm mit allen Sequenzen bis zum Gewicht $d = 6$

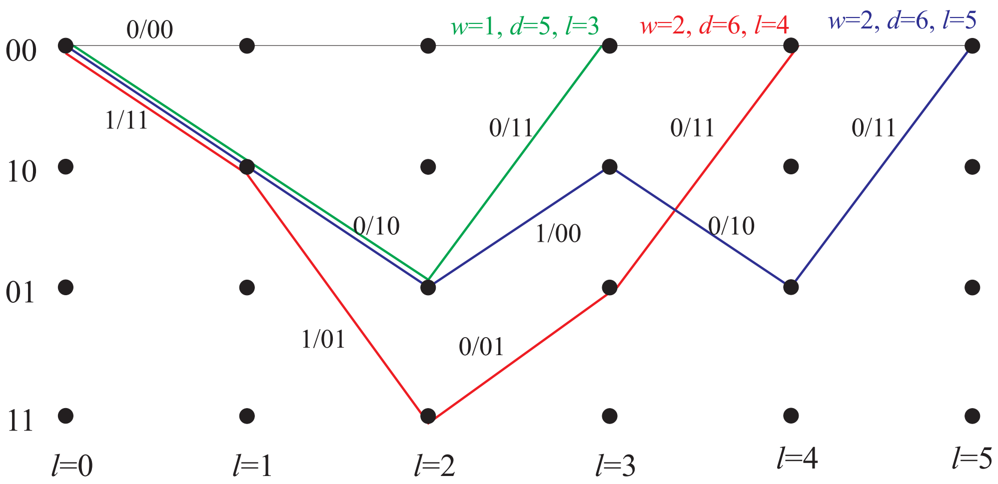


Bild 4.12: Ausschnitt des Trellisdiagramms für Faltungscode aus Bild 4.2 zur Veranschaulichung der freien Distanz

Die oben beschriebene Vorgehensweise kann mathematisch recht anschaulich mit Hilfe von Matrizen dargestellt werden. Es gilt

$$T\,(W,D,L) = \sum_{p=0}^{\infty} \mathbf{a}\mathbf{S}^p\mathbf{b}\,. \tag{4.23}$$

Dabei beschreibt der Vektor **a** den Start im Trellisdiagramm vom Nullzustand in alle übrigen Zustände mit den Parametern W, D und L. Da die Selbstschleife des Nullzustands entsprechend Bild 4.11 aufgetrennt wurde, existiert nur ein Übergang zum Zustand S_{10} mit den Parametern WD^2L und **a** hat die Form

$$\mathbf{a} = \begin{bmatrix} 0 & WD^2L & 0 \end{bmatrix} . \tag{4.24}$$

Die Matrix **S** stellt die Übergänge zwischen den Zuständen S_{01} bis S_{11} (ohne Nullzustand) dar und **b** gibt die Übergänge von allen Zuständen in den Nullzustand an. Für den oben betrachteten Code gilt:

$$\mathbf{S} = \begin{array}{c} \begin{bmatrix} 0 & WL & 0 \\ DL & 0 & WDL \\ DL & 0 & WDL \end{bmatrix} \end{array} \quad \begin{array}{l} \text{alter Zustand } S_{01} \\ \text{alter Zustand } S_{10} \\ \text{alter Zustand } S_{11} \end{array} \tag{4.25}$$

$$\mathbf{b} = \begin{bmatrix} D^2L \\ 0 \\ 0 \end{bmatrix} \quad \begin{array}{l} \text{Zustand } S_{01} \\ \text{Zustand } S_{10} \\ \text{Zustand } S_{11} \end{array} \tag{4.26}$$

Vom Zustand 10 sind die Zustände 01 und 11 zu erreichen, wodurch sich die Einträge in der 2. Zeile von **S** erklären lassen. Für jeden Übergang im Trellisdiagramm ist mit **S** zu multiplizieren, so dass für eine Sequenz der Länge L der Exponent von **S** genau $p = L - 2$ beträgt. Der Vektor **b** schließt das Trellis letztendlich im Nullzustand ab, da dieser nur über den Zustand S_{01} zu erreichen ist, enthält **b** nur in der ersten Spalte einen von Null verschiedenen Eintrag. Wir wollen nun für das obige Beispiel alle Sequenzen bis zur Länge $\ell \leq 5$ bestimmen, die im Nullzustand beginnen und enden. Wir erhalten

$$\begin{aligned}
T_{\ell=3}(W,D,L) &= \mathbf{aSb} = \begin{bmatrix} 0 & WD^2L & 0 \end{bmatrix} \begin{bmatrix} 0 & WL & 0 \\ DL & 0 & WDL \\ DL & 0 & WDL \end{bmatrix} \begin{bmatrix} D^2L \\ 0 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} WD^3L^2 & 0 & W^2D^3L^2 \end{bmatrix} \begin{bmatrix} D^2L \\ 0 \\ 0 \end{bmatrix} \\
&= WD^5L^3
\end{aligned}$$

$$\begin{aligned}
T_{\ell=4}(W,D,L) &= \mathbf{aS}^2\mathbf{b} = \begin{bmatrix} WD^3L^2 & 0 & W^2D^3L^2 \end{bmatrix} \begin{bmatrix} 0 & WL & 0 \\ DL & 0 & WDL \\ DL & 0 & WDL \end{bmatrix} \begin{bmatrix} D^2L \\ 0 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} W^2D^4L^3 & W^2D^3L^3 & W^3D^4L^3 \end{bmatrix} \begin{bmatrix} D^2L \\ 0 \\ 0 \end{bmatrix} \\
&= W^2D^6L^4
\end{aligned}$$

$$\begin{aligned}
T_{\ell=5}(W,D,L) &= \mathbf{aS}^3\mathbf{b} = \begin{bmatrix} W^2D^4L^3 & W^2D^3L^3 & W^3D^4L^3 \end{bmatrix} \begin{bmatrix} 0 & WL & 0 \\ DL & 0 & WDL \\ DL & 0 & WDL \end{bmatrix} \begin{bmatrix} D^2L \\ 0 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} W^2D^4L^4 + W^3D^5L^4 & W^3D^4L^4 & W^3D^4L^4 + W^4D^5L^4 \end{bmatrix} \begin{bmatrix} D^2L \\ 0 \\ 0 \end{bmatrix} \\
&= W^2D^6L^5 + W^3D^7L^5
\end{aligned}$$

$$\Rightarrow T_{\ell\leq 5}(W,D,L) = WD^5L^3 + W^2D^6L^4 + W^2D^6L^5 + W^3D^7L^5$$

Wichtige Parameter zur Beschreibung der Distanzeigenschaften:

- Anzahl der Sequenzen mit bestimmten Hamming-Gewicht d:

$$a_d = \sum_w \sum_\ell T_{w,d,\ell} \tag{4.27}$$

- Anzahl der 'Informationsbit gleich Eins' ($w = 1$) für alle Sequenzen mit Hamming-Gewicht d

$$\left.\frac{\partial T(W,D,L=1)}{\partial W}\right|_{W=1} = \sum_d \left(\sum_w \sum_\ell \cdot T_{w,d,\ell} \right) \cdot D^d = \sum_d c_d \cdot D^d$$

$$\Longrightarrow \quad c_d = \sum_w \sum_\ell w \cdot T_{w,d,\ell} \tag{4.28}$$

Beispiel: Codes mit $R_c = 1/2$, $L_c = 3$

NSC-Code: $G_1(D) = 1 + D + D^2$, $G_2(D) = 1 + D^2$
RSC-Code: $G_1(D) = 1$, $G_2(D) = \frac{1+D^2}{1+D+D^2}$

- Für diesen speziellen NSC-Code existieren für jedes Eingangsgewicht w nur Sequenzen mit einem einzigen bestimmten Ausgangsgewicht $d(w)$ (diese Eigenschaft ist nicht zu verallgemeinern)
- Kleinste Distanz $d_{\min} = d_f = 5$ wird bei NSC-Code für $w = 1$ erreicht
- Anzahl der Pfade mit größeren Distanzen nimmt exponentiell zu
- Bei RSC-Code erst ab $w \geq 2$ Ausgangssequenzen mit endlichem Gewicht ($d_f = 5$ für $w = 2$)
- Besonderheit: endliche Gewichte der Ausgangssequenzen nur für gerade Eingangsgewichte

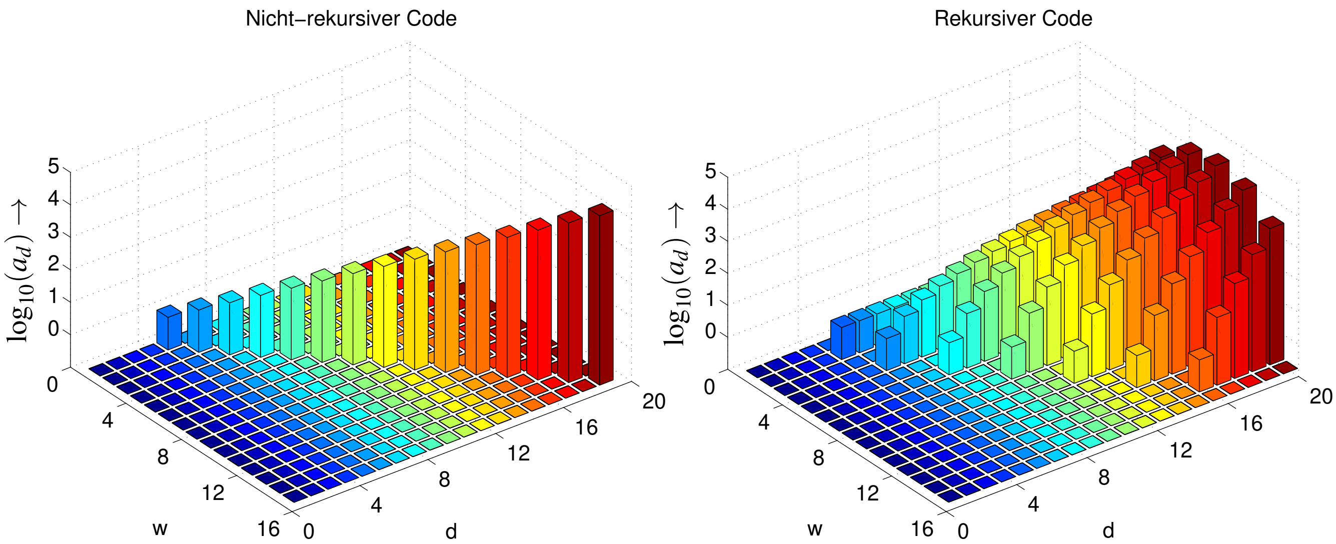


Bild 4.13: Distanzspektren für nicht-rekursiven und rekursiven Faltungscode

4.6 Abschätzung der Fehlerwahrscheinlichkeit

Aus Kapitel 3 ist schon von den Blockcodes bekannt, dass ein Fehler auftritt, wenn die bedingte Wahrscheinlichkeit für die korrekte Codesequenz $\mathbf{x}$ kleiner als für eine andere Folge $\mathbf{a} \neq \mathbf{x}$ ist ($P(\mathbf{y} \mid \mathbf{x}) < P(\mathbf{y} \mid \mathbf{a})\ \forall \mathbf{a} \neq \mathbf{x}$).

Unter Verwendung der Gleichungen (4.17) und (4.19) und Anwendung der *Union Bound* erhalten wir folgende Wahrscheinlichkeit für eine Fehlentscheidung

$$\begin{aligned} P_w &= P\left(\ln p(\mathbf{y}|\mathbf{x}) < \ln p(\mathbf{y}|\mathbf{a})\right) = P\left(\sum_{\ell=0}^{N-1} \gamma(\mathbf{y}(\ell) \mid \mathbf{x}(\ell)) < \sum_{\ell=0}^{N-1} \gamma(\mathbf{y}(\ell) \mid \mathbf{a}(\ell))\right) \\ &= P\left(\sum_{\ell=0}^{N-1} \mathbf{y}(\ell)\mathbf{x}(\ell) < \sum_{\ell=0}^{N-1} \mathbf{y}(\ell)\mathbf{a}(\ell)\right) = P\left(\sum_{\ell=0}^{N-1}\sum_{i=1}^{n} y_i(\ell)x_i(\ell) < \sum_{\ell=0}^{N-1}\sum_{i=1}^{n} y_i(\ell)a_i(\ell)\right) \\ &= P\Bigg(\sum_{\ell=0}^{N-1}\sum_{i=1}^{n} \underbrace{2\left(a_i(\ell) - x_i(\ell)\right)}_{=\begin{cases} 4\sqrt{E_s} & \text{für } a_i(\ell) \neq x_i(\ell) \\ 0 & \text{sonst} \end{cases}} \cdot y_i(\ell) > 0\Bigg) \end{aligned} \tag{4.29}$$

Unter der Annahme, dass die Nullsequenz gesendet wurde, nehmen die Codesymbole nur den Wert $x_i(\ell) \equiv -\sqrt{E_s}$ an. Außerdem sollen sich $\mathbf{a}$ und $\mathbf{x}$ in genau d Stellen unterscheiden, d.h. die Hamming-Distanz beträgt $d_H(\mathbf{a},\mathbf{x}) = w_H(\mathbf{a}) = d$. Dann gibt es in Gl. (4.29) nur d Summanden ungleich Null und die paarweise Fehlerwahrscheinlichkeit P_d für zwei Sequenzen mit der Distanz d lautet

$$P_d = P\Big(\underbrace{\underbrace{\sum_l \sum_i}_{x_i(\ell)\neq a_i(\ell)} y_i(\ell)}_{Y} > 0\Big) \tag{4.30}$$

Die Summe über d Empfangswerte y_i ist eine gaußverteilte Zufallsgröße Y (zentraler Grenzwertsatz) mit

$$\text{Mittelwert: } \bar{Y} = -d \cdot \sqrt{E_s/T_s}$$
$$\text{Varianz: } \sigma_Y^2 = d \cdot N_0/2/T_s\,.$$

Die Wahrscheinlichkeit für ein Verwechseln zweier Sequenzen mit der Hamming-Distanz d zueinander ergibt sich dann zu

$$\Longrightarrow P_d = \frac{1}{2} \cdot \operatorname{erfc}\left(\sqrt{d\frac{E_s}{N_0}}\right) = \frac{1}{2} \cdot \operatorname{erfc}\left(\sqrt{dR_c\frac{E_b}{N_0}}\right)\,. \tag{4.31}$$

Abschätzung der Sequenzfehlerwahrscheinlichkeit
Zur Berechnung der Auftrittswahrscheinlichkeit für einen Decodierfehler sind jetzt wie bei den Blockcodes alle Sequenzen zu betrachten. Hierzu kann das schon bekannte Distanzspektrum herangezogen werden, d.h. speziell den Koeffizienten a_d aus Gl. (4.27). Wir erhalten den Ausdruck

$$\boxed{P_w \leq \sum_d a_d \cdot P_d = \frac{1}{2} \cdot \sum_d a_d \cdot \operatorname{erfc}\left(\sqrt{dR_c\frac{E_b}{N_0}}\right)}\,. \tag{4.32}$$

Abschätzung der Bitfehlerwahrscheinlichkeit
In der Praxis ist häufig auch die Bitfehlerrate P_b von Interesse. Zur Bestimmung von P_b ist jedoch noch die Anzahl der fehlerhaften uncodierten Informationsbit erforderlich. Die Annahme der gesendeten Nullsequenz erlaubt nun die ausschließliche Berücksichtigung der Einsen in der Informationssequenz. Diese Information steckt im Exponenten von W des Distanzspektrums und somit im Koeffizienten c_d. Wir erhalten den Ausdruck

$$\boxed{P_b \leq \frac{1}{2} \cdot \sum_d c_d \cdot \operatorname{erfc}\left(\sqrt{dR_c\frac{E_b}{N_0}}\right)}\,. \tag{4.33}$$

Die analytische Abschätzung der Bitfehlerrate hängt von den Distanzeigenschaften des jeweiligen Codes (Koeffizienten c_d) und den Kanaleigenschaften (Fehlerwahrscheinlichkeit P_d) ab. Für andere Kanäle als den AWGN-Kanal, z.B. BSC oder Schwundkanäle, sind in Gl. (4.33) nur die P_d neu zu berechnen, ansonsten behält Gl. (4.33) ihre Gültigkeit.

Beispiel: 1/2-ratiger Faltungscode mit $L_c = 3$, $g_1 = 7_8$, $g_2 = 5_8$

$$\begin{aligned}
T(W,D,L) &= \sum_{d=5}^{\infty} W^{d-4} \cdot D^d \cdot L^{d-2} \cdot (1+L)^{d-5} \\
T(W=1,D,L=1) &= \sum_{d=5}^{\infty} 2^{d-5} D^d \qquad \Rightarrow \quad \boxed{a_d = 2^{d-5}} \\
\left.\frac{\partial T(W,D,L=1)}{\partial W}\right|_{W=1} &= \sum_{d=5}^{\infty} 2^{d-5} \cdot (d-4) \cdot D^d \quad \Rightarrow \quad \boxed{c_d = (d-4) \cdot 2^{d-5}}
\end{aligned}$$

$$P_b \le \frac{1}{2} \cdot \sum_{d=5}^{\infty} (d-4) \cdot 2^{d-5} \cdot \operatorname{erfc}\left(\sqrt{dR_c \frac{E_b}{N_0}}\right) \tag{4.34}$$

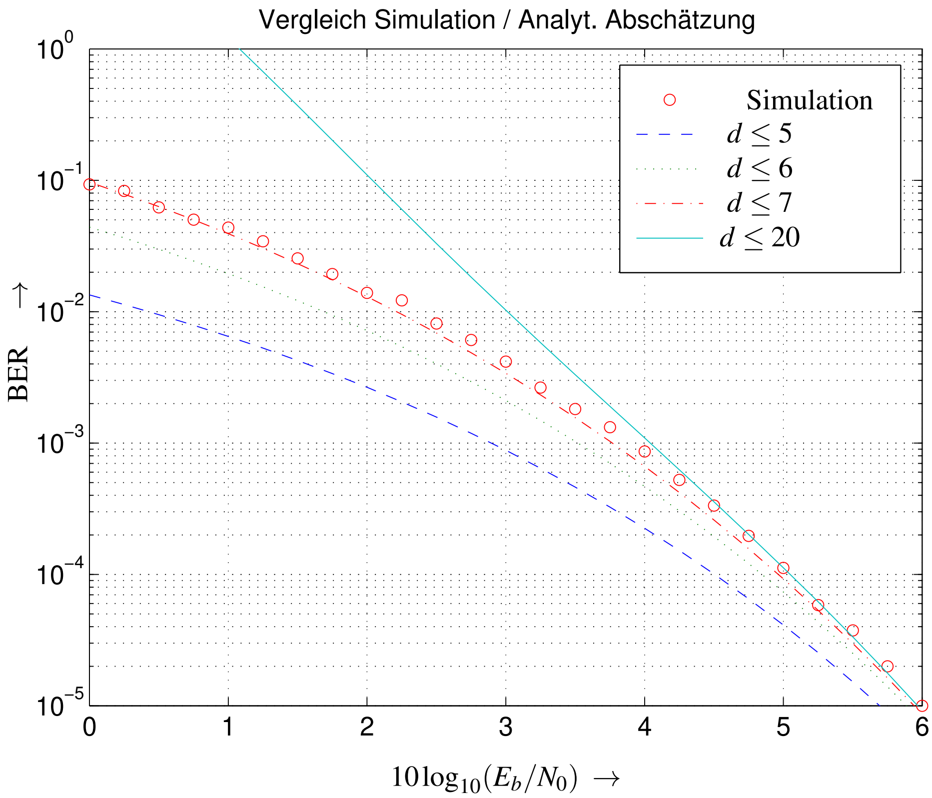


Bild 4.14: Vergleich von Simulationsergebnissen mit der analytischen Abschätzung der Bitfehlerrate für Faltungscode mit $g_1 = 5_8$, $g_2 = 7_8$ und Übertragung über einen AWGN-Kanal

- Asymptotische Bitfehlerrate ($E_b/N_0 \rightarrow \infty$) allein durch freie Distanz d_f bestimmt
- Für Bitfehlerrate im 'mittleren' Bereich gesamtes Distanzspektrum erforderlich
- Für große Fehlerraten /kleine Signal-Rausch-Abstände ist *Union Bound* sehr ungenau

Zum Codiergewinn:
Die Beurteilung eines Kanalcodierungsverfahrens erfolgt in der Regel über den Codiergewinn. Dieser wird

noch einmal in den Bildern 4.15 veranschaulicht. Im linken Diagramm sind die Bitfehlerkurven für den uncodierten Fall und den im Beispiel behandelten Faltungscode (vor und nach der Decodierung) angegeben. Durch die Kanalcodierung mit der Coderate $R_c = 1/2$ wird die Energie E_b eines Informationsbit auf 2 Codebit verteilt, d.h. bei konstantem E_b ergibt sich ein E_b/N_0-Verlust von 3 dB auf dem Kanal (vor der Decodierung). Der Code muss nun diesen Verlust mehr als kompensieren, damit nach der Decodierung noch ein Gewinn übrig bleibt. In unserem Beispiel beträgt dieser ca. 3.5 dB bei einer Fehlerrate von $P_b = 10^{-5}$.

Das rechte Diagramm zeigt die Verhältnisse in Abhängigkeit von E_s/N_0, d.h. die Energie pro Codebit ist konstant. Dann beträgt der Gewinn ca. 6.5 dB, allerdings wird hier nicht der höhere Bandbreitenbedarf durch die Codierung berücksichtigt. In der Literatur ist die erste Darstellung die gebräuchliche.

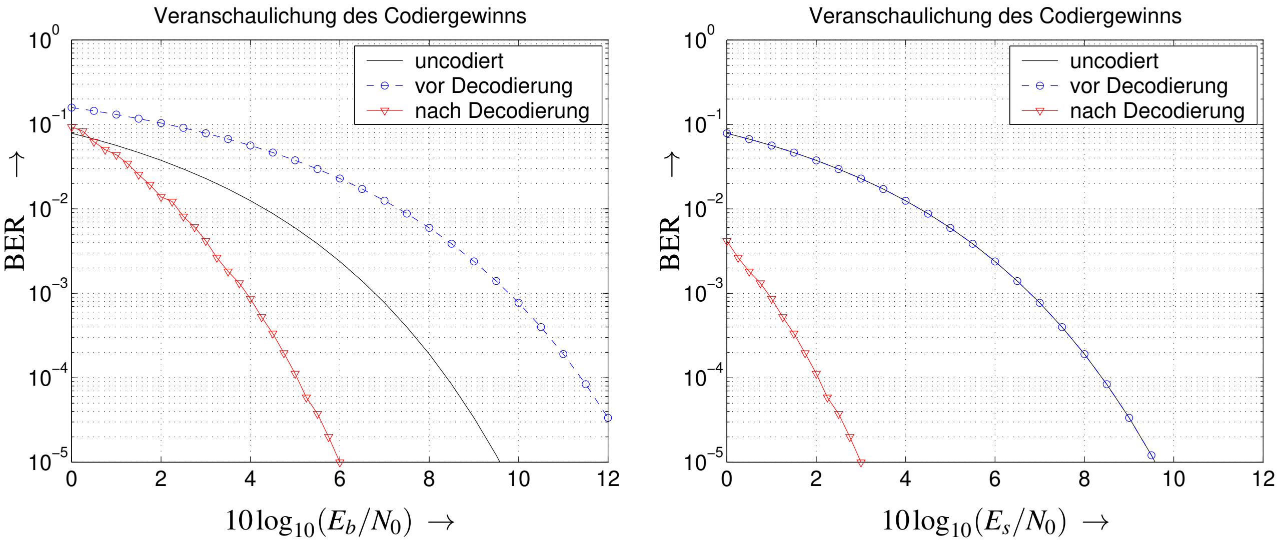


Bild 4.15: Veranschaulichung des Codiergewinns für Faltungscode mit $g_1 = 5_8$, $g_2 = 7_8$ beim AWGN-Kanal

4.7 Beispiele für die Leistungsfähigkeit von Faltungscodes

Einfluss der Quantisierung

Zunächst soll noch einmal der Einfluss der Quantisierung am Eingang eines Empfängers untersucht werden. Durch die Quantisierung vor der Decodierung geht Information unwiderruflich verloren, was im Fall einer *Hard-Decision* in Bild 4.16 deutlich zu sehen ist. Eine 3-Bit-Quantisierung (8-stufig) weist dagegen nur noch kleine Verluste gegenüber dem unquantisierten Fall auf.

Einfluss der Einflusslänge L_c

Nun soll der Einfluss der *Constraint Length* L_c näher erläutert werden. Wie schon mehrfach erwähnt wurde, nimmt die Leistungsfähigkeit von Faltungscodes mit steigender Einflusslänge zu (s. Bild 4.17). Allerdings steigt mit wachsendem L_c auch der Decodieraufwand exponentiell an, da die Anzahl der Zustände im Trellisdiagramm von der Speicherlänge des Codierers abhängt. Es ist also ein Kompromiß zwischen hohem Codiergewinn und praktikablen Decodieraufwand zu finden.

Die analytische Abschätzung mit Hilfe der *Union Bound* ist unabhängig von L_c für geringe Signal-Rausch-Abstände sehr ungenau, bei steigendem Signal-Rausch-Abstand nimmt die Genauigkeit zu. Außerdem reicht es bei kleinen Bitfehlerraten aus, nur wenige Distanzen des Spektrums zu berücksichtigen. Während also für hohe Signal-Rausch-Abstände die Monte-Carlo-Simulationen sehr aufwendig sind und sich somit eine analytische Abschätzung der Bitfehlerrate anbietet, sind Monte-Carlo-Simulationen für niedrige Signal-Rausch-Abstände zu bevorzugen.

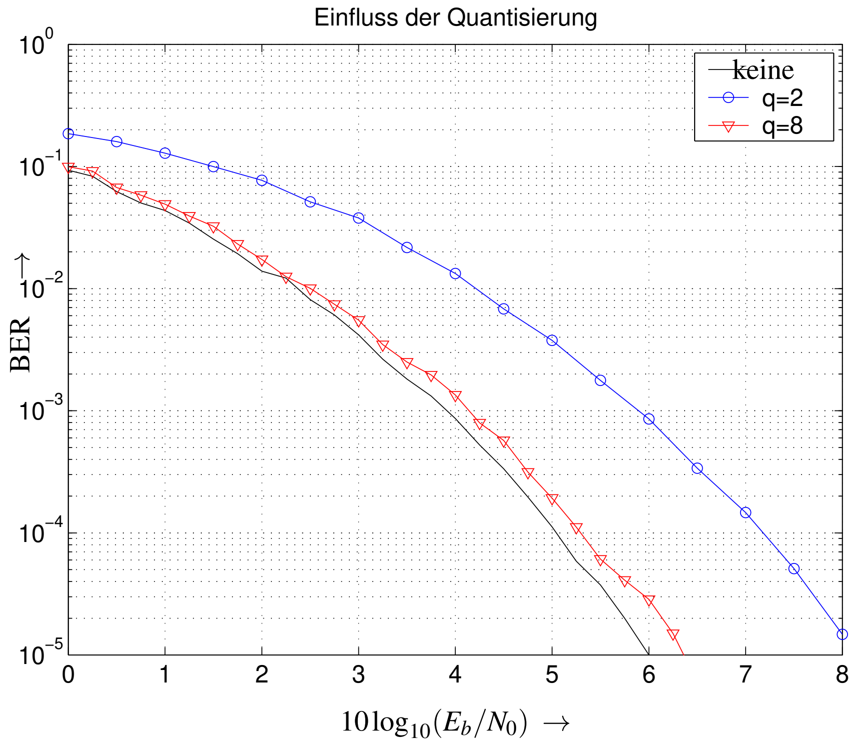


Bild 4.16: Bitfehlerraten für verschiedene Quantisierungsstufen für Faltungscode mit $g_1 = 5_8$, $g_2 = 7_8$ und Übertragung über einen AWGN-Kanal

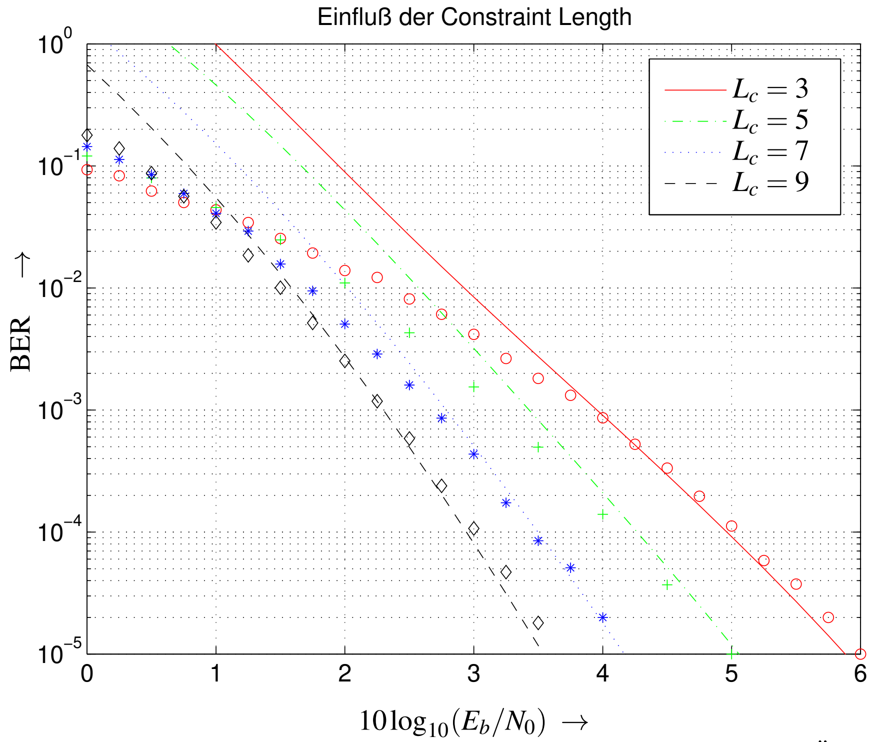


Bild 4.17: Bitfehlerraten für verschiedene Einflusslängen für 1/2-ratige Faltungscode bei Übertragung über einen AWGN-Kanal

Einfluss der Coderate

Zum Abschluss soll noch die Coderate R_c betrachtet werden. Mit sinkender Coderate (mehr Redundanz, höhere Bandbreite) nimmt die Leistungsfähigkeit von Faltungscodes zu. In Bild 4.18 wird dabei größere Bandbreite durch die Abzissenskalierung auf E_b/N_0 berücksichtigt. Codes der Rate $R_c > 1/2$ wurden durch Punktierung des halbratigen Codes konstruiert.

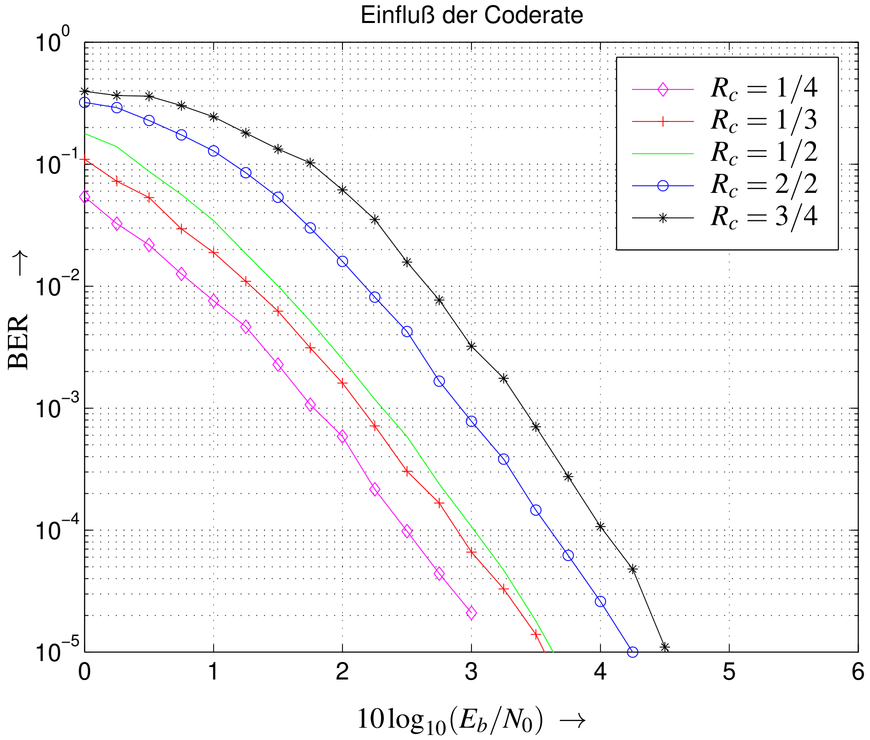


Bild 4.18: Bitfehlerraten für verschiedene Coderaten für Faltungscode mit Einflusslänge $L_c = 9$ und Übertragung über einen AWGN-Kanal

Literaturverzeichnis

[Bla83] R.E. Blahut. *Theory and Practice of Error Control Codes*. Addison-Wesley, 1983.

[Bla98] R.E. Blahut. *Algebraic Codes for Data Transmission*. Cambridge University Press, 1998.

[Bos98] M. Bossert. *Kanalcodierung*. Teubner, Stuttgart, 1998.

[Fri96] B. Friedrichs. *Kanalcodierung*. Springer Verlag, Berlin, 1996.

[Joh92] R. Johannesson. *Informationstheorie – Grundlage der (Tele-)Kommunikation*. Addison-Wesley, Lund, Schweden, 1992.

[Kam04] K. D. Kammeyer. *Nachrichtenübertragung*. Teubner, Stuttgart, 3. Auflage, 2004.

[KK01] K.-D. Kammeyer und V. Kühn. *MATLAB in der Nachrichtentechnik*. Schlembach, Weil der Stadt, 2001.

[Küh06] V. Kühn. *Wireless Communications over MIMO Channels: Application to CDMA and Multiple Antenna Systems*. Wiley, West Sussex, U.K., 2006.

[Pro01] J.G. Proakis. *Digital Communications*. McGraw-Hill, 4. Auflage, 2001.