Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung*

Norbert Stolte

D17
Darmstädter Dissertation
Januar 2002

*This document is also available in English. Please feel free to contact the author.

Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung

Vom Fachbereich
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des Grades
Doktor-Ingenieur genehmigte

Dissertation

von

Dipl.-Ing. Norbert Stolte
Groß-Umstadt

Referent:	Prof. Dr.-Ing. Bernhard Dorsch
Korreferent:	Prof. Dr.-Ing. Martin Bossert
Eingereicht am:	26. Oktober 2001
Disputation am:	9. Januar 2002

D17
Darmstädter Dissertation
Januar 2002

Kostenlos zu beziehen als PDF-File über
EPDA (Elektronische Publikationen der TU Darmstadt)
Fachbereich Elektrotechnik und Informationstechnik
http://elib.tu-darmstadt.de
oder
OPAC-Recherche bei der Deutschen Bibliothek Frankfurt am Main
zur Zeit: http://dbf-opac.ddb.de

This document is also available in English.
Please feel free to contact the author at norbert.stolte@gmx.de

Kurzfassung

In dieser Arbeit werden Codes betrachtet, die allein durch rekursive Anwendung der $|u|u+v|$-Konstruktion, auch als PLOTKIN-Konstruktion bezeichnet, generiert werden. Der Schwerpunkt liegt hier sowohl auf der Konstruktion als auch auf der Decodierung von Codes dieser Klasse, die die Klasse der binären REED-MULLER (RM) Codes enthält. Bezüglich der auftretenden Störungen werden nur die beiden Sonderfälle des binären symmetrischen Kanals (BSC) und des additiven weißen gaußschen Rauschkanals (AWGN) betrachtet.

Alle hier vorgestellten Codes sind Untercodes von RM-Codes und lassen sich als verallgemeinert verkettete Codes beschreiben. Für die zur Decodierung notwendige Zuverlässigkeitsübergabe an die Decoder der äußeren Codes wird neben der optimalen auch eine suboptimale Methode beschrieben und bewertet. Aufbauend auf diesen Methoden wird sowohl ein neues sequentielles als auch ein listengestütztes Decodierverfahren vorgeschlagen, die für alle Codes dieser Klasse geeignet sind. Es können damit beim AWGN-Kanal mit geringem Aufwand alle RM-Codes und deren Untercodes bis zu einer Länge von $N = 128$ Codesymbolen annähernd optimal decodiert werden. Speziell für RM-Codes wird darüber hinaus auch eine kombinierte Listen- und Permutationsdecodierung vorgeschlagen, womit beim AWGN-Kanal auch für alle Codes der Länge $N = 256$ und beim BSC bis zur Länge $N = 512$ nahezu optimale Wortfehlerwahrscheinlichkeiten erzielt werden.

Um auch bei größeren Codelängen ebenfalls gute Decodierergebnisse zu erzielen, werden zwei verschiedene Methoden zur Anpassung des Codes an die verwendeten Decoder vorgestellt. Beide Methoden ermöglichen für Codelängen $N = 2^m$ die Konstruktion von Codes beliebiger Raten K/N, $K \in \{1, 2, \ldots, N\}$. Unter anderem die Berücksichtigung der bei Multilevel-Codes bekannten Ergebnisse führt so zu Codes, die zusammen mit dem hier vorgestellten Listendecodierverfahren auch bei deutlich über der Cutoff-Rate liegenden Coderaten kleine Fehlerwahrscheinlichkeiten ermöglichen.

Abstract

In this thesis codes are considered which are exclusively generated by the $|u|u+v|$-construction, also known as PLOTKIN-construction. It is focused both on the construction and the decoding of codes of this class. This class also contains the binary REED-MULLER (RM) codes. Concerning the channel distortion two special cases are considered, the additive white GAUSSIAN noise (AWGN) channel and the binary symmetric channel (BSC).

All codes presented are subcodes of RM codes and can be described as generalized concatenated codes. In general reliability values are required for the decoding of the outer codes. An optimal and a suboptimal calculation method of these values are given and evaluated. Based on these methods a new sequential and a list-decoding algorithm are proposed which are applicable to all codes of this class. This enables for the AWGN channel close to optimal decoding of all RM codes and their subcodes of length up to $N = 128$ code symbols.

Furthermore, especially for the RM codes a combined list and permutation decoding algorithm is proposed. With this algorithm close to optimal word error probabilities are achieved for

the AWGN channel and the BSC for all RM codes of length up to $N = 256$ and $N = 512$, respectively.

In order to obtain good decoding results also for larger code lengths two different methods are presented to match the code to the decoder. Both methods enable the construction of codes of length $N = 2^m$ with arbitrary rates K/N, $K \in \{1, 2, \ldots, N\}$. Taking into account results known from multilevel codes leads to codes which together with the proposed list-decoding algorithm give small error rates, even if the code rate is well above the cutoff-rate.

The complete document is also available in English. Please feel free to contact the author.

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter in der Arbeitsgruppe von Prof. Dr.-Ing. Bernhard Dorsch am Institut für Nachrichtentechnik der Technischen Universität Darmstadt, sowie während meiner Tätigkeit als Doktorand in der Arbeitsgruppe von Dr. Robert Schweikert am Institut für Nachrichtentechnik des Deutschen Zentrums für Luft- und Raumfahrt (DLR) in Oberpfaffenhofen.

Wissenschaftliche Freiheit, jenseits von Vorgaben und einzwängenden Randbedingungen, war im Nachhinein betrachtet notwendig für das Gelingen dieser Arbeit. Da ich sie in der Arbeitsgruppe von Herrn Prof. Dorsch vorgefand, gebührt ihm mein besonderer Dank. Auch für die positive Atmosphäre in der Gruppe von Herrn Dr. Schweikert möchte ich mich an dieser Stelle bedanken.

Sehr gefreut hat mich auch die Bereitschaft von Herrn Prof. Dr.-Ing. Martin Bossert für die Übernahme des Korreferats.

Ganz entscheidenden Anteil an dieser Arbeit hat unzweifelhaft Herr Dr.-Ing. Ulrich Sorger, ohne dessen Anregungen und Impulse wesentliche Teile der Arbeit, wenn nicht sogar die Arbeit an sich nicht möglich gewesen wäre. Auch Herr Prof. Dr. Eugene Krouk von der State University of Aerospace Instrumentation in St. Petersburg hat durch seine Vorträge und Einzelgespräche viel zum Gelingen dieser Arbeit beigetragen. Herr Dr.-Ing. Stefan Brück hat nicht nur die leidige Aufgabe des Korrekturlesens dieser Arbeit übernommen. Die Diskussionen mit ihm waren gerade in der Anfangsphase meiner Tätigkeit besonders wichtig.

Weiterhin gilt mein Dank allen Kollegen von Darmstadt und Oberpfaffenhofen für das positive Umfeld, das mich auf die letzten fünfeinhalb Jahre auch in menschlicher Hinsicht positiv zurückblicken läßt.

Abschließend und ganz besonders bedanken möchte ich mich bei meiner Frau Yuko, die durch ihre Unterstützung sicherlich ihren Teil zu dieser Arbeit beigetragen hat.

Groß-Umstadt, im Januar 2002 *Norbert Stolte*

Inhaltsverzeichnis

1. Einleitung

Digitale Daten, gleich welcher Art, sind bei ihrer Übermittlung bzw. Speicherung in vielen Fällen unbekannten, zufälligen Störungen ausgesetzt. Man muß daher generell davon ausgehen, daß sie anschließend nicht mehr fehlerfrei zur Verfügung stehen. Daher werden in der Praxis oft fehlerkorrigierende Codes verwendet, bei denen zusätzlich zu den Nutzdaten redundante Prüfdaten mit übertragen bzw. gespeichert werden, mit deren Hilfe dann Fehler erkannt und wenn möglich auch korrigiert werden können. Falls die Verarbeitung der Daten in blockorientierter Weise erfolgt bedeutet dies, daß zu jedem Datenblock ein dazugehöriger Prüfblock berechnet wird. Man spricht dann auch von Blockcodes.

Es erscheint einleuchtend, daß zum einen die Sicherheit der Daten mit zunehmender Anzahl der Prüfdaten größer wird, aber auch daß so keine hundertprozentige Sicherheit gegenüber Fehlern gewährleistet werden kann. In seiner revolutionären Arbeit hat CLAUDE E. SHANNON 1948 jedoch gezeigt, daß theoretisch eine beliebig kleine Fehlerwahrscheinlichkeit erreicht werden kann, wenn die Blocklänge genügend groß ist, sofern nur das Verhältnis zwischen Nutzdatenblocklänge zu Gesamtblocklänge einen bestimmten oberen Grenzwert nicht überschreitet. Dieses Verhältnis wird im allgemeinen als (Nutzdaten-) Rate und die obere Grenze als *Kanalkapazität* bezeichnet, bezugnehmend auf die abstrakte nachrichtentechnische Betrachtung, daß die Daten über einen "Kanal" übertragen werden. Ist nahezu fehlerfreie Datenrückgewinnung bei Nutzdatenraten, die wesentlich kleiner als die obere Grenze sind, noch einfach zu realisieren, so steigt der dazu benötigte Aufwand für Raten nahe der Kanalkapazität stark an. Als eine "Grenze" zwischen *mit akzeptablem Aufwand erreichbaren Coderaten* und *nur mit sehr großem Aufwand erreichbaren Coderaten* galt noch bis Anfang der 90er Jahre die Cutoff-Rate (deut.: Grenz-Rate), die im allgemeinen deutlich unter der Kanalkapazität liegt.

Die Methoden, mit denen *codiert*, d.h. die Prüfdaten berechnet werden können, sind vielfältig. Neben algebraischen Methoden, bei denen aus den Nutzdaten anhand von Polynomen über endlichen Zahlenkörpern die Prüfdaten berechnet werden, existieren auch sehr simple Verfahren, die auf einfachen Verknüpfungen der Datensymbole beruhen. Zudem hat es sich bei großer Blocklänge als sinnvoll erwiesen, die Codierung in zwei oder mehreren Schritten durchzuführen, indem erst kurze Codeblöcke mit jeweils eigenen Prüfdaten berechnet werden, um diese dann im zweiten Schritt miteinander zu verkoppeln.

Die Frage, welche Methoden besser sind, kann im allgemeinen nicht beantwortet werden, da neben der speziellen Anwendung auch der nötige Rechenaufwand zur Decodierung, d.h. Datenrückgewinnung betrachtet werden muß. Zwar bieten komplexe Codierverfahren im allgemeinen das größere Potential an Datensicherheit, aber wenn die Nutzung dieses Potentials unrealisierbar großen Aufwand bei der Decodierung voraussetzt, können simple Verfahren mit realisierbaren Algorithmen zur Nutzung des geringeren Potentials insgesamt bessere Ergebnisse liefern.

Ein Beispiel hierfür sind die sog. *Turbo*-Codes, die '93 zum ersten Mal vorgestellt wurden [3] und die durch eine (meist parallele) Verkopplung einfacher Faltungscodes generiert werden. Turbo-Codes großer Länge lassen sich sehr effizient decodieren, so daß damit auch bei Raten nahe der Kanalkapazität (und somit größer als die Cutoff-Rate) eine annähernd fehlerfreie Datenrückgewinnung möglich ist. Die wesentlich neue Idee, die mit den Turbo-Codes eingeführt wurde, ist das Prinzip der *iterativen Decodierung*. Basierte die klassische Decodierung von verkoppelten Codes noch auf schrittweise getroffenen Entscheidungen für die Teilcodes, wird bei der iterativen Decodierung nur mit Wahrscheinlichkeiten gearbeitet, die in mehreren Durchläufen zwischen den einzelnen Decodern der Teilcodes ausgetauscht werden. Leider erfordert die iterative Decodierung relativ große Codelängen, so daß Codes mit einer Länge von weniger als 500 Codesymbolen nur schlecht mit diesem Verfahren decodiert werden können. Für den AWGN-Kanal[1] und Codelängen > 2000 Symbole sind jedoch nach Kenntnis des Autors die besten Ergebnisse alle durch iterative Decodierung erzielt worden.

In dieser Arbeit werden binäre Blockcodes betrachtet, bei denen die Prüfdaten ausschließlich durch die $|u|u+v|$-Konstruktion (auch als PLOTKIN-Konstruktion bekannt), d.h. durch die beiden simplen Operationen *Wiederholung* und *Exor*[2]*-Verknüpfung* von Daten erzeugt werden. Binär bedeutet, daß der gesamte Block aus Nutzdaten und Prüfdaten nur zwei verschiedene Symbole, z.B. Nullen und Einsen, enthält. Trotz der simplen Operationen können damit durch rekursive Konstruktion Codes mit relativ großem Korrekturpotential konstruiert werden, wie z.B. die REED-MULLER (RM) Codes, die für kurze Blocklängen mit zu den besten Codes gehören. Es existieren auch sehr einfache Decodieralgorithmen mit geringer Komplexität, wie z.B. der rekursive Algorithmus von KABATYANSKY [29], die aber leider nur für kurze Blocklängen (≤ 32 Bits) bzw. kleine Coderaten das zur Verfügung stehende Potential annähernd nutzen. Für längere Blocklängen nutzen diese Decodierverfahren nur einen Bruchteil des tatsächlichen Korrekturpotentials der RM-Codes aus. Dies führte in einem bekannten Lehrbuch zur Kanalcodierung aus dem Jahr 1983 zu der Bemerkung:

"A Reed-Muller code was used to transmit the Mariner phototgraph of Mars in 1972. Today, a more powerful code would be preferred." [4, Chapter 3]

Zwar wurden auch schon Versuche unternommen, die zusammen mit den Turbo-Codes vorgestellte iterative Decodierung auch auf die Codeklasse der RM-Codes anzuwenden [39], [34], die damit erzielten Verbesserungen waren aber nur gering. Eine von LUCAS, BOSSERT und DAMMANN [33] vorgestellte Erweiterung des rekursiven Algorithmus von KABATYANSKY zu einem Listendecodierverfahren ermöglicht eine nahezu optimale Decodierung von RM-Codes bis zu einer Länge von 64 Codesymbolen und ergibt zudem für den Code der Länge 512 und Rate 0.5 fast gleiche Ergebnisse wie die iterative Decodierung. Sowohl die iterative Decodierung als auch die Listendecodierung nach LUCAS et al. sind jedoch beim letzten Code noch weit von einer vollständigen Nutzung des Codepotentials entfernt.

Wesentlich bessere Ergebnisse konnten vom Autor zusammen mit SORGER mit einem modifizierten Listen-Decodierverfahren erreicht werden [57], [56], so daß alle RM-Codes bis zu einer Länge von 128 Symbolen nahezu optimal decodiert werden können. Bei noch größeren Codelängen und Raten wesentlich von Null verschieden ist jedoch auch mit diesem Algorithmus der notwendige Aufwand für eine nahezu optimale Decodierung von RM-Codes sehr groß, so

[1] Dieser Kanal wird in Kapitel 2 näher beschrieben.
[2] Die Rechenregeln der Exor-Operation ($\oplus$-Operation) lauten: $0+0=1+1=0$ und $0+1=1+0=1$.

daß hier nach anderen Wegen gesucht werden muß. Wie im Verlauf der Arbeit gezeigt wird, ermöglicht eine kombinierte Permutations- und Listendecodierung des RM-Codes mit Länge 512 und Rate $1/2$ zumindest für die Übertragung über den BSC[3] eine fast optimale Ausnutzung des Korrekturpotentials.

Die Listendecodierung von $|u|u+v|$-verketteten Codes ist generell ein wesentlicher Punkt dieser Arbeit. Die Fokussierung auf dieses Verfahren rührt aus den hier gewonnenen Erkenntnissen, daß es zum einen für diese Codeklasse als sehr geeignet erscheint und zum anderen, daß die Cutoff-Rate des Übertragungskanals für dieses Verfahren, wie sich zeigen wird, keine beschränkende Größe ist. Es können im Gegenteil auch Codes, deren Raten deutlich über der Cutoff-Rate liegen, noch mit relativ kleiner Liste fast fehlerfrei decodiert werden. Allerdings ist es dazu mit wachsender Codelänge notwendig, eine gemeinsame Betrachtung von Codierung und Decodierung vorzunehmen. Dies entspricht hier einer Anpassung des Codes an das Listendecodierverfahren. Erste Überlegungen dazu, die schon von DUMER und SHABUNOV [11], [12] vorgestellt wurden, werden in dieser Arbeit verallgemeinert. Eine der wesentlichen Erweiterungen resultiert aus der Betrachtung der $|u|u+v|$-Konstruktion als Multilevel-Code, womit sich die z.B. von WACHSMANN [63] vorgeschlagenen Konstruktionsprinzipien hier anwenden lassen.

Die Arbeit ist wie folgt gegliedert:

- In Kapitel 2 werden allgemein bekannte **Grundlagen** der Kanalcodierung erläutert.
- In Kapitel 3 wird zu Beginn die PLOTKIN-Konstruktion vorgestellt. Es folgt die Definition zweier verschiedener Möglichkeit, die **Zuverlässigkeitswerte** bei rekursiver Decodierung an die äußeren Decoder zu übergeben sowie ein Vergleich beider Möglichkeiten. Der Begriff des *äquivalenten Signal-zu-Rauschverhältnisses* wird eingeführt und zum Abschluß die rekursive Codekonstruktion anhand des klassisches Kriteriums, nach dem die Mindestdistanz des Codes zu maximieren ist, vorgestellt.
- In Kapitel 4 werden notwendige Eigenschaften der Klasse der **REED-MULLER Codes** gegeben.
- In Kapitel 5 wird die **bitweise Mehrstufendecodierung** erläutert und gezeigt, wie die Wortfehlerwahrscheinlichkeit für dieses Decodierverfahren mit Hilfe des äquivalenten Signal-zu-Rauschverhältnisses sehr exakt geschätzt werden kann. Es folgt die Beschreibung der sequentiellen bzw. **Listendecodierung** von $|u|u+v|$-verketteten Codes. Nach einer Erklärung der generellen Problematik bei Listendecodierung von langen REED-MULLER Codes wird hier zur Lösung die kombinierte Permutations- und Listendecodierung vorgeschlagen.
- Der Schwerpunkt des 6. Kapitels liegt auf der Konstruktion von langen Codes, die sich noch mit vertretbarem Aufwand fast optimal decodieren lassen. Es werden hierzu zwei verschiedene **Konstruktionskriterien** angegeben.
- Zum Abschluß der Arbeit folgen im 7. Kapitel noch einige **Simulationsergebnisse** für Codes verschiedener Längen.

[3] Auch dieser Kanal wird in Kapitel 2 näher beschrieben.

2. Grundlagen der Kanalcodierung

In diesem Kapitel werden die Grundlagen der Kanalcodierung erläutert und notwendigen Definitionen für die späteren Kapitel gegeben. Da alle in diesem Kapitel vorgestellten Prinzipien und Eigenschaften in der Standardliteratur zu finden sind, ist die Darstellung auf das Notwendigste beschränkt, auf Beweise wird verzichtet.

Zu dem Zweck, digitale Daten gegen auftretende Fehler zu schützen, wird anstelle des ursprünglichen Datenblocks der Länge K ein um Prüfinformation erweiterter Block der Länge $N \geq K$ übertragen bzw. gespeichert. Die Zuordnung von Vektoren der Länge K zu Vektoren der Länge N bezeichnet man als Codierung, wobei zu jedem Datenvektor $\boldsymbol{i} = (i_1, i_2, \ldots, i_K)$ im Sinne der Eindeutigkeit *ein* Codevektor $\boldsymbol{c} = (c_1, c_2, \ldots, c_N)$ gehört. Für binäre Daten- und Codesymbole, d.h. $i_k, c_k \in \{+1, -1\}$ ergibt sich damit, daß aus der Menge aller möglichen 2^N Vektoren nur 2^K gültige Codevektoren ausgewählt werden. Die Menge aller 2^K Codevektoren bezeichnet man als Code C. Die Nutzdaten- bzw. Coderate ergibt sich damit zu $R = K/N$. Bei linearen Codes kann die Codierung mit Hilfe einer $K \times N$ Matrix $\boldsymbol{G}$ (auch Generatormatrix genannt) erfolgen, die die lineare Abbildung des Raumes mit Dimension K in den linearen Unterraum (Coderaum) der gleichen Dimension innerhalb des Raumes mit Dimension N definiert. Damit gilt

$$\boldsymbol{i} \cdot \boldsymbol{G} = \boldsymbol{c}.$$

Die Rechnung erfolgt hier im Galois Feld GF(2) mit $+1$ als dem neutralen Element bezüglich der Addition[1].

Der durch die Übertragung empfangene bzw. gelesene Vektor $\boldsymbol{y} = (y_1, y_2, \ldots, y_N)$ steht dann dem Decoder zur Verfügung. Abhängig von der praktischen Realisierung können die Symbole z.B. zweiwertig, d.h. $y_k \in \{\pm 1\}$, mehrwertig oder auch reellwertig, d.h. $y_k \in \mathbb{R}$ sein. Der Decoder hat dann die Aufgabe zu lösen, aus dem möglicherweise verfälschten Empfangsvektor $\boldsymbol{y}$ die ursprünglichen Daten $\boldsymbol{i}$ oder äquivalent den ursprünglichen Codevektor $\boldsymbol{c} \in C$ zu schätzen. Er benötigt dazu Kenntnis über die statistischen Eigenschaften der Störung, die durch die Übergangswahrscheinlichkeiten $P(\boldsymbol{y}|\boldsymbol{c})$ für diskrete y_i bzw. durch die Wahrscheinlichkeitsdichten $f(\boldsymbol{y}|\boldsymbol{c})$ bei reellwertigen Empfangssymbolen gegeben ist. In dieser Arbeit werden zwei

[1] In dieser Arbeit wird von der üblichen Darstellung der Elemente in GF(2) als Null und Eins abgewichen und stattdessen die Menge $\{\pm 1\}$ mit Eins als dem neutralem Element der Addition verwendet. Dies ermöglicht im weiteren Verlauf eine kompakte und einheitliche Beschreibung von AWGN-Kanal und BSC. Die Tabellen für die Addition und die Multiplikation sind daher wie folgt:

$+$	$+1$	-1
$+1$	$+1$	-1
-1	-1	$+1$

$*$	$+1$	-1
$+1$	$+1$	$+1$
-1	$+1$	-1

theoretische Sonderfälle von Störungen betrachtet, die sich beide dadurch auszeichnen, daß die Störung von Symbol zu Symbol statistisch unabhängig ist. Solche Kanäle werden als gedächtnislos bezeichnet und es gilt z.B. bei diskreten y_k

$$P(\boldsymbol{y}|\boldsymbol{c}) = \prod_{k=1}^{N} P(y_k|c_k).$$

Dabei können die Übergangswahrscheinlichkeiten von k abhängen. Falls sie nicht von k abhängen, ist damit für zweiwertige Empfangssymbole y_k die Übergangswahrscheinlichkeit $P(\boldsymbol{y}|\boldsymbol{c})$ durch die Symbolfehlerwahrscheinlichkeiten $p_{+1} = P(y_k \neq +1|c_k = +1)$ und $p_{-1} = P(y_k \neq -1|c_k = -1)$ eindeutig beschrieben. Falls gilt $p_{+1} = p_{-1} = p$, spricht man vom binären symmetrischen Kanal (engl. Binary Symmetric Channel, BSC) mit Symbolfehlerwahrscheinlichkeit p. Für reellwertige Empfangssymbole beschränkt sich diese Arbeit auf den additiven weißen gaußschen Rausch-Kanal (engl. Additive White Gaussian Noise Channel, AWGN Channel). Die Übertragung erfolgt hier zeitkontinuierlich, und die Wahrscheinlichkeitsdichten für die zeitdiskreten Werte nach der Abtastung des Signals hinter dem signalangepaßten Filter sind gaußverteilt[2]

$$f(y_k|c_k) = \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left[\frac{-(y_k - c_k)^2}{2\sigma_0^2}\right]. \tag{2.1}$$

Hierbei ist die Varianz σ_0^2 bei Übertragung mit der Signalempfangsleistung S_0 und Symboldauer T_S über einen Kanal mit zweiseitiger Rauschleistungsdichte $N_0/2$ gegeben durch

$$\sigma_0^2 = \frac{N_0}{2 \cdot S_0 \cdot T_\mathrm{S}} = \frac{N_0}{2 \cdot E_\mathrm{S}}.$$

Das Signal-zu-Rauschverhältnis (SNR) S/N zum Abtastzeitpunkt ergibt sich mit der Signalenergie pro Symbol E_S zu

$$\frac{S}{N} = \frac{1}{\sigma_0^2} = \frac{2E_\mathrm{S}}{N_0}.$$

Obwohl in der Realität in vielen Fällen die Störungen korreliert sind, haben die beiden obigen Kanalmodelle ihre Berechtigung, da sie für alle Kanalmodelle mit gleicher Symbolfehlerwahrscheinlichkeit bzw. mittlerer Rauschleistung den schlimmsten Fall darstellen.

2.1. Optimale Decodierregel

Dem Decoder steht das geschriebene bzw. gesendete Codewort $\boldsymbol{c}$ nicht zur Verfügung, er muß aus dem Lese- bzw. Empfangsvektor $\boldsymbol{y}$ die ursprünglichen Daten schätzen. Der optimale Decodieralgorithmus (im Sinne der Fehlerwahrscheinlichkeit) wählt als Decodierentscheidung unter Kenntnis der Übergangswahrscheinlichkeiten und des Vektors $\boldsymbol{y}$ dasjenige Codewort $\widehat{\boldsymbol{c}}$, das die Fehlerwahrscheinlichkeit $P(\widehat{\boldsymbol{c}} \neq \boldsymbol{c}|\boldsymbol{y})$ minimiert. Falls alle Codeworte $\boldsymbol{c}$ mit gleicher Wahrscheinlichkeit gesendet wurden, ist diese optimale Entscheidungsregel äquivalent zu

$$\widehat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c} \in C} P(\boldsymbol{y}|\boldsymbol{c}). \tag{2.2}$$

[2] In der Literatur auch als Normalverteilung $\mathcal{N}(c_k, \sigma_0)$ mit Mittelwert c_k und Standardabweichung σ_0 bezeichnet.

Ein solcher Decoder wird auch als Maximum-Likelihood-Decoder (ML-Decoder, deut.: Maximaler-Wahrscheinlichkeits-Decoder) bezeichnet und dessen Decodierentscheidung im folgenden mit $\boldsymbol{c}_{\text{ML}}$. Bei Übertragung über den BSC bzw. den AWGN-Kanal ist die Gesamtwahrscheinlichkeit $P(\boldsymbol{y}|\boldsymbol{c})$ durch das Produkt der Einzelwahrscheinlichkeiten gegeben. Für den BSC mit Fehlerwahrscheinlichkeit p gilt dann

$$P(\boldsymbol{y}|\boldsymbol{c}) = p^{\sum_k |y_k - c_k|/2} \cdot (1-p)^{N - \sum_k |y_k - c_k|/2}$$

und für den AWGN-Kanal

$$P(\boldsymbol{y}|\boldsymbol{c}) = \left(\frac{1}{\sqrt{2\pi}\sigma_0}\right)^N \exp\left[\frac{-\sum_k (y_k - c_k)^2}{2\sigma_0^2}\right].$$

Der Maximierung von $P(\boldsymbol{y}|\boldsymbol{c})$ entspricht daher beim BSC mit $p < 0.5$ der Minimierung von $\sum_k |y_k - c_k|/2$ bzw. beim AWGN-Kanal der Minimierung von $\sum_k (y_k - c_k)^2$. Da beim BSC mit $y_k, c_k \in \{+1, -1\}$ gilt

$$|y_k - c_k|/2 = (y_k - c_k)^2/4,$$

kann die Entscheidungsregel des ML-Decoders (2.2) für beide in dieser Arbeit betrachteten Kanäle auch folgendermaßen geschrieben werden

$$\boldsymbol{c}_{\text{ML}} = \arg\min_{\boldsymbol{c} \in C} \sum_{k=1}^{N} (y_k - c_k)^2. \tag{2.3}$$

Gemäß (2.3) ist für die ML-Decodierentscheidung also nur der quadratische euklidische Abstand $d_{\text{E}}^2(\boldsymbol{y}, \boldsymbol{c})$ zwischen dem Empfangsvektor $\boldsymbol{y}$ und den verschiedenen Codewörtern $\boldsymbol{c}$ maßgebend, unabhängig von der Fehlerwahrscheinlichkeit p oder der Varianz σ_0^2.

Vor der Decodierung kann, da die $c_k \in \{+1, -1\}$ zweiwertig sind, das Minimum des euklidischen Abstands über alle Codeworte $d_{\text{E}}(\boldsymbol{y}, \boldsymbol{c}_{\text{ML}})$ nach unten abgeschätzt werden, d.h. mit $a_k = \operatorname{sign} y_k$ gilt

$$d_{\text{E}}^2(\boldsymbol{y}, \boldsymbol{c}_{\text{ML}}) = \min_{\boldsymbol{c} \in C} \sum_{k=1}^{N} (y_k - c_k)^2 \geq \sum_{k=1}^{N} (y_k - a_k)^2 = d_{\text{bias}}^2.$$

Gleichheit gilt, falls die 'harte' Vorzeichenentscheidung von $\boldsymbol{y}$ ein Codewort ergibt. Allgemein gilt, daß jedes $c_k \neq a_k$ zu einer Erhöhung von $d_{\text{E}}^2(\boldsymbol{y}, \boldsymbol{c})$ um den Wert

$$(y_k - c_k)^2 - (y_k - a_k)^2 = 4|y_k| \quad \text{für} \quad c_k \neq a_k$$

führt, so daß man mit $w_k = |y_k|$ für die quadratische euklidische Distanz für alle $\boldsymbol{c}$ schreiben kann

$$d_{\text{E}}^2(\boldsymbol{y}, \boldsymbol{c}) = d_{\text{bias}}^2 + 4 \sum_{\{k: c_k \neq a_k\}} w_k. \tag{2.4}$$

Damit ergibt sich die dritte äquivalente ML-Entscheidungsregel, die für spätere Kapitel benötigt wird:

$$\boldsymbol{c}_{\mathrm{ML}} = \arg\min_{\boldsymbol{c}\in C} \sum_{\{k:c_k\neq a_k\}} w_k. \tag{2.5}$$

Das Finden des wahrscheinlichsten Codewortes $\boldsymbol{c}_{\mathrm{ML}}$ kann auf verschiedene Arten erfolgen. Die direkte Methode, alle Codewörter der Reihe nach mit der Empfangssequenz zu vergleichen und dann das beste zu wählen, kann aus verständlichen Gründen nur bei einer kleinen Anzahl von Codewörtern angewendet werden. Wenn der Code durch ein kompaktes Trellis beschrieben werden kann, dann kann die ML-Decodierung auch anhand dieses Trellis mit Hilfe des VITERBI-Algorithmus erfolgen. Die Komplexität, d.h. die Größe des minimalen Trellis, hängt hier nicht nur allein von Anzahl der Codeworte ab, sondern in besonderer Weise von den Verknüpfungen zwischen Informations- und Prüfsymbolen und nicht zuletzt auch von der Reihenfolge der Symbole. Allgemein sind jedoch nur Codes mit einem relativ geringen Korrekturpotential mit Hilfe des VITERBI-Algorithmus decodierbar, da hohes Korrekturpotential aus einer starken Verflechtung der Codesymbole untereinander und damit aus einem großen Codetrellis resultiert. Damit wird auch die Komplexität des VITERBI-Algorithmus bei langen Codes zu groß und die ML-Decodierung unrealisierbar.

Abhilfe bieten hier Algorithmen, die nicht in jedem Fall eine ML-Decodierung garantieren. Damit wird dann zwar nicht das komplette Korrekturpotential des Codes ausgenutzt, aber die verminderte Decodierkomplexität ermöglicht die Verwendung von besseren Codes, so daß in manchen Fällen insgesamt die Datensicherheit erhöht wird. In dieser Arbeit wird in erster Linie ein sequentieller bzw. Listenalgorithmus verwendet.

2.2. Codeparameter

An dieser Stelle einige notwendige Definitionen, die in fast jedem Lehrwerk zur Kanalcodierung zu finden sind:

Definition 2.1 Die Anzahl von Stellen, an denen sich zwei gleichlange, diskrete Vektoren $\boldsymbol{a}, \boldsymbol{b}$ unterscheiden, bezeichnet man als **Hammingdistanz** $d_{\mathrm{H}}(\boldsymbol{a}, \boldsymbol{b})$. □

Definition 2.2 Die Anzahl von Stellen eines diskreten Vektors $\boldsymbol{a}$, die ungleich Eins[3] sind, bezeichnet man als **Hamminggewicht** $w_{\mathrm{H}}(\boldsymbol{a})$. □

Definition 2.3 Die **Mindest-Hammingdistanz** $d_{\mathrm{H,min}}(C)$ eines Codes C ist die minimale Hammingdistanz zwischen beliebigen, verschiedenen Codeworten $\boldsymbol{a}, \boldsymbol{b} \in C$, d.h.

$$d_{\mathrm{H,min}}(C) = \min_{\substack{\boldsymbol{a},\boldsymbol{b}\in C \\ \boldsymbol{a}\neq\boldsymbol{b}}} d_{\mathrm{H}}(\boldsymbol{a}, \boldsymbol{b}).$$

□

[3] Üblicherweise wird das Hamminggwicht als Anzahl der Stellen ungleich Null definiert, da als Codesymbole Elemente aus der Menge $\{0, 1\}$ verwendet werden. Um eine einheitliche Beschreibung von BSC und AWGN-Kanal geben zu können, werden in dieser Arbeit jedoch als Codesymbole Elemente aus der Menge $\{\pm 1\}$, mit $+1$ als dem neutralen Element bezüglich der Addition in GF(2), verwendet (s. Seite 5).

Definition 2.4 Das **Mindest-Hamminggewicht** $w_{\mathrm{H,min}}(C)$ eines Codes C ist das Minimum aller Hamminggewichte in einem Code, d.h.

$$w_{\mathrm{H,min}}(C) = \min_{\boldsymbol{c} \in C} w_{\mathrm{H}}(\boldsymbol{c}).$$

□

Satz 2.1 In einem linearen Code ist das Mindestgewicht gleich der Mindestdistanz. □

In gleicher Weise kann die Mindestdistanz im euklidischen Raum $d_{\mathrm{E,min}}(C)$ definiert werden. Für den Spezialfall eines binären Codes mit Codesymbolen aus der Menge $\{\pm 1\}$ gilt für $\boldsymbol{a}, \boldsymbol{b} \in C$

$$d_{\mathrm{E}}^2(\boldsymbol{a}, \boldsymbol{b}) = 4 d_{\mathrm{H}}(\boldsymbol{a}, \boldsymbol{b})$$

und daher auch $d_{\mathrm{E,min}}^2(C) = 4 d_{\mathrm{H,min}}(C)$.

Bei kleiner Symbolfehlerwahrscheinlichkeit p bzw. kleiner Varianz σ_0^2 ist die Mindestdistanz eines Codes ein Maß für seine Korrekturfähigkeit. Bei den betrachteten Kanälen kann man die Störungen allgemein durch einen zufälligen Störungsvektor $\boldsymbol{n} = (n_1, n_2, \ldots, n_N)$ beschreiben, der auf den Codevektor addiert wird

$$\boldsymbol{y} = \boldsymbol{c} + \boldsymbol{n},$$

wobei beim BSC $n_k \in \mathrm{GF}(2) = \{\pm 1\}$ und beim AWGN-Kanal $n_k \in \mathbb{R}$ ist, und die Addition entsprechend des Zahlenraumes beim BSC in GF(2) und beim AWGN-Kanal in $\mathbb{R}$ ausgeführt wird. Die Gesamtwahrscheinlichkeit $P(\boldsymbol{n})$ ist unabhängig vom gesendeten Codewort und gegeben durch das Produkt der Einzelwahrscheinlichkeiten $P(n_k)$. Beim BSC ist $P(n_k = -1) = p$, beim AWGN-Kanal gilt die Unabhängigkeit entsprechend für die Wahrscheinlichkeitsdichten, und n_k ist normalverteilt: $n_k \sim N(0, \sigma_0^2)$. Ein Störungsvektor kann bei ML-Decodierung nur dann zu einem Decodierfehler führen, d.h. $\boldsymbol{c}_{\mathrm{ML}} \neq \boldsymbol{c}$, falls beim BSC gilt

$$w_{\mathrm{H}}(\boldsymbol{n}) \geq d_{\mathrm{H,min}}(C)/2 \tag{2.6}$$

bzw. bei Übertragung über den AWGN-Kanal[4]

$$||\boldsymbol{n}|| \geq d_{\mathrm{E,min}}(C)/2 = \sqrt{d_{\mathrm{H,min}}(C)}. \tag{2.7}$$

Natürlich führt nicht jeder Störungsvektor mit $w_{\mathrm{H}}(\boldsymbol{n}) \geq d_{\mathrm{H,min}}(C)/2$ bzw. $||\boldsymbol{n}|| \geq d_{\mathrm{E,min}}(C)/2$ zu einem Decodierfehler, bei ML-Decodierung können im Gegenteil noch viele weitere Störungsvektoren korrigiert werden. Jedoch führt in vielen Fällen die Begrenzung der durch den Decoder korrigierten Störungsvektoren zu einer wesentlichen Verringerung der Decodierkomplexität, natürlich zu Lasten der Datensicherheit. Innerhalb der Klasse der suboptimalen Decodierverfahren werden Decoder, die nur Fehlervektoren bis zur halben Mindestdistanz korrigieren und sich bei allen Störungsvektoren, die die Ungleichungen (2.6) und (2.7) erfüllen, nicht für ein Codewort entscheiden (sog. Decodierversagen), als Begrenzte-Mindestdistanz-Decoder (engl.: Bounded Minimum Distance Decoder, BMD-Decoder) bezeichnet.

[4] Die Norm eines Vectors $||\boldsymbol{n}||$ ist gegeben durch $||\boldsymbol{n}|| = \sqrt{\sum_k n_k^2}$

2.3. Verallgemeinerte Codeverkettung

Kurze Codes werden im allgemeinen direkt konstruiert, indem man z.B. die Generatormatrix angibt oder bei Polynomcodes, die hier nicht behandelt werden, das Generator- bzw. Prüfpolynom. Bei größeren Codelängen erweist es sich aber als sinnvoll, durch Verkettung kurzer Codes den Gesamtcode zu konstruieren. Im einfachsten Fall einer zweistufigen Verkettung besteht der Gesamtcode dann aus einem *inneren Code*, dessen Codesymbole über den Kanal übertragen werden und dessen Informationssymbole gleich der Codesymbole einer oder mehrerer *äußerer Codes* sind. Die zu codierende Information $\boldsymbol{i}$ wird damit in zwei Stufen zuerst von den äußeren Codes und dann vom inneren Code codiert. Die folgende Beschreibung von verallgemeinert verketteten Codes (engl.: Generalized Concatenated Codes, GC-Codes) ist angelehnt an ZINOV'EV, der sie in [73] für Codes, auch nichtlineare, über beliebigem Alphabet gegeben hat. Hier wird allerdings nur der Spezialfall von binären linearen Codes betrachtet.

Definition 2.5 Es sei A ein binärer (N_A, K_A, D_A)-Code der Länge N_A, Minimum-Hammingdistanz D_A und 2^{K_A} Codeworten. Mit den m gleichlangen Codes A_l, $1 \leq l \leq m$ mit Parametern (N_A, K_{A_l}, D_{A_l}) sei $\boldsymbol{M}$ eine $N_A \times m$ Matrix, deren l-te Spalte[5] ein beliebiges Codewort des Codes A_l enthalte. Weiter sei durch die Generatormatrix $\boldsymbol{B}$ ein binärer $(N_B, K_B = m, D_B)$-Code B gegeben. Die Codevektoren des **verketteten Codes** sind dann durch zeilen- oder spaltenweises Auslesen der Matrix $\boldsymbol{C}$, mit

$$\boldsymbol{M} \cdot \boldsymbol{B} = \boldsymbol{C}$$

gegeben. □

In bezug auf die oben gegebene Einteilung werden die Codes A_l als äußere Codes und B als innerer Code bezeichnet. Um die Mindestdistanz des Gesamtcodes zu bestimmen, müssen die Distanzen der verschiedenen Untercodes von B betrachtet werden. Da hier nur mit linearen inneren und äußeren Codes gearbeitet wird, genügt es, nur die Distanzen der linearen Untercodes von B zu untersuchen.

Satz 2.2 Sei $d^{(l)}_{\text{H,min}}$ die Mindestdistanz des Codes $B^{(l)}$ mit

$$B^{(l)} = \{\boldsymbol{b} : \boldsymbol{b} = (1, 1, \ldots, 1, i_l, \ldots, i_m) \cdot \boldsymbol{B}, i_k \in \{\pm 1\}\}.$$

Damit ist $B^{(1)} = B$, $d^{(1)}_{\text{H,min}} = D_B$ und $d^{(m+1)}_{\text{H,min}} = \infty$. Es gilt dann für die Mindestdistanz des Gesamtcodes $D_{\text{H,ges}}$

$$D_{\text{H,ges}} \geq \min_l (d^{(l)}_{\text{H,min}} \cdot D_{A_l}). \tag{2.8}$$

und der Gesamtcode hat die Parameter $(N, K, D) = (N_A \cdot N_B, \sum_l K_{A_l}, D_{\text{H,ges}})$. □

2.4. Theoretische Grenzen der digitalen Nachrichtenübertragung

Der wohl berühmteste Begriff der digitalen Nachrichtenübermittelung ist die sog. **Kanalkapazität** [50], die 1948 von SHANNON eingeführt wurde. Anschaulich beschreibt die Kanalkapazität

[5] In [73] enthält jede Zeile von $\boldsymbol{M}$ ein Codewort

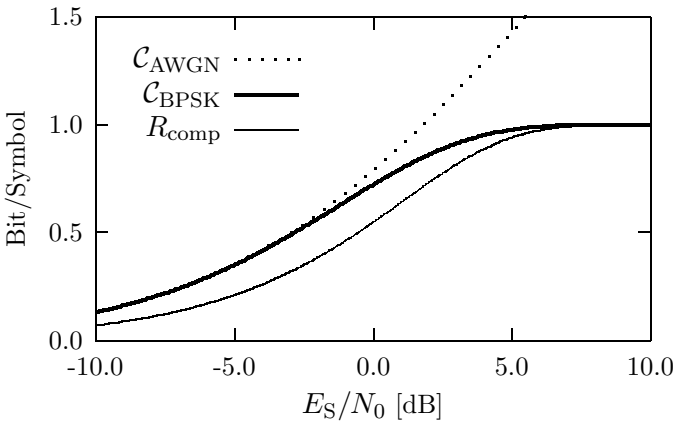

Abbildung 2.1.: Kanalkapazitäten und R_{comp} für BPSK-Übertragung, alle für den AWGN-Kanal

$\mathcal{C}$ die maximale Datenrate, mit der über den betreffenden Kanal mit beliebig kleiner Restfehlerwahrscheinlichkeit übertragen werden kann. Es handelt sich hierbei um eine theoretische Grenze, da u.a. in der Herleitung zufällige Codes[6] verwendet werden, die zwar für $N \to \infty$ sehr gute Korrektureigenschaften haben, bei denen aber für große Codelängen weder effiziente Codier- noch Decodieralgorithmen existieren. Die wesentliche Bedeutung der Kanalkapazität resultiert aus der Aussage, daß zum einen für jede Rate $R < \mathcal{C}$ Codes existieren, mit denen zumindest für $N \to \infty$ eine beliebig kleine Fehlerwahrscheinlichkeit $P(\boldsymbol{c}_{\text{ML}} \neq \boldsymbol{c}|\boldsymbol{y}) > 0$ erreicht werden kann und zum anderen, daß es für $R > \mathcal{C}$ keinen Code mit beliebig kleiner Fehlerwahrscheinlichkeit gibt.

Größen, die in den Wert von $\mathcal{C}$ einfließen, sind z.B. die Rauschleistung auf dem Kanal sowie das verwendete Symbolalphabet, die Empfangsleistung und die Art der Detektion auf der Empfängerseite. Für gedächtnislose Kanäle mit diskreten Eingangsalphabet X und Ausgangsalphabet Y ist die Kanalkapazität in Bit pro Symbol gegeben durch das Maximum des Erwartungswerts

$$\mathcal{C} = \max_{P(x)} E_{X,Y} \left\{ \log_2 \frac{P(y,x)}{P(x)P(y)} \right\} \tag{2.9}$$

über alle möglichen Eingangssymbolwahrscheinlichkeiten $P(x)$. Bei kontinuierlichen Eingangs- oder Ausgangswerten x, y sind die Wahrscheinlichkeiten durch die entsprechenden Wahrscheinlichkeitsdichten zu ersetzen. Im AWGN-Kanal wird die größte Kanalkapazität bei vorgegebener mittlerer Signalempfangsenergie pro Symbol E_S durch wertkontinuierliche, gaußverteilte x erreicht und ist gegeben durch (siehe z.B. [19, Abschnitt 2.5])

$$\mathcal{C}_{\text{AWGN}} = \frac{1}{2} \log_2(1 + 2 \cdot E_S/N_0).$$

Die Kanalkapazität für BPSK-Übertragung, d.h. $X = \{\pm 1\}$ läßt sich nicht in geschlossener Form angeben, der Erwartungswert muß numerisch berechnet werden. Die Wahrscheinlichkeit, die den Erwartungswert maximiert, ist hier durch $P(x = 1) = P(x = -1) = 0.5$ gegeben.

[6] Bei zufälligen Codes werden die Codewörter zufällig gewählt. Der Code C besitzt damit keinerlei Struktur und es kann daher z.B. auch keine Generatormatrix angegeben werden.

Eine weitere wichtige Größe der Nachrichtenübertragung ist die sog. computational **Cutoff-Rate**[7] R_{comp}, die besonders bei der sequentiellen und der Listendecodierung von Bedeutung ist und für alle Kanäle kleiner oder maximal gleich zur Kanalkapazität ist. Von JACOBS und BERLEKAMP [27] wurde gezeigt, daß bei sequentieller Decodierung für $R > R_{\text{comp}}$ die mittlere Anzahl der Operationen, die zur fehlerfreien Decodierung benötigt werden, gegen unendlich geht und damit die sequentielle Decodierung unpraktikabel wird. Für Listendecodierung wurde von FORNEY [14] gezeigt, daß die Forderung nach einer mit der Codelänge N exponentiell fallenden mittleren Listengröße, die in jedem Fall das gesendete Codewort enthält, auf die Bedingung[8] $R \leq R_{\text{comp}}$ führt. Außerdem konnte FORNEY zeigen, daß bei Raten $R < R_{\text{comp}}$ der Gewinn an Fehlerwahrscheinlichkeit eines Listendecoders gegenüber einem klassischen Decoder, der sich immer nur für ein einziges Codewort entscheidet, groß ist, daß aber für $R > R_{\text{comp}}$ nur ein geringer Gewinn zu erwarten ist. Unter anderem daher wird die Cutoff-Rate auch als eine Grenze für praktikable Datenübertragung (ohne Rückkanal) betrachtet, da für $R_{\text{comp}} < R < \mathcal{C}$ zwar theoretisch eine fehlerfreie Datenübertragung möglich ist, dies aber nur durch einen erheblich erhöhten Decodieraufwand erreicht werden kann[9].

R_{comp} ist allgemein gegeben durch

$$R_{\text{comp}} = \max_{P(x)} \left[-\log_2 \sum_{y \in Y} \left(\sum_{x \in X} P(x) \sqrt{P(y|x)} \right)^2 \right] \tag{2.10}$$

wobei für kontinuierliche Eingangs- oder Ausgangswerte x, y die Summen in Integrale übergehen. Für BPSKÜbertragung über den AWGN-Kanal, d.h. $X = \{\pm 1\}$, läßt sich R_{comp} in geschlossener Form angeben

$$R_{\text{comp}} = 1 - \log_2(1 + \mathrm{e}^{-E_S/N_0}). \tag{2.11}$$

Alle Größen sind im Bild 2.1 für Übertragung über den AWGN-Kanal gezeigt.

[7] Dieser Wert wird auch als R_0-Kriterium bezeichnet

[8] Streng genommen wurde in [14] gezeigt, daß $R < \mathcal{C}_{0,1}$ mit der 'zero-error' Kapazität $\mathcal{C}_{0,1} \leq R_{\text{comp}}$ gelten muß.

[9] Wie schon in der Einleitung erwähnt, sind mit den sog. *Turbo-Codes* auch bei vertretbarer Decodierkomplexität Raten $R > R_0$ möglich.

3. Die PLOTKIN-Konstruktion

In diesem Kapitel wird das wesentliche Basiselement der in dieser Arbeit betrachteten Codekonstruktion vorgestellt. Nach der Beschreibung folgt die Definition zweier verschiedener Methoden, die Zuverlässigkeitswerte an die Decoder der äußeren Codes weiter zu reichen. Beide Methoden sind im wesentlichen nicht neu, wobei jedoch die auf der euklidischen Distanz basierende Übergabe gegenüber der von SCHNABL und BOSSERT [49] vorgestellten leicht verändert ist. Daran anschließend wird der Begriff des äquivalenten SNR eingeführt. Dieses Kapitel schließt mit einer Einführung in die rekursive Codekonstruktion.

3.1. Beschreibung

Die binäre $|u|u+v|$-Konstruktion kann auf verschiedene Arten beschrieben werden. Zuerst definiert wurde sie schon 1951 von PLOTKIN [44][1]. Aus der Sicht der GC-Codes ist sie eine spezielle Form der Codeverkettung, bei der zwei binäre äußere Codes U und V gleicher Länge mit einem inneren Code der Länge $N_B = 2$, $K_B = 2$ und Distanz $D_B = 1$ verknüpft werden [49]. Es wird (vergl. Def. 2.5) $A_1 = V$ und $A_2 = U$ gewählt und die Generatormatrix von B ist gegeben durch[2]

$$\boldsymbol{B} = \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix}.$$

Der Code B enthält als Codewörter somit alle 2-Tupel $\boldsymbol{b} = (b_1, b_2)$ mit $b_k \in \{\pm 1\}$. Mit den Codeworten $\boldsymbol{v} = (v_1, v_2, \ldots, v_N) \in V$ und $\boldsymbol{u} = (u_1, u_2, \ldots, u_N) \in U$ liest sich das Produkt $\boldsymbol{MB} = \boldsymbol{C}$ als[3]

$$\begin{bmatrix} v_1 & u_1 \\ v_2 & u_2 \\ \vdots & \vdots \\ v_N & u_N \end{bmatrix} \cdot \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} u_1 & u_1 \oplus v_1 \\ u_2 & u_2 \oplus v_2 \\ \vdots & \vdots \\ u_N & u_N \oplus v_N \end{bmatrix}$$

und das Codewort des verketteten Codes C ergibt sich z.B. bei spaltenweisem Auslesen der Matrix $\boldsymbol{C}$ zu

$$\boldsymbol{c} = (\boldsymbol{u}, \boldsymbol{u} + \boldsymbol{v}) = (u_1, u_2, \ldots, u_N, u_1 \oplus v_1, u_2 \oplus v_2, \ldots, u_N \oplus v_N).$$

[1] In [67] wird angemerkt, daß dieser Arikel schon im Jahre 1951 als Forschungsbericht erschienen ist.

[2] Auch hier werden als Elemente aus GF(2) die Symbole $\{\pm 1\}$ verwendet, mit der Eins als dem neutralen Element bezüglich der Addition (s. auch Fußnote 1 auf Seite 5)

[3] Rechnung in GF(2), $\oplus$ bezeichnet die Exclusive-Oder-Operation (EXOR), die hier äquivalent zur Addition der Elemente in GF(2) ist.

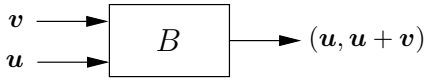

Abbildung 3.1.: Block zur vektoriellen Verknüpfung von $\boldsymbol{u}$ und $\boldsymbol{v}$

Betrachtet man die Codesymbolpaare $\boldsymbol{c}_k = (c_k, c_{N+k})$ für jedes k getrennt voneinander, kann die Codierung auch folgendermaßen beschrieben werden: mit der Menge $B^{(1)} = B$ der binären 2-Tupel $\boldsymbol{b} \in \{(\pm 1, \pm 1)\}$ wird durch $v_k \in \{\pm 1\}$ aus den disjunkten Untermengen $B_\nu^{(2)} \subset B^{(1)}$, $\nu \in \{\pm 1\}$ mit

$$\begin{aligned} B_{+1}^{(2)} &= \{(+1,+1),(-1,-1)\} \\ B_{-1}^{(2)} &= \{(+1,-1),(-1,+1)\} \end{aligned}$$

die Menge $B_{v_k}^{(2)}$ ausgewählt, u_k wählt dann im zweiten Schritt aus $B_{v_k}^{(2)}$ den Vektor $\boldsymbol{b}^{(v_k,u_k)} = (u_k, u_k \oplus v_k) \in B_{v_k}^{(2)}$, dessen Komponenten anschließend dem Paar $\boldsymbol{c}_k = (c_k, c_{N+k})$ zugewiesen werden. Die Verknüpfung der beiden Vektoren $\boldsymbol{u}$ und $\boldsymbol{v}$ zum Vektor $(\boldsymbol{u}, \boldsymbol{u} + \boldsymbol{v})$ ist in Bild 3.1 durch einen Block dargestellt.

Genauer betrachtet wird durch die Codierung mit B, verglichen mit einem simplen Aneinanderhängen der beide Codeworte $\boldsymbol{u}$ und $\boldsymbol{v}$, keine zusätzliche Redundanz eingefügt, B ist ein $(2, 2, 1)$-Code. Durch die Überlagerung von $\boldsymbol{u}$ und $\boldsymbol{v}$ werden jedoch die Verknüpfungen der Codesymbole untereinander verstärkt, und damit hat der resultierende $|u|u + v|$-Code ein wesentliches größeres Korrekturpotential als der gleichlange $|u|v|$-Code.

Der Code $B^{(1)} = B$ (vergl. Satz 2.2) enthält alle 2-Tupel als Codewörter und daher gilt $d_{\text{H,min}}^{(1)} = 1$. $B^{(2)}$ dagegen enthält nur die Codewörter $(1, 1)$ und $(-1, -1)$, damit ist $d_{\text{H,min}}^{(2)} = 2$, und daher kann die Mindestdistanz des verketteten Codes gemäß (2.8) nach unten abgeschätzt werden zu

$$D_{\text{H,min}} \geq \min(D_{\text{H,min}}^{(V)}, 2D_{\text{H,min}}^{(U)}),$$

mit der Mindestdistanz $D_{\text{H,min}}^{(U)}$ des Codes U und der Mindestdistanz $D_{\text{H,min}}^{(V)}$ des Codes V. Allerdings sind für beliebige aber feste $\boldsymbol{v}_0 \in V$ und $\boldsymbol{u}_0 \in U$ die Untercodes $(\boldsymbol{u}, \boldsymbol{u} + \boldsymbol{v}_0)$, $\boldsymbol{u} \in U$ und $(\boldsymbol{u}_0, \boldsymbol{u}_0 + \boldsymbol{v})$, $\boldsymbol{v} \in V$ in C enthalten und damit kann $D_{\text{H,min}}$ nicht größer sein als $2D_{\text{H,min}}^{(U)}$ bzw. $D_{\text{H,min}}^{(V)}$, weshalb gilt

$$D_{\text{H,min}} = \min(D_{\text{H,min}}^{(V)}, 2D_{\text{H,min}}^{(U)}). \tag{3.1}$$

Die graphische Darstellung der Codewörter von B in der 2-dimensionalen Ebene (s. Abb. 3.2) verdeutlicht die Verwandtschaft der $|u|u + v|$-Konstruktion mit der QPSK-Modulation [46, Chapter 4].

Falls die Komponenten des Vektors $\boldsymbol{b} = (b_1, b_2)$ als Inphase- und Quadratur-Komponente I und Q betrachtet werden, ergeben sich automatisch die gleichen Punkte im zweidimensionalen Signalraum, d.h. daß die Verkettung mit dem inneren Code B äquivalent auch als QPSK-Modulation betrachtet werden kann, falls die Zuordnung der Codesymbole (u_i, v_i) zu den Signalpunkten $\in \mathbb{R}^2$ entsprechend gewählt wird. Die Verbindung von Codierung und Modulation wurde u.a. von UNGERBOECK [62] untersucht. Bei der von ihm vorgeschlagenen Konstruktion

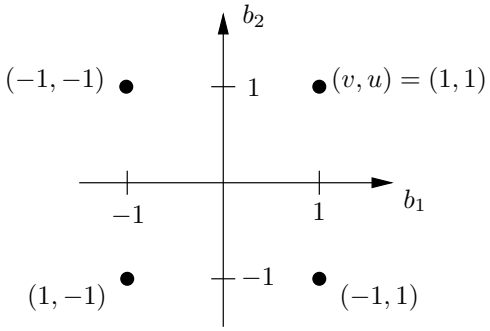

Abbildung 3.2.: Graphische Darstellung der Codeworte von B in Abhängigkeit von (v,u)

wird die Menge der Signalpunkte stufenweise so partitioniert, daß auf jeder Stufe die minimale euklidische Distanz in jeder Menge maximal wird. Auf jeder Stufe werden dann die verschiedenen Mengen nummeriert und diese Nummerierung dann durch einen Code geschützt. Diese Partitionierungsregel ist in Abb. 3.3 exemplarisch für die vier Signalpunkte der QPSK-Modulation ausgeführt und es zeigt sich, daß die $|u|u+v|$-Konstruktion, obwohl viel früher bekannt, die Kriterien der sog. *Ungerboeck-Partitionierung* inhärent erfüllt.

3.2. Zuverlässigkeitsübergabe

In diesem Abschnitt wird auf die Decodierung, d.h. auf die Schätzung der Symbole u_k und v_k in Abhängigkeit von $\boldsymbol{y}$ eingegangen. Der optimale Decodieralgorithmus im Sinne der kleinsten Wortfehlerwahrscheinlichkeit bestimmt natürlich unter Kenntnis der Codes U und V gemäß Gl. (2.5) das wahrscheinlichste Codewort $\boldsymbol{c}_{\mathrm{ML}}$ und dann daraus den Informationsvektor $\boldsymbol{i}$. In Abschnitt 2.1 wurde allerdings schon angesprochen, daß aus Komplexitätsgründen gerade bei längeren Codes die ML-Decodierung nicht durchführbar ist. Wie noch später gezeigt wird (s. Korollar 5.1, S. 44), kann bei BMD-Decodierung allerdings der Vektor $\boldsymbol{v} \in V$ unabhängig von $\boldsymbol{u}$ entschieden werden und dann unter Kenntnis von $\boldsymbol{v}$ das Codewort $\boldsymbol{u}$ (sog. Zweistufendecodierung) [49],[67]. Dazu werden im ersten Schritt aus dem Empfangsvektor $\boldsymbol{y}$ die Symbole $v_k \in \{\pm 1\}$ getrennt voneinander und unabhängig von $\boldsymbol{u}$ geschätzt und Zuverlässigkeitswerte der einzelnen Schätzungen berechnet. In die Decodierung von V fließen dann die Schätzungen als auch deren Zuverlässigkeitswerte ein. In gleicher Weise werden dann im zweiten Schritt, abhängig von der Entscheidung $\widehat{\boldsymbol{v}} \in V$, die Symbole u_k geschätzt und Zuverlässigkeitswerte berechnet, die bei der Decodierung von U verwendet werden.

Vereinfachend werden im folgenden nur die zwei Symbole c_k und c_{N+k} des Codewortes betrachtet, die sich aus den Symbolen u_k und v_k wie folgt ergeben

$$\boldsymbol{c}_k = (c_k, c_{N+k}) = (u_k, u_k \oplus v_k).$$

Falls beim BSC mit $y_i \in \{\pm 1\}$ kein Fehler aufgetreten ist, d.h. $\boldsymbol{y}_k = (y_k, y_{N+k}) = (c_k, c_{N+k})$, kann v_k leicht aus den Empfangssymbolen zurückgewonnen werden, d.h. bei Rechnung in GF(2) gilt

$$\widetilde{v}_k = y_k \oplus y_{N+k} = u_k \oplus (u_k \oplus v_k) = v_k,$$

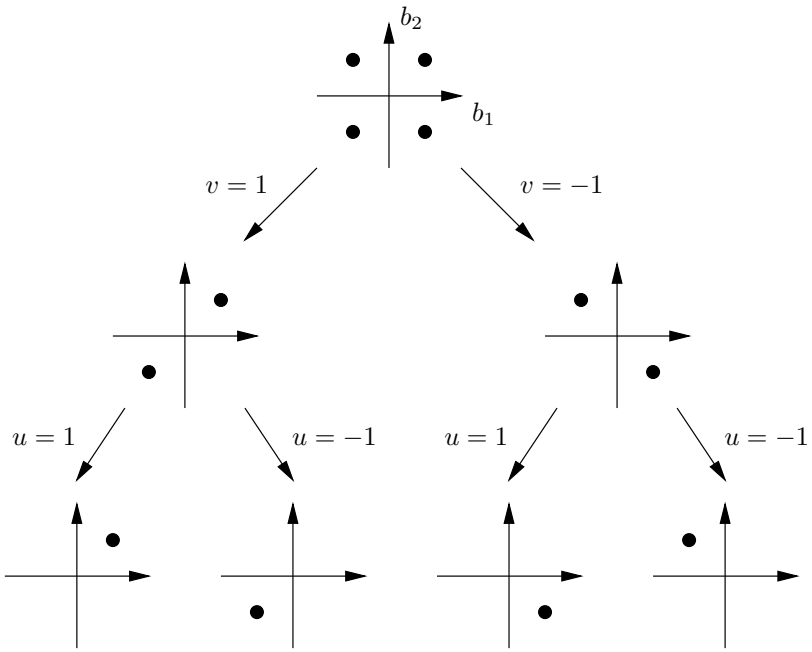

Abbildung 3.3.: Partitionierung der Codeworte von B

da das Symbol v_k nur in y_{N+k} enthalten ist, u_k aber symmetrisch in beiden Symbolen y_k und y_{N+k}. Aus der Sicht der Codeverkettung ist die Schätzung $\widetilde{v}_k$ identisch mit dem Index $\nu \in \{\pm 1\}$ der Untermenge $B_\nu^{(2)}$ (s. Seite 14), die den Vektor $\boldsymbol{y}_k$ enthält. Beim AWGN-Kanal mit $y_k \in \mathbb{R}$ kann v_k in gleicher Weise geschätzt werden, indem die Menge $B_\nu^{(2)}$ bestimmt wird, die den *wahrscheinlichsten* Vektor $\boldsymbol{b} \in \{(\pm 1, \pm 1)\}$ enthält, d.h. mit $\boldsymbol{b}^{(u,v)} = (u, u \oplus v)$ und $B_\nu^{(2)} = \{\boldsymbol{b}^{(u,v)} : v = \nu, u = \pm 1\}$ gilt

$$\widetilde{v}_k = \arg\min_{\nu} \left\{ \min_{\boldsymbol{b} \in B_\nu^{(2)}} (\boldsymbol{y}_k - \boldsymbol{b})^2 \right\}. \tag{3.2}$$

Dies ist sowohl beim BSC als auch beim AWGN-Kanal mit $a_k = \operatorname{sign} y_k$ gleichwertig zur Schätzung

$$\widetilde{v}_k = a_k \cdot a_{N+k}, \tag{3.3}$$

da die Vorzeichenentscheidung der Empfangswerte y_k und y_{N+k} direkt den wahrscheinlichsten Vektor $\boldsymbol{b} \in B^{(1)}$ liefert und die Untermenge $B_v^{(2)}$, die diesen Vektor enthält, durch $v = b_1 \oplus b_2$ gegeben ist. Es handelt sich bei $\widetilde{v}_k$ um eine harte Entscheidung ohne Zuverlässigkeitswerte. Da beim BSC die Zuverlässigkeit aller Empfangswerte y_k gleich ist und damit auch die Zuverlässigkeit aller Schätzungen $\widetilde{v}_k$, ist es hier sicherlich nicht sinnvoll, zusätzlich noch Zuverlässigkeitswerte zu berechnen. Beim AWGN-Kanal gilt dies nicht, die Zuverlässigkeit der Schätzungen hängt stark von den Werten $y_k \in \mathbb{R}$ ab und hier sollen für eine effektive Decodierung Zuverlässigkeiten angegeben werden. Wie diese berechnet werden können, wird in den beiden folgenden Abschnitten näher beschrieben.

Nach der Decodierung von V werden im zweiten Schritt ausgehend von der Annahme, daß die Decodierentscheidung $\widehat{\boldsymbol{v}} \in V$ richtig ist, die Symbole u_k ebenfalls getrennt voneinander, aber in

Abhängigkeit von $\widehat{\boldsymbol{v}}$ geschätzt und dem Decoder von U zur Verfügung gestellt. Die Schätzung der Symbole u_k kann in gleicher Weise erfolgen wie im ersten Schritt die Schätzung der Symbole v_k, d.h. daß aus der Menge der möglichen Vektoren $\{\boldsymbol{b}^{(u,v)} : v = \widehat{v}_k\}$ der wahrscheinlichste ermittelt wird:

$$\widetilde{u}_k = \arg\min_u (\boldsymbol{y}_k - \boldsymbol{b}^{(u,\widehat{v}_k)})^2.$$

Aber schon hier wird bei genauerer Betrachtung deutlich, daß die harten Entscheidungen $\widetilde{u}_k$ auch beim BSC nicht ausreichen. Falls bei der Übertragung innerhalb des 2-Tupels $\boldsymbol{y}_k$ ein Fehler aufgetreten und die Entscheidung $\widehat{v}_k$ richtig ist, gilt $d_{\mathrm{E}}^2(\boldsymbol{y}_k, \boldsymbol{b}^{(1,\widehat{v}_k)}) = d_{\mathrm{E}}^2(\boldsymbol{y}_k, \boldsymbol{b}^{(-1,\widehat{v}_k)})$, damit sind beide Schätzungen $\widetilde{u}_k = 1$ und $\widetilde{u}_k = -1$ gleichwahrscheinlich und dies wird im allgemeinen als 'Auslöschung' (Zuverlässigkeitswert gleich Null) behandelt.

In den Folgen beiden Abschnitten werden zwei verschiedene Methoden vorgestellt, Zuverlässigkeitswerte zu bestimmen, die sowohl für den AWGN-Kanal als auch den BSC geeignet sind.

3.2.1. Auf Wahrscheinlichkeit basierte Zuverlässigkeitswerte

Die Zuverlässigkeiten können auf verschiedene Arten definiert werden. Ein Weg [8][9][49] ist, die Wahrscheinlichkeiten $P(v_k = 1|\boldsymbol{y}_k)$ und $P(v_k = -1|\boldsymbol{y}_k)$ zu verwenden. Um diese zu berechnen, werden die Wahrscheinlichkeiten $P(c_k|y_k)$ benötigt, und da die Codesymbole c_k zweiwertig sind, kann anstelle auch der Quotient $P(c_k = 1|y_k)/P(c_k = -1|y_k)$ benutzt werden. Für gleichwahrscheinliche Codesymbole, d.h. $P(c_k = 1) = P(c_k = -1)$ gilt

$$\frac{P(c_k = 1|y_k)}{P(c_k = -1|y_k)} = \frac{P(y_k|c_k = 1)P(c_k = 1)}{P(y_k|c_k = -1)P(c_k = 1)} = \frac{P(y_k|c_k = 1)}{P(y_k|c_k = -1)}.$$

Beim BSC ist $P(c_k|y_k)$ bzw. $P(y_k|c_k)$ direkt durch die Symbolfehlerwahrscheinlichkeit p, im Fall des AWGN-Kanals ist $f(y_k|c_k)$ durch Gl. (2.1) gegeben. Beim AWGN-Kanal bietet sich zur Darstellung zudem der Übergang in den log-Bereich an, da die Wahrscheinlichkeitsdichten $f(y_k|c_k) \geq 0$ Exponentialfunktionen sind,

$$g_k = \ln \frac{P(y_k|c_k = 1)}{P(y_k|c_k = -1)} \overset{\text{AWGN}}{=} 2y_k/\sigma_0^2.$$

Für die Umkehrung gilt

$$P(c_k = \pm 1|y_k) = \frac{\mathrm{e}^{\pm g_k/2}}{\mathrm{e}^{g_k/2} + \mathrm{e}^{-g_k/2}}.$$

Es ist weiter sinnvoll, die Differenz der Wahrscheinlichkeiten wie in [8],[9]

$$h_k = P(c_k = 1|y_k) - P(c_k = -1|y_k) = \tanh \frac{g_k}{2}$$

einzuführen. Bei den Größen y_k, g_k und h_k handelt es sich um gleichberechtigte Werte, da sie sich ohne Verlust ineinander umrechnen lassen. Daher kann die optimale Entscheidungsregel des ML-Decoders nach Gl. (2.5) auch in Abhängigkeit z.B. des Vektors $\boldsymbol{h} = (h_1, h_2, \ldots, h_{2N})$ angegeben werden. Wie im Anhang Abschnitt A.1 gezeigt wird, ist (2.5) äquivalent zu

$$\boldsymbol{c}_{\mathrm{ML}} = \arg\max_{\boldsymbol{c} \in C} \prod_k (1 + c_k \cdot h_k). \tag{3.4}$$

Obwohl die Rauschvarianz σ_0^2 in die Zuverlässigkeitswerte h_k einfließt und damit die Entscheidung nach Gl. (3.4) scheinbar von σ_0^2 abhängt, ist die ML-Entscheidung natürlich weiterhin unabhängig von der Rauschvarianz, da aus (3.4) direkt (2.5) abgeleitet werden kann. Auf den Einfluß von σ_0^2 bei der Entscheidung von $\widehat{\boldsymbol{v}}$ wird in Abschnitt 3.2.3 noch genauer eingegangen.

Die Berechnung der Zuverlässigkeiten der Schätzung $\widetilde{v}_k$ erfolgt am einfachsten mit Hilfe der h_k, denn für die Differenz der Wahrscheinlichkeiten $h_k^{(v)}$ gilt

$$\begin{aligned} h_k^{(v)} &= P(v_k = 1|\boldsymbol{y}_k) - P(v_k = -1|\boldsymbol{y}_k) \\ &= P(c_k = 1|y_k) \cdot P(c_{N+k} = 1|y_{N+k}) + P(c_k = -1|y_k) \cdot P(c_{N+k} = -1|y_{N+k}) \\ &\quad -P(c_k = 1|y_k) \cdot P(c_{N+k} = -1|y_{N+k}) - P(c_k = -1|y_k) \cdot P(c_{N+k} = 1|y_{N+k}) \\ &= h_k \cdot h_{N+k}. \end{aligned} \tag{3.5}$$

Da das Vorzeichen von $h_k^{(v)} \in \mathbb{R}$ identisch mit der aus Gl. (3.3) gewonnen Schätzung $\widetilde{v}_k$ ist, muß diese nicht separat durchgeführt werden.

Basierend auf der Entscheidung von $\widehat{\boldsymbol{v}} \in V$ ergeben sich die Zuverlässigkeiten für $\widetilde{u}_k$ mit $c_k = u_k$ und[4] $c_{N+k} = u_k \cdot v_k$ wie folgt

$$\mathrm{e}^{g_k^{(u)}} = \frac{P(u_k = 1|\boldsymbol{y}_k, \widehat{v}_k)}{P(u_k = -1|\boldsymbol{y}_k, \widehat{v}_k)} = \frac{P(\boldsymbol{y}_k|u_k = 1, \widehat{v}_k)}{P(\boldsymbol{y}_k|u_k = -1, \widehat{v}_k)} = \frac{P(y_k|c_k = 1)}{P(y_k|c_k = -1)} \cdot \frac{P(y_{N+k}|c_{N+k} = \widehat{v}_k)}{P(y_{N+k}|c_{N+k} = -\widehat{v}_k)}.$$

Dies kann auch folgendermaßen geschrieben werden

$$\mathrm{e}^{g_k^{(u)}} = \frac{P(y_k|c_k = 1)}{P(y_k|c_k = -1)} \cdot \left(\frac{P(y_{N+k}|c_{N+k} = 1)}{P(y_{N+k}|c_{N+k} = -1)} \right)^{\widehat{v}_k} = \mathrm{e}^{g_k} \cdot \mathrm{e}^{\widehat{v}_k \cdot g_{N+k}}$$

und damit ergibt sich

$$g_k^{(u)} = g_k + \widehat{v}_k \cdot g_{N+k}. \tag{3.6}$$

Rechnet man dieses Ergebnis auf die Wahrscheinlichkeitsdifferenz $h_k^{(u)}$ um, so ergibt sich

$$h_k^{(u)} = \tanh\left(\frac{g_k + \widehat{v}_k \cdot g_{N+k}}{2}\right) = \frac{\tanh\frac{g_k}{2} + \widehat{v}_k \cdot \tanh\frac{g_{N+k}}{2}}{1 + \widehat{v}_k \cdot \tanh\frac{g_k}{2} \cdot \tanh\frac{g_{N+k}}{2}} = \frac{h_k + \widehat{v}_k \cdot h_{N+k}}{1 + \widehat{v}_k \cdot h_k \cdot h_{N+k}}. \tag{3.7}$$

Auch hier ist das Vorzeichen von $g_k^{(u)}$ bzw. $h_k^{(u)}$ mit der Schätzung $\widetilde{u}_k$ identisch.

Die ML-Schätzung ist gemäß (3.4) durch das Produkt der Faktoren $(1 + c_k \cdot h_k)$ vollständig bestimmt. Betrachtet man einen Block $\boldsymbol{c}_k = (c_k, c_{N+k}) = (u_k, u_k \cdot v_k)$ separat, ergibt sich der Faktor

$$(1 + c_k \cdot h_k) \cdot (1 + c_{N+k} \cdot h_{N+k}) = 1 + v_k h_k h_{N+k} + u_k h_k + u_k v_k h_{N+k}.$$

Der gleiche Faktor ergibt sich ausgehend von den $h_k^{(v)}$ und $h_k^{(u)}$ als

$$(1 + v_k \cdot h_k^{(v)}) \cdot (1 + u_k \cdot h_k^{(u)}) = 1 + v_k h_k h_{N+k} + u_k h_k + u_k v_k h_{N+k}$$

[4] Die Addition in GF(2) ist hier gleich der Multiplikation in $\mathbb{R}$.

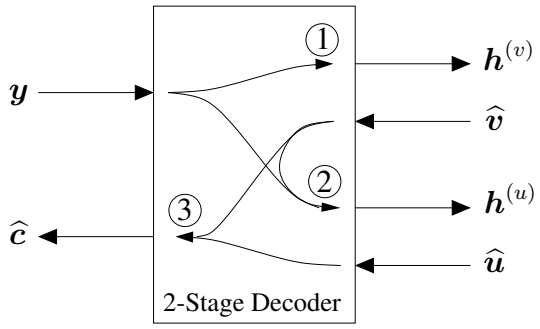

Abbildung 3.4.: Einzelne Schritte bei der Übergabe der Zuverlässigkeitswerte bei Zweistufendecodierung

und damit kann das ML-Codewort auch unter der alleinigen Kenntnis der Zuverlässigkeiten der Symbole $\widetilde{u}_k$ und $\widetilde{v}_k$ der äußeren Codes U und V durch folgende Entscheidungsregel bestimmt werden

$$\boldsymbol{c}_{\mathrm{ML}} = \arg\max_{\boldsymbol{u},\boldsymbol{v}} \prod_k (1 + v_k \cdot h_k^{(v)}) \prod_k (1 + u_k \cdot h_k^{(u)}). \tag{3.8}$$

Natürlich fließen in (3.8) auch die Zuverlässigkeiten der Empfangssymbole h_k mit ein, da insbesondere bei der Maximierung für jedes $\boldsymbol{v}$ die Zuverlässigkeiten $h_k^{(u)}$ aus den h_k neu berechnet werden müssen.

Die einzelnen Schritte zur Berechnung der Zuverlässigkeitswerte sind in Bild 3.4 zu einem Block zusammengefaßt. Schritt Nr. 1 erfolgt hier gemäß Gl. (3.5) und Schritt Nr. 2 gemäß Gl. (3.7). Zum Schluß werden die Vektoren $\widehat{\boldsymbol{v}}$ und $\widehat{\boldsymbol{u}}$ zur Schätzung $\widehat{\boldsymbol{c}}$ zusammengefaßt.

3.2.2. Auf euklidischer Distanz basierte Zuverlässigkeitswerte

Die Zuverlässigkeiten für die Schätzungen $\widetilde{u}_k$ und $\widetilde{v}_k$ können auch bezogen auf die euklidische Distanz zwischen Empfangsvektor und geschätztem Codewort definiert werden, wie schon vom Autor zusammen mit SORGER in [54][57] vorgestellt. Nach Gl. (2.5)

$$\boldsymbol{c}_{\mathrm{ML}} = \arg\min_{\boldsymbol{c}\in C} \sum_{\{k:c_k \neq a_k\}} w_k$$

werden bei der ML-Entscheidung die Beträge $w_k = |y_k|$ der Positionen k aufsummiert, an denen die Entscheidung $\widehat{c}_k$ nicht im Vorzeichen mit dem Empfangswert übereinstimmt, sprich ein Übertragungsfehler aufgetreten ist. Jede Entscheidung $\widehat{c}_k \neq \operatorname{sign} y_k$ erhöht die Summe um w_k. Um bei der Decodierung von V verschiedene Hypothesen für $\boldsymbol{v}$ bewerten zu können, bietet sich als Zuverlässigkeitswert $w_k^{(v)}$ für die Schätzung $\widetilde{v}_k$ das Anwachsen der Summe für $\widehat{v}_k \neq \widetilde{v}_k$ an. Die Schätzung $\widetilde{v}_k$ resultiert aus den beiden Empfangswerten $\boldsymbol{y}_k = (y_k, y_{N+k})$. Falls nun für eine bestimmte Hypothese $\widehat{v}_k \neq \widetilde{v}_k$ gilt, ist es bei unbekanntem u_k jedoch nicht möglich festzustellen, welches Vorzeichen der beiden Werte y_k oder y_{N+k} falsch ist. Allerdings gilt für $\widehat{v}_k \neq \widetilde{v}_k$, daß innerhalb des Blocks $\boldsymbol{y}_k$ genau *ein* Fehler $a_k \neq \widehat{c}_k$ oder $a_{N+k} \neq \widehat{c}_{N+k}$ aufgetreten sein muß, wenn die Hypothese richtig ist. Da der Empfangswert mit kleinerem

Betrag auch zugleich der unzuverlässigere mit höherer Fehlerwahrscheinlichkeit ist, setzt man sinnvollerweise

$$w_k^{(v)} = \min(w_k, w_{N+k}). \tag{3.9}$$

Zu dem gleichen Ergebnis kommt man ausgehend von Gl. (3.2) wenn $w_k^{(v)}$ als *minimaler Zuwachs* der quadratischen euklidischen Distanz, falls die Schätzung $\boldsymbol{c}_k \in B_{\widetilde{v}_k}^{(2)}$ falsch ist (vergl. Gl. (3.2)), definiert wird

$$w_k^{(v)} = \frac{1}{4} \min_{\boldsymbol{b} \in B_{-\widetilde{v}_k}^{(2)}} (\boldsymbol{y}_k - \boldsymbol{b})^2 - \frac{1}{4} \min_{\boldsymbol{b} \in B_{\widetilde{v}_k}^{(2)}} (\boldsymbol{y}_k - \boldsymbol{b})^2.$$

Der Faktor $1/4$ resultiert aus dem Weglassen des Faktors 4 in Gl. (2.5).

Die Zuverlässigkeitswerte $w_k^{(u)}$ ergeben sich auf gleiche Weise als Zuwachs der quadratischen euklidischen Distanz, falls die Schätzung $\widetilde{u}_k$ falsch ist, in Abhängigkeit von der bereits getroffenen Entscheidung $\widehat{v}_k$ zu

$$w_k^{(u)} = \frac{1}{4}(\boldsymbol{y}_k - \boldsymbol{b}^{(-\widetilde{u}_k, \widehat{v}_k)})^2 - \frac{1}{4}(\boldsymbol{y}_k - \boldsymbol{b}^{(\widetilde{u}_k, \widehat{v}_k)})^2. \tag{3.10}$$

Mit $\boldsymbol{b}^{(u,v)} = (u, u \cdot v)$ erhält man daraus

$$w_k^{(u)} = \widetilde{u}_k \cdot (y_k + \widehat{v}_k \cdot y_{N+k}) = |y_k + \widehat{v}_k \cdot y_{N+k}|.$$

In der gleichen Weise, in der sich die Empfangswerte y_k in Vorzeichen a_k und Betrag w_k aufgespalten lassen, kann man umgekehrt die soeben definierten Zuverlässigkeiten $w_k^{(u)}, w_k^{(v)} \in \mathbb{R}^+$ mit den Schätzungen $\widetilde{u}_k, \widetilde{v}_k \in \{\pm 1\}$ zu neuen Empfangswerten $y_k^{(u)}, y_k^{(v)} \in \mathbb{R}$ für die äußeren Codes U und V vereinigen[5]

$$y_k^{(v)} = \widetilde{v}_k \cdot w_k^{(v)} = a_k \cdot a_{N+k} \cdot \min(w_k, w_{N+k}) \tag{3.11}$$

$$y_k^{(u)} = \widetilde{u}_k \cdot w_k^{(u)} = y_k + \widehat{v}_k \cdot y_{N+k}. \tag{3.12}$$

Bei der Decodierung mit auf Distanz basierten Zuverlässigkeitswerten ergeben sich damit die gleichen Schritte wie in Bild 3.4 gezeigt, nur daß hier die Vektoren $\boldsymbol{h}^{(v)}$ und $\boldsymbol{h}^{(u)}$ durch $\boldsymbol{y}^{(v)}$ bzw. $\boldsymbol{y}^{(u)}$ zu ersetzen sind.

Bis auf einen fehlenden Faktor von $1/2$ in Gl. (3.12) ist die hier definierte Zuverlässigkeitsübergabe mit der von SCHNABL und BOSSERT in [49] angegebenen identisch. Aber nur aus der hier gegebenen Definitionen folgt automatisch, daß der quadratische euklidische Abstand zwischen Empfangsvektor $\boldsymbol{y}$ und Codewort $\boldsymbol{c}$, welches durch $\boldsymbol{u}$ und $\boldsymbol{v}$ eindeutig bestimmt ist, auch gegeben ist durch

$$d_{\mathrm{E}}^2(\boldsymbol{y}, \boldsymbol{c}) = d_{\mathrm{bias}}^2 + 4 \sum_{\{k: v_k \neq \widetilde{v}_k\}} w_k^{(v)} + 4 \sum_{\{k: u_k \neq \widetilde{u}_k\}} w_k^{(u)}, \tag{3.13}$$

[5] Da gilt

$$a_k \cdot a_{N+k} \cdot \min(w_k, w_{N+k}) = \frac{1}{2}(|y_k + y_{N+k}| - |y_k - y_{N+k}|)$$

ist die Übergabe nach Gl. (3.11) prinzipiell mit der in [70] definierten Metrik identisch.

und daß damit die ML-Entscheidungsregel Gl. (2.5) auch geschrieben werden kann als

$$\boldsymbol{c}_{\mathrm{ML}} = \arg\min_{\boldsymbol{v},\boldsymbol{u}} \sum_{\{k:v_k \neq \widetilde{v}_k\}} w_k^{(v)} + \sum_{\{k:u_k \neq \widetilde{u}_k\}} w_k^{(u)}. \tag{3.14}$$

Diese Eigenschaft ist später bei der sequentiellen bzw. Listendecodierung wichtig.

3.2.3. Vergleich der Zuverlässigkeitswerte

Ein Vergleich zwischen den in Abschnitt 3.2.2 mit den in Abschnitt 3.2.1 gefundenen Zuverlässigkeiten ergibt, daß einerseits Gl. (3.12) äquivalent zu (3.6) ist, und daß man andererseits, da

$$g_k^{(v)} = 2 \cdot \operatorname{artanh}\left(\tanh\frac{g_k}{2} \cdot \tanh\frac{g_{N+k}}{2}\right)$$

und für den Grenzwert

$$\lim_{|x|\to\infty} \operatorname{artanh}\left(\tanh x \cdot \tanh y\right) \to \operatorname{sign} x \cdot \operatorname{sign} y \cdot |y|,$$

gilt, $y_k^{(v)}$ nach Gl. (3.11) als näherungsweise Berechnung von $g_k^{(v)}$ für $|g_k| \gg |g_{N+k}|$ oder $|g_k| \ll |g_{N+k}|$ betrachten kann.

Die Unterschiede zwischen den verschiedenen Zuverlässigkeitswerten sollen an folgendem Beispiel verdeutlicht werden.

Beispiel 3.1 Betrachtet wird ein Code C der Länge 4 mit $V = \{(1,1),(-1,-1)\}$ (Wiederholcode) und $U = \{(\pm 1,\pm 1)\}$, d.h. U enthält alle 2-Tupel. Aus dem Empfangsvektor $\boldsymbol{y} = (y_1, y_2, y_3, y_4)$ ergeben sich die auf den Wahrscheinlichkeiten basierten Zuverlässigkeitswerte für $\widetilde{v}_1$ und $\widetilde{v}_2$ wie folgt

$$\begin{aligned} h_1^{(v)} &= h_1 \cdot h_3 = \tanh\frac{y_1}{\sigma_0^2} \cdot \tanh\frac{y_3}{\sigma_0^2} \\ h_2^{(v)} &= h_2 \cdot h_4 = \tanh\frac{y_2}{\sigma_0^2} \cdot \tanh\frac{y_4}{\sigma_0^2}. \end{aligned}$$

In der ML-Entscheidungsregel für das wahrscheinlichste Codewort $\boldsymbol{c} \in C$ nach (3.8) werden neben den Zuverlässigkeiten für v auch die Zuverlässigkeiten für u berücksichtigt. Falls im Vorgriff auf Kapitel 5.2 $\widehat{\boldsymbol{v}}$ unabhängig von $\boldsymbol{u}$ entschieden wird, ist für den hier gegebenen Fall, daß U alle 2^N möglichen 2-Tupel enthält, das wahrscheinlichste Codewort $\boldsymbol{v} \in V$ durch

$$\begin{aligned} \widehat{\boldsymbol{v}} &= \arg\max_{\boldsymbol{v}\in V} P(\boldsymbol{v}|\boldsymbol{y}) \\ &= \arg\max_{\boldsymbol{v}\in V} \prod_k (1 + v_k h_k^{(v)}) \end{aligned} \tag{3.15}$$

gegeben, d.h. $\widehat{\boldsymbol{v}}$ maximiert die vordere Hälfte des Terms in (3.8). Es läßt sich leicht zeigen, daß dies hier auf folgende Entscheidungsregel reduziert werden kann

$$\widehat{\boldsymbol{v}} = \begin{cases} (+1,+1) & \text{für} \quad h_1h_3 + h_2h_4 \geq 0 \\ (-1,-1) & \text{für} \quad h_1h_3 + h_2h_4 < 0. \end{cases}$$

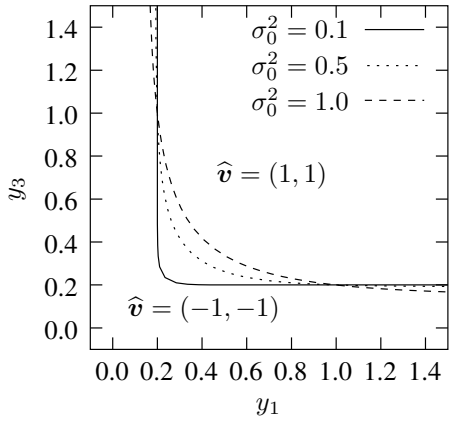

Abbildung 3.5.: Entscheidungsregionen bei wahrscheinlichkeitsbasierter Zuverlässigkeitsübergabe durch $h_k^{(v)}$ (s. Bsp. 3.1)

Die Unabhängigkeit der Entscheidung (3.8) von σ_0^2 gilt hier allerdings nicht mehr. Für $(y_2, y_4) = (-0.2, 1)$ sind in Abb. 3.5 die Entscheidungsregionen für $\widehat{\boldsymbol{v}} = (+1, +1)$ bzw. $\widehat{\boldsymbol{v}} = (-1, -1)$ in Abhängigkeit von y_1, y_3 für verschiedene σ_0^2 zu sehen, und da die Regionen punktsymmetrisch zum Ursprung liegen, ist hier nur der erste Quadrant gezeigt. Man erkennt, daß für kleine σ_0^2 die Entscheidungsregion für $\widehat{\boldsymbol{v}} = (1, 1)$ näherungsweise durch die Geraden $y_1 = 0.2$ und $y_3 = 0.2$ begrenzt wird und damit fast identisch mit der Entscheidungsregion ist, die durch die Regel

$$\widehat{\boldsymbol{v}} = \arg\min_{\boldsymbol{v}} \sum_{\{k: v_k \neq \widetilde{v}_k\}} w_k^{(v)} \tag{3.16}$$

mit $y_2^{(v)} = -0.2$ gegeben ist. Aufschlußreich ist der Vergleich für $\sigma_0^2 = 1$ und $\boldsymbol{y} = (0.3, -0.2, 0.3, 1)$. Das wahrscheinlichste Codewort $\boldsymbol{c} \in C$ für diese Werte ist $\boldsymbol{c}_{\text{ML}} = (1, 1, 1, 1)$ unabhängig von σ_0^2 und damit wäre $\widehat{\boldsymbol{v}} = (1, 1)$. Die auf Wahrscheinlichkeiten basierte Entscheidung gemäß (3.15) ergibt $\widehat{\boldsymbol{v}} = (-1, -1)$ während die auf euklidischer Distanz basierte Entscheidung gemäß Gl. (3.16) $\widehat{\boldsymbol{v}} = (1, 1)$ lautet. Hier wird deutlich, daß für uncodierte $\boldsymbol{u}$ die Entscheidung nach (3.16) gleich ist mit dem $\widehat{\boldsymbol{v}}$, das aus $\boldsymbol{c}_{\text{ML}}$ resultiert, während (3.15) abhängig von der Rauschvarianz das gesamtwahrscheinlichste $\widehat{\boldsymbol{v}}$ über alle Codeworte ergibt.

□

Die Wahrscheinlichkeitsdichten für $h_k^{(v)}$ lassen sich nur schwer bestimmen, da hier das *Produkt* zweier Zufallsvariablen gebildet wird. Die Verteilung von $y_k^{(v)}$ kann dagegen leicht bestimmt werden und ist gegeben durch (s. Anhang A.5)

$$f(y_k^{(v)}|v_k) = \frac{2}{\sqrt{2\pi\sigma_0^2}} \cdot \tag{3.17}$$
$$\left(\exp\left[-\frac{(y_k^{(v)} - v_k)^2}{2\sigma_0^2} \right] \cdot Q\left(\frac{|y_k^{(v)}| - 1}{\sigma_0} \right) + \exp\left[-\frac{(y_k^{(v)} + v_k)^2}{2\sigma_0^2} \right] \cdot Q\left(\frac{|y_k^{(v)}| + 1}{\sigma_0} \right) \right)$$

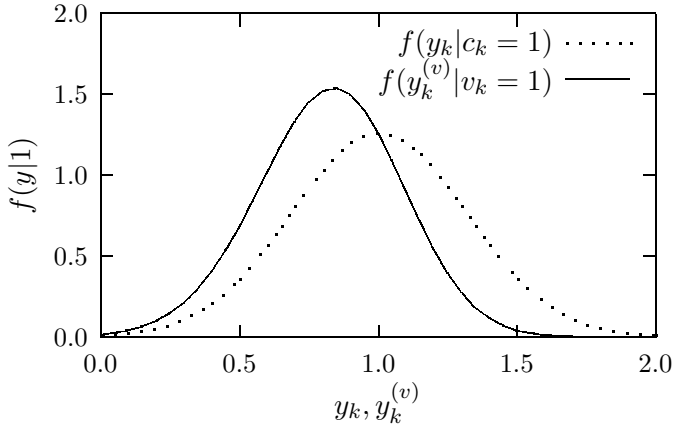

Abbildung 3.6.: Wahrscheinlichkeitsdichten für $\sigma_0^2 = 0.1$ (dies entspricht einem E_S/N_0 von etwa 7 dB)

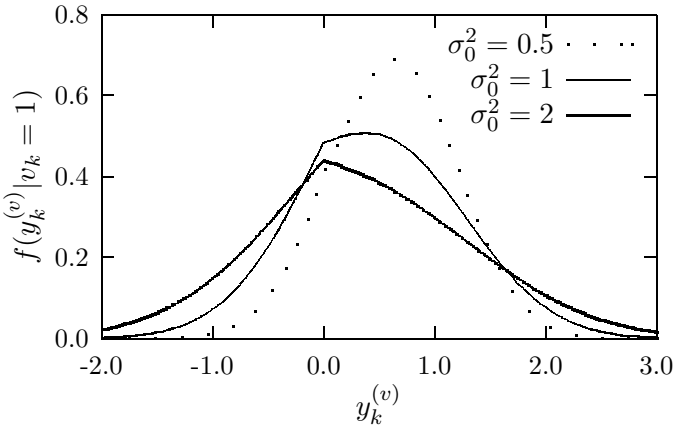

Abbildung 3.7.: Wahrscheinlichkeitsdichten $f(y_k^{(v)}|v_k = 1)$ für verschiedene σ_0^2 (die gewählten Werte entsprechen einem E_S/N_0 von etwa 0 dB, -3 dB und -6 dB)

In Bild 3.6 ist die Wahrscheinlichkeitsdichte $f(y_k^{(v)}|v_k = 1)$ für $\sigma_0^2 = 0.1$ gezeigt und man erkennt, daß diese näherungsweise gaußverteilt ist. Für größere Rauschvarianzen gilt dies allerdings nicht mehr, wie in Bild 3.7 zu sehen ist.

In die Wahrscheinlichkeitsdichte $f(y_k^{(u)}|u_k)$ fließt die Wahrscheinlichkeit ein, ob die Schätzung $\widehat{v}_k$ richtig ist oder nicht. Sie ist damit generell von der Fehlerwahrscheinlichkeit des Decoders für V abhängig. Falls davon ausgegangen wird, daß die Entscheidung $\widehat{v}_k$ richtig ist, ergibt sich allerdings eine einfache Verteilung. Für richtige Schätzung, d.h. $\widehat{v}_k = v_k$ ist nämlich $y_k^{(u)}$ das Ergebnis der Addition zweier gaußverteilter Größen y_k und $v_k \cdot y_{N+k}$ mit gleichem Mittelwert $E\{y_k\} = E\{v_k \cdot y_{N+k}\} = u_k$ und gleicher Varianz σ_0^2. Damit ist auch $y_k^{(u)}$ gaußverteilt und hat den Erwartungswert $E\{y_k^{(u)}\} = 2u_k$ und die Varianz $2\sigma_0^2$.

Kapazitiver Vergleich

Durch die Übergabe von Zuverlässigkeitswerten $h_k^{(v)}, h_k^{(u)}$ bzw. $y_k^{(v)}, y_k^{(u)}$ an die Decoder der äußeren Codes entstehen sog. *äquivalente Kanäle* für die Übertragung der Codeworte $\boldsymbol{v}$ und $\boldsymbol{u}$. Für diese Kanäle können in gleicher Weise, wie in Abschnitt 2.4 beschrieben, die Kanalkapazitäten $\mathcal{C}_h^{(v)}$, $\mathcal{C}_h^{(u)}$ bei wahrscheinlichkeitsbasierter Übergabe sowie $\mathcal{C}_y^{(v)}$, $\mathcal{C}_y^{(u)}$ bei distanzbasierter Übergabe als auch für beide Fälle die Cutoff-Raten $R_{\text{comp}}^{(v)}$ und $R_{\text{comp}}^{(u)}$ angegeben werden[6]. In [64] wurde gezeigt, daß bei Multilevel-Codes die Kanalkapazität mit Mehrstufendecodierung erreicht werden kann, d.h. daß für $N \to \infty$ eine beliebig kleine Wortfehlerwahrscheinlichkeit möglich ist, wenn die Raten der äußeren Codes entsprechend angepaßt gewählt werden. Wichtig ist hierbei, daß bei a-priori gegebenen Wahrscheinlichkeiten der Sendesymbole[7] die Summe der Kanalkapazitäten der äquivalenten Kanäle $\mathcal{C}^{(i)}$ für die äußeren Codes (bei richtiger 'verlustloser' Zuverlässigkeitsübergabe) der Gesamtkapazität $\mathcal{C}_0$ entspricht. Eine solche 'verlustlose' Übergabe ist per Definition durch die in Abschnitt 3.2.1 beschriebene, auf Wahrscheinlichkeiten basierte Zuverlässigkeitsübergabe gegeben, da mit den $h_k^{(v)}$ und $h_k^{(u)}$ direkt die wahrscheinlichsten Codewörter $\boldsymbol{v} \in V$ und $\boldsymbol{u} \in U$ gemäß Gl. (3.15) bestimmt werden können[8]. Falls die $|u|u+v|$-Konstruktion als Multilevel-Code betrachtet wird, entsprechen die Codeworte des inneren Codes $\boldsymbol{b} \in B$ einem QPSK-Symbol (s. Bild 3.2). Die Wahrscheinlichkeitsverteilung der Symbole $\boldsymbol{b}$, die die Transinformation maximiert, ist gegeben durch $P(\boldsymbol{b}) = 1/4\ \forall \boldsymbol{b} \in B$, und da diese durch gleichwahrscheinliche Codesymbole v_k und u_k erreicht werden kann, ist $\mathcal{C}_0$ durch die Kanalkapazität bei QPSK-Übertragung $\mathcal{C}_{\text{QPSK}}$ gegeben. Weiter gilt $\mathcal{C}_{\text{QPSK}} = 2 \cdot \mathcal{C}_{\text{BPSK}}$ (z.B. [19, Abschnitt 2.5]) und damit gilt für die Kanalkapazitäten der äquivalenten Kanäle $\mathcal{C}_h^{(v)}$

[6] Da die Wahrscheinlichkeitsdichten für die $y_k^{(v)}, y_k^{(u)}$ bekannt sind, können dazu direkt Gl. (2.9) und (2.10) ausgewertet werden. Bei den h_k handelt es sich um Wahrscheinlichkeiten, daher sind $\mathcal{C}_h^{(v)}$ und $\mathcal{C}_h^{(u)}$ direkt durch die Transinformationen $I(Y_k, Y_{N+k}; V_k)$ bzw. $I(Y_k, Y_{N+k}; U_k|V_k)$ gegeben (s. auch [64]).

[7] Im allgemeinen erfolgt die Berechnung der Kanalkapazität durch eine Maximierung der Transinformation über die Wahrscheinlichkeiten der Sendesymbole. Bei Multilevel-Codes kann diese Verteilung, die die Transinformation maximiert, nicht durch unabhängige Codierung der äußeren Codes erreicht werden. Aus Komplexitätsgründen ist aber gerade die Unabhängigkeit der Encoder für die äußeren Codes wünschenswert. Daher wird hier auf die Maximierung der Transinformation verzichtet und statt dessen die Kanalkapazität bei gegebener Symbolwahrscheinlichkeit, die aus der unabhängigen Codierung der äußeren Codes resultiert, betrachtet (s. auch [64]).

[8] Die Bestimmung des wahrscheinlichsten Codewortes $\boldsymbol{u} \in U$ erfolgt in gleicher Weise, indem in Gl. (3.15) $\boldsymbol{v}$ durch $\boldsymbol{u}$ ersetzt wird.

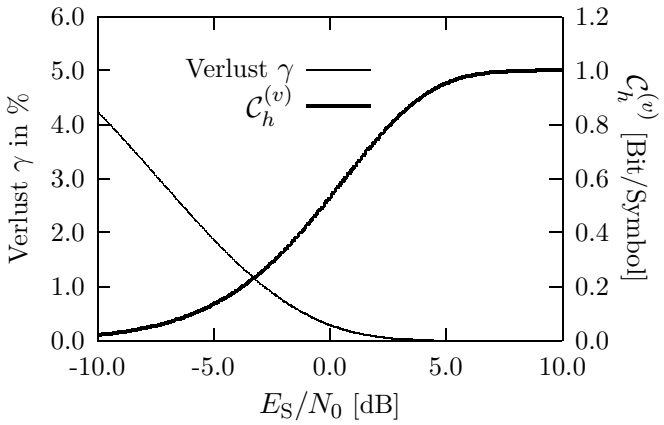

Abbildung 3.8.: Verlust γ (linke Skala) und $\mathcal{C}_h^{(v)}$ (rechte Skala) in Abhängigkeit von E_S/N_0

und $\mathcal{C}_h^{(u)}$ bei Zuverlässigkeitsübergabe durch h_k

$$\mathcal{C}_{\text{BPSK}} = \frac{1}{2}(\mathcal{C}_h^{(v)} + \mathcal{C}_h^{(u)}) \tag{3.18}$$

Die Übergabe der Zuverlässigkeiten mit Hilfe der $y_k^{(u)}$ ist ebenfalls optimal, da wie schon erwähnt aus den $y_k^{(u)}$ ohne Verlust die Zuverlässigkeitswerte $h_k^{(u)}$ berechnet werden können. Damit ergibt sich $\mathcal{C}_y^{(u)} = \mathcal{C}_h^{(u)}$ und da für das Verhältnis von Signal-zu-Rauschleistung für diesen äquivalenten Kanal gilt $\text{SNR}_u = 2{\cdot}\text{SNR}$, kann die Kapazität direkt zu $\mathcal{C}_y^{(u)} = \mathcal{C}_{\text{BPSK}}(2{\cdot}\text{SNR})$ bestimmt werden. Auf der anderen Seite handelt es sich bei der in Abschnitt 3.2.2 beschriebenen, auf dem quadratischen Abstand basierten Zuverlässigkeitsübergabe $y_k^{(v)}$ um ein suboptimales Verfahren, da sie nur eine Näherung für die $h_k^{(v)}$ sind. Es stellt sich allerdings die Frage, wie groß der Verlust bei der Übergabe der Zuverlässigkeit durch $y_k^{(v)}$ anstatt durch $h_k^{(v)}$ ist. Der Verlust γ an Kanalkapazität bei der Übergabe durch $y_k^{(v)}$ anstatt durch $h_k^{(v)}$

$$\mathcal{C}_y^{(v)} = (1-\gamma)\mathcal{C}_h^{(v)}$$

ist in Bild 3.8 gezeigt, wobei hier $\mathcal{C}_y^{(v)}$ ausgehend von der Wahrscheinlichkeitsdichte (3.17) numerisch bestimmt wurde und $\mathcal{C}_h^{(v)}$ aus Gl. (3.18) bestimmt werden kann. Für große SNR ist $\gamma \approx 0$ und daher gilt $\mathcal{C}_h^{(v)} \approx \mathcal{C}_y^{(v)}$, in Einklang mit der Feststellung aus Beispiel 3.1, daß für große SNR die Entscheidungsregionen gleich sind und daher beide Arten der Zuverlässigkeitsübergabe praktisch äquivalent. Zwar steigt für kleine E_S/N_0 der Verlust an, aber er bleibt im betrachteten Bereich in der Größe von wenigen Prozentpunkten und ist daher auch für kleinere SNR praktisch vernachlässigbar, zudem da $\mathcal{C}_h^{(v)}$ selbst sehr klein wird und daher die zulässige Coderate für V fast Null ist. Daher kann für den interessierenden Bereich geschrieben werden

$$\mathcal{C}_{\text{BPSK}} \approx \frac{1}{2}(\mathcal{C}_y^{(v)} + \mathcal{C}_y^{(u)}). \tag{3.19}$$

Wie im Anhang A.8 gezeigt wird, gilt allerdings bei Übertragung über den BSC Gl. (3.19) exakt. Es ist hier anzumerken, daß eine entsprechende Gleichung für die Cutoff-Raten R_{comp}, $R_{\text{comp}}^{(v)}$

und $R_{\text{comp}}^{(u)}$ nicht gilt. So wurde z.B. schon in [41] festgestellt, daß für Multilevel-Codes die Summe der Cutoff-Raten der äquivalenten Kanäle *größer* als R_{comp} sein kann.

Abschließend kann gesagt werden, daß beide Methoden der Zuverlässigkeitsübergabe praktisch identisch sind, da der 'mittlere Verlust an Information' durch die auf euklidischer Distanz basierten Werte gegenüber den auf Wahrscheinlichkeit basierten vernachlässigt werden kann. Der Rechenaufwand bei der Bestimmung der Zuverlässigkeitswerte ist allerdings bei den auf euklidischer Distanz basierten Werten $y_k^{(v)}$ deutlich geringer, es wird nur das Minimum aus zwei Zahlen gesucht und eine Vorzeichenkorrektur durchgeführt, wogegen zur Bestimmung der $h_k^{(v)}$ zwei reelle Zahlen miteinander multipliziert werden müssen.

3.2.4. Äquivalentes Signal-zu-Rauschverhältnis

Die äquivalenten Kanäle werden durch die Wahrscheinlichkeitsdichten $f(y_k^{(v)}|v_k)$ und $f(y_k^{(u)}|u, \widehat{v}_k = v_k)$ beschrieben. Es bietet sich aber an, für die beiden Kanäle jeweils ein *äquivalentes SNR* einzuführen, das anstelle der Wahrscheinlichkeitsdichten die Kanäle charakterisiert.

Wie schon erwähnt, sind für richtig geschätztes $\widehat{v}_k$ die Werte $y_k^{(u)}$ gaußverteilt mit $E\{y_k^{(u)}\} = 2u_k$ und Varianz $2\sigma_0^2$ und damit gilt für das SNR_u des äquivalenten Kanals zur Übertragung von $\boldsymbol{u}$

$$\text{SNR}_u = 2 \cdot \text{SNR}.$$

Da für große SNR auch $y_k^{(v)}$ näherungsweise gaußverteilt ist (s. Bild 3.6), kann auch hier ein äquivalentes SNR bestimmt werden. Zwar kann dies durch Bestimmung des Mittelwerts und der Varianz von $y_k^{(v)}$ geschehen, hier wird jedoch eine andere Vorgehensweise gewählt. Ausgangspunkt ist die Annahme, daß in Gl. (3.19) das Gleichheitszeichen gilt und gesucht ist nun das SNR_v, daß diese Gleichung erfüllt, wenn auch $y_k^{(v)}$ gaußverteilt ist. Damit ergibt sich die Forderung für SNR_v

$$\mathcal{C}_{\text{BPSK}}(\text{SNR}) \stackrel{!}{=} \frac{1}{2}(\mathcal{C}_{\text{BPSK}}(\text{SNR}_v) + \mathcal{C}_{\text{BPSK}}(\text{SNR}_u)). \tag{3.20}$$

Das so bestimmte SNR_v stellt sicher, daß die Gesamtkapazität erhalten bleibt. Für obige Gleichung kann keine geschlossene Lösung angegeben werden, sie muß nummerisch gelöst werden. Für große bzw. kleine SNR-Werte lassen sich allerdings Näherungen für SNR_v in geschlossener Form angeben. So gilt für $\text{SNR} \gg 1$ (s. Anhang A.7)

$$\text{SNR}_v \approx \text{SNR} - 2\ln 2 \tag{3.21}$$

bzw. für $\text{SNR} \ll 1$ (s. Anhang A.6)

$$\text{SNR}_v \approx \text{SNR}^2. \tag{3.22}$$

In gleicher Weise kann man $[E_S/N_0]_v$ als äquivalentes E_S/N_0 definieren. In Bild 3.9 ist $[E_S/N_0]_v$ mit den beiden obigen Näherungen gezeigt.

Auch wenn zur Charakterisierung des äquivalenten Kanals für die Übertragung von $\boldsymbol{v}$ der Wert $[E_S/N_0]_v$ so definiert wird, daß Gl. (3.20) erfüllt ist, bedeutet dies nicht zwangsläufig, daß

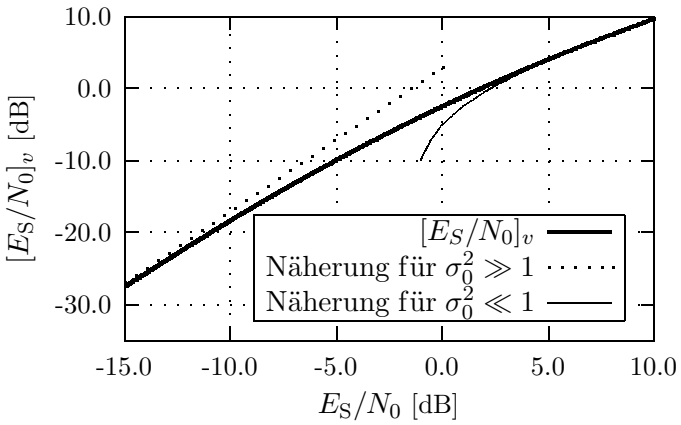

Abbildung 3.9.: Äquivalentes $[E_S/N_0]_v$ und Näherung für kleines und großes SNR gemäß Gl. (3.22) und (3.21)

damit auch alle anderen wichtigen Eigenschaften des Kanals richtig beschrieben werden. Eine neben der Kanalkapazität wichtige charakteristische Größe jedes Kanals ist die Cutoff-Rate. Eine Möglichkeit, die Näherung der wahren Verteilung von $y_k^{(v)}$ durch eine Gaußverteilung mit Varianz $1/(2 \cdot [E_S/N_0]_v)$ zu beurteilen, kann durch den Vergleich der tatsächlichen Cutoff-Rate $R_{\text{comp}}^{(v)}$ bei distanzbasierter Übergabe mit der geschätzten Cutoff-Rate $\widetilde{R}_{\text{comp},y}^{(v)}$ unter der Annahme gaußverteilter $y_k^{(v)}$ geschehen. Die exakte Cutoff-Rate $R_{\text{comp},y}^{(v)}$ kann mit Gl. (2.10) berechnet werden, für die Näherung $\widetilde{R}_{\text{comp},y}^{(v)}$ ergibt sich dagegen der Wert

$$\widetilde{R}_{\text{comp},y}^{(v)} = 1 - \log_2(1 + \mathrm{e}^{-[E_S/N_0]_v}).$$

Im folgenden wird der Fehler γ_R bei der Berechnung der Cutoff-Rate durch die Näherung

$$\widetilde{R}_{\text{comp},y}^{(v)} = (1 + \gamma_R) R_{\text{comp},y}^{(v)}$$

betrachtet, ähnlich wie der Verlust γ beim Vergleich der Kapazitäten. Der Verlauf von γ_R ist in Bild 3.10 zusammen mit dem exakten Wert der Cutoff-Rate gezeigt. Der Fehler liegt im interessierenden Bereich in der Größe von wenigen Prozentpunkten und ist für Cutoff-Raten > 0.2 praktisch zu vernachlässigen.

Zur besseren Beurteilung der Beschreibung der Kanaleigenschaften durch das äquivalente SNR könnte man noch den Fehlerexponenten [20] heranziehen und dessen exakten Verlauf mit dem sich aus der Gaußverteilung ergebenden Wert vergleichen. Da aber der lineare Bereich des Fehlerexponenten durch die Cutoff-Rate bestimmt ist und ferner der Schnittpunkt mit der Ratenachse durch die Kanalkapazität gegeben ist, welche per Definition durch die Wahl von $[E_S/N_0]_v$ erhalten bleibt, werden sich hier keine wesentlich anderen Erkenntnisse ergeben. Auch hier werden die Unterschiede zwischen exaktem Verlauf und dem durch das äquivalente SNR genäherten Fehlerexponenten nur gering sein.

Damit kann abschließend festgestellt werden, daß die Charakterisierung der äquivalenten Kanäle für die äußeren Codes durch die äquivalenten SNRs eine geeignete Beschreibung sein kann. Wie noch später in Abschnitt 5.2.3 gezeigt wird, kann damit auch die Fehlerwahrscheinlichkeit bei der Decodierung der äußeren Codes mit ausreichender Genauigkeit bestimmt werden.

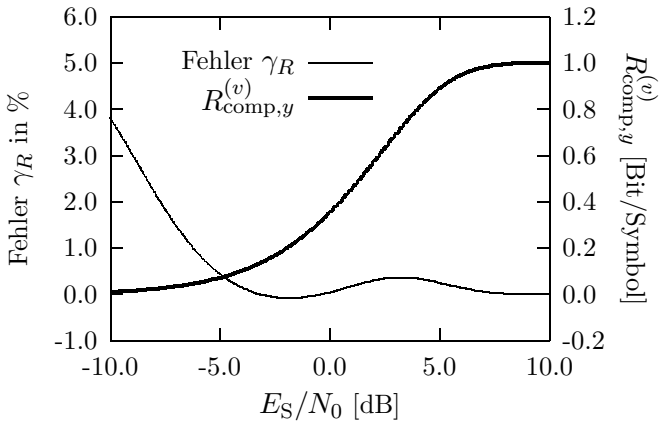

Abbildung 3.10.: Fehler bei Berechnung von $R^{(v)}_{\text{comp},y}$ durch Gl. (2.11) mit $[E_S/N_0]_v$ im Vergleich zur exakten Rechnung mit (2.10) und der Verteilungsdichte $f(y_k^{(v)}|v_k)$

3.3. Codekonstruktionen

Die $|u|u+v|$-Konstruktion ist, wie schon beschrieben, eine spezielle Form der Codeverkettung. Als äußere Codes U, V können alle binären Codes verwendet werden. So wurden z.B. in [40] binäre Faltungscodes als äußere Codes gewählt, es können aber auch alle binären Blockcodes wie z.B. BCH-Codes verwendet werden. Ein besonderer Fall, auf den in Kapitel 4 noch näher eingegangen wird, ist die Verwendung von binären Reed-Muller Codes als äußere Codes, da diese selbst durch mehrfache $|u|u+v|$-Verkettung konstruiert werden können. Unabhängig vom Decodierverfahren ist es aber aufgrund von Gl. (3.1) generell sinnvoll, die Distanz des Codes V größer als die Distanz von U zu wählen, damit die Distanz des Gesamtcodes nicht zu klein ist.

Es ist natürlich möglich, daß die äußeren Codes U und V selbst binäre GC-Codes sind. Falls nun die inneren Codes beider Codes auch als $|u|u+v|$-Konstruktion betrachtet werden können, liegt der Fall einer zweifachen $|u|u+v|$-Verkettung vor.

3.3.1. Rekursive PLOTKIN-Konstruktion

Wie oben angesprochen ist es möglich, auch die Codes U und V durch die $|u|u+v|$-Verkettung jeweils zweier äußerer Codes zu konstruieren, und auch diese äußeren Codes können ebenfalls durch die $|u|u+v|$-Verkettung konstruiert werden. Wenn diese Kette weiter entwickelt wird, kommt man damit zur rekursiven $|u|u+v|$-Konstruktion. In Abb. 3.11 ist schematisch eine 3-stufige $|u|u+v|$-Verkettung mit den Symbolen der Codeworte $\boldsymbol{c}^{(i)} = (c_1^{(i)}, c_2^{(i)}, \ldots, c_N^{(i)})$, $i \in \{1, \ldots, 8\}$ der acht äußeren Codes $C^{(1)}, \ldots, C^{(8)}$ gezeigt. Jeder Block entspricht hier einem Encoder, durch den die beiden Eingangssymbolfolgen $(u_1, u_2, \ldots)$ und $(v_1, v_2, \ldots)$ auf die Ausgangsfolge $(u_1, u_2, \ldots, u_1+v_1, u_2+v_2, \ldots)$ abgebildet werden. Mit jeder Stufe verdoppelt sich dadurch die Codelänge, so daß der Gesamtcode die Länge $8N$ hat.

Das Gesamtsystem kann auch als GC-Code mit 8 äußeren Codes und einem inneren Code der

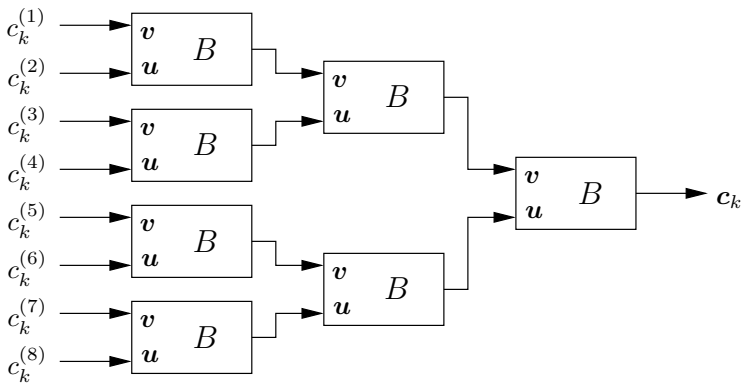

Abbildung 3.11.: Schematische Darstellung einer 3-stufigen rekursiven $|u|u+v|$-Verkettung

Länge 8 mit der Generatormatrix

$$\boldsymbol{B}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

aufgefaßt werden. Die Codeworte des inneren Codes $\boldsymbol{c}_k = (c_k, c_{N+k}, \ldots, c_{7N+k})$ sind durch die Vorschrift $\boldsymbol{c}_k = (c_k^{(1)}, \ldots, c_k^{(8)}) \cdot \boldsymbol{B}_8$ bestimmt, und die Codeworte $\boldsymbol{c} \in C$ können in der gleichen Weise wie in Abschnitt 3.1 beschrieben durch spaltenweises Auslesen der Matrix $\boldsymbol{C} = \boldsymbol{M}\boldsymbol{B}_8$ gebildet werden, wobei hier in der i-ten Spalte von $\boldsymbol{M}$ ein Codewort des Codes $C^{(i)}$ steht. Allgemein ergibt sich bei einer m-stufigen Verkettung die Codelänge des inneren Codes und Anzahl der äußeren Codes zu 2^m, die Gesamtcodelänge ist $2^m N$. Es läßt sich leicht zeigen, daß die Generatormatrix des inneren Codes $\boldsymbol{B}_{2^{m+1}}$ bei $(m+1)$-stufiger Verkettung aus der Generatormatrix $\boldsymbol{B}_{2^m}$ des durch m-stufige Verkettung konstruierten inneren Codes B_{2^m} rekursiv gebildet werden kann

$$\boldsymbol{B}_{2^{m+1}} = \begin{bmatrix} \mathbf{1}_{2^m} & \boldsymbol{B}_{2^m} \\ \boldsymbol{B}_{2^m} & \boldsymbol{B}_{2^m} \end{bmatrix},$$

$\mathbf{1}_{2^m}$ bezeichnet hier eine $2^m \times 2^m$-Matrix, deren Elemente alle 1 sind.

Die 2^m äußeren Codes sind nun wieder beliebig wählbar. Da allerdings durch die multiple Verkettung für große m die Codelänge schnell anwächst, bieten sich kurze äußere Codes an, zudem diese auch einfach decodiert werden können. Im einfachsten Fall, für $N = 1$, stehen damit für die $C^{(i)}$ nur zwei verschiedene Codes zur Verfügung. Dies sind der $(1, 1, 1)$-Code mit den beiden Codeworten $\{(1), (-1)\}$ und der $(1, 0, \infty)$-Code, der nur eines der beiden Codeworte (1) oder (-1) enthält. Die Dimension K des Gesamtcodes ergibt sich dann aus der Anzahl der Codes $C^{(i)}$ mit Dimension $K^{(i)} = 1$.

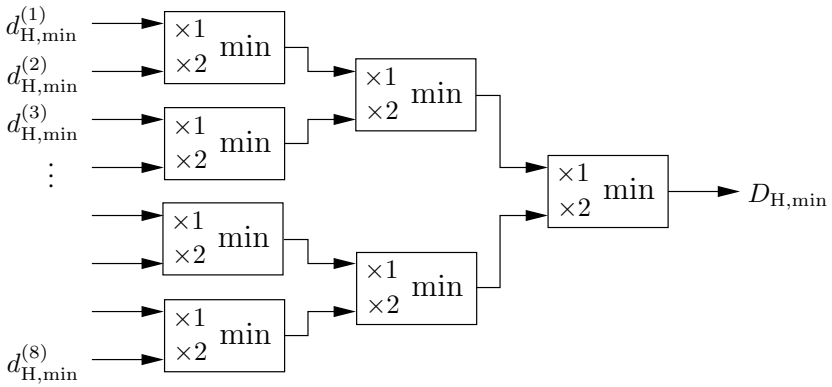

Abbildung 3.12.: Schematische Darstellung der Bestimmung von $D_{\mathrm{H,min}}$ bei 3-stufiger Verkettung

3.3.2. Konstruktion für maximale Mindestdistanz

Bei fester Wahl von m hängen die Eigenschaften des Codes natürlich in entscheidender Weise von der Wahl der äußeren Codes ab. Das 'klassische' Kriterium für die Auswahl der Codes $C^{(i)}$ ist die sich ergebende Mindest-Hammingdistanz $D_{\mathrm{H,min}}$ des verketteten Codes, da für große SNR die Fehlerwahrscheinlichkeit in erster Linie durch die Mindestdistanz des Codes bestimmt wird. Die Berechnung von $D_{\mathrm{H,min}}$ aus den Distanzen $d^{(i)}_{\mathrm{H,min}}$ der äußeren Codes läßt sich ähnlich wie Codierung anhand eines Blockschaltbildes verdeutlichen. Dies ist in Abb. 3.12 exemplarisch für die 3-stufige Verkettung gezeigt. Hierbei wird in jedem Block das Minimum aus den mit dem entsprechenden Faktor gewichteten Eingangsdistanzen an den Ausgang gegeben. Für dieses Beispiel ergibt sich $D_{\mathrm{H,min}}$ zu

$$D_{\mathrm{H,min}} = \min(d^{(1)}_{\mathrm{H,min}}, 2 \cdot d^{(2)}_{\mathrm{H,min}}, 2 \cdot d^{(3)}_{\mathrm{H,min}}, 4 \cdot d^{(4)}_{\mathrm{H,min}}, \ldots, 8 \cdot d^{(8)}_{\mathrm{H,min}}).$$

Falls die Codelänge der äußeren Codes wie oben angegeben $N = 1$ gewählt und die Dimension K des Gesamtcodes vorgegeben wird, müssen K der 2^m äußeren Codes die Dimension $K^{(i)} = 1$ haben, für die anderen $(2^m - K)$ Codes muß $K^{(i)} = 0$ gelten. Als Codiervorschrift, d.h. die Zuordnung zwischen Information und Codewort der äußeren Codes, definiert man sinnvollerweise $\boldsymbol{c}^{(i)} = \boldsymbol{i}^{(i)}$, damit enthält dann der Vektor $(c^{(1)}_1, c^{(2)}_1, \ldots, c^{(2^m)}_1)$ direkt die zu codierenden Informationssymbole $(i_1, \ldots, i_K)$ an den Positionen i mit $K^{(i)} = 1$. Durch rekursives Anwenden von Gl. (3.1) läßt sich leicht zeigen, daß für $D_{\mathrm{H,min}}$ gilt

$$D_{\mathrm{H,min}} = \min_{\{i:K^{(i)}=1\}} w_{\mathrm{H}}(\boldsymbol{B}_{2^m,i}),$$

wobei $\boldsymbol{B}_{2^m,i}$ die i-te Zeile von $\boldsymbol{B}_{2^m}$ bezeichnet. Das Gewicht der i-ten Zeile $\boldsymbol{B}_{2^m,i}$ kann ebenfalls durch die schematische Darstellung 3.12 ermittelt werden. Es gilt $w_{\mathrm{H}}(\boldsymbol{B}_{2^m,i}) = 2^{m-\rho}$ wobei $(m - \rho)$ gleich der Anzahl der Blöcke ist, die die Distanz $d^{(i)}_{\mathrm{H,min}}$ verdoppeln (d.h. bei denen auf dem Weg von $d^{(i)}_{\mathrm{H,min}}$ zu $D_{\mathrm{H,min}}$ die Distanz mit 2 multipliziert wird). Die Anzahl der Zeilen in $\boldsymbol{B}_{2^m}$ mit Gewicht $2^{m-\rho}$ ist damit $\binom{m}{m-\rho}$, da für eine Zeile mit Gewicht $2^{m-\rho}$ aus m Blöcken genau $(m - \rho)$ Stück ausgewählt werden müssen, durch die die Distanz verdoppelt

wird[9]. Um die Distanz des Gesamtcodes bei gegebenem K zu maximieren, muß die Dimension genau der Codes $C^{(i)}$ zu $K^{(i)} = 1$ gesetzt werden, deren zugehörige Zeilenvektoren $\boldsymbol{B}_{2^m,i}$ zu den K Zeilen mit größtem Gewicht gehören.

Die Distanz des Gesamtcodes ergibt sich dann zu $D_{\text{H,min}} = 2^{m-r}$ mit dem kleinsten $r \in [0, m]$, das die Gleichung

$$K \leq \sum_{\rho=0}^{r} \begin{pmatrix} m \\ m-\rho \end{pmatrix}$$

erfüllt. Obige Konstruktion schließt die Klasse der binären REED-MULLER Codes mit ein, denn für $K = \sum_{\rho=0}^{r} \binom{m}{m-\rho}$ werden damit alle Positionen i, deren zugehöriger Zeilenvektor $\boldsymbol{B}_{2^m,i}$ ein Gewicht größer oder gleich 2^{m-r} besitzt, zu Informationsstellen. Wenn dann alle anderen Stellen $c_1^{(i)} = 1$ gesetzt werden, ergibt sich als Gesamtcode der REED-MULLER Code $\mathrm{RM}(r, m)$. Im Fall der 3-stufigen Verkettung mit z.B. $K = 4$ erhält man durch die Wahl $K^{(i)} = 1$ für $i \in \{4, 6, 7, 8\}$ damit den $(8, 4, 4)$-Code $\mathrm{RM}(1, 3)$. Der dazugehörige Eingangsvektor, gebildet aus den Codesymbolen der äußeren Code liest sich dann

$$(c_1^{(1)}, c_1^{(2)}, \ldots, c_1^{(8)}) = (1, 1, 1, i_1, 1, i_2, i_3, i_4)$$

wobei $c_1^{(i)} = 1$ für $i \in \{1, 2, 3, 5\}$ gewählt wurde.

Generell hat der nach diesem 'klassischen' Prinzip konstruierte Code bei gegebenem N und K das größte Korrekturpotential[10]. Es wird sich jedoch noch zeigen (s. Abschn. 5.3.4 bzw. Kapitel 6), daß gerade bei langen Codes dieses Potential nicht genutzt werden kann und daher andere Kriterien bei der Bestimmung der Codes mit $K^{(i)} = 1$ geeigneter sind.

[9] Die Anzahl der Zeilen mit Gewicht $2^{m-\rho}$ kann natürlich auch, wie in der Literatur über Reed-Muller-Codes vielfach gezeigt, auf andere Art wie z.B.mit Hilfe Boolscher Funktionen oder Euklidischer Geometrien bestimmt werden. [67, Chapter 13][43, Chapter 16]

[10] Für $K < \sum_{\rho=0}^{r} \binom{m}{m-\rho}$ ergeben sich zwar bei gleicher Mindestdistanz verschiedene Möglichkeiten, die Codes mit $K^{(i)} = 1$ zu wählen, eine Suche wie z.B. in [60] nach dem besten $(64, 40, 8)$-Subcode des RM-Code mit Parametern $(64, 42, 8)$ ergab jedoch, daß alle dort betrachteten Subcodes bei ML-Decodierung fast indentische Korrekturfähigkeit besitzen.

4. REED-MULLER Codes

REED-MULLER (RM) Codes gehören zu den frühen Codes die schon in den ersten sechs Jahren nach SHANNONs Artikel über die Theorie der Nachrichtenübertragung veröffentlicht wurden. Von MULLER wurden sie 1954 als Klasse von Codes zur Fehlerentdeckung veröffentlicht [36], REED gab im gleichen Jahr einen einfachen Decodieralgorithmus an [47]. Eine ausführliche Beschreibung dieser Codeklasse kann in [67, Kapitel 13] gefunden werden, dieser Abschnitt wiederholt einige der bekannten Eigenschaften.

4.1. Definition und Eigenschaften

Eine der einfachsten Definitionen von RM-Codes beruht auf boolschen Funktionen[1] mit m Variablen.

Definition 4.1 Es sei $\boldsymbol{x} = (x_m, \ldots, x_1)^T$ ein Vektor mit den Elementen[2] $x_i \in \mathrm{GF}(2) = \{\pm 1\}$ und $F_{r,m}$ die Menge aller boolschen Funktionen $f(x_1, x_2, \ldots, x_m)$ in m Variablen vom Grad $\leq r$. Weiter sei $\boldsymbol{c} = (c_1, c_2, \ldots c_N)$ der Vektor der Länge $N = 2^m$ mit den Funktionswerten von $f(\boldsymbol{x})$ über alle verschiedenen $\boldsymbol{x} \in \{\pm 1\}^m$, d.h. $c_k = f(\boldsymbol{x}_k)$, $k = 1, 2, \ldots, 2^m$. Dann ist der **REED-MULLER Code** der Ordnung r und Länge 2^m gegeben durch

$$\mathrm{RM}(r, m) = \{\boldsymbol{c} : f(\boldsymbol{x}) \in F_{r,m}\}$$

□

Allgemein beschreibt die Funktion $f(\boldsymbol{x})$ eine Abbildung des Raumes $\mathrm{GF}(2)^m$ in den Raum $\mathrm{GF}(2)$, der Wertebereich umfaßt zwei Elemente. Die Elemente aus $\mathrm{GF}(2)^m$ werden hierdurch in zwei Mengen X und $\overline{X}$ aufgeteilt

$$\begin{aligned} X &= \{\boldsymbol{x} : f(\boldsymbol{x}) = -1\} \\ \overline{X} &= \{\boldsymbol{x} : f(\boldsymbol{x}) = 1\} \end{aligned}$$

und man bezeichnet $f(\boldsymbol{x})$ als *Charakteristische Funktion* der Menge X. Der Vektor $\boldsymbol{c}$ ist damit der *Inzidenzvektor* der Menge X, d.h. es gilt $c_k = -1$ genau dann, wenn $\boldsymbol{x}_k \in X$. Diese

[1] Als boolsche Funktion wird hier ein Polynom

$$f(\boldsymbol{x}) = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_1 x_2 + \ldots + b_{2^m-1} x_1 x_2 \cdots x_m$$

mit $f(\boldsymbol{x}), b_i, x_i \in \mathrm{GF}(2)$ bezeichnet.

[2] Auch hier werden die Elemente aus GF(2) mit {+1,-1} bezeichnet, mit +1 als dem Nullelement (s. auch Fußnote 1 auf Seite 5).

Betrachtung führt auf die Verwandtschaft von RM-Codes mit euklidischen Geometrien (EG). Ein wesentliches Ergebnis dieser Betrachtung ist folgender Satz:

Satz 4.1 Die Codeworte $\boldsymbol{c} \in \mathrm{RM}(r,m)$ mit kleinstem Gewicht sind die Inzidenzvektoren der Flächen von Dimension $(m-r)$ in $\mathrm{EG}(m,2)$, d.h. in der euklidischen Geometrie der Dimension m über $\mathrm{GF}(2)$. □

Daraus ergibt sich weiter

Satz 4.2 Für die **Mindestdistanz** eines RM-Codes mit Parametern r und m gilt

$$d_{\mathrm{H,min}} = 2^{m-r}.$$

□

Im allgemeinen ist die Nummerierung der $\boldsymbol{x}_k$ und daher die Reihenfolge der c_k nicht vorgegeben, in vielen Fällen werden allerdings die $\boldsymbol{x}_k$ mit der binären Darstellung von $(k-1)$ verknüpft, d.h. z.B. für $(k-1) = 4$ oder in binärer Schreibweise $(k-1) = \ldots 0100$ ergibt sich mit der Zuordnung $0 \rightarrow 1$ und $1 \rightarrow -1$ der Vektor $\boldsymbol{x}_5 = (\ldots, 1, -1, 1, 1)^T$. Umgekehrt gibt damit die binäre Schreibweise der $\boldsymbol{x}_k = (x_m, \ldots, x_2, x_1)^T$ die Reihenfolge vor.

Beispiel 4.1 Für $m = 3$ und $r = 1$ ist die Menge $F_{1,3}$ gegeben durch

$$F_{1,3} = \{b_0 + b_1x_1 + b_2x_2 + b_3x_3 | b_i \in \{1, -1\}\}.$$

Da die b_i beliebig aus $\{1, -1\}$ gewählt werden können, enthält $F_{1,3}$ $2^4 = 16$ verschiedene Polynome und damit ist der Code $\mathrm{RM}(1,3)$ ein $(8,4,4)$-Code. Die Codiervorschrift aus Def. 4.1 kann auch in Vektor-Matrix-Schreibweise angegeben werden, und falls die Reihenfolge der $\boldsymbol{x}_k$ wie oben beschrieben durch die binäre Darstellung vorgegeben ist, liest sich dies

$$\boldsymbol{c} = (b_0, b_3, b_2, b_1) \cdot \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}.$$

Dadurch ergibt sich automatisch die Generatormatrix $\boldsymbol{G}$, wenn die b_i als Information aufgefaßt werden. Vergleicht man diese Generatormatrix mit der Matrix $\boldsymbol{B}_8$ aus Abschnitt 3.3.1, erkennt man sofort, daß $\boldsymbol{G}$ genau die Zeilen aus $\boldsymbol{B}_8$ enthält, deren Gewicht größer oder gleich 4 ist. □

Aus der Verallgemeinerung dieses Beispiels folgt automatisch folgender Satz (vergl. Abschn. 3.3.2):

Satz 4.3 Die **Dimension** K eines RM-Codes mit Parametern r und m ist gegeben durch

$$K = \sum_{\rho=0}^{r} \binom{m}{\rho}.$$

□

Eine für diese Arbeit wichtige Eigenschaft der REED-MULLER Codes ist die Tatsache, daß sie rekursiv mit Hilfe der $|u|u+v|$-Verkettung konstruiert werden können. Diese Eigenschaft kann direkt aus der Definition der RM-Codes mit Hilfe boolscher Funktionen hergeleitet werden. So gilt für jede boolsche Funktion $f(\boldsymbol{x})$ mit $\boldsymbol{x} = (x_m, x_{m-1}, \ldots, x_1) \in \mathrm{GF}(2)^m$ vom maximalen Grad r, da die Variable x_m bei dem Teil der Summanden, die x_m als Faktor enthalten, ausgeklammert werden kann

$$f(\boldsymbol{x}) = x_m \cdot f'(\boldsymbol{x}') + f''(\boldsymbol{x}')$$

mit $\boldsymbol{x}' \in \mathrm{GF}(2)^{m-1}$. Die beiden Funktionen $f'(\boldsymbol{x}')$ und $f''(\boldsymbol{x}')$ haben maximal Grad $(r-1)$ bzw. Grad r. Damit können REED-MULLER Codes auch wie folgt definiert werden

$$\mathrm{RM}(r,m) = \{\boldsymbol{c} : f(\boldsymbol{x}) = x_m \cdot f'(\boldsymbol{x}') + f''(\boldsymbol{x}'), f'(\boldsymbol{x}') \in F_{r-1,m-1}, f''(\boldsymbol{x}') \in F_{r,m-1}\},$$

und da die Mengen $F_{r-1,m-1}$ und $F_{r,m-1}$ die Codes $\mathrm{RM}(r-1,m-1)$ bzw. $\mathrm{RM}(r,m-1)$ generieren, gilt in gleicher Weise aufgrund

$$\begin{aligned} f(\boldsymbol{x}) &= x_m \cdot f'(\boldsymbol{x}') + \underbrace{[x_m + (x_m + (-1))]}_{(-1)} \cdot f''(\boldsymbol{x}') \\ &= x_m \cdot [f'(\boldsymbol{x}') + f''(\boldsymbol{x}')] + (x_m + (-1)) \cdot f''(\boldsymbol{x}') \end{aligned}$$

die Definition

$$\mathrm{RM}(r,m) = \{\boldsymbol{c} : \boldsymbol{c} = |\boldsymbol{c}''|\boldsymbol{c}'' + \boldsymbol{c}'|, \boldsymbol{c}' \in \mathrm{RM}(r-1,m-1), \boldsymbol{c}'' \in \mathrm{RM}(r,m-1)\},$$

falls die Nummerierung der $\boldsymbol{x}_k$, wie oben beschrieben, mit der binären Darstellung verknüpft ist. Es können damit alle RM-Codes aus RM-Codes der halben Länge als äußeren Codes und der $|u|u+v|$-Verkettung generiert werden, und für die Generatormatrix von $\mathrm{RM}(r,m)$ gilt in gleicher Weise (vergl. Abschn. 3.3.1)

$$\boldsymbol{G}_{r,m} = \begin{bmatrix} \boldsymbol{G}_{r,m-1} & \boldsymbol{G}_{r,m-1} \\ \boldsymbol{1}_{2^{m-1}} & \boldsymbol{G}_{r-1,m-1} \end{bmatrix}$$

wobei auch hier $\boldsymbol{1}_{2^{m-1}}$ eine $(2^{m-1} \times 2^{m-1})$-Matrix bezeichnet, deren Elemente alle 1 sind.

Obwohl sich RM-Codes sehr einfach beschreiben lassen, ist die Gewichtsverteilung A_w, d.h. die Anzahl der Codeworte mit Gewicht w, nur bei wenigen Codes vollständig bekannt. Für $r = 0, 1, 2$ sowie zu den dazu dualen Codes $r = m-1, m-2, m-3$ ist die Gewichtsverteilung bekannt ($r = 0, 1$ trivial, $r = 2$ s. [53]). Weiterhin bekannt ist die Gewichtsverteilung der Codes $\mathrm{RM}(3,7)$ [58], $\mathrm{RM}(3,8)$ und $\mathrm{RM}(4,8)$ [31], [48], $\mathrm{RM}(3,9)$ und $\mathrm{RM}(5,9)$ [59]. Für alle weiteren Codes kann zumindest die Anzahl der Codeworte mit Gewicht $w < 2d_{\mathrm{H,min}}(C)$ angegeben werden [30].

4.2. Permutationen

In Kapitel 5 wird ein Algorithmus vorgestellt, der verschiedene Permutationen des Empfangsvektors betrachtet. Eine Permutation Π wird allgemein durch eine Zuordnungsvorschrift $\Pi(k)$

mit $\Pi(k) \neq \Pi(j)$ für $k \neq j$ festgelegt. Der sich aus dem Vektor $\boldsymbol{y} = (y_1, y_2, \ldots, y_N)$ durch die Permutation Π ergebende Vektor $\boldsymbol{y}^{(p)}$ ist definiert als

$$\boldsymbol{y}^{(p)} = (y_{\Pi(1)}, y_{\Pi(2)}, \ldots, y_{\Pi(N)}).$$

In der gleichen Weise können die Codeworte $\boldsymbol{c} \in \mathrm{RM}(r, m)$ permutiert werden. Von besonderem Interesse sind allerdings diejenigen Permutationen, durch die alle Codewörter des Codes in andere Codewörter des gleichen Codes überführt werden. Die Gruppe der Permutationen mit dieser Eigenschaft wird auch als Automorphismus-Gruppe bezeichnet. Gemäß Def. 4.1 werden die Codesymbole eines RM-Codes durch die Vorschrift $c_k = f(\boldsymbol{x}_k)$, $\boldsymbol{x}_k \in \mathrm{GF}(2)^m$ mit einem Polynom f gebildet. Für eine beliebige aber feste Ordnung der Elemente $\boldsymbol{x}_k$ können mit Hilfe der Gruppe der affinen Transformationen $\mathrm{GA}(m)$ verschiedene Permutationen angegeben werden. Es sei

$$\boldsymbol{x}_k = \boldsymbol{A} \cdot \boldsymbol{x}_j + \boldsymbol{b}$$

mit einer nichtsingulären $(m \times m)$-Matrix $\boldsymbol{A}$ mit Elementen $A_{ij} \in \mathrm{GF}(2)$ und einem Vektor $\boldsymbol{b} \in \mathrm{GF}(2)^m$. Mit $f \in F_{r,m}$ und $\boldsymbol{A}^{-1}$ als Inverser von $\boldsymbol{A}$ ist, wie sich leicht zeigen läßt, auch $\boldsymbol{c}^{(p)}$ mit

$$c_k^{(p)} = c_{\Pi(k)} = f(\boldsymbol{A}^{-1} \cdot (\boldsymbol{x}_k - \boldsymbol{b})) = f(\boldsymbol{x}_j)$$

ein Codewort von $\mathrm{RM}(r, m)$, denn der Grad von f vergrößert sich durch die Substitution $\boldsymbol{x}_k \rightarrow (\boldsymbol{A}^{-1} \cdot (\boldsymbol{x}_k - \boldsymbol{b}))$ nicht. Durch Wahl von $\boldsymbol{A}$ und $\boldsymbol{b}$ ist nun die Permutation Π wie folgt festgelegt

$$\Pi(k) = j \quad \text{falls} \quad \boldsymbol{x}_k = \boldsymbol{A} \cdot \boldsymbol{x}_j + \boldsymbol{b}.$$

Für RM-Codes mit $1 \leq r \leq m - 2$ ist die Automorphismus-Gruppe, d.h. die Gruppe aller Permutationen, durch die die Codeworte auf andere Codeworte abgebildet werden, durch $\mathrm{GA}(m)$ gegeben [67, Chapter 13]. Für die Mächtigkeit von $\mathrm{GA}(m)$ gilt

$$|\mathrm{GA}(m)| = 2^m \cdot (2^m - 2^0)(2^m - 2^1) \cdots (2^m - 2^{m-1}).$$

Eine Untermenge der affinen Transformationen ist durch die Gruppe der linearen Transformationen $\mathrm{GL}(m)$, die sich aus $\mathrm{GA}(m)$ mit[3] $\boldsymbol{b} = \boldsymbol{1}$ ergeben. Alle sich daraus ergebenden Permutationen haben die Eigenschaft, daß diejenige Komponente c_κ mit $\boldsymbol{x}_\kappa = \boldsymbol{1}$ nicht permutiert wird. Falls nun die Nummerierung der $\boldsymbol{x}_k$ durch die binäre Schreibweise festgelegt ist, gilt damit für alle Permutationen, die durch $\mathrm{GL}(m)$ gebildet werden, $c_1^{(p)} = c_1$.

Bezüglich der Mächtigkeit von $\mathrm{GL}(m)$ kann man leicht zeigen

$$|\mathrm{GL}(m)| = (2^m - 2^0)(2^m - 2^1) \cdots (2^m - 2^{m-1}).$$

[3] Auch hier ist das Nullement bezüglich der Addition durch den Vektor $(1, 1, \ldots, 1)$ gegeben.

5. Decodieralgorithmen

In diesem Kapitel werden die in dieser Arbeit verwendeten Decodieralgorithmen vorgestellt. Die betrachteten Verfahren der Mehrstufen- und der sequentiellen Decodierung sind im wesentlichen nicht neu, sie wurden schon bei der Decodierung anderer Codes angewendet. Allerdings sind bei der hier betrachteten Codeklasse bisher fast ausschließlich die unter Abschnitt 5.2.1 und 5.2.2 aufgeführten, auf Mehrheitsentscheidung basierten Algorithmen angewendet worden, und dies nur für den Spezialfall von REED-MULLER Codes [9], [49]. Eine Abwandlung des in Abschnitt 5.3 vorgestellten sequentiellen Algorithmus wurde schon von LUCAS, BOSSERT und DAMMANN [33] für die Decodierung von REED-MULLER Codes verwendet. Die dort vorgeschlagene Version ist aber besonders bei langen Codes sehr ineffizient und besitzt eine große Decodierkomplexität, worauf in Abschnitt 5.3.3 noch näher eingegangen wird.

Dieses Kapitel beginnt mit der Vorstellung des klassischen REED-Algorithmus und der von KABATYANSKY bzw. von SCHNABL und BOSSERT vorgestellten rekursiven Decodierung von REED-MULLER Codes. Es wird gezeigt, daß die Wortfehlerwahrscheinlichkeit dieses Algorithmus mit Hilfe des im vorigen Kapitel eingeführten 'äquivalenten SNR' sehr exakt geschätzt werden kann. Darauf folgt die Erweiterung des Algorithmus zum sequentiellen bzw. Listendecoder. Es wird das strukturelle Problem des Zusammenspiels zwischen RM-Code und Listendecodierung gezeigt, was dann zum Abschluß dieses Kapitels zur Permutationsdecodierung führt, die für den Code RM$(4, 9)$, zumindest für den BSC, Wortfehlerraten nahe der des ML-Decoders ermöglicht.

5.1. Problem der Decodierung

Generell besteht die Aufgabe des Decoders, wie schon im Kapitel 2 erläutert, darin, das gesendete bzw. gespeicherte Codewort zu schätzen. Der ML-Decoder kann dazu die Entscheidungsregel (2.5) verwenden, die bei gleichwahrscheinlichen Codeworten optimal im Sinne der minimalen Wortfehlerwahrscheinlichkeit ist. Eine Methode der ML-Decodierung ist der VITERBI-Algorithmus [15], wozu allerdings ein Trellis für den Code angegeben werden muß. Für Blockcodes kann dieses Trellis z.B. ausgehend von der Prüfmatrix aufgestellt werden [68]. Speziell bei REED-MULLER Codes kann die Tatsache, daß sie mit Hilfe der $|u|u+v|$-Verkettung konstruiert werden können und die sich dabei ergebenden Teilcodes U und $U+V$ identisch sind, zur Aufstellung eines Trellis genutzt werden (sog. 'Squaring Construction' [16]), wobei sich allerdings parallele Übergänge zwischen den Zuständen ergeben. Ein ähnliches Verfahren zur Aufstellung des Trellis von Blockcodes wurde in [18] vorgestellt. Oft ist jedoch die ML-Decodierung zu aufwendig und zu komplex, da sich gerade gute Codes mit einem großem Korrekturpotential durch ein großes Trellis auszeichnen und damit die ML-Decodierung zu rechenaufwendig wird. Im Fall der RM-Codes sind bisher bei Raten R wesentlich von Null bzw. Eins

verschieden[1] nur für Codes bis zu einer Länge von $N = 64$ ML-Decoder realisiert worden (siehe z.B. [61]).

In vielen Fällen werden daher suboptimale Decodierverfahren eingesetzt, die nicht immer das Finden des wahrscheinlichsten Codewortes garantieren. Bei einer großen Gruppe der suboptimalen Decodierverfahren wird das Problem aufgeteilt und dann für jeden dieser Teile eine Lösung gesucht. Im allgemeinen werden dabei die Teilergebnisse nicht unabhängig voneinander betrachtet, sondern schon getroffene Entscheidungen für das Lösen der restlichen Teilprobleme benutzt. In diese Klasse der suboptimalen Decodierverfahren fällt u.a. die im folgenden Abschnitt betrachtete Mehrstufendecodierung wie z.B. der Algorithmus von REED zur Decodierung von RM-Codes (s. Abschnitt 5.2.1). Für die in dieser Arbeit betrachteten Codes besteht z.B. auch die Möglichkeit, im ersten Schritt aus den Zuverlässigkeitswerten für V das wahrscheinlichste Codewort $\boldsymbol{v} \in V$ zu suchen, ohne den Code U zu berücksichtigen, und dann im zweiten Schritt abhängig von der getroffenen Entscheidung für $\boldsymbol{v}$ das wahrscheinlichste Codewort $\boldsymbol{u} \in U$ zu ermitteln. Dieses Verfahren wird auch als *Closest Coset Decoding* bezeichnet [22] und führt bei rekursiver Anwendung direkt zu den in [29] bzw. [49] vorgestellten Decodierverfahren, auf die in Abschnitt 5.2.2 eingegangen wird.

5.2. Mehrstufendecodierung

Beim oben erwähnten, mit *Closet Coset Decoding* bezeichneten Verfahren handelt es sich um eine Zweistufendecodierung. Die Mehrstufendecodierung (MSD) hat in besonderer Weise bei der Decodierung von GC-Codes und von Multilevel-Codes Bedeutung erlangt. So kann bei vielen GC-Codes die BMD-Decodierung durch eine MSD realisiert werden. Zudem kann, worauf noch später eingegangen wird, für alle Kanäle mit MSD die Kanalkapazität erreicht werden, sofern die Raten der äußeren Codes an die Kapazitäten der sich ergebenden äquivalenten Kanäle angepaßt sind.

5.2.1. Klassische Mehrheits-Decodierung von REED-MULLER Codes

Der von REED [47] vorgeschlagene Algorithmus zur Decodierung von RM-Codes bei Übertragung über den BSC ist allgemein der bekannteste und wird in fast jedem Lehrbuch, das das Thema REED-MULLER Codes behandelt, erklärt. Er ist sehr einfach und zeichnet sich durch geringe Komplexität aus, da er ausschließlich auf Mehrheitsentscheidungen (engl.: Majority-Logic) basiert[2]. Allgemein ist der REED-Algorithmus zur Decodierung von allen binären RM-Codes und deren Untercodes geeignet und kann somit bei der Decodierung aller in dieser Arbeit betrachteten Codes eingesetzt werden.

Ausgangspunkt ist die Tatsache, daß die Zeilen der Generatormatrix eines RM-Codes die Inzidenzvektoren von Flächen in der euklidischen Geometrie über $\mathrm{GF}(2)$ sind. In $\mathrm{RM}(r, m)$

[1] Bei Coderaten nahe Null bzw. Eins ist bekannterweise auch bei langen Codes die ML-Decodierung relativ einfach zu realisieren. So können $\mathrm{RM}(1, m)$-Codes auch effizient mit Hilfe der Hadamard-Transformation bzw. des FFT-Algorithmus ML-decodiert werden.

[2] Der bei der Datenübertragung vom Satelliten Mariner 9 eingesetzte $(32, 5, 16)$-Code $\mathrm{RM}(1, 5)$ wurde allerdings durch einen FFT-Algorithmus decodiert [67, Chapter 14].

repräsentieren die Zeilen mit Gewicht $w_{\mathrm{H}} = 2^{m-r}$ verschiedene Flächen mit Dimension $(m-r) = \log_2(w_{\mathrm{H}})$, sog. $(m-r)$-Flächen, die zueinander nicht parallel sind. Bei der in Kapitel 4 gegebenen Konstruktion ist der Punkt $(-1,-1\ldots,-1)$ in allen dieser $(m-r)$-Flächen enthalten. In $\mathrm{EG}(m,2)$, der euklidischen Geometrie der Dimension m über $\mathrm{GF}(2)$, gibt es nun zu jeder $(m-r)$-Fläche eine r-Fläche, die mit der $(m-r)$-Fläche nur den gemeinsamen Punkt $(-1,-1\ldots,-1)$ besitzt und die Flächen aller weiteren Zeilen mit Gewicht $\geq 2^{m-r}$ in keinem oder in $2^i, i \geq 1$ Punkten schneidet. Diese Eigenschaft besitzen auch die $2^{m-r}-1$ parallelen Verschiebungen dieser r-Fläche , d.h. jede schneidet die $(m-r)$-Fläche in genau einem Punkt, die anderen $(m-\rho)$-Flächen, $\rho \geq r$ allerdings in einer geraden Anzahl von Punkten. Damit eignen sich die 2^{m-r} Inzidenzvektoren dieser parallelen r-Flächen $\boldsymbol{t}_j = (t_{j,1}, t_{j,2}, \ldots, t_{j,2^m}), j = 1,\ldots,2^{m-r}$ als Prüfvektoren für das entsprechende Informationssymbol, denn die 2^{m-r} Tests[3]

$$\widetilde{b}_{i,j} = \langle \boldsymbol{y} \cdot \boldsymbol{t}_j \rangle \quad ; \quad j = 1,\ldots,2^{m-r} \tag{5.1}$$

liefern 2^{m-r} unabhängige Aussagen über das Informationssymbol[4] b_i. Falls nun für die Anzahl der Übertragungsfehler $w_{\mathrm{H}}(\boldsymbol{n})$ gilt

$$w_{\mathrm{H}}(\boldsymbol{n}) \leq \frac{1}{2}2^{m-r} - 1, \tag{5.2}$$

liefert die Mehrheit der Tests das richtige Ergebnis. Auf diese Weise können alle[5] $\binom{m}{r}$ Symbole b_i, deren zugehörige Zeilenvektoren der Generatormatrix das Gewicht 2^{m-r} besitzen, geschätzt werden, indem[6]

$$\widehat{b}_i = \begin{cases} 1 & \text{für } \sum_j \widetilde{b}_{i,j} \geq 0 \\ -1 & \text{für } \sum_j \widetilde{b}_{i,j} < 0 \end{cases} \tag{5.3}$$

gesetzt wird. Unter der Annahme, daß diese richtig sind, werden sie dann vom Empfangsvektor $\boldsymbol{y}$ abgezogen und im zweiten Schritt die Symbole b_i geschätzt, deren Zeilenvektoren das Gewicht 2^{m-r+1} besitzen. Die Decodierung läuft damit in $L = r$ Stufen ab, man spricht auch von L-Step-Majority-Logic-Decoding (deut.: L-Stufen-Mehrheits-Decodierung).

Dieser Algorithmus gehört damit in die Klasse der Mehrstufen-Decodierer, die auf einer unteren Stufe getroffenen Entscheidungen werden auf den höheren Stufen verwendet. Allerdings werden auf jeder Stufe die $\binom{m}{\rho}$ Informationssymbole b_i unabhängig voneinander geschätzt, was den Vorteil der Parallelisierbarkeit besitzt, aber auch den Nachteil mit sich bringt, daß eventuelle Abhängigkeiten in den Entscheidungen nicht berücksichtigt werden und dadurch die Wahrscheinlichkeit einer Fehlentscheidung ansteigt. Gleichung (5.2) zeigt zwar, daß alle Fehlermuster, deren Gewicht kleiner als die halbe Mindestdistanz des Codes ist, korrigiert werden können, jedoch besitzen lange RM-Codes verglichen mit anderen binären Codes wie z.B. den BCH-Codes eine relative kleine Mindestdistanz und daher ist der Algorithmus von REED nur bei kurzen RM-Codes interessant.

[3] Mit $\langle \boldsymbol{x} \cdot \boldsymbol{y} \rangle$ wird hier das innere Produkt $\sum_k x_k \cdot y_k$ bezeichnet. Die Rechnung wird hier in $\mathrm{GF}(2)$ durchgeführt.
[4] Die Reihenfolge der b_i stimmt hier nicht mit der Reihenfolge aus Bsp. 4.1 überein.
[5] Falls es sich um einen Untercode handelt, bei dem nicht alle $\binom{m}{r}$ Symbole b_i genutzt werden, vermindert sich die Anzahl der zu schätzenden Symbole entsprechend.
[6] Die Addition der $\widetilde{b}_{i,j} \in \{\pm 1\}$ erfolgt hier in $\mathbb{R}$.

Bei der Übertragung über den AWGN-Kanal ergeben sich aus den unterschiedlichen Zuverlässigkeiten der einzelnen Empfangssymbole y_k auch unterschiedliche Zuverlässigkeiten für die einzelnen Tests $\widetilde{b}_{i,j}$. Aufgrund der in Abschnitt 3.2.1 durchgeführten Überlegungen sind z.B. die mit Zuverlässigkeiten gemäß Gl. (3.5) gewichteten Tests gegeben durch

$$\widetilde{h}_{i,j} = \prod_{\{k:t_{j,k}=-1\}} h_k.$$

Die h_k sind hier die auf Wahrscheinlichkeiten basierten Zuverlässigkeitswerte. Die Entscheidungsregel Gl. (5.3) verändert sich dann zu

$$\widehat{b}_i = \begin{cases} 1 & \text{für } \prod_j (1+\widetilde{h}_{i,j}) \geq \prod_j (1-\widetilde{h}_{i,j}) \\ -1 & \text{sonst,} \end{cases} \tag{5.4}$$

da das wahrscheinlichste b_i gemäß der ML-Regel Gl. (3.4) sich durch

$$\widehat{b}_i = \arg\max_{b_i} \prod_j (1 + b_i \cdot \widetilde{h}_{i,j})$$

bestimmt. In [9] wurde gezeigt, daß die Entscheidungsregel (5.4) für $(m-r) \to \infty$ auch näherungsweise wie folgt geschrieben werden kann

$$\widehat{b}_i = \begin{cases} 1 & \text{für } \sum_j \widetilde{h}_{i,j} \geq 0 \\ -1 & \text{für } \sum_j \widetilde{h}_{i,j} < 0 \end{cases}. \tag{5.5}$$

Damit ergibt sich strukturell die gleiche Entscheidungsregel wie die, die sich bei Verwendung von auf der euklidischen Distanz basierten Zuverlässigkeitswerten

$$\widetilde{y}_{i,j} = \prod_{\{k:t_{j,k}=-1\}} \operatorname{sign} y_k \cdot \min_{\{k:t_{j,k}=-1\}} |y_k|$$

zu

$$\widehat{b}_i = \begin{cases} 1 & \text{für } \sum_j \widetilde{y}_{i,j} \geq 0 \\ -1 & \text{für } \sum_j \widetilde{y}_{i,j} < 0 \end{cases}$$

ergeben hätte. Entscheidend für die Beurteilung des Algorithmus ist jedoch folgender Satz [9]

Satz 5.1 Sei $\varepsilon = P(\widehat{b}_i \neq b_i) < 1/2$ eine beliebige feste Fehlerwahrscheinlichkeit bei Übertragung über den BSC bzw. den AWGN-Kanal und Decodierung mit dem Algorithmus nach REED, wobei für den AWGN-Kanal die Entscheidungsregel (5.4) für die Schätzung der Informationssymbole b_i verwendet wird. Die dazugehörigen Rauschvarianzen, die für beide Kanäle die gleiche Fehlerwahrscheinlichkeit ε ergeben, seien σ^2_{BSC} bzw. σ^2_{AWGN}. Für $N \to \infty$ und feste Coderate R, $0 < R < 1$ ergibt sich[7]

$$\frac{Q(1/\sigma_{\text{AWGN}})}{Q(1/\sigma_{\text{BSC}})} \to \frac{4}{\pi}$$

□

[7] Die Q-Funktion ist gegeben durch $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \mathrm{e}^{-u^2/2} \mathrm{d}u$

Für kleine Rauschvarianzen ergeben sich daraus mit $2E_S/N_0 = 1/\sigma_0^2$ und der Näherung $Q(x) \approx e^{-x^2/2}$ die SNR, die in beiden Kanälen zur gleichen Fehlerwahrscheinlichkeit führen zu

$$\left.\frac{E_S}{N_0}\right|_{\text{AWGN}} \approx \left.\frac{E_S}{N_0}\right|_{\text{BSC}} - \ln\frac{4}{\pi}.$$

D.h. für große SNR ist der Gewinn an Signalleistung im AWGN-Kanal im Vergleich zur Übertragung über den BSC praktisch zu vernachlässigen. Damit wird klar, daß der REED-Algorithmus für lange RM-Codes mit einer Rate R wesentlich von Null verschieden das im Code vorhandene Korrekturpotential nicht annähernd vollständig ausnutzt, sondern daß im Gegenteil dieser Algorithmus asymptotisch schlecht ist. Das wird auch bestätigt durch die Tatsache, daß für große SNR, feste Coderate R und $N \to \infty$ gilt [9]

$$\sigma^2_{\text{BSC}} \sim \sigma^2_{\text{AWGN}} \sim \frac{1}{m}. \tag{5.6}$$

Damit *wächst* bei festem SNR die Fehlerwahrscheinlichkeit mit der Codelänge $N = 2^m$.

5.2.2. Rekursive Bitweise Mehrstufendecodierung

Bei der im obigen Abschnitt beschriebenen Mehrstufendecodierung nach REED basiert die Schätzung von Informationssymbolen auf höheren Stufen auf den bereits getroffenen Entscheidungen für niedere Stufen. Auf jeder Stufe werden jedoch die Informationssymbole unabhängig von den Symbolen der gleichen Stufe geschätzt. Mit dem Ziel, die Wortfehlerwahrscheinlichkeit zu verringern ist es jedoch sinnvoll, die getroffenen Entscheidungen sofort für alle späteren Schätzungen zu verwenden. Eine Möglichkeit dazu bietet die Betrachtung von RM-Codes als mehrfach verkettete GC-Codes, wie sie von KABATYANSKY [29] allgemein für den semikontinuierlichen Kanal vorgestellt und deren Zuverlässigkeitsübergabe später von SCHNABL und BOSSERT [49] für den AWGN-Kanal spezialisiert wurde. Prinzipiell ist der im folgenden beschriebene Algorithmus eine direkte Folge der rekursiven $|u|u+v|$-Konstruktion und er läßt sich auf die Decodierung sowohl von RM-Codes als auch deren Untercodes anwenden.

Wie in Abschnitt 3.2 beschrieben, können bei einer zweistufigen Decodierung in der ersten Stufe aus den Empfangswerten y_k nach Gl. (3.3) die Symbole v_k geschätzt werden, wobei es bei Übertragung über den AWGN-Kanal sinnvoll ist, noch zusätzlich Zuverlässigkeitswerte nach Gl. (3.5) oder (3.11) anzugeben. Zur Decodierung von $\boldsymbol{v} \in V$ kann nun z.B. die Entscheidungsregel (3.4), die das ML-Codewort des Gesamtcodes C bestimmt, angewendet werden, wobei die Summation über die $h_k^{(v)}$ durchgeführt wird (vergl. Bsp. 3.1)

$$\widehat{\boldsymbol{v}} = \arg\max_{\boldsymbol{v}\in V} \prod_k (1 + v_k h_k^{(v)}). \tag{5.7}$$

Falls die Zuverlässigkeiten durch die $y_k^{(v)}$ nach Gl. (3.11) übergeben werden, ergibt sich mit $w_k^{(v)} = |y_k^{(v)}|$

$$\widehat{\boldsymbol{v}} = \arg\min_{\boldsymbol{v}\in V} \sum_{\{k: v_k \neq \widetilde{v}_k\}} w_k^{(v)} \tag{5.8}$$

wobei wie schon in Bsp. 3.1 gezeigt wurde, die Entscheidungen nach Gl. (5.7) und (5.8) nicht in jedem Fall identisch sind. Auf der zweiten Stufe erfolgt die Decodierung von U abhängig von der getroffenen Entscheidung $\widehat{\boldsymbol{v}}$ gemäß

$$\widehat{\boldsymbol{u}} = \arg\max_{\boldsymbol{u}\in U} \prod_k (1 + u_k h_k^{(u)}) \tag{5.9}$$

bei auf Wahrscheinlichkeiten basierter Zuverlässigkeitsübergabe nach Gl. (3.7) bzw. gemäß

$$\widehat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}\in U} \sum_{\{k:u_k\neq\widetilde{u}_k\}} w_k^{(u)}, \tag{5.10}$$

falls die Zuverlässigkeitsübergabe nach Gl. (3.12) erfolgt. Da sich die Zuverlässigkeitswerte $h_k^{(u)}$ und $y_k^{(u)}$ verlustlos ineinander umrechnen lassen, liefern natürlich beide Regeln (5.9) und (5.10) für gleiches $\widehat{\boldsymbol{v}}$ auch die gleiche Entscheidung $\widehat{\boldsymbol{u}}$. Wenn sich, wie bei RM-Codes, die äußeren Codes U, V auch durch die $|u|u+v|$-Konstruktion beschreiben lassen, kann die Entscheidung für $\boldsymbol{v}$ und $\boldsymbol{u}$ ebenfalls in zwei Schritten erfolgen, indem die Zuverlässigkeitswerte auf die gleiche Weise nach außen weitergereicht werden. Führt man diese Zerlegung rekursiv weiter, gelangt man beim Code $\mathrm{RM}(r, m)$ mit Dimension K schließlich zu K Wiederholcodes als äußeren Codes, bzw. falls auch diese Codes verallgemeinert als $|u|u + v|$-verkettete Codes aufgefaßt werden, zu 2^m Codes der Länge $N^{(i)} = 1$ mit Dimension $K^{(i)} \in \{0, 1\}$, $i \in \{1, \ldots 2^m\}$, wobei die K Codes $C^{(i)}$ mit $K^{(i)} = 1$ jeweils ein Informationssymbol b_i repräsentieren (vergl. Abschnitt 3.3.2). Diese Codes der Länge $N^{(i)} = 1$ können sehr einfach decodiert werden und es ergibt sich damit für die Gesamtkomplexität des Algorithmus die Ordnung[8] $\mathcal{O}(N \log_2 N)$ mit der Gesamtcodelänge $N = 2^m$ [8]. Da bei diesem Algorithmus die Codes mit Länge $N^{(i)} = 1$ und damit auch die Informationsbits b_i der Reihe nach entschieden werden, wird dieser Algorithmus im folgenden auch als *bitweise Mehrstufendecodierung* (bitweise MSD) bezeichnet.

Beispiel 5.1 Für den Code der Länge $N = 8$ ist in Bild 5.1 schematisch der Ablauf bei bitweiser MSD dargestellt. Aus dem Empfangsvektor $\boldsymbol{y}$ werden, wie in in Bild 3.4 (S. 19) gezeigt, rekursiv die Zuverlässigkeitswerte für die äußeren Codes berechnet, zuerst für den Code $C^{(1)}$. Die Entscheidung für alle äußeren Code erfolgt gemäß

$$\widehat{\boldsymbol{c}}^{(i)} = \begin{cases} (-1) & \text{für} \quad h_1^{(i)} < 0 \wedge K^{(i)} = 1 \\ (1) & \text{sonst} \end{cases} .$$

Nach der Entscheidung für $C^{(i)}$ werden die Zuverlässigkeitswerten für den nächsten Code $C^{(i+1)}$ berechnet. Wenn alle acht äußeren Codes entschieden sind, kann die Schätzung $\widehat{\boldsymbol{c}}$ bestimmt werden. □

[8] Die größte Komplexität ergibt sich für den voll besetzten Code mit $K^{(i)} = 1, \forall i$. Um auf jeder Stufe die Vektoren der Zuverlässigkeitswerte für die äußeren Codes U und V zu berechnen, muß bei auf euklidischer Distanz basierter Zuverlässigkeitsübergabe für die $y_k^{(v)}$ $\frac{N}{2}$-mal das Mininmum aus zwei Werten bestimmt und eine Vorzeichenkorrektur durchgeführt werden, für die $y_k^{(u)}$ sind $\frac{N}{2}$ Additionen/Subtraktionen nötig. Bei einem Code der Länge N kann die Decodierung $\log_2 N$-mal jeweils an die äußeren Codes weitergereicht werden. Zur Decodierung der äußeren Codes mit Länge 1 ist nur ein Vergleich erforderlich. Falls die Übergabe durch die h_i erfolgt, sind auf jeder Stufe eine Multiplikation, eine Division und zwei Additionen erforderlich. Auch hier ist die Komplexität proportional zu $N \log_2 N$.

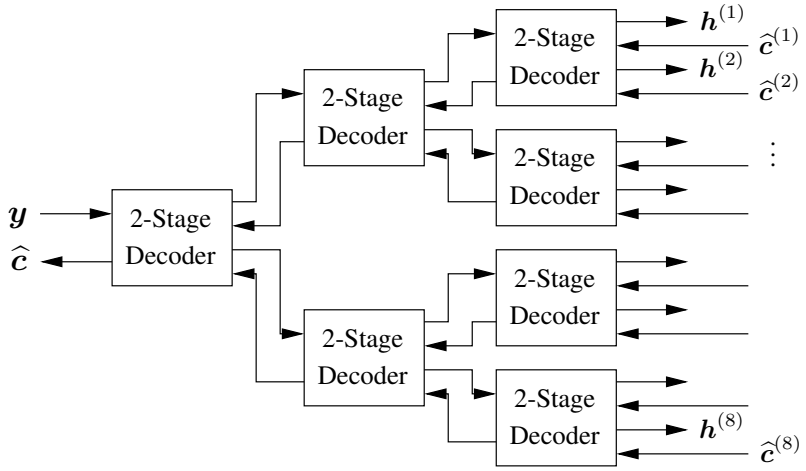

Abbildung 5.1.: Darstellung der bitweisen MSD für einen Code der Länge $N = 8$ bei wahrscheinlichkeitsbasierter Zuverlässigkeitsübergabe

Natürlich kann, wenn eine höhere Komplexität zulässig ist, die Zerlegung auf einer Zwischenstufe abgebrochen und die ML-Decodierung der sich dabei ergebenden äußeren Codes durchgeführt werden. Speziell bei langen RM-Codes ergibt sich dadurch eine wesentlich verbesserte Fehlerwahrscheinlichkeit, wobei jedoch hierzu die Zerlegung schon vor dem Erreichen der Wiederholcodes $\mathrm{RM}(0, m')$ bzw. der Parity-Check-Codes $\mathrm{RM}(m'-1, m')$ abgebrochen werden sollte[9,10].

Prinzipiell führt der in diesem Abschnitt vorgestellte rekursive Algorithmus auch bei einer Zerlegung in äußere Codes bis zur Länge $N^{(i)} = 1$ zu einer etwas geringeren Wortfehlerwahrscheinlichkeit als der REED-Algorithmus, da bereits getroffene Entscheidungen sofort bei allen anderen Entscheidungen berücksichtigt werden. Allerdings ist bei RM-Codes die *erste* zu treffende Entscheidung für den *ersten* Code $C^{(i)}$ mit $K^{(i)} = 1$ nicht zuverlässiger als die Entscheidung der Informationssymbole im ersten Schritt beim REED-Algorithmus und die Wortfehlerwahrscheinlichkeiten beider Algorithmen unterscheiden sich daher nicht wesentlich. Aus diesem Grund ist bei RM-Codes mit konstanter Coderate und $m \to \infty$ der rekursive Algorithmus für $N^{(i)} = 1$, $\forall i$ genauso wie der REED-Algorithmus asymptotisch schlecht.

Von zumindest theoretischem Interesse ist folgender Satz, der auch schon in ähnlicher Weise von KABATYANSKY [29] für den semikontinuierlichen Kanal mit etwas anderer Zuverlässigkeitsübergabe gezeigt wurde:

Satz 5.2 . Es sei $\boldsymbol{c} \in C$ das gesendete und $\widehat{\boldsymbol{c}} \in C$ das durch die bitweise MSD gefundene Co-

[9] In [49] wurde die Zerlegung neben den Wiederholcodes $\mathrm{RM}(0, m')$ auch schon bei den Parity-Check-Codes $\mathrm{RM}(m'-1, m')$ abgebrochen und diese ML-decodiert. Das führt im Vergleich zur Zerlegung bis zu Codes der Länge $N = 1$ nur bei der Verwendung von auf Wahrscheinlichkeiten basierten Zuverlässigkeitswerten zu einer Verbesserung. Bei der Verwendung von auf euklidischer Distanz basierten Zuverlässigkeitswerten ergibt sich keine Verbesserung, da in diesem Fall der nächste äußere Code U der Code $\mathrm{RM}(m'-1, m'-1)$ ist, der keine Redundanz enthält und damit ist die rekursive Decodierung äquivalent zur ML-Decodierung (siehe auch Bsp. 3.1). Natürlich kann in diesem Fall die ML-Decodierung unter Umständen aufwandsgünstiger als mit dem rekursiven Algorithmus realisiert werden.

[10] In [10] wurde die Zerlegung jeweils bei den Codes $\mathrm{RM}(1, m')$ bzw. $\mathrm{RM}(m'-1, m')$ abgebrochen.

dewort unter Verwendung distanzbasierter Zuverlässigkeitsübergabe. Sei $\Omega_N = \{1, 2, \ldots, N\}$. Wenn für alle Mengen $\omega \subseteq \Omega_N$ mit $|\omega| = d_{\mathrm{H,min}}(C)$ gilt

$$\sum_{k \in \omega} c_k \cdot y_k > 0, \tag{5.11}$$

dann ist $\boldsymbol{c} = \widehat{\boldsymbol{c}}$.

Beweis:

1. Falls Bedingung (5.11) erfüllt ist, gilt auch für alle $\omega \subseteq \Omega_{N/2}$ mit $|\omega| = d_{\mathrm{H,min}}(C)$

$$\sum_{k \in \omega} v_k \cdot y_k^{(v)} > 0,$$

   denn es läßt sich leicht zeigen, daß wenn obige Bedingung für die $y_k^{(v)}$ nicht erfüllt ist, auch eine Menge ω gefunden werden kann, für die Gl. (5.11) nicht erfüllt ist.

2. Falls die Bedingung (5.11) erfüllt ist, folgt mit der gleichen Argumentation für alle $\omega \subseteq \Omega_{N/2}$ mit $|\omega| = d_{\mathrm{H,min}}(C)/2$

$$\sum_{k \in \omega} u_k \cdot y_k^{(u)} > 0.$$

Rekursives Anwenden von Punkt 1 und 2 ergibt schließlich, daß bei Zerlegung von C in äußere Wiederholcodes diese richtig Entschieden werden. Da weiterhin die bitweise MSD von Wiederholcodes gleichwertig zur ML-Decodierung ist, folgt die Behauptung.

□

Ein direktes Ergebnis dieses Satzes ist folgendes Korollar, das auch schon von SCHNABL und BOSSERT [49] bewiesen wurde:

Korollar 5.1 Wenn für das gesendete Codewort $\boldsymbol{c}$ gilt

$$d_{\mathrm{E}}(\boldsymbol{y}, \boldsymbol{c}) < \frac{d_{\mathrm{E,min}}(C)}{2} = \sqrt{d_{\mathrm{H,min}}(C)},$$

dann wird $\boldsymbol{c}$ durch bitweise MSD mit distanzbasierter Zuverlässigkeitsübergabe gefunden.

Beweis: Obige Ungleichung kann umgeformt werden zu

$$d_{\mathrm{E}}^2(\boldsymbol{y}, \boldsymbol{c}) = \sum_k (y_k - c_k)^2 < d_{\mathrm{H,min}}(C)$$

und damit gilt auch für alle $\omega \subseteq \Omega_N$, $|\omega| = d_{\mathrm{H,min}}(C)$

$$\sum_{k \in \omega} (y_k - c_k)^2 < d_{\mathrm{H,min}}(C).$$

Daraus kann direkt Gl. (5.11) abgeleitet werden.

□

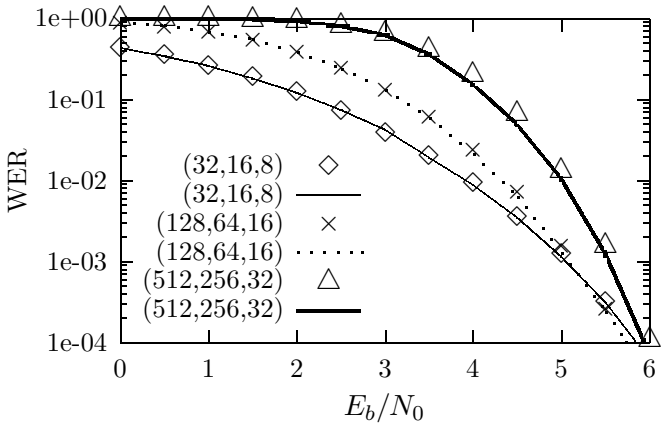

Abbildung 5.2.: Wortfehlerwahrscheinlichkeiten für den AWGN-Kanal für verschiedene RM-Codes mit Parametern (N, K, D) bei rekursiver Decodierung und Zerlegung in äußere Codes mit Länge $N^{(i)} = 1, \forall i$ (Linien beziehen sich auf wahrscheinlichkeitsbasierte, Marker auf distanzbasierte Zuverlässigkeitsübergabe)

Danach werden alle Vektoren $\boldsymbol{y}$, die bei Soft-Decision-BMD-Decodierung richtig decodiert werden, auch bei bitweiser MSD richtig entschieden. Für den AWGN-Kanal hat dieses Korollar jedoch nur geringe Bedeutung, da hier Soft-Decision-BMD-Decodierung wesentlich schlechtere Wortfehlerraten liefert als Hard-Decision-BMD-Decodierung[11].

In Bild 5.2 sind die Wortfehlerwahrscheinlichkeiten einiger RM-Codes mit Rate $R = 0.5$ bei Decodierung mit dem rekursiven Algorithmus für die beiden Zuverlässigkeitsübergaben gezeigt. Zwar ergeben sich bei (optimaler) wahrscheinlichkeitsbasierter Zuverlässigkeitsübergabe etwas kleinere Wortfehlerwahrscheinlichkeiten als bei Verwendung der auf euklidischer Distanz basierten Zuverlässigkeitswerte, aber über den ganzen betrachteten Bereich sind die Unterschiede praktisch zu vernachlässigen. Damit bestätigt sich die Erkenntnis aus Abschnitt 3.2.3, daß die beiden Prinzipien der Zuverlässigkeitsübergabe praktisch gleichwertig sind, wobei allerdings die distanzbasierte Übergabe wesentlich aufwandsgünstiger durchgeführt werden kann und daher vorgezogen werden sollte.

Allgemein ist für diesen Algorithmus und Decodierung langer RM-Codes der Rate $R = 0.5$ das notwendige SNR für moderate Fehlerraten relativ hoch. Nur bei kurzen RM-Codes liefert der hier beschriebene Decodieralgorithmus akzeptable Ergebnisse.

Eine erste Erweiterung dieses Algorithmus in Richtung Listendecodierung wurde von LUCAS, BOSSERT und DAMMANN vorgeschlagen [33]. Anstatt bei der rekursiven Decodierung und Zerlegung in äußere Wiederholcodes gemäß [49] jeweils nur *eine* Entscheidung für alle späteren Decodierschritte weiterzuverwenden, wird hier mit Codewortlisten gearbeitet. Auf jeder Zerlegungsstufe wird zuerst eine Liste $\mathcal{L}_V \subseteq V$ gebildet. Für jedes Codewort $\boldsymbol{v} \in \mathcal{L}_V$ wird im nächsten Schritt eine Liste $\mathcal{L}_U \subseteq U$ gebildet. Aus der Vereinigung aller $|\mathcal{L}_V| \cdot |\mathcal{L}_U|$ Listen ergibt sich schließlich die Liste $\mathcal{L}$, $|\mathcal{L}| \leq |\mathcal{L}_V| \cdot |\mathcal{L}_U|$, die auf der nächsthöheren Stufe verwendet wird. Auf der obersten Stufe wird schließlich das wahrscheinlichste Codewort $\boldsymbol{c} \in \mathcal{L}$ ausgewählt. Es läßt sich leicht zeigen, daß, falls für alle Zerlegungsstufen gilt $\mathcal{L}_V = V$, $|\mathcal{L}_U| \geq 1$ und $|\mathcal{L}| \geq 1$,

[11] In der Tat werden bei Hard-Decision-BMD-Decodierung viele Vektoren $\boldsymbol{y}$ richitg entschieden, die bei Soft-Decision-BMD-Decodierung zu einem Decodierversagen führen.

dieser Algorithmus immer das ML-Codewort $\boldsymbol{c}_{\mathrm{ML}}$ findet.

5.2.3. Abschätzung der Wortfehlerwahrscheinlichkeit bei bitweiser Mehrstufendecodierung

Die Wortfehlerwahrscheinlichkeit bei Decodierung mit dem oben beschriebenen rekursiven Algorithmus und Zerlegung in äußere Codes der Länge $N^{(i)} = 1, \forall i$ kann auf einfache Art durch die Wahrscheinlichkeitsdichten der auf euklidische Distanz basierten Zuverlässigkeitswerte abgeschätzt werden. Bei auf euklidischer Distanz basierter Zuverlässigkeitsübergabe sind für eine einstufige Zerlegung in zwei äußere Codes $C^{(1)} = V$ und $C^{(2)} = U$ die Wahrscheinlichkeitsdichten $f(y_k^{(v)}|v_k = \pm 1)$ und $f(y_k^{(u)}|u_k = \pm 1, \widehat{v}_k = v_k)$ in Abschnitt 3.2.3 ermittelt worden. Prinzipiell können aus den 2^ρ Dichten der Zuverlässigkeitswerte $y_k^{(i)}$, $i = 1, \ldots, 2^\rho$ der ρ-ten Zerlegungsstufe die $2^{\rho+1}$ Dichten der Zuverlässigkeitswerte[12] $\widetilde{y}_k^{(2i-1)}, \widetilde{y}_k^{(2i)}$ der $(\rho+1)$-ten Stufe durch numerisches Auswerten der Integrale[13,14] (s. Anhang A.5)

$$\begin{aligned} f(\widetilde{y}_k^{(2i-1)}|\widetilde{c}_k^{(2i-1)} = \pm 1) &= 2 \cdot f_{y_k^{(i)}}(\widetilde{y}_k^{(2i-1)}|c_k^{(i)} = \widetilde{c}_k^{(2i-1)}) \int_{|\widetilde{y}_k^{(2i-1)}|}^{\infty} f(y_k^{(i)}|c_k^{(i)} = 1)\mathrm{d}y_k^{(i)} \\ &\quad + 2 \cdot f_{y_k^{(i)}}(-\widetilde{y}_k^{(2i-1)}|c_k^{(i)} = \widetilde{c}_k^{(2i-1)}) \int_{-\infty}^{-|\widetilde{y}_k^{(2i-1)}|} f(y_k^{(i)}|c_k^{(i)} = 1)\mathrm{d}y_k^{(i)} \end{aligned}$$

bzw.[15]

$$\begin{aligned} f(\widetilde{y}_k^{(2i)}|\widetilde{c}_k^{(2i)} = \pm 1) &= f(y_k^{(i)}|c_k^{(i)} = \pm 1) * f(y_k^{(i)}|c_k^{(i)} = \pm 1) \\ &= \int_{-\infty}^{\infty} f_{y_k^{(i)}}(y_k^{(i)}|c_k^{(i)} = \pm 1) \cdot f_{y_k^{(i)}}(\widetilde{y}_k^{(2i)} - y_k^{(i)}|c_k^{(i)} = \pm 1)\mathrm{d}y_k^{(i)} \end{aligned}$$

bestimmt werden. Bei einer rekursiven Zerlegung in äußere Codes der Länge $N^{(i)} = 1$ ist es damit möglich, die Wahrscheinlichkeitsdichten der an die äußeren Decoder übergebenen Zuverlässigkeitswerte rekursiv zu berechnen und daraus die Wahrscheinlichkeit einer Fehlentscheidung[16]

$$P_e^{(i)} = P(\widehat{\boldsymbol{c}}^{(i)} \neq \boldsymbol{c}^{(i)}) = \begin{cases} \int_{-\infty}^{0} f(y_1^{(i)}|c_1^{(i)} = 1)\mathrm{d}y_1^{(i)} & \text{für } K^{(i)} = 1 \\ 0 & \text{für } K^{(i)} = 0 \end{cases}$$

[12] Die $\widetilde{y}_k^{(2i-1)}$ entsprechen hier den Werten $y_k^{(v)}$ der ersten Zerlegungsstufe und die $\widetilde{y}_k^{(2i)}$ entsprechen den $y_k^{(u)}$ der zweiten Zerlegungsstufe.

[13] Die hier angegebenen Dichten setzen natürlich voraus, daß auf der ρ-ten Stufe die Codes $C^{(j)}$, $j < i$ und auf der $(\rho+1)$-ten Stufe die Codes $\widetilde{C}^{(j)}$, $j < 2i-1$ bzw. $j < 2i$ richtig entschieden wurden.

[14] Aus Gründen der Übersichtlichkeit wird bei den Wahrscheinlichkeitsdichten dort, wo Mißverständnisse ausgeschlossen sind, wie bisher auch schon auf den Index verzichtet.

[15] Da mit $\widetilde{y}_k^{(2i)} = y_k^{(i)} + \widetilde{c}_k^{(2i-1)} y_{N+k}^{(i)}$ die Größe $\widetilde{y}_k^{(2i)}$ die Summe zweier Zufallsgrößen ist, ergibt sich die Dichte durch die Faltung der Einzeldichten, falls wie vorausgesetzt, die Entscheidung $\widehat{\boldsymbol{c}}^{(2i-1)}$ richtig ist.

[16] Auch hier wird vorausgesetzt, daß für $K^{(i)} = 1$ die Codeworte $C^{(i)} \in \{(\pm 1)\}$ mit gleicher Wahrscheinlichkeit ausgewählt werden, und daß die Fehlerwahrscheinlichkeit nicht vom ausgwählten Codeworte abhängt.

zu bestimmen, da in diesem Fall die Decodierentscheidung für jeden einzelnen Code mit $K^{(i)} = 1$ nur aufgrund des Vorzeichens des entsprechenden Zuverlässigkeitswertes $y_1^{(i)}$ gefällt wird. Das Gesamtcodewort $\boldsymbol{c} \in \mathrm{RM}(r, m)$ wird nur dann richtig entschieden, wenn alle Einzelentscheidungen für alle äußeren Codes richtig sind. Damit ergibt sich unter der Annahme, daß die Einzelentscheidungen voneinander statistisch unabhängig getroffen werden,

$$P_e = P(\widehat{\boldsymbol{c}} \neq \boldsymbol{c}) = 1 - \prod_i (1 - P_e^{(i)}). \tag{5.12}$$

Diese Methode ist bei kurzen Codes und kleinem SNR sehr effizient, auch wenn sie tendenziell etwas zu große Wortfehlerraten liefert[17]. Bei langen Codes erfordert die oben beschriebene Methode insbesondere bei großem SNR jedoch eine sehr genaue numerische Approximation der Wahrscheinlichkeitsdichten in den Randbereichen, und dies auf allen Zwischenstufen, die bei der Rekursion durchlaufen werden. Daher werden die Ergebnisse für große SNR zunehmend ungenau.

Eine Alternative bietet hier das in Abschnitt 3.2.4 vorgestellte äquivalente SNR, das die äquivalenten Kanäle für die Übertragung der beiden äußeren Codeworte bei Zweistufendecodierung charakterisiert. Wie schon erwähnt, können damit besonders bei großem SNR die Kanaleigenschaften sehr genau beschrieben werden, da dann auch der äquivalente Kanal für die Übertragung von V näherungsweise ein Gaußkanal ist. In gleicher Weise, in der die Wahrscheinlichkeitsdichten der an die Decoder der äußeren Codes gegebenen Zuverlässigkeitswerte rekursiv berechnet werden können, lassen sich auch die äquivalenten SNR-Werte rekursiv berechnen. Dies ist in Bild 5.3 schematisch für eine 3-stufige Zerlegung in 8 äußere Codes gezeigt. Dieser rekursiven Bestimmung der äquivalenten SNR-Werte liegt die Annahme zu Grunde, daß alle Zuverlässigkeitswerte auf jeder Stufe gaußverteilt sind, was strenggenommen nur für die Werte im untersten Weg (der Weg zu $\mathrm{SNR}^{(8)}$) gilt, da nur hier in jedem Schritt die Zuverlässigkeitswerte addiert werden.

Bei einer Zerlegung in 2^m äußere Codes der Länge $N^{(i)} = 1$ lassen sich auf diese Art 2^m verschiedene Werte $\mathrm{SNR}^{(i)}$, $i \in \{1, \ldots, 2^m\}$ für die einzelnen äquivalenten Kanäle bestimmen. Die Fehlerwahrscheinlichkeit bei der Decodierung der einzelnen Codes unter der Annahme, daß alle vorherigen Entscheidungen richtig sind, ergibt sich dann näherungsweise durch die BPSK-Fehlerwahrscheinlichkeit bei Übertragung über den AWGN-Kanal [18]

$$P_e^{(i)} = P(\widehat{\boldsymbol{c}}^{(i)} \neq \boldsymbol{c}^{(i)}) \approx \begin{cases} Q(1/\sigma_0^{(i)}) & \text{für } K^{(i)} = 1 \\ 0 & \text{für } K^{(i)} = 0 \end{cases} \tag{5.13}$$

mit der äquivalenten Rauschvarianz $[\sigma_0^{(i)}]^2 = 1/\mathrm{SNR}^{(i)}$. Die damit gewonnenen Ergebnisse werden in Bild 5.4 mit den tatsächlichen Wortfehlerwahrscheinlichkeiten verglichen. Trotz der Annahmen und Vereinfachungen, die der Schätzung zu Grunde liegen, sind die Ergebnisse sehr

[17] In der Tat sind die Einzelentscheidungen nicht voneinander statistisch unabhängig, da sich der zu korrigierende Rauschvektor $\boldsymbol{n} \in \mathbb{R}^N$ nicht verändert. So schränkt z.B. eine richtige Entscheidung für den ersten Code mit $K^{(i)} = 1$ die Menge der möglichen Rauschvektoren $\boldsymbol{n}$, die aufgetreten sein können, ein, und damit ist die Wahrscheinlichkeit, daß auch alle folgenden Entscheidungen für die anderen Codes richtig sind, etwas größer als bei statistisch unabhängigen Entscheidungen.

[18] Die Q-Funktion ist gegeben durch $Q(y) = \frac{1}{\sqrt{2\pi}} \int_y^\infty \mathrm{e}^{-\beta^2/2} \mathrm{d}\beta$.

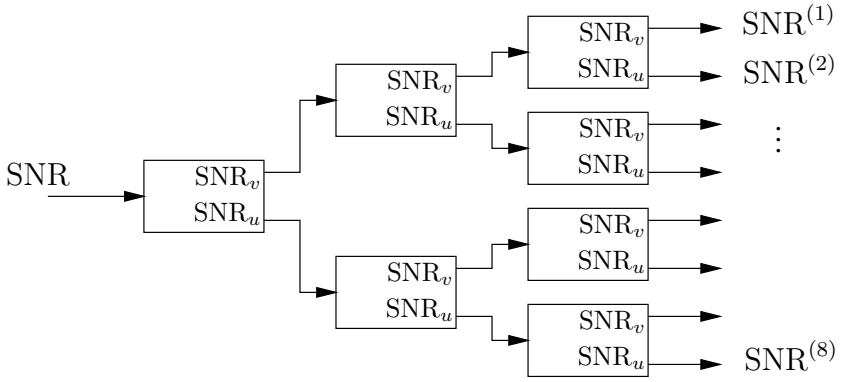

Abbildung 5.3.: Rekursive Berechnung des äquivalenten SNR für die äußeren Codes bei 3-stufiger Zerlegung.

exakt. Damit ist neben den Betrachtungen in Abschnitt 3.2.4 auch die rekursive Bestimmung der äquivalenten SNR-Werte zur Charakterisierung der äquivalenten Kanäle gerechtfertigt.

Durch die oben beschriebene Methode ist es möglich, die Wortfehlerwahrscheinlichkeiten auch in hohen SNR-Bereichen, die durch Simulation auf Grund der dazu notwendigen Rechenzeit nicht mehr erreicht werden können, sehr genau zu bestimmen. Zwar kann für große SNR die Wortfehlerwahrscheinlichkeit auch durch die des BMD-Decoders abgeschätzt werden, die damit erzielten Ergebnisse sind aber wesentlich schlechter als bei der hier vorgestellten Methode (vergl. [49]).

Viel wichtiger als die reine Abschätzung der WER ist jedoch die Optimierung des Codes mit dem Ziel, die WER zu verringern. Es zeigt sich nämlich, daß in der Menge der durch die $|u|u+v|$-Verkettung konstruierten Codes die RM-Codes nicht die beste WER erzielen, wenn das oben beschriebene rekursive Decodierverfahren angewendet wird. Die hier vorgestellte Methode, die WER mit Hilfe der äquivalenten SNR zu schätzen, ermöglicht ein sehr effizientes Verfahren, um bei gegebenen Parametern N und K den Code zu finden, der für dieses Decodierverfahren die kleinste WER liefert. Der Leser wird hierzu auf Abschnitt 6.1 verwiesen.

5.3. Sequentielle- und Listendecodierung

Bei dem im vorigen Abschnitt vorgestellten Decodieralgorithmus werden aus dem Empfangsvektor $\boldsymbol{y}$ rekursiv bis zur gewünschten Zerlegungsstufe die Zuverlässigkeitsvektoren weitergegeben. Diese Zuverlässigkeitswerte werden dann verwendet, um aus den Codeworten des jeweiligen äußeren Codes *ein* Codewort auszuwählen und diese Decodierentscheidung dann für alle folgenden Schritte zu verwenden. Es wird sich also jedes mal immer für *ein* Codewort entschieden. Dies ist sinnvoll, wenn beginnend bei der ersten jede zu treffende Entscheidung sehr zuverlässig und mit hoher Sicherheit getroffen werden kann. Aber schon bei mittlerem SNR ist jedoch z.B. bei RM-Codes die Fehlerwahrscheinlichkeit bereits bei der ersten Entscheidung relativ groß, was zu vergleichsweise großer Gesamtfehlerwahrscheinlichkeit führt. Es erscheint einleuchtend, daß z.B. ein sequentielles Decodierverfahren, bei dem einmal getroffene Entscheidungen revidiert werden können, die Decodierfehlerwahrscheinlichkeit wesentlich verringern

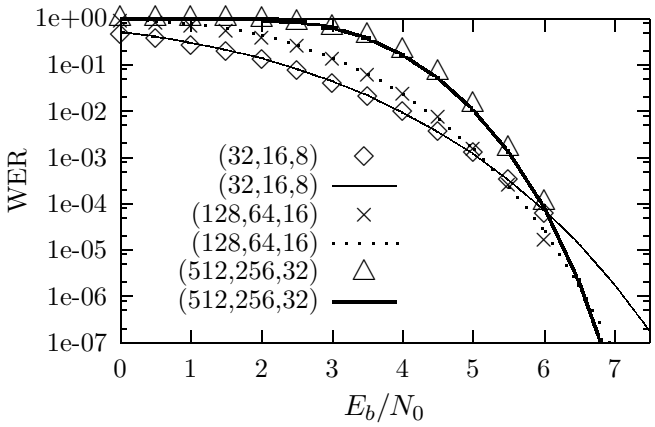

Abbildung 5.4.: Simulierte und geschätzte Wortfehlerwahrscheinlichkeiten für den AWGN-Kanal für verschiedene RM-Codes mit Parametern (N, K, D) bei rekursiver Decodierung und Zerlegung in äußere Codes mit Länge $N^{(i)} = 1, \forall i$, distanzbasierte Zuverlässigkeitsübergabe (Marker beziehen sich auf Simulationsergebnisse, Linien auf Schätzung mit Gl. (5.12) und (5.13)).

kann. In gleiche Weise kann statt einer Entscheidung für ein einzelnes Codewort mit Codewortlisten gearbeitet werden, so daß nur dann ein vermeidbarer[19] Decodierfehler auftritt, wenn das wahrscheinlichste Codewort nicht in der Liste enthalten ist. Listendecodierung für RM-Codes in der hier vorgestellten Art mit distanzbasierter Metrik wurde bereits vom Autor zusammen mit SORGER in [54], [57] bzw. [55], [56] vorgeschlagen. Später wurde der gleiche Algorithmus auch von DUMER und SHABUNOV in [11] und [12] für wahrscheinlichkeitsbasierte Metrik veröffentlicht.

Allgemein zum ersten Mal vorgeschlagen wurde die sequentielle Decodierung von WOZENCRAFT [69]. Sie setzt die Beschreibung des Codes durch einen Codebaum mit einem Startpunkt (Wurzel des Codebaumes) voraus, der während des Decodiervorganges durchlaufen wird. Zu jedem Codewort $\boldsymbol{c} = (c_1, c_2, \ldots, c_N)$ gehört ein Pfad durch den Baum von der Wurzel zu einem Blatt. Codeworte, deren Pfade den gleichen Anfangsteil besitzen, gleichen sich auch in einer entsprechenden Anzahl von Codesymbolen. Beim sequentiellen Algorithmus handelt es sich prinzipiell um einen Algorithmus, der den besten Pfad durch den Baum sucht. Ausgehend von der Wurzel wird in jedem Schritt nur der Pfad mit der besten Metrik verlängert[20], bis schließlich ein Codewort gefunden wird. Zur Beurteilung der verschiedenen Pfade unterschiedlicher Länge wurde von FANO [13] auf Grund von informationstheoretischen Überlegungen heuristisch die folgende additive Metrik für einen Pfad mit Codesymbolfolge $(c_1, c_2, \ldots, c_\tau)$ vorgeschlagen

$$\lambda(c_1, c_2, \ldots, c_\tau) = \sum_{k=1}^{\tau} \log_2 \frac{P(y_k|c_k)}{P(y_k)} - R, \tag{5.14}$$

mit $1 \leq \tau \leq N$. Von MASSEY [35] konnte gezeigt werden, daß, falls die Struktur des nicht

[19] Jeder Algorithmus ist von seiner Fehlerrate durch die des ML-Decoders, der sich für das wahrscheinlichste Codewort entscheidet, nach unten begrenzt.

[20] Beim Algorihmus nach FANO wird allerdings mit einer Schwelle gearbeitet, die im Verlauf der Decodierung dynamisch angepaßt wird. Es wird damit nicht immer der Pfad mit der besten Metrik verlängert, sondern beim Decodieren auf den Codebaum vorwärts und rückwärts gelaufen.

untersuchten Teils des Codebaumes nicht bekannt ist, aus der Menge der verschiedenen Pfade unterschiedlicher Länge derjenige mit größter FANO-Metrik zugleich der wahrscheinlichste Pfad ist. Der Zusammenhang zwischen der Metrik von FANO und der Entscheidungsregel des ML-Decoders Gl. (2.2) wird deutlich, wenn die Metrik für Codesequenzen der Länge N betrachtet wird. Hier ergibt sich

$$\lambda(c_1, c_2, \ldots, c_N) = \sum_{k=1}^{N} \log_2 \frac{P(y_k|c_k)}{P(y_k)} - R = -\underbrace{N \cdot R}_{K} + \log_2 P(\boldsymbol{y}|\boldsymbol{c}) - \log_2 \prod_k P(y_k).$$

In diesem Fall unterscheidet sich die Metrik nach FANO durch die zwei Konstanten $-K$ und $-\log \prod_k P(y_k)$ von der ML-Entscheidungsregel. Daher sollte es prinzipiell möglich sein, diese Metrik auch in anderer Form darzustellen, da sich die ML-Entscheidungsregel auch auf verschiedene Arten formulieren läßt.

So kann die FANO-Metrik z.B. bei einem binären Code mit $c_k \in \{\pm 1\}$ auch in folgender Weise in Abhängigkeit von den h_k ausgedrückt werden (s. Anhang A.2)

$$\lambda(c_1, c_2, \ldots, c_\tau) = \sum_{k=1}^{\tau} \log_2(1 + c_k h_k) - R,$$

was, wie oben erwähnt, die Darstellung des Codes in Form eines Baumes voraussetzt, bei dem Pfade mit gleichem Anfangsteil zu Codeworten gehören, deren erste Codesymbole übereinstimmen.

Damit die in Abschnitt 5.2.2 vorgestellte bitweise MSD zu einem sequentiellen bzw. Listendecodierverfahren erweitert werden kann, ist eine Metrik erforderlich, damit auch hier verschiedene Decodierhypothesen miteinander vergleichen werden können. Dazu bieten sich u.a. die Gleichungen (3.8) bzw. (3.14) als Ausgangspunkt an. Der prinzipielle Unterschied des im folgenden vorgestellten sequentiellen Algorithmus zur klassischen sequentiellen Decodierung mit der FANO-Metrik besteht darin, daß hier nicht die Codesymbole c_k sondern die Codeworte der äußeren Codes $\boldsymbol{c}^{(i)}$ der Reihe nach geschätzt werden. Mit den $N = 2^m$ äußeren Codes $C^{(i)}$, $i \in \{1, \ldots, 2^m\}$, $K^{(i)} \in \{0, 1\}$ ergibt sich der Codebaum der Länge N dann auf herkömmliche Weise; Verzweigungen treten genau an den Stellen i mit $K^{(i)} = 1$ auf, und ein Pfad durch den Baum ist durch die Folge der Codeworte der äußeren Codes $(\boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)}, \ldots, \boldsymbol{c}^{(N)})$ bestimmt. Die Reihenfolge, in der die äußeren Codes betrachtet werden, ist damit prinzipiell vorgegeben

$$i := 1 \rightarrow 2 \rightarrow \ldots \rightarrow N,$$

wobei allerdings entsprechend des sequentiellen Decodierprinzips dynamisch entschieden wird, welcher Pfad verlängert wird.

Auch bei der Listendecodierung werden die äußeren Codes in dieser Reihenfolge betrachtet. Anstatt aber immer nur den Pfad der besten Hypothese zu verlängern, werden hier zu jedem Zeitpunkt i alle $L_{i-1} \in \mathbb{N}$ Pfade (L_i entspricht der Listengröße an der Stelle i) der in der Liste enthaltenen Hypothesen gemeinsam verlängert und von den sich dabei ergebenden $2^{K^{(i)}} \cdot L_{i-1}$ neuen Hypothesen nur die besten L_i behalten, alle anderen Hypothese werden verworfen und bei allen folgenden Decodierschritten nicht mehr betrachtet. Zudem gibt es bei diesem Verfahren nur eine Richtung, die dynamische Richtungsänderung der sequentiellen Decodierung ist hier nicht möglich.

Alle sequentiellen Decodieralgorithmen haben gemeinsam, daß Hypothesen unterschiedlicher Länge τ miteinander verglichen werden. Des weiteren wird bei der in [28], [71] vorgestellten Version zwar von einem notwendigerweise in der Größe begrenzten Speicher ausgegangen, allerdings kann dieser zumindest bei großem SNR so ausgelegt werden, daß nur selten die Grenze der Speicherkapazität erreicht wird und eine Hypothese verloren geht. Dagegen werden bei der Listendecodierung zum einen nur Hypothesen mit gleicher Anzahl von geschätzten Codesymbolen bzw. Informationssymbolen betrachtet, und auf der anderen Seite werden per Definition zu jedem Zeitpunkt eine gewisse Anzahl von Hypothesen verworfen, die für alle folgende Decodierschritte verloren sind.

5.3.1. Auf Wahrscheinlichkeit basierte Metrik

In Abschnitt 3.2.1 wurde gezeigt, daß das Kriterium des ML-Decoders bei einer Zerlegung in zwei äußere Codes U und V und wahrscheinlichkeitsbasierter Zuverlässigkeitsübergabe wie folgt geschrieben werden kann

$$\boldsymbol{c}_{\mathrm{ML}} = \arg\max_{\boldsymbol{u},\boldsymbol{v}} \prod_k (1 + v_k \cdot h_k^{(v)}) \prod_k (1 + u_k \cdot h_k^{(u)}).$$

Damit ist die ML-Entscheidungsregel als Produkt zweier Faktoren darstellbar. Ebenso kann die ML-Entscheidungsregel bei einer Zerlegung von C in 2^m äußere Codes $C^{(i)}$ der Länge $N^{(i)} = 1$ und rekursiver Berechnung der Zuverlässigkeitswerte $h_1^{(i)}$ nach Gl. (3.5) und Gl. (3.7) in 2^m Faktoren aufgespalten werden

$$\boldsymbol{c}_{\mathrm{ML}} = \arg \max_{\substack{\boldsymbol{c}^{(i)} \in C^{(i)} \\ i=1,\ldots,2^m}} \prod_i (1 + c_1^{(i)} \cdot h_1^{(i)}). \tag{5.15}$$

Zum Auswerten des obigen Produktes müssen alle 2^m Codeworte $\boldsymbol{c}^{(i)}$ bekannt sein. Bei einem sequentiellen Decodierverfahren müssen aber Hypothesen, die nur einen Teil des zu decodierenden Codewortes repräsentieren, auf die Wahrscheinlichkeit hin richtig zu sein, geprüft und bewertet werden. Falls nun durch eine solche Hypothese die ersten $\tau \leq 2^m$ Codeworte $\boldsymbol{c}^{(i)}$, $i \in \{1, \ldots, \tau\}$ festgelegt sind, ergibt sich wie im Anhang A.3 gezeigt wird, die zur FANO-Metrik äquivalente Metrik für die sequentielle Decodierung

$$\lambda_h(\boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)}, \ldots, \boldsymbol{c}^{(\tau)}) = \sum_{i=1}^{\tau} \log_2(1 + c_1^{(i)} \cdot h_1^{(i)}) - K^{(i)},$$

in der die ersten τ Faktoren aus Gl. (5.15) enthalten sind. Diese Metrik wächst wie die FANO-Metrik mit zunehmender Wahrscheinlichkeit der Hypothese und ist sowohl für Übertragung über den BSC als auch über den AWGN-Kanal gültig. Sie gleicht auf den ersten Blick der im Abschnitt A.2 vorgestellten Schreibweise der FANO-Metrik, doch diese geht wie schon erwähnt davon aus, daß die Codesymbole c_k der Reihe nach geschätzt werden. Bei einer sequentiellen Decodierung der äußeren Codes $C^{(i)}$, $1 \leq i \leq N$ sind jedoch alle c_k so lange undefiniert, bis der letzte Code $C^{(N)}$ entschieden ist. Als ein wesentlicher Unterschied zur FANO-Metrik, deren Bias-Term R konstant ist, ergibt sich hier ein variabler Bias $K^{(i)}$.

Die hier vorgestellte Metrik eignet sich auch, um bei der Listendecodierung zu entscheiden, welche der $2^{K^{(i)}} L_{i-1}$ Hypothesen zum Zeitpunkt i zu den L_i wahrscheinlichsten gehören und

die Liste dementsprechend zu ordnen. Da allerdings immer nur Hypothesen mit gleicher Anzahl geschätzter äußerer Codes verglichen werden, reicht es aus, nur über die Terme $\log_2(1 + c_1^{(i)} \cdot h_1^{(i)})$ zu summieren, die $K^{(i)}$ können weggelassen werden. Es ergibt sich damit die Metrik für Listendecodierung[21]

$$\Lambda_h(\boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)}, \ldots, \boldsymbol{c}^{(\tau)}) = \sum_{i=1}^{\tau} \log_2(1 + c_1^{(i)} \cdot h_1^{(i)}).$$

5.3.2. Auf euklidischer Distanz basierte Metrik

In Abschnitt 3.2.2 wurde folgende ML-Entscheidungsregel bei einer Zerlegung in zwei äußere Codes vorgestellt

$$\boldsymbol{c}_{\mathrm{ML}} = \arg\min_{\boldsymbol{v},\boldsymbol{u}} \sum_{\{k:v_k \neq \widetilde{v}_k\}} w_k^{(v)} + \sum_{\{k:u_k \neq \widetilde{u}_k\}} w_k^{(u)}.$$

Genau wie im obigen Abschnitt ist es möglich, die ML-Entscheidungsregel bei rekursiver Zerlegung in 2^m äußere Codes der Länge $N^{(i)} = 1$ und rekursiver Berechnung der Zuverlässigkeitswerte $y_k^{(i)}$ der äußeren Codesymbole wie folgt zu schreiben

$$\boldsymbol{c}_{\mathrm{ML}} = \arg \min_{\substack{\boldsymbol{c}^{(i)} \in C^{(i)} \\ i \in \{1,\ldots,2^m\}}} \sum_{\{i:c_1^{(i)} \neq a_1^{(i)}\}} w_1^{(i)}, \tag{5.16}$$

mit $a_1^{(i)} = \operatorname{sign} y_1^{(i)}$ und $w_1^{(i)} = |y_1^{(i)}|$. Auch hier läßt sich eine Metrik für die sequentielle Decodierung angeben, in der zur Beurteilung einer Hypothese für die ersten τ Codeworte die ersten τ Summanden der ML-Entscheidungsregel berücksichtigt werden. Eine solche, auf der distanzbasierten Zuverlässigkeitsübergabe beruhenden Metrik für den AWGN-Kanal kann wie folgt geschrieben werden (s. Anhang A.4)

$$\lambda_y^{(\mathrm{AWGN})}(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(\tau)}) = \frac{-2}{\ln 2 \cdot \sigma_0^2} \sum_{\substack{\{i:c_1^{(i)} \neq a_1^{(i)}\} \\ i \leq \tau}} w_1^{(i)} + \sum_{i=1}^{\tau} \min\left(1, \frac{|y_1^{(i)}|}{\ln 2 \cdot \sigma_0^2}\right) - K^{(i)}$$

Es handelt sich hier um eine auf Grund von heuristischen Überlegungen definierte Metrik. Die entsprechende Metrik für Übertragung über den BSC ist durch

$$\lambda_y^{(\mathrm{BSC})}(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(\tau)}) = (\log_2 p - \log_2 q) \cdot \sum_{\substack{\{i:c_1^{(i)} \neq a_1^{(i)}\} \\ i \leq \tau}} w_1^{(i)} + \sum_{i=1}^{\tau} 1 - K^{(i)}$$

gegeben.

Falls diese Metrik bei der Listendecodierung eingesetzt wird, wird sinnvollerweise nur die Summe über die $w_1^{(i)}$ betrachtet und jeweils die zweite Summe vernachlässigt. Für den BSC ergibt

[21] Obwohl weder in [11] noch [12] explizit eine Metrik angegeben wurde, ist anzunehmen, daß dort diese Metrik verwendet wurde.

sich dies automatisch, da bei allen Hypothesen die Anzahl der geschätzten äußeren Codes gleich ist. Generell ist nach Abschnitt 3.2.3 bei Zerlegung von C in zwei äußere Codes U und V und der Annahme, daß der Code U ein $(N/2, N/2, 1)$-Code sei, das zum wahrscheinlichsten Codewort $\boldsymbol{c}_{\mathrm{ML}} \in C$ gehörende Wort $\boldsymbol{v} \in V$ dadurch bestimmt, daß es die Summe in Gl. (3.16) minimiert. Ein Vergleich mit der Metrik nach Gl. (A.6) ergibt, daß sich auch hier alle Hypothesen nur in der vorderen Summe unterscheiden und damit für beliebige τ die Hypothese $(\widehat{\boldsymbol{c}}^{(1)}, \widehat{\boldsymbol{c}}^{(2)}, \ldots, \widehat{\boldsymbol{c}}^{(\tau)})$, die zum wahrscheinlichsten Codewort $\in \{C : \boldsymbol{c}^{(i)} = \pm 1, i > \tau\}$ gehört, allein durch das Auswerten der Summe über die $w_1^{(i)}$, $i \leq \tau$ bestimmt werden kann[22]. So ergibt sich die Metrik für Listendecodierung zu

$$\Lambda_y(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(\tau)}) = \sum_{\substack{\{i : c_1^{(i)} \neq a_1^{(i)}\} \\ i \leq \tau}} w_1^{(i)}, \tag{5.17}$$

die sowohl für den AWGN-Kanal als auch den BSC geeignet ist.

5.3.3. Vergleich zwischen Listen- und sequentieller Decodierung

Die Wortfehlerraten bei sequentieller (Stack-) Decodierung und Verwendung der hier vorgestellten Metriken λ_h und $\lambda_y^{(\mathrm{AWGN})}$ im Vergleich zur bitweisen MSD und zur ML-Decodierung[23] sind in Bild 5.5 gezeigt. Auf der einen Seite kann für den hier beispielhaft untersuchten RM-Code $\mathrm{RM}(3, 7)$ mit Parametern $(N, K, D) = (128, 64, 16)$ gegenüber der bitweisen MSD ein zusätzlicher Codiergewinn von mehr als 1 dB bei WER $= 10^{-4}$ erzielt werden, und dies praktisch unabhängig von der verwendeten Metrik. Auf der anderen Seite weichen die erreichten Wortfehlerraten auch bei Verwendung der Metrik λ_h, die entsprechend den Überlegungen von MASSEY für Variable-Length-Codes (deut.: Codes mit variabler Länge) [35] hergeleitet wurde, deutlich von der theoretisch möglichen WER des ML-Decoders ab. Das gleiche Verhalten, dessen Ursache nicht in der Speicherbegrenzung liegt, wurde u.a. auch schon in [1] für den Fall der sequentiellen Decodierung von Blockcodes beobachtet. Dort lieferte eine modifizierte Metrik mit dem konstantem Bias-Term R bessere Wortfehlerraten als die nach MASSEY hergeleitete Metrik für Variable-Length-Codes mit dem variablem Bias $K^{(i)}$. Bei dem hier vorgestellten sequentiellen Decodierverfahren ergibt die Verwendung eines konstanten Bias jedoch keine wesentlich veränderten Fehlerraten. Wie später noch gezeigt wird, scheint die Metrik λ_h allgemein zu optimistisch zu sein, so daß lange Pfade zu gut bewertet werden.

Wesentlich bessere Ergebnisse werden allerdings mit Hilfe eines Listendecodierverfahrens erreicht. Die WER bei Listendecodierung sind in Bild 5.6 gezeigt, hier wurde für alle Zeitpunkte τ die gleiche Listengröße gewählt, d.h. $L_\tau = L, \forall \tau$. Schon bei $L = 20$ ist damit für praktisch alle Wortfehlerraten das dazu notwendige SNR nur um wenige zehntel Dezibel größer als bei optimaler SDML-Decodierung. Auch hier ergeben sich, wie bei allen vorherigen Beispielen, für beide Zuverlässigkeitsübergaben bzw. für beide Metriken Λ_h und Λ_y fast die gleichen Ergebnisse. Insgesamt erweist sich damit, zumindest für diesen Code, die Listendecodierung als das

[22] Auch wenn die Metrik (A.6) aus den im Abschnitt A.4 erläuterten Gründen nicht für die sequentielle Decodierung geeignet ist, so wurde sie doch ausgehend von Gl. (A.4) unter der Vorgabe hergeleitet, die Hypothesen entsprechend der Wahrscheinlichkeit des dazugehörigen wahrscheinlichsten Codewortes $\boldsymbol{c}'_{\mathrm{ML}}$, $\boldsymbol{c}' \in \{C : \boldsymbol{c}^{(1)} = \widehat{\boldsymbol{c}}^{(1)}, \ldots, \boldsymbol{c}^{(\tau)} = \widehat{\boldsymbol{c}}^{(\tau)}, \boldsymbol{c}^{(i)} = \pm 1, i > \tau\}$ zu ordnen.

[23] Genau genommen handelt es sich hier um eine durch Simulation gewonnene untere Schranke für die WER des ML-Decoders, die allerdings sehr genau ist.

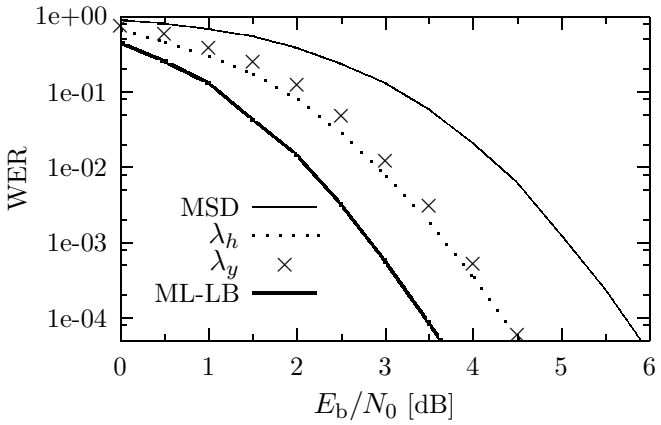

Abbildung 5.5.: Vergleich der sequentiellen Decodierung mit der bitweisen MSD sowie einer unteren Schranke für Soft-Decision ML-Decodierung (ML-LB) für den Code $\mathrm{RM}(3,7)$ bei Übertragung über den AWGN-Kanal.

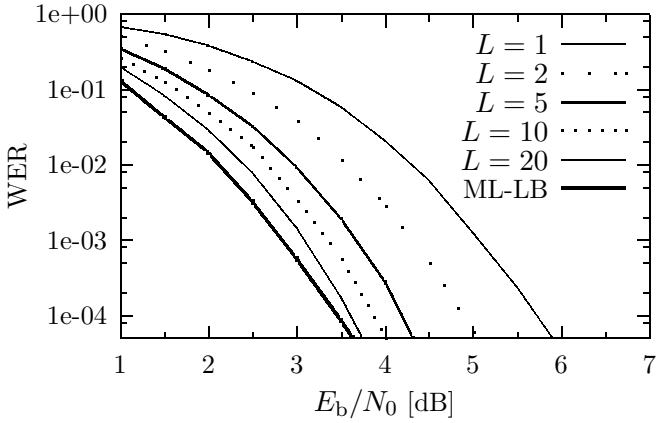

Abbildung 5.6.: WER für den Code $\mathrm{RM}(3,7)$ bei Listendecodierung mit verschiedenen Listengrößen L und eine untere Grenze bei Soft-Decision ML-Decodierung (ML-LB), alles für den AWGN-Kanal

geeignetere und leistungsfähigere Decodierverfahren. Die Verminderung des notwendigen SNR gegenüber der bitweisen MSD muß allerdings mit einer erhöhten Decodierkomplexität erkauft werden. So erhöht sich der Rechenaufwand um den Faktor L, so daß sich in erster Näherung eine Gesamtkomplexität der Ordnung $\mathcal{O}(LN \log N)$ ergibt, wenn der zusätzliche Aufwand zum Sortieren der Listenelemente vernachlässigt wird.

Die Wortfehlerrate des Listendecoders mit $L_\tau = 20, \forall \tau$ liegt, wie oben gezeigt, nahe an der des ML-Decoders, der allgemein durch $L_\tau = \infty, \forall \tau$ gegeben ist. Es zeigt sich, daß es nicht sinnvoll ist, die Listengröße L_τ zu allen Zeitpunkten i auf den gleichen Wert zu begrenzen. Zur Untersuchung der notwendigen Listengröße zu den verschiedene Zeitpunkten eignet sich die additive Metrik Λ_y sehr gut, da für den quadratischen euklidischen Abstand zwischen Empfangsvektor $\boldsymbol{y}$ und jedem Codewort $\boldsymbol{c}'$, daß für die ersten τ äußeren Codeworte mit einer Hypothese $(\widehat{\boldsymbol{c}}^{(1)}, \widehat{\boldsymbol{c}}^{(2)}, \ldots, \widehat{\boldsymbol{c}}^{(\tau)})$ übereinstimmt, nach Gl. (3.13) gilt

$$d_\mathrm{E}^2(\boldsymbol{y}, \boldsymbol{c}') \geq d_\mathrm{bias}^2 + 4 \cdot \Lambda_y(\widehat{\boldsymbol{c}}^{(1)}, \widehat{\boldsymbol{c}}^{(2)}, \ldots, \widehat{\boldsymbol{c}}^{(\tau)}), \quad \forall \boldsymbol{c}'$$

mit $\boldsymbol{c}' \in C' = \{C : \boldsymbol{c}^{(1)} = \widehat{\boldsymbol{c}}^{(1)}, \ldots, \boldsymbol{c}^{(\tau)} = \widehat{\boldsymbol{c}}^{(\tau)}\}$ und $d^2_{\text{bias}} = \sum_{k=1}^{N}(y_k - a_k)^2$. Damit kann während des Decodiervorganges zu jedem Zeitpunkt eine untere Grenze für den quadratischen euklidischen Abstand zwischen dem Empfangsvektor und allen Decodierentscheidung $\widehat{\boldsymbol{c}}$, die bei den ersten τ äußeren Codes mit einer bestimmten Hypothese übereinstimmen, angegeben werden. Ein Genie-Aided Decoder[24] (deut: Geist-unterstützter Decoder), der den wahren quadratischen Abstand $d^2_{\text{E}}(\boldsymbol{y}, \boldsymbol{c})$ zwischen Sendevektor $\boldsymbol{c}$ und Empfangsvektor $\boldsymbol{y}$ kennt, kann nun alle Hypothesen vernachlässigen, für die

$$4 \cdot \Lambda_y(\widehat{\boldsymbol{c}}^{(1)}, \widehat{\boldsymbol{c}}^{(2)}, \ldots, \widehat{\boldsymbol{c}}^{(\tau)}) > d^2_{\text{E}}(\boldsymbol{y}, \boldsymbol{c}) - d^2_{\text{bias}} \tag{5.18}$$

gilt, da diese Hypothesen nicht zum richtigen Codewort und auch nicht zum ML-Codewort gehören können[25]. Ein Genie-Aided Decoder, der nun zu jedem Zeitpunkt τ die Listengröße L_τ in Abhängigkeit von $\boldsymbol{y}$ und $\boldsymbol{c}$ dynamisch so wählt, daß genau die Hypothesen, die Gl. (5.18) erfüllen, aus der Liste herausfallen, alle anderen aber in der Liste verbleiben, ist damit ein ML-Decoder. Diese minimale Listengröße, die die ML-Decodierung garantiert, wird im folgenden mit $L_\tau^{(\text{ML})}(\boldsymbol{y}, \boldsymbol{c})$ bezeichnet. Weiterhin entscheidet sich ein Genie-Aided Decoder, dessen Listengröße durch einen maximalen Wert $L_\tau^{(\max)}$ nach oben begrenzt ist und der daher die Listengröße zu

$$L_\tau = \min\{L_\tau^{(\text{ML})}(\boldsymbol{y}, \boldsymbol{c}), L_\tau^{(\max)}\}$$

wählt, genau dann nicht für das ML-Codewort (bzw. gesendete Codewort), wenn sich auch der Decoder mit $L_\tau = L_\tau^{(\max)}, \forall \tau$ nicht für das ML-Codewort (bzw. gesendete Codewort) entscheidet. Beide Decoder sind daher bezüglich ihrer WER äquivalent[26].

Exemplarisch ist in Bild 5.7 für den Code $\text{RM}(3, 7)$ und einer rekursiven Zerlegung in 64 äußere Wiederholcodes[27] die mittlere Listengröße[28] $\overline{L}_\tau^{(\text{ML})}$ und die mittlere Anzahl der betrachten Pfade in Abhängigkeit von τ für verschiedene E_b/N_0-Werte gezeigt. Neben der Abhängigkeit der mittleren erforderlichen Listengröße vom Signal-zu-Rauschverhältnis ist der Unterschied zwischen $\tau = 22$, $\tau = 23$ und $\tau = 24$ markant (die ermittelten Werte bei $E_b/N_0 = 2$ dB sind $\overline{L}_{\tau=22}^{(\text{ML})} \approx 23.7$, $\overline{L}_{\tau=23}^{(\text{ML})} \approx 11.7$ und $\overline{L}_{\tau=24}^{(\text{ML})} \approx 7.5$). Da sich die $K = 64$ Informationssymbole für den hier betrachteten Code zwischen Codes U und V aufteilen zu[29] $K^{(v)} = 22$ und $K^{(u)} = 42$ ergibt sich, daß nur zu Beginn bei der Decodierung von V eine große Liste erforderlich ist und daß ferner, falls das Codewort $\boldsymbol{v} \in V$ bekannt ist, U mit einer relativ kleinen Liste entschieden werden kann. Damit ist der von LUCAS et al. in [33] vorgeschlagene Listendecoder mit $L = |\mathcal{L}_V| = |\mathcal{L}_U|$ sehr ineffektiv, da hier nur $|\mathcal{L}_V|$ verschiedene Codeworte $\boldsymbol{v} \in V$ generiert werden, aber bis zu $|\mathcal{L}_V| \cdot |\mathcal{L}_U| = L^2$ verschiedene Codeworte $\boldsymbol{u} \in U$. Der größte Teil der

[24] Es handelt sich hierbei um einen abstrakten Decoder, dem ein 'allwissender' Geist hilfreiche Informationen liefert, die das Decodierproblem erleichtern. Dieses Modell vereinfacht in manchen Fällen die theoretische Betrachung.

[25] Für den AWGN-Kanal als auch den BSC gilt immer $d^2_{\text{E}}(\boldsymbol{y}, \boldsymbol{c}_{\text{ML}}) \leq d^2_{\text{E}}(\boldsymbol{y}, \boldsymbol{c})$.

[26] Falls die WER eines Decoders, der die Metrik Λ_y verwendet, mit Hilfe einer Monte-Carlo-Simulation ermittelt werden soll, läßt sich mit dieser Methode auch die Rechenzeit erheblich verkürzen.

[27] Die Metrik schreibt sich in diesem Fall

$$\Lambda_y(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(\tau)}) = \sum_{i \leq \tau} \sum_{\{k : c_k^{(i)} \neq a_k^{(i)}\}} w_k^{(i)}.$$

[28] Dies Größe wurde durch eine Monte-Carlo-Simulation ermittelt, die maximale Listengröße war dabei auf 1000 Hypothesen begrenzt.

[29] Es sind $V = \text{RM}(2, 6)$ und $U = \text{RM}(3, 6)$.

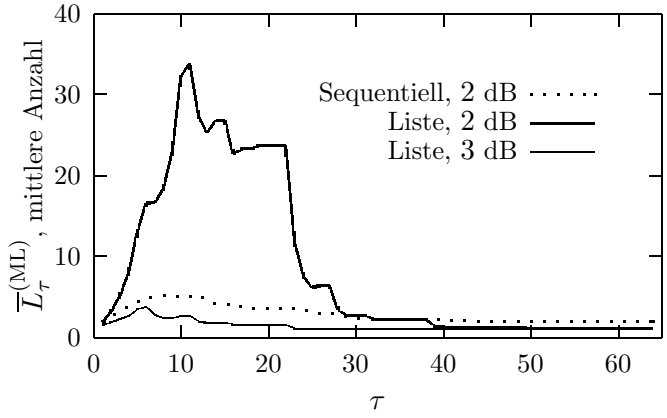

Abbildung 5.7.: Mittlere Listengröße $\overline{L}_\tau^{(\mathrm{ML})}$ bei Genie-Aided ML-Decodierung und mittlere Anzahl der verlängerten Pfade bei sequentieller Decodierung mit der Metrik λ_h für verschiedene E_b/N_0-Werte, alle für den Code $\mathrm{RM}(3,7)$ und Übertragung über den AWGN-Kanal

Rechenoperationen wird hier an unnötiger Stelle durchgeführt, sinnvoller wäre hier die Wahl $|\mathcal{L}_V| \gg |\mathcal{L}_U|$.

Des weiteren ist für das beispielhaft gewählte SNR von $E_b/N_0 = 2$ dB die mittlere Anzahl der untersuchten Pfade bei sequentieller Decodierung wesentlich geringer als die mittlere Listengröße, die für die ML-Decodierung erforderlich ist. Daraus resultiert, daß durch die in Abschnitt 5.3.1 angegebene Metrik die kurzen Pfade zu schlecht und die langen Pfade zu gut bewertet werden, so daß der Decoder nicht lange genug im vorderen Teil des Codebaumes verweilt und zu schnell dem Ende entgegen läuft.

Für den realen Decoder ergeben sich daraus verschiedene Möglichkeiten, die Listengröße L_τ zu wählen:

- Eine direkte Methode besteht in der vom SNR abhängigen Wahl $L_\tau = \kappa \cdot L_\tau^{(\mathrm{ML})}$ mit einem frei wählbaren Parameter $\kappa > 0$, wobei die Leistungsfähigkeit, aber auch die Komplexität des Decoders mit κ wächst.

- Anderseits kann aber auch das Prinzip der Bounded-Distance Decodierung (deut.: begrenzte Distanz-Decodierung) in die Wahl der Listengröße einfließen und so z.B. nur die die Hypothesen weiter verwendet werden, für die mit einer gewählten oberen Grenze $d_{\max}^2$

$$4 \cdot \Lambda_y(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(\tau)}) + d_{\text{bias}}^2 \overset{!}{\leq} d_{\max}^2$$

  gilt. Z.B. für $d_{\max} = d_{\mathrm{E,min}}(C)/2$ ergibt sich daraus ein BMD-Decoder (der bei der Wahl $L_\tau = 1, \forall i$ gegeben ist). Allgemein wachsen auch hier die Leistungsfähigkeit sowie die Decodierkomplexität mit $d_{\max}^2$. Für $d_{\max} \geq d_{\mathrm{E,min}}(C)$ ergibt diese Methode jedoch 'paradoxerweise' eine mit dem SNR *wachsende* mittlere Listengröße, was sich einfach erklären läßt, da z.B. für fehlerfrei empfangenes $\boldsymbol{y}$ alle minimalgewichtigen Codeworte in der List bleiben, wogegen es bei gestörtem Empfangsvektor nur wenige Codeworte gibt, die die obige Forderung erfüllen. Damit erscheint diese Methode wenig geeignet zu sein.

- Bei Verwendung der Metrik Λ_h kann z.B., da diese mit der Wahrscheinlichkeit verknüpft ist, $P(\boldsymbol{y}|\boldsymbol{c}^{(1)},\ldots,\boldsymbol{c}^{(\tau)})$ nach unten beschränkt werden. Mit Gl. (A.2) kann damit für alle Hypothesen gefordert werden[30]

$$P(\boldsymbol{y}|\boldsymbol{c}^{(1)},\ldots,\boldsymbol{c}^{(\tau)}) = \prod_k P(y_k) \cdot \prod_{i\le\tau}(1 + c_1^{(i)} h_1^{(i)}) \ge P_{\min}$$

mit einer unteren Grenzwahrscheinlichkeit $P_{\min}$. Dies ist gleichwertig zur Forderung

$$\sum_k \log_2 P(y_k) + \Lambda_h(\boldsymbol{c}^{(1)},\ldots,\boldsymbol{c}^{(\tau)}) \ge \log_2 P_{\min}$$

Diese Methode erscheint allerdings aus dem gleichem Grund wie die 2. Methode wenig geeignet.

5.3.4. Mittlere Listengröße für HD-ML-Decodierung von REED-MULLER Codes

Wie im Anhang A.10 gezeigt wird, kann bei Übertragung über den BSC für eine Zerlegung in zwei äußere Codes U und V die mittlere Größe der Liste $L_v^{(\mathrm{ML})}(\boldsymbol{y},\boldsymbol{c})$, d.h. die mittlere Anzahl der Hypothesen in der Liste für V bei Genie-Aided Decodierung, berechnet werden. Die damit gewonnenen Ergebnisse für den $\mathrm{RM}(3,7)$-Code sind zusammen mit simulierten[31] Werten in Bild 5.8 gezeigt. Die dazu erforderliche Gewichtsverteilung für den Code $V = \mathrm{RM}(2,6)$ kann gemäß [53] bestimmt werden. Für $E_\mathrm{b}/N_0 > 4$ dB ist die erforderliche Listengröße relativ klein, sie steigt jedoch mit sinkendem SNR stark an. Bei kleinem SNR ist die Größe $\overline{L}_v^{(\mathrm{ML})}$ für diesen Code allerdings kein geeignetes Maß für die Dimensionierung der Listengröße eines realen Decoders, da z.B. die Wortfehlerrate des ML-Decoders bei Übertragung über den BSC und $E_\mathrm{b}/N_0 = 3$ dB größer als 0.1 ist und somit die Wahl dieses SNR nicht sinnvoll ist. Die mittlere Listengröße wird bei kleinem SNR in großem Maße von den Störungsvektoren $\boldsymbol{n}$ bestimmt, die die Korrekturfähigkeit des Codes überschreiten und bei denen eine fehlerfreie Decodierung unmöglich ist. Andererseits ist in den SNR-Bereichen mit $\mathrm{WER}_\mathrm{ML} \le 10^{-3}$ die erforderliche Listengröße klein genug, so daß bei diesem Code für beide in dieser Arbeit betrachteten Kanäle mit einem realen Decoder eine Fehlerrate nahe der des ML-Decoders erreichbar ist.

Beim Code $\mathrm{RM}(4,9)$ mit Parametern $(512,256,32)$ liegen die Verhältnisse dagegen anders. Wie in der Ergänzung im Abschnitt A.10 gezeigt, ist es bei rekursiver Zerlegung in vier äußere Codes $C^{(1)},\ldots,C^{(4)}$ ebenfalls möglich, die mittlere erforderliche Listengröße $\overline{L}_{C^{(1)}}^{(\mathrm{ML})}$ zu bestimmen. Auch hier kann die dazu erforderliche Gewichtsverteilung des Codes $C^{(1)} = \mathrm{RM}(2,7)$ gemäß [53] bestimmt werden. Ein Vergleich mit Bild 5.11 ergibt, daß das Anwachsen der mittleren Listengröße für $E_\mathrm{b}/N_0 < 5$ dB nicht wie beim $\mathrm{RM}(3,7)$-Code durch nichtkorrigierbare

[30] Z.B. für den AWGN-Kanal ist die Wahrscheinlichkeitsdichte gegeben durch

$$f(y_k) = \frac{1}{2} \cdot \frac{1}{\sqrt{2\pi\sigma_0^2}} \left(e^{-\frac{(y_k-1)^2}{2\sigma_0^2}} + e^{-\frac{(y_k+1)^2}{2\sigma_0^2}} \right)$$

[31] Bei dieser Monte-Carlo Simulation wurde die Listengröße auf maximal 10.000 Hypothesen begrenzt.

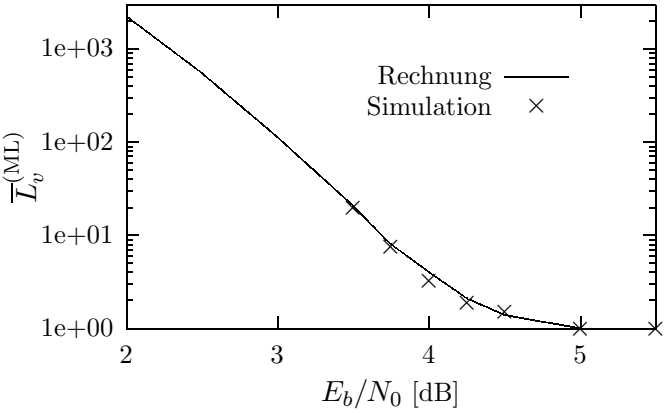

Abbildung 5.8.: Mittlere Listengröße $\overline{L}_v^{(\mathrm{ML})}$ für den Code $\mathrm{RM}(3,7)$ und Übertragung über den BSC

Störungsvektoren verursacht wird. Es handelt sich hier vielmehr um ein strukturelles Problem[32] des Zusammenspiels zwischen Code und Decoder bei langen Codes. Für kurze RM-Codes ist eine unabhängige ML-Decodierung der beiden Codes U und V nicht viel schlechter als eine ML-Decodierung des Gesamtcodes, wogegen bei großen Codelängen eine gemeinsame Decodierung notwendig ist. Zudem fällt auf, daß die SNR-Schwelle, unter der die Listengröße ansteigt, im Vergleich zu Bild 5.8 zu größeren SNR-Werten hin verschoben ist. Dieses auch für den AWGN-Kanal beobachtbare Phänomen, daß für lange Codes bei fester Rate R die mittlere minimal erforderliche Listengröße zur ML-Decodierung mit der Codelänge wächst, geht konform mit der Erkenntnis, daß bei festem SNR und fester Coderate die Fehlerwahrscheinlichkeit des REED-Decoders sowie des Decoders in [49] asymptotisch mit der Codelänge zunimmt [9]. Obwohl der in diesem Abschnitt vorgestellte Listendecoder im allgemeinen wesentlich geringere Wortfehlerraten erreicht als der Decoder nach REED, ist es daher auch mit diesem Verfahren und praktikablen Listengrößen prinzipiell nicht möglich, bei langen RM-Codes Fehlerraten nahe der des ML-Decoders zu erzielen. Auch der hier vorgestellte Listendecoder scheint bei fester Coderate und fester maximaler Listengröße für wachsende Codelängen asymptotisch schlecht zu sein.

Damit kann mit dem Listendecoder im allgemeinen nur bei relativ kurzen Codes ($N \leq 256$) oder relativ kleiner bzw. großer Coderate eine Fehlerrate nahe der des ML-Decoders erreicht werden. Ab Codelängen ≥ 512 Codesymbole wird für RM-Codes mit Coderaten $R = 0.5$ der Unterschied zwischen der theoretisch möglichen WER und der mit Listendecodierung praktisch

[32] Dieses Problem kann auch anschaulich wie folgt erklärt werden: bei z.B. $E_\mathrm{b}/N_0 = 3.5$ dB ist $p = 0.067$ und damit ergibt sich mit der Codelänge $N = 512$ eine mittlere Anzahl der Fehler im Empfangsvektor $\boldsymbol{y}$ von $\overline{e} = 34.5$ Fehlern. Die Wahrschleinlichkeit, daß 35 Fehler aufgetreten sind, liegt bei 7% und in 10% der Fälle sind bei $e = 35$ Empfangsfehlern und Zerlegung in vier äußere Codes im Vektor $\boldsymbol{y}^{(1)}$ noch $e^{(1)} = 33$ Fehler enthalten. Wenn man nun berücksichtigt, daß es sich bei $C^{(1)}$ um einen Code mit Codelänge $N^{(1)} = 128$ und $K^{(1)} = 29$ handelt, und die GILBERT-VARSHAMOV-Distanz für diese Werte bei $d_{\mathrm{GV}}^{(1)} = 33$ liegt, bedeutet dies, daß selbt ein Genie-Aided Decoder, der die Anzahl der Fehler in $\boldsymbol{y}^{(1)}$ kennt, in 0.7% der Fälle alle Codeworte von $C^{(1)}$, die in der Kugel mit dem Radius $d_{\mathrm{GV}}^{(1)}$ um den Vektor $\boldsymbol{y}^{(1)}$ liegen, in der Liste behalten muß. Falls nun mehr als 35 Fehler aufgetreten sind, vergrößert sich der Radius der Kugel entsprechend. Da aber die mittlere Anzahl der Codeworte in der Liste mit dem überschreiten des Radius $d_{\mathrm{GV}}^{(1)}$ stark ansteigt, ist es damit praktisch unmöglich, alle Kanditaten in Liste zu behalten.

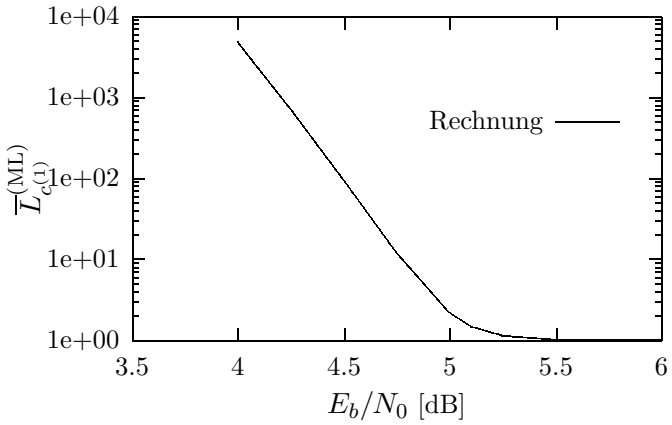

Abbildung 5.9.: Mittlere Listengröße $\overline{L}^{(\mathrm{ML})}_{c^{(1)}}$ für den Code $\mathrm{RM}(4,9)$ und Übertragung über den BSC

erreichbaren Fehlerrate immer größer, und dies für beide in dieser Arbeit betrachteten Kanäle.

Trotz der prinzipiellen Schwäche des Listendecodierverfahrens für $N \to \infty$ ermöglicht es die im folgenden Abschnitt vorgestellte Erweiterung des Algorithmus auch für Codes mittlerer Länge, die aufgrund der zu großen erforderlichen Listengröße mit der oben beschriebenen Methode nur schlecht decodiert werden können, bei Übertragung über den BSC fast optimale Fehlerraten zu erzielen.

5.4. Permutationsdecodierung für REED-MULLER Codes

Das wesentliche Problem im Zusammenspiel zwischen Code und Decoder bei der Decodierung von RM-Codes ist, wie oben beschrieben, die sehr große erforderliche Listengröße zu Beginn der Decodierung, da z.B. beim BSC die Anzahl der Fehler in $\boldsymbol{y}^{(1)}$ häufig nahe oder sogar über der Distanz d_{GV} des Codes $C^{(1)}$ liegt (siehe Fußnote 32 auf Seite 58). Bei gegebener Anzahl e der aufgetretenen Fehler in $\boldsymbol{y}$ hängt die Anzahl $e^{(1)}$ der Fehler in $\boldsymbol{y}^{(1)}$ jedoch von der Position im Vektor $\boldsymbol{y}$ ab, da z.B. bei Übertragung über den BSC nach Gl. (3.3) die Schätzung $\widetilde{v}_k$ nur dann falsch ist, wenn entweder a_k oder $a_{k+N^{(v)}}$ falsch ist, nicht aber, wenn beide Werte falsch sind. Allgemein gilt $e^{(1)} \leq e$ und bei günstiger Anordnung der e Fehler kann $e^{(1)}$ wesentlich kleiner als e sein, bei ungünstiger Anordnung jedoch kann sogar $e^{(1)} = e$ gelten[33]. Da RM-Codes eine sehr symmetrische Struktur besitzen, ist die Gruppe der Permutationen, durch die Codeworte in andere Codeworte des gleichen Codes abgebildet werden, relativ groß und es bietet sich an, diejenige Permutation zu suchen, bei der $e^{(1)}$ klein ist. Bereits in [52] wurde ein Algorithmus zur Decodierung von RM-Codes vorgestellt, bei dem verschiedene Permutationen des Empfangsvektors betrachtet werden.

Nach Abschn. 4.1 ist die Automorphismus-Gruppe für RM-Codes durch die Gruppe der affinen Transformationen $\mathrm{GA}(m)$ gegeben. Wie dort vorgestellt, gilt für die Permutation Π, durch die

[33] Dies gilt natürlich nur unter der Bedingung $e \leq N^{(1)}$.

$\boldsymbol{y}$ in $\boldsymbol{y}^{(p)}$ mit $y_k^{(p)} = y_{\Pi(k)}$ abgebildet wird[34]

$$\Pi(k) = j \quad \text{falls} \quad \boldsymbol{x}_k = \boldsymbol{A} \cdot \boldsymbol{x}_j + \boldsymbol{b}.$$

Zum einen führen für einen gegebenen RM-Code allerdings nicht alle Permutationen notwendigerweise zu verschiedenen Codewörtern, durch viele Permutationen werden die Codeworte auf sich selbst abgebildet. Zum anderen sind in der Menge P_{GA} der Permutationen, die ein Codewort in ein anderes abbilden, nur diejenigen von Interesse, durch die bei Zerlegung in äußere Codes $C^{(1)}, C^{(2)}, \ldots$ die Anzahl der Fehler $e^{(1)}$ im Vektor $\boldsymbol{y}^{(1)}$ variiert.

Im ersten Schritt wird im folgenden eine Zerlegung in zwei äußere Codes U und V betrachtet. Nach Gl. (3.3) ergeben sich die Komponenten von $\widetilde{\boldsymbol{v}} = (\widetilde{v}_1, \widetilde{v}_2, \ldots, \widetilde{v}_{N^{(v)}})$, falls die $\boldsymbol{x}_k$ wie in Kapitel 4 beschrieben gemäß ihrer binären Darstellung geordnet sind, zu[35]

$$\widetilde{v}_k = a_k \cdot a_{N^{(v)}+k}$$

mit $a_k = \operatorname{sign} y_k$. Die Schätzung $\widetilde{\boldsymbol{v}}^{(p)}$, die sich aus dem Vektor $\boldsymbol{y}^{(p)}$ ergibt, bestimmt sich in gleicher Weise zu

$$\widetilde{v}_k^{(p)} = a_k^{(p)} \cdot a_{N^{(v)}+k}^{(p)} = a_{\Pi(k)} \cdot a_{\Pi(N^{(v)}+k)}. \tag{5.19}$$

Falls nun nach der Permutation bei der Bestimmung der Symbole $\widetilde{v}_k^{(p)}$ paarweise die gleichen Werte a_k und $a_{N^{(v)}+k}$ multipliziert werden wie bei der Bestimmung von $\widetilde{v}_k$, gilt $e^{(v,p)} = e^{(v)}$ mit der Anzahl $e^{(v,p)}$ der Fehler in $\widetilde{\boldsymbol{v}}^{(p)}$. Allgemein gilt für jede Permutation Π, daß bei Wahl von $\boldsymbol{A}$ und $\boldsymbol{b}$ die $N^{(v)}$ Paare $(a_{\Pi(k)}, a_{\Pi(N^{(v)}+k)})$ durch folgendes zu erfüllende Gleichungssystem bestimmt sind

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{A} \cdot \boldsymbol{x}_{\Pi(k)} + \boldsymbol{b} \\ \boldsymbol{x}_{N^{(v)}+k} &= \boldsymbol{A} \cdot \boldsymbol{x}_{\Pi(N^{(v)}+k)} + \boldsymbol{b}. \end{aligned}$$

Für die hier angenommene Ordnung der $\boldsymbol{x}_k$ ergibt sich mit dem Vektor $\boldsymbol{x}_{N^{(v)}+1} = (-1, 1, 1, \ldots, 1)^T$ (s. auch Bsp. 4.1)

$$\boldsymbol{x}_{N^{(v)}+k} = \boldsymbol{x}_k + \boldsymbol{x}_{N^{(v)}+1}.$$

Damit kann das obige Gleichungssystem wie folgt geschrieben werden

$$\begin{aligned} \boldsymbol{x}_{\Pi(k)} &= \boldsymbol{A}^{-1} \cdot (\boldsymbol{x}_k - \boldsymbol{b}) \\ \boldsymbol{x}_{\Pi(N^{(v)}+k)} &= \underbrace{\boldsymbol{A}^{-1} \cdot (\boldsymbol{x}_k - \boldsymbol{b})}_{\boldsymbol{x}_{\Pi(k)}} + \boldsymbol{A}^{-1} \cdot \boldsymbol{x}_{N^{(v)}+1}, \end{aligned}$$

wobei $\boldsymbol{A}^{-1}$ die inverse Matrix zu $\boldsymbol{A}$ ist. Hieraus resultiert, daß die Bildung der Paare unabhängig von $\boldsymbol{b}$ ist (der Vektor $\boldsymbol{b}$ beeinflußt nur die Reihenfolge der Symbole in $\widetilde{\boldsymbol{v}}^{(p)}$, nicht aber deren Werte). Daher reicht es aus, nur die Gruppe der linearen Transformationen $\mathrm{GL}(m)$ und die sich damit ergebende Menge der Permutationen $P_{\mathrm{GL}} \subset P_{\mathrm{GA}}$ zu betrachten. Weiterhin ist für die Paarbildung nur die erste Spalte von $\boldsymbol{A}^{-1}$ von Bedeutung, da vom Vektor $\boldsymbol{x}_{N^{(v)}+1}$ nur die erste

[34] Rechnung in $\mathrm{GF}(2)^m$
[35] Rechnung in $\mathbb{R}$

Komponente gleich -1 ist, alle anderen Spalten haben genau wie $\boldsymbol{b}$ nur auf die Reihenfolge der Symbole $\widetilde{v}_k^{(p)}$ Einfluß. Für die Untermenge $P^* \subset P_{\mathrm{GL}}$ der Permutationen, die zu verschiedener Anzahl von Fehlern $e^{(v,p)}$ führen, folgt damit

$$|P^*| \leq 2^m - 1$$

da die erste Spalte von $\boldsymbol{A}^{-1}$ beliebig aber ungleich $\mathbf{1}$ gewählt werden kann. Daraus ergibt sich folgender Satz

Satz 5.3 Für einen REED-MULLER-Code der Länge $N = 2^m$ gibt es insgesamt nur $2^m - 1$ verschiedene Möglichkeiten, die $N^{(v)} = 2^{m-1}$ Paare $(a_{\Pi(k)}, a_{\Pi(N^{(v)}+k)})$ in Gl. (5.19) zu bilden, und unter allen $2^m - 1$ Permutationen mit verschiedenen Paaren gibt es für $k \neq j$ genau eine Permutation, bei der die Symbole a_k und a_j ein Paar bilden. Jedes Symbol wird genau einmal mit jedem anderen Symbol kombiniert. □

Eine Möglichkeit, die $2^m - 1$ verschieden Matrizen $\boldsymbol{A}$ zu bilden, die die Permutationen mit den $2^m - 1$ verschieden Paaren definieren, ist folgende:

- Betrachtet wird zuerst das erste Symbol a_1 mit dem dazugehörigen Vektor $\boldsymbol{x}_1 = (1, 1, \ldots, 1)$. Mit $\boldsymbol{b} = \mathbf{1}$ ist $\boldsymbol{A} \cdot \boldsymbol{x}_1 = \boldsymbol{x}_1$ und damit ist $\Pi(1) = 1$. Das erste Symbol wird daher nicht permutiert.

- Damit nach der Permutation a_1 und a_k, $2 \leq k \leq 2^m$ ein Paar bilden, d.h. $\Pi(N^{(v)} + 1) = k$, muß gelten

  $$\boldsymbol{A} \cdot \boldsymbol{x}_k = \boldsymbol{x}_{N^{(v)}+1} = (-1, 1, 1, \ldots, 1)^T.$$

  Gesucht ist also eine Matrix $\boldsymbol{A}$, die bei gegebenen a_k, bzw. $\boldsymbol{x}_k = (x_m, x_{m-1}, \ldots, x_1)^T$, $k \neq 1$ die obige Gleichung erfüllt.

- Für den gewählten Vektor $\boldsymbol{x}_k$ gelte $x_\kappa = -1$ und $x_\mu = 1, \forall \mu > \kappa$. Ausgehend von der Einheitsmatrix wird nun die erste mit der $(m-\kappa+1)$-ten Spalte vertauscht und zur neuen $(m - \kappa + 1)$-ten Spalte der Vektor $(1, \ldots, 1, x_{\kappa-1}, \ldots, x_1)^T$ addiert.

Die daraus resultierende Matrix $\boldsymbol{A}$ erfüllt obige Gleichung, wie sich leicht zeigen läßt.

Beispiel 5.2 Für $m = 5$ und $\boldsymbol{x}_k = (1, 1, -1, 1, -1)^T$ ergibt sich $\kappa = 3$ und mit $(m-\kappa+1) = 3$ nach obiger Konstruktion folgende Matrix $\boldsymbol{A}$

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

wobei hier die erste mit der dritten Spalte der Einheitsmatrix vertauscht und sowie der Vektor $(1, 1, 1, 1, -1)^T$ zur neuen dritten Spalte addiert wurde. □

Bei rekursiver Zerlegung von $\mathrm{RM}(r,m)$ in vier äußere Codes $C^{(1)},\ldots,C^{(4)}$ ist die zu Gl. (5.19) äquivalente Gleichung zur Schätzung der $a_k^{(1)}$

$$a_k^{(1)} = a_k \cdot a_{N^{(1)}+k} \cdot a_{2N^{(1)}+k} \cdot a_{3N^{(1)}+k},$$

d.h. hier werden vier Symbole a_k miteinander multipliziert, um einen Schätzwert zu erhalten. Wie im Anhang A.11 gezeigt, gilt für die Menge P der Permutationen, die verschiedene Gruppen $(a_{\Pi(k)},\ldots,a_{\Pi(3N^{(1)}+k)})$ ergeben

$$|P| = \frac{(2^m-2^0)(2^m-2^1)}{(2^2-2^0)(2^2-2^1)}.$$

Eine Verallgemeinerung dieser Ergebnisse liefert folgender Satz

Satz 5.4 Für einen REED-MULLER-Code der Länge $N=2^m$ und eine Zerlegung in 2^ρ äußere Codes $C^{(1)},\ldots,C^{(2^\rho)}$ gibt es insgesamt

$$|P| = \frac{(2^m-2^0)(2^m-2^1)\cdots(2^m-2^{\rho-1})}{(2^\rho-2^0)(2^\rho-2^1)\cdots(2^\rho-2^{\rho-1})}$$

verschiedene Möglichkeiten, die verschieden Gruppen aus jeweils 2^ρ Symbolen a_k zur Bestimmung der Werte $a_k^{(1)}$ zu bilden.

Beweis: Jede Permutation entspricht einer μ-Fläche mit $\mu=\rho$ in $\mathrm{EG}(m,2)$, die durch den Punkt $(1,1,\ldots,1)$ geht. Die Anzahl der Permutationen ist somit gleich zur Anzahl der ρ-Flächen durch den Ursprung. □

Auf den Code $\mathrm{RM}(4,9)$ angewendet bedeutet dies, daß es bei Zerlegung in zwei äußere Codes $|P| = (512-1) = 511$ und bei Zerlegung in vier äußere Codes $|P| = 511\cdot 510/6 = 43435$ verschiedene Möglichkeiten gibt, die Paare aus jeweils zwei bzw. vier Symbolen a_k zu bilden.

Für $\rho = 2$ können die verschiedenen Matrizen $\boldsymbol{A}$, die die $|P|$ dazugehörigen Permutationen definieren, wie folgt konstruiert werden, wenn auch hier $\boldsymbol{b}=\mathbf{1}$ gewählt wird:

- Ausgehend von zwei beliebigen Vektoren $\boldsymbol{x}_{k_1} = (x_m^{(k_1)},\ldots,x_1^{(k_1)})^T$ und $\boldsymbol{x}_{k_2} = (x_m^{(k_2)},\ldots,x_1^{(k_2)})^T$ mit den Eigenschaften $x_{\kappa_1}^{(k_1)} = -1$, $x_\mu^{(k_1)} = 1, \forall \mu > \kappa_1$ und $x_{\kappa_2}^{(k_2)} = -1$, $x_\mu^{(k_2)} = 1, \forall \mu > \kappa_2$ und $x_{\kappa_2}^{(k_1)} = 1$ sowie $\kappa_1 > \kappa_2$ werden zwei Matrizen $\boldsymbol{A}_1$ und $\boldsymbol{A}_2$ wie folgt gebildet:

- Ausgangspunkt jeder Matrix ist die Einheitsmatrix $\boldsymbol{E}$. Zur Bildung von $\boldsymbol{A}_1$ wird die erste mit der $(m-\kappa_1+1)$-ten Spalte von $\boldsymbol{E}$ vertauscht und zur neuen $(m-\kappa_1+1)$-ten Spalte der Vektor $(1,\ldots,1,x_{\kappa_1-1}^{(k_1)},\ldots,x_1^{(k_1)})^T$ addiert. In der gleichen Weise wird zur Bildung von $\boldsymbol{A}_2$ die zweite mit der $(m-\kappa_2+1)$-ten Spalte von $\boldsymbol{E}$ vertauscht und zur neuen $(m-\kappa_2+1)$-ten Spalte der Vektor $(1,\ldots,1,x_{\kappa_2-1}^{(k_2)},\ldots,x_1^{(k_2)})^T$ addiert.

- Die Matrix $\boldsymbol{A}$ ergibt sich nun durch das Produkt $\boldsymbol{A} = \boldsymbol{A}_2 \cdot \boldsymbol{A}_1$.

Es kann nun leicht gezeigt werden, daß folgendes Gleichungssystem erfüllt ist

$$\begin{aligned} \boldsymbol{x}_{2N^{(1)}+1} &= \boldsymbol{A} \cdot \boldsymbol{x}_{k_1} \\ \boldsymbol{x}_{N^{(1)}+1} &= \boldsymbol{A} \cdot \boldsymbol{x}_{k_2}, \end{aligned}$$

da gilt $\boldsymbol{A}_2 \cdot \boldsymbol{x}_{2N^{(1)}+1} = \boldsymbol{x}_{2N^{(1)}+1}$ und $\boldsymbol{A}_1 \cdot \boldsymbol{x}_{k_2} = \boldsymbol{x}_{k_2}$. Zu jedem κ_1 und κ_2 gibt es 2^{κ_1-2} verschiedene Vektoren $\boldsymbol{x}_{k_1}$ und 2^{κ_2-1} verschiedene Vektoren $\boldsymbol{x}_{k_2}$, die die oben geforderten Eigenschaften besitzen. Die Anzahl der verschiedenen Matrizen $\boldsymbol{A}$ ist damit

$$\sum_{\kappa_1=2}^{m} \sum_{\kappa_2=1}^{\kappa_1-1} 2^{\kappa_1-2} \cdot 2^{\kappa_2-1} = |P|$$

Beispiel 5.3 Für $m = 5$ und $\boldsymbol{x}_{k_1} = (1, -1, 1, 1, -1)^T$ und $\boldsymbol{x}_{k_1} = (1, 1, -1, -1, -1)^T$ ergeben sich mit $\kappa_1 = 4$, $(m - \kappa_1 + 1) = 2$ und $\kappa_2 = 3$, $(m - \kappa_2 + 1) = 3$ folgende Matrizen

$$\boldsymbol{A}_1 = \begin{bmatrix} 1 & -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 \\ 1 & 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

und

$$\boldsymbol{A}_2 = \begin{bmatrix} -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \end{bmatrix}.$$

□

Beispielhaft werden nun zum Schluß dieses Abschnitts die durch Listendecodierung verschiedener Permutationen erzielbaren Ergebnisse für den Code $\mathrm{RM}(4, 9)$ mit Parametern $(N, K, D) = (512, 256, 32)$ vorgestellt. Es wird von einer Übertragung über den BSC ausgegangen. Wie in Abb. 5.9 gezeigt, wächst für diesen Code die mittlere Listengröße zur ML-Decodierung zu Beginn des Decodiervorgangs sehr stark, falls das Signal-zu-Rauschverhältnis unter $E_\mathrm{b}/N_0 = 5$ dB fällt. Es werden daher im folgenden diejenigen Permutationen verwendet, die bei Zerlegung in vier äußere Codes verschiedene Gruppen bei der Berechnung der $\widetilde{v}_k^{(1)}$ ergeben. Weil jedoch die Menge P der verschiedenen Permutationen mit dieser Eigenschaft sehr groß ist ($|P| = 43435$), werden bei der Decodierung $n_p \leq |P|$ zufällig aus P gewählte Permutationen des Empfangsvektors $\boldsymbol{y}$ erzeugt und jede dieser Permutationen mit Hilfe eines Decoders[36] mit Listengröße $L_\tau = L, \forall \tau$ decodiert. Die Parameter L und n_p in Abb. 5.10 und Abb. 5.11 sind paarweise jeweils so gewählt, daß das Produkt $L \cdot n_p$ gleich und damit die Gesamtkomplexität der Decodierung auch in etwa gleich ist. Zum Vergleich ist auch die WER für den Decoder mit $L = 1$ und $n_p = 1$ gegeben. Es ergibt sich, daß für $n_p = 1$ selbst bei relativ großer Liste (hier $L = 1000$) die WER bei mehr als $e = 35$ Übertragungsfehlern stark ansteigt (hier sei nochmal auf Fußnote 32 auf Seite 58 hingewiesen). Damit ist es praktisch unmöglich, mit realisierbarem

[36] Hier wird nur der Decoder, der die Metrik Λ_y verwendet, betrachtet.

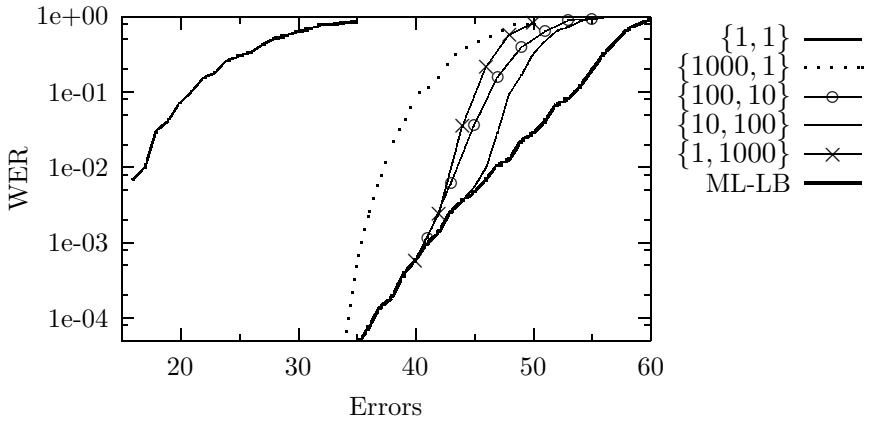

Abbildung 5.10.: Wortfehlerraten für den Code $\mathrm{RM}(4, 9)$ für verschiedene Listengrößen L und Anzahl der zufällig gewählten Permutationen n_p, bezeichnet mit $\{L, n_p\}$, in Abhängigkeit von den aufgetretenen Kanalfehlern

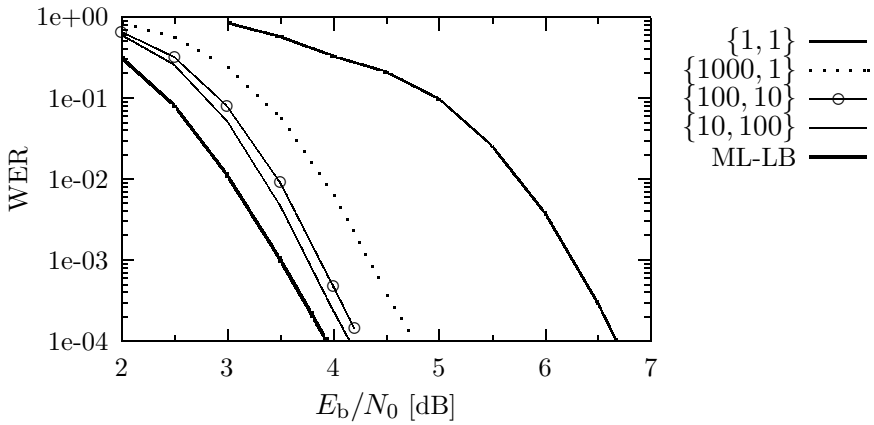

Abbildung 5.11.: Wortfehlerraten für den Code $\mathrm{RM}(4, 9)$ für verschiedene Listengrößen L und Anzahl der zufällig gewählten Permutationen n_p, bezeichnet mit $\{L, n_p\}$, alles für den BSC

Aufwand eine Fehlerrate nahe der des HDML-Decoders zu erzielen. Auch werden die besten Wortfehlerraten nicht mit dem $\{1, 1000\}$-Decoder erzielt, sondern durch eine Kombination von Listen- und Permutationsdecodierung. Die Wortfehlerraten für $L = 10$ und $n_p = 100$ sind bis ca. $e \leq 45$ Übertragungsfehler praktisch mit der des HDML-Decoders identisch[37].

Die gleichen Ergebnisse sind in Bild 5.11 in Abhängigkeit von E_b/N_0 gezeigt. In Übereinstimmung mit dem Resultat aus Abb. 5.9 steigt die WER des Decoders mit $L = 1000$ und $n_p = 1$ beim Unterschreiten von $E_\mathrm{b}/N_0 = 5$ dB stark an. Auf der anderen Seite ist für den $\{10, 100\}$-Decoder der Verlust an SNR gegenüber HDML-Decodierung für WER $= 10^{-4}$ nur etwa 0.2 dB. Insgesamt ergibt sich damit gegenüber der bitweisen MSD ($\{L = 1, n_p = 1\}$) ein zusätzlicher Codiergewinn von ca. 2.5 dB.

Die Wortfehlerrate bei Übertragung über den AWGN-Kanal ist in Bild 5.12 in Abhängigkeit

[37] Es sei hier darauf hingewiesen, daß es sich um einen Code mit Mindesthammingdistanz $d_{\mathrm{H,min}} = 32$ handelt und daher WER $= 0$ nur für $e \leq 15$ garantiert ist.

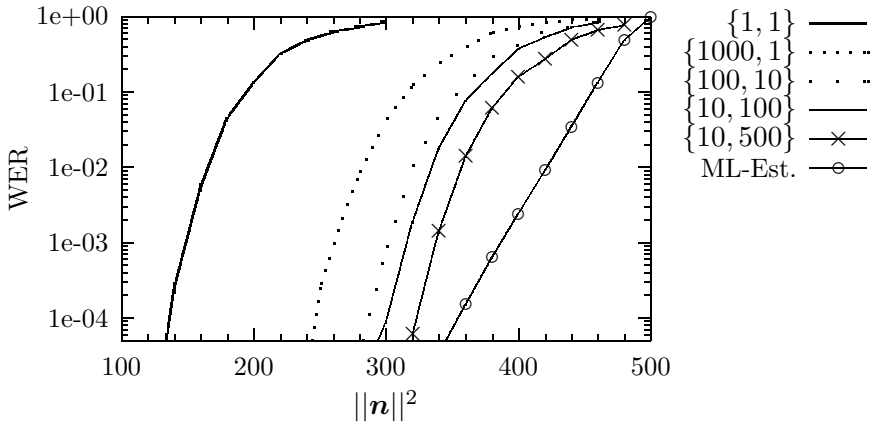

Abbildung 5.12.: Wortfehlerraten für den Code $\mathrm{RM}(4,9)$ für verschiedene Listengrößen L und Anzahl der zufällig gewählten Permutationen n_p, bezeichnet mit $\{L, n_p\}$ in Abhängigkeit von $||\boldsymbol{n}||^2 = d_{\mathrm{E}}^2(\boldsymbol{y}, \boldsymbol{c})$.

von $||\boldsymbol{n}||^2 = d_{\mathrm{E}}^2(\boldsymbol{y}, \boldsymbol{c})$ und in Bild 5.13 in Abhängigkeit vom SNR gezeigt. Die Schätzung für die Wortfehlerwahrscheinlichkeit des ML-Decoders (ML-Est.) wurde in Bild 5.12 mit Hilfe der Sphere-Bound von HUGHES [25] und in Bild 5.13 mit Hilfe der Tangential-Bound von BERLEKAMP [2] gewonnen, wobei jeweils in der Summe nur die Codeworte mit Gewicht $w_{\mathrm{H}}(\boldsymbol{c}) < 2d_{\mathrm{H,min}}(C)$ berücksichtigt wurden [30], da die Gewichtsverteilung des Codes $\mathrm{RM}(4,9)$ nicht bekannt ist. Die relativen Unterschiede zwischen dem $\{1000,1\}$-Decoder und dem $\{10,100\}$-Decoder sind für den AWGN-Kanal ähnlich wie für den BSC, jedoch bleibt hier bezüglich des notwendigen SNR für WER $= 10^{-4}$ eine Differenz zwischen dem ML-Decoder und dem $\{10,100\}$-Decoder von schätzungsweise 1 dB. Auch bei Erhöhung von n_p auf 500 Permutationen ist für WER $= 10^{-4}$ noch ein um ca. 0.5 dB größeres SNR notwendig als bei ML-Decodierung[38],[39].

5.5. Gemeinsame Listen- und Permutationsdecodierung für REED-MULLER Codes

Beim im vorhergehenden Abschnitt vorgestellten Verfahren werden verschiedene Permutationen des Empfangsvektors unabhängig voneinander decodiert. Erkenntnisse aus der Decodierung einer Permutation werden daher bei allen anderen Permutationen nicht berücksichtigt. Durch die regelmäßige Struktur von REED-MULLER Codes ist es jedoch möglich, zumindest eine bestimmte Anzahl verschiedener Permutationen gemeinsam zu betrachten und dadurch die Sortierung der Elemente in der Liste wesentlich zu verbessern. Das hier vorgestellte Verfahren beruht im wesentlichen auf den vom Autor zusammen mit SORGER in [54] und [57] vorgestellten Prinzipien und kann ebenso wie das in[40] [23] vorgestellte Verfahren als eine spezielle Form des

[38] In der Tat ist dem Autor auch keine anderes praktikables Decodierverfahren bekannt, mit dem für diesen Code bessere Ergebnisse erzielt werden.

[39] Siehe auch Fußnote 27 auf S. 89

[40] Ein wesentlicher Unterschied zu dem in [23] vorgestellten Verfahren ist, daß dort zur Abschätzung der Kostenfunktion h vom aktuellen Knoten zum Zielknoten nur die Gewichtsverteilung des Codes verwendet wurde, wogegen

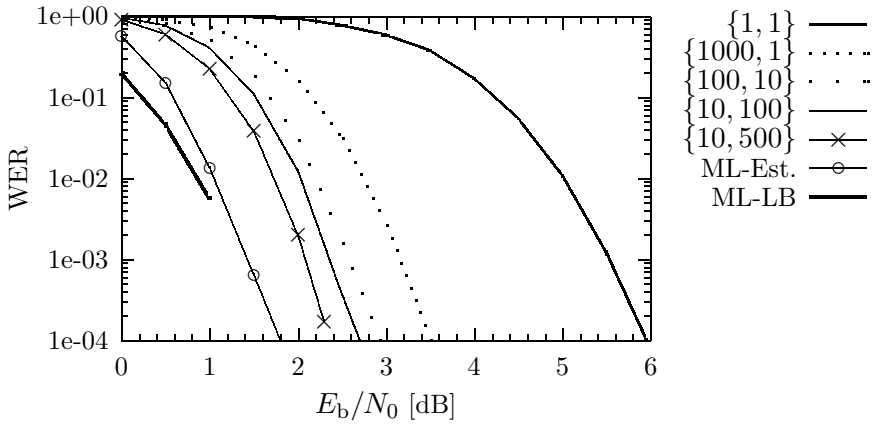

Abbildung 5.13.: Wortfehlerraten für den Code $\mathrm{RM}(4, 9)$ für verschiedene Listengrößen L und Anzahl der zufällig gewählten Permutationen n_p, bezeichnet mit $\{L, n_p\}$, alles für den AWGN-Kanal

A*-Algorithmus, der in Abschnitt 5.5.1 erläutert wird, betrachtet werden.

Einführend betrachtet wird im folgenden eine Zerlegung des Codes $C = \mathrm{RM}(r, m)$ in vier äußere Codes $C^{(1)} = \mathrm{RM}(r-2, m-2)$, $C^{(2)} = C^{(3)} = \mathrm{RM}(r-1, m-2)$ und $C^{(4)} = \mathrm{RM}(r, m-2)$ gleicher Länge $N^{(1)} = \ldots = N^{(4)} = 2^{m-2}$. In diesem Fall ist die Generatormatrix des inneren Codes B_4 (s. Abschnitt 3.3.1 S. 28) wie folgt

$$\boldsymbol{B}_4 = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 \end{bmatrix}.$$

Die Codeworte $\boldsymbol{c} \in \mathrm{RM}(r, m)$ ergeben sich durch spaltenweises Auslesen der Matrix $\boldsymbol{C}$, deren k-te Zeile durch das Codewort den inneren Codes $\boldsymbol{c}_k = (c_k^{(1)}, \ldots, c_k^{(4)}) \cdot \boldsymbol{B}_4$ gegeben ist. Empfangen sei der Vektor $\boldsymbol{y} = (y_1, y_2, \ldots, y_N)$. Zur besseren Darstellung wird nun der Vektor

$$\boldsymbol{y}_k = (y_{k,1}, y_{k,2}, y_{k,3}, y_{k,4}) = (y_k, y_{N^{(1)}+k}, y_{2N^{(1)}+k}, y_{3N^{(1)}+k})$$

eingeführt. Sowohl für den AWGN-Kanal als auch den BSC bestimmen sich die Zuverlässigkeitswerte $y_k^{(1)}$ für die Decodierung von $C^{(1)}$ aus dem Vektor $\boldsymbol{y}_k$ durch

$$y_k^{(1)} = a_{k,1} \cdot a_{k,2} \cdot a_{k,3} \cdot a_{k,4} \cdot \min(w_{k,1}, w_{k,2}, w_{k,3}, w_{k,4})$$

mit $a_{k,i} = \mathrm{sign}\,(y_{k,i})$ und $w_{k,i} = |y_{k,i}|$. Mit der Schätzung $\widehat{\boldsymbol{c}}^{(1)} \in C^{(1)}$ ergeben sich die $y_k^{(2)}$ aus den Werten $y_{k,i}$ und den $\widehat{c}_k^{(1)} \in \{\pm 1\}$ zu

$$y_k^{(2)} = a_{k,1} \cdot a_{k,3} \cdot \min(w_{k,1}, w_{k,3}) + \widehat{c}_k^{(1)} \cdot a_{k,2} \cdot a_{k,4} \cdot \min(w_{k,2}, w_{k,4}),$$

mit deren Hilfe der Code $C^{(2)}$ decodiert werden kann. Bei dem in Abschnitt 5.2.2 vorgestellten Verfahren sind die nächsten Decodierschritte wie folgt:

bei dem hier vorgestellten Verfahren auch die Struktur des Codes mit berücksichtigt wird.

1. Aus $\widehat{\boldsymbol{c}}^{(1)}$ und $\widehat{\boldsymbol{c}}^{(2)}$ Bestimmen des Codewortes $\widehat{\boldsymbol{v}} \in V = \mathrm{RM}(r-1, m-1)$ bei Zerlegung von C in zwei äußere Codes U und V gemäß der Vorschrift $\widehat{\boldsymbol{v}} = |\widehat{\boldsymbol{c}}^{(2)}|\widehat{\boldsymbol{c}}^{(2)} + \widehat{\boldsymbol{c}}^{(1)}|$

2. Berechnung der Zuverlässigkeitswerte für die Decodierung von U und daraus rekursiv die Zuverlässigkeitswerte $y_k^{(3)}$ zur Decodierung von $C^{(3)}$.

3. usw. ...

Die so ermittelten Zuverlässigkeitswerte $y_k^{(3)}$ sind von der Entscheidung $\widehat{\boldsymbol{c}}^{(2)}$ abhängig. Aufgrund der Symmetrie der Matrix $\boldsymbol{B}_4$ ist es jedoch möglich, von $\widehat{\boldsymbol{c}}^{(2)}$ unabhängige Zuverlässigkeitswerte[41] $y_k'^{(3)}$ zu bestimmen

$$y_k'^{(3)} = a_{k,1} \cdot a_{k,2} \cdot \min(w_{k,1}, w_{k,2}) + \widehat{c}_k^{(1)} \cdot a_{k,3} \cdot a_{k,4} \cdot \min(w_{k,3}, w_{k,4}),$$

die allerdings nicht die gleiche Zuverlässigkeit wie die oben unter Punkt 2 berechneten besitzen. Es ist weiterhin möglich, die von $\widehat{\boldsymbol{c}}^{(2)}$ und $\widehat{\boldsymbol{c}}^{(3)}$ unabhängigen Zuverlässigkeitswerte[42] $y_k'^{(2\oplus 3)}$ des Summencodes

$$C^{(2\oplus 3)} = \{\boldsymbol{c}^{(2\oplus 3)} : \boldsymbol{c}^{(2\oplus 3)} = \boldsymbol{c}^{(2)} + \boldsymbol{c}^{(3)}, \boldsymbol{c}^{(2)} \in C^{(2)}, \boldsymbol{c}^{(3)} \in C^{(3)}\}$$

gemäß

$$y_k'^{(2\oplus 3)} = a_{k,2} \cdot a_{k,3} \cdot \min(w_{k,2}, w_{k,3}) + \widehat{c}_k^{(1)} \cdot a_{k,1} \cdot a_{k,4} \cdot \min(w_{k,1}, w_{k,4})$$

zu ermitteln. Daß die so gewonnen Werte $y_k'^{(2\oplus 3)}$ zum Code $C^{(2\oplus 3)}$ gehören, kann leicht gesehen werden, wenn die zu $\boldsymbol{c}_k = (c_k^{(1)}, \ldots, c_k^{(4)}) \cdot \boldsymbol{B}_4$ gleichwertige Codierungsvorschrift

$$\boldsymbol{c}_k = (c_k^{(1)}, c_k^{(2)} + c_k^{(3)}, c_k^{(3)}, c_k^{(4)}) \cdot \boldsymbol{B}_4'$$

mit

$$\boldsymbol{B}_4' = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix}$$

betrachtet wird[43]. Da es sich bei den Codes $C^{(2)}$ und $C^{(3)}$ um die gleichen linearen Codes $\mathrm{RM}(r-1, m-2)$ handelt, ist auch $C^{(2\oplus 3)} = \mathrm{RM}(r-1, m-2)$. Für die weitere Betrachtung wird, um die Notation zu vereinfachen, $\boldsymbol{y}^{(2)} = \boldsymbol{y}'^{(2)}$ geschrieben.

Theorem 5.1 Für alle Codeworte $\boldsymbol{c} \in C'$ des Untercodes

$$C' = \{\boldsymbol{c} \in \mathrm{RM}(r, m) : \boldsymbol{c}^{(1)} = \widehat{\boldsymbol{c}}^{(1)}, \boldsymbol{c}^{(2)} = \widehat{\boldsymbol{c}}^{(2)}, \boldsymbol{c}^{(3)} = \widehat{\boldsymbol{c}}^{(3)}\}$$

[41] Das Primzeichen kennzeichnet hier, daß diese Zuverlässigkeitswerte unabhängig von $\widehat{\boldsymbol{c}}^{(2)}$ sind.

[42] Mit dem Zeichen $(i_1 \oplus i_2)$ wird hier nicht eine Rechnenvorschrift im Sinne einer Exor-Operation bezeichnet, sondern es wird ein neues Symbol eingeführt.

[43] Hier wurde ausgehend von $\boldsymbol{B}_4$ die zweite zur dritten Zeile addiert.

gilt für den quadratischen euklidischen Abstand $d_{\mathrm{E}}^2(\boldsymbol{y}, \boldsymbol{c})$ zum Empfangsvektor $\boldsymbol{y}$

$$d_{\mathrm{E}}^2(\boldsymbol{y}, \boldsymbol{c}) \geq d_{\mathrm{bias}}^2 + 4 \cdot \Lambda_y(\widehat{\boldsymbol{c}}^{(1)}) + 4 \cdot \alpha \cdot \sum_{i \in \{2,3,2\oplus 3\}} \sum_{\{k: \widehat{c}_k^{(i)} \neq a_k'^{(i)}\}} w_k'^{(i)} \tag{5.20}$$

mit $\alpha = 1/2$ und $\widehat{\boldsymbol{c}}^{(2\oplus 3)} = \widehat{\boldsymbol{c}}^{(2)} + \widehat{\boldsymbol{c}}^{(3)}$.

Beweis: Da natürlich für alle $\boldsymbol{c} \in C'$ und alle $i \in \{2, 3, 2 \oplus 3\}$ gilt

$$\Lambda_y(\widehat{\boldsymbol{c}}^{(1)}, \widehat{\boldsymbol{c}}^{(i)}) = \Lambda_y(\widehat{\boldsymbol{c}}^{(1)}) + \sum_{\{k: \widehat{c}_k^{(i)} \neq a_k'^{(i)}\}} w_k'^{(i)}$$

und da in der hinteren Summe in Gl. (5.20) über alle drei möglichen Erweiterungen i summiert wird, ist die Behauptung sicherlich für $\alpha = 1/3$ richtig. Zu zeigen ist nun, daß Gl. (5.20) auch für $\alpha = 1/2$ richtig ist. Nach Gl. (3.10) sind die $w'^{(i)}$, $i \in \{2, 3, 2 \oplus 3\}$ in Abhängigkeit von $\widehat{c}_k^{(1)}$ definiert als minimaler Zuwachs in $d_{\mathrm{E}}^2(\boldsymbol{y}, \boldsymbol{c})$, falls die Schätzung $\widehat{c}_k^{(i)} = a_k'^{(i)}$ falsch ist und sich der Decoder für $\widehat{c}_k^{(i)} = -a_k'^{(i)}$ entscheidet. Mit der Menge $B_\nu^{(2)} = \{\boldsymbol{b} : \boldsymbol{b} = (\nu, \pm 1, \pm 1, \pm 1) \cdot B\}$ sei $\boldsymbol{b}_k^{(\nu)} \in B_\nu^{(2)}$ für festes ν der Vektor mit kleinstem quadratischen Abstand zu $\boldsymbol{y}_k$, d.h.

$$\boldsymbol{b}_k^{(\nu)} = \arg \min_{\boldsymbol{b} \in B_\nu^{(2)}} d_{\mathrm{E}}^2(\boldsymbol{y}_k, \boldsymbol{b}).$$

Aus der Definition der $y_k'^{(i)}$ folgt automatisch, daß für $\nu = \widehat{c}_k^{(1)}$ und einem noch zu bestimmenden $a_k'^{(4)}$ gilt

$$\boldsymbol{b}_k^{(\nu)} = (\widehat{c}_k^{(1)}, a_k'^{(2)}, a_k'^{(3)}, a_k'^{(4)}) \cdot B_4 = (\widehat{c}_k^{(1)}, a_k'^{(2\oplus 3)}, a_k'^{(3)}, a_k'^{(4)}) \cdot B_4'$$

Nun ist der Vektor $(a_k'^{(2)}, a_k'^{(3)}, a_k'^{(2\oplus 3)})$ genau wie der Vektor $(c_k^{(2)}, c_k^{(3)}, c_k^{(2\oplus 3)})$ ein Codevektor des Codes G mit der Generatormatrix

$$\boldsymbol{G} = \begin{bmatrix} -1 & 1 & -1 \\ 1 & -1 & -1 \end{bmatrix}.$$

Da es sich bei G um einen Equal-Weight Code[44] mit Mindest-Hammingdistanz $w_{\mathrm{H,min}}(G) = 2$ handelt, können daher in Gl. (5.20) für jedes k nur zwei der drei Symbole $c_k^{(i)}$ ungleich dem zugehörigen $a_k^{(i)}$ gewählt werden. Daraus ergibt sich das Theorem. □

Dieses Theorem kann in seiner ursprünglichen Form nach Gl. (5.20) nur dann angewendet werden, wenn neben der Entscheidung $\widehat{\boldsymbol{c}}^{(1)}$ auch die Codeworte $\widehat{\boldsymbol{c}}^{(2)}$ und $\widehat{\boldsymbol{c}}^{(3)}$ entschieden sind. Eine direkte Folge des Theorems ist jedoch, da bei unbekannten $\boldsymbol{c}^{(2)}, \boldsymbol{c}^{(3)}$ einfach das Minimum über alle möglichen Codeworte bestimmt werden kann, daß für alle Codeworte $\boldsymbol{c} \in C''$ des Untercodes

$$C'' = \{\boldsymbol{c} \in \mathrm{RM}(r, m) : \boldsymbol{c}^{(1)} = \widehat{\boldsymbol{c}}^{(1)}\}$$

die Ungleichung

$$d_{\mathrm{E}}^2(\boldsymbol{y}, \boldsymbol{c}) \geq d_{\mathrm{bias}}^2 + 4 \cdot \Lambda_y(\widehat{\boldsymbol{c}}^{(1)}) + 4 \cdot \alpha \cdot \min_{\substack{\boldsymbol{c}^{(2)}, \boldsymbol{c}^{(3)} \\ \boldsymbol{c}^{(2\oplus 3)} = \boldsymbol{c}^{(2)} + \boldsymbol{c}^{(3)}}} \sum_{i \in \{2,3,2\oplus 3\}} \sum_{\{k: c_k^{(i)} \neq a_k'^{(i)}\}} w_k'^{(i)}$$

[44] Bei einem Equal-Weight Code besitzen alle Codewörter außer dem Vektor $(1, 1, \ldots, 1)$ das gleiche Gewicht.

ebenfalls mit $\alpha = 1/2$ gilt. Weiter ergibt sich, da die rechte Seite der Ungleichung nicht größer werden kann, wenn die Minimierung über alle $\boldsymbol{c}^{(2)}, \boldsymbol{c}^{(3)}$ und $\boldsymbol{c}^{(2\oplus 3)}$ anstatt gemeinsam, getrennt und unabhängig voneinander durchgeführt wird, folgender Satz

Satz 5.5 Für alle Codeworte $\boldsymbol{c} \in C''$ gilt

$$d_{\mathrm{E}}^2(\boldsymbol{y}, \boldsymbol{c}) \geq d_{\mathrm{bias}}^2 + 4 \cdot \Lambda_y(\widehat{\boldsymbol{c}}^{(1)}) + 4 \cdot \frac{1}{2} \cdot \sum_{i \in \{2,3,2\oplus 3\}} \min_{\boldsymbol{c}^{(i)} \in C^{(i)}} \sum_{\{k: c_k^{(i)} \neq a_k'^{(i)}\}} w_k'^{(i)} \tag{5.21}$$

□

Wesentlich bei diesem Ergebnis ist, daß in Gl. (5.21) über den Distanzzuwachs durch die unabhängige Decodierung der *drei* Codes $C^{(2)}, C^{(3)}, C^{(2\oplus 3)}$ bzw. der Vektoren $\boldsymbol{y}'^{(2)}, \boldsymbol{y}'^{(3)}, \boldsymbol{y}'^{(2\oplus 3)}$ summiert, die Summe anschließend aber nur mit $\alpha = 1/2$ multipliziert wird. Ein Vergleich mit der im vorigen Abschnitt vorgestellten Decodierung verschiedener Permutationen ergibt, daß hier drei verschiedene Permutationen, die verschiedene Codeworte $\boldsymbol{c}^{(2)}$ aber gleiche Codeworte $\boldsymbol{c}^{(1)}$ ergeben, gemeinsam betrachtet werden. Mit dieser Methode kann nun die Metrik $\Lambda_y(\widehat{\boldsymbol{c}}^{(1)})$ bei der Beurteilung der verschiedenen Hypothesen für $\boldsymbol{c}^{(1)}$ wesentlich verbessert werden, natürlich auf Kosten höherer Komplexität.

Die Verallgemeinerung dieses Satzes in eine Zerlegung in 2^s äußere Codes wird im Anhang A.12 vorgestellt und es resultiert daraus folgender Satz

Satz 5.6 Für eine Zerlegung von $\mathrm{RM}(r, m)$ in 2^s äußere Codes $C^{(1)}, \ldots, C^{(2^s)}$ mit der Generatormatrix des inneren Codes[45] $\boldsymbol{B}_{2^s} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{2^s})^T$ sei F die Menge aller Inzidenzvektoren $\boldsymbol{f}^{(i)}$, $i = 1, 2, \ldots, 2^s - 1$ der $(s-1)$-Flächen in $\mathrm{EG}(s, 2)$, die durch den Punkt $\boldsymbol{x} = (-1, -1, \ldots, -1) \in \mathrm{EG}(s, 2)$ gehen. Mit den Zuverlässigkeitswerten

$$y_k'^{(i)} = \prod_{\{n: f_n^{(i)} = 0\}} a_{kn} \cdot \min_{\{n: f_n^{(i)} = 0\}} w_{kn} + \widehat{c}_k^{(1)} \cdot \prod_{\{n: f_n^{(i)} = 1\}} a_{kn} \cdot \min_{\{n: f_n^{(i)} = 1\}} w_{kn}$$

und dem Code $Q = C^{(2)} = \mathrm{RM}(r - s + 1, m - s)$ gilt für alle Codeworte des Codes C''

$$d_{\mathrm{E}}^2(\boldsymbol{y}, \boldsymbol{c}) \geq d_{\mathrm{bias}}^2 + 4 \cdot \Lambda_y(\widehat{\boldsymbol{c}}^{(1)}) + 4 \cdot \frac{1}{2^{s-1}} \cdot \sum_{i=1}^{2^s - 1} \min_{\boldsymbol{q} \in Q} \sum_{\{k: q_k \neq a_k'^{(i)}\}} w_k'^{(i)} \tag{5.22}$$

mit $a_k'^{(i)} = \operatorname{sign} y_k'^{(i)}$ und $w_k'^{(i)} = |y_k'^{(i)}|$.

Beweis: siehe Anhang A.12 □

Auch hier ergibt sich eine Verbesserung der Metrik $\Lambda_y(\widehat{\boldsymbol{c}}^{(1)})$, da in Gl. (5.22) über den Distanzzuwachs durch die unabhängige Decodierung der $2^s - 1$ Vektoren $\boldsymbol{y}'^{(i)}$ summiert, aber die Summe anschließend nur durch 2^{s-1} dividiert wird. Grundsätzlich ist es damit bei der Listendecodierung und Zerlegung in 2^m äußere Codes $C^{(1)}, \ldots, C^{(2^m)}$ der Länge $N^{(i)} = 1$ möglich,

[45] Die Zeilen von $\boldsymbol{B}_{2^s}$ sind bekanntermaßen Flächen in $\mathrm{EG}(s, 2)$.

die Metriken der verschiedenen Hypothesen für $\boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)}, \ldots, \boldsymbol{c}^{(\tau)}$ in der Liste zu verbessern, wenn für ein beliebiges $s \in \{1, 2, \ldots, r\}$gilt[46]

$$\sum_{i=1}^{\tau} K^{(i)} = \dim(\mathrm{RM}(r-s, m-s)).$$

Die Minimierung in den Gleichungen (5.21) und (5.22) erfordert eine ML-Decodierung der Codes $C^{(i)}$, $i \in \{2, 3, 2 \oplus 3\}$ bzw. des Codes Q, die selbstverständlich nur für Codes mit kleiner Anzahl von Codeworten durchführbar ist. Es handelt sich jedoch beim Code Q ebenfalls um einen RM-Code, so daß dieser auch mit Hilfe des in Abschnitt 5.3 vorgestellten Listendecoders rekursiv decodiert werden kann. Da die Metrik

$$\Lambda_y(\boldsymbol{q}) = \sum_{\{k: q_k \neq a_k'^{(i)}\}} w_k'^{(i)}$$

additiv ist, muß daher für eine untere Abschätzung $\Lambda_y^{(\mathrm{LB})}(\boldsymbol{y}'^{(i)}) \leq \min_{\boldsymbol{q}} \Lambda_y(\boldsymbol{q})$ nicht der ganze Codebaum von Q durchlaufen werden. So kann z.B. die Decodierung nur bis zu einer bestimmten Tiefe τ durchgeführt werden, und dann als untere Abschätzung $\Lambda_y^{(\mathrm{LB})}$ die kleinste Metrik in Liste verwendet werden. Eine andere, etwas flexiblere Methode, besteht in einem vollständigen Durchlaufen des Codebaumes bei begrenzter Listengröße, wobei die verwendete untere Abschätzung $\Lambda_y^{(\mathrm{LB})}$ nicht größer sein darf als die minimale Metrik Λ_y aller Hypothesen, die verworfen wurden. Beide Methoden erfordern zusätzlich zur Hauptliste der Größe L eine zweite Liste zur Decodierung von Q, deren Größe im folgenden mit L_{2nd} bezeichnet wird.

Aus diesen Überlegungen ergibt sich daher für die sequentielle bzw. Listendecodierung der 2^m äußeren Codes der Länge $N^{(i)} = 1$ die verbesserte Gesamtmetrik $\Lambda_y^{\star}(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(\tau)})$, $\tau \leq 2^m$ für die *gemeinsame Listen- und Permutationsdecodierung* zu[47]

$$\Lambda_y^{\star}(\widehat{\boldsymbol{c}}^{(1)}, \ldots, \widehat{\boldsymbol{c}}^{(\tau)}) = \max\left\{ \Lambda_y(\widehat{\boldsymbol{c}}^{(1)}, \ldots, \widehat{\boldsymbol{c}}^{(\tau)}), \Lambda_y(\widehat{\boldsymbol{c}}^{(1)}, \ldots, \widehat{\boldsymbol{c}}^{(2^{m-s})}) + \frac{1}{2^{s-1}} \cdot \sum_{i=1}^{2^s-1} \Lambda_y^{(\mathrm{LB})}(\boldsymbol{y}'^{(i)}) \right\},$$

mit dem kleinsten $s \geq 0$, für das gilt

$$\sum_{i=1}^{\tau} K^{(i)} \geq \dim(\mathrm{RM}(r-s, m-s)).$$

Bei der zweiten Methode erhöht sich für jedes $s \leq r$ und jeden Eintrag in der Liste L die Gesamtkomplexität des Algorithmus um einen Term von der Ordnung $\mathcal{O}((2^s - 1) L_{\mathrm{2nd}} N_Q \log_2 N_Q)$, mit

[46] Die Funktion $\dim(C)$ ergibt die Dimension des Codes C, was gleich ist zur Anzahl der Informationstellen K eines (N, K, D)-Codes.

[47] Diese Metrik kann noch weiter verbessert werden, da für ein bestimmtes i die untere Abschätzung $\Lambda_y^{(\mathrm{LB})}(\boldsymbol{y}'^{(i)})$ durch $\Lambda_y(\widehat{\boldsymbol{c}}^{(2^{m-s}+1)}, \ldots, \widehat{\boldsymbol{c}}^{(\tau)})$ ersetzt werden kann.

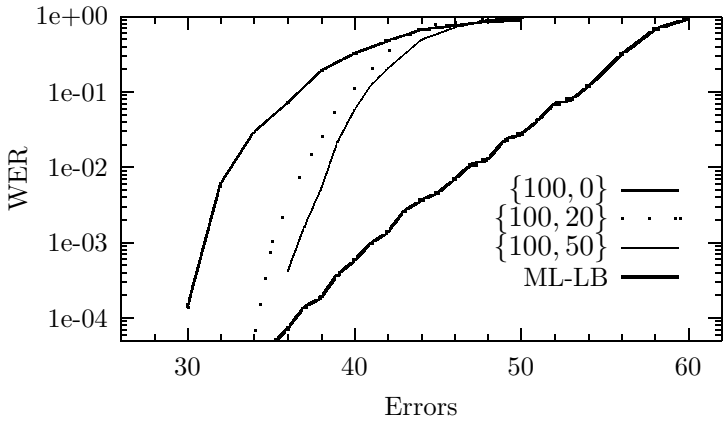

Abbildung 5.14.: Wortfehlerraten bei kombinierter Permutationsdecodierung für den Code RM$(4, 9)$ für verschiedene Listengrößen L und L_{2nd}, bezeichnet mit $\{L, L_{\text{2nd}}\}$

$N_Q = 2^{m-s}$ der Länge des Codes Q , so daß schließlich die Gesamtkomplexität von der Ordnung

$$\mathcal{O}(L(N \log_2 N + \sum_{s \le r}(2^s - 1) L_{\text{2nd}} N_Q \log_2 N_Q))$$

ist.

Zum Abschluß dieses Abschnitts werden einige mit diesem Algorithmus erzielten Ergebnisse vorgestellt. Bei den in Bild 5.14 gezeigten Wortfehlerwahrscheinlichkeiten wurde jeweils für alle $s \le r$ eine Listendecodierung der $2^s - 1$ verschiedenen Vektoren $\boldsymbol{y}'^{(i)}$ mit Listengröße L_{2nd} durchgeführt. Ferner wurde als untere Abschätzung $\Lambda_y^{(\text{LB})}(\boldsymbol{y}'^{(i)}) \le \min_{\boldsymbol{q}} \Lambda_y(\boldsymbol{q})$ das Minimum aus den Metriken aller verworfenen Hypothesen und der Decodierentscheidung für Q verwendet. Für $L_{\text{2nd}} = 0$ ist damit auch $\Lambda_y^{(\text{LB})}(\boldsymbol{y}^{(i)}) = 0$, und der gemeinsame Permutationsdecoder ist identisch mit der einfachen Listendecodierung nach Abschnitt 5.3. Bei gleicher Komplexität sind die allein mit gemeinsamer Permutationsdecodierung erzielten Wortfehlerraten schlechter als die im vorigen Abschnitt vorgestellten Ergebnisse (vergl. Bild 5.10). Dies liegt zum einen an der geringen Anzahl von Permutationen, die hier gemeinsam betrachtet werden, es sind dies $|P| = 1, 3, 7, 15$ für $s = 1, 2, 3, 4$. Zum anderen ist für alle Permutationen die Anzahl der Fehler in $\boldsymbol{y}^{(1)}$ gleich. Sinnvoller wäre es, zusätzlich noch verschiedene Permutationen zu betrachten, die verschiedene Vektoren $\boldsymbol{y}^{(1)}$ ergeben. Dies wurde allerdings im Rahmen dieser Arbeit nicht weiter untersucht.

Die hier vorgestellte Methode, verschiedene Permutationen gemeinsam zu betrachten und die verschiedenen Ergebnisse zur Verbesserung der Metrik zu verwenden, kann auch als eine spezielle Form des A*-Algorithmus betrachtet werden, der in der "Künstlichen Intelligenz" schon lange bekannt ist.

5.5.1. Der A*-Algorithmus[48]

In diesem Abschnitt soll nur eine kleine Einführung in das Gebiet der Suchalgorithmen gegeben werden, um den Begriff "A*-Algorithmus" etwas näher zu erläutern. Für eine vollständigere

[48] Sprich: 'A-Star-Algorithmus'

Einführung sei z.B. auf [37] verwiesen.

Die sequentielle Decodierung gehört zur Klasse der Suchalgorithmen, die bei einem gegebenen Baum (oder auch Graph) und gegebener Kostenfunktion zwischen zwei benachbarten Baumknoten versuchen, den Weg mit geringsten 'Kosten' von einem Startknoten s zu einem Zielknoten z zu finden. Die verschiedenen Suchalgorithmen lassen sich unterteilen in zwei Klassen: bei Algorithmen der ersten Klasse wird der Baum komplett durchlaufen, wogegen bei denen der zweiten, basierend auf einer Metrik zur Bewertung der verschiedenen Pfade, nur ein Teil des ganzen Baumes durchsucht wird. Der sequentielle Algorithmus, der nur der Pfad mit der besten Metrik weiterverfolgt, gehört damit in die zweite Klasse.

Die minimalen Kosten über alle mögliche Pfade zwischen zwei Knoten n_i und n_j werde mit $k(n_i, n_j) \geq 0$ bezeichnet. Damit sind die minimalen Kosten zwischen Start- und Zielknoten $k(s, z)$. Für einen beliebigen Knoten n sei $f(n) \geq 0$ das Minimum der Kosten über alle Pfade, die durch den Knoten n gehen. Falls sich nun die Kosten wie eine additive Metrik verhalten, kann f aufgespalten werden in

$$f(n) = g(n) + h(n), \tag{5.23}$$

mit den Kosten des optimalen Pfades vom Startknoten s zu n, $g(n) \geq 0$ und den Kosten des optimalen Pfades vom Knoten n zum Zielknoten z, $h(n) \geq 0$. Wie schnell der optimale Pfad mit kleinster Kostenfunktion gefunden werden kann, hängt entscheidend von der Kenntnis über die Funktion f ab. Wenn beim Durchlaufen den Baumes für jeden Knoten n nur die Funktion $g(n)$ ermittelt wird, und damit die Metrik zur Bewertung der verschiedenen Hypothesen nur die 'Vergangenheit' berücksichtigt, kann unter Umständen ein schlechter Pfad sehr lange verfolgt werden, was zu einem hohen Suchaufwand führt. Dagegen kann (theoretisch) bei Kenntnis von $g(n)$ und $h(n)$ zu jedem Zeitpunkt immer die richtige Entscheidung für den optimalen Knoten n getroffen werden, was die Anzahl der untersuchten Knoten allgemein wesentlich verringert. Natürlich ist es, von Ausnahmen abgesehen[49], nicht möglich, beide Funktionen g und h zu kennen, ohne den ganzen Baum analysiert zu haben. Es ist aber in manchen Fällen möglich, die Funktion $h(n)$ zu schätzen und mit dieser Schätzung den Suchaufwand zu reduzieren. Generell werden Algorithmen, die eine Schätzung $\widetilde{h}(n)$ verwenden, als A-Algorithmen bezeichnet. Dabei kann die Schätzung $\widetilde{h}(n)$ sowohl größer als auch kleiner als der richtige Wert $h(n)$ sein. Für alle Algorithmen aus der Gruppe der A*-Algorithmen gilt jedoch immer $\widetilde{h}(n) \leq h(n)$. Es wird damit immer eine untere Abschätzung von $h(n)$ verwendet. Während A-Algorithmen, die immer den Pfad mit der geringsten Kostenfunktion verlängern, nicht notwendigerweise den optimalen Pfad finden, ist dies bei A*-Algorithmen garantiert[50].

Die auf euklidischer Distanz basierte Metrik $\Lambda_y(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(\tau)})$ nach Gl. (5.17) ist somit eine Realisierung der Funktion $g(n)$ für den Knoten, der durch den Hypothesenvektor $(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(\tau)})$ gegeben ist, und der in Abschnitt 5.3.2 vorgestellte Algorithmus verwendet damit nur die Kenntnis über den bereits untersuchten Teil des Codebaumes[51]. Ein Vergleich von Gl. (5.23) mit der im vorigen Abschnitt vorgestellten Metrik nach Gl. (5.22) ergibt, daß es sich bei der Summe über die $2^s - 1$ verschiedenen Distanzzuwächse durch die unabhängige Decodierung der Codes

[49] Eine Ausnahme ist z.B. durch die Kostenfunktion zwischen benachbarten Knoten $k(n_i, n_{i+1}) = const, \forall i$ gegeben.

[50] Natürlich nur, wenn genügend große Rechenkapazität und ein genügend großer Speicher zur Verfügung steht.

[51] Da mit $\widetilde{h}(n) = 0$ immer die Bedingung $\widetilde{h}(n) \leq h(n)$ erfüllt ist, kann dieser Algorithmus verallgemeinert natürlich auch als A*-Algorithmus betrachtet werden.

Q um eine Realisierung der Schätzung von $\widetilde{h}(n) \leq h(n)$ handelt. Damit ist der der in Abschnitt 5.5 eingeführte Algorithmus eine spezielle Form des A*-Algorithmus.

Zusammenfassung

In diesem Kapitel wurden die wesentlichen in dieser Arbeit verwendeten Decodierverfahren vorgestellt. Es zeigte sich, daß die Wortfehlerwahrscheinlichkeit bei Übertragung über den AWGN-Kanal und bitweiser Mehrstufendecodierung sehr exakt mit Hilfe des äquivalenten SNR numerisch bestimmt werden kann. Für die Listendecodierung von rekursiv konstruierten $|u|u+v|$-Codes wurden zwei verschiedene Metriken zur Beurteilung der verschiedenen Hypothesen eingeführt. Es ist damit möglich, für alle RM-Code bis zu Länge $N \leq 128$ eine Decodierung mit Fehlerraten nahe der des ML-Decoders zu erzielen.

Das prinzipielle Problem, daß gerade zu Beginn des Decodiervorgangs eine sehr große Liste erforderlich ist, verhindert allgemein eine nahezu optimale Decodierung von langen RM-Codes mit Raten wesentlich von Null oder Eins verschieden. Aber zumindest für Codes mittlere Länge kann durch Decodierung verschiedener Permutationen auch eine fast optimale Wortfehlerwahrscheinlichkeit erreicht werden. Wie später im Kapitel 7 gezeigt, können so bei Übertragung über den AWGN-Kanal auch alle RM-Codes der Länge $N = 256$ mit geringem Aufwand nahezu ML-decodiert werden, Dagegen ist, wie in diesem Kapitel gezeigt, beim Code RM$(4, 9)$ selbst bei Verwendung von 500 Permutationen noch ein um ca. 0.5 dB höheres SNR notwendig als bei ML-Decodierung.

Für RM-Codes noch größerer Länge sind jedoch auch mit diesen Verfahren keine guten Fehlerwahrscheinlichkeiten zu erzielen. Zwar können mit langen RM-Codes bei ML-Decodierung relativ kleine Wortfehlerraten erzielt werden, die hier vorgestellten Decodierverfahren sind aber nicht geeignet, um das Korrekturpotential langer RM-Code auszunutzen. Um auch bei großen Codelängen gute Wort- bzw. Bitfehlerwahrscheinlichkeiten erzielen zu können, muß daher der Code an den Decoder angepaßt werden. In nächsten Kapitel werden daher zwei Kriterien zur Konstruktion von $|u|u+v|$-Codes vorgestellt, um lange Codes an die Listendecodierung anzupassen.

6. Verschiedene Optimierungsstrategien der Codekonstruktion

Nachdem im vorigen Kapitel verschiedene Decodierverfahren vorgestellt wurden, beschäftigen sich die folgenden Abschnitte mit der Optimierung der Codekonstruktion. Auf der einen Seite sind innerhalb der durch die rekursive $|u|u+v|$-Konstruktion gegebenen Codeklasse die REED-MULLER Codes diejenigen mit dem besten Korrekturpotential. Da sich jedoch nach den Erkenntnissen aus Abschnitt 5.3.4 lange RM-Codes mit einer Rate wesentlich von Null bzw. Eins verschieden nur sehr schlecht mit dem hier vorgestellten Listendecodierverfahren decodieren lassen[1], bietet es sich an, auf das Decodierverfahren angepaßte Codes zu konstruieren. Generell erweist sich die rekursive $|u|u+v|$-Konstruktion als sehr flexibel, und es lassen so aus der großen Fülle verschiedener Codes jeweils diejenigen auswählen, die für ein gegebenes Decodierverfahren mit gegebener Decodierkomplexität geeignet sind. Bei allen hier konstruierten Codes handelt es sich um Unter- bzw. Übercodes von RM-Codes. Allgemein besitzen diese Codes zwar ein geringeres Korrekturpotential als die RM-Codes, für feste Codelänge und Coderate sind die erzielten Fehlerraten bei gegebener Decodierkomplexität jedoch gerade bei großen Codelängen wesentlich besser.

Erste Schritte in diese Richtung wurden auch schon von DUMER und SHABUNOV unternommen [11], [12]. Die von ihnen vorgestellten Ergebnisse beschränkten sich im wesentlichen darauf, ausgehend von einem gegebenen RM-Code mit Dimension K einen Untercode mit Dimension $K_{\text{sub}} \leq K$ zu konstruieren, indem bei einer Zerlegung des RM-Code in äußere Wiederholcodes $C^{(i)}$ mit $K^{(i)} = 1$, $\forall i$ zusätzlich alle $K^{(i)} = 0$ für $i \leq K - K_{\text{sub}}$ gesetzt werden[2]. Obwohl schon mit dieser Methode für lange Codes bessere Wortfehlerraten als mit dem RM-Code erzielt werden können, ist eine gezieltere Auswahl der Wiederholcodes, für die $K^{(i)} = 0$ gesetzt wird, sinnvoll. Abhängig von der Decodierkomplexität, d.h. vom Decodierverfahren, werden in diesem Abschnitt zwei Strategien vorgestellt, um dafür geeignete äußere Codes zu finden.

6.1. Optimierte Konstruktion für bitweise Mehrstufendecodierung

Die rekursive bitweise Mehrstufendecodierung wurde in Abschnitt 5.2.2 vorgestellt. Zudem wurde in Abschnitt 5.2.3 gezeigt, daß mit Gl. (5.12) und z.B. (5.13), basierend auf dem äquivalenten SNR für die äquivalenten Kanäle zur Übertragung der äußeren Codes $C^{(i)}$, $1 \leq i \leq N$,

[1] Es sollte angemerkt merkt, daß dem Autor zu diesem Zeitpunkt auch kein anderes praktikables Decodierverfahren bekannt ist, daß der hier vorgestellten Listendecodierung diesbezüglich überlegen ist.

[2] Es wurden in [11] auch andere Methoden zur Konstruktion von Untercodes vorgestellt, die aber nur geringe Verbesserung brachten.

die Wortfehlerrate bei bitweiser MSD sehr genau geschätzt werden kann. Eine wesentliche Erkenntnis daraus ist, daß die WER des Gesamtcodes sehr exakt aus den Einzelfehlerwahrscheinlichkeiten $P_e^{(i)}$ der äußeren Codes mit $K^{(i)} = 1$ ermittelt werden kann, da die Gesamtentscheidung nur dann richtig ist, wenn jede einzelne Entscheidung für jeden äußeren Code mit $K^{(i)} = 1$ richtig ist. Generell sind die Einzelfehlerwahrscheinlichkeiten der äußeren Codes sehr verschieden, und obwohl tendenziell diejenigen Positionen i, deren zugehörige Zeilenvektoren $\boldsymbol{B}_{2^m,i}$ (s. Abschn. 3.3.2) das größte Gewicht besitzen, auch eine kleine Einzelfehlerwahrscheinlichkeit haben, gibt es keinen direkten Zusammenhang zwischen dem Gewicht $w_{\mathrm{H}}(\boldsymbol{B}_{2^m,i})$ und $P_e^{(i)}$. Für $K < N$ bietet es sich daher an, neben dem klassischen Konstruktionsprinzip nach Abschnitt 3.3.2, durch das die Mindestdistanz des Codes maximiert wird, das Prinzip der *minimalen Wortfehlerwahrscheinlichkeit bei bitweiser MSD* einzuführen. Die sich daraus ergebende Konstruktionsvorschrift für einen (N, K)-Code, $N = 2^m$, $K \in \{1, 2, \ldots, 2^m\}$ lautet wie folgt:

Schritt 1: Abhängig von den Kanaleigenschaften und der verwendeten Zuverlässigkeitsübergabe Bestimmung der Einzelfehlerwahrscheinlichkeit $P_e^{(i)} = P(\widehat{\boldsymbol{c}}^{(i)} \neq \boldsymbol{c}^{(i)})$ für alle $i \in \{1, 2, \ldots, N\}$

Schritt 2: Auswahl der K Positionen i mit kleinster Einzelfehlerwahrscheinlichkeit $P_e^{(i)}$. Für diese Positionen wird $K^{(i)} = 1$ gesetzt, für alle anderen wird $K^{(i)} = 0$ gewählt.

Der nach diesem Prinzip gefundene Code wird im folgenden als *optimierter Code für bitweise MSD*, oder kurz OCBM bezeichnet[3]. Da durch die Optimierung keine strukturellen Veränderungen bezüglich der multiplen rekursiven Verkettung vorgenommen werden, ist auch für den OCBM die Decodierkomplexität wie beim RM-Code von der Ordnung[4] $O(N \log_2 N)$.

Bezüglich Schritt Nr. 1 der Konstruktionsvorschrift ist zu betonen, daß die Kanaleigenschaften wie z.B. das SNR mit in die Konstruktion einfließen. Zudem können die $P_e^{(i)}$ auf verschiedene Arten ermittelt werden, so z.B. bei distanzbasierter Zuverlässigkeitsübergabe aus den Wahrscheinlichkeitsdichten $f(y_1^{(i)}|c_1^{(i)})$, aus dem rekursiv bestimmten äquivalenten $\mathrm{SNR}^{(i)}$ (s. Bild 5.3 auf S. 48) oder auch direkt mit Hilfe einer Monte-Carlo-Simulation. Die letzte Methode bietet sich gerade bei wahrscheinlichkeitsbasierter Zuverlässigkeitsübergabe an, da hier die Wahrscheinlichkeitsdichten der rekursiv berechneten Zuverlässigkeitswerte nicht bekannt sind. Allerdings ist die rekursive Bestimmung der Dichten $f(y_1^{(i)}|c_1^{(i)})$ nur bei relativ kurzen Codes mit wenigen Rekursionsstufen ausreichend exakt möglich. Wie später noch gezeigt, sind für Codes der Länge $N = 2^{15}$ die durch das leicht zu bestimmende äquivalente SNR ermittelten $P_e^{(i)}$ ausreichend exakt (s. Bild 6.3), so daß sich für diese Längen auf diese Methode beschränkt werden kann. Erst bei sehr langen Codes wird die Beschreibung der äquivalenten Kanäle durch das äquivalente SNR zunehmend ungenau und so wird die Ermittlung der Einzelfehlerwahrscheinlichkeiten durch eine Monte-Carlo-Simulation notwendig.

[3] Da es verschiedene Verfahren gibt, den Code auf bitweise MSD zu optimieren, und diese Verfahren aufgrund der ihnen innewohnenden Ungenauigkeiten im allgemeinen leicht verschiedene optimierte Code ergeben, wird hier nicht zwischen diesen einzeln Codes unterschieden, sondern für alle gefundenen Code die Bezeichnung 'OCBM' verwendet.

[4] Da der gefundene OCBM keine so regelmäßige Struktur besitzt wie der RM-Codes, kann in der Praxis unter Umständen ein etwas aufwandsgünstigerer Decodieralgotithmus für den RM-Code realisiert werden.

Die in [24] vorgeschlagene, effektive Technik zur Bestimmung der Bitfehlerrate bei Maximum-A-posteriori-Decodierung durch Monte-Carlo-Simulation kann auch hier zur Bestimmung der Einzelfehlerwahrscheinlichkeiten eingesetzt werden, falls die Zuverlässigkeitsübergabe wahrscheinlichkeitsbasiert ist. In diesem Fall ergeben die aus dem Empfangsvektor $\boldsymbol{y}$ rekursiv berechneten Werte $h_1^{(i)}$ direkt die Wahrscheinlichkeit

$$P(c_1^{(i)} = \pm 1 | \boldsymbol{y}, \widehat{\boldsymbol{c}}^{(1)}, \ldots, \widehat{\boldsymbol{c}}^{(i-1)}) = \frac{1 \pm h_1^{(i)}}{2}$$

und es lassen sich damit sog. 'weiche Fehler' zählen. Die geforderte Genauigkeit bei der Bestimmung von

$$P(\widehat{c}_1^{(i)} \neq c_1^{(i)} | \widehat{\boldsymbol{c}}^{(1)}, \ldots, \widehat{\boldsymbol{c}}^{(i-1)}) = E_{\boldsymbol{y}}\{P(\widehat{c}_1^{(i)} \neq c_1^{(i)} | \boldsymbol{y}, \widehat{\boldsymbol{c}}^{(1)}, \ldots, \widehat{\boldsymbol{c}}^{(i-1)})\}$$

kann damit durch wesentlich weniger Durchläufe im Vergleich zur herkömmlichen Technik, die $P_e^{(i)}$ durch Zählen 'harter Fehler' zu ermitteln, erreicht werden.

Bild 6.1 vergleicht die tatsächliche WER des durch Monte-Carlo-Simulation gefundenen OCBM mit Parametern $N = 512, K = 256$ mit der nach Gl. (5.12) geschätzten Fehlerrate, wobei die $P_e^{(i)}$ aus der Simulation gewonnen wurden (Kurve (a)). Das Signal-zu-Rauschverhältnis wurde zu $E_\mathrm{b}/N_0 = 3.4$ dB gewählt. Es ist zwar eine große Anzahl von Durchläufen T notwendig, um den tatsächlichen Wert der WER mit Gl. (5.12) genau zu bestimmen, aber der gefundene OCBM ist schon bei sehr kleiner Anzahl T (praktisch schon bei $T = 1$) sehr nahe am Optimum. Gerade hier zeigt sich der Vorteil, zur Bestimmung der $P_e^{(i)}$ die Technik der 'weichen Fehler' einzusetzen, denn obwohl die tatsächliche WER im Bereich von 10^{-3} liegt und damit bei $T = 1$ mit großer Wahrscheinlichkeit noch kein einziger 'harter Fehler' gezählt wird, erlauben es die für $T = 1$ gewonnen Näherungen für die $P_e^{(i)}$, fast die richtigen Positionen zu $K^{(i)} = 1$ zu setzen. Auch erscheint die Wahl derjenigen Positionen mit $K^{(i)} = 1$ nicht besonders kritisch zu sein, so daß es viele Codes mit annähernd gleicher Wortfehlerwahrscheinlichkeit gibt. Zum Vergleich ist in Bild 6.1 auch die mit Hilfe des äquivalenten SNR geschätzte WER des damit gefundenen OCBM gezeigt (Kurve (c)). Sie ist sehr genau und auch der damit gefundene Code ist für diese Codeparameter N und K praktisch optimal.

In Bild 6.2 werden die Wortfehlerraten der mit dem äquivalenten SNR gefundenen OCBM verglichen mit den Fehlerraten für die beiden RM-Codes $\mathrm{RM}(3, 5)$ und $\mathrm{RM}(4, 9)$. Die Zuverlässigkeitsübergabe ist in allen Fällen distanzbasiert. Für die gegebenen Parameter N und K wurde für jeden SNR-Wert jeweils der dafür beste Code bestimmt. Die durch die Optimierung erzielten Verbesserungen sind beim $(32, 16)$-OCBM gegenüber dem RM-Code gleicher Länge und gleicher Anzahl der Informationsstellen sehr gering. Die Tatsache, daß zudem der gefundene OCBM für $E_\mathrm{b}/N_0 \geq 3$ dB exakt der Code $\mathrm{RM}(3, 5)$ ist, unterstreicht die Erkenntnis, daß die bitweise MSD für kurze RM-Codes ein sehr effektives Decodierverfahren ist. Deutlich anders sind die Verhältnisse für[5] $(N, K) = (512, 256)$. Hier kann durch die Optimierung verglichen mit dem Code $\mathrm{RM}(4, 9)$ ein zusätzlicher Codiergewinn von über 2 dB bei WER $= 10^{-4}$ erzielt werden, und dies ohne Erhöhung der Decodierkomplexität. Die durch diese Optimierung gefundenen Codes haben im Bereich $1.2 \text{ dB} \leq E_\mathrm{b}/N_0 \leq 10$ dB eine Mindestdistanz von $d_{\mathrm{H,min}}(\mathrm{OCBM}) = 16$,

[5] Das hier für diesen Code gewonnene Ergebnisse wurde auch schon in [38] (einer vom Autor angeleiteten Arbeit) dokumentiert.

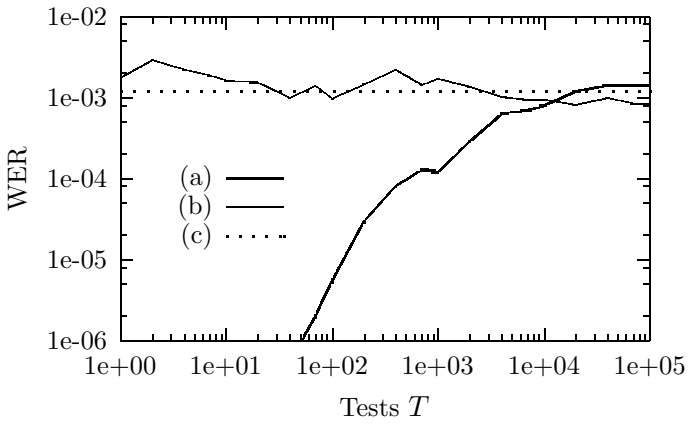

Abbildung 6.1.: Vergleich zwischen tatsächlicher und geschätzter WER bei bitweiser MSD eines durch Monte-Carlo-Simulation gefundenen (512,256)-OCBM bei $E_\mathrm{b}/N_0 = 3.4$ dB in Abhängigkeit der Testdurchläufe: (a) geschätzte WER, (b) tatsächliche WER des Codes, der nach T Tests gefunden wurde, (c) geschätzte WER des mit Hilfe des äquivalenten SNR gefundenen Codes.

die nur halb so groß wie die Mindestdistanz des RM-Codes von $d_{\mathrm{H,min}}(\mathrm{RM}(4,9)) = 32$. Auch wenn am Ende diese Abschnitts noch genauer gezeigt wird, daß der OCBM für moderate SNR-Werte geringeres Korrekturpotential besitzt als der auf maximale Distanz hin optimierte Code, so kann dies hier schon aufgrund eines Vergleichs der Mindestdistanzen richtig vermutet werden.

Für den Bereich der Wortfehlerraten, die in der Praxis interessant sind[6], steigt der Codiergewinn bei bitweiser MSD der OCBM gegenüber den RM-Codes mit gleicher Dimension K für große Codelängen stark an, da zum einen für feste Coderate, festes SNR und wachsende Codelänge die WER der OCBM immer besser wird[7], und zum anderen die mit den RM-Codes erzielten Fehlerraten immer schlechter werden. Beispielhaft für $R = 0.5$ sind in Bild 6.3 die Wortfehlerraten bei bitweiser MSD einiger Codes der Länge $N = 2^m$ gezeigt. Bis zu einer Länge von $N = 32768$ zeigt sich eine sehr exakte Übereinstimmung zwischen dem aus Gl. (5.12) zusammen mit dem äquivalenten SNR geschätzten WER und dem durch die Simulation ermittelten, so daß für diese Längen die numerische Schätzung zur Bestimmung der Wortfehlerraten praktisch ausreicht.

Eine wesentliche Erkenntnis aus diesen Ergebnissen ist, daß Satz 5.1 (s. S. 40) für den OCBM scheinbar[8] nicht gilt, d.h. auch bei bitweiser MSD werden die mit den OCBM erzielten Wortfehlerraten für konstante Rate und festes SNR mit wachsender Codelänge nicht schlechter, sondern im Gegenteil immer besser.

Tabelle 6.1 vergleicht die Distanzen der RM-Codes mit Länge $N = 2^m$ und Rate $R = 0.5$ mit denen der OCBM bei einer Wortfehlerrate von WER $= 10^{-4}$ mit bitweiser MSD. Auch hier wird deutlich, daß sich bei moderatem SNR der Unterschied zwischen OCBM und RM-Code mit wachsender Codelänge immer mehr vergrößert. Ist bei einer Länge von $N = 32$ und $E_\mathrm{b}/N_0 \geq 3$

[6] Da nach Satz 6.1 für SNR $\to \infty$ der OCBM gleich dem RM-Code ist, ist für feste N und K der asymptotische Codiergewinn (SNR $\to \infty$, WER $\to 0$) von OCBM und RM-Code bei bitweiser MSD gleich.

[7] Zwar konnte in dieser Arbeit nicht gezeigt werden, daß dies immer gilt, aber für alle untersuchten Coderaten sank bei fester Coderate die WER mit der Codelänge.

[8] Leider konnte dies im Rahmen dieser Arbeit nicht allgemein bewiesen werden.

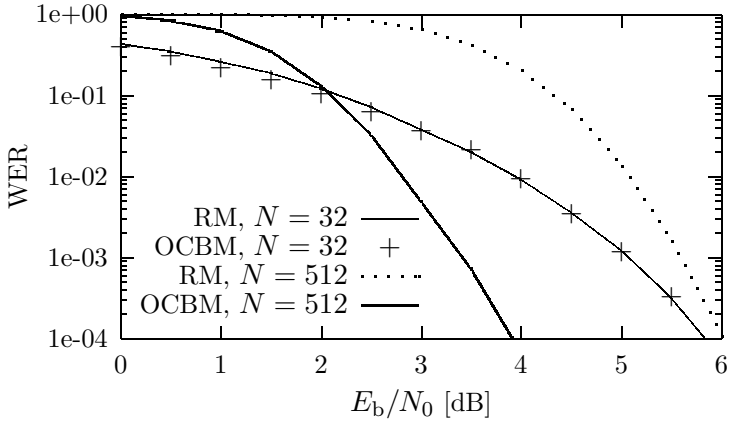

Abbildung 6.2.: Vergleich der Wortfehlerwahrscheinlichkeiten von RM-Codes und auf minimale WER optimierte Codes (bezeichnet mit OCBM) der Rate $R = 0.5$ bei bitweiser MSD mit distanzbasierter Zuverlässigkeitsübergabe bei Übertragung über den AWGN-Kanal. Für jeden SNR-Wert wurde der dazu beste OCBM bestimmt.

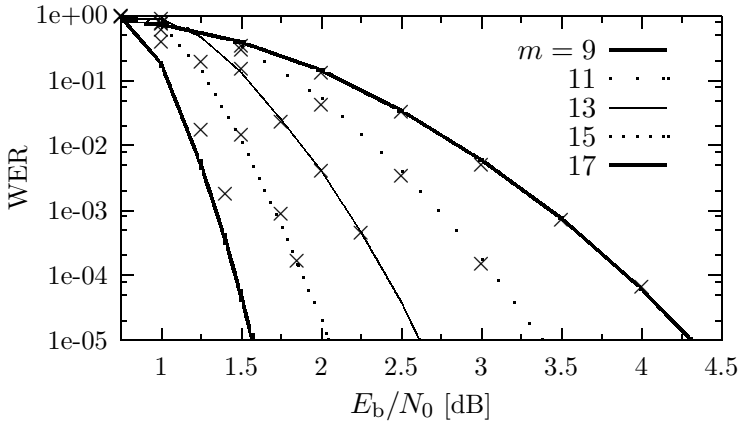

Abbildung 6.3.: Verschiedene OCBM der Länge $N = 2^m$ mit Rate $R = 0.5$ bei Übertragung über den AWGN-Kanal und bitweise MSD mit distanzbasierter Zuverlässigkeitsübergabe. Linien sind mit Hilfe den äquivalenten SNR geschätzte Wortfehlerraten, Marker simulierte Werte der mit Hilfe des äquivalenten SNR gefundenen OCBM

$D_{\mathrm{H,min}}$	$m = 9$	$m = 11$	$m = 13$	$m = 15$	$m = 17$
RM-Code	32	64	128	256	512
OCBM für WER $= 10^{-4}$	16	16	32	32	32

Tabelle 6.1.: Distanzen der in Bild 6.3 gezeigten OCBM bei WER $= 10^{-4}$ bei bitweiser MSD im Vergleich zu den RM-Codes gleicher Länge $N = 2^m$ und Rate $R = 0.5$.

der OCBM noch gleich dem RM-Code und hat daher auch die gleiche Mindestdistanz, so ist für $N = 2^{17}$ die Distanz des RM-Codes 16-mal größer als die des OCBM.

Während für moderate SNR-Werte die gesamte Gewichtsverteilung des Codes die Wortfehlerrate beeinflußt, wird sie für große SNR-Werte in erster Linie durch die Mindestdistanz des Codes bestimmt. Zudem ist für große SNR-Werte das Finden des richtigen Codewortes und damit die ML-Decodierung einfacher als bei kleinem SNR. Daß der OCBM für große SNR größtmögliche Distanz hat, ergibt sich aus folgendem Satz:

Satz 6.1 Für $N = 2^m$ und $K = \dim(\mathrm{RM}(r, m))$, $0 \le r \le m$ ist der für SNR $\to \infty$ mit Hilfe des äquivalenten SNR gefundene OCBM gleich dem Code $\mathrm{RM}(r, m)$.

Beweis: Da die geschätzten Einzelfehlerwahrscheinlichkeiten $P_e^{(i)}$ mit wachsendem äquivalenten SNR monoton fallen, genügt es, nur die SNR-Werte zu betrachten. Bei Zerlegung in zwei äußere Codes U und V gilt allgemein $\mathrm{SNR}^{(u)} = 2 \cdot \mathrm{SNR}$ und für große SNR nach Anhang A.7

$$\mathrm{SNR}^{(v)} \approx \mathrm{SNR} - 2 \ln 2.$$

Die Mindesthammingdistanz des Codes $\mathrm{RM}(r, m)$ sei D und betrachtet werden bei Zerlegung in 2^m äußere Codes nun zwei beliebige Codes $C^{(i_1)}$ und $C^{(i_2)}$, für deren zugehörige Zeilenvektoren der Generatormatrix des inneren Codes $\boldsymbol{B}_{2^m}$ gilt $w_{\mathrm{H}}(\boldsymbol{B}_{2^m,i_1}) = D$ und $w_{\mathrm{H}}(\boldsymbol{B}_{2^m,i_2}) = D/2$. Es läßt sich leicht zeigen, daß für die rekursiv berechneten äquivalenten SNR-Werte gilt

$$\begin{aligned} \mathrm{SNR}^{(i_1)} &\ge D \cdot [\mathrm{SNR} - (\log_2 N - \log_2 D) \cdot 2 \ln 2] \\ \mathrm{SNR}^{(i_2)} &\le D/2 \cdot \mathrm{SNR} - [\log_2 N - \log_2(D/2)] \cdot 2 \ln 2 \end{aligned}$$

und damit

$$\lim_{\mathrm{SNR} \to \infty} \frac{\mathrm{SNR}^{(i_1)}}{\mathrm{SNR}^{(i_2)}} = 2.$$

Daher gilt für die Einzelfehlerwahrscheinlichkeiten $P_e^{(i_1)} < P_e^{(i_2)}$, und in Schritt 2 der Konstruktionsvorschrift des OCBM werden bei der Auswahl der K äußeren Codes, für die $K^{(i)} = 1$ gesetzt wird, diejenigen bevorzugt, deren zugehöriges Gewicht $w_{\mathrm{H}}(\boldsymbol{B}_{2^m,i})$ am größten ist. Für SNR $\to \infty$ ist somit der optimierte Code für bitweise MSD gleich dem Code mit größter Distanz, der wiederum durch den RM-Code gegeben ist. □

Für große SNR geht damit die Konstruktionsvorschrift für den optimalen Code für bitweise MSD über in das klassische Prinzip, den Code mit größter Distanz zu konstruieren.

Zur Beurteilung der in Bild 6.3 gezeigten Ergebnisse lohnt ein Vergleich mit anderen Codierverfahren. So benötigt nach HAGENAUER der von der NASA bei der *Galileo*-Mission eingesetzte,

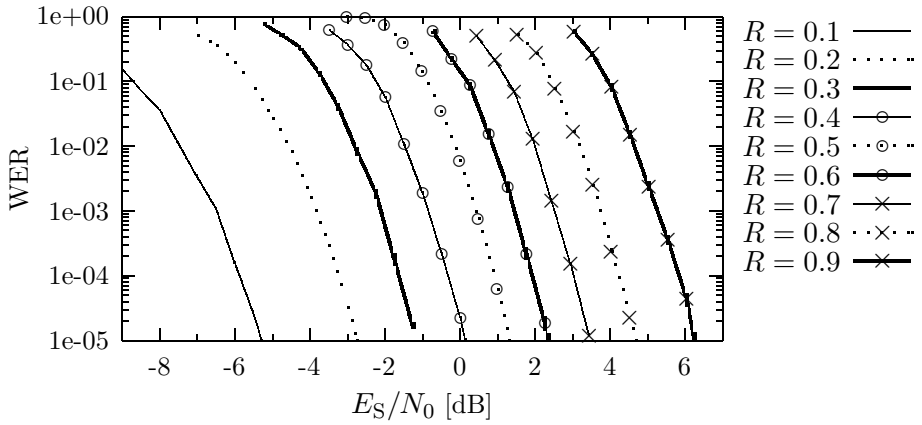

Abbildung 6.4.: Simulierte Wortfehlerraten für OCBM mit Länge $N = 512$ und bitweiser MSD, alles für den AWGN-Kanal

verkettete Code aus innerem Faltungscode der Rate 0.5, Gedächtnislänge $M = 6$ und zwei äußeren $(255, 233)$-REED-SOLOMON Codes für WER $= 10^{-3}$ ein SNR von $E_b/N_0 = 2.7\,\text{dB}$ und für BER $= 10^{-5}$ ein SNR von $E_b/N_0 = 2.6\,\text{dB}$ [66]. Hierbei wurde das Standardverfahren zur Decodierung angewendet, bei dem im ersten Schritt der Faltungscode mit Hilfe des VITERBI-Algorithmus und im zweiten Schritt die einzelnen REED-SOLOMON Codes unabhängig voneinander decodiert werden. Dagegen sind für den vergleichbaren $(8192, 4096)$-OCBM mit systematischer Codierung die entsprechenden SNR-Werte bei bitweiser MSD $E_b/N_0 = 2.17\,\text{dB}$ für WER $= 10^{-3}$ und[9] $E_b/N_0 = 2.25\,\text{dB}$ für BER $= 10^{-5}$, und damit ergibt sich zusätzlicher Codiergewinn von $0.5\,\text{dB}$ bzw. $0.35\,\text{dB}$. Noch deutlicher sind die Unterschiede bei dem von der ESA bei bei *Giotto*-Mission zum Kometen Halley eingesetzten Code mit acht äußeren $(255, 233)$-REED-SOLOMON Codes. Hier wird nach HAGENAUER für WER $= 10^{-3}$ ein SNR von $E_b/N_0 = 2.4\,\text{dB}$ und für BER $= 10^{-5}$ ein SNR von $E_b/N_0 = 2.35\,\text{dB}$ benötigt, falls das gleiche Decodierverfahren eingesetzt wird[10]. Dagegen sind für den vergleichbaren $(32768, 16384)$-OCBM die entsprechenden SNR-Werte $E_b/N_0 = 1.75\,\text{dB}$ für WER $= 10^{-3}$ und $E_b/N_0 = 1.85\,\text{dB}$ für BER $= 10^{-5}$.

Um die Flexibilität der Codekonstruktion zu verdeutlichen, sind in Bild 6.4 die Wortfehlerraten in Abhängigkeit von der Coderate für auf bitweise MSD optimierte Codes mit $N = 512$ gezeigt.

Selbstverständlich kann jeder OCBM, auch wenn er in Hinblick auf minimale Wortfehlerrate bei bitweiser MSD optimiert wurde, mit Hilfe des in Abschnitt 5.3 vorgestellten Listendecoders decodiert werden. In Bild 6.5 sind die damit erzielten Ergebnisse für $(N, K) = (512, 256)$ zusammen mit den Ergebnissen bei Listendecodierung des RM-Codes mit gleichen Parametern gezeigt. Es wurde in allen Fällen die Metrik Λ_y verwendet. Bei Wortfehlerraten $> 10^{-4}$ liefert eine Listendecodierung des RM-Codes mit Listengröße $L = 100$ schlechtere Ergebnisse als der OCBM mit Listengröße $L = 1$, was der bitweisen MSD entspricht. Allerdings ist der zusätzliche Codiergewinn für den OCBM beim Übergang von bitweiser MSD zur Listendecodierung nicht

[9] Siehe Kapitel 7.

[10] Natürlich werden in [66] auch verbesserte, iterative Decodierverfahren beschrieben. Beim besten, dem Autor bekannten Ergebnis für diese Codeklasse ist mit leicht unterschiedlicher Wahl der acht REED-SOLOMON Codes und iterativer Decodierung ein SNR von $E_b/N_0 = 1.6\,\text{dB}$ für BER $= 10^{-5}$ notwendig [26].

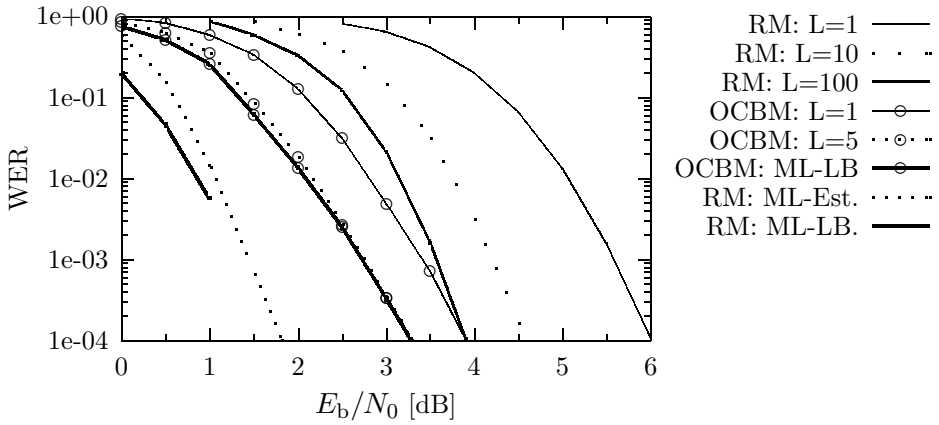

Abbildung 6.5.: Vergleich zwischen dem RM-Code $\mathrm{RM}(4, 9)$ und dem OCBM für $(N, K) = (512, 256)$ bei Listendecodierung, alles für den AWGN-Kanal. Für den RM-Code ist neben einer unteren Grenze (ML-LB) auch eine Schätzung (ML-Est.) für die WER bei ML-Decodierung gegeben.

groß. Schon bei $L = 5$ ist die Wortfehlerrate praktisch mit der ML-Decoders identisch, und eine weitere Erhöhung der Listengröße bringt keine Verbesserung. Für den RM-Code ist ebenfalls eine, durch Monte-Carlo-Simulation ermittelte untere Grenze sowie eine numerisch bestimmte Schätzung[11] für die WER des ML-Decoders gegeben. Danach ist bei SDML-Decodierung ein SNR von $E_b/N_0 \approx 1.8$ dB für WER $= 10^{-4}$ erforderlich, was relativ genau 2 dB unter dem erforderlichen SNR bei HDML-Decodierung liegt (s. Bild 5.11 auf Seite 64). Es zeigt sich, daß für den Fall des OCBM die bitweise Mehrstufendecodierung die Korrekturfähigkeit des Codes sehr gut ausnutzt, so daß der zusätzliche Codiergewinn durch eine optimale ML-Decodierung des Codes nicht groß ist. Das Korrekturpotential des RM-Code liegt dagegen weit über dem des OCBM, doch durch die bitweise MSD sowie durch Decodierung mit Listengröße $L = 100$ wird dieses Potential nicht annähernd ausgenutzt.

6.2. Optimierte Konstruktion für Mehrstufendecodierung auf Codewortebene

Die im vorigen Abschnitt vorgestellte Optimierungsstrategie hat das Ziel, den Code so anzupassen, daß bei bitweiser Mehrstufendecodierung die kleinste Wortfehlerrate erzielt wird. Der Übergang zur Listendecodierung, die in Abschn. 5.3 eingeführt wurde, ermöglicht eine gewisse Menge von verschiedenen Informationsbitfolgen gemeinsam zu betrachten und aus dieser Menge die besten Folgen auszuwählen. Damit wird für große Listenlängen die WER zunehmend nicht von den Einzelfehlerwahrscheinlichkeiten $P_e^{(i)}$ bei Zerlegung in äußere Codes der Länge $N^{(i)} = 1$, sondern von den Wortfehlerwahrscheinlichkeiten der Codes auf den Zwischenstufen der rekursiven Zerlegung bestimmt sowie der dazu notwendigen Listengröße.

[11] Für RM-Codes ist die Anzahl der Codeworte vom Gewicht $w_H(\boldsymbol{c}) < 2d_H(C)$ bekannt [30]. Für kleine Rauschleistung kann damit zusammen mit der Tangential-Bound von BERLEKAMP [2] die WER des ML-Decoders geschätzt werden.

Bei der Suche nach dem Code mit kleinster WER bei Listendecodierung mit vorgegebenen Listengrößen L_τ müßte eigentlich für jedes τ die Wahrscheinlichkeit, daß die richtige Folge $(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(\tau)})$ nicht in der Liste enthalten ist, analysiert werden. Dies ist generell nur sehr schwer, wenn überhaupt möglich. Daher werden im folgenden die verschiedenen Zerlegungsstufen betrachtet. Zur Codekonstruktion wird nun vereinfachend angenommen, daß auf einer bestimmten Stufe[12] ρ, $0 \leq \rho \leq \log_2 N$ für jeden der äußeren Codes $C^{(i)}$, $i = 1, 2, \ldots, 2^\rho$ jeweils nur ein Codewort für die folgenden Decodierschritte weiterverwendet würde, d.h. für jeden dieser Codes würde nicht eine Liste generiert, sondern eine Decodierentscheidung getroffen. Damit wäre der Decoder ein klassischer Mehrstufendecoder. Die bereits von PORTUGHEIS und später z.B. auch von WACHSMANN für Multilevel-Codes (MLC) vorgeschlagene Methode [45], [41], [64], die Raten der Codes an die sich durch die Verteilung der Zuverlässigkeitswerte ergebenden Kanalkapazitäten bzw. die Cutoff-Raten der äquivalenten Kanäle anzupassen, bietet sich nun auch hier an. Ein solcher, rekursiv konstruierter $|u|u+v|$-Code, dessen Raten der äußeren Codes unter Berücksichtigung der Kapazitäten bzw. der Cutoff-Raten gewählt wurden, wird im folgenden als Raten-angepaßter Code (RAC) bezeichnet. Es wurde schon erwähnt, daß für den Fall der MLC bei optimaler, d.h. wahrscheinlichkeitsbasierter Zuverlässigkeitsübergabe die Summe der Kanalkapazitäten der äquivalenten Kanäle gleich der Kanalkapazität des Übertragungskanals ist (s. z.B. [64]). Dies begründet auch die Tatsache, daß mit MSD die Kanalkapazität erreicht werden kann, sofern die Raten der äußeren Codes entsprechend zu $R^{(i)} = \mathcal{C}^{(i)}, \forall i$ gewählt werden. Es ist daher für diesen Fall nicht erforderlich, für den ganzen Code eine ML-Decodierung durchzuführen, man kann in suboptimaler Weise die einzelnen Teilcodes $C^{(i)}$ der Reihe nach entscheiden. Trotzdem müssen aber, um die Kanalkapazität zu erreichen, alle Teilcodes für sich optimal decodiert werden, und so stellt sich zumindest an dieser Stelle die Frage, ob durch die Zerlegung ein Gewinn an Decodierkomplexität erzielt werden kann.

Vergleicht man die ρ-stufige rekursive $|u|u+v|$-Konstruktion mit den MLC, so muß berücksichtigt werden, daß im Fall der rekursiven $|u|u+v|$-Verkettung jedes Codewort des inneren Codes mit Länge 2^ρ einem 2^ρ-närem Modulationssymbol des MLC entspricht, es werden jeweils die Codesymbole von 2^ρ äußeren Codes zu einem Symbol bzw. Vektor verknüpft. Daher liest sich die Tatsache, daß bei optimaler Zuverlässigkeitsübergabe die Summe der Kanalkapazitäten der äquivalenten Kanäle $\mathcal{C}^{(i)}$ gleich der Kanalkapazität des Übertragungskanals $\mathcal{C}$ ist, für den Fall der ρ-stufigen $|u|u+v|$-Verknüpfung als

$$\mathcal{C} = \frac{1}{2^\rho} \sum_{i=1}^{2^\rho} \mathcal{C}^{(i)}$$

während im Fall der MLC der Faktor $1/2^\rho$ wegfällt.

Das Gleichheitszeichen in obiger Gleichung gilt jedoch nicht für die Cutoff-Raten, deren Summe größer als die Cutoff-Rate des Übertragungskanals sein kann. Daß für die beiden in dieser Arbeit betrachteten Kanäle und $\rho = 1$ die Summe der Cutoff-Raten nicht nur *anwachsen kann*, sondern für alle sinnvollen Fehlerwahrscheinlichkeiten p bzw. SNR *garantiert anwächst*, besagt folgender Satz:

Satz 6.2 . Für den AWGN-Kanal sowie den BSC und einer Zerlegung von C in zwei äußere Codes U und V gilt für die Summe der Cutoff-Raten der äquivalenten Kanäle, $R_{\text{comp}}^{(u)}, R_{\text{comp}}^{(v)}$ für

[12] Es wird hier von Null gezählt, d.h. für die oberste Stufe (keine Zerlegung) ist $\rho = 0$.

$0 < E_S/N_0 < \infty$ bzw. $0 < p < 0.5$

$$R_{\text{comp}} < \frac{1}{2}(R_{\text{comp}}^{(u)} + R_{\text{comp}}^{(v)})$$

mit der Cutoff-Rate bei BPSK-Übertragung über den AWGN-Kanal bzw. den BSC R_{comp}.

Beweis: Für den AWGN-Kanal siehe[13] [41, Chapter 2]. Der dortige Beweis für 4-PSK ist hier direkt anwendbar. Der Beweis für den BSC ist im Anhang A.9 gegeben. □

Diese wichtige Eigenschaft ist gerade für die sequentielle- bzw. Listendecodierung der äußeren Codes wichtig, da hier die Cutoff-Rate die begrenzende Größe in Hinblick auf die erreichbare WER bei begrenzter Decodierkomplexität ist. (s. Seite 12). So ist z.B. für Coderate $R = 0.5$, BPSK-Übertragung über den AWGN-Kanal und sequentieller[14] Decodierung ohne Zerlegung, d.h. klassischer sequentieller Schätzung der Codesymbole, ein Signal-zu-Rauschverhältnis von $E_b/N_0 \geq 2.47\,\text{dB}$ notwendig, während bei ML-Decodierung für $N \to \infty$ ein SNR von $E_b/N_0 \geq 0.19\,\text{dB}$ ausreicht. Da bei Zerlegung in zwei äußere Codes, d.h. $\rho = 1$ die mittlere Cutoff-Rate

$$\overline{R}_{\text{comp}} = \frac{1}{2^\rho} \sum_{i=1}^{2^\rho} R_{\text{comp}}^{(i)} \tag{6.1}$$

über R_{comp} liegt, ist damit umgekehrt auch ein geringeres SNR notwendig, um bei sequentieller Decodierung der äußeren Codes (zumindest theoretisch) fehlerfrei decodieren zu können. So ergibt sich aus Bild[15] 6.6 für $\rho = 1$ ein notwendiges SNR von $E_b/N_0 \approx 2\,\text{dB}$ für $\overline{R}_{\text{comp}} = 0.5$ und damit ein zusätzlicher Gewinn von fast $0.5\,\text{dB}$ gegenüber der Betrachtung für $\rho = 0$. Natürlich müssen dazu die Raten der äußeren Codes zu $R^{(i)} = R_{\text{comp}}^{(i)}$ gewählt werden. Wenn nun die Cutoff-Rate als Kriterium für praktikable Codierverfahren betrachtet wird, ergibt sich damit ein Komplexitätsgewinn gegenüber $\rho = 0$.

Zur Berechnung von $\overline{R}_{\text{comp}}$ nach Gl. (6.1) bietet sich auch hier das äquivalente SNR an, da der Fehler zwischen tatsächlicher Cutoff-Rate $R_{\text{comp}}^{(i)}$ und dem mit Hilfe des äquivalenten SNR geschätzten $\widetilde{R}_{\text{comp}}^{(i)}$ nur gering ist (s. Bild 3.10 auf S. 28). Auf diese Weise kann, wie zur Berechnung von Bild 6.6 geschehen, die mittlere Cutoff-Rate auch bei Zerlegung in mehr als zwei äußere Codes näherungsweise bestimmt werden. Für große ρ wird die Näherung natürlich zunehmend ungenau, aber bis $\rho = 7$ liefern die damit ermittelten Werte brauchbare Abschätzungen für das tatsächlich notwendige SNR, um kleine Wortfehlerwahrscheinlichkeiten zu erzielen.

Der große Unterschied zwischen rekursiver $|u|u+v|$-Konstruktion zu den MLC ist, daß der Betrachtungspunkt, an dem die äquivalenten Kanäle analysiert und die Raten der äußeren Code angepaßt werden, bei MLC durch den Übergang von binären Codesymbolen zum Modulationssymbol festgelegt ist, während bei der $|u|u+v|$-Konstruktion jede beliebige Zerlegungsstufe ρ zur Ratenanpassung gewählt werden kann. So kann der Effekt, daß die mittlere Cutoff-Rate über der Cutoff-Rate des Übertragungskanals liegt, bei MLC nur wenig genutzt werden. Die

[13] Nach [41] soll der Beweis schon in [72] gegeben sein.

[14] Auch wenn in Abschnitt 5.3 gezeigt wurde, daß für die untersuchte Codeklasse mit Listendecodierung wesentlich bessere Ergebnisse erzielt werden als mit der sequentiellen Decodierung, so wird hier doch beispielhaft das sequentielle Verfahren betrachtet, da hier die Cutoff-Rate eine 'harte' Grenze ist. R_{comp} ist zwar bei Listendecodierung keine 'harte' Grenze, dient aber doch als Orientierung für praktikable Verfahren.

[15] Außer für $\rho = 0$ sind alle Kurven für $\overline{R}_{\text{comp}}$ mit Hilfe des äquivalenten SNR geschätzt.

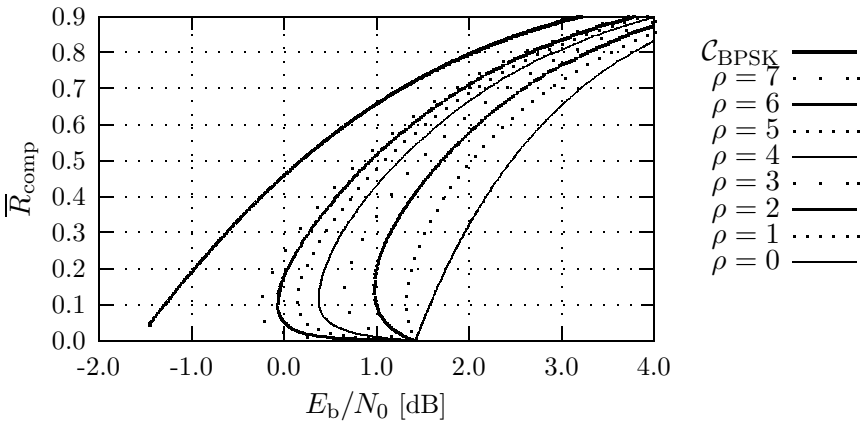

Abbildung 6.6.: Basierend auf dem äquivalenten SNR geschätztes notwendiges E_b/N_0 für gemittelte Cutoff-Raten der äquivalenten Kanäle in Abhängigkeit von ρ und die Kanalkapazität für BPSK-Übertragung, alles für den AWGN-Kanal

rekursive $|u|u+v|$-Konstruktion bietet jedoch die Möglichkeit, diesen Effekt durch Wahl von ρ zu verstärken, so daß nach Bild 6.6 z.B. für $\overline{R}_{\text{comp}} = 0.5$ bei $\rho = 7$ nur ein SNR von näherungsweise $E_\text{b}/N_0 \approx 0.8\,\text{dB}$ erforderlich ist. Für die Codelänge der äußeren Codes gilt dann $N^{(i)} = N/2^\rho, \forall i$. Hier zeigt sich auch die Grenze bei der Wahl von ρ, denn für große ρ sind die äußeren Codes sehr kurz und damit ist die Cutoff-Rate für die Beurteilung der Fehlerwahrscheinlichkeit bzw. der Decodierkomplexität nicht mehr geeignet. Generell muß daher bei der Wahl von ρ ein Kompromiß zwischen erforderlichem SNR für eine bestimmte Cutoff-Rate und der Codelänge der äußeren Codes gesucht werden.

Allgemein kann das notwendige SNR für $\overline{R}_{\text{comp}} = R$ als untere Grenze angesehen werden, um mit dem hier vorgestellten Listendecodierverfahren zumindest theoretisch kleine Wortfehlerraten erzielen zu können. Das tatsächliche SNR wird daher über diesem Wert liegen und so ist für ein gegebenes SNR die mittlere Cutoff-Rate größer als die tatsächliche Coderate. In diesem Fall kann die Regel $R^{(i)} = R^{(i)}_{\text{comp}}$, da für die mittlere Wortfehlerwahrscheinlichkeit, gemittelt über alle möglichen Codes $C^{(i)}$ mit Rate $R^{(i)}$ gilt [19][16]

$$\overline{P}^{(i)}_w \leq 2^{-N^{(i)}(R^{(i)}_{\text{comp}} - R^{(i)})}$$

in kanonischer Weise geändert werden zu[17]

$$R^{(i)}_{\text{comp}} - R^{(i)} = \min(R^{(i)}_{\text{comp}}, \Delta) \quad \forall i. \tag{6.2}$$

Die Größe[18] $\Delta \in \mathbb{R}^+$ ergibt sich dabei aus der Randbedingung $R = 2^{-\rho} \sum_i R^{(i)}$. Damit sind

[16] Die eigentliche Aussage des R_0-Theorems ist, daß mindestens ein Code mit dieser Wortfehlerwahrscheinlichkeit existiert. Diese Aussage resultiert jedoch aus der Erkenntnis, daß dies auch für die mittlere WER über alle zufällig gewählten Codes gilt.

[17] Dies ist eine vereinfachte Form der *Raten Design Regel Nr. 3* von WACHSMANN [63], bei der die Raten der äußeren Codes so angepaßt werden, daß die Fehlerexponenten nach GALLAGER [20] für alle äquivalenten Kanäle gleich sind. Es wird hier jedoch die einfache Form des Fehlerxponenten gewählt, da, wie sich später herausstellen wird, sich schon mit dieser Regel sehr gute Codes konstruieren lassen, die nahe am Optimum sind.

[18] Falls das tatsächliche SNR unter dem notwendigen SNR für $\overline{R}_{\text{comp}} = R$ liegt, ist verallgemeinert $\Delta \in \mathbb{R}$. Bei der Bestimmung der Raten $R^{(i)}$ ist dann zu beachten, daß $R^{(i)} \leq 1$ gelten muß.

die Fehlerwahrscheinlichkeiten $\overline{P}_w^{(i)}$, zumindest gemäß dieser einfachen Abschätzung, näherungsweise gleich. Falls alle so gefundenen Raten $R^{(i)}$ echt größer Null sind, sind sie identisch mit den Raten, die nach der in [45, Chapter 5] von PORTUGHEIS für M-PSK gegebenen Regel gefunden werden, sofern diese an die $|u|u+v|$-Verkettung angepaßt wird[19].

Ein neben dem Parameter ρ weiterer Freiheitsgrad ist die Wahl der äußeren Codes $C^{(i)}$, $i = 1, 2, \ldots, 2^\rho$. Zwar ist deren Codelänge durch $N^{(i)} = N/2^\rho$, $\forall i$ vorgegeben, und die Rate sollte die Cutoff-Rate des dazugehörigen äquivalenten Kanals nicht überschreiten. Aus der Vielzahl der verschiedenen durch die rekursive $|u|u+v|$-Konstruktion generierbaren Codes mit gleichen Parametern $(N^{(i)}, K^{(i)})$ sind jedoch geeignete auszuwählen. Allgemein können alle RM-Codes bis zu einer Länge von $N^{(i)} \leq 128$ auch mit kleiner Listengröße nahezu ML-decodiert werden (s. Bild 5.6 S. 54), bei größeren Codelängen gilt dies jedoch nicht mehr[20]. Daher wurden in dieser Arbeit bei der auf die Cutoff-Raten angepaßten Konstruktion Codelängen $N^{(i)} \in \{64, 128\}$ betrachtet und bei jeweils gegebenen $N^{(i)}$ und $K^{(i)}$ der Code $C^{(i)}$ mit maximaler Mindestdistanz gewählt[21].

So ergibt sich bei vorgegebenen SNR, $N = 2^m, K \in \{1, 2, \ldots, N\}$ und $0 \leq \rho \leq m$ folgendes Konstruktionsverfahren

Schritt 1: Aus dem SNR des Kanals Bestimmung der Cutoff-Raten $R_{\text{comp}}^{(i)}$ der 2^ρ äquivalenten Kanäle für die Codes $C^{(i)}$

Schritt 2: Aus den Cutoff-Raten Bestimmung der Hilfsgröße $\Delta \in \mathbb{R}^+$, so daß gilt

$$K = \sum_i N^{(i)} \cdot \max(0, R_{\text{comp}}^{(i)} - \Delta).$$

Es wird $K^{(i)} = N^{(i)} \cdot \max(0, R_{\text{comp}}^{(i)} - \Delta)$ gesetzt.

Schritt 3: Aus der Menge der $(N^{(i)}, K^{(i)})$-Codes mit maximaler Distanz Auswahl desjenigen Code für $C^{(i)}$, dessen Einzelfehlerwahrscheinlichkeiten bei rekursiver Zerlegung in Codes der Länge 1 minimal sind.

Der nach diesem Prinzip gefundene Code wird im folgenden als *linear Raten-angepaßter Code* mit Parameter ρ (LRAC(ρ)) bezeichnet, da die Größe Δ linear in die Ratenbestimmung einfließt und damit für alle äußeren Codes die Differenz zwischen Cutoff-Rate und tatsächlicher Coderate gleich ist.

Für $\rho = 1$ und $(N, K) = (512, 256)$ sind in Bild 6.7 die sich mit diesem Verfahren ergebenden Raten für die beiden Codes $C^{(1)} = V$ und $C^{(2)} = U$ gezeigt[22]. Bei $E_b/N_0 \approx 2$ dB, das dem notwendigen SNR für $\overline{R}_{\text{comp}} = 0.5$ entspricht, ist $\Delta = 0$ und es ergibt sich $K^{(v)} = 69$ und

[19] Der Fall $\Delta > R_{\text{comp}}^{(i)}$ wurde in [45] allerding nicht korrekt behandelt, hier würden sich negative Coderaten $R^{(i)}$ ergeben.

[20] So ist beim RM-Code mit Parametern $(N, K, D) = (256, 93, 32)$ eine Listendecodierung mit $L \approx 250$ noch ca. 0.7 dB schlechter als eine ML-Decodierung [55], [56].

[21] Falls es bei gegebenen $(N^{(i)}, K^{(i)})$ mehrere Codes mit gleicher Mindestdistanz gibt, dann wurde derjenige ausgewählt, der bei bitweiser MSD die kleinste Fehlerwahrscheinlichkeit ergibt, d.h. dessen Einzelfehlerwahrscheinlichkeiten $P_e^{(i)}$ bei Zerlegung in Codes der Länge 1 minimal sind.

[22] Siehe Fußnote 18

$K^{(u)} = 187$. Nahezu die gleichen Werte ergeben sich bei der Konstruktion des OCBM nach Abschnitt 6.1, für dessen äußere Codes $K^{(v)} = 66$ und $K^{(u)} = 190$ gilt. Für gleiche Werte $K^{(v)}$ und $K^{(u)}$ liegt der Unterschied zwischen OCBM und LRAC(1) in der Wahl der äußeren Codes V, U, die beim LRAC im Hinblick auf maximale Mindestdistanz gewählt werden und beim OCBM im Hinblick auf kleine Fehlerwahrscheinlichkeit bei bitweiser MSD. Weiter ergibt sich bei $E_\mathrm{b}/N_0 \approx 6$ dB für den LRAC(1) $K^{(v)} = 93$ und $K^{(v)} = 163$, so daß hier der LRAC gleich dem REED-MULLER Code ist. Daß für ein bestimmtes SNR der LRAC gleich dem Code mit maximaler Mindestdistanz ist, gilt allerdings nur für die beiden Spezialfälle $\rho \in \{0, 1\}$, für größere ρ gibt es in der Regel kein SNR, bei dem beide Codes identisch sind.

Für große SNR zeigt sich auch die Schwäche bei der Konstruktion des LRAC: da die Cutoff-Raten beider äquivalenten Kanäle für große SNR gegen Eins streben, nähern sich für große SNR auch beide Raten $R^{(v)}$ und $R^{(u)}$ aneinander an. Allgemein gilt daher für den LRAC(ρ)[23]

$$\lim_{\mathrm{SNR}\to\infty} K^{(i)} = 2^{-\rho} \cdot K.$$

und damit hat der LRAC für große SNR, im Gegensatz zum OCBM, nicht maximale Distanz. Für große SNR wird die Wortfehlerrate jedoch in erster Linie von der Mindestdistanz des Codes bestimmt, und da nach Korollar 5.1 (S. 44) bei der bitweisen MSD alle Fehlervektoren $\boldsymbol{n}$ mit

$$||\boldsymbol{n}|| < \frac{d_{\mathrm{E,min}}(C)}{2}$$

korrigiert werden, liefert asymptotisch eine bitweise MSD des OCBM kleinere Wortfehlerraten als eine ML-Decodierung des LRAC. Weil die Codestruktur nur begrenzt in die Konstruktion des LRAC einfließt, ergibt sich für große SNR ein Code, der zwar mit geringer Komplexität (d.h. kleiner Liste) nahezu ML-decodierbar ist, dessen Gesamteigenschaften aber nicht optimal sind. Somit eignet sich die *linear Raten-angepaßte Konstruktion* hauptsächlich für kleine SNR-Werte, bei denen die Wortfehlerrate in erster Linie von der Decodierbarkeit und nicht von der Mindestdistanz des Codes bestimmt ist.

In Bild 6.8 wird die erzielbare Wortfehlerrate des LRAC für $\rho = 3$, womit die besten Ergebnisse erzielt wurden, mit der des OCBM für $(N, K) = (512, 256)$ verglichen. Zur Decodierung wurde in beiden Fällen ein Listendecoder nach Abschnitt 5.3 mit der Metrik Λ_y verwendet. Die Listengrößen wurden beim OCBM zu $L_\tau = 5$, $\tau \in \{1, 2, \ldots, 512\}$ und beim LRAC zu $L_\tau = 10$, $\tau \in \{1, 2, \ldots, 512\}$ gewählt, und die sich damit ergebenden Fehlerraten sind für WER ≤ 0.1 praktisch mit der des ML-Decoders identisch (vergl. Bild 6.5), so daß eine Decodierung mit größerer Liste nicht zu einer wesentlichen Verbesserung führt. Die an die Coderaten angepaßte Konstruktion ermöglicht hier zwar einen zusätzlichen Codiergewinn von ca. 0.25 dB gegenüber einer ML-Decodierung des OCBM, aber der begrenzende Faktor im Hinblick auf die erreichbare WER ist in beiden Fällen die Korrekturfähigkeit des Codes und nicht die notwendige Komplexität zur Decodierung. Eine Decodierung mit größerer Liste ist somit nur bei einer gleichzeitigen Verbesserung des Codes sinnvoll.

Um die Korrekturfähigkeit des Codes zu verbessern, ist es notwendig, die Ratenaufteilung der äußeren Codes nicht allein an den Cutoff-Raten der äquivalenten Kanäle zu orientieren. Die

[23] Leider konnte nicht gezeigt werden, daß dieser Zusammenhang auch gilt, wenn die Raten der äußeren Codes anstatt mit der einfachen Form des Fehlerexponenten (gemäß Gl. (6.2)) mit den Isoquanten des GALLAGER-Fehlereponenten gemäß der *Raten Design Regel Nr. 3* von WACHSMANN [63] bestimmt werden.

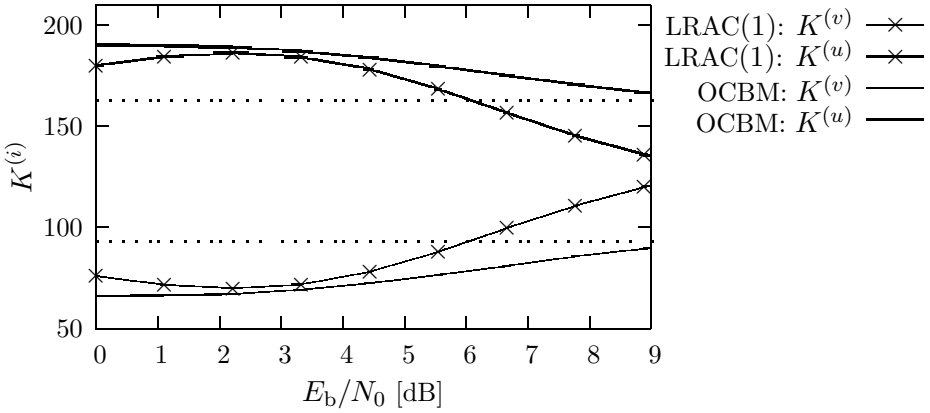

Abbildung 6.7.: Aufteilung der K Informationssymbole auf die Codes U und V für verschiedene Konstruktionsverfahren eines $(512, 256)$-Codes für den AWGN-Kanal. Die gestrichelten Linien geben die Werte $K^{(u)} = 163$ (oben) und $K^{(v)} = 93$ (unten) des Codes $\mathrm{RM}(4, 9)$ an.

Korrekturfähigkeit des Codes mit größter Mindestdistanz kann jedoch bei langen Codes, wie in Abschnitt 5.3.4 gezeigt wurde, mit dem hier vorgestellten Listendecodierverfahren nicht ausgenutzt werden. Es muß daher eine Ratenaufteilung gefunden werden, die sich an beiden Extremen (der Aufteilung gemäß der Cutoff-Raten sowie gemäß der maximalen Distanz) orientiert. Die optimale Ratenaufteilung für gegebene Listengröße und gegebenes SNR kann jedoch nicht berechnet werden, es ist eine Aufteilung der Informationssymbole *per Hand* auf die einzelnen äußeren Codes $C^{(i)}$ erforderlich. Einige auf diese Art gewonnen Ergebnisse wurden auch hier schon in[24] [38] dokumentiert. Als Ausgangspunkt kann ein Code, der unter der Bedingung $R^{(i)} \leq R^{(i)}_{\text{comp}}, \forall i$ nahezu[25] maximales Korrekturpotential besitzt, dienen. Ein solcher Code kann gefunden werden, indem für alle i, deren Cutoff-Rate $R^{(i)}_{\text{comp}}$ kleiner als die dazugehörige Rate $R^{(i)}_{D_{\max}}$ des Codes mit maximaler Distanz ist, zu $R^{(i)} = R^{(i)}_{\text{comp}}$ gesetzt wird, bei den restlichen äußeren Codes ist $R^{(i)}$ möglichst klein aber größer als $R^{(i)}_{D_{\max}}$ zu wählen. Für diesen Code ist allerdings die erforderliche Listengröße für eine nahezu optimale Decodierung sehr groß, da die Raten einiger äußerer Codes identisch mit den dazugehörigen Cutoff-Raten sind. Der beste Code für gegebenes L und SNR muß daher durch den Vergleich der tatsächlichen Fehlerraten verschiedener Codes, deren Coderaten $R^{(i)}$ um den Ausgangscode variieren, bestimmt werden. Dieser Code wird im folgenden als *per Hand Raten-angepaßter Code* mit Parameter ρ (HRAC(ρ)) bezeichnet. In den meisten Fällen sind beim gefundenen HRAC die Raten, für die beim Ausgangscode $R^{(i)} = R^{(i)}_{\text{comp}}$ gilt, etwas reduziert und im Gegenzug die restlichen Raten etwas angehoben.

In Tabelle 6.2 sind für $L = 100$ und $L = 500$ jeweils die $K^{(i)}$ derjenigen Codes aufgelistet, deren ermittelte[26] WER unter den verschiedenen getesteten Codes für $E_b/N_0 = 2.5$ dB am geringsten war. Zum Vergleich sind die Werte $N^{(i)} \cdot R^{(i)}_{\text{comp}}$, die Werte des LRAC sowie des Codes

[24] Diese Arbeit wurde vom Autor angeleitet.

[25] Es ist allgemein sehr schwer, unter den gegebenen Randbedingungen den Code mit *maximalem* Korrekturpotential zu bestimmen.

[26] Die verschiedenen Wortfehlerraten wurden hier durch Monte-Carlo-Simulationen bestimmt.

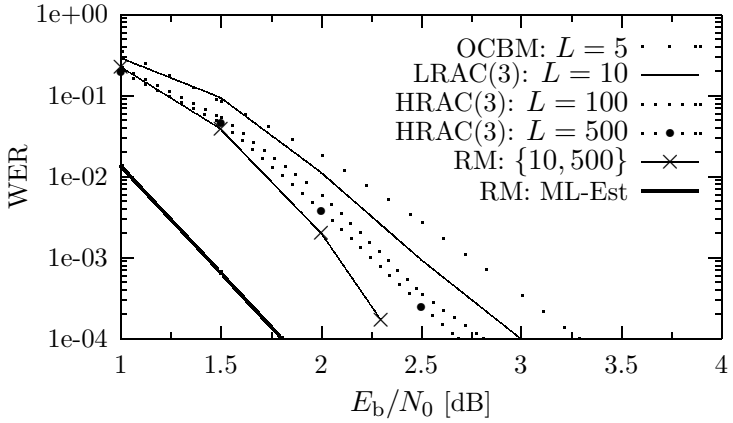

Abbildung 6.8.: Wortfehlerraten verschiedener Codes mit $(N, K) = (512, 256)$ bei Übertragung über den AWGN-Kanal. Für den RM-Code ist neben einer Schätzung der WER bei ML-Decodierung (ML-Est.) auch die WER des Permutationsdecoders mit $\{L, n_p\} = \{10, 500\}$ gegeben. Bezüglich der Codes HRAC(3) s. Tabelle 6.2

$K^{(i)}$	1	2	3	4	5	6	7	8
$N^{(i)} \cdot R^{(i)}_{\text{comp}}$	2	17	22	52	29	57	59	64
LRAC(3)	0	10	16	45	23	51	53	58
HRAC(3), $L = 100$	2	14	19	42	22	50	50	57
HRAC(3), $L = 500$	3	15	19	42	22	49	49	57
RM$(4, 9)$	7	22	22	42	22	42	42	57

Tabelle 6.2.: Aufteilung der $K = 256$ Informationssymbole auf die äußeren Codes bei $E_b/N_0 = 2.5$ dB für verschiedene $(512, 256)$-Codes

RM$(4, 9)$ aufgeführt. Bis auf eine Ausnahme (die Rate $R^{(1)}$ des HRAC für $L = 500$) gilt für alle Raten der gefunden Codes $R^{(i)} \leq R^{(i)}_{\text{comp}}$. Weiterhin sind die $K^{(i)}$ beider HRAC für $i = 4, 5, 8$ identisch mit denen des RM-Codes $R^{(i)}_{D_{\max}}$, eine Erhöhung dieser Raten führt automatisch zu einer größeren Fehlerwahrscheinlichkeit. Der zusätzliche Codiergewinn des HRAC für $L = 500$ gegenüber dem LRAC ist mit ca. 0.3 dB bei WER $= 10^{-4}$ allerdings sehr gering[27], verglichen mit der dazu notwendigen Vergrößerung der Liste um den Faktor 50. Zumindest für die hier betrachteten SNR-Werte im Bereich um 3 dB ist damit der LRAC für $(N, K) = (512, 256)$ ein guter Kompromiß zwischen Decodierkomplexität und Wortfehlerwahrscheinlichkeit.

Während beim $(512, 256)$-Code der zusätzliche Codiergewinn beim Übergang vom OCBM zum LRAC nur gering ist (ca. 0.25 dB bei WER = 10^{-4}, was bei fester Rauschleistung einer Verminderung der notwendigen Signalleistung von weniger als 6% entspricht), nimmt er bei Codes mit großer Codelänge zu. Beispielhaft sind in Bild 6.9 die Wortfehlerraten des OCBM und LRAC(4) mit Codelänge $N = 2048$ und Rate $R = 0.5$ zusammen mit einer unteren Gren-

[27] Es sollte angemerkt werden, daß selbst die besten Ergebnisse, die dem Autor für binäre Codes vergleichbarer Länge und Coderate bekannt sind, bei WER$= 10^{-4}$ ein ähnliches SNR erfordern. So war beispielsweise in [17] bei der Decodierung eines $(504, 252)$-LDPC Codes ein SNR von $E_b/N_0 = 2.55$ dB erforderlich und in [65] bei einem $(N, K) = (500, 220)$-Faltungscode ($R = 0.44$) mit Gedächtnislänge 30 ein SNR von $E_b/N_0 = 2.34$ dB.

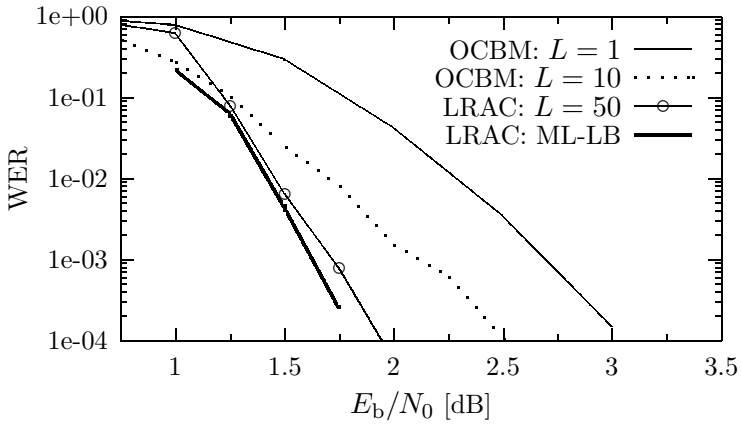

Abbildung 6.9.: Wortfehlerraten verschiedener Codes mit $(N, K) = (2048, 1024)$ bei Übertragung über den AWGN-Kanal

ze bei ML-Decodierung des LRAC (ML-LB) gezeigt. Für die gewählten Listengrößen von $L = 10$ bzw. $L = 50$ sind die erzielten Wortfehlerraten in beiden Fällen nahe an denen des ML-Decoders, so daß auch hier durch eine Vergrößerung nur noch ein geringer Gewinn möglich ist. Hier ist der zusätzliche Codiergewinn des LRAC gegenüber dem OCBM bei WER = 10^{-4} etwa 0.5 dB (was einer Reduzierung der notwendigen Signalleistung um ca. 11% entspricht). Ein weiterer Beleg für die Nützlichkeit des äquivalenten SNR zur Charakterisierung der äquivalenten Kanäle ist ein Vergleich mit Bild 6.6, wonach für $\rho = 4$ ein notwendiges SNR von $E_b/N_0 \approx 1.25\,\text{dB}$ für $\overline{R}_{\text{comp}} = 0.5$ geschätzt wird. Dieser Wert stimmt relativ gut mit dem Punkt überein, an dem die Wortfehlerwahrscheinlichkeitskurve des LRAC nach unten knickt.

Zusammenfassung

In diesem Kapitel wurden prinzipiell zwei verschieden Konstruktionsverfahren für rekursiv konstruierte $|u|u+v|$-Codes vorgestellt. Es zeigte sich, daß das *äquivalente SNR* für Codelängen bis 2^{15} Codesymbole geeignet ist, um bei bitweiser Mehrstufendecodierung den Code mit kleinster Wortfehlerwahrscheinlichkeit zu finden. Auch für die Listendecodierung kann mit Hilfe des äquivalenten SNR die Cutoff-Rate hinreichend genau abgeschätzt werden, um für Codes bis zur Länge 2048, und wie im nächsten Kapitel gezeigt wird auch bei größeren Codelängen, die notwendigen Coderaten der äußeren Codes zu bestimmen.

Generell erweisen sich beide hier vorgestellten Konstruktionsverfahren als sehr vorteilhaft. Zum einen liefert die bitweise Mehrstufendecodierung der darauf optimierten Codes gerade bei großen Codelängen relativ kleine Fehlerwahrscheinlichkeiten. Zwar gehören die damit erzielten Ergebnisse nicht zu den besten, sie sind aber bei großen Codelängen durchaus mit anderen Verfahren wie z.B. den verketteten Systemen aus innerem Faltungscode und äußeren REED-SOLOMON Codes vergleichbar. Auf der anderen Seite ermöglicht eine auf die Listendecodierung angepaßte Codekonstruktion eine weitere Verbesserung der Ergebnisse, wenn zur Decodierung eine größere Komplexität zulässig ist.

Eine weitere wesentliche Erkenntnis diese Kapitels ist die Tatsache, daß die Cutoff-Rate des Übertragungskanals auch bei Anwendung eines Listendecoders, wie er in Abschnitt 5.3 vorge-

stellt wurde, keine begrenzende Größe für die Coderate ist.

7. Simulationsergebnisse

In diesem Kapitel werden zum Abschluß der Arbeit noch einige Ergebnisse präsentiert, um die Leistungsfähigkeit der vorgestellten Decodier- und Konstruktionsverfahren zu verdeutlichen. Der folgende erste Abschnitt beschränkt sich hierbei auf RM-Codes mittlerer Länge, bei denen mit Listendecodierung verschiedener Permutationen des Empfangsvektors nahezu optimale Wortfehlerwahrscheinlichkeiten erzielt werden können. Im zweiten Abschnitt werden noch einmal die verschiedenen Konstruktionsverfahren des letzten Kapitels am Beispiel von Codes der Rate $R = 0.5$ miteinander verglichen.

7.1. REED-MULLER Codes

In diesem Abschnitt werden RM- Codes der Länge $N = 256$ und $N = 512$ betrachtet. Alle Ergebnisse setzen eine Übertragung über den AWGN-Kanal voraus.

In Bild 7.1 ist die Wortfehlerwahrscheinlichkeit für den Code RM$(3, 8)$ mit Parametern $(N, K, D) = (256, 93, 32)$ in Abhängigkeit vom quadratischen euklidischen Abstand zwischen Empfangsvektor und gesendetem Codewort gezeigt. Es wird die Notation $\{L, n_p\}$ mit der Listengröße L und Anzahl der zufällig gewählten Permutationen n_p verwendet. Aus den $|P| = 255$ verschiedenen Permutationen, die bei einer Zerlegung in zwei äußere Codes U und V zu verschieden Paaren bei der Bestimmung der $\widetilde{v}_k$, $k = 1, 2, \ldots, 128$ führen, wurden für jeden Empfangsvektor $\boldsymbol{y}$ jeweils neu n_p Permutationen zufällig ausgewählt. Prinzipiell können für diesen Code durch eine reine Listendecodierung ohne Permutationen relativ gute Ergebnisse erzielt werden, bei sehr großer Liste sogar nahe am Optimum. Es zeigt sich jedoch auch hier der Vorteil durch Decodierung verschiedener Permutationen, so daß mit $L = 10$ und $n_p = 25$ Wortfehlerraten nahe der des ML-Decoders erreicht werden.

Die Wortfehlerwahrscheinlichkeiten für den Code RM$(4, 8)$ mit Parametern $(N, K, D) = (256, 163, 16)$ sind in Bild 7.2 gezeigt. Ebenso wie beim Code RM$(3, 9)$ wurden die Permutationen im Hinblick auf verschiedene Paare bei der Bestimmung der $y_k^{(v)}$, $k = 1, 2, \ldots, 128$ gewählt. Mit $\{L, n_p\} = \{10, 25\}$ ist auch hier nahezu eine ML-Decodierung möglich. Generell sind damit alle RM-Code mit der Länge $N \leq 256$ nahezu ML-decodierbar, da die erforderliche Komplexität für eine ML-Decodierung der Codes aus dieser Gruppe für die beiden hier untersuchten RM$(3, 8)$ und RM$(4, 8)$ am größten ist.

Für eine Wortfehlerrate nahe der des ML-Decoders sind im Fall des RM$(3, 9)$ Codes mit Parametern $(N, K, D) = (512, 130, 64)$ wesentlich mehr Permutationen als bei den RM-Codes der halben Länge erforderlich. So ist das notwendige SNR für WER $= 10^{-4}$ bei Decodierung mit $\{L, n_p\} = \{10, 500\}$ ähnlich wie bei Code RM$(4, 9)$ noch etwa 0.5 dB größer als bei ML-Decodierung (s. Bild 7.4). Allerdings ist damit, im Vergleich zum $\{1, 1\}$-Decoder, der mit dem

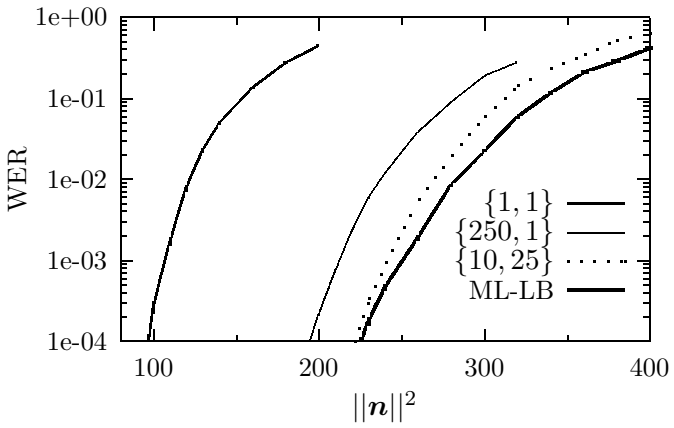

Abbildung 7.1.: Wortfehlerraten für den Code RM$(3, 8)$ für verschiedene Listengrößen L und Anzahl der Permutationen n_p, bezeichnet mit $\{L, n_p\}$, in Abhängigkeit von $||\boldsymbol{n}||^2 = d_{\mathrm{E}}^2(\boldsymbol{y}, \boldsymbol{c})$

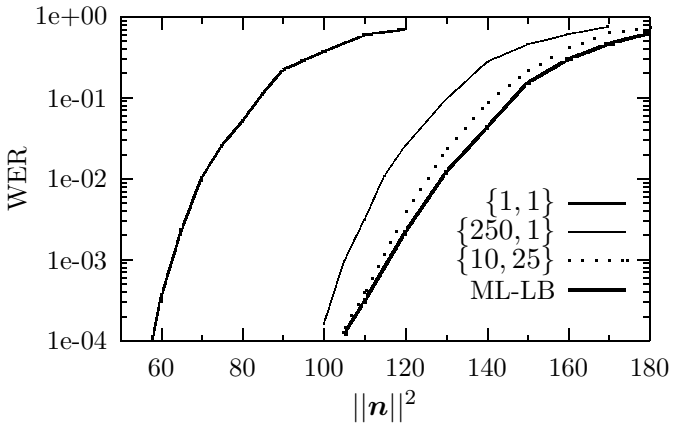

Abbildung 7.2.: Wortfehlerraten für den Code RM$(4, 8)$ für verschiedene Listengrößen L und Anzahl der Permutationen n_p, bezeichnet mit $\{L, n_p\}$, in Abhängigkeit von $||\boldsymbol{n}||^2 = d_{\mathrm{E}}^2(\boldsymbol{y}, \boldsymbol{c})$

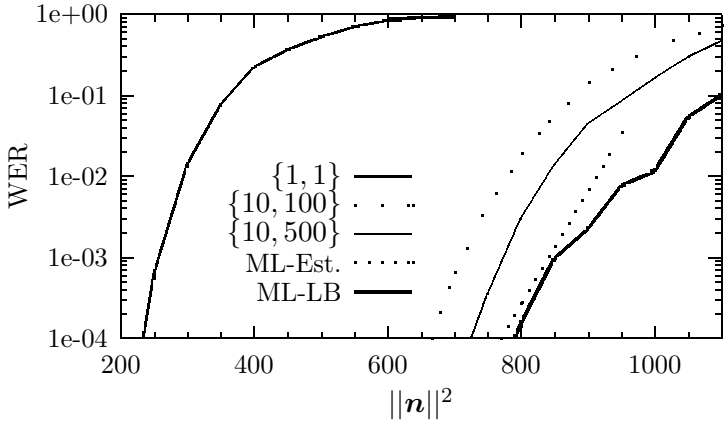

Abbildung 7.3.: Wortfehlerraten für den Code RM$(3,9)$ für verschiedene Listengrößen L und Anzahl der Permutationen n_p, bezeichnet mit $\{L, n_p\}$, in Abhängigkeit von $||\boldsymbol{n}||^2 = d_{\mathrm{E}}^2(\boldsymbol{y}, \boldsymbol{c})$

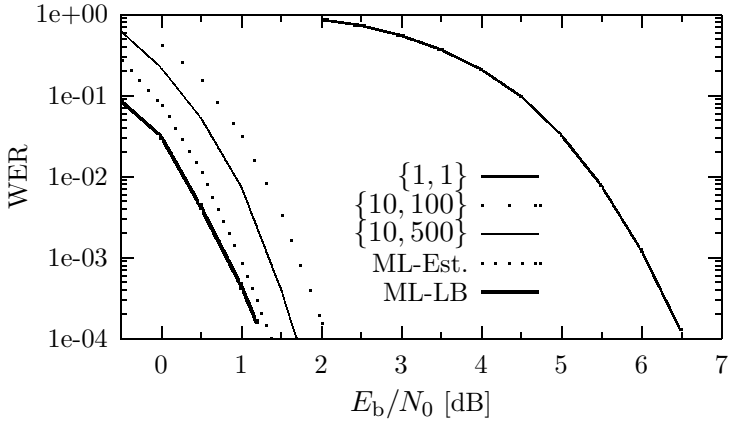

Abbildung 7.4.: Wortfehlerraten für den Code RM$(3,9)$ für verschiedene Listengrößen L und Anzahl der Permutationen n_p, bezeichnet mit $\{L, n_p\}$ bei Übertragung über den AWGN-Kanal

in [49] vorgestellten Decoder identisch ist, ein zusätzlicher Codiergewinn von fast 5 dB möglich. Die Schätzung für die Wortfehlerwahrscheinlichkeit des ML-Decoders (ML-Est.) wurde in Bild 7.3 mit der Sphere-Bound von HUGHES [25] und in Bild 7.4 mit der Tangential-Bound von BERLEKAMP [2], jeweils unter Verwendung der Codeworte mit Gewicht $w_{\mathrm{H}}(\boldsymbol{c}) < 128$ [30] ermittelt.

7.2. Für Mehrstufendecodierung optimierte Codes verschiedener Länge und Rate 1/2

Im vorigen Kapitel wurden die beiden in dieser Arbeit vorgestellten Konstruktionsverfahren anhand von Codes der Länge $N = 512$ und $N = 2048$ und der Rate $R = 0.5$ eingeführt. In diesem Abschnitt werden nun die mit Listendecodierung erzielbaren Fehlerraten bei Übertragung über den AWGN-Kanal, vor allem für größere Codelängen, miteinander verglichen.

Die vorgestellten Bitfehlerwahrscheinlichkeiten gehen in allen Fällen von einer systematischen Codiervorschrift aus[1]. Dies bringt bekanntermaßen einen zusätzlichen Vorteil im Verhältnis zwischen Wort- und Bitfehlerrate, da so bei (nahezu) ML-Decodierung in vielen Fällen bei einem Wortfehler nur $K \cdot D/N$ Bitfehler auftreten und somit die Bitfehlerrate bis zum Faktor D/N kleiner als die Wortfehlerrate sein kann. Da alle hier untersuchten Codes im Verhältnis zur Codelänge nur eine kleine Mindestdistanz besitzen, ergibt sich daraus gerade bei großen Codelängen eine wesentlich verbesserte BER als bei nichtsystematischer Codierung.

Codes der Länge 1024

Die Wort- bzw. Bitfehlerraten des OCBM bzw. LRAC(4) mit Parametern $(N, K) = (1024, 512)$ sind in den Bildern 7.5 und 7.6 gezeigt. Zum Vergleich ist für den LRAC auch eine unter Grenze für die WER des ML-Decoders (ML-LB) angegeben. Der Gewinn an SNR beim Übergang von $L = 1$ zu $L = 5$ beträgt bei OCBM, wie auch schon beim $(512, 256)$-OCBM (vergl. Bild 6.5) ca. 0.5 dB bei WER $= 10^{-4}$. Etwa der gleiche Gewinn ist beim Übergang vom OCBM zum LRAC zu erzielen. Die erreichten Wortfehlerraten mit Listengröße $L = 5$ für den OCBM bzw. $L = 20$ für den LRAC sind sehr nahe an denen des ML-Decoders, so daß eine weitere Vergrößerung der Liste nur geringen zusätzlichen Gewinn bringen kann.

Bei noch größerer zulässiger Decodierkomplexität empfiehlt sich daher eine Optimierung der Raten der äußeren Codes per Hand, wie es in Abschnitt 6.2 beispielhaft für $N = 512$ vorgestellt wurde.

[1] Es läßt sich leicht zeigen, daß wenn genau für die $i \in \{i_1, i_2, \ldots, i_K\}$ gilt $K^{(i)} = 1$, dann bilden die Positionen $k \in I_{\mathrm{S}} = \{N - i_1, N - i_2, \ldots, N - i_K\}$ des Codevektors $\boldsymbol{c} \in C$ eine Informationsmenge, d.h. die zugehörigen Spalten der Generatormatrix sind linear unabhängig und die Symbole c_k, $k \in I_{\mathrm{S}}$ können daher beliebig gewählt werden. Die Codierung kann durch eine einfache Soft-BMD-Decodierung des Vektors $\boldsymbol{y}$ mit

$$y_k = \begin{cases} c_k = \pm 1 & \text{für } K^{(N-k)} = 1 \\ 0 & \text{sonst} \end{cases}, \forall k$$

erfolgen.

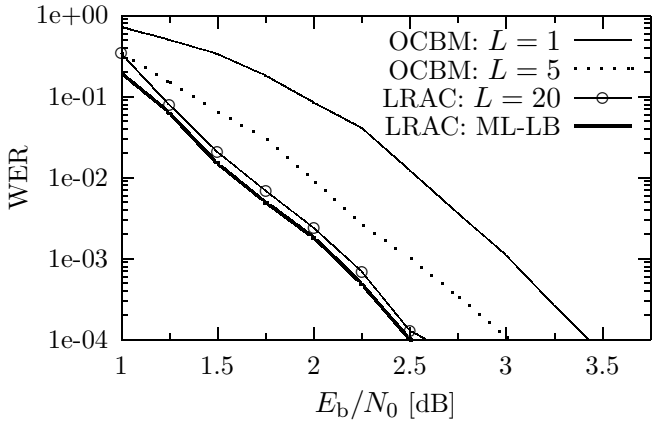

Abbildung 7.5.: Wortfehlerraten für Codes mit $(N, K) = (1024, 512)$ bei Übertragung über den AWGN-Kanal. Der LRAC wurde für $\rho = 4$ konstruiert.

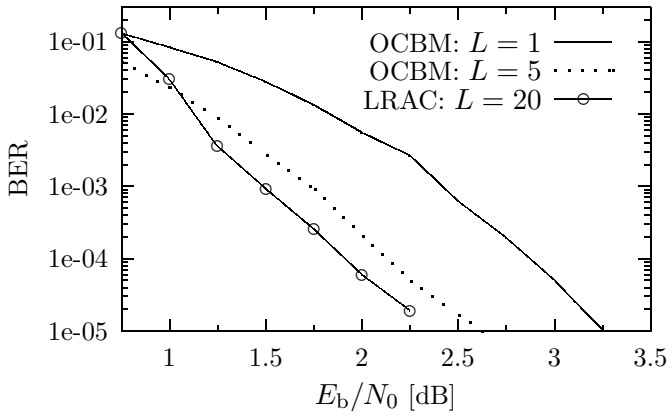

Abbildung 7.6.: Bitfehlerraten für Codes mit $(N, K) = (1024, 512)$ bei Übertragung über den AWGN-Kanal. Der LRAC wurde für $\rho = 4$ konstruiert.

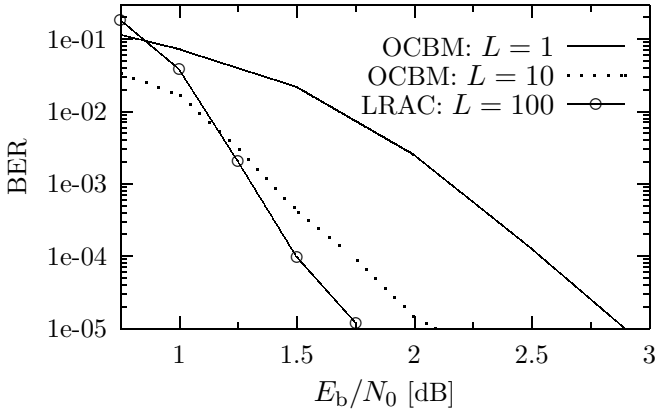

Abbildung 7.7.: Bitfehlerraten für Codes mit $(N, K) = (2048, 1024)$ bei Übertragung über den AWGN-Kanal. Der LRAC wurde für $\rho = 4$ konstruiert.

Codes der Länge 2048

Die Wortfehlerwahrscheinlichkeit bei Länge $N = 2048$ wurde schon in Bild 6.9 gezeigt. Zur Vollständigkeit wird in Bild 7.7 die Bitfehlerwahrscheinlichkeit bei der Decodierung der gleichen Codes gegeben, wobei hier für den LRAC(4) die Liste auf $L = 100$ erhöht wurde.

Es sollte angemerkt werden, daß beim bestem dem Autor für diese Codelänge bekannten Ergebnis für Turbo-Codes das notwendige SNR für WER $= 10^{-4}$ bzw. BER $= 10^{-5}$ jeweils nur um weniger als 0.3 dB geringer ist [6].

Codes der Länge 4096

Ist die erforderliche Listengröße beim LRAC für eine Wortfehlerwahrscheinlichkeit nahe der des ML-Decoders für $N = 512$ noch relativ klein, so ist dazu bei $N = 4096$ eine wesentlich größere Liste notwendig. Nach Bild 7.8 ist das notwendige SNR für WER $= 10^{-4}$ mit $L = 50$ ca. 0.25 dB größer als bei optimaler Decodierung. Hier wurde $\rho = 5$ gewählt, wodurch die zur Anpassung betrachteten 32 äußeren Codes die Länge $N^{(i)} = 128$ haben. Allgemein erweist es sich mit wachsender Codelänge als zunehmend schwieriger, das Korrekturpotential selbst des LRAC auszunutzen. Dies führt zwar für die hier betrachtete Codelänge von $N = 4096$ noch nicht zu einer wesentlich verschlechterten Wortfehlerwahrscheinlichkeit, aber da für eine kleine Bitfehlerwahrscheinlichkeit eine Wortfehlerrate nahe der des ML-Decoders erforderlich ist[2], sind die so erzielten Bitfehlerraten deutlich schlechter.

Ein Möglichkeit, um auch für diese Codelänge kleine *Bitfehlerraten* zu erzielen, besteht in der Wahl eines LRAC mit geringerem Korrekturpotential, bei dem auch mit kleiner Listengröße Wortfehlerraten nahe der des ML-Decoders erreichbar sind. Trotz einer geringfügig schlechteren WER kann somit unter Umständen ein Gewinn bezüglich der Bitfehlerrate erzielt werden. Nach dem in Abschnitt 6.2 vorgestellten Konstruktionsverfahren für den LRAC werden die

[2] Für eine kleine Bitfehlerrate ist neben einer systematischen Codierung bei dem hier betrachteten Decodierverfahren auch eine WER nahe der des ML-Decoders erforderlich. Dies läßt sich damit erklären, daß, falls bei der Listendecodierung das richtige bzw. das ML-Codewort aus der Liste fällt, die Decodierentscheidung $\widehat{c}$ unter Umständen weit weg vom empfangenen Vektor liegt und somit viele Bitfehler auftreten.

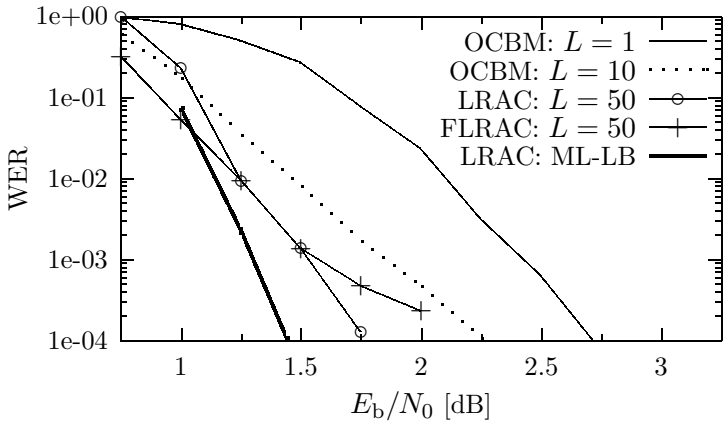

Abbildung 7.8.: Wortfehlerraten für Codes mit $(N, K) = (4096, 2048)$ bei Übertragung über den AWGN-Kanal. Der FLRAC wurde für $E_b/N_0 = 1.25$ dB konstruiert.

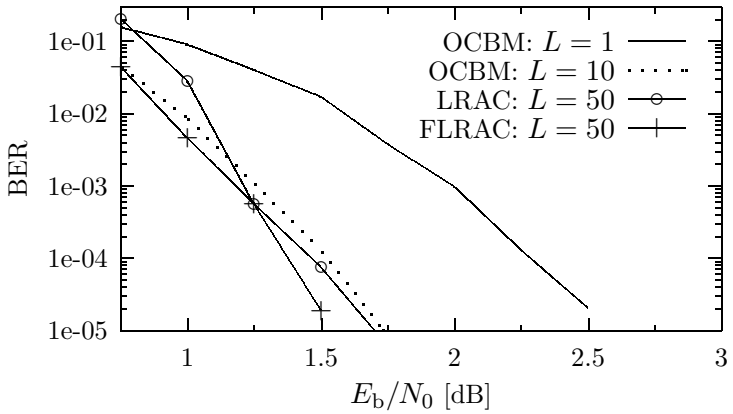

Abbildung 7.9.: Bitfehlerraten für Codes mit $(N, K) = (4096, 2048)$ bei Übertragung über den AWGN-Kanal. Der FLRAC wurde für $E_b/N_0 = 1.25$ dB konstruiert.

Raten der äußeren Codes an das jeweilige SNR des Kanals angepaßt. Für kleine SNR-Werte wächst das Korrekturpotential des so konstruierten Codes mit E_b/N_0. Ein für alle SNR-Werte an ein kleines, festes E_b/N_0 angepaßter LRAC besitzt daher die gewünschte Eigenschaft und ermöglicht u.U. eine kleine Bitfehlerrate. Ein solcher Code wird im folgenden als *fester linear Raten-angepaßter Code* (FLRAC) bezeichnet.

Wie auch schon bei der Ratenanpassung der äußeren Codes *per Hand* nach Abschnitt 6.2 kann hier keine optimale Regel für das richtige E_b/N_0 zur Konstruktion des FLRAC angegeben werden. Als ein geeigneter SNR-Wert für die Parameter $(N, K) = (4096, 2048)$ erscheint $E_b/N_0 = 1.25$ dB, da hier die Wortfehlerwahrscheinlichkeitskurve des LRAC von der des ML-Decoders divergiert. Der so gefundene FLRAC liefert in der Tat, trotz leicht größerer WER für $E_b/N_0 = 1.5$ dB, eine bessere BER als der LRAC (s. Bild 7.9).

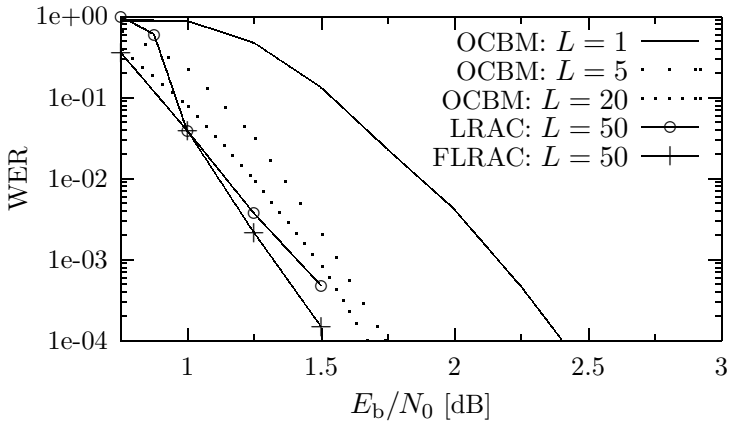

Abbildung 7.10.: Wortfehlerraten für Codes mit $(N, K) = (8192, 4096)$ bei Übertragung über den AWGN-Kanal. Der FLRAC wurde für $E_b/N_0 = 1.0\,\text{dB}$ konstruiert

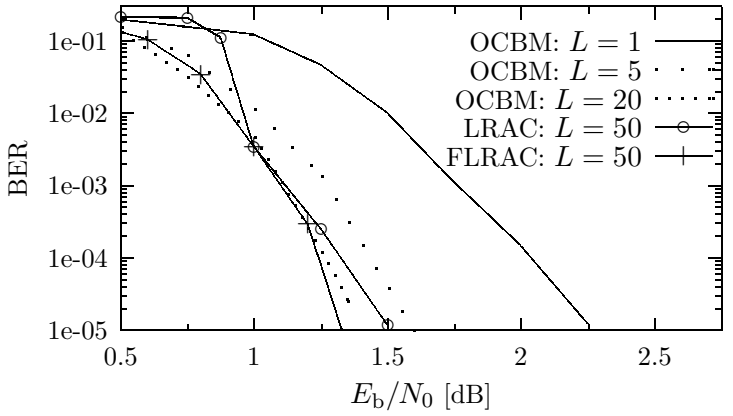

Abbildung 7.11.: Bitfehlerraten für Codes mit $(N, K) = (8192, 4096)$ bei Übertragung über den AWGN-Kanal. Der FLRAC wurde für $E_b/N_0 = 1.0\,\text{dB}$ konstruiert.

Codes der Länge 8192

Für Codes der Länge $N = 8192$ sind die Fehlerraten in den Bildern 7.10 bzw. 7.11 gezeigt. Das oben für Codes Länge $N = 4096$ beschriebene Problem, daß das Korrekturpotential selbst des LRAC($\rho = 6$) nicht ausreichend genutzt werden kann, führt hier bei $N = 8192$ nicht nur zu einer Verschlechterung der Bit-, sondern auch der Wortfehlerraten. Mit der gleichen Methode wie oben kann auch hier für den betrachteten Bereich eine Verbesserung erzielt werden. Mit dem für $E_b/N_0 = 1.0\,\text{dB}$ konstruierten FLRAC ist ein SNR von $E_b/N_0 \approx 1.6\,\text{dB}$ für WER $= 10^{-4}$ und $E_b/N_0 \approx 1.35\,\text{dB}$ für BER $= 10^{-5}$ notwendig.

Es fällt jedoch auf, daß der Unterschied zwischen LRAC und OCBM in bezug auf die mit vernünftigen Listengrößen erreichbaren Fehlerwahrscheinlichkeiten, der bei kurzen Codes sehr ausgeprägt ist, hier fast verschwunden ist. Für sehr große Codelängen ist damit nur noch der OCBM interessant.

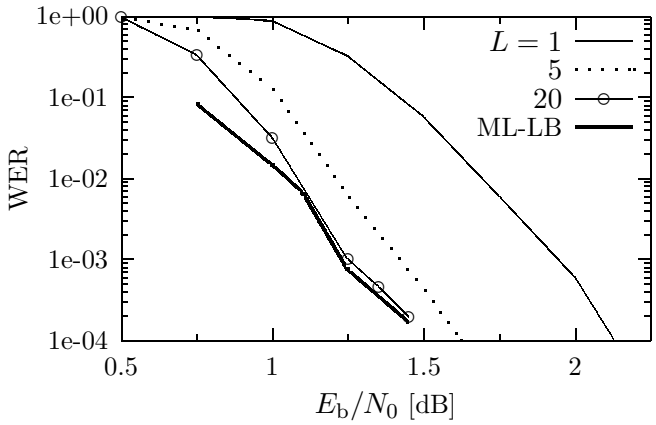

Abbildung 7.12.: Wortfehlerraten der OCBM mit $(N, K) = (16384, 8192)$ bei Übertragung über den AWGN-Kanal

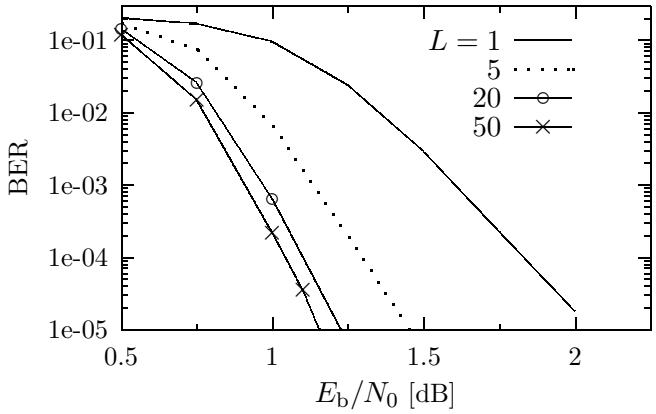

Abbildung 7.13.: Bitfehlerraten der OCBM mit $(N, K) = (16384, 8192)$ bei Übertragung über den AWGN-Kanal

Codes der Länge 16384

Für Codelänge $N = 2^{14}$ werden hier nur die erzielten Ergebnisse für den OCBM gezeigt, da die für kleine Fehlerraten erforderliche Listengröße im Fall des LRAC sehr groß werden. Beim OCBM ist für eine fast optimale Wortfehlerwahrscheinlichkeit eine Listengröße von ca. $L = 20$ ausreichend. Der 'Knick' in der Wortfehlerwahrscheinlichkeitskurve des ML-Decoders bei $E_b/N_0 \approx 1.2\,\text{dB}$ resultiert aus der unterschiedlichen Mindestdistanz des OCBM bei $E_b/N_0 = 1.1\,\text{dB}$ und $E_b/N_0 = 1.2\,\text{dB}$ von $d_{\text{H,min}} = 16$ bzw. $d_{\text{H,min}} = 32$. Der steile Abfall der WER-Kurve im Bereich um $E_b/N_0 \approx 1.1\,\text{dB}$ wird durch eine starke Verbesserung der Codeeigenschaften mit wachsendem SNR verursacht, die sich beim Überschreiten von $1.2\,\text{dB}$ nur noch wenig verändern.

Die Bitfehlerrate verbessert sich bei einer Vergrößerung der Liste von $L = 20$ auf $L = 50$ nur geringfügig. So ist für BER $= 10^{-5}$ nur ein SNR von $E_b/N_0 = 1.15\,\text{dB}$ erforderlich.

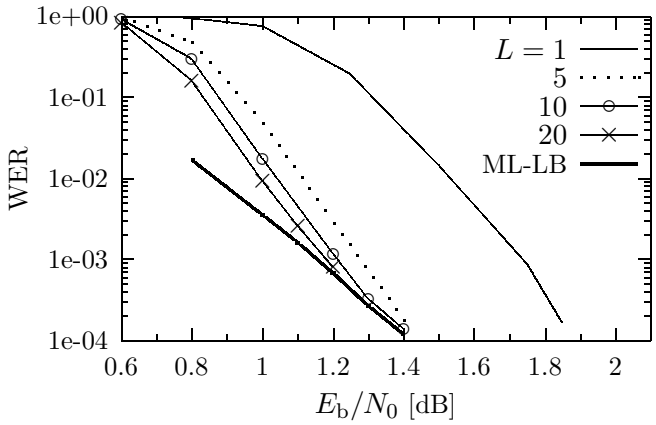

Abbildung 7.14.: Wortfehlerraten der OCBM mit $(N, K) = (32768, 16384)$ bei Übertragung über den AWGN-Kanal

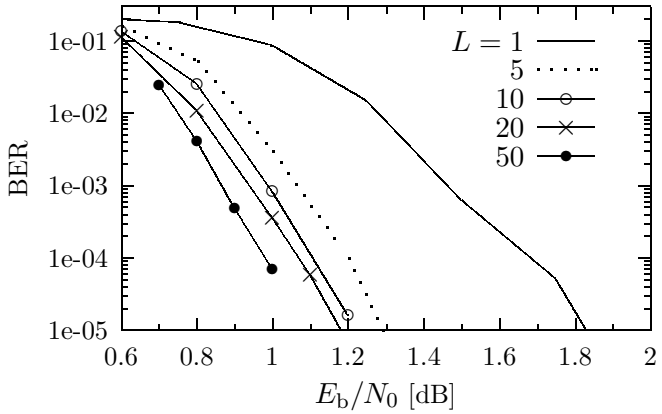

Abbildung 7.15.: Bitfehlerraten der OCBM mit $(N, K) = (32768, 16384)$ bei Übertragung über den AWGN-Kanal

Codes der Länge 32768

Auch für diese Codelänge werden nur die Ergebnisse des OCBM gezeigt. Durch den flachen Verlauf der WER-Kurve ist für Listengröße $L = 5$ das erforderliche SNR für WER $= 10^{-4}$ nur unwesentlich größer als bei ML-Decodierung. Eine größere Liste ist daher nur im Hinblick auf eine kleine Bitfehlerrate bei niedrigem SNR sinnvoll. Mit $L = 50$ kann hier bei $E_b/N_0 = 1.0\,\text{dB}$ eine BER $< 10^{-4}$ erreicht werden.

7.2.1. Vergleich der Ergebnisse mit SHANNONs Sphere-Packing-Bound

Im letzten Abschnitt dieses Kapitels werden nun die Fehlerraten, die mit den in dieser Arbeit vorgestellten Codekonstruktions- und Decodierverfahren erreicht wurden, mit der Sphere-Packing-Bound (engl.: Raum-Packungs-Grenze) von SHANNON [51] verglichen. Diese untere Grenze für die Wortfehlerwahrscheinlichkeit bei ML-Decodierung in Abhängigkeit von der Coderate gilt unter der Annahme, daß alle Codeworte die gleiche Gesamtenergie E_{ges} besitzen. Diese

Energie kann beliebig über das Codewort verteilt sein. Binäre Codes erfüllen diese Bedingung, für sie gilt aber zusätzlich die weitere Einschränkung, daß die empfangene Signalenergie für alle Symbole exakt gleich ist, denn es gilt $E_S = E_{ges}/N$. Daher liegen selbst perfekte Codes, die auf der HAMMING-Grenze liegen, nicht auf der von SHANNON hergeleiteten unteren Grenze. Die in Bild 7.16 gezeigte Sphere Bound[3] gilt für Codes der Rate $R = 0.5$. Zum Vergleich sind auch einige Ergebnisse für Turbo-Codes der Rate $R = 0.5$ gezeigt, wobei die Werte für $K = 300$ und $K = 1000$ aus [6] und die Werte für $K = 10200$ und $K = 16384$ aus [7] entnommen sind.

So verdeutlicht sich auch hier das schon im Verlauf der Arbeit mehrfach gezeigte Bild:

- Bei kurzen Codelängen $(N \leq 512)$ ist für eine kleine Fehlerwahrscheinlichkeit in erster Linie die Korrekturfähigkeit des Codes ausschlaggebend. Daher erzielen hier die RM-Codes die besten Ergebnisse.
- Bei mittleren Codelängen $(1024 \leq N \leq 8192)$ wird die erreichbare Fehlerwahrscheinlichkeit zunehmend auch von der Decodierbarkeit des Code beeinflußt. Hier liefern die nach Abschnitt 6.2 auf Codewortebene optimierten Codes die besten Resultate.
- Bei noch größeren Codelängen wird die Fehlerwahrscheinlichkeit fast ausschließlich von der Decodierbarkeit bestimmt. Daher sind hier die OCBM vorzuziehen.

Trotz der großen Verbesserung im Vergleich zur klassischen Konstruktion für maximale Mindestdistanz und zur Decodierung von $|u|u+v|$-Codes nach KABATYANSKY, die in dieser Arbeit erzielt wurden, können für Codelängen $N \geq 2000$ die mit den Turbo-Codes erzielten Fehlerraten nicht erreicht werden. Der wesentliche Vorteil gegenüber den Turbo-Codes liegt jedoch in der Flexibilität der rekursiven $|u|u+v|$-Konstruktion. So sind, wie schon in Bild 6.4 auf Seite 81 gezeigt, sowohl für den OCBM als auch den LRAC alle Coderaten $R = K/N$, $K \in \{1, 2, \ldots, N\}$, $N = 2^m$ einfach realisierbar, ohne die Codestruktur durch Punktierung oder ähnliche Maßnahmen zu verändern. Dagegen muß im Fall der Turbo-Codes für jede Rate ein eigener Code konstruiert werden.

Eine weitere Erkenntnis dieser Arbeit ist, daß auch mit nichtiterativen Decodierverfahren, wie z.B. der hier vorgestellten Listendecodierung, mit realisierbarem Aufwand Coderaten wesentlich über der Cutoff-Rate erreicht werden können.

[3] Die Werte wurden aus [7] entnommen.

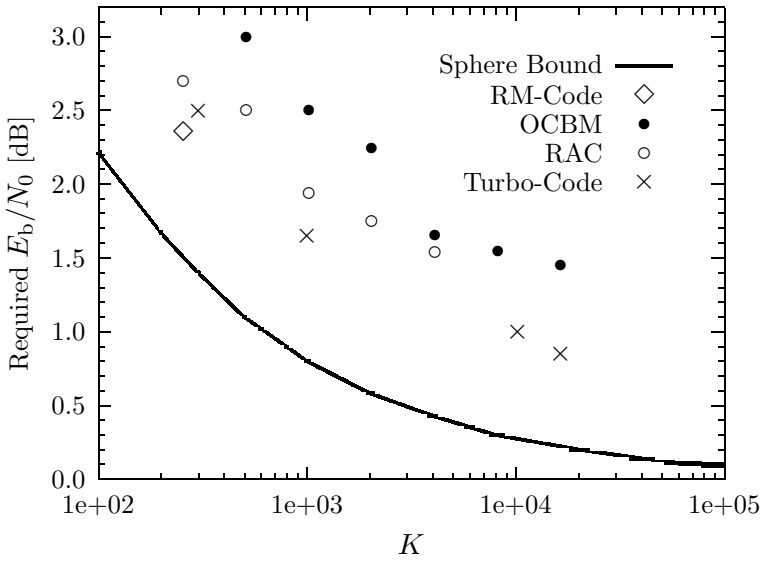

Abbildung 7.16.: Vergleich der in dieser Arbeit erzielten Ergebnisse mit der Sphere Packing Bound von SHANNON und einigen Ergebnissen für Turbo-Codes für WER $= 10^{-4}$ bei Übertragung über den AWGN-Kanal

8. Zusammenfassung

In dieser Arbeit wurde die Codeklasse der rekursiv konstruierten $|u|u+v|$-Codes untersucht. Der Schwerpunkt lag hierbei sowohl auf der Konstruktion als auch der Decodierung von Codes dieser Klasse.

Die rekursive Konstruktion, die auch mit Hilfe der verallgemeinerten Codeverkettung beschrieben werden kann, ermöglicht eine Zerlegung der $|u|u+v|$-Codes in innere und äußere Codes. Dies war Ausgangspunkt der vorgestellten Decodierverfahren, die mit der bei GC-Codes oft angewendeten Mehrstufendecodierung verwandt sind. So ist auch bei den in dieser Arbeit vorgestellten Verfahren für eine effektive Decodierung eine korrekte Weitergabe der Zuverlässigkeitswerte an die Decoder der äußeren Codes notwendig, und dies nicht nur bei Übertragung über den AWGN-Kanal. Durch informationstheoretische Betrachtung als auch durch Simulationsergebnisse wurde gezeigt, daß eine hier definierte, suboptimale Zuverlässigkeitsübergabe nahezu gleichwertig zur optimalen Übergabe ist.

Ausgehend von diesem Ergebnis wurde eine Methode zur Bestimmung des *äquivalenten SNR* vorgestellt, womit die Wortfehlerwahrscheinlichkeit bei *bitweiser Mehrstufendecodierung* numerisch sehr exakt geschätzt werden kann. Eine Erweiterung der bitweisen MSD ist die hier vorgestellte sequentielle- bzw. Listendecodierung. Ausgehend von den Zuverlässigkeitsübergaben konnten für beide Decodierverfahren sowohl optimale als auch suboptimale Metriken hergeleitet werden. Es hat sich gezeigt, daß unabhängig von der verwendeten Metrik das Listendecodierverfahren der sequentiellen Decodierung in bezug auf die erreichte Fehlerwahrscheinlichkeit überlegen ist. Können schon mit dem von LUCAS, BOSSERT und DAMMANN [33] gegebenen Decodieralgorithmus für REED-MULLER (RM) Codes bis zu einer Länge von $N = 64$ fast optimale Fehlerwahrscheinlichkeiten erreicht werden, so ermöglicht das hier vorgeschlagene Verfahren, auch bei allen RM-Codes bis zu einer Länge von $N = 128$ mit geringem Decodieraufwand Fehlerwahrscheinlichkeiten nahe denen des Maximum-Likelihood Decoders zu erzielen.

Allgemein erlaubt die regelmäßige Struktur der RM-Codes die Decodierung verschiedener Permutationen des Empfangsvektors. Es wurden für das verwendete Decodierverfahren geeignete Permutationen gefunden, so daß für Codelängen von 256 und 512 Codesymbolen beim AWGN-Kanal für eine Wortfehlerwahrscheinlichkeit von 10^{-4} nur ein um wenige zehntel Dezibel höheres Signal-zu-Rauschverhältnis (SNR) notwendig ist als bei optimaler Decodierung. Natürlich erfordert dies einen erhöhten Decodieraufwand.

Weiterhin wurden in dieser Arbeit zwei verschiedene Kriterien vorgestellt, um an die betrachteten Decodierverfahren angepaßte Codes zu konstruieren. So können auf sehr einfache Weise Codes jeder beliebigen Rate $R = K/N$, $K \in \{1, 2, \ldots, N\}$, $N = 2^m$ realisiert werden. Die so gefundenen Codes (es handelt sich in allen Fällen um Untercodes hochratiger RM-Codes) haben

zwar ein geringeres Korrekturpotential als andere RM-Codes gleicher Rate, die erzielten Wortfehlerwahrscheinlichkeiten sind aber bei großen Codelängen wesentlich besser. So werden bei Codelängen von $N = 8192$ und $N = 32768$ auch ohne Listendecodierung bessere Ergebnisse erzielt als mit verketteten Codes aus innerem Faltungs- und äußeren REED-SOLOMON Codes. Falls eine Listendecodierung vorgenommen wird, ist z.B. für den Code der Länge $N = 32768$ nur noch ein SNR von $E_\mathrm{b}/N_0 = 1.0\,\mathrm{dB}$ für eine Bitfehlerrate von weniger als 10^{-4} notwendig, und damit nur ein um ca. $0.4\,\mathrm{dB}$ größeres SNR als bei dem ersten für Turbo-Codes präsentierten Ergebnis von '93 [3]. Vor allem bei kürzeren Codelängen sind, wie im letzten Kapitel gezeigt, mit den Turbo-Codes vergleichbare Fehlerwahrscheinlichkeiten erreichbar.

Viel wichtiger als der exakte Vergleich mit den mit Turbo-Codes erzielten Ergebnissen (und dem Ringen um eine Verringerung des notwendigen SNR um einige zehntel dB) ist die Erkenntnis, daß generell auch mit **nichtiterativen Decodierverfahren** bei vertretbarer Komplexität wesentlich über der Cutoff-Rate liegende Coderaten erreichbar sind.

So kann abschließend die in der Einleitung erwähnte Bemerkung aus [4] korrigiert werden zu:

A Reed-Muller code was used to transmit the Mariner phototgraph of Mars in 1972. Today, a subcode of a Reed-Muller code would be preferred.

A. Anhang

A.1. Herleitung der ML-Entscheidungsregel in Abhängigkeit von h_k

In diesem Abschnitt wird hier eine weitere äquivalente Entscheidungsregel des ML-Decoders hergeleitet, diesmal in Abhängigkeit von den Wahrscheinlichkeitsdifferenzen $h_k = P(c_k = 1|y_k) - P(c_k = -1|y_k)$. Nach Abschnitt 2.1 entscheidet sich der ML-Decoder im Fall des gedächtnislosen Kanals für das Codewort $\boldsymbol{c} \in C$, daß folgendes Produkt maximiert

$$\boldsymbol{c}_{\mathrm{ML}} = \arg\max_{\boldsymbol{c}\in C} \prod_k P(y_k|c_k).$$

Für $P(c_k = 1) = P(c_k = -1) = 0.5$ kann bei der Entscheidung anstelle der $P(y_k|c_k)$ auch[1] $P(c_k|y_k)$ oder auch $2P(c_k|y_k) = P(c_k|y_k) + P(c_k|y_k)$ verwendet werden, und da mit $c_k \in \{\pm 1\}$ gilt

$$P(c_k|y_k) + P(c_k|y_k) = P(c_k|y_k) + 1 - P(-c_k|y_k) = 1 + c_k \cdot h_k,$$

lautet eine äquivalente ML-Entscheidungsregel

$$\boldsymbol{c}_{\mathrm{ML}} = \arg\max_{\boldsymbol{c}\in C} \prod_k (1 + c_k \cdot h_k).$$

A.2. Die Metrik nach FANO in Abhängigkeit von h_k

Die FANO-Metrik ist für einen Pfad mit der dazugehörigen Codesequenz $(c_1, c_2, \ldots, c_\tau)$ gegeben durch

$$\lambda(c_1, c_2, \ldots, c_\tau) = \sum_{k=1}^{\tau} \log \frac{P(y_k|c_k)}{P(y_k)} - R.$$

Für $c_k \in \{\pm 1\}$ und $P(c_k = 1) = P(c_k = -1) = 0.5$ gilt

$$\frac{P(y_k|c_k)}{P(y_k)} = 2 \cdot P(c_k) \frac{P(y_k|c_k)}{P(y_k)} = 2 \cdot P(c_k|y_k) = 1 + c_k \cdot h_k$$

[1] Hier sollte angemerkt werden, daß die Beziehung $P(\boldsymbol{y}|\boldsymbol{c}) = \prod_k P(y_k|c_k)$ nicht in gleicher Weise für das Produkt der $\prod_k P(c_k|y_k)$ gilt. Trotzdem kann die Wahrscheinlichkeit $P(y_k|c_k)$ aus der Wahrscheinlichkeit für ein Codesymbol $P(c_k|y_k)$ berechnet werden. Zudem kann, da $P(y_k|c_k)$ unabhängig von $P(c_k)$ ist, eine beliebige Wahrscheinlichkeitsverteilung der c_k angenommen werden. Hier wurde aufgrund der Problemstellung $P(c_k = 1) = P(c_k = -1) = 0.5$ gewählt.

und damit ist die FANO-Metrik äquivalent zu

$$\lambda(c_1, c_2, \ldots, c_\tau) = \sum_{k=1}^{\tau} \log(1 + c_k \cdot h_k) - R.$$

A.3. FANO-Typ-Metrik für sequentielle Decodierung der äußeren Codes

In diesem Abschnitt wird eine Metrik des Decoders hergeleitet, bei dem die Codeworte der äußeren Codes sequentiell geschätzt werden. Es wird der Fall einer n-stufigen rekursiven Zerlegung in äußere Codes der Länge $N^{(i)} = 1$, $1 \leq i \leq N$ betrachtet, die Gesamtcodelänge ist damit $N = 2^n$. Die sich ergebende Metrik gleicht auf den ersten Blick der FANO-Metrik, allerdings geht diese von einem Algorithmus aus, bei dem die Codesymbole der Reihe nach geschätzt werden und somit durch jede Hypothese zumindest ein Teil der Codesequenz festgelegt ist. Hier werden jedoch die Codeworte der *äußeren* Codes sequentiell geschätzt, und da jedes einzelne Codesymbol des verketteten Codes c_k erst dann festgelegt ist, wenn auch der letzte äußere Code $C^{(N)}$ entschieden ist, kann die FANO-Metrik hier nicht direkt angewendet werden.

Angenommen wird ein Algorithmus, der die Codeworte $\boldsymbol{c}^{(i)} = (c^{(i)})$ der äußeren Codes $C^{(i)}$ der Reihe nach schätzt, indem in jedem Decodierschritt immer nur der Pfad mit der besten Metrik um eine Schätzung $\boldsymbol{c}^{(i)} \in C^{(i)}$ verlängert wird. Da hier eine Zerlegung des Codes in äußere Codes der Länge $N^{(i)} = 1, \forall i$ betrachtet wird, muß zwischen den beiden Möglichkeiten $C^{(i)} = \{(+1), (-1)\}$ für $K^{(i)} = 1$ bzw. $C^{(i)} = \{(+1)\}$ oder $C^{(i)} = \{(-1)\}$ für $K^{(i)} = 0$ unterschieden werden. Im Codebaum der Länge N gibt es damit genau dann eine Verzweigung an der Stelle $i \in \{1, 2, \ldots, N\}$, falls für den entsprechenden Code gilt $K^{(i)} = 1$. Ein Pfad der Länge τ durch den Codebaum ist durch die Folge der Codesymbole $\boldsymbol{p} = (c^{(1)}, c^{(2)}, \ldots, c^{(\tau)})$ festgelegt.

Die hier vorgestellte Herleitung geht von einem ähnlichen Modell aus, wie es in [35] von MASSEY angegeben wurde. Das Problem, aus der Menge der Hypothesen $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M\}$, $\boldsymbol{p}_m = (c_m^{(1)}, c_m^{(2)}, \ldots, c_m^{(\tau_m)})$ mit unterschiedlicher Länge τ_m, $1 \leq m \leq M$ die Hypothese mit größter Wahrscheinlichkeit zu finden, wird mit einem Übertragungssystem verglichen, bei dem ein in gleicher Weise rekursiv konstruierter Code mit M Codeworten verwendet wird. Das m-te Codewort dieses Codes wird gebildet, indem die ersten τ_m äußeren Codeworte entsprechend der m-ten Hypothese zu $\boldsymbol{c}^{(i)} = (c_m^{(i)})$, $i \in \{1, \ldots, \tau_m\}$ gewählt werden, die restlichen äußeren Codeworte $\boldsymbol{c}^{(i)}$, $i \in \{\tau_m + 1, \ldots, N\}$ werden zufällig und statistisch unabhängig voneinander mit $P(c^{(i)} = +1) = P(c^{(i)} = -1) = 0.5$ gewählt (siehe Bild A.1). Bei der Decodierung sind neben dem Empfangsvektor $\boldsymbol{y}$ nur die $\boldsymbol{p}_m$ bekannt, der Empfänger kennt von den $\boldsymbol{c}_m^{(i)}$, $i > \tau_m$ nur die Statistik, nach der sie ausgewählt werden, nicht aber die konkrete Realisierung.

Die optimale Decodierregel lautet nun

$$\widehat{m} = \arg\max_m P(m|\boldsymbol{y}) = \arg\max_m P(m, \boldsymbol{y}),$$

und damit hängt die Entscheidungsregel nur von $P(m, \boldsymbol{y})$ ab. Weiter kann geschrieben werden

$$P(m, \boldsymbol{y}) = P(c_m^{(1)}, \ldots, c_m^{(\tau)}, \boldsymbol{y}) = \sum_{\substack{c_m^{(i)} \in \{\pm 1\} \\ i > \tau_m}} P(c_m^{(1)}, \ldots, c_m^{(\tau_m)}, c_m^{(\tau_m+1)}, \ldots, c_m^{(N)}, \boldsymbol{y}).$$

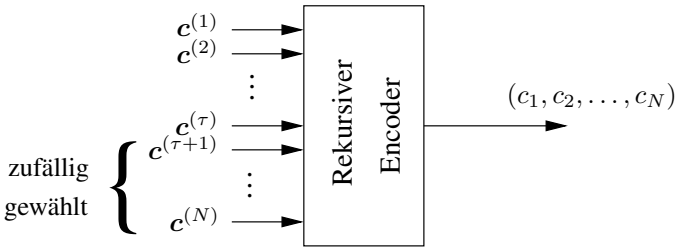

Abbildung A.1.: Bildung des m-ten Codewortes

Da die Codesymbole $c_m^{(i)}$, $i \in \{\tau_m + 1, \ldots, N\}$ unabhängig von den $c_m^{(i)}$, $i \in \{1, \ldots, \tau_m\}$ gewählt werden, gilt

$$P(m, \boldsymbol{y}) = \sum_{\substack{c_m^{(i)} \\ i>\tau_m}} P(\boldsymbol{y}|c_m^{(1)}, \ldots, c_m^{(N)}) \cdot P(c_m^{(1)}, \ldots, c_m^{(\tau_m)}) \cdot P(c_m^{(\tau_m+1)}, \ldots, c_m^{(N)}) \qquad \text{(A.1)}$$

und da der Kanal gedächtnislos ist, kann weiter geschrieben werden

$$P(m, \boldsymbol{y}) = \sum_{\substack{c_m^{(i)} \\ i>\tau_m}} \prod_k P(y_k|c_k) \cdot P(c_m^{(1)}, \ldots, c_m^{(\tau_m)}) \cdot P(c_m^{(\tau_m+1)}, \ldots, c_m^{(N)}),$$

wobei hier die Codesymbole c_k, $k \in \{1, \ldots, N\}$ durch die äußeren Codes $\boldsymbol{c}_m^{(i)}$, $i \in \{1, \ldots, N\}$ gemäß der rekursiven Codiervorschrift bestimmt sind. Mit $P(c_k = 1) = P(c_k = -1) = 0.5$ gilt wie in Abschnitt A.1 gezeigt (siehe auch Fußnote 1 auf Seite 107)

$$P(y_k|c_k) = 2 \cdot P(y_k) \cdot P(c_k|y_k) = P(y_k) \cdot (1 + c_k h_k)$$

und damit

$$P(m, \boldsymbol{y}) = \underbrace{\prod_k P(y_k)}_{\text{const}} \cdot \sum_{\substack{c_m^{(i)} \\ i>\tau_m}} \prod_k (1 + c_k h_k) \cdot P(c_m^{(1)}, \ldots, c_m^{(\tau_m)}) \cdot P(c_m^{(\tau_m+1)}, \ldots, c_m^{(N)}).$$

Aus Gl. (5.15) folgt, daß bei einer rekursiven Zerlegung in äußere Codes der Länge $N^{(i)} = 1$ das Produkt $\prod_k (1 + c_k h_k)$ ebenfalls rekursiv aus den Zuverlässigkeiten für die äußeren Codesymbole $h_1^{(i)}$ berechnet werden kann, und damit ergibt sich unter der Annahme, daß auch die $c_m^{(i)}$ für $i \in \{1, \ldots, \tau_m\}$ voneinander statistisch unabhängig sind

$$\begin{aligned} P(m, \boldsymbol{y}) &= \text{const} \cdot \sum_{\substack{c_m^{(j)} \\ j>\tau_m}} \prod_i (1 + c_m^{(i)} h_1^{(i)}) \cdot P(c_m^{(i)}) \\ &= \text{const} \cdot \prod_{i \le \tau_m} (1 + c_m^{(i)} h_1^{(i)}) \cdot P(c_m^{(i)}) \cdot \sum_{\substack{c_m^{(j)} \\ j>\tau_m}} \prod_{i>\tau_m} (1 + c_m^{(i)} h_1^{(i)}) \cdot P(c_m^{(i)}). \end{aligned}$$

und wenn bei der hinteren Summe die Summation mit der Produktbildung vertauscht wird

$$P(m, \boldsymbol{y}) = \text{const} \cdot \prod_{i \le \tau_m} (1 + c_m^{(i)} h_1^{(i)}) \cdot P(c_m^{(i)}) \cdot \prod_{i>\tau_m} \sum_{c_m^{(i)} \in \{\pm 1\}} (1 + c_m^{(i)} h_1^{(i)}) \cdot P(c_m^{(i)}).$$

Z.B. für den Faktor des hinteren Produktes mit $i = \tau_m + 1$ gilt

$$\sum_{c_m^{(\tau_m+1)} \in \{\pm 1\}} (1 + c_m^{(\tau_m+1)} h_1^{(\tau_m+1)}) \cdot P(c_m^{(\tau_m+1)}) = \frac{1}{2}(1 + h_1^{(\tau_m+1)}) + \frac{1}{2}(1 - h_1^{(\tau_m+1)}) = 1$$

so daß geschrieben werden kann

$$P(m, \boldsymbol{y}) = \text{const} \cdot \prod_{i \le \tau_m} (1 + c_m^{(i)} h_1^{(i)}) \cdot P(c_m^{(i)}). \tag{A.2}$$

Unter der Bedingung, daß an den Positionen i mit $K^{(i)} = 1$ die $c_m^{(i)} \in \{\pm 1\}$ gleichwahrscheinlich sind, d.h. $P(c_m^{(i)}) = 2^{-K^{(i)}}$, ergibt sich durch Übergang in den Log-Bereich und Weglassen der Konstante schließlich die additive Metrik

$$\lambda_h(\boldsymbol{c}^{(1)}, \boldsymbol{c}^{(2)}, \ldots, \boldsymbol{c}^{(\tau)}) = \sum_{i=1}^{\tau} \log_2(1 + c_1^{(i)} \cdot h_1^{(i)}) - K^{(i)}, \tag{A.3}$$

die sowohl für den AWGN-Kanal als auch bei Übertragung über den BSC angewendet werden kann.

A.4. Distanzbasierte Metrik für sequentielle Decodierung der äußeren Codes

In diesem Abschnitt wird eine weitere Metrik für einen Algorithmus hergeleitet, bei dem genau wie im Abschnitt A.3 die Codeworte der äußeren Codes sequentiell geschätzt werden. Wie dort gezeigt wurde, ist die wahrscheinlichste Hypothese bei rekursiver Zerlegung in äußere Codes der Länge $N^{(i)} = 1$ gegeben durch (s. Gl. (A.1))

$$\widehat{m} = \arg\max_m \sum_{\substack{c_m^{(i)} \\ i > \tau_m}} P(\boldsymbol{y} | c_m^{(1)}, \ldots, c_m^{(N)}) \cdot P(c_m^{(1)}, \ldots, c_m^{(\tau_m)}) \cdot P(c_m^{(\tau_m+1)}, \ldots, c_m^{(N)}).$$

Anstatt nun in obiger Gleichung für jede Hypothese m über alle unbekannten Codesymbole $c_m^{(i)}$, $i > \tau_m$ zu summieren, kann vereinfacht folgende suboptimale Regel betrachtet werden

$$\widehat{m}_{\text{sub}} = \arg\max_m \max_{\substack{c_m^{(i)} \\ i > \tau_m}} P(\boldsymbol{y} | c_m^{(1)}, \ldots, c_m^{(N)}) \cdot P(c_m^{(1)}, \ldots, c_m^{(\tau_m)}) \cdot P(c_m^{(\tau_m+1)}, \ldots, c_m^{(N)}), \tag{A.4}$$

d.h. für jedes m wird das wahrscheinlichste Codewort über alle $c_m^{(i)}$, $i > \tau_m$ bestimmt und dann das wahrscheinlichste m ausgewählt.

A.4.1. Metrik für den AWGN-Kanal

Für den AWGN-Kanal ist obige Gleichung äquivalent zu

$$\widehat{m}_{\text{sub}} = \arg\max_m \max_{\substack{c_m^{(i)} \\ i > \tau_m}} \prod_k \exp\left[-\frac{(y_k - c_k)^2}{2\sigma_0^2}\right] \cdot \underbrace{P(c_m^{(1)}, \ldots, c_m^{(\tau_m)})}_{\prod_{i=1}^{\tau_m} 2^{-K(i)}} \cdot \underbrace{P(c_m^{(\tau_m+1)}, \ldots, c_m^{(N)})}_{2^{-(N-\tau_m)}},$$

wobei auch hier die Codesymbole c_k durch die äußeren Codeworte bestimmt sind. Bildung des Logarithmus (zur Basis 2) ergibt mit der Zerlegung

$$\exp\left[-\frac{(y_k - c_k)^2}{2\sigma_0^2}\right] = \exp\left[-\frac{y_k^2 + c_k^2}{2\sigma_0^2}\right] \cdot \exp\left[\frac{2y_k c_k}{2\sigma_0^2}\right] = \text{const} \cdot \exp\left[\frac{y_k c_k}{\sigma_0^2}\right]$$

und Weglassen der Konstanten

$$\widehat{m}_{\text{sub}} = \arg\max_m \max_{\substack{c_m^{(i)} \\ i>\tau_m}} \frac{1}{\ln 2 \cdot \sigma_0^2} \sum_k y_k \cdot c_k - \left[\sum_{i=1}^{\tau_m} K^{(i)}\right] + \tau_m. \tag{A.5}$$

Die vordere Summe über die Produkte $y_k c_k$ kann für $c_k \in \{\pm 1\}$ mit $w_k = |y_k|$ und $a_k = \operatorname{sign} y_k$ wie folgt geschrieben werden

$$\sum_k y_k \cdot c_k = \sum_k w_k - 2 \sum_{\{k: c_k \neq a_k\}} w_k,$$

womit die erste Summe der rechten Seite unabhängig von m ist. Die zweite Summe kann gemäß Gl. (3.13) umgeformt werden zu

$$\sum_{\{k: c_k \neq a_k\}} w_k = \sum_{\substack{\{i: c_m^{(i)} \neq a_1^{(i)}\} \\ i \leq \tau_m}} w_1^{(i)} + \sum_{\substack{\{i: c_m^{(i)} \neq a_1^{(i)}\} \\ i > \tau_m}} w_1^{(i)}$$

Wenn berücksichtigt wird, daß beim Maximum in Gl. (A.5) die zweite Summe der rechten Seite in obiger Gleichung Null ist, d.h. wenn $c_m^{(i)} = a_1^{(i)}$ für $i > \tau_m$ gewählt wird, ergibt sich eine erste suboptimale additive Metrik für den AWGN-Kanal

$$\lambda_{\text{sub}}(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(\tau)}) = \frac{-2}{\ln 2 \cdot \sigma_0^2} \sum_{\substack{\{i: c_1^{(i)} \neq a_1^{(i)}\} \\ i \leq \tau}} w_1^{(i)} + \left[\sum_{i=1}^{\tau} 1 - K^{(i)}\right]. \tag{A.6}$$

Eine genaue Betrachtung dieser Metrik zeigt, daß sie in dieser Form nicht für den AWGN-Kanal mit $y_k \in \mathbb{R}$ geeignet ist. Eine charakteristische Eigenschaft jeder Metrik für den AWGN-Kanal sollte sein, daß sie für $y_k \approx 0$ und $K^{(i)} > 0$ 'fällt', d.h. kleiner wird[2], da y_k keine Information über das gesendete Codesymbol c_k liefert und damit die 'Unsicherheit' beim Verlängern jedes Pfades steigt. Dies führt bei Verwendung einer geeigneten Metrik zu dem für sequentielle Decodierung typischen Verhalten, daß bei Bündelauslöschungen an der entsprechenden Stelle des Codebaumes viele verschiedene Pfade verlängert werden, da kein Pfad gegenüber dem anderen bevorzugt werden kann. Für $y_k \approx 0$ gilt auch $w_k \approx 0$. Die obige Metrik (A.6) wächst für $w_1^{(\tau)} = 0$ um den Term $1 - K^{(\tau)} \geq 0$ an, und damit wird die Metrik des besten Pfades nicht für $w_1^{(\tau)} = 0$ wie gefordert kleiner, so daß auch im nächsten Schritt nur dieser Pfad verlängert wird. Dies würde bei Bündelauslöschungen dazu führen, daß nur ein einziger Pfad verlängert wird. Im folgenden wird eine Möglichkeit gezeigt, die Metrik (A.6) zu verbessern.

[2]Hier wird vorausgesetzt, das immer der Pfad mit der *größten* Metrik verlängert wird.

Vergleich mit der Metrik aus Anhang A.3

Es ist sinnvoll, die bei sequentieller Decodierung der Codesymbole c_k zu Gl. (A.6) äquivalente Metrik mit der FANO-Metrik zu vergleichen. Gleichung (A.4) lautet bei herkömmlicher Decodierung, d.h. wenn die Codesymbolfolge c_k sequentiell geschätzt wird

$$\widehat{m} = \arg\max_{m} \max_{\substack{c_k \\ k>\tau_m}} P(\boldsymbol{y}|c_1,\ldots,c_N) \cdot P(c_1,\ldots,c_{\tau_m}) \cdot P(c_{\tau_m+1},\ldots,c_N).$$

Mit den gleichen Überlegungen wie oben ergibt sich daraus eine suboptimale Metrik für den AWGN-Kanal

$$\lambda_{\text{sub}}(c_1,\ldots,c_\tau) = \frac{-2}{\ln 2 \cdot \sigma_0^2} \cdot \sum_{\substack{\{k:c_k \neq a_k\} \\ k\leq\tau}} w_k + \left[\sum_{k=1}^{\tau} 1 - R\right] \tag{A.7}$$

mit der Coderate R, die in dieser Form genau aus den gleichen Gründen wie oben beschrieben nicht für den AWGN-Kanal geeignet ist. Die FANO-Metrik nach (5.14) kann für den AWGN-Kanal geschrieben werden als

$$\begin{aligned}\lambda(c_1,\ldots,c_\tau) &= \sum_{k=1}^{\tau} \log_2\left(2\frac{\mathrm{e}^{y_k c_k/\sigma_0^2}}{\mathrm{e}^{y_k c_k/\sigma_0^2}+\mathrm{e}^{-y_k c_k/\sigma_0^2}}\right) - R \\ &= \sum_{k=1}^{\tau} \log_2\left(\frac{\mathrm{e}^{y_k c_k/\sigma_0^2}}{\mathrm{e}^{y_k c_k/\sigma_0^2}+\mathrm{e}^{-y_k c_k/\sigma_0^2}}\right) + 1 - R.\end{aligned}$$

Für $\sigma_0^2 \ll w_k$ gilt näherungsweise

$$\log_2\left(\frac{\mathrm{e}^{y_k c_k/\sigma_0^2}}{\mathrm{e}^{y_k c_k/\sigma_0^2}+\mathrm{e}^{-y_k c_k/\sigma_0^2}}\right) \approx \begin{cases} 0 & \text{für } c_k = a_k \\ \frac{-2w_k}{\ln 2\cdot\sigma_0^2} & \text{für } c_k \neq a_k \end{cases}$$

und damit ist (A.7) eine Näherung für die FANO-Metrik für $\sigma_0^2 \ll w_k$. Daher kann die distanzbasierte Metrik (A.6) ebenfalls als Näherung der auf Wahrscheinlichkeiten basierten Metrik (A.3) für $\sigma_0^2 \ll w_k$ angesehen werden.

Für $w_k \approx 0$ ergibt sich jedoch mit der Näherung $\log_2(1+\mathrm{e}^x) \approx 1 + x/(2\ln 2)$

$$\log_2\left(\frac{\mathrm{e}^{y_k c_k/\sigma_0^2}}{\mathrm{e}^{y_k c_k/\sigma_0^2}+\mathrm{e}^{-y_k c_k/\sigma_0^2}}\right) = \underbrace{\log_2 1}_{=0} - \log_2(1+\mathrm{e}^{-2y_k c_k/\sigma_0^2}) \approx -1 + \frac{y_k c_k}{\ln 2 \cdot \sigma_0^2},$$

woraus sich eine auch für den AWGN-Kanal geeignete Näherung der FANO-Metrik ergibt

$$\lambda_y(c_1,\ldots,c_\tau) = \frac{-2}{\ln 2 \cdot \sigma_0^2} \cdot \sum_{\substack{\{k:c_k \neq a_k\} \\ k\leq\tau}} w_k + \left[\sum_{k=1}^{\tau} \min\left(1, \frac{w_k}{\ln 2 \cdot \sigma_0^2}\right) - R\right].$$

Heuristisch kann daher für die sequentielle Decodierung der äußeren Codes folgende Metrik vorgeschlagen werden

$$\lambda_y^{(\text{AWGN})}(\boldsymbol{c}^{(1)},\ldots,\boldsymbol{c}^{(\tau)}) = \frac{-2}{\ln 2 \cdot \sigma_0^2} \sum_{\substack{\{i:c_1^{(i)} \neq a_1^{(i)}\} \\ i\leq\tau}} w_1^{(i)} + \left[\sum_{i=1}^{\tau} \min\left(1, \frac{w_1^{(i)}}{\ln 2 \cdot \sigma_0^2}\right) - K^{(i)}\right], \tag{A.8}$$

die für den Fall $y_1^{(i)} \approx 0$ und $K^{(i)} > 0$ im Vergleich zu (A.6) die geforderte Eigenschaft hat, daß sie kleiner wird.

A.4.2. Metrik für den BSC

Bei Übertragung über den BSC mit $y_k \in \{\pm 1\}$ folgt aus Gl. (A.4) unter Verwendung von (s. Abschnitt 2.1)

$$P(\boldsymbol{y}|c_m^{(1)},\ldots,c_m^{(N)}) = p^e q^{N-e} = p^{\sum_N (y_k - c_k)^2/4} \cdot q^{N - \sum_N (y_k - c_k)^2/4},$$

wobei e die Anzahl der Übertragungsfehler $y_k \neq c_k$ bezeichnet, und anschließender Logarithmierung

$$\begin{aligned}
\widehat{m}_{\text{sub}} &= \arg\max_m \max_{\substack{c_m^{(i)} \\ i>\tau_m}} \frac{\log_2 q - \log_2 p}{2} \sum_k y_k \cdot c_k - \left[\sum_{i=1}^{\tau_m} K^{(i)}\right] + \tau_m \\
&= \arg\max_m \max_{\substack{c_m^{(i)} \\ i>\tau_m}} (\log_2 p - \log_2 q) \sum_{\{k:c_k \neq a_k\}} w_k - \left[\sum_{i=1}^{\tau_m} K^{(i)}\right] + \tau_m.
\end{aligned}$$

Da auch hier der Term $\sum w_k$ durch die an die äußeren Decoder übergebenen Zuverlässigkeiten $y_k^{(i)}$ bestimmt werden kann, lautet die suboptimale Metrik für den BSC schließlich

$$\lambda_y^{(\text{BSC})}(\boldsymbol{c}^{(1)},\ldots,\boldsymbol{c}^{(\tau)}) = (\log_2 p - \log_2 q) \cdot \sum_{\substack{\{i:c_1^{(i)} \neq a_1^{(i)}\} \\ i \leq \tau}} w_1^{(i)} + \left[\sum_{i=1}^{\tau} 1 - K^{(i)}\right]. \tag{A.9}$$

A.5. Wahrscheinlichkeitsdichten für $y_k^{(v)}$ beim AWGN-Kanal

In diesem Abschnitt werden die Wahrscheinlichkeitsdichten $f(y_k^{(v)}|v_k)$ bei Übertragung über den AWGN-Kanal bestimmt. Im ersten Schritt werden dazu die Wahrscheinlichkeiten $P(y_k^{(v)} \geq y|v_k = 1)$ für $y \geq 0$ bzw. $P(y_k^{(v)} \leq y|v_k = 1)$ für $y \leq 0$ bestimmt, die mit $\boldsymbol{c}_k = (c_k, c_{N+k})$ gegeben sind durch

$$\begin{aligned}
P(y_k^{(v)} \gtrless y|v_k = 1) &= \frac{1}{2} P(y_k^{(v)} \gtrless y|\boldsymbol{c}_k = (1,1)) + \frac{1}{2} P(y_k^{(v)} \gtrless y|\boldsymbol{c}_k = (-1,-1)) \\
&= P(y_k^{(v)} \gtrless y|\boldsymbol{c}_k = (1,1)).
\end{aligned}$$

Die zweite Gleichung folgt aus der Symmetrie des Problems. In Bild A.2 sind die Gebiete für $\boldsymbol{y}_k = (y_k, y_{N+k})$ gezeigt, die mit

$$y_k^{(v)} = \operatorname{sign} y_k \cdot \operatorname{sign} y_{N+k} \cdot \min(|y_k|, |y_{N+k}|)$$

zu $y_k^{(v)} \geq y \geq 0$ führen. Mit

$$\begin{aligned}
A_y^{(1)} &= \{\boldsymbol{y}_k : y_k \geq y \wedge y_{N+k} \geq y\} \\
A_y^{(3)} &= \{\boldsymbol{y}_k : y_k \leq -y \wedge y_{N+k} \leq -y\}
\end{aligned}$$

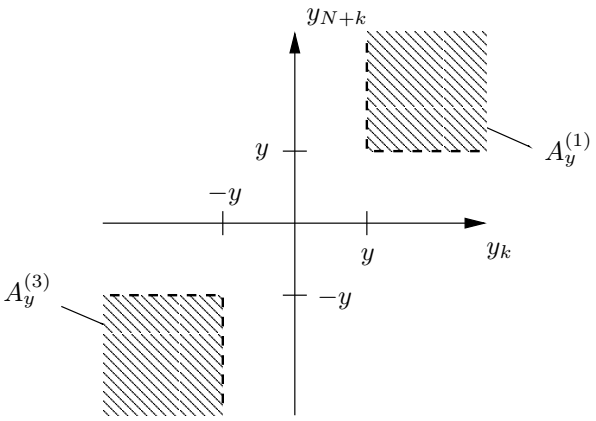

Abbildung A.2.: Gebiete $A_y^{(1)}, A_y^{(3)}$ für die Bestimmung von $P(y_k^{(v)} > y|v_k = 1)$ für $y \geq 0$

gilt

$$P(y_k^{(v)} \geq y|v_k = 1) = P(\boldsymbol{y}_k \in A_y^{(1)}|\boldsymbol{c}_k = (1,1)) + P(\boldsymbol{y}_k \in A_y^{(3)}|\boldsymbol{c}_k = (1,1)).$$

Da y_k und y_{N+k} statistisch unabhängig und gaußverteilt sind, kann man weiter schreiben

$$\begin{aligned} &P(y_k^{(v)} \geq y|v_k = 1) = \\ &\quad \int_y^\infty \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left[-\frac{(y_k-1)^2}{2\sigma_0^2}\right] \mathrm{d}y_k \cdot \int_y^\infty \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left[-\frac{(y_{N+k}-1)^2}{2\sigma_0^2}\right] \mathrm{d}y_{N+k} \\ &\quad + \int_{-\infty}^{-y} \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left[-\frac{(y_k-1)^2}{2\sigma_0^2}\right] \mathrm{d}y_k \cdot \int_{-\infty}^{-y} \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left[-\frac{(y_{N+k}-1)^2}{2\sigma_0^2}\right] \mathrm{d}y_{N+k} \end{aligned}$$

Anwendung der Produktregel und der Symmetrieeigenschaft

$$\int_{-\infty}^{-y} \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left[-\frac{(y_k-1)^2}{2\sigma_0^2}\right] \mathrm{d}y_k = \int_y^\infty \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left[-\frac{(y_k+1)^2}{2\sigma_0^2}\right] \mathrm{d}y_k$$

ergibt für $y \geq 0$

$$\begin{aligned} &f_{y_k^{(v)}}(y|v_k = 1) \\ &= -\frac{\mathrm{d}P(y_k^{(v)} \geq y|v_k = 1)}{\mathrm{d}y} \\ &= \frac{2}{\sqrt{2\pi\sigma_0^2}} \left(\exp\left[-\frac{(y-1)^2}{2\sigma_0^2}\right] \cdot Q(\frac{y-1}{\sigma_0}) + \exp\left[-\frac{(y+1)^2}{2\sigma_0^2}\right] \cdot Q\left(\frac{y+1}{\sigma_0}\right)\right) \end{aligned}$$

mit

$$Q(y) = \int_y^\infty \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{\beta^2}{2}\right] \mathrm{d}\beta.$$

Die gleichen Überlegungen führen für $y \leq 0$ ausgehen d von $P(y_k^{(v)} \leq y|v_k = 1)$ zu

$$\begin{aligned} &f_{y_k^{(v)}}(y|v_k = 1) = \\ &\quad \frac{2}{\sqrt{2\pi\sigma_0^2}} \left(\exp\left[-\frac{(y-1)^2}{2\sigma_0^2}\right] \cdot Q\left(\frac{-y-1}{\sigma_0}\right) + \exp\left[-\frac{(y+1)^2}{2\sigma_0^2}\right] \cdot Q\left(\frac{-y+1}{\sigma_0}\right)\right) \end{aligned}$$

so daß man allgemein für alle y schreiben kann

$$f_{y_k^{(v)}}(y|v_k = 1) = \frac{2}{\sqrt{2\pi\sigma_0^2}} \left(\exp\left[-\frac{(y-1)^2}{2\sigma_0^2} \right] \cdot Q\left(\frac{|y|-1}{\sigma_0} \right) + \exp\left[-\frac{(y+1)^2}{2\sigma_0^2} \right] \cdot Q\left(\frac{|y|+1}{\sigma_0} \right) \right).$$

Aufgrund der Symmetrie $f_{y_k^{(v)}}(y|v_k = 1) = f_{y_k^{(v)}}(-y|v_k = -1)$ folgt dann schließlich

$$f_{y_k^{(v)}}(y|v_k) = \frac{2}{\sqrt{2\pi\sigma_0^2}} \left(\exp\left[-\frac{(y-v_k)^2}{2\sigma_0^2} \right] \cdot Q\left(\frac{|y|-1}{\sigma_0} \right) + \exp\left[-\frac{(y+v_k)^2}{2\sigma_0^2} \right] \cdot Q\left(\frac{|y|+1}{\sigma_0} \right) \right).$$

A.6. Näherung des äquivalenten SNR für $y_k^{(v)}$ für große Rauschvarianz

In diesem Abschnitt wird abhängig vom SNR $= 2E_S/N_0$ eine Näherung des äquivalenten SNR für $y_k^{(v)}$ für SNR $\ll 1$ hergeleitet. Die Näherung beruht auf den Annahmen, daß im AWGN-Kanal die Summe der Kanalkapazitäten der äquivalenten Kanäle für die äußeren Codes bei Mehrstufendecodierung $\mathcal{C}^{(v)}$ und $\mathcal{C}^{(u)}$ gleich der Gesamtkapazität bei BPSK-Übertragung ist, d.h.

$$\mathcal{C}_{\text{BPSK}} = \frac{1}{2}(\mathcal{C}_y^{(v)} + \mathcal{C}_y^{(u)}) \tag{A.10}$$

und daß zudem auch $y_k^{(v)}$ gaußverteilt sei. Der Faktor $1/2$ berücksichtigt die Tatsache, daß die äußeren Codes nur die halbe Länge besitzen. Für kleine SNR ist die Kanalkapazität bei BPSK-Übertragung fast gleich der Kapazität bei Übertragung mit wertkontinuierlichem gaußverteiltem Eingangsalphabet (s. Bild 2.1)

$$\mathcal{C}_{\text{Gauss}} = \frac{1}{2}\log_2(1 + 2E_S/N_0).$$

Da gemäß (3.12) $y_k^{(u)}$ durch $y_k^{(u)} = y_k + \widehat{v}_k \cdot y_{N+k}$ gegeben ist, ist für richtiges $\widehat{v}_k$ auch $y_k^{(u)}$ gaußverteilt, es ergibt sich das Signal-zu-Rauschverhältnis für $y_k^{(u)}$ zu $\text{SNR}_u = 2 \cdot \text{SNR}$. Falls angenommen wird, daß auch $y_k^{(v)}$ gaußverteilt ist, kann nun SNR_v als äquivalentes SNR für $y_k^{(v)}$ so bestimmt werden, das Gl. (A.10) erfüllt ist. Mit der Näherung

$$\log_2(1+x) \approx \frac{1}{\ln 2}(x - \frac{x^2}{2})$$

kann damit geschrieben werden

$$2 \cdot \text{SNR} - \text{SNR}^2 \stackrel{!}{=} \text{SNR}_v - \underbrace{\frac{\text{SNR}_v^2}{2}}_{\approx 0} + \text{SNR}_u - \frac{\text{SNR}_u^2}{2}.$$

Wie schon angedeutet, kann er zweite Summand der rechten Seite vernachlässigt werden, da für kleine SNR gilt $\mathrm{SNR}_v \ll \mathrm{SNR}$. Damit ergibt sich

$$\mathrm{SNR}_v \approx \mathrm{SNR}^2$$

oder in logarithmischer Darstellung

$$\mathrm{SNR}_v\ [\mathrm{dB}]\ \approx 2 \cdot \mathrm{SNR}\ [\mathrm{dB}].$$

Für $[E_\mathrm{S}/N_0]_v$ als äquivalentem E_S/N_0 für $y_k^{(v)}$ ergibt sich daraus

$$\left[\frac{E_\mathrm{S}}{\mathrm{N}_0}\right]_v [\mathrm{dB}] \approx 2 \cdot \frac{E_\mathrm{S}}{N_0}[\mathrm{dB}] + 10 \cdot \log_{10} 2 \approx 2 \cdot \frac{E_\mathrm{S}}{N_0}[\mathrm{dB}] + 3\ \mathrm{dB}$$

Vergleich mit auf Wahrscheinlichkeiten basierten Zuverlässigkeitswerten

Für die auf Wahrscheinlichkeiten basierten Zuverlässigkeiten h_k wurde in [9] für den Erwartungswert $E\{h_k\}$ und $E\{h_k^2\}$ gezeigt, daß für $\sigma_0^2 \to \infty$

$$E\{h_k\} \sim E\{h_k^2\} \sim \sigma_0^{-2}$$

gilt. Für alle SNR gilt für das Verhältnis von $E\{h_k^{(v)}\}^2$ und Varianz von $h_k^{(v)}$, $V\{h_k^{(v)}\}$ [9, Gl. (16)]

$$\frac{E\{h_k^{(v)}\}^2}{V\{h_k^{(v)}\}} = \frac{E\{h_k\}^2 \cdot E\{h_{N+k}\}^2}{E\{h_k^2\} \cdot E\{h_{N+k}^2\} - (E\{h_k\} \cdot E\{h_{N+k}\})^2} = \frac{E\{h_k\}^4}{E\{h_k^2\}^2 - E\{h_k\}^4}.$$

Da aus $\sigma_0^2 \to \infty$ folgt $E\{h_k\}^2 \ll E\{h_k^2\}$, kann auch hier geschrieben werden

$$\frac{E\{h_k^{(v)}\}^2}{V\{h_k^{(v)}\}} \approx \frac{E\{h_k\}^4}{E\{h_k^2\}^2} \approx \left(\frac{E\{h_k\}^2}{V\{h_k\}}\right)^2,$$

so daß die obige Näherung für das äquivalente SNR auch für die auf Wahrscheinlichkeiten basierten Zuverlässigkeitswerte gilt.

A.7. Näherung des äquivalenten SNR für $y_k^{(v)}$ für kleine Rauschvarianz

In diesem Abschnitt wird abhängig vom $\mathrm{SNR} = 2E_\mathrm{S}/N_0$ eine Näherung des äquivalenten SNR für $y_k^{(v)}$ für $\mathrm{SNR} \gg 1$ hergeleitet. Die Näherung beruht auf den gleichen Annahmen wie im Abschnitt A.6, d.h. daß im AWGN-Kanal die Summe der Kanalkapazitäten der äquivalenten Kanäle für die äußeren Codes bei Mehrstufendecodierung $\mathcal{C}^{(v)}$ und $\mathcal{C}^{(u)}$ gleich der Gesamtkapazität bei BPSK-Übertragung ist, d.h.

$$\mathcal{C}_\mathrm{BPSK} = \frac{1}{2}(\mathcal{C}_y^{(v)} + \mathcal{C}_y^{(u)})$$

und daß zudem auch $y_k^{(v)}$ gaußverteilt sei. Die Gesamtkapazität $\mathcal{C}_{\text{BPSK}}$ ist mit $x = \sqrt{\frac{2E_S}{N_0}}$ gegeben durch (s. z.B. [32, Abschnitt 6.2.1])

$$\begin{aligned}\mathcal{C}_{\text{BPSK}} &= -\frac{1}{2}\log_2(\frac{4}{\mathrm{e}}) - \frac{1}{2\sqrt{2\pi}}\int_{-\infty}^{\infty}\left[\mathrm{e}^{-\frac{(y-x)^2}{2}} + \mathrm{e}^{-\frac{(y+x)^2}{2}}\right]\cdot\log_2\left[\mathrm{e}^{-\frac{(y-x)^2}{2}} + \mathrm{e}^{-\frac{(y+x)^2}{2}}\right]\mathrm{d}y \\ &= A_1 + A_2\int_{-\infty}^{\infty}\left[\mathrm{e}^{-\frac{(y-x)^2}{2}} + \mathrm{e}^{-\frac{(y+x)^2}{2}}\right]\cdot\log_2\left[\mathrm{e}^{-\frac{(y-x)^2}{2}} + \mathrm{e}^{-\frac{(y+x)^2}{2}}\right]\mathrm{d}y\end{aligned}$$

mit den Konstanten A_1 und A_2. Durch die Symmetrie des Integranden läßt sich auch schreiben

$$\mathcal{C}_{\text{BPSK}} = A_1 + A_2\int_{-\infty}^{\infty} 2\cdot\mathrm{e}^{-\frac{(y-x)^2}{2}}\cdot\log_2\left[\mathrm{e}^{-\frac{(y-x)^2}{2}} + \mathrm{e}^{-\frac{(y+x)^2}{2}}\right]\mathrm{d}y$$

und da gilt $\log(a+b) = \log a + \log(1+\frac{b}{a})$ kann man mit $a = \mathrm{e}^{-\frac{(y-x)^2}{2}}$ weiter schreiben

$$\begin{aligned}\mathcal{C}_{\text{BPSK}} = A_1 + \\ 2A_2\underbrace{\int_{-\infty}^{\infty}\mathrm{e}^{-\frac{(y-x)^2}{2}}\log_2\mathrm{e}^{-\frac{(y-x)^2}{2}}\mathrm{d}y}_{\text{unabhängig von } x} + 2A_2\int_{-\infty}^{\infty}\mathrm{e}^{-\frac{(y-x)^2}{2}}\log_2\left[1+\mathrm{e}^{\frac{-(y+x)^2+(y-x)^2}{2}}\right]\mathrm{d}y.\end{aligned}$$

Wie angedeutet ist das erste Integral unabhängig von x und daher kann obige Gleichung mit den Konstanten A_3, A_4 weiter vereinfacht werden zu

$$\begin{aligned}\mathcal{C}_{\text{BPSK}} &= A_3 + A_4\int_{-\infty}^{\infty}\mathrm{e}^{-\frac{(y-x)^2}{2}}\log_2\left[1+\mathrm{e}^{-2xy}\right]\mathrm{d}y \\ &= A_3 + A_4\cdot\mathrm{e}^{-\frac{x^2}{2}}\int_{-\infty}^{\infty}\mathrm{e}^{-\frac{y^2}{2}}\cdot e^{xy}\log_2\left[1+\mathrm{e}^{-2xy}\right]\mathrm{d}y \qquad \text{(A.11)}\end{aligned}$$

Der Verlauf des Integranden aus Gl. (A.11) ist in Bild A.3 für einige SNR-Werte gezeigt. Es ist zu erkennen, daß das Maximum des Integranden bei $y \approx 0$ liegt und er für große SNR nur in der Nähe von $y = 0$ wesentlich von Null verschieden ist.

Damit kann mit $\mathrm{e}^{-\frac{y^2}{2}} \approx 1$ Gleichung (A.11) auch Näherungsweise geschrieben werden als

$$\mathcal{C}_{\text{BPSK}} \approx A_3 + A_4\cdot\mathrm{e}^{-\frac{x^2}{2}}\int_{-\infty}^{\infty} e^{xy}\log_2\left[1+\mathrm{e}^{-2xy}\right]\mathrm{d}y$$

Mit der Substitution $t = \mathrm{e}^{xy}$ folgt daraus

$$\mathcal{C}_{\text{BPSK}} \approx A_3 + A_4\cdot\mathrm{e}^{-\frac{x^2}{2}}\int_{0}^{\infty}\frac{1}{x}\log_2\left[1+\frac{1}{t^2}\right]\mathrm{d}t$$

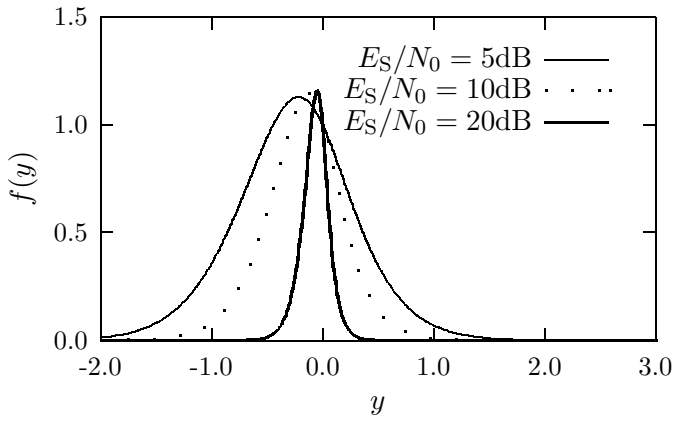

Abbildung A.3.: Integrand $f(y) = \mathrm{e}^{-\frac{y^2}{2}} \cdot e^{xy} \log_2\left[1 + \mathrm{e}^{-2xy}\right]$ aus Gl. (A.11)

und mit $\int_0^\infty \ln(1 + \frac{1}{t^2})\mathrm{d}t = \pi$ [21, Nr. 4.222.1] ergibt sich schließlich

$$C_{\text{BPSK}} \approx A_3 + A_5 \cdot \frac{1}{x}\mathrm{e}^{-\frac{x^2}{2}} \tag{A.12}$$

Da gemäß (3.12) $y_k^{(u)}$ durch $y_k^{(u)} = y_k + \widehat{v}_k \cdot y_{N+k}$ gegeben ist, ist für richtiges $\widehat{v}_k$ auch $y_k^{(u)}$ gaußverteilt, es ergibt sich wie im vorherigen Abschnitt $\text{SNR}_u = 2 \cdot \text{SNR}$. Unter der Annahme, daß auch $y_k^{(v)}$ gaußverteilt ist, kann wie oben SNR_v als äquivalentes SNR für $y_k^{(v)}$ so bestimmt werden, das Gl. (A.10) erfüllt ist. Für große SNR ergibt sich damit mit (A.12) die Forderung

$$\frac{2}{x}\mathrm{e}^{-\frac{x^2}{2}} \stackrel{!}{=} \frac{1}{x_v}\mathrm{e}^{-\frac{x_v^2}{2}} + \underbrace{\frac{1}{x_u}\mathrm{e}^{-\frac{x_u^2}{2}}}_{\approx 0 \text{ für SNR} \gg 1}$$

wobei $x_v^2 = \text{SNR}_v$ und $x_u^2 = \text{SNR}_u$ gesetzt wird. Wie schon angedeutet, kann für große SNR der zweite Summand der rechten Seite vernachlässigt werden und damit ergibt sich SNR_v als Lösung der Gleichung

$$\text{SNR} + \ln \text{SNR} - 2\ln 2 \stackrel{!}{=} \text{SNR}_v + \ln \text{SNR}_v.$$

Mit wachsendem SNR wird $\text{SNR} \approx \text{SNR}_v$, damit ist $\ln(\text{SNR}/\text{SNR}_v) \approx 0$ und es kann letztendlich das äquivalente SNR für $y_k^{(v)}$ direkt angegeben werden

$$\text{SNR}_v \approx \text{SNR} - 2\ln 2$$

oder $[E_S/N_0]_v$ als äquivalentes E_S/N_0 für $y_k^{(v)}$

$$\left[\frac{E_S}{N_0}\right]_v \approx \frac{E_S}{N_0} - \ln 2.$$

A.8. Beweis der Exaktheit von Gl. (3.19) im BSC

In diesem Abschnitt wird gezeigt, daß bei Übertragung über den BSC und Zuverlässigkeitsübergabe gemäß (3.11) und (3.12) das Gleichheitszeichen in Gl. (3.19) gilt. Streng genommen muß dies allein schon auf Grund der Tatsache gelten, daß mit den auf Wahrscheinlichkeiten basierten Zuverlässigkeitsübergaben $h_k^{(v)}, h_k^{(u)}$ Gleichung (3.18) gilt und auch hier nur zwei- bzw. dreiwertige Symbole und damit nicht mehr Information an die Decoder von V und U übergeben wird als durch die $y_k^{(v)}, y_k^{(u)}$. Trotzdem soll der Beweis hier der Vollständigkeit halber geführt werden.

Bei Übertragung über den BSC mit $y_k \in \{\pm 1\}$ ist auch $y_k^{(v)} \in \{\pm 1\}$ zweiwertig, für den äquivalenten Kanal ergibt sich die Fehlerwahrscheinlichkeit p' aus der Fehlerwahrscheinlichkeit p des BSC zu

$$p' = P(y_k^{(v)} \neq v_k) = 2p \cdot (1-p).$$

Die Zuverlässigkeitswerte für U sind allerdings dreiwertig, $y_k^{(u)} \in \{-2, 0, +2\}$ mit den Übergangswahrscheinlichkeiten

$$\begin{aligned} P(y_k^{(u)} = 2u_k) &= (1-p)^2 \\ P(y_k^{(u)} = 0) &= 2p \cdot (1-p) \\ P(y_k^{(u)} = -2u_k) &= p^2. \end{aligned}$$

Die Summe der Transinformationen kann nun unter der Annahme, daß alle Symbole v_k und u_k gleichwahrscheinlich sind, durch die Entropien H ausgedrückt werden

$$\begin{aligned} & I(Y_k^{(v)}; V_k) + I(Y_k^{(u)}; U_k | V_k) \\ &= H(Y_k^{(v)}) - H(Y_k^{(v)}|V_k) + H(Y_k^{(u)}|V_k) - H(Y_k^{(u)}|U_k, V_k) \\ &= 1 + E\{\log_2 P(y_k^{(v)}|v_k)\} - E\{\log_2 P(y_k^{(u)}|v_k)\} + E\{\log_2 P(y_k^{(u)}|v_k, u_k)\} \end{aligned}$$

Einsetzen der Wahrscheinlichkeiten ergibt

$$\begin{aligned} & I(Y_k^{(v)}; V_k) + I(Y_k^{(u)}; U_k | V_k) \\ &= 1 + p' \log_2 p' + (1-p') \log_2 (1-p') \\ &\quad - \Big[\frac{1}{2}(1 - 2p + 2p^2) \log_2 \frac{1}{2}(1 - 2p + 2p^2) + 2p \cdot (1-p) \log_2 2p \cdot (1-p) \\ &\quad + \frac{1}{2}(1 - 2p + 2p^2) \log_2 \frac{1}{2}(1 - 2p + 2p^2) \Big] \\ &\quad + (1-p)^2 \log_2 (1-p)^2 + 2p \cdot (1-p) \log_2 2p \cdot (1-p) + p^2 \log_2 p^2 \end{aligned}$$

Durch weitere Rechnung folgt schließlich

$$\begin{aligned} I(Y_k^{(v)}; V_k) + I(Y_k^{(u)}; U_k | V_k) &= 2 \cdot (1 + (1-p) \log_2 (1-p) + p \log_2 p) \\ &= I(Y_k, Y_{N+k}; C_k, C_{N+k}) \end{aligned}$$

was zu beweisen war.

A.9. Beweis von Satz 6.2 für den BSC

In diesem Abschnitt wird gezeigt, daß beim BSC mit Symbolfehlerwahrscheinlichkeit $p = 1-q$ und Cutoff-Rate R_{comp} und bei einer Zerlegung von C in zwei äußere Codes U, V für die Summe der Cutoff-Raten der äquivalenten Kanäle gilt

$$R_{\text{comp}} < \frac{1}{2}(R_{\text{comp}}^{(U)} + R_{\text{comp}}^{(V)}),$$

falls $0 < p < 1/2$. Die Cutoff-Rate für den BSC ist allgemein gegeben durch (z.B. [19, Abschnitt 2.3])

$$R_{\text{comp}} = 1 - \log_2(1 + 2\sqrt{pq}).$$

Sowohl bei distanzbasierter als auch bei wahrscheinlichkeitsbasierter Zuverlässigkeitsübergabe handelt es sich beim äquivalenten Kanal für V ebenfalls um einen BSC, allerdings mit Symbolfehlerwahrscheinlichkeit $p^{(v)} = 2pq$. Damit ist die Cutoff-Rate diese Kanals

$$R_{\text{comp}}^{(V)} = 1 - \log_2(1 + 2\sqrt{2pq(q^2+p^2)}).$$

Da beide Zuverlässigkeitsübergaben für die Decodierung von U gleichwertig sind, muß hier nur die distanzbasierte Übergabe betrachtet werden. Das Ausgangsalphabet des äquivalenten Kanals zur Übertragung von U ist damit dreiwertig, $y_k^{(u)} \in \{-2, 0, +2\}$ mit den Übergangswahrscheinlichkeiten $P(y_k^{(u)}|u_k)$ gemäß Bild A.4. Die Cutoff-Rate für diesen Kanal ergibt sich aus Gl. (2.10) zu

$$\begin{aligned} R_{\text{comp}}^{(U)} &= 1 - \log_2\left[1 + \sum_{y_k^{(u)} \in \{-2,0,2\}} \sqrt{P(y_k^{(u)}|u_k = 1) \cdot P(y_k^{(u)}|u_k = -1)}\right] \\ &= 1 - \log_2(1 + 4pq). \end{aligned}$$

Nun gilt es zu zeigen, daß für $0 < p < 1/2$ gilt

$$\log_2(1 + 2\sqrt{2pq \cdot (p^2+q^2)}) + \log_2(1+4pq) < 2\log_2(1 + 2\sqrt{pq}).$$

Nach einigen Umformungen ergibt sich daraus für $p \neq 0$ die Ungleichung

$$(1+4pq)\sqrt{p^2+q^2} < \sqrt{2}.$$

Da mit $q = 1-p$ gilt

$$\frac{\mathrm{d}}{\mathrm{d}p}(1 + 4p(1-p))\sqrt{p^2+(1-p)^2} = \frac{-48 \cdot (p-\frac{1}{2})^3}{2\sqrt{p^2+(1-p)^2}},$$

liegt das (einzige) Maximum der linken Seite obiger Ungleichung bei $p = 1/2$. Einsetzen ergibt schließlich

$$(1+4pq)\sqrt{p^2+q^2}\Big|_{p=1/2} = \sqrt{2},$$

womit für alle $0 < p < 1/2$ die Ungleichung wie gefordert erfüllt ist.

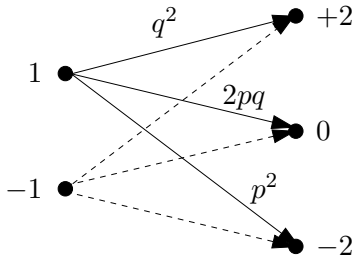

Abbildung A.4.: Übergangswahrscheinlichkeiten für den äquivalenten Kanal für U

A.10. Berechnung der mittleren Listengröße $L_v^{(\mathrm{ML})}$ bei HDML-Decodierung

In diesem Abschnitt wird für den BSC und Zerlegung des Codes C in zwei äußere Codes U und V die mittlere Anzahl der Codeworte $\boldsymbol{v} \in V$ in der Liste $\mathcal{L}_v$ bestimmt, die bei Genie-Aided Decodierung notwendig ist, um eine ML-Decodierung zu garantieren. Es wird hierzu die Gewichtsverteilung $A_d^{(v)}$ des Codes V benötigt.

Wie in Abschnitt 5.3.3 gezeigt wurde muß ein Genie-Aided Decoder, der die wahre Anzahl e der aufgetretenen Übertragungsfehler kennt, um eine ML-Decodierung zu garantieren nur die Hypothesen $\widehat{\boldsymbol{v}} = (\widehat{v}_1, \widehat{v}_2, \ldots, \widehat{v}_N)$ in der Liste behalten, für die gilt

$$4 \cdot \Lambda_y(\widehat{\boldsymbol{v}}) \leq d_{\mathrm{E}}^2(\boldsymbol{y}, \boldsymbol{c}) - d_{\mathrm{bias}}^2 \overset{\mathrm{BSC}}{=} 4 \cdot d_{\mathrm{H}}(\boldsymbol{y}, \boldsymbol{c}) = 4e.$$

Ohne Beschränkung kann angenommen werden, daß der Vektor $\boldsymbol{v} = (1, 1, \ldots, 1)$ gesendet wurde. Sei $\boldsymbol{v}_d \in V$ ein Codewort mit Hamminggewicht d und $e^{(v)}$ die Anzahl der Positionen $\widetilde{v}_k^{(v)} \neq 1$ (s. Abschn. 3.2). Zur leichteren Betrachtung werde ferner angenommen, daß bei $\boldsymbol{v}_d$ die ersten d Komponenten gleich -1 sind. Damit kann geschrieben werden

$$e^{(v)} = e_1^{(v)} + e_2^{(v)},$$

wobei $e_1^{(v)}$ die Anzahl der Positionen $\widetilde{v}_k^{(v)} \neq 1$ mit $k \leq d$ und $e_2^{(v)}$ die Anzahl der Positionen $\widetilde{v}_k^{(v)} \neq 1$ mit $k > d$ bezeichnet. Für den Hammingabstand zwischen $\boldsymbol{v}_d$ und dem Vektor der geschätzten Symbole $\widetilde{\boldsymbol{v}} = (\widetilde{v}_1, \widetilde{v}_2, \ldots, \widetilde{v}_N)$ gilt nun

$$d_{\mathrm{H}}(\boldsymbol{v}_d, \widetilde{\boldsymbol{v}}) = (d - e_1^{(v)}) + e_2^{(v)}$$

und damit das Codewort $\boldsymbol{v}_d$ in der Liste verbleibt, müssen daher die beiden folgenden Bedingungen erfüllt sein

$$\begin{aligned} e_1^{(v)} &\geq \frac{d - e + e^{(v)}}{2} \\ e_2^{(v)} &\leq \frac{e + e^{(v)} - d}{2}. \end{aligned}$$

Für die Wahrscheinlichkeit, daß das Codewort $\boldsymbol{v}_d$ in der Liste $\mathcal{L}_v$ verbleibt unter der Bedingung, daß e und $e^{(v)}$ Fehler aufgetreten sind , kann nun geschrieben werden

$$P(\boldsymbol{v}_d \in \mathcal{L}_v | e, e^{(v)}) = \frac{\sum_{e_2^{(v)}=0}^{(e+e^{(v)}-d)/2} \binom{d}{e_1^{(v)}} \binom{N-d}{e_2^{(v)}}}{\binom{N}{e^{(v)}}}.$$

Summierung über alle Codeworte $\boldsymbol{v}$ ergibt für die mittlere Anzahl L_v der Hypothesen in der Liste $\mathcal{L}_v$ unter der Bedingung e und $e^{(v)}$

$$E\{L_v | e, e^{(v)}\} = \sum_d A_d^{(v)} \cdot P(\boldsymbol{v}_d \in \mathcal{L}_v | e, e^{(v)}),$$

wobei $A_d^{(v)}$ das Distanzspektrum des Codes V bezeichnet. Weiter kann für die mittlere Listengröße unter der Bedingung, daß e Fehler aufgetreten sind, geschrieben werden

$$\begin{aligned} E\{L_v|e\} &= \sum_d A_d^{(v)} \cdot P(\boldsymbol{v}_d \in \mathcal{L}_v | e) \\ &= \sum_d A_d^{(v)} \cdot \sum_{e^{(v)}} P(\boldsymbol{v}_d \in \mathcal{L}_v | e, e^{(v)}) \cdot P(e^{(v)}|e) \end{aligned}$$

so daß schließlich für den BSC mit Fehlerwahrscheinlichkeit p folgt

$$\begin{aligned} E\{L_v\} &= \sum_e E\{L_v|e\} \cdot P(e) \\ &= \sum_{e=0}^{2N} E\{L_v|e\} \cdot \binom{2N}{e} p^e (1-p)^{2N-e}. \end{aligned}$$

Die verbleibende, noch zu bestimmende Funktion $P(e^{(v)}|e)$ ergibt sich wie folgt: es werden die N Paare (y_k, y_{N+k}) betrachtet, aus denen sich gemäß Gl. (3.11) $y_k^{(v)}$ berechnet. Bei $e^{(v)}$ Fehlern $\widetilde{v}_k \neq 1$ müssen bei exakt $e^{(v)}$ solcher Paare genau ein Fehler aufgetreten sein, alle anderen Paare müssen entweder fehlerfrei sein oder zwei Fehler enthalten und damit müssen die restlichen $(e - e^{(v)})$ Fehler paarweise auftreten. Damit ergibt sich die Anzahl $\mathcal{A}(e, e^{(v)})$ der möglichen Ereignisse mit e Empfangsfehlern und $e^{(v)}$ Fehler in $\widetilde{\boldsymbol{v}}$ zu[3]

$$\mathcal{A}_N(e, e^{(v)}) = \binom{N}{e^{(v)}} \binom{2}{1}^{e^{(v)}} \binom{N - e^{(v)}}{(e - e^{(v)})/2}$$

und schließlich die gesuchte Wahrscheinlichkeit

$$P(e^{(v)}|e) = \frac{\mathcal{A}_N(e, e^{(v)})}{\binom{2N}{e}}.$$

[3]Hier wird vorausgesetzt, daß $\binom{n}{k}$ nur dann ungleich Null ist, falls n und k ganze Zahlen sind.

Ergänzung

Rekursiv kann natürlich auf die gleiche Weise auch die mittlere Listengröße $L_{c^{(1)}}^{(\mathrm{ML})}$ bei einer Zerlegung von C in vier äußere Codes $C^{(1)}, \ldots, C^{(4)}$ der Länge N bestimmt werden. Der Gesamtcode hat damit die Länge $4N$. Es sei $e^{(1)}$ die Anzahl der Fehler in $\widetilde{\boldsymbol{y}}^{(1)}$. Dann ist die Wahrscheinlichkeit $P(e^{(1)}|e)$ gegeben durch

$$P(e^{(1)}|e) = \sum_{e^{(v)}} P(e^{(1)}|e^{(v)}) \cdot P(e^{(v)}|e)$$

mit

$$P(e^{(1)}|e^{(v)}) = \frac{\mathcal{A}_N(e^{(v)}, e^{(1)})}{\binom{2N}{e^{(v)}}}$$

wenn in $P(e^{(v)}|e)$ entsprechend N durch $2N$ ersetzt wird. Die weitere Rechnung erfolgt dann wie oben.

A.11. Anzahl der Permutation bei Zerlegung in vier äußere Codes

In diesem Abschnitt wird für RM-Codes die Anzahl der Permutationen bestimmt, die bei einer Zerlegung in vier äußeres Codes $C^{(1)}, \ldots, C^{(4)}$ zu verschiedenen Paaren bei der Bestimmung der Werte $a_k^{(1)}$ führen. Allgemein ist, wenn die $\boldsymbol{x}_k$ entsprechend ihrer binären Darstellung geordnet sind

$$a_k^{(1)} = a_k \cdot a_{N^{(1)}+k} \cdot a_{2N^{(1)}+k} \cdot a_{3N^{(1)}+k},$$

mit $N^{(1)}$ der Codelänge von $C^{(1)}$. Aus den gleichen Überlegungen wie in Abschnitt 5.4 ist bei Wahl von $\boldsymbol{A}$ und $\boldsymbol{b}$ die Erfüllung folgendes Gleichungssystem erforderlich, damit nach der Permutation Π die Symbole $a_{\Pi(k)}$, $a_{\Pi(N^{(1)}+k)}$, $a_{\Pi(2N^{(1)}+k)}$, $a_{\Pi(3N^{(1)}+k)}$ bei der Berechnung von $a_k^{(1)}$ miteinander multipliziert werden

$$\begin{aligned}
\boldsymbol{x}_k &= \boldsymbol{A} \cdot \boldsymbol{x}_{\Pi(k)} + \boldsymbol{b} \\
\boldsymbol{x}_{N^{(1)}+k} &= \boldsymbol{A} \cdot \boldsymbol{x}_{\Pi(N^{(1)}+k)} + \boldsymbol{b} \\
\boldsymbol{x}_{2N^{(1)}+k} &= \boldsymbol{A} \cdot \boldsymbol{x}_{\Pi(2N^{(1)}+k)} + \boldsymbol{b} \\
\boldsymbol{x}_{3N^{(1)}+k} &= \boldsymbol{A} \cdot \boldsymbol{x}_{\Pi(3N^{(1)}+k)} + \boldsymbol{b}.
\end{aligned}$$

Berücksichtigt man auch hier, daß gilt

$$\begin{aligned}
\boldsymbol{x}_{N^{(1)}+k} &= \boldsymbol{x}_k + \boldsymbol{x}_{N^{(1)}+1} \\
\boldsymbol{x}_{2N^{(1)}+k} &= \boldsymbol{x}_k + \boldsymbol{x}_{2N^{(1)}+1} \\
\boldsymbol{x}_{3N^{(1)}+k} &= \boldsymbol{x}_k + \boldsymbol{x}_{3N^{(1)}+1} = \boldsymbol{x}_k + \boldsymbol{x}_{N^{(1)}+1} + \boldsymbol{x}_{2N^{(1)}+1}
\end{aligned}$$

mit

$$\begin{aligned}
\boldsymbol{x}_{N^{(1)}+1} &= (1, -1, 1, 1, \ldots, 1)^T \\
\boldsymbol{x}_{2N^{(1)}+1} &= (-1, 1, 1, 1, \ldots, 1)^T,
\end{aligned}$$

so folgt hieraus das zu erfüllende Gleichungssystem

$$\begin{aligned}
\boldsymbol{x}_{\Pi(k)} &= \boldsymbol{A}^{-1} \cdot (\boldsymbol{x}_k - \boldsymbol{b}) \\
\boldsymbol{x}_{\Pi(N^{(1)}+k)} &= \boldsymbol{x}_{\Pi(k)} + \boldsymbol{A}^{-1} \cdot \boldsymbol{x}_{N^{(1)}+1} \\
\boldsymbol{x}_{\Pi(2N^{(1)}+k)} &= \boldsymbol{x}_{\Pi(k)} + \boldsymbol{A}^{-1} \cdot \boldsymbol{x}_{2N^{(1)}+1} \\
\boldsymbol{x}_{\Pi(3N^{(1)}+k)} &= \boldsymbol{x}_{\Pi(k)} + \boldsymbol{A}^{-1} \cdot (\boldsymbol{x}_{N^{(1)}+1} + \boldsymbol{x}_{2N^{(1)}+1}).
\end{aligned}$$

Damit ergibt sich, daß auch hier der Vektor $\boldsymbol{b}$ sowie alle Spalten von $\boldsymbol{A}$ mit Ausnahme der ersten beiden nur die Reihenfolge der Symbole in $\boldsymbol{a}^{(1)}$ beeinflussen, die Gruppen aber unverändert bleiben. Für die Auswahl der Gruppen $(a_{\Pi(k)}, \ldots, a_{\Pi(3N^{(1)}+k)})$ sind nur die ersten beiden Spalten von $\boldsymbol{A}$ entscheidend, für die es $(2^m - 2^0)(2^m - 2^1)$ Möglichkeiten gibt. Allgemein gilt damit für die Anzahl der Permutationen P, die verschiedene Gruppen ergeben

$$|P| = \frac{(2^m - 2^0)(2^m - 2^1)}{(2^2 - 2^0)(2^2 - 2^1)},$$

da jeweils $(2^2 - 2^0)(2^2 - 2^1)$ Matrizen $\boldsymbol{A}$ mit unterschiedlichen ersten beiden Spalten zu den gleichen Gruppen führen[4]. Es ist offensichtlich, daß nicht alle Paare $(a_{\Pi(k)}, \ldots, a_{\Pi(3N^{(1)}+k)})$ mit beliebigen Gruppierungen der Symbole a_k möglich sind.

A.12. Verallgemeinerung von Theorem 5.1

In diesem Abschnitt wird das in Abschnitt 5.5 für eine Zerlegung in vier äußere Codes vorgestellte Theorem 5.1 verallgemeinert. Es wird hierzu auf die gleiche Notation zurückgegriffen, d.h. bei Zerlegung in 2^s äußere Codes $C^{(1)}, C^{(2)}, \ldots, C^{(2^s)}$ sei $\boldsymbol{c}_k = (c_k^{(1)}, c_k^{(2)}, \ldots, c_k^{(2^s)}) \cdot \boldsymbol{B}_{2^s}$ die k-te Zeile der Matrix $\boldsymbol{C}$ und der dazugehörige Empfangsvektor sei $\boldsymbol{y}_k = (y_{k1}, y_{k2}, \ldots, y_{k2^s})$.

Weiterhin wird hier der Koordinatenvektor $\boldsymbol{x} = (x_1, x_2, \ldots, x_s)^T$, $x_m \in \{\pm 1\}$ der euklidischen Geometrie der Dimension s über $\mathrm{GF}(2)$, $\mathrm{EG}(s,2)$ benötigt, und die $(s \times 2^s)$-Matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{2^s})$ enthalte in ihren Spalten die verschiedenen $\boldsymbol{x}_n \in \mathrm{EG}(s,2)$, geordnet entsprechend der binären Schreibweise wie in Kapitel 4 beschrieben

$$\boldsymbol{X} = \begin{bmatrix} 1 & 1 & 1 & 1 & \ldots & -1 & -1 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & 1 & -1 & -1 & \ldots & -1 & -1 \\ 1 & -1 & 1 & -1 & \ldots & 1 & -1 \end{bmatrix}.$$

X_{mn} bezeichne das Element in der m-ten Zeile und n-ten Spalte von $\boldsymbol{X}$. Eine Fläche der Dimension μ ist allgemein bestimmt durch alle Punkte $\boldsymbol{x}$, die ein Gleichungssystem von $(s - \mu)$ unabhängigen linearen Gleichungen

$$\sum_{m=1}^{s} a_{jm} x_m = b_j \quad \text{für} \quad j = 1, 2, \ldots, (s - \mu)$$

[4] Dieses Problem ist äquivalent zur Definition von Ebenen mit Dimension 2 in der Euklidischen Geometrie (EG) der Dimension m über $\mathrm{GF}(2)$. In $\mathrm{EG}(m,2)$ gibt es genau $|P|$ verschiedene Ebenen der Dimension 2.

mit $a_{jm}, b_j \in \{\pm 1\}$ erfüllen. In gleicher Weise ist der Inzidenzvektor $\boldsymbol{f} = (f_1, \ldots, f_{2^s})$ einer μ-Fläche durch das Produkt über $(s-\mu)$ Faktoren

$$f_n = \prod_{j=1}^{s-\mu} \left(b_j + 1 + \sum_{m=1}^{s} a_{jm} x_{nm} \right),$$

mit $\boldsymbol{x}_n = (x_{n1}, x_{n2}, \ldots, x_{ns})$ bestimmt. Es wird nun die Radix-2-Darstellung der Integer $h \in \{0, 1, \ldots, 2^s - 1\}$ definiert zu

$$h = \delta_1^{(h)} \cdot 2^0 + \delta_2^{(h)} \cdot 2^1 + \ldots + \delta_s^{(h)} \cdot 2^{s-1},$$

$\delta_m^{(h)} \in \{0, 1\}$ und das Radix-2-Gewicht entsprechend als

$$\mathrm{w}_2(h) = \sum_m \delta_m^{(h)}.$$

Mit diesen Definitionen ist der Vektor $\boldsymbol{\beta}^{(h)} = (\beta_1^{(h)}, \beta_2^{(h)}, \ldots, \beta_L^{(h)})$ der Länge $L = 2^s$ mit den Elementen[5]

$$\beta_n^{(h)} = X_{1,n}^{\delta_1^{(h)}} \cdot X_{2,n}^{\delta_2^{(h)}} \cdots X_{s,n}^{\delta_s^{(h)}}$$

der Inzidenzvektor der μ-Fläche in $\mathrm{EG}(s, 2)$ mit Dimension $\mu = s - \mathrm{w}_2(h)$, die durch Punkte $\boldsymbol{x}$ gegeben ist, die die Gleichung

$$x_1^{\delta_1^{(h)}} \cdot x_2^{\delta_2^{(h)}} \cdots x_s^{\delta_s^{(h)}} = -1$$

erfüllen.

Nun ist bei einer Zerlegung in 2^s äußere Codes die Generatormatrix $\boldsymbol{B}_L$ des inneren Codes B_L gegeben durch

$$\boldsymbol{B}_L = (\boldsymbol{\beta}^{(L-1)}, \boldsymbol{\beta}^{(L-2)}, \ldots, \boldsymbol{\beta}^{(0)})^T.$$

Basierend auf der Entscheidung $\widehat{\boldsymbol{c}}^{(1)}$ können auch hier bei einer Zerlegung in 2^s Codes, wie im Abschnitt 5.5 für eine Zerlegung in vier äußere Codes vorgestellt, von $\boldsymbol{y}_k$ abhängige Zuverlässigkeitswerte für Codesymbole $c_k^{(i)}$ derjenigen Codes $C^{(i)}$ berechnet werden, deren zugehöriger Zeilenvektor $\boldsymbol{\beta}^{(L-i)}$ das Hamminggewicht $w_{\mathrm{H}}(\boldsymbol{\beta}^{(L-i)}) = 2$ besitzt. Genau die s Zeilenvektoren $\boldsymbol{\beta}^{(L-i)}$ mit $\mathrm{w}_2(L-i) = s-1$, was gleichwertig ist zur Bedingung $\mathrm{w}_2(i-1) = 1$, besitzen das Hamminggewicht 2. Diese Vektoren sind gleichzeitig Inzidenzvektoren von Linien in $\mathrm{EG}(s, 2)$, alle diese Linien sind nicht parallel zueinander und schneiden sich im Punkt $\boldsymbol{x}_L = (-1, -1, \ldots, -1)$. In $\mathrm{EG}(s, 2)$ gibt es zu jeder durch $\boldsymbol{\beta}^{(L-i)}$, $\mathrm{w}_2(i-1) = 1$ definierten Linie eine $(s-1)$-Fläche, die mit dieser Linie nur den Punkt $\boldsymbol{x}_L$ gemeinsam hat aber alle anderen nicht parallelen Linien $\boldsymbol{\beta}^{(L-i')}$, $i' \neq i$ in zwei Punkten schneidet. Die dazu parallel verschobene $(s-1)$-Fläche schneidet die Linie $\boldsymbol{\beta}^{(L-i)}$ in dem zweiten Punkt der Linie[6] und alle anderen Linien $\boldsymbol{\beta}^{(L-i')}$, $i' \neq i$ nicht. Der folgende Satz liefert eine Verallgemeinerung dieses Sachverhaltes.

[5] Da das Eins-Element durch (-1) gegeben ist, gilt hier $1^0 = (-1)^0 = -1$

[6] In $\mathrm{EG}(s, 2)$ besteht eine Linie aus genau zwei Punkten.

Satz A.1 Die $(s-1)$-Fläche, die für gegebenes i mit $\mathrm{w}_2(i-1)=1$ mit der durch den Inzidenzvektor $\boldsymbol{\beta}^{(L-i)}$ definierten Linie nur den Punkt $\boldsymbol{x}_L=(-1,-1,\ldots,-1)$ gemeinsam hat und alle anderen durch $\boldsymbol{\beta}^{(L-i')}$, $\mathrm{w}_2(i'-1)\geq 1$, $i'\neq i$ definierten Flächen in $2^j, j\geq 1$ Punkten schneidet, ist durch den Inzidenzvektor $\boldsymbol{f}^{(i)}=(f_1^{(i)},f_2^{(i)},\ldots f_L^{(i)})$, $f_n^{(i)}\in\{\pm 1\}$ mit

$$\boldsymbol{f}^{(i)}=\boldsymbol{\beta}^{(i-1)}$$

gegeben.

Beweis:

1. Die zu $\boldsymbol{f}^{(i)}$ gehörende μ-Fläche hat wie gefordert die Dimension $\mu=s-\mathrm{w}_2(i-1)=s-1$.
2. Die zu $\boldsymbol{f}^{(i)}$ gehörende Fläche schneidet die Fläche von $\boldsymbol{\beta}^{(L-i)}$ nur im Punkt $\boldsymbol{x}_L$, denn mit $\boldsymbol{x}_n=(x_{n1},x_{n2},\ldots,x_{ns})$ ist[7]

$$\begin{aligned} f_n^{(i)}\cdot\beta_n^{(L-i)} &= x_{n1}^{\delta_1(i-1)+\delta_1(L-i)}\cdot x_{n2}^{\delta_2(i-1)+\delta_2(L-i)}\cdots x_{ns}^{\delta_s(i-1)+\delta_s(L-i)} \\ &= x_{n1}\cdot x_{n2}\cdots x_{ns} \end{aligned}$$

   da für alle m gilt[8] $\delta_m(L-i)=\delta_m(L-1-(i-1))=1-\delta_m(i-1)$.

3. Weiterhin schneidet die zu $\boldsymbol{f}^{(i)}$ gehörende Fläche alle anderen μ-Flächen, $\mu\geq 1$, die durch $\boldsymbol{\beta}^{(L-i')}$ mit $\mathrm{w}_2(i'-1)\geq 1$, $i'\neq i$ definiert sind, neben dem Punkt $\boldsymbol{x}_L$ auch in mindestens einem weiteren Punkt, da es sich bei der Radix-2-Darstellung von $(i'-1)$ nicht um das 2er-Komplement von $(L-1)$ handelt und damit mindestens ein Exponent der rechten Seite von

$$f_n^{(i)}\cdot\beta_n^{(L-i')}=x_{n1}^{\delta_1(i-1)+\delta_1(L-i)}\cdot x_{n2}^{\delta_2(i-1)+\delta_2(L-i)}\cdots x_{ns}^{\delta_s(i-1)+\delta_s(L-i)}$$

   gleich Null ist. Daher bilden die Schnittpunkte ein Fläche mit Dimension ≥ 1.

□

Die zum Inzidenzvektor $\boldsymbol{f}^{(i_1)}$, $\mathrm{w}_2(i_1-1)=1$ gehörende $(s-1)$-Fläche und deren parallele Verschiebung haben damit, wie schon in Kapitel 4 beschrieben, die Eigenschaft, daß sie für alle $i_2\neq i_1$ und $\mathrm{w}_2(i_2-1)\geq 1$ die zu $\boldsymbol{\beta}^{(L-i_2)}$ gehörenden Flächen nicht oder in einer geraden Anzahl von Punkten schneiden. Im ersten Schritt können so in bekannter Weise die verschiedenen Zuverlässigkeitswerte[9] $y_k'^{(i)}$ zur Decodierung der s Codes $C^{(i)}$ mit $\mathrm{w}_2(i-1)=1$ berechnet werden

$$y_k'^{(i)}=\prod_{\{n:f_n^{(i)}=0\}}a_{kn}\cdot\min_{\{n:f_n^{(i)}=0\}}w_{kn}+\widehat{c}_k^{(1)}\cdot\prod_{\{n:f_n^{(i)}=1\}}a_{kn}\cdot\min_{\{n:f_n^{(i)}=1\}}w_{kn}$$

[7] Für $x\in\mathrm{GF}(2)$ ist $x^2=x\cdot x=x$.

[8] Es gilt $L-i+(i-1)=L-1$ und $\mathrm{w}_2(L-1)=s$, damit ist in der Radix-2-Darstellung $L-i$ das 2er-Komplement zu $i-1$.

[9] Das Primzeichen kennzeichnet hier, daß die verschiedenen Zuverlässigkeitswerte nur von der Entscheidung $\widehat{c}_k^{(1)}$, nicht aber von den anderen Entscheidungen $\widehat{c}_k^{(i)}$, $i>1$ abhängen.

mit $a_{kn} = \operatorname{sign} y_{kn}$ und $w_{kn} = |y_{kn}|$.

Es ist[10] $\overline{\boldsymbol{f}}^{(i)} = (\boldsymbol{-1}) + \boldsymbol{f}^{(i)}$ der Inzidenzvektor der $(s-1)$-Fläche, die zur $(s-1)$-Fläche mit dem Inzidenzvektor $\boldsymbol{f}^{(i)}$ parallel ist. Im folgenden werden die Inzidenzvektoren[11]

$$\begin{aligned} \boldsymbol{f}^{(i_1 \oplus i_2)} &= (\boldsymbol{-1}) + \overline{\boldsymbol{f}}^{(i_1)} + \overline{\boldsymbol{f}}^{(i_2)} \\ \overline{\boldsymbol{f}}^{(i_1 \oplus i_2)} &= \overline{\boldsymbol{f}}^{(i_1)} + \overline{\boldsymbol{f}}^{(i_2)} \end{aligned}$$

betrachtet. Es soll nun gezeigt werden, daß durch

$$y_k'^{(i_1 \oplus i_2)} = \prod_{\{n: f_n^{(i_1 \oplus i_2)} = 0\}} a_{kn} \cdot \min_{\{n: f_n^{(i_1 \oplus i_2)} = 0\}} w_{kn} + \widehat{c}_k^{(1)} \cdot \prod_{\{n: f_n^{(i_1 \oplus i_2)} = 1\}} a_{kn} \cdot \min_{\{n: f_n^{(i_1 \oplus i_2)} = 1\}} w_{kn} \tag{A.13}$$

in der Tat die Zuverlässigkeitswerte für die Decodierung des Codes

$$C^{(i_1 \oplus i_2)} = \{\boldsymbol{c}^{(i_1 \oplus i_2)} : \boldsymbol{c}^{(i_1 \oplus i_2)} = \boldsymbol{c}^{(i_1)} + \boldsymbol{c}^{(i_2)}, \boldsymbol{c}^{(i_1)} \in C^{(i_1)}, \boldsymbol{c}^{(i_2)} \in C^{(i_2)}\}$$

bestimmt werden können. Ohne Beschränkung der Allgemeinheit wird angenommen, daß $\boldsymbol{f}^{(i_1)}$ durch die Funktion $f_n^{(i_1)} = x_{n1}$ sowie $\boldsymbol{f}^{(i_2)}$ durch $f_n^{(i_2)} = x_{n2}$ bestimmt ist. Damit ist $\boldsymbol{f}^{(i_1 \oplus i_2)}$ der Inzidenzvektor der $(s-1)$-Fläche

$$(-1) + (-1 + x_{n1}) + (-1 + x_{n2}) = (-1) + x_{n1} + x_{n2}.$$

In $\mathrm{EG}(s, 2)$ ist die Dimension einer Schnittfläche zwischen einer beliebigen $(s-1)$-Fläche und einer μ-Fläche mit $\mu \geq 2$ immer größer oder gleich eins, und sie besteht damit aus einer geraden Anzahl von Punkten. Damit beeinflussen die $c_k^{(i)}$ mit $\mathrm{w}_2(i-1) > 1$ die Berechnung der $y_k'^{(i_1 \oplus i_2)}$ nicht. Durch die obigen Annahmen bestimmen sich die Komponenten des Inzidenzvektor $\boldsymbol{\beta}^{(L-i_1)}$ zu $\beta_n^{(L-i_1)} = \prod_{m \neq 1} x_{nm}$ und die Komponenten von $\boldsymbol{\beta}^{(L-i_2)}$ zu $\beta_n^{(L-i_2)} = \prod_{m \neq 2} x_{nm}$. Damit ergibt sich

$$\begin{aligned} f_n^{(i_1 \oplus i_2)} \cdot \beta_n^{(L-i_1)} &= [(-1) + x_{n1} + x_{n2}] \cdot x_{n2} x_{n3} \cdots x_{ns} \\ &= \prod_m x_{nm}, \end{aligned}$$

d.h. die Linie $\boldsymbol{\beta}^{(L-i_1)}$ (und dies gilt auch für die Linie $\boldsymbol{\beta}^{(L-i_2)}$) schneidet die durch $\boldsymbol{f}^{(i_1 \oplus i_2)}$ gegebene $(s-1)$-Fläche nur im Punkt $\boldsymbol{x}_L$. Für alle weiteren Linien $\beta_n^{(L-i')} = \prod_{m \neq m'} x_{nm}$, $m' \notin \{1, 2\}$ gilt jedoch

$$\begin{aligned} f_n^{(i_1 \oplus i_2)} \cdot \beta_n^{(L-i')} &= [(-1) + x_{n1} + x_{n2}] \cdot \prod_{m \neq m'} x_{nm} \\ &= \prod_{m \neq m'} x_{nm} \end{aligned}$$

[10] Rechnung in $\mathrm{GF}(2)$

[11] Wie schon in Abschnitt 5.5 wird mit dem Zeichen $(i_1 \oplus i_2)$ nicht eine Rechnenvorschrift im Sinne einer Exor-Operation bezeichnet, sondern es wird ein neues Symbol eingeführt.

und sie sind daher in der $(s-1)$-Fläche enthalten. Da weiterhin gilt

$$[(-1+x_{n1})+(-1+x_{n2})]\cdot\left(\prod_{m\neq 1}x_{nm}+\prod_{m\neq 2}x_{nm}\right)=\prod_{m\neq 1}x_{nm}+\prod_{m\neq 2}x_{nm},$$

ist schließlich die zum Inzidenzvektor $\boldsymbol{\beta}'=\boldsymbol{\beta}^{(L-i_1)}+\boldsymbol{\beta}^{(L-i_2)}$ gehörende Linie in $\overline{\boldsymbol{f}}^{(i_1\oplus i_2)}$ enthalten. Für beliebige i_1,i_2, $\mathrm{w}_2(i_1-1)=\mathrm{w}_2(i_2-1)=1$ ist damit gezeigt, daß Gl. (A.13) richtig ist, wenn die ursprüngliche Codierungsvorschrift $\boldsymbol{c}_k=(c_k^{(1)},c_k^{(2)},\ldots,c_k^{(2^s)})\cdot\boldsymbol{B}_{2^s}$ verglichen wird mit der äquivalenten Regel

$$\boldsymbol{c}_k=(c_k^{(1)},\ldots,c_k^{(i_1-1)},c_k^{(i_1\oplus i_2)},c^{(i_1+1)},\ldots,c_k^{(2^s)})\cdot\boldsymbol{B}'_{2^s}$$

mit der Matrix $\boldsymbol{B}'_{2^s}$, die aus $\boldsymbol{B}_{2^s}$ durch Addition von $\boldsymbol{\beta}^{(L-i_1)}$ zur i_2-ten Zeile hervorgeht. Verallgemeinert man dieses Ergebnis, so folgt

Satz A.2 Für eine beliebige Menge $\{i_1,i_2,\ldots,i_q\}\subseteq\{i:\mathrm{w}_2(i-1)=1\}$ sei

$$\boldsymbol{f}^{(i_1\oplus i_2\oplus\ldots\oplus i_q)}=(\boldsymbol{-1})+\overline{\boldsymbol{f}}^{(i_1)}+\ldots+\overline{\boldsymbol{f}}^{(i_q)},$$

so sind die Zuverlässigkeitswerte für die Decodierung des Summencodes

$$C^{(i_1\oplus i_2\ldots\oplus i_q)}=\{\boldsymbol{c}^{(i_1\oplus i_2\ldots)}:\boldsymbol{c}^{(i_1\oplus i_2\ldots)}=\boldsymbol{c}^{(i_1)}+\ldots+\boldsymbol{c}^{(i_q)},\boldsymbol{c}^{(i_1)}\in C^{(i_1)},\boldsymbol{c}^{(i_2)}\in C^{(i_2)},\ldots\}$$

gegeben durch

$$y_k'^{(i_1\oplus i_2\oplus\ldots)}=\prod_{\{n:f_n^{(i_1\oplus i_2\ldots)}=0\}}a_{kn}\cdot\min_{\{n:f_n^{(i_1\oplus i_2\ldots)}=0\}}w_{kn}+\widehat{c}_k^{(1)}\cdot\prod_{\{n:f_n^{(i_1\oplus i_2\ldots)}=1\}}a_{kn}\cdot\min_{\{n:f_n^{(i_1\oplus i_2\ldots)}=1\}}w_{kn}$$

□

Insgesamt gibt es

$$\sum_{j=1}^{s}\binom{s}{j}=2^s-1$$

verschiedene Untermengen $\{i_1,i_2,\ldots,i_q\}\subseteq\{i:\mathrm{w}_2(i-1)=1\}$ und auch 2^s-1 verschiedene $(s-1)$-Flächen, die durch den Punkt $\boldsymbol{x}_L$ gehen. Damit korrespondiert zu jeder Menge $\{i_1,\ldots,i_q\}$ eine bestimmte $(s-1)$-Fläche und umgekehrt. Aus der gleichen Argumentation wie im Beweis zu Theorem 5.1 folgt nun, daß mit $I=\{i_1,i_2,\ldots,i_Q\}=\{i:\mathrm{w}_2(i-1)=1\}$ der Vektor der Länge 2^s-1

$$(a_k'^{(i_1)},a_k'^{(i_2)},a_k'^{(i_1\oplus i_2)},a_k'^{(i_3)},\ldots,a_k'^{(i_1\oplus i_2\oplus\ldots\oplus i_Q)}),$$

$a_k'^{(.)}=\operatorname{sign}y_k'^{(.)}$ genauso wie der Vektor

$$(c_k^{(i_1)},c_k^{(i_2)},c_k^{(i_1\oplus i_2)},c_k^{(i_3)},\ldots,c_k^{(i_1\oplus i_2\oplus\ldots\oplus i_Q)})$$

ein Codewort des Codes G mit der Generatormatrix

$$\boldsymbol{G} = \begin{bmatrix} -1 & 1 & -1 & \dots & 1 & -1 \\ 1 & -1 & -1 & \dots & -1 & -1 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 1 & 1 & 1 & \dots & -1 & -1 \end{bmatrix}$$

ist. Es handelt sich bei G um einen Equal-Weight-Code mit Hammingdistanz $d_{\mathrm{H,min}}(G) = 2^{s-1}$. Daraus ergibt sich schließlich, nachdem die verschiedenen Codes und Inzidenzvektoren von 1 bis $2^s - 1$ durchnummeriert werden, Satz 5.6.

Abkürzungen und Variablen

Verwendete Abkürzungen

AWGN	Additive White Gaussian Noise (additiver weißer gaußscher Rausch-)
BER	Bit Error Rate (Bitfehlerrate)
BMD	Bounded Minimum Distance (begrenzte Mindestdistanz)
BPSK	Binary Phase Shift Keying
BSC	Binary Symmetric Channel (binärer symmetrischer Kanal)
EG	Euklidische Geometrie
Est.	Estimate (Schätzung)
FFT	Fast FOURIER-Transformation (schnelle FOURIER-Transformation)
FLRAC	Fester linear Raten-angepaßter Code
GA	Gruppe der affinen Transformationen
GC	Generalized Concatenated (verallgemeinert verkettet)
GL	Gruppe der linearen Transformationen
GF	GALOIS-Feld
HDML	Hard-Decision-Maximum-Likelihood (maximale Wahrscheinlichkeit bei harter Entscheidung)
HRAC	per Hand Raten-angepaßter Code
LB	Lower Bound (untere Grenze)
LRAC	Linear Raten-angepaßter Code
ML	Maximum-Likelihood (maximale Wahrscheinlichkeit)
MLC	Multilevel-Code
MSD	Multistage Decoding (Mehrstufendecodierung)

OCBM	Optimierter Code für bitweise Mehrstufendecodierung
QPSK	Quaternary Phase Shift Keying
RAC	Raten-angepaßter Code
RM	REED-MULLER
SD	Soft-Decision ('weiche' Entscheidung)
SDML	Soft-Decision-Maximum-Likelihood (maximale Wahrscheinlichkeit bei weicher Entscheidung)
SNR	Signal to Noise Ratio (Signal-zu-Rausch-Verhältnis)
WER	Word Error Rate (Wortfehlerrate)

Verwendete lateinische Formelzeichen

a	Skalar a
$\boldsymbol{a}$	Vektor $\boldsymbol{a}$
$\boldsymbol{A}$	Matrix $\boldsymbol{A}$
$\boldsymbol{B}$	Generatormatrix des inneren Codes
C	Code (Menge der Codeworte)
$C^{(i)}$	i-ter äußere Code
$\mathcal{C}$	Kanalkapazität
$\mathcal{C}^{(i)}$	Kanalkapazität des i-ten äquivalenten Kanals
c_k	Codesymbol des Codevektors $\boldsymbol{c}$
$\boldsymbol{c}$	Codewort
$\widehat{\boldsymbol{c}}$	Decodierentscheidung für $\boldsymbol{c}$
D_{H}	minimale Hammingdistanz des Codes
$d_{\mathrm{E}}^2(.)$	quadratische euklidische Distanz
d_{GV}	GILBERT-VARSHAMOV-Distanz
$d_{\mathrm{H}}(.)$	Hammingdistanz
$d_{\mathrm{H,min}}(.)$	Mindest-Hammingdistanz
e	Anzahl der Kanalfehler beim BSC

E_b	Empfangsenergie pro Informationsbit
E_S	Empfangsenergie pro Sendesymbol
g_k	vorzeichenbehafteter Zuverlässigkeitswert des Empfangswertes y_k
h_k	vorzeichenbehafteter Zuverlässigkeitswert des Empfangswertes y_k
$h_k^{(u)}$	vorzeichenbehafteter Zuverlässigkeitswert für U
$h_k^{(v)}$	vorzeichenbehafteter Zuverlässigkeitswert für V
I_S	Informationsmenge
$I(X;Y)$	Transinformation zwischen X und Y
i_k	Informationssymbol
K	Anzahl der Informationssymbole im Codewort
$K^{(i)}$	Anzahl der Informationssymbole des i-ten äußeren Codes
L	Listengröße
$L_\tau^{(ML)}(\boldsymbol{c},\boldsymbol{y})$	Listengröße bei Genie-Aided Decodierung
$\overline{L}_\tau^{(ML)}$	mittlere Listengröße bei Genie-Aided Decodierung
$\mathcal{L}$	eine Liste bei der Listendecodierung
$\mathbb{N}$	Raum der natürlichen Zahlen
N	Codelänge
$N^{(i)}$	Länge des i-ten äußeren Codes
N_0	Spektrale Rauschleistungsdichte (einseitig)
$\boldsymbol{n}$	Störungs- oder Rauschvektor
n_p	Anzahl der Permutationen bei Permutationsdecodierung
$\mathcal{O}(f(n))$	Ordnung, d.h. der Ausdruck verhält sich für $n \to \infty$ wie die Funktion $a \cdot f(n)$
$\mathbb{R}$	Raum der reellen Zahlen
$\mathbb{R}^+$	Raum der nicht negativen reellen Zahlen
R	Coderate
R_comp	Cutoff-Rate
$R_\mathrm{comp}^{(i)}$	Cutoff-Rate des i-ten äquivalenten Kanals
$\overline{R}_\mathrm{comp}$	mittlere Cutoff-Rate

$\mathrm{RM}(r,m)$	REED-MULLER Code der Länge 2^m und Ordnung r
SNR_u	äquivalentes SNR des äquivalenten Kanals zur Übertragung von U
SNR_v	äquivalentes SNR des äquivalenten Kanals zur Übertragung von V
$\mathrm{SNR}^{(i)}$	äquivalentes SNR des äquivalenten Kanals zur Übertragung von $C^{(i)}$
U	äußerer Code bei Zerlegung in zwei äußere Codes
$\boldsymbol{u}$	Codewort des Codes U
$\widehat{\boldsymbol{u}}$	Decodierentscheidung für $\boldsymbol{u}$
$\widetilde{u}_k$	aus $\boldsymbol{y}$ (ohne Fehlerkorrektur) geschätztes Symbol u_k
V	äußerer Code bei Zerlegung in zwei äußere Codes
$\boldsymbol{v}$	Codewort des Codes V
$\widehat{\boldsymbol{v}}$	Decodierentscheidung für $\boldsymbol{v}$
$\widetilde{v}_k$	aus $\boldsymbol{y}$ (ohne Fehlerkorrektur) geschätztes Symbol v_k
$w_{\mathrm{H}}(.)$	Hamminggewicht
w_k	Zuverlässigkeitswert des Empfangswertes y_k (ohne Vorzeichen)
$w_k^{(u)}$	Zuverlässigkeitswert für die Schätzung $\widetilde{u}_k$
$w_k^{(v)}$	Zuverlässigkeitswert für die Schätzung $\widetilde{v}_k$
y_k	Empfangskomponente des Vektors $\boldsymbol{y}$
$y_k^{(u)}$	vorzeichenbehafteter Zuverlässigkeitswert für U
$y_k^{(v)}$	vorzeichenbehafteter Zuverlässigkeitswert für V
$\boldsymbol{y}$	Empfangsvektor

Verwendete griechische Formelzeichen

$\Lambda_h(.)$	wahrscheinlichkeitsbasierte Metrik für die Listendecodierung
$\Lambda_y(.)$	distanzbasierte Metrik für die Listendecodierung
$\lambda_h(.)$	wahrscheinlichkeitsbasierte Metrik für die sequentielle Decodierung
$\lambda_y(.)$	distanzbasierte Metrik für die sequentielle Decodierung
$\Pi(.)$	Permutation
ρ	Zerlegungsstufe bei rekursiver Zerlegung
σ_0^2	Rauschvarianz für gaußverteiltes Rauschen

Operatoren

$\boldsymbol{x}^T$	Transponierte des Vektors $\boldsymbol{x}$
$\|\|\boldsymbol{n}\|\|$	Norm des Vektors $\boldsymbol{n}$ (s. Fußnote 4 auf S. 9)
$\|M\|$	Mächtigkeit der Menge M (Anzahl der Elemente in M)
$\|\boldsymbol{a}\|\boldsymbol{b}\|$	Vektor, dessen vorderer Teil aus $\boldsymbol{a}$ und dessen hinterer Teil aus $\boldsymbol{b}$ besteht
$\oplus$	Exor-Addition, Addition in GF(2)
$E\{.\}$	Erwartungswert
$\text{sign}\,(.)$	Signum-Funktion (Vorzeichen)

Literaturverzeichnis

[1] L. E. Aguado and P. G. Farrell: On Hybrid Stack Decoding Algorithms for Block Codes, *IEEE Trans. on Inform. Theory*, Vol. 44, No. 1, pp. 398-409, January 1998

[2] Elwyn R. Berlekamp: The Technology of Error-Correcting Codes, *Proceedings of the IEEE*, Vol. 68, No. 5, pp. 564-593, May 1980

[3] C. Berrou, A. Glavieux and P. Thitimajshima: Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes, *Proc. of ICC '93*, Geneva, Switzerland, pp. 1064-1070, May 1993

[4] Richard E. Blahut: *Theory and Practice of Error Control Codes*, Addison-Wesley Publishing Company, Reading, Massachusetts, USA, 1993

[5] E. L. Blokh and V. V. Zyablov: Coding of Generalized Concatenated Codes, *Problemy Peredachi Informatsii*, Vol. 10, No. 3, pp. 45-50, July-September 1974

[6] Slim Chaoui and Ingmar Land: Comparison of Convolutional Coupled Codes and Partially Systematic Turbo Codes for Medium Code Lengths, accepted at the *4th International ITG Conference on Source and Channel Coding*, Berlin, Germany, January 28-30, 2002

[7] S. Dolinar, D. Divsalar and F. Pollara: Code Performance as a Function of Block Size, *The Telecommuncations and Mission Operations Progress Report* 42-133, January-March 1998, NASA Code 315-91-20-20-53, May 15, 1998

[8] Illya Dumer: Recursive decoding of Reed-Muller codes, *Proceedings of 37th Annual Allerton Conference on Communication, Control and Computing*, Monticello IL, U.S.A., 22-24 Sept. 1999, pp. 61-69

[9] Illya Dumer and Rafail Kichevskiy: Soft-Decision Majority Decoding of Reed-Muller Codes, *IEEE Trans. on Inform. Theory*, Vol. 46, No. 1, pp. 258-264, January 2000

[10] Illya Dumer and Kirill Shabunov: Recursive Decoding of Reed-Muller Codes, *Proceedings of the International Symposium on Information Theory 2000*, Sorrento, Italy, page 63, June 2000

[11] Illya Dumer and Kirill Shabunov: Recursive Construction and their maximum likelyhood decoding, *Proceedings of 38th Annual Allerton Conference on Communication, Control and Computing*, Monticello IL, U.S.A., 4-6 Oct. 2000, pp. 71-81

[12] Illya Dumer and Kirill Shabunov: Near optimum Decoding for Subcodes of Reed-Muller Codes, *Proceedings of the International Symposium on Information Theory 2001,* Washington DC, USA, 24-29 June 2001

[13] Robert M. Fano: A Heuristic Discussion of Probabilistic Decoding, *IEEE Transactions on Inform. Theory*, Vol. IT-9, pp. 64-74, April 1963

[14] G. David Forney Jr.: Exponential Error Bounds for Erasure, List and Decision Feedback Schemes, *IEEE Trans. on Information Theory*, Vol. IT-14, No. 2, pp. 206-230, March 1968

[15] G. D. Forney Jr.: The Viterbi Algorithm, *Proceedings of the IEEE*, 61(3), pp. 268-278, March 1973

[16] G. D. Forney Jr.: Coset Codes - Part II: Binary Lattices and Related Codes, *IEEE Trans. Inform. Theory*, Vol. 34, No. 5, pp. 1152-1187, September 1988

[17] Marc Fossorier: Near Maximum-Likelihood Decoding of Low-Density Parity Check Codes, *International Symposium on Information Theory and Its Applications*, Honolulu, Hawaii, U.S.A, pp. 184-187, November 2000

[18] T. Fujiwara, H. Yamamoto, T. Kasami, Shu Lin: A Trellis-Based Recursive Maximum-Likelihood Decoding Algorithm for Binary Linear Block Codes, *IEEE Trans. Inform. Theory*, Vol. 44, No. 2, pp. 714-729, March 1998

[19] Bernd Friedrichs: *Kanalcodierung*, Springer-Verlag, Berlin, 1996

[20] R. G. Gallager: A Simple Derivation of the Coding Theorem and Some Applications, *IEEE Trans. Inform. Theory,* Vol. IT-11, No. 1. pp. 3-18, January 1965

[21] I. S. Gradshteyn and I. M. Ryzhik: *Table of Integrals, Series and Products*, Academic Press, San Diego, 1980

[22] Farhad Hemmati: Closest Coset Decoding of $|u|u+v|$ Codes, *IEEE Journal on Selected Areas in Communications*, Vol. 7, No 6, pp. 982-988, August 1989.

[23] Y. Han and C. Hartmann and C. C. Chen: Efficient Priority-First Search Maximum-Likelihood Soft-Decision Decoding of Linear Block Codes, *IEEE Trans. Inform. Theory*, Vol. IT-39, pp. 1514 - 1523, 1993

[24] P. Höher, I. Land and U. Sorger: Log-Likelihood Values and Monte Carlo Simulation - Some Fundamental Results, *Proceedings International Symposium on Turbo Codes and Ralated Topics*, Brest, France, pp. 43-46, Sept. 2000

[25] Brian Hughes: On the Error Probability of Signals in Additive White Gaussian Noise, *IEEE Trans. Inform. Theory*, Vol. 37, No. 1, pp. 151-155, January 1991

[26] Simon Hüttinger and Johannes Huber: Hierarchical near-EEP Codedesign for improved decoding performance, *Winter School on Coding and Information Theory 2000*, University of Ulm, Reisenburg, Germany, December 17-20, 2000

[27] Irwin Mark Jacobs and Elwyn R. Berlekamp: A Lower Bound to the Distribution of Computation for Sequential Decoding, *IEEE Trans. Inform. Theory,* Vol. IT-13, No. 2, pp. 167-174, April 1967

[28] Jelinek, F.: A fast sequential decoding algorithm using a stack, *IBM J. Res. Dev*, Vol. 13, pp. 675-685, 1969,

[29] G. A. Kabatyansky: On decoding Reed-Muller codes in semicontinuous channel, *Proc. Second International Workshop Algebraic and Combinatorial Coding Theory*, Leningrad, USSR, pp. 87-91, 1990

[30] Tadao Kasami, Nobuki Tokura: On the Weight Structure of Reed-Muller Codes, *IEEE Trans. on Inform. Theory*, Vol. IT-16, No. 6, pp. 752-759, Nov. 1970

[31] T. Kasami, N. Tokura, S. Azumi: *On the weight distribution of Reed-Muller Codes*, (in Japanese), Faculty Elec. Eng., Osaka Univ., Osaka, Japan 1971

[32] Ulrich Kreßel: *Informationstheoretische Beurteilung digitaler Übertragungsverfahren mit Hilfe des Fehlerexponenten*, Dissertation, VDI Verlag, Düsseldorf, Reihe 10 Nr. 121, 1989

[33] R. Lucas and M. Bossert and A. Dammann: Improved Soft-Decision Decoding of Reed-Muller Codes as Generalized Multiple Concatenated Codes, *ITG Fachbericht, Codierung für Quelle, Kanal und Übertragung*, Aachen, Germany, pp. 137-141, 1998

[34] Rainer Lucas: *On Iterative Soft-Decision Decoding of Linear Binary Block Codes*, Ph. D. dissertation, VDI Verlag, Düsseldorf Germany, Reihe 10 Nr. 511, 1997

[35] James L. Massey: Variable-Length Codes and the Fano Metric, *IEEE Trans. on Inform.* Theory, Vol. IT-18, No. 1, pp. 196-198, January 1972

[36] D. E. Muller: Application of Boolean Algebra to Switching Circuit Design and to Error Detection, *IRE Transactions on Electronic Compute*rs, Vol. EC-3, pp. 6-12, Sept. 1954

[37] N. Nilsson: *Principles of Artificial Intelligence*, Springer Verlag, Berlin, 1982

[38] Robert Oestreich: *Untersuchung und Optimierung von multiplen $|u|u+v|$-verketteten Codes bei begrenzter Decodierkomplexität*, Student research project, University of Technology Darmstadt, Department no. 18, Institute for Communications Technology, Germany June 2001

[39] Elke Offer: *Decodierung mit Qualitätsinformation bei verketteten Codiersystemen*, Ph. D. dissertation, VDI Verlag, Düsseldorf Germany, Reihe 10 Nr. 443, 1996

[40] R. Pellizzoni and A. Spalvieri: Binary Multilevel Coset Codes Based on Reed-Muller Codes, *IEEE Transactions on Communications*, Vol. 42, No. 7, pp. 2357-2360, July 1994

[41] J. Persson: *Multilevel coding based on convolutional code*s, Ph. D. dissertation, Lund University, Department of Information Theory, Lund, Sweden, June 1996

[42] V. S. Pless and W. C. Huffman: *Handbook of Coding Theory*, Vol. 1, Elsevier Science B.V., Amsterdamm, 1998

[43] V. S. Pless and W. C. Huffman: *Handbook of Coding Theory*, Vol. 2, Elsevier Science B.V., Amsterdamm, 1998

[44] Morris Plotkin: Binary Codes with Specified Minimum Distance, *IRE Transactions on Inform. Theory*, Vol. 6, pp. 445-450, 1960

[45] Jaime Portugheis: *Generalized Concatenated Codes for M-PSK Modulation*, Ph. D. dissertation, University of Technology Darmstadt [Department no. 19], 1992 [without publisher]

[46] John G. Proakis: *Digital Communications*, 3rd Edition, McGraw-Hill, New York, 1995

[47] Irving S. Reed: A Class of Multiple-Error-Correcting Codes and the Decoding Scheme, *IRE Transactions onf Information Theory*, Vol. PGIT-4, pp. 38-49, Sept. 1954

[48] Dilip V. Sarwate: *Weight Enumeration of Reed-Muller Codes and Cosets*, Ph. D. dissertation, Dep. Elec. Eng., Princeton Univ., Princeton, N.J., Sept. 1973

[49] Gottfried Schnabl and Martin Bossert: Soft-Decision Decoding of Reed-Muller Codes as Generalized Multiple Concatenated Codes, *IEEE Trans. Inform. Theory*, Vol. IT-41, pp. 304-308, 1995

[50] C. E. Shannon: A Mathematical Theory of Communication, *Bell System Technical Journal*, Vol. 27, No. 3-4, pp. 6379-423 and pp. 623-656, July and October 1948

[51] C. E. Shannn: Probability of Error for Optimal Codes in a Gaussian Channel, *Bell Syst. Tech. Journal*, Vol. 38 , pp. 611-656, 1959

[52] V. M. Sidel'nikov and A. S. Pershakov: Decoding of Reed-Muller Codes with a large Number of Errors, *Problemy Peredachi Informatsii*, Vol. 28, No. 3, pp. 80-94, July-September 1992

[53] N. J. A. Sloane and E. R. Berlekamp: Weight enumerator for second-order Reed-Muller codes, *IEEE Trans. Inform. Theory*, Vol. IT-16 No. 6, pp. 745-751, November 1970

[54] Norbert Stolte and Ulrich Sorger and Gunther Sessler: Sequential Stack Decoding of Binary Reed-Muller Codes, *3rd ITG Conference on Source and Channel Coding*, January 2000, Munich, Germany, pp. 63-69

[55] Norbert Stolte and Ulrich Sorger: Soft-Decision Stack Decoding of Bianary Reed-Muller Codes with "Look-Ahead" Technique*, 7th International Workshop on Algebraic an Combinatorial Coding Theory*, 18-24 June 2000, Bansko, Bulgaria, pp. 293-298

[56] Norbert Stolte and Ulrich Sorger: "Look-Ahead" Soft-Decision Decoding of Binary Reed-Muller Codes, *International Symposium on Information Theory and Its Applications*, Honolulu, Hawaii, U.S.A, November 5-8, 2000, pp. 327-330

[57] Norbert Stolte and Ulrich Sorger: Sequential Decoding of Binary Reed-Muller Codes, *International Journal of Electronics and Communications (AEÜ)*, Vol. 54 No. 6, pp. 412-420, 2000

[58] M. Sugino and Y. Ienaga and N. Tokura and T. Kasami: Weight Distribution of (128,64) Reed-Muller code, *IEEE Trans. Inform. Theory*, Vol. IT-17, pp. 627-628, September 1971

[59] T. Sugita and T. Kasami and T. Fujiwara: Weight Distribution of the Third and Fifth Order Reed-Muller Codes of Length 512, *Technical Report NAIST IS-TR96006*, Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara 630-01, Japan, 1996

[60] H. Tokushige, T. Takata, T. Kasami: On the Number of Minimum Weight Codewords of Subcodes of Reed-Muller Codes, *IEICE Trans. on Fundamentals,* Vol. E81-A, No. 10, October 1998, pp. 1990-1997

[61] K. Tomiyasu, M. Maeda, H. Yamamoto, T. Kodame, T. Fujiwara: On Single IC Chip Implementation of a Recursive Maximum-Likelihood Decoding Algorithm for a (64,40) Reed-Muller Subcode, *International Symposium on Information Theory and Its Applications*, Honolulu, Hawaii, U.S.A., pp. 794-797, November 2000

[62] G. Ungerboeck: Channel Coding with Multilevel/Phase Signals, *IEEE Trans. Inform. Theory*, Vol. IT-28, pp. 55-66, 1982

[63] U. Wachsmann: Coded Modulation: Theoretical Concepts and Praktical Design Rules, Ph. D. dissertation, *Berichte aus der Kommunikations- und Informationstechnik*, Band 9, D 29, Shaker Verlag, Aachen 1999

[64] U. Wachsmann, R. F. H. Fischer, J. Huber: Mulitlevel Codes: Theoretical Concepts and Practical Design Rules, *IEEE Transactions on Information Theory*, Vol. 45 No. 5, pp. 1361 - 1391, July 1999

[65] Lei Wei and Honghui Qi: Near-Optimal Limited-Search Detection on ISI/CDMA Channels and Decoding of Long Convolutional Codes, *IEEE Trans. on Inform. Theory*, Vol. 46, No. 4, pp. 1459-1482, July 2000

[66] Stephen B. Wicker and Vijay K. Bhargava (Eds.): *Reed-Solomon Codes and Their Applications*, New York, IEEE Press 1994

[67] F. J. MacWilliams and N. J. A. Sloane: *The Theory of Error-Correcting Codes*, North-Holland, New York, 1983

[68] J.K. Wolf: Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis, *IEEE Trans. Inform. Theory*, IT-24, No. 1, pp. 76-80, Jan. 1978

[69] J. M. Wozencraft: Sequential Decoding for Reliable Communications, Res. Lab. of Electronics, MIT, Cambridge, Mass., Technical Rep. 325, 1957

[70] V. V. Zyablov and S. L. Portnoi: Fast Maximum-Likelyhood Decoding of Reed-Muller Codes, *Problemy Peredachi Informatsii*, Vol. 27, No. 4, pp. 39-50, October-December 1991

[71] K. Sh. Zigangirov: Some sequential decoding procedures, *Probl. Peredachi Informatsii*, Vol. 2, pp. 13-25, 1966

[72] K. Sh. Zigangirov and R. Johannesson: A Trellis Coding Scheme based on Signal Alphabet Splitting, *Prob. Peredachi Informatsii* (Engl. translation), Vol. 28, No. 4, pp. 14-23 Oct.-Dec. 1992

[73] V. A. Zinov'ev: Generalized cascade codes, *Problemy Peredachi Informatsii*, Vol. 12 no. 1, pp. 5-15, 1976

Index