Soft–Decision Decodierung langer Blockcodes mit Informationsmengen

Vom Fachbereich Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung der Würde eines
Doktor–Ingenieurs (Dr.–Ing.)
genehmigte

Dissertation

von

M. E. E. Dulte José de Barros
(geboren am 28. Februar 1965 in Curitiba / Paraná / Brasilien)

Referent:	Prof. Dr.–Ing. Bernhard G. Dorsch
Korreferent:	Prof. Dr.–Ing. Evgueni A. Krouk
Tag der Einreichung:	19. Juni 2000
Tag der mündlichen Prüfung:	18. Juli 2000

D17
Darmstädter Dissertation
Darmstadt 2000

Gewidmet meiner Frau

Jussimara,

meinen Eltern

Irene und **Dante**

und meiner Schwester

Jane

Danksagungen

Die vorliegende Arbeit entstand während meiner Zeit als Stipendiat des DAAD (Deutschen Akademischen Austauschdienstes) am Institut für Nachrichtentechnik der Technischen Universität Darmstadt.

Offensichtlich ist die Art und Weise, wie ein Brasilianer seine Gefühle äußert, ganz unterschiedlich als die eines Deutschen. Hier ist ein Versuch, meine große Dankbarkeit an die Personen und Organisationen auszudrücken, die sicherlich zum Gelingen meiner Dissertation wesentlich beitrugen:

Zunächst geht mein ganz besonderer Dank an Herrn Prof. Dr.–Ing. Bernhard G. Dorsch, der diese Arbeit betreut hat, für die beträchtliche Erweiterung meines Horizontes und seine vielfältige und stimulierende Unterstützung in allen Phasen meiner Promotion. Von ihm habe ich sowohl fachlich als auch menschlich sehr viel gelernt. Bei Herrn Prof. Dr.–Ing. Evgueni Avramovich Krouk von der *St. Petersburg State University of Aerospace Instrumentation* bedanke ich mich sehr herzlich nicht nur für die Bereitschaft, das Korreferat zu übernehmen, sondern auch für die nützlichen Kommentare, die wertvollen theoretischen Einsichten und die zahlreichen, fruchtbaren Ideen während der Entstehung dieser Arbeit.

Sehr dankbar bin ich dem DAAD für die Förderung in Form eines viereinhalbjährigen Stipendiums sowie der CAPES (*Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior*), die die Flugtickets bezahlt hat. Besonders erwähnen möchte ich hier Frau Helga Wahre vom DAAD, die mir bei allen denkbaren Schwierigkeiten bezüglich des Stipendiums vielmals geholfen hat.

Weiterhin gilt mein Dank der Technischen Universität CEFET–PR in Curitiba / Paraná / Brasilien, an der ich seit Oktober 1991 als Hochschullehrer tätig bin, für die Zulassung zu meiner Promotion sowie meinen Kollegen Professoren, die meine Vorlesungen in Brasilien während meines Aufenthaltes in Deutschland übernommen haben.

Für die mühsame Aufgabe der Durchsicht des Manuskriptes an seinem Anfang danke ich Frau Dr. habil. Britta Hufeisen vom Sprachenzentrum, die mir ständig bei meinen sprachlichen Zweifeln zur Seite stand und mit viel Engagement einen sicherlich kryptischen Text auf sprachliche Mängel durchsah.

Viele hilfreiche Ratschläge sowie interessante Fachdiskussionen und die daraus entstandenen Anregungen für diese Arbeit verdanke ich Herrn Dr. Sergei Valentinovich Fedorenko von der *St. Petersburg State University of Aerospace Instrumentation*.

Allen meinen derzeitigen und ehemaligen Kolleginnen und Kollegen am Institut gebührt mein Dank für ihre Hilfsbereitschaft und das hervorragende Arbeitsklima. Besonders verpflichtet bin ich den Herren Dr.–Ing. Ulrich Sorger und Dipl.–Ing. Norbert Stolte, die mir immer zur Verfügung standen, für ihre unschätzbare und geduldige fachliche Unterstützung, vielerlei Hinweise und kritische Anmerkungen auf formale Fehler und dafür, daß ich an ihren umfassenden, ausgezeichneten Fachwissen teilhaben durfte.

Ich verneige mich in tiefer Dankbarkeit vor meinem Freund Dr.–Ing. Stefan Brück, der mir bei allen möglichen Gelegenheiten am Institut stets behilflich war, mit dem ich viele anregende Fachgespräche führte, der das sehr aufwendige Korrekturlesen der Arbeit übernahm und mich in jeder Hinsicht bei ihrer Entstehung unterstützte.

Ein besonderes Dankeschön ergeht an meine geliebte Familie, meine Mutter Irene, meinen Vater Dante und meine Schwester Jane, für das mir in dieser arbeitsreichen Zeit entgegengebrachte Verständnis und dafür, daß sie mich zum Durchhalten ermutigten.

Schließlich bedanke ich mich von ganzem Herzen bei meiner Frau Jussimara, die in diesen letzten Jahren zahllose Stunden auf mich verzichten mußte und mich dennoch immer mit ihrer Geduld und Motivation unterstützt hat.

Darmstadt, im Juli 2000 *Dulte José de Barros*

“Exercise is the beste intrument in learnyng.”

Robert Recorde
(The Whetstone of Witte, 1557)

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

Abkürzung **Seite**

AWGN–Kanal (*Additive White Gaussian Noise Channel*): Additiver weißer gaußscher Rauschkanal 21

BCH-Code: Bose–Ray-Chaudhuri–Hocquenghem–Code 80

BER (*Bit Error Rate*): Bitfehlerrate 36

BMD–Decodierung (*Bounded–Minimum–Distance Decoding*): Begrenzte–Minimalabstand–Decodierung 1

BPSK (*Binary Phase Shift Keying*): Binäre Phasenumtastung 21

DC (*Discrete Channel*): Diskreter Kanal 21

eBCH-Code (*extended Bose–Ray-Chaudhuri–Hocquenghem–Code*): erweiterter Bose–Ray-Chaudhuri–Hocquenghem–Code 80

GF (*Galois Field*): Galoisfeld 8

GMD (*Generalized Minimum Distance*): verallgemeinerter Minimalabstand 44

HDBMD–Decodierung (*Bounded–Minimum–Distance Hard–Decision Decoding*): Hard–Decision–Begrenzte–Minimalabstand–Decodierung 2

HDMDD (*Minimum–Distance Hard–Decision Decoding*): Hard–Decision–Minimalabstand–Decodierung 60

IS (*Information Set*): Informationsmenge 17

LB (*Lower Bound*): untere Schranke 37

MAP–Decodierer (*Maximum–a–posteriori Decoder*): Maximum–a–posteriori–Decodierer 30

MDS-Code: Maximum–Distance–Separable–Code 18

ML–Decodierer (*Maximum–Likelihood Decoder*): Maximum–Likelihood–Decodierer 30

MLD (*Maximum–Likelihood Decoding*): Maximum–Likelihood–Decodierung 2

MRB (*Most Reliable Basis*): zuverlässigste Basis 56

PS (*Parity Set*): Prüfmenge 18

RS-Code: Reed–Solomon–Code 18

SDMLD (*Maximum–Likelihood Soft–Decision Decoding*): Soft–Decision–Maximum–Likelihood–Decodierung 2

SNR (*Signal–to–Noise Ratio*): Signal zu Rauschverhältnis 22

WER (*Word Error Rate*): Wortfehlerrate 36

Symbolverzeichnis

Konstanten:

$\mathbf{0}$	Nullwort
e	Basis der natürlichen Logarithmen 2.718281828459045 ...
$\boldsymbol{I}$	Einheitsmatrix
κ	beliebige Konstante

Operatoren:

$\arg\max\{\cdot\}$	Argument, das die Funktion maximiert
$\max\{\cdot\}$	Maximum
$\min\{\cdot\}$	Minimum
$o(\cdot)$	Landau–Operator „klein o“
$p(\cdot)$	Wahrscheinlichkeitsdichte
$\Pr(\cdot)$	Wahrscheinlichkeit
$\Pr(\cdot \mid \cdot)$	bedingte Wahrscheinlichkeit
$\overline{\Pr}\{\cdot\}$	gesamte Wahrscheinlichkeit einer Menge
$\overline{(\cdot)}$	Komplement einer Menge (Komplementärmenge)
$(\cdot)^{\mathrm{T}}$	transponierte Matrix
$\lvert\cdot\rvert$	Betrag eines Skalars oder Mächtigkeit (Kardinalzahl) einer Menge
$\lfloor\cdot\rfloor$	größte ganze Zahl kleiner oder gleich dem Argument
$\lceil\cdot\rceil$	kleinste ganze Zahl größer oder gleich dem Argument
$\lVert\cdot\rVert$	euklidische Norm eines Vektors
$\lVert\cdot\rVert_\infty$	Maximumnorm eines Vektors
$\langle\cdot,\cdot\rangle$	Skalarprodukt

Funktionen:

$\mathrm{bit}(i,\boldsymbol{\alpha})$	Bit an der i-ten Stelle eines binären Musters $\boldsymbol{\alpha}$ aus $\mathbb{B}^h$ (ist

	äquivalent zu α_i)
$\mathrm{erfc}(\cdot)$	komplementäre Fehlerfunktion
$\exp(\cdot)$	Exponentialfunktion
$f(\cdot), g(\cdot)$	beliebige Funktion
$\mathrm{H}_2(\cdot)$	binäre Entropiefunktion
$\mathrm{lb}(\cdot)$	binärer (dyadischer) Logarithmus
$\ln(\cdot)$	natürlicher (Neperscher) Logarithmus
$\mathrm{sign}(\cdot)$	Signumfunktion (Sprungfunktion), siehe Gleichung (2.38)
$\Gamma(\cdot)$	Gamma–Funktion
$\binom{n}{k}$	Binomialkoeffizient n über k

Mengen:

$\boldsymbol{\mathcal{A}}$	Menge der 2^{n-k} wahrscheinlichsten Fehlervektoren $\boldsymbol{a}$
$\mathcal{B}$	Block, siehe Unterabschnitt 2.2.7
$\mathbb{B}^h$	Menge der 2^h verschiedenen binären Muster der Länge h (ist äquivalent zu $\mathbb{F}_2^h$)
$\mathcal{C}$	Blockcode
$\mathcal{C}^h$	Menge der h besten geschätzten Codewörtern $\hat{\boldsymbol{c}}_1, \hat{\boldsymbol{c}}_2, \ldots, \hat{\boldsymbol{c}}_h$
$\mathcal{D}_0(\cdot,\cdot)$	Indexmenge der übereinstimmenden Stellen, siehe Gleichung (2.41)
$\mathcal{D}_1(\cdot,\cdot)$	Indexmenge der sich unterscheidenden Stellen, siehe Gleichung (2.42)
$\mathcal{D}_{\boldsymbol{\alpha}}$	Indexmenge, siehe Gleichung (3.28)
$\mathcal{E}(r, \boldsymbol{\beta}, \boldsymbol{v})$	diskretes Ellipsoid mit Hammingradius r und Zuverlässigkeitsvektor $\boldsymbol{\beta}$ um einen Vektor $\boldsymbol{v}$
$\mathcal{F}$	Menge von n Elementen, siehe Unterabschnitt 2.2.7
$\mathbb{F}_q$	Galoisfeld (endlicher Zahlenkörper) mit q Elementen
$\mathbb{F}_q^n$	Menge der n-dimensionalen (Zeilen-)Vektoren bzw. Blöcke mit Komponenten aus $\mathbb{F}_q$
$\mathbb{F}_q^{k,n}$	Menge der (k,n)-dimensionalen Matrizen mit Elementen aus $\mathbb{F}_q$
$\mathcal{I}$	Informationsmenge
$\mathcal{J}(\cdot,\cdot), \mathcal{J}'(\cdot,\cdot)$	beliebige Indexmenge
$\mathcal{J}(\cdot,\cdot)^{(\delta)}$	Indexmenge der δ unzuverlässigsten Stellen von $\mathcal{J}(\cdot,\cdot)$, siehe Unterabschnitt 2.4.2
$\boldsymbol{\mathcal{K}}$	Menge von Blöcken $\mathcal{B}$, siehe Unterabschnitt 2.2.7
$\mathcal{L}$	geordnete Menge von Stellen, siehe Gleichung (4.12)

$\boldsymbol{m}$	Menge von 2^h beliebigen Kardinalzahlen $m_{\boldsymbol{\alpha}}$
$\boldsymbol{\mathcal{M}}$	Menge der Nebenklassenführer (Restklassenführer oder *Coset Leader*)
$\mathbb{M}$	Menge derjenigen Mengen $\boldsymbol{m}$ von 2^h Kardinalzahlen $m_{\boldsymbol{\alpha}}(\hat{\boldsymbol{x}}, \boldsymbol{y})$, die Gleichung (3.39) erfüllen
$\mathbb{M}_{\min}$	Menge minimaler Mengen $\boldsymbol{m}$ in $\mathbb{M}$ von 2^h Kardinalzahlen $m_{\boldsymbol{\alpha}}(\hat{\boldsymbol{x}}, \boldsymbol{y})$
$\mathcal{N}$	Nebenklasse (Restklasse oder *Coset*)
$\mathbb{N}$	Menge der natürlichen Zahlen
$\mathbb{N}^+$	Menge der natürlichen Zahlen ohne Null
$\mathcal{P}$	Prüfmenge
$\mathcal{Q}$	Indexmenge, siehe Gleichung (3.48)
$\mathcal{R}_C(\cdot)$	Decodierbereich innerhalb eines Hyperkreiskegels
$\mathcal{R}_G(\cdot)$	GMD–Decodierbereich
$\mathcal{R}_S(\cdot)$	Decodierbereich innerhalb einer Hyperkugel
$\mathbb{R}$	unendlicher Körper der reellen Zahlen
$\mathbb{R}^+$	unendlicher Körper der positiven reellen Zahlen (ohne Null)
$\mathbb{R}_0^+$	unendlicher Körper der positiven reellen Zahlen mit Null
$\mathbb{R}^n$	euklidischer Vektorraum
$\mathbb{R}^3$	dreidimensionaler euklidischer Vektorraum
$\mathcal{S}(r, \boldsymbol{v})$	Kugel mit Hammingradius r um einen Vektor $\boldsymbol{v}$
T	Teilmenge, siehe Unterabschnitt 2.2.7
$\mathcal{T}(h, \mathcal{W}, \mathcal{X}^h)$	Menge von geschätzten Sendefolgen in $\mathbb{R}^n$, die weit entfernt von den besten, bereits in den vorherigen Iterationen von Ψ getesteten geschätzten Sendefolgen $\hat{\boldsymbol{x}}_{\boldsymbol{1}}, \hat{\boldsymbol{x}}_{\boldsymbol{2}}, \ldots, \hat{\boldsymbol{x}}_{\boldsymbol{h}}$ liegen, entspricht $\mathcal{T}'(h, \mathcal{W}, \mathcal{C}^h)$ in $\mathbb{F}_2^n$
$\mathcal{T}'(h, \mathcal{W}, \mathcal{C}^h)$	Schnittmenge aller h Mengen derjenigen geschätzten Codewörter um $\hat{\boldsymbol{c}}_{\boldsymbol{i}}$ in $\mathbb{F}_2^n$, die einen Hammingabstand d_H zu $\hat{\boldsymbol{c}}_{\boldsymbol{i}}$ größer als w_i aufweisen, entspricht $\mathcal{T}(h, \mathcal{W}, \mathcal{X}^h)$ in $\mathbb{R}^n$
$\mathcal{U}(n, k, t)$	(n, k, t)-Überdeckungsmenge
$\mathcal{V}(\cdot)$	Voronoi–Region
$\mathcal{W}$	Gewichtsprofil
$\mathcal{X}$	modulierter Code
$\mathcal{X}^h$	Menge der h besten geschätzten Sendefolgen $\hat{\boldsymbol{x}}_{\boldsymbol{1}}, \hat{\boldsymbol{x}}_{\boldsymbol{2}}, \ldots, \hat{\boldsymbol{x}}_{\boldsymbol{h}}$
$\mathbb{Z}$	Menge der ganzen Zahlen
$\emptyset$	leere Menge
$\{0, 1\}$	Menge mit den (reellen) Elementen 0 und 1 (entspricht den

	Elementen aus $\mathbb{F}_2$)
$\{0,1\}^n$	Menge der n-dimensionalen (Zeilen-)Vektoren bzw. Blöcke mit Komponenten aus $\{0,1\}$ (entspricht dem Galoisfeld $\mathbb{F}_2^n$)
$\{\pm 1\}$	Menge mit den (reellen) Elementen -1 und $+1$
$\{\pm 1\}^n$	Menge der n-dimensionalen (Zeilen-)Vektoren bzw. Blöcke mit Komponenten aus $\{\pm 1\}$

Symbole aus $\mathbb{F}_q$:

a_i	beliebiger Koeffizient
c_i	Komponente i eines Codewortes $\boldsymbol{c}$
$g_{i,j}$	Element in der Zeile i und Spalte j einer Generatormatrix $\boldsymbol{G}$
u_i	Komponente i eines Infowortes $\boldsymbol{u}$
v_i, v_i'	Komponente i eines Hard–Decision Vektors $\boldsymbol{v}$ bzw. $\boldsymbol{v'}$ oder Komponente i eines beliebigen Vektors $\boldsymbol{v}$ bzw. $\boldsymbol{v'}$
α_i	Bit an der i-ten Stelle eines binären Musters $\boldsymbol{\alpha}$ aus $\mathbb{B}^h$

Symbole aus $\mathbb{N}$:

d_G	Gilbertabstand
$d_H(\cdot,\cdot)$	Hammingabstand
$d_{H\min}$	Minimalabstand
d_V	Varshamovabstand
h	Anzahl der geschätzten Codewörtern $\hat{\boldsymbol{c}}_1, \hat{\boldsymbol{c}}_2, \ldots, \hat{\boldsymbol{c}}_h$ bzw. Sendefolgen $\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \ldots, \hat{\boldsymbol{x}}_h$
i	Index
I	Anzahl der betrachteten Fehlervektoren, siehe Abschnitt 4.1
$\mathcal{I}_{\max}$	maximale Anzahl der gebildeten Informationsmengen $\mathcal{I}$
j	Index
k	Dimension eines Codes $\mathcal{C}$
l	Index
$m_{\boldsymbol{\alpha}}$	beliebige Kardinalzahl
$m_{\boldsymbol{\alpha}}(\cdot,\cdot)$	Kardinalzahl, siehe Gleichung (3.31)
n	Blocklänge eines Codes $\mathcal{C}$
q	Stufenzahl der Eingangs- oder Ausgangsalphabet
t	Packungsradius
$\lvert\mathcal{U}(n,k,t)\rvert$	Mächtigkeit (Kardinalzahl) einer (n,k,t)-Überdeckungsmenge
w_i	i-tes Hamminggewicht innerhalb eines Gewichtsprofils $\mathcal{W}$

$w_H(\cdot)$	Hamminggewicht
w_{Hmin}	Minimalgewicht
ι	Variable, die die Anzahl der gebildeten Informationsmengen $\mathcal{I}$ bezeichnet
ν	Anzahl von ausgewählten Blöcken $\mathcal{B}$ von $n-k$ Stellen
ξ	Anzahl von Codewörtern $\boldsymbol{c}$, die dieselbe Menge von k Komponenten an k beliebig ausgewählten Stellen von $\boldsymbol{c}$ besitzen
ρ	Überdeckungsradius
ϱ	Pivotindex

Symbole aus $\mathbb{Z}$:

δ, δ_i	von dem Gewichtsprofil $\mathcal{W}$ und der Indexmenge $\mathcal{D}_1(\cdot,\cdot)$ abhängige Zahl oder beliebige Zahl

Skalare:

$d_E(\cdot,\cdot)$	euklidischer Abstand
$d_E^2(\cdot,\cdot)$	quadratischer euklidischer Abstand
$d_{E\min}$	euklidischer Minimalabstand
$d_\beta(\cdot,\cdot)$	gewichteter Hammingabstand
$d_\mathcal{E}(\boldsymbol{\beta}, \boldsymbol{v}, \boldsymbol{v}')$	ellipsoidischer Abstand zwischen $\boldsymbol{v}$ und $\boldsymbol{v}'$
E_b	mittlere Energie pro empfangenem Infobit
E_c	mittlere Energie pro empfangenem Codebit
N_0	einseitige Rauschleistungsdichte des Störsignals
p_c	codierte Fehlerwahrscheinlichkeit, siehe Gleichung (5.13)
p_i	Wahrscheinlichkeit einer Stelle i
P_{BMD}	WER einer HDBMD–Decodierung
P_{COR}	WER basiert auf der *Cutoff–Rate* R_0
P_{LB}	untere Simulationsschranke für die WER einer SDMLD
P_{MDD}	WER einer HDMDD
P_{MLD}	WER einer SDMLD
P_u	Fehlerrate einer uncodierten Übertragung
p_{Δ^2}	simulierte BER
P_{Δ^2}	simulierte WER
p_σ	BER in Abhängigkeit von der Varianz σ^2, siehe Gleichung (2.79b)
P_σ	WER in Abhängigkeit von der Varianz σ^2, siehe

	Gleichung (2.79a)
P_Ψ	WER eines Decodierers Ψ
r_i	Komponente i eines Empfangsvektors $\boldsymbol{r}$
R	Rate eines Blockcodes $\mathcal{C}$
R_0	*Computational Cutoff–Rate*
$w_\beta(\cdot)$	gewichtetes Hamminggewicht
x_i, x_i'	Komponente i einer Sendefolge $\boldsymbol{x}$ bzw. $\boldsymbol{x'}$
$\hat{x}_i$	Komponente i einer geschätzten Sendefolge $\hat{\boldsymbol{x}}$
y_i, y_i'	Komponente i eines durch das Log–Likelihood–Wahrscheinlichkeitsverhältnis definierten Vektors $\boldsymbol{y}$ bzw. $\boldsymbol{y'}$ oder Komponente i eines beliebigen Vektors $\boldsymbol{y}$ bzw. $\boldsymbol{y'}$
$\tilde{y}_i$	Komponente i eines normierten Vektors $\tilde{\boldsymbol{y}}$
${y^0}_i$	Komponente i eines Einheitsvektors $\boldsymbol{y}^0$
z_i	Komponente i eines Vektors $\boldsymbol{z}$, der die Hard–Decision Information eines Vektors $\boldsymbol{y}$ zusammenfaßt
z_i'	Komponente i eines Konkurrenz–Vektor $\boldsymbol{z'}$
β_i	Komponente i eines Vektors $\boldsymbol{\beta}$, der die Zuverlässigskeits-information eines Vektors $\boldsymbol{y}$ zusammenfaßt
γ_i	siehe Gleichung (4.10b)
Δ^2	quadratischer Abstand
ϵ	beliebige Zahl $0 < \epsilon \leq 1$, siehe Gleichung (5.11)
η_1	linearer Normierungsfaktor
η_2	quadratischer Normierungsfaktor
η_3	kubischer Normierungsfaktor
λ	Parameter, siehe Gleichung (4.10)
Λ	Normierungsfaktor, siehe Gleichung (2.78)
μ	Erwartungswert
ρ_S	Radius einer Hyperkugel
σ^2	Varianz
ς	Argument einer Funktion
τ	Hilfsvariable
Υ	Korrelationsschwelle
ϕ_i	Zufallsgröße
Φ	Zufallsvariable, siehe Gleichung (2.76)
$\varphi(\cdot,\cdot)$	Winkel zwischen zwei Vektoren
$\varphi_{\max}$	Winkel eines Hyperkreiskegels
$\chi(\cdot, n, \cdot)$	χ^2-Wahrscheinlichkeitsdichte mit dem Freiheitsgrad n, siehe Gleichung (2.80)

Vektoren:

$\boldsymbol{a}, \boldsymbol{a}_i$	Vektor aus der Menge $\boldsymbol{\mathcal{A}}$ der 2^{n-k} wahrscheinlichsten Fehlervektoren
$\boldsymbol{c}, \boldsymbol{c}', \boldsymbol{c}_i$	Codewort
$\hat{\boldsymbol{c}}, \hat{\boldsymbol{c}}_i$	geschätztes Codewort
$\boldsymbol{c}'(\mathcal{I})$	von der Informationsmenge $\mathcal{I}$ abhängiges Codewort
$\boldsymbol{c}_{ML}$	wahrscheinlichst gesendetes Codewort
$\boldsymbol{g}_i$	Zeilenvektor in der Zeile i einer Generatormatrix $\boldsymbol{G}$
$\boldsymbol{r}$	Empfangsvektor
$\boldsymbol{s}$	Syndrom
$\boldsymbol{u}$	Infowort
$\hat{\boldsymbol{u}}$	geschätztes Infowort
$\boldsymbol{u}_{[\mathcal{I}]}$	durch die Projektion eines Vektors $\boldsymbol{v}$ auf seine Koordinaten in einer Informationsmenge $\mathcal{C}$ bestimmtes Infowort
$\boldsymbol{v}, \boldsymbol{v}', \boldsymbol{v}_i$	Hard–Decision Information des Vektors $\boldsymbol{y}$ in $\mathbb{F}_2^n$ oder beliebiger Vektor in $\mathbb{F}_q^n$
$\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{x}''$	Sendefolge
$\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}_i$	geschätzte Sendefolge
$\hat{\boldsymbol{x}}_{LB}$	durch Simulation geschätzte Sendefolge
$\boldsymbol{x}_{ML}$	wahrscheinlichste Sendefolge
$\boldsymbol{y}, \boldsymbol{y}'$	durch das Log–Likelihood–Wahrscheinlichkeitsverhältnis definierter Vektor oder beliebiger Vektor in $\mathbb{R}^n$
$\tilde{\boldsymbol{y}}$	normierter Vektor
$\boldsymbol{y}^0$	Einheitsvektor
$\boldsymbol{z}$	Hard–Decision Information des Vektors $\boldsymbol{y}$ in $\mathbb{R}^n$
$\boldsymbol{z}'$	Konkurrenz–Vektor in $\mathbb{R}^n$
$\boldsymbol{\alpha}, \boldsymbol{\alpha}'$	binäres Muster aus $\mathbb{B}^h$
$\boldsymbol{\beta}$	Zuverlässigskeitsinformation des Vektors $\boldsymbol{y}$

Matrizen:

$\boldsymbol{G}$	Generatormatrix
$\boldsymbol{G}'(\mathcal{I})$	durch die Informationsmenge $\mathcal{I}$ bestimmte Generatormatrix
$\boldsymbol{H}$	Prüfmatrix
$\boldsymbol{P}$	beliebige Matrix

Sonstiges:

Ξ	beliebige Bedingung
Ξ_ϵ	Suboptimales Akzeptanzkriterium basierend auf zwei Codewörtern, siehe Gleichung (5.11)
Ξ_{GMD}	Bedingung eines GMD–Tests
$\Xi_{\text{Hyperkreiskegel}}$	Bedingung eines Hyperkreiskegel–Tests
$\Xi_{\text{Hyperkugel}}$	Bedingung eines Hyperkugel–Tests
Ξ_{Kasami}	Akzeptanzkriterium nach Kasami et al. basierend auf mehreren Codewörtern
$\Xi_{\text{Kasami}(\hat{\boldsymbol{x}}_\mathbf{1},\hat{\boldsymbol{x}}_\mathbf{2})}$	Akzeptanzkriterium nach Kasami et al. basierend auf zwei Codewörtern
Ξ_{SDMLD}	Bedingung eines Soft–Decision–Maximum–Likelihood–Akzeptanzkriteriums
Ξ_{Syndrom}	Bedingung eines Syndrom–Tests
Ξ_{TP}	Bedingung eines Akzeptanzkriteriums nach Taipale und Pursley
Ψ	beliebiger Decodierer
Ψ_{LB}	spezieller Decodierer in einer Simulation, der eine WER P_{LB} hat
$\overset{\text{exp}}{=}$	gleiche exponentielle Ordnung, siehe Definition (4.4)
$\forall$	Allquantor
$\Longrightarrow$	Implikation
$\Longleftrightarrow$	Äquivalenz

Kapitel 1

Einleitung

Seit Entstehung der Informations- und Codierungstheorie im Jahre 1948 durch die Arbeiten von Shannon [Sha48a, Sha48b] haben die algebraischen Blockcodes und die Blockcodes in euklidischen Vektorräumen zwei unterschiedliche Entwicklungsrichtungen genommen.

Die algebraischen Blockcodes werden häufig mit folgenden Konzepten in Verbindung gebracht [FV96]:

- Symbole aus endlichen Körpern,
- Hammingabstand,
- q-stufige symmetrische Kanäle,
- Hard–Decision,
- Begrenzte–Minimalabstand–Decodierung (BMD–Decodierung, *Bounded–Minimum–Distance Decoding*) und
- algebraische Decodierverfahren.

Im Gegensatz dazu sind die Blockcodes in euklidischen Vektorräumen normalerweise verbunden mit:

- Symbolen aus euklidischen Vektorräumen,
- euklidischem Abstand,
- gaußschen Rauschkanälen,

- Soft–Decision,
- Maximum–Likelihood–Decodierung (MLD, *Maximum–Likelihood Decoding*) und
- nicht algebraischen Decodierverfahren.

Der Vorteil des algebraischen Ansatzes ist, daß dieser die Konstruktion von langen und leistungsfähigen Blockcodes mit komplexen mathematischen Strukturen erlaubt. Diese Blockcodes können dann mit algebraischen Decodieralgorithmen bis zum halben Minimalabstand decodiert werden, wobei die Komplexität einer Decodierung in Abhängigkeit vom Minimalabstand des Codes nur polynomial zunimmt.

Die algebraischen Decodierverfahren haben jedoch Nachteile. Bei gaußschen Kanälen beträgt der Verlust durch Ausnutzung von Hard–Decision anstatt Soft–Decision üblicherweise 2 bis zu asymptotisch 3 dB [Fri95]. Während kurze algebraische Blockcodes normalerweise optimal sind, werden sie allmählich schlechter als zufällig gewählte Codes bei wachsender Blocklänge [Bos98].

Außerdem ist bekannt [FV96], daß eine Hard–Decision–Begrenzte–Minimalabstand–Decodierung (HDBMD–Decodierung, *Bounded–Minimum–Distance Hard–Decision Decoding*) asymptotisch wesentlich schlechter ist als eine Soft–Decision–Maximum–Likelihood–Decodierung (SDMLD, *Maximum–Likelihood Soft–Decision Decoding*). Das hat zur Folge, daß die Kanalkapazität mit einem HDBMD–Decodierprinzip nicht erreicht werden kann [Fri95].

Wegen dieser Nachteile werden heutzutage stets häufiger Blockcodes mittlerer Länge bei gaußschen Kanälen mit SDMLD benutzt. Diese Blockcodes in euklidischen Vektorräumen ermöglichen eine Fehlerkorrektur über den halben Minimalabstand hinaus, und somit könnte die Kanalkapazität theoretisch mit der Anwendung eines sehr langen Blockcodes erreicht werden.

Dagegen steigt die Komplexität einer SDMLD exponentiell. Falls sie für lange Blockcodes verwendet wird, ist sie mit so erheblichem Decodieraufwand verbunden, daß sie undurchführbar wird.

Trotz dieses Nachteiles hat Evseev im Jahre 1983 für Hard–Decision Kanäle bewiesen [Evs83], daß der größte Teil des Decodieraufwandes eingespart werden könnte, wenn lediglich eine näherungsweise optimale Decodierung durchgeführt würde. Dieser wichtige Beitrag war der einführende zu einem ganz neuen Forschungsgebiet, nämlich der Suche nach Prozeduren zur näherungsweisen optimalen Decodierung mit geringer Komplexität.

Das Ergebnis von Evseev hat Dumer später für Soft–Decision Känale verallgemeinert [Dum96b]. Kurz danach hat er in einer Reihe von Artikeln [Dum96a, Dum97a, Dum97b, Dum98a, Dum98b, Dum98c, Dum] einen probabilistischen Algorithmus vorgeschlagen, der bei näherungsweiser SDMLD die gewünschte Einsparung der Decodierkomplexität im asymptotischen Fall ermöglicht.

1.1 Motivation

Ziel der vorliegenden Arbeit ist, ein allgemeines Soft–Decision Decodierverfahren vorzustellen, das geeignet ist, einen beliebigen linearen Blockcode weit über den halben Minimalabstand hinaus mit einer akzeptablen Komplexität zu decodieren.

Durch einen leicht änderbaren Parameter wird der Aufwand des Decodierverfahrens gesteuert, das sich schrittweise in wiederholten Iterationen demselben Ergebnis wie dem einer SDMLD nähert. Dieser Parameter begrenzt die maximale Anzahl der Iterationen, die durchgeführt werden können, und somit die gesamte Decodierkomplexität der Methode bzw. die Güte der Decodierung im Vergleich zu einer SDMLD. Wird ein großer Aufwand erlaubt, so wird beinahe eine SDMLD durchgeführt und umgekehrt.

Als Ausgangspunkt für den Entwurf dieses Algorithmus waren die obenerwähnten theoretischen Überlegungen von Dumer im asymptotischen Fall. Einer der wesentlichsten Beiträge dieser Arbeit ist zu zeigen, wie der probabilistische Ansatz von Dumer nicht nur im asymptotischen Fall, sondern auch für konkrete Codes (d. h. Codes endlicher Länge) benutzt werden kann.

Darüber hinaus werden einige allgemeine Bedingungen, sogenannte Akzeptanzkriterien hergeleitet, die es erlauben, daß eine iterative Decodierung möglicherweise früher abgebrochen werden kann. Genaugenommen sind diese Bedingungen unabhängig von der Decodiermethode und somit in anderen denkbaren Decodieralgorithmen anwendbar. Sie erweisen sich als sehr effizient bei der Decodierung kurzer Codes. Insbesondere bei langen und komplexen Codes werden diese Bedingungen hingegen sehr streng.

Deshalb wird eine Lösung zur Bewältigung dieses Nachteiles vorgeschlagen und genauer untersucht, die eine beträchtliche Verminderung der Anzahl der Iterationen für lange Codes ermöglicht, ohne die Leistungsfähigkeit des Decodieralgorithmus in der Praxis sichtbar zu beeinträchtigen. Damit kann die durchschnittliche Anzahl der Iterationen für lange Codes drastisch ver-

ringert werden.

Da ein Mangel an zuverlässigen Ergebnissen zu Vergleichszwecken bei den Simulationen langer Codes bisher vorhanden ist, wurden einige Referenzkurven für solche Codes in dieser Arbeit dargestellt, die erstmals so präzis erzeugt wurden.

1.2 Gliederung der Arbeit

Im einzelnen wird der Aufbau dieser Arbeit folgendermaßen gegliedert:

- Das nächste Kapitel fängt mit einer mathematischen Darstellung aller Begriffe der Kanalcodierung an, die für das Verständnis des weiteren Verlaufes der Arbeit notwendig sind. Danach werden sowohl die wesentlichsten Grundlagen für eine Decodierung mithilfe von Informationsmengen vermittelt als auch das in den Simulationen von Kapitel 5 verwendete Übertragungssystem vorgestellt. Der darauffolgende Abschnitt befaßt sich mit Zuverlässigkeitsmaßen für Soft–Decision. Dann wird eine SDMLD kurz diskutiert und mittels des Begriffes der Voronoi–Regionen geometrisch interpretiert. Es folgt eine Erläuterung von Zufallsvariablen, die als ein anderer Beitrag dieser Arbeit durch extrem schnelle Zufallszahlengeneratoren implementiert wurden. Dabei werden einerseits gleichverteilte Zufallszahlengeneratoren, die im Decodieralgorithmus von Kapitel 4 nötig sind, und andererseits gaußverteilte Zufallszahlengeneratoren, die für die Simulationen von Kapitel 5 benutzt werden, in Betracht gezogen. Ferner wird eine Methode im vorletzten Abschnitt vorgeschlagen, die für Simulationen langer Codes bei kleinen Fehlerwahrscheinlichkeiten geeignet ist. Schließlich wird eine untere Simulationsschranke für SDMLD angegeben.

- Im ersten Abschnitt von Kapitel 3 werden zunächst die wichtigsten Akzeptanzkriterien auf Basis eines Codewortes hergeleitet. Die umfassendsten Herleitungen werden im Anhang durchgeführt. Im zweiten Abschnitt wird eine vereinheitliche Notation bereitgestellt, die die Beschreibung eines allgemeinen Akzeptanzkriteriums basierend auf mehreren Codewörtern ermöglicht. Daraus wird anstelle einer sehr aufwendigen Herleitung eine algorithmische Darstellung eines Akzeptanzkriteriums gezeigt, das auf zwei Codewörtern beruht. Danach werden die

Akzeptanzkriterien auf Basis von mehr als zwei Codewörtern kurz beschrieben. Im Abschluß des Kapitels werden weitere Akzeptanzkriterien diskutiert.

- In Kapitel 4 wird auf das Hauptproblem der aufwandsgünstigen Decodierung in Soft–Decision Kanälen eingegangen. Zuerst wird das Ergebnis von Evseev [Evs83] erörtert sowie einige probabilistische Hard–Decision Decodierverfahren präsentiert. Nachher wird die Verallgemeinerung der Ergebnisse für Soft–Decision Kanäle durch den Beitrag von Dumer [Dum96b] erwähnt und nicht zuletzt sein probabilistischer Algorithmus [Dum96a, Dum97a, Dum97b, Dum98a, Dum98b, Dum98c, Dum] genauer berücksichtigt. In Abschnitt 4.4 wird schließlich die allgemeine iterative Soft–Decision Decodiermethode dargelegt, die einen beliebigen linearen Blockcode endlicher Länge mit einer akzeptablen Komplexität decodieren kann. Dieser wesentliche Abschnitt wird weiter in drei Unterabschnitte gegliedert: Im ersten Unterabschnitt wird die theoretische probabilistische Methode von Dumer modifiziert, um eine effizientere Erzeugung von Informationsmengen zu erzielen. Die eigentliche Erzeugung von Informationsmengen wird im nachfolgenden Unterabschnitt betrachtet und durch ein Beispiel verdeutlicht. Unterabschnitt 4.4.3 beschäftigt sich mit der algorithmischen Beschreibung des gesamten Decodierverfahrens unter Verwendung eines der Akzeptanzkriterien von Kapitel 3. Letztendlich werden die bedeutsamsten Punkte des Kapitels in zusammengefaßter Form wiederholt.

- Ausgehend vom in Abschnitt 4.4 angegeben Decodierverfahren werden schwerpunktmäßig die Simulationsergebnisse in Kapitel 5 behandelt. Da Bewertungen des Decodierverfahrens mit langen Codes bei kleinen Fehlerwahrscheinlichkeiten angestrebt werden, wird ein großer Wert auf die Ausführungszeiten der Simulationsprogramme gelegt. Deswegen werden einige Techniken zur Optimierung von Programmen im ersten Abschnitt erklärt. Dann werden Implementierungsaspekte, die die Simulationszeiten beträchtlich beeinflussen können, sowie die Realisierung aller Schritte des Algorithmus von Unterabschnitt 4.4.3 detailliert. Einer der wichtigsten Beiträge dieser Arbeit ist, durch die in Abschnitt 5.3 durchgeführten Untersuchungen deutlich zeigen zu können, daß die korrekte Auswahl einer geeigneten sogenannten Normierung, die für die Berechnung einiger Wahrscheinlichkeiten im Decodieralgorithmus

notwendig ist, eine entscheidende Rolle hinsichtlich Verkleinerung der Fehlerwahrscheinlichkeit bzw. Decodierkomplexität spielt. Im darauffolgenden Abschnitt werden alle in Kapitel 3 hergeleiteten Akzeptanzkriterien ausführlich analysiert. Erwähntermaßen werden die Akzeptanzkriterien allmählich schlechter mit der wachsenden Steigerung der Länge des angewendeten Codes. Als Beitrag zur Verminderung dieses Nachteils folgt ein suboptimales Akzeptanzkriterium, das die Anzahl der Iterationen für lange Codes erheblich reduziert. Im nächsten Abschnitt werden hauptsächlich vollständige Simulationsergebnisse eines langen Codes evaluiert und mit sowohl einer HDBMD–Decodierung als auch theoretischen bzw. praktischen Grenzen verglichen. Zum Schluß werden die wichtigsten Ergebnisse nochmals rekapituliert.

- Letztlich werden sowohl die wesentlichsten Untersuchungen und Ergebnisse zusammengefaßt als auch ein Ausblick auf mögliche Erweiterungen und Verbesserungen des entwickelten Decodierverfahrens sowie andere verwandte Forschungsarbeiten gegeben.

Kapitel 2

Grundlagen

Im folgenden werden zunächst einige für das Verständnis der Kapitel 3 und 4 notwendige Begriffe vermittelt und Vereinbarungen zur Bezeichnungsweise getroffen. In Abschnitt 2.2 werden Grundbegriffe der Decodierung mit Hilfe von Informationsmengen behandelt, und eine allgemeine Decodiermethode basierend auf Informationsmengen wird vorgestellt. Danach wird das Übertragungssystem dargestellt, das als Ausgangspunkt für die Überlegungen dieser Arbeit dient. Die Zuverlässigkeitsmaße für Soft–Decision werden in Abschnitt 2.4 angegeben, auf denen einige Herleitungen der Akzeptanzkriterien in Kapitel 3 beruhen. Es folgt eine kurze Erläuterung einer SDMLD, die geometrisch mit dem Begriff der Voronoi–Region in Abschnitt 2.6 erklärt wird. Sowohl die gleichverteilten als auch die gaußverteilten Zufallszahlengeneratoren werden in Abschnitt 2.7 berücksichtigt, wobei ein großer Wert auf statistische Eigenschaften sowie Geschwindigkeit gelegt wurde. Dann wird eine Simulationsmethode im vorletzten Abschnitt vorgeschlagen, die besonders für lange Blockcodes mit kleiner Wortfehlerwahrscheinlichkeit geeignet ist. Anschließend wird eine Simulationsschranke für SDMLD erörtert. Sie ermöglicht die Beurteilung der Leistungsfähigkeit eines beliebigen Decodieralgorithmus.

2.1 Grundbegriffe der Kanalcodierung

In diesem Abschnitt werden grundlegende Begriffe der algebraischen Kanalcodierung definiert, die eine zentrale Stellung in dieser Arbeit einnehmen. Weitere Begriffe und Eigenschaften werden beispielsweise in [Bos98, Fri95] behandelt.

2.1.1 Galoisfeld

Ein Galoisfeld $\mathbb{F}_q$ (GF, *Galois Field*, ebenfalls endlicher Zahlenkörper genannt [Fri95]) ist eine Menge mit q Elementen, zwischen denen zwei Rechenoperationen (auch boolesche Verknüpfungen [Stö99] oder Junktoren [Bar97] genannt) erklärt sind, die üblicherweise als Addition + (ebenfalls Exklusiv–Oder, ausschließendes Entweder–Oder oder Antivalenz genannt [Bar97]) und als Multiplikation $\cdot$ (auch logisches Produkt, logisches Und oder Konjunktion genannt [Bar97]) geschrieben werden [Fri95].

Die Theorie von Galoisfeldern spielt keine zentrale Rolle in dieser Arbeit, weil die hier verwendeten Codes binär sind. Für eine mathematische Einführung in Galoisfelder wird auf [Fri95] verwiesen.

Mit $\mathbb{F}_q^n$ wird die Menge aller q^n verschiedenen n-Tupel (bzw. Wörter der Länge n bzw. (Zeilen-)Vektoren bzw. Blöcke) mit Komponenten aus $\mathbb{F}_q$ bezeichnet:

$$\mathbb{F}_q^n = \{(a_0, \ldots, a_{n-1}) \,|\, a_0, \ldots, a_{n-1} \in \mathbb{F}_q\}. \tag{2.1}$$

Entsprechend steht $\mathbb{F}_q^{k,n}$ für die Menge der (k, n)-dimensionalen Matrizen mit Elementen aus $\mathbb{F}_q$.

2.1.2 Blockcode

Ein Blockcode $\mathcal{C}$ der Länge n mit q-stufigen Symbolen ist eine Menge von Vektoren, die aus n Komponenten $c_0, \ldots, c_{n-1} \in \mathbb{F}_q$ bestehen und als Codewörter $\boldsymbol{c} = (c_0, \ldots, c_{n-1}) \in \mathcal{C}$, $\mathcal{C} \subseteq \mathbb{F}_q^n$ bezeichnet werden.

Bei einem $(n, k)_q$-Blockcode $\mathcal{C}$ gilt $\boldsymbol{u} = (u_0, \ldots, u_{k-1}) \in \mathbb{F}_q^k$ für das Infowort sowie $\boldsymbol{c} = (c_0, \ldots, c_{n-1}) \in \mathbb{F}_q^n$ für das Codewort.

2.1.3 Identische Blockcodes

Zwei Blockcodes $\mathcal{C}_1$ und $\mathcal{C}_2$ heißen identisch, falls die Codemenge $\mathcal{C}_1$ identisch mit der Codemenge $\mathcal{C}_2$ ist [Fri95].

2.1.4 Äquivalente Blockcodes

Zwei Blockcodes $\mathcal{C}_1$ und $\mathcal{C}_2$ der Länge n mit q-stufigen Symbolen heißen äquivalent, falls nach einer bestimmten Vertauschung der n Komponenten $c_0, \ldots, c_{n-1} \in \mathbb{F}_q$ aller Codewörter $\boldsymbol{c} \in \mathcal{C}_1$ diese vertauschte Codemenge identisch mit der Codemenge $\mathcal{C}_2$ ist [Fri95].

2.1.5 Linearer Code

Ein Code $\mathcal{C}$ über $\mathbb{F}_q$ heißt linearer Code (oder Gruppe–Code [CS99]), wenn jede Linearkombination von Codewörtern wieder ein Codewort ergibt [Fri95]:

$$\boldsymbol{c_1}, \ldots, \boldsymbol{c_j} \in \mathcal{C}, \quad a_1, \ldots, a_j \in \mathbb{F}_q \quad \Longrightarrow \quad \sum_{i=1}^{j} a_i \, \boldsymbol{c_i} \in \mathcal{C}. \tag{2.2}$$

2.1.6 Systematische Codierung

Vorausgesetzt wird ein $(n,k)_q$-Code $\mathcal{C}$. Die Zuordnung zwischen Infowörtern $\boldsymbol{u} \in \mathbb{F}_q^k$ und Codewörtern $\boldsymbol{c} \in \mathcal{C}$ heißt systematische Codierung, wenn die k Stellen eines Infowortes $\boldsymbol{u}$ explizit ein Teil des Codewortes $\boldsymbol{c}$ sind. Die übrigen $n-k$ Stellen von $\boldsymbol{c}$ heißen dann Prüfstellen [Fri95], Redundanzstellen [Bos98] oder Kontrollstellen [Pet67].

2.1.7 Generatormatrix

Eine (k,n)-dimensionale Matrix $\boldsymbol{G} \in \mathbb{F}_q^{k,n}$ heißt Generatormatrix für den linearen $(n,k)_q$-Code $\mathcal{C}$, wenn

$$\mathcal{C} = \left\{ \boldsymbol{u}\,\boldsymbol{G} \;\middle|\; \boldsymbol{u} \in \mathbb{F}_q^k \right\} \tag{2.3}$$

gilt, wobei

$$\boldsymbol{G} = \begin{pmatrix} \boldsymbol{g}_0 \\ \vdots \\ \boldsymbol{g}_{k-1} \end{pmatrix} = \begin{pmatrix} g_{0,0} & \cdots & g_{0,n-1} \\ \vdots & & \vdots \\ g_{k-1,0} & \cdots & g_{k-1,n-1} \end{pmatrix} \tag{2.4}$$

ist. Die Generatormatrix $\boldsymbol{G}$ erzeugt den Code $\mathcal{C}$ und liefert gleichzeitig eine Codiervorschrift, durch die ein Infowort $\boldsymbol{u}$ auf das Codewort $\boldsymbol{c}$ in folgender Form abgebildet werden kann:

$$\boldsymbol{c} = \boldsymbol{u}\,\boldsymbol{G}. \tag{2.5}$$

2.1.8 Prüfmatrix

Eine $(n-k,n)$-dimensionale Matrix $\boldsymbol{H} \in \mathbb{F}_q^{n-k,n}$ heißt Prüfmatrix (*Parity Check Matrix*) für den linearen $(n,k)_q$-Code $\mathcal{C}$, wenn

$$\mathcal{C} = \left\{ \boldsymbol{c} \in \mathbb{F}_q^n \;\middle|\; \boldsymbol{c}\,\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{0} \right\} \tag{2.6}$$

gilt, wobei das Nullwort $\boldsymbol{0} = (0, \ldots, 0)$ die Länge $n-k$ besitzt.

2.1.9 Syndrom

Vorausgesetzt wird ein $(n,k)_q$-Code $\mathcal{C}$ mit einer Prüfmatrix $\boldsymbol{H} \in \mathbb{F}_q^{n-k,n}$. Ein Syndrom $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$ wird definiert als Multiplikation eines Vektors $\boldsymbol{v} \in \mathbb{F}_q^n$ mit der transponierten Prüfmatrix $\boldsymbol{H}$:

$$\boldsymbol{s} = \boldsymbol{v}\,\boldsymbol{H}^{\mathrm{T}}. \tag{2.7}$$

Das Syndrom $\boldsymbol{s}$ besitzt somit die Länge $n-k$ und ist genau dann das Nullwort $\mathbf{0}$, wenn der Vektor $\boldsymbol{v}$ ein Codewort aus $\mathcal{C}$ ist.

Die Ausdrücke Prüfvektor und Korrektor werden ebenfalls anstatt Syndrom verwendet [Pet67].

2.1.10 Hamminggewicht

Das Hamminggewicht $w_H(\boldsymbol{v})$ eines Vektors $\boldsymbol{v}$ wird definiert als die Anzahl der Komponenten, die ungleich Null sind:

$$w_H(\boldsymbol{v}) = \sum_{i \,|\, v_i \neq 0} 1. \tag{2.8}$$

Es wird auch Hammingnorm genannt [Bos98].

2.1.11 Hammingabstand

Der Hammingabstand $d_H(\boldsymbol{v}, \boldsymbol{v}')$ zwischen zwei Vektoren $\boldsymbol{v}$ und $\boldsymbol{v}'$ wird definiert als die Anzahl der Komponenten, in denen sich diese Vektoren unterscheiden:

$$d_H(\boldsymbol{v}, \boldsymbol{v}') = \sum_{i \,|\, v_i \neq v'_i} 1. \tag{2.9}$$

Er wird ebenfalls als Hammingdistanz bezeichnet [Fri95, Bos98].

Für den Zusammenhang zwischen Hamminggewicht und Hammingabstand gilt:

$$w_H(\boldsymbol{v}) = d_H(\boldsymbol{v}, \mathbf{0}) \quad \text{mit} \quad \mathbf{0} = (0, \ldots, 0). \tag{2.10}$$

2.1.12 Minimalgewicht

Das Minimalgewicht $w_{H\min}$ eines Codes $\mathcal{C}$ wird definiert als das kleinste Hamminggewicht aller Codewörter aus $\mathcal{C}$, außer dem Nullwort $\mathbf{0}$:

$$w_{H\min} = \min\{w_H(\boldsymbol{c}) \,|\, \boldsymbol{c} \in \mathcal{C} \setminus \{\mathbf{0}\}\}. \tag{2.11}$$

2.1.13 Minimalabstand

Der Minimalabstand $d_{H\min}$ eines Codes $\mathcal{C}$ wird definiert als der kleinste Hammingabstand zwischen allen Codewörtern aus $\mathcal{C}$:

$$d_{H\min} = \min\{d_H(\boldsymbol{c}, \boldsymbol{c}') \,|\, \boldsymbol{c}, \boldsymbol{c}' \in \mathcal{C}, \boldsymbol{c} \neq \boldsymbol{c}'\}. \tag{2.12}$$

Da lineare Codes einen Vektorraum darstellen, ist bei diesen Codes das Minimalgewicht $w_{H\min}$ gleich dem Minimalabstand $d_{H\min}$.

Der Minimalabstand $d_{H\min}$ wird auch Minimaldistanz [Fri95] oder Mindestdistanz [Bos98] genannt und ist ein wichtiger Parameter, der die Güte eines Codes $\mathcal{C}$ bestimmt. Je größer der Minimalabstand $d_{H\min}$ ist, und je stärker sich also die Codewörter voneinander unterscheiden, um so besser ist der Code $\mathcal{C}$.

Ein Code $\mathcal{C}$ mit Länge n, Dimension k, Minimalabstand $d_{H\min}$ und q-stufigen Symbolen wird dann als $(n, k, d_{H\min})_q$-Code $\mathcal{C}$ bezeichnet.

2.1.14 Rate

Als Rate R eines $(n, k)_q$-Codes $\mathcal{C}$ wird das Verhältnis von k zu n bezeichnet [Fri95]:

$$R = \frac{k}{n} \leq 1, \tag{2.13}$$

wobei $R = 1$ eine uncodierte Übertragung bedeutet.

2.1.15 Gewichtsprofil

Ein Gewichtsprofil (*Weight Profile*) $\mathcal{W}$ eines linearen Codes $\mathcal{C}$ der Länge n mit Minimalabstand $d_{H\min}$ ist eine geordnete Menge aller verschiedenen Hamminggewichte der Codewörter aus diesem Blockcode $\mathcal{C}$:

$$\mathcal{W} = \{w_0, w_1, \ldots, w_l\},\ w_0 < w_1 < \cdots < w_l, \tag{2.14a}$$

$$w_0 = 0,\ w_1 = d_{H\min},\ \cdots,\ w_l \leq n. \tag{2.14b}$$

Als Beispiel besitzt ein $(7, 4, 3)_2$-Code $\mathcal{C}$ 1 Codewort mit Hamminggewicht 0, 7 Codewörter mit Hamminggewicht 3, 7 Codewörter mit Hamminggewicht 4 und 1 Codewort mit Hamminggewicht 7. Daraus folgt $\mathcal{W} = \{0, 3, 4, 7\}$.

2.1.16 Kugel

Als Kugel $\mathcal{S}(r, \boldsymbol{v})$ mit Hammingradius r um einen Vektor $\boldsymbol{v} \in \mathbb{F}_q^n$ wird die Menge aller Vektoren $\boldsymbol{v'}$ verstanden, die einen Hammingabstand d_H zu $\boldsymbol{v}$ kleiner oder gleich r aufweisen:

$$\mathcal{S}(r, \boldsymbol{v}) = \{\boldsymbol{v'} \in \mathbb{F}_q^n \mid d_H(\boldsymbol{v}, \boldsymbol{v'}) \leq r\}. \tag{2.15}$$

Selbstverständlich ist $\mathcal{S}(0, \boldsymbol{v}) = \{\boldsymbol{v}\}$ und $\mathcal{S}(n, \boldsymbol{v}) = \mathbb{F}_q^n$.

Aus der Kombinatorik [Fri95] folgt

$$|\mathcal{S}(r, \boldsymbol{v})| = \sum_{i=0}^{r} (q-1)^i \binom{n}{i}, \tag{2.16}$$

wobei $|\mathcal{S}(r, \boldsymbol{v})|$ die Anzahl von Elementen der Menge $\mathcal{S}(r, \boldsymbol{v})$ (die Mächtigkeit oder Kardinalzahl) bezeichnet.

Wird eine Kugel um ein Codewort $\boldsymbol{c}$ eines $(n, k)_q$-Codes $\mathcal{C}$ betrachtet, dann wird sie Korrekturkugel [Bos98], Korrigierkugel oder Korrigierbereich genannt [TH93].

2.1.17 Packungsradius

Der Packungsradius (*Packing Radius*) t eines Codes $\mathcal{C}$ mit Minimalabstand $d_{H\min}$ ist der größtmögliche Radius, so daß sich keine Korrekturkugeln überlappen. Alle Vektoren, die innerhalb einer Korrekturkugel $\mathcal{S}(t, \boldsymbol{c})$ liegen, können eindeutig einem Codewort $\boldsymbol{c} \in \mathcal{C}$, das dem Mittelpunkt dieser Kugel entspricht, zugeordnet werden [Bos98]. Somit ermöglicht der Code $\mathcal{C}$ die Korrektur von

$$t = \left\lfloor \frac{d_{H\min} - 1}{2} \right\rfloor \tag{2.17}$$

Fehler, wobei $\lfloor \varsigma \rfloor$ die größte ganze Zahl bezeichnet, die kleiner oder gleich ς ist. Ist $d_{H\min}$ ungerade, so gilt

$$t = \frac{d_{H\min} - 1}{2}, \tag{2.18a}$$

und falls $d_{H\min}$ gerade ist, dann ist

$$t = \frac{d_{H\min}}{2} - 1. \tag{2.18b}$$

2.1.18 Überdeckungsradius

Der Überdeckungsradius (*Covering Radius*) ρ eines $(n,k)_q$-Codes $\mathcal{C}$ ist der kleinstmögliche Radius, so daß alle Vektoren $\boldsymbol{v}$ aus $\mathbb{F}_q^n$ in einer der Korrekturkugeln $\mathcal{S}(\rho, \boldsymbol{c})$ um ein Codewort $\boldsymbol{c} \in \mathcal{C}$ liegen [TH93]:

$$\rho = \max_{\boldsymbol{v} \in \mathbb{F}_q^n} \left\{ \min_{c \in \mathcal{C}} d_H(\boldsymbol{v}, \boldsymbol{c}) \right\}. \tag{2.19}$$

Somit ist ρ der maximale Hammingabstand d_H eines Vektors $\boldsymbol{v}$ zum nächstliegenden Codewort $\boldsymbol{c}$.

Der Überdeckungsradius ρ ist größer oder gleich dem Packungsradius t. Für perfekte Codes gilt die Gleichheit, weil die Korrekturkugeln $\mathcal{S}(\rho, \boldsymbol{c})$ in diesem Fall den gesamten Raum überdecken [Bla83].

2.1.19 Gilbertabstand

Der Gilbertabstand d_G [Bar98] eines nicht linearen Codes $\mathcal{C}$ mit Länge n, q-stufigen Symbolen und Mächtigkeit M (Anzahl der Codewörter) wird durch die Gilbert–Schranke [Gil52, Bos98] folgendermaßen definiert:

$$d_G = \max\left\{ d \;\middle|\; M \sum_{i=0}^{d-1} (q-1)^i \binom{n}{i} \leq q^n \right\}. \tag{2.20}$$

Somit bedeutet d_G die größte ganze Zahl, für die die Multiplikation von M mit der Summe kleiner oder gleich q^n ist.

Im allgemeinen Fall gilt der Gilbertabstand d_G für nicht lineare Codes. Ist der Code $\mathcal{C}$ jedoch linear, dann ist ihre Mächtigkeit $M = q^k$ und Gleichung (2.20) kann in Gleichung (2.21a) vereinfacht werden [Dum96b, Dum97b]:

$$d_G = \max\left\{ d \;\middle|\; \sum_{i=0}^{d-1} (q-1)^i \binom{n}{i} \leq q^{n-k} \right\}. \tag{2.21a}$$

Des weiteren kann Gleichung (2.21a) zu Gleichung (2.21b) umgeformt werden [Dum98a, Dum98c]:

$$d_G = \min\left\{ d \;\middle|\; \sum_{i=0}^{d} (q-1)^i \binom{n}{i} > q^{n-k} \right\}. \tag{2.21b}$$

2.1.20 Varshamovabstand

Der Varshamovabstand d_V [Dum96b] eines $(n,k)_q$-Codes $\mathcal{C}$ kann mittels der Varshamov–Schranke [CC81, Fri95, Bos98] wie folgt geschrieben werden:

$$d_V = \max\left\{ d \;\middle|\; \sum_{i=0}^{d-2} (q-1)^i \binom{n-1}{i} < q^{n-k} \right\}. \tag{2.22a}$$

Der Varshamovabstand ist eigentlich eine Verfeinerung des Gilbertabstandes speziell für lineare Codes. Nach [Bos98] handelt es sich dabei um leicht unterschiedliche Ansätze derselben Idee.

Gleichung (2.22a) ist äquivalent zur folgenden Gleichung [Bos98]:

$$d_V = \min\left\{ d \;\middle|\; \sum_{i=0}^{d-2} (q-1)^i \binom{n-1}{i} \geq q^{n-k-1} \right\}. \tag{2.22b}$$

Offensichtlich ist der Varshamovabstand immer größer als der Gilbertabstand:

$$d_V \geq d_G + 1. \tag{2.23}$$

Im asymptotischen Fall gilt die Gleichheit.

2.1.21 Nebenklasse

Vorausgesetzt wird ein $(n,k)_q$-Code $\mathcal{C}$. Eine Nebenklasse $\mathcal{N}_i$ (*Coset*) eines beliebigen Vektors $\boldsymbol{v} \in \mathbb{F}_2^n$ wird definiert als die Summe dieses Vektors $\boldsymbol{v}$ und aller Codewörter $\boldsymbol{c}$ des Codes $\mathcal{C}$ [Fri95]:

$$\mathcal{N}_i = \boldsymbol{v} + \mathcal{C} = \{\boldsymbol{v} + \boldsymbol{c} \,|\, \boldsymbol{c} \in \mathcal{C}\}. \tag{2.24}$$

Folgende Eigenschaften gelten für Nebenklassen $\mathcal{N}$ [Dor97, Bos98]:

- Jede Nebenklasse $\mathcal{N}_i$ besteht aus q^k verschiedenen Vektoren.
- Die Differenz zweier beliebiger Vektoren aus $\mathcal{N}_i$ ergibt stets ein Codewort $\boldsymbol{c}$ aus $\mathcal{C}$.
- Es existieren insgesamt q^{n-k} Nebenklassen $\mathcal{N}$, die eine eindeutige disjunkte Zerlegung von $\mathbb{F}_q^n$ bilden:

$$\begin{aligned}\mathbb{F}_q^n &= \bigcup_{i=0}^{q^{n-k}-1} \mathcal{N}_i \\ &= \mathcal{C} \cup \{\boldsymbol{v_1} + \mathcal{C}\} \cup \{\boldsymbol{v_2} + \mathcal{C}\} \cup \cdots \cup \{\boldsymbol{v}_{\boldsymbol{q^{n-k}-1}} + \mathcal{C}\}.\end{aligned} \tag{2.25}$$

  Dabei gilt $\boldsymbol{v_i} \notin \mathcal{C}$, außer $\boldsymbol{v_0} = \boldsymbol{0}$.

Aus der letzten Eigenschaft folgt, daß zwei Nebenklassen entweder gleich oder verschieden sind (keine teilweise Überlappung), und jeder beliebige Vektor $\boldsymbol{v}$ in einer und nur einer Nebenklasse $\mathcal{N}_i$ auftaucht.

Eine Nebenklasse wird ebenfalls Restklasse genannt [Bos98].

2.1.22 Nebenklassenführer

Vorausgesetzt wird die obenerwähnte Definition von Nebenklasse. Eine Nebenklasse $\mathcal{N}_i$ wird geordnet, so daß der Vektor an der ersten Stelle von $\mathcal{N}_i$ als Nebenklassenführer (*Coset Leader*) dienen kann [Fri95]. Jeder Vektor einer Nebenklasse $\mathcal{N}_i$ kann als Nebenklassenführer verwendet werden. Dies ändert an den Vektoren von $\mathcal{N}_i$ nichts, es permutiert sie nur [Dor97].

Damit die Fehlerwahrscheinlichkeit einer Decodierung verringert werden kann, ist es üblich, den wahrscheinlichsten Fehlervektor einer Nebenklasse $\mathcal{N}_i$ als Nebenklassenführer auszuwählen. Werden die Nebenklassenführer aller Nebenklassen $\mathcal{N}$ auf diese Weise festgelegt, dann wird eine MLD nach Abschnitt 2.5 durchgeführt.

Bei dem in dieser Arbeit verwendeten Übertragungssystem, das in Abschnitt 2.3 dargestellt wird, haben die wahrscheinlichsten Fehlervektoren die geringsten Hamminggewichte:

$$\Pr(\boldsymbol{v_j}) \geq \Pr(\boldsymbol{v_l}) \implies w_H(\boldsymbol{v_j}) \leq w_H(\boldsymbol{v_l}), \quad \forall\ \boldsymbol{v_j}, \boldsymbol{v_l} \in \mathcal{C}. \tag{2.26}$$

Deswegen wird als Nebenklassenführer derjenige Vektor $\boldsymbol{v_i}$ einer Nebenklasse $\mathcal{N}_i$ festgelegt, der das kleinste Hamminggewicht besitzt [Fri95]:

$$\mathcal{N}_i = \boldsymbol{v_i} + \mathcal{C}, \quad w_H(\boldsymbol{v_i}) \leq w_H(\boldsymbol{v}), \quad \forall\ \boldsymbol{v} \in \mathcal{N}_i. \tag{2.27}$$

Diese Nebenklassenführer $\boldsymbol{v}_i$ kleinsten Hamminggewichtes sind nicht notwendigerweise eindeutig in einer Nebenklasse $\mathcal{N}_i$ bestimmt. Falls mehrere mit kleinstem Hamminggewicht existieren, wird einer davon beliebig ausgewählt. In der ersten Nebenklasse $\mathcal{N}_0 = \mathcal{C}$ ist $\boldsymbol{v}_\mathbf{0} = \mathbf{0}$ dagegen selbstverständlich eindeutig. In der vorliegenden Arbeit wird die Menge der Nebenklassenführer mit $\boldsymbol{\mathcal{M}}$ bezeichnet.

Ein Nebenklassenführer wird auch als Restklassenführer [Bos98] oder als Anführer [Fri95] bezeichnet.

2.2 Grundbegriffe der Decodierung mit Informationsmengen

Im folgenden werden einige Definitionen hinsichtlich der Decodierung mit Informationsmengen diskutiert. Diese Definitionen werden besonders zur Beschreibung der Algorithmen in Kapitel 4 verwendet.

2.2.1 Elementare Zeilenoperationen

Vorausgesetzt wird ein $(n,k)_q$-Code $\mathcal{C}$ mit einer Generatormatrix $\boldsymbol{G} \in \mathbb{F}_q^{k,n}$. Die folgenden sogenannten elementaren Zeilenoperationen in $\boldsymbol{G}$ sind erlaubt, ohne daß der Code $\mathcal{C}$ geändert wird [Fri95, Bar97, BSMM99]:

- Vertauschen von Zeilen.
- Multiplikation einer Zeile mit einem von Null verschiedenen Skalar.
- Addition einer mit einem Skalar multiplizierten Zeile zu einer anderen Zeile.

2.2.2 Kanonische Staffelform

Es sei $\boldsymbol{G} \in \mathbb{F}_q^{k,n}$ eine Generatormatrix für den $(n,k)_q$-Code $\mathcal{C}$. Die kanonische Staffelform (*Echelon Canonical Form*) wird erreicht, wenn $\boldsymbol{G}$ mit den obenerwähnten elementaren Zeilenoperationen in folgende Form überführt werden kann:

$$\boldsymbol{G} = (\boldsymbol{I}\ \boldsymbol{P}), \tag{2.28}$$

wobei $\boldsymbol{I} \in \mathbb{F}_q^{k,k}$ eine Einheitsmatrix ist und $\boldsymbol{P} \in \mathbb{F}_q^{k,n-k}$ eine $(k, n-k)$-dimensionale Matrix bezeichnet.

Gemäß Unterabschnitt 2.1.6 entspricht diese Form einer systematischen Codierung und wird ebenfalls reduzierte Staffelform [Pet67], Zeilennormalform oder gaußsche Normalform [Fri95] genannt.

2.2.3 Informationsmenge

In einem $(n, k)_q$-Code $\mathcal{C}$ wird eine Informationsmenge (IS, *Information Set*) definiert als eine beliebige Menge $\mathcal{I}$ von k Stellen eines Codewortes $\boldsymbol{c} \in \mathcal{C}$, so daß die dazugehörigen k Spalten der Generatormatrix $\boldsymbol{G} \in \mathbb{F}_q^{k,n}$ linear unabhängig sind. Aus der Kenntnis dieser k Stellen können die restlichen $n-k$ Stellen dann eindeutig bestimmt werden [CG90].

Wenn die Generatormatrix $\boldsymbol{G}$ des Codes $\mathcal{C}$ in die kanonische Staffelform nach (2.28) überführt werden kann, dann bilden die k ersten Stellen eine Informationsmenge $\mathcal{I}$ [CC81]. Eine andere beliebige Menge $\mathcal{I}'$ von k Stellen kann eine Informationsmenge bilden, wenn $\boldsymbol{G}$ mit elementaren Zeilenoperationen gemäß Unterabschnitt 2.2.1 in eine andere Generatormatrix $\boldsymbol{G}'$ überführt werden kann, so daß das Hamminggewicht w_H ihrer entsprechenden k Spalten gleich Eins wird.

Normalerweise bilden k beliebig ausgewählte Stellen eines Codewortes $\boldsymbol{c} \in \mathcal{C}$ nicht stets eine Informationsmenge $\mathcal{I}$.

Als Beispiel sei $\boldsymbol{G}$ folgende Generatormatrix eines $(7, 4, 3)_2$-Codes $\mathcal{C}$:

$$\boldsymbol{G} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \tag{2.29}$$

Mit elementaren Zeilenoperationen wird $\boldsymbol{G}$ in die kanonische Staffelform nach (2.28) überführt:

$$\boldsymbol{G}'(\mathcal{I}) = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}, \tag{2.30}$$

wobei die Informationsmenge $\mathcal{I} = \{0, 1, 2, 3\}$ ist. Hingegen bilden die Stellen $\{0, 1, 2, 5\}$ keine Informationsmenge, denn diese vier Spalten von $\boldsymbol{G}$ hängen linear voneinander ab.

2.2.4 MDS-Code

Maximum–Distance–Separable–Codes (MDS-Codes) sind eine wichtige Klasse von $(n, k, n-k+1)_q$-Codes $\mathcal{C}$, bei denen ein Codewort $\boldsymbol{c} \in \mathcal{C}$ durch k beliebig ausgewählte Stellen eindeutig bestimmt werden kann [Fri95]. Das hat zur Folge, daß alle beliebig ausgewählten Blöcke von k Stellen in diesem Code $\mathcal{C}$ immer eine Informationsmenge $\mathcal{I}$ bilden.

Die einzigen binären MDS-Codes sind die trivialen: die $(n, 1, n)_2$-Wiederholungscodes, die $(n, n-1, 2)_2$-Parity–Check–Codes und der $(n, n, 1)$-Code ohne Redundanz [Bos98]. Im Gegensatz dazu bilden die q-stufigen MDS-Codes eine breite Klasse von wesentlich praktischerer Bedeutung. Beispielsweise besitzen die Reed–Solomon–Codes (RS-Codes) [LC83, Bla83, ML85, Fri95, Bos98] diese MDS-Eigenschaft.

2.2.5 Prüfmenge

Basierend auf dem obengenannten Konzept von Informationsmenge wird eine Prüfmenge (PS, *Parity Set*) definiert als die Menge $\mathcal{P}$ der nicht zur Informationsmenge $\mathcal{I}$ gehörenden restlichen $n-k$ Stellen eines Codewortes $\boldsymbol{c} \in \mathcal{C}$ [CC81].

2.2.6 Allgemeine Decodierung mit Informationsmengen

Vorausgesetzt werden ein $(n, k)_q$-Code $\mathcal{C}$ mit einer Generatormatrix $\boldsymbol{G}$ und ein Vektor $\boldsymbol{v}$, der decodiert werden soll. Eine allgemeine Prozedur zur Decodierung mit Hilfe von Informationsmengen kann folgendermaßen formuliert werden [Bar91a, Bar93]:

- **Schritt 1:** Bilde eine Menge von verschiedenen Informationsmengen gemäß einer beliebigen Methode.
- **Schritt 2:** Wähle eine Informationsmenge $\mathcal{I}$ aus der oben gebildeten Menge aus.
- **Schritt 3:** Überführe die Generatormatrix $\boldsymbol{G}$ mit elementaren Zeilenoperationen in eine andere Generatormatrix $\boldsymbol{G}'(\mathcal{I})$, so daß die durch $\mathcal{I}$ bestimmten Spalten von $\boldsymbol{G}'(\mathcal{I})$ Hamminggewicht Eins besitzen.

- **Schritt 4:** Bilde ein Infowort $\boldsymbol{u}_{[\mathcal{I}]}$, das diejenigen k Komponenten aus dem Vektor $\boldsymbol{v}$ enthält, die durch $\mathcal{I}$ bestimmt werden.
- **Schritt 5:** Berechne ein Codewort $\boldsymbol{c}'(\mathcal{I})$ durch eine ähnliche Codiervorschrift wie Gleichung (2.5): $\boldsymbol{c}'(\mathcal{I}) = \boldsymbol{u}_{[\mathcal{I}]}\,\boldsymbol{G}'(\mathcal{I})$.
- **Schritt 6:** Erzeuge eine Liste von Codewörtern $\boldsymbol{c}'(\mathcal{I})$ nach Schritten 2 bis 5 mittels der im Schritt 1 gebildeten Menge.
- **Schritt 7:** Suche in dieser Liste dasjenige geschätzte Codewort $\hat{\boldsymbol{c}}$, das den geringsten Hammingabstand zum Vektor $\boldsymbol{v}$ aufweist.
- **Schritt 8:** Decodiere $\boldsymbol{v}$ als $\hat{\boldsymbol{c}}$.

Gemäß [CC81, Bar93] kann anstelle der Codiervorschrift im Schritt 5 auch das Syndrom anhand der Prüfmatrix nach Gleichung (2.7) verwendet werden.

Wie die Menge von verschiedenen Informationsmengen in Schritt 1 erzeugt wird, damit die Codewörter $\boldsymbol{c}'(\mathcal{I})$ in Schritt 5 mit großer Wahrscheinlichkeit nahe an Vektor $\boldsymbol{v}$ liegen, ist sicherlich die schwierigste Aufgabe bei der Decodierung mit Informationsmengen.

2.2.7 Überdeckungsmenge

Vorausgesetzt werden drei positive ganze Zahlen n, k und t, wobei $n \geq k \geq t > 0$ gilt. Es seien $\mathcal{F}$ eine Menge mit n Elementen und $\boldsymbol{\mathcal{K}}$ eine Menge von Teilmengen $\mathcal{B}$ von $\mathcal{F}$, jede Teilmenge $\mathcal{B}$ habe k Elemente. Gemäß [CG81] wird eine (n, k, t)-Überdeckungsmenge (*Covering Set*) definiert als ein Paar $(\mathcal{F}, \boldsymbol{\mathcal{K}})$, so daß es für jede Teilmenge $T \subseteq \mathcal{F}$ mit t Elementen mindestens ein $\mathcal{B} \in \boldsymbol{\mathcal{K}}$ mit $T \subseteq \mathcal{B}$ gibt.

Normalerweise werden die Elemente von $\mathcal{F}$ als Punkte, die Elemente von $\boldsymbol{\mathcal{K}}$ als Blöcke $\mathcal{B}$ und die Teilmenge T als t-Teilmenge bezeichnet [Bar93]. Wenn alle t-Teilmengen in exakt einem Block $\mathcal{B} \in \boldsymbol{\mathcal{K}}$ enthalten sind, dann wird eine (n, k, t)-Überdeckungsmenge ein (n, k, t)-Steiner–System genannt [CS99].

In dieser Arbeit wird eine (n, k, t)-Überdeckungsmenge mit $\mathcal{U}(n, k, t)$ bezeichnet. Entsprechend steht $|\mathcal{U}(n, k, t)|$ für die Anzahl von Blöcken $\mathcal{B}$ einer (n, k, t)-Überdeckungsmenge (Mächtigkeit oder Kardinalzahl).

Eine einfache untere Schranke für $|\mathcal{U}(n, k, t)|$ wird im folgenden beschrieben: Bei n Stellen ist die Gesamtanzahl unterschiedlicher Fehlermuster $\binom{n}{t}$.

Da ein Block $\mathcal{B}$ k Elemente besitzt, beträgt die Anzahl von unterschiedlichen Fehlermustern, die durch einen einzigen Block $\mathcal{B}$ überdeckt werden können, höchstens $\binom{k}{t}$. Somit kann die untere Schranke wie folgt angegeben werden [CC81]:

$$|\mathcal{U}(n,k,t)| \geq \frac{\binom{n}{t}}{\binom{k}{t}}. \tag{2.31}$$

Für die Mächtigkeit einer (n,k,t)-Überdeckungsmenge gilt folgender Zusammenhang [Bar98]:

$$k\,|\mathcal{U}(n,k,t)| \geq n\,|\mathcal{U}(n-1,k-1,t-1)|. \tag{2.32}$$

Speziell für $t = 1$ gilt

$$|\mathcal{U}(n,k,1)| = \left\lceil \frac{n}{k} \right\rceil. \tag{2.33}$$

Damit kann die untere Schranke (2.31) etwas verbessert werden [Sch64]:

$$|\mathcal{U}(n,k,t)| \geq \left\lceil \frac{n}{k} \cdot \left\lceil \frac{n-1}{k-1} \cdot \ldots \cdot \left\lceil \frac{n-t+1}{k-t+1} \right\rceil \cdots \right\rceil \right\rceil, \tag{2.34}$$

wobei $\lceil \varsigma \rceil$ die kleinste ganze Zahl bezeichnet, die größer oder gleich ς ist.

Falls eine gleichverteilte Auswahl von Blöcken $\mathcal{B}$ durchgeführt wird, kann eine obere Schranke für $|\mathcal{U}(n,k,t)|$ durch folgende Gleichung berechnet werden [ES74, ASE92]:

$$|\mathcal{U}(n,k,t)| \leq \left(\ln \binom{k}{t} + 1 \right) \frac{\binom{n}{t}}{\binom{k}{t}}. \tag{2.35}$$

2.3 Übertragungssystem

Abbildung 2.1 zeigt ein vereinfachtes digitales Übertragungssystem, das auf der Darstellung in [Fri95] basiert. Sowohl die Quellencodierung als auch die Kryptographie werden in der vorliegenden Arbeit nicht berücksichtigt.

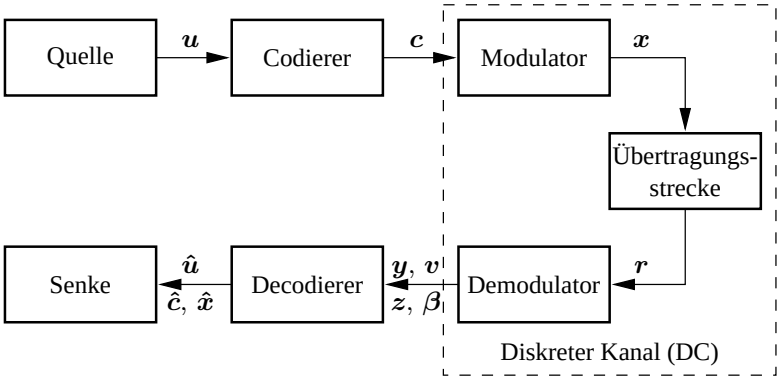

Abbildung 2.1: Das digitale Übertragungssystem.

Überdies sei der (zeit)diskrete Kanal (DC, *Discrete Channel*) wertkontinuierlich, zeitinvariant, gedächtnislos, symmetrisch und binär [Bos98, Fri95]. Er kann als eine Zusammenfassung von Modulator, Übertragungsstrecke und Demodulator gesehen werden. Er läßt sich erzeugen durch

- eine binäre Phasenumtastung (BPSK, *Binary Phase Shift Keying*) als Modulationsverfahren [Kam96],
- einen additiven weißen gaußschen Rauschkanal (AWGN–Kanal, *Additive White Gaussian Noise Channel*) [Pro95] und
- eine Soft–Decision Demodulation.

Das Infowort $\boldsymbol{u} = (u_0, \ldots, u_{k-1})$, $u_j \in \mathbb{F}_2$, $0 \leq j < k$ wird durch den Codierer eines $(n, k)_2$-Codes $\mathcal{C}$ in das Codewort $\boldsymbol{c} = (c_0, \ldots, c_{n-1}) \in \mathcal{C}$, $c_i \in \mathbb{F}_2$, $0 \leq i < n$ umgewandelt. Im Modulator wird dem Codewort $\boldsymbol{c}$ eine Sendefolge $\boldsymbol{x} = (x_0, \ldots, x_{n-1}) \in \mathcal{X}$, $x_i = (-1)^{c_i} \in \{\pm 1\}$ zugeordnet. Der modulierte Code $\mathcal{X}$ ist die Menge der Codewörter $\mathcal{C}$ nach der Modulation und kann als eine Abbildung des Codes $\mathcal{C}$ aus $\mathbb{F}_2^n$ in die Menge $\{\pm 1\}^n$ aus dem euklidischen Vektorraum $\mathbb{R}^n$ verstanden werden. Durch die auftretenden Störsignale in der Übertragungsstrecke wird $\boldsymbol{x}$ zum Empfangsvektor $\boldsymbol{r} = (r_0, \ldots, r_{n-1})$, $r_i \in \mathbb{R}$ möglicherweise verfälscht.

Mit N_0 wird die einseitige Rauschleistungsdichte des Störsignals, und mit E_c wird die mittlere Energie pro empfangenem Codebit bezeichnet. Die Varianz $\sigma^2 = N_0 \,/\, 2$ entspricht also der Varianz des Rauschens, und E_c wird auf Eins normiert.

Der Demodulator liefert dem Decodierer einen Vektor $\boldsymbol{y} = (y_0, \ldots, y_{n-1})$, $y_i \in \mathbb{R}$, der durch das Log–Likelihood–Wahrscheinlichkeitsverhältnis definiert wird:

$$y_i = \kappa \ln \left(\frac{\Pr(r_i \,|\, 1)}{\Pr(r_i \,|\, -1)} \right), \tag{2.36}$$

wobei $\kappa \in \mathbb{R}^+$ eine Konstante ist, die ausschließlich vom Signal zu Rauschverhältnis (SNR, *Signal–to–Noise Ratio*) pro empfangenem Codebit E_c/N_0 abhängt.

Das Vorzeichen jeder Komponente y_i des Vektors $\boldsymbol{y}$ kann als Hard–Decision interpretiert und in einem Vektor $\boldsymbol{v} = (v_0, \ldots, v_{n-1})$ zusammengefaßt werden, wobei $v_i \in \mathbb{F}_2$ folgendermaßen angegeben wird:

$$v_i = \begin{cases} 0 & \text{für} \quad y_i > 0 \\ 1 & \text{für} \quad y_i \leq 0. \end{cases} \tag{2.37}$$

Entsprechend kann der Betrag jeder Komponente y_i als Zuverlässigkeitsinformation β_i verstanden und als ein Vektor $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_{n-1})$, $\beta_i = |y_i| \in \mathbb{R}_0^+$ geschrieben werden, wobei $\mathbb{R}_0^+$ die Menge der positiven reellen Zahlen mit Null bezeichnet.

Außerdem entspricht das Vorzeichen jeder Komponente y_i ebenfalls der Komponente z_i eines Vektors $\boldsymbol{z} = (z_0, \ldots, z_{n-1})$, $z_i = \operatorname{sign}(y_i) \in \{\pm 1\}$ im euklidischen Vektorraum $\mathbb{R}^n$, wobei

$$\operatorname{sign}(\varsigma) = \begin{cases} +1 & \text{für} \quad \varsigma > 0 \\ -1 & \text{für} \quad \varsigma \leq 0 \end{cases} \tag{2.38}$$

die Signumfunktion bezeichnet [Bar97].

Da z_i nur entweder -1 oder $+1$ sein kann, bestehen folgende Zusammenhänge zwischen den Komponenten der Vektoren $\boldsymbol{\beta}$ und $\boldsymbol{z}$:

$$y_i = z_i \, \beta_i, \tag{2.39a}$$

$$\beta_i = z_i \, y_i, \tag{2.39b}$$

und offensichtlich gelten folgende Gleichungen für v_i:

$$v_i = \frac{1 - \operatorname{sign}(y_i)}{2}, \tag{2.40a}$$

$$= \frac{1 - z_i}{2}. \tag{2.40b}$$

Aufgrund der eindeutigen Zuordnung zwischen Infowörtern, Codewörtern und Sendefolgen kann anstelle des Infowortes $\boldsymbol{u}$ auch das Codewort $\boldsymbol{c} \in \mathcal{C}$ oder die Sendefolge $\boldsymbol{x} \in \mathcal{X}$ auf der Empfangsseite geschätzt werden.

Der Decodierer verwendet $\boldsymbol{y}$, um ein möglichst fehlerfreies Infowort bzw. Codewort oder eine möglichst fehlerfreie Sendefolge an seinem Ausgang liefern zu können.

2.4 Zuverlässigkeitsmaße für Soft–Decision

In diesem Abschnitt werden u. a. einige Maße für die Berechnung des Abstandes zwischen Codewörtern vorgestellt. Diese Maße ziehen die Zuverlässigkeitsinformation des Empfangsvektors in Betracht. Weitere Maße werden in [Bos98, Kab91] erörtert.

2.4.1 Eigenschaften von Indexmengen

Mit $\mathcal{D}_0(\boldsymbol{x}, \boldsymbol{y})$ wird eine Indexmenge von Stellen bezeichnet, an welchen eine Sendefolge $\boldsymbol{x}$ mit einem Vektor $\boldsymbol{y}$ im Vorzeichen übereinstimmt [KTK+95, KKTL95a, KKTL95b]:

$$\mathcal{D}_0(\boldsymbol{x}, \boldsymbol{y}) = \{i \,|\, \operatorname{sign}(x_i) = \operatorname{sign}(y_i),\, 0 \leq i < n\}. \tag{2.41}$$

Entsprechend bezeichnet $\mathcal{D}_1(\boldsymbol{x}, \boldsymbol{y})$ eine Indexmenge von Stellen, an welchen sich eine Sendefolge $\boldsymbol{x}$ von einem Vektor $\boldsymbol{y}$ durch das Vorzeichen unterscheidet [KTK+95, KKTL95a, KKTL95b]:

$$\mathcal{D}_1(\boldsymbol{x}, \boldsymbol{y}) = \{i \,|\, \operatorname{sign}(x_i) \neq \operatorname{sign}(y_i),\, 0 \leq i < n\}. \tag{2.42}$$

Als Beispiel kann die Definition des Hammingabstandes $d_H(\boldsymbol{v}, \boldsymbol{v'})$ nach Gleichung (2.9) ebenfalls wie folgt angegeben werden:

$$\begin{aligned} d_H(\boldsymbol{v}, \boldsymbol{v'}) &= \sum_{i \in \mathcal{D}_1(\boldsymbol{v}, \boldsymbol{v'})} 1 \\ &= |\mathcal{D}_1(\boldsymbol{v}, \boldsymbol{v'})|, \end{aligned} \tag{2.43}$$

wobei $|\cdot|$ die Mächtigkeit (Kardinalzahl) einer Menge bezeichnet.

In Anhang A.1 wird gezeigt, daß folgender Zusammenhang zwischen zwei beliebigen Codewörtern $\boldsymbol{c}, \boldsymbol{c'} \in \mathcal{C}$, den entsprechenden Sendefolgen $\boldsymbol{x}, \boldsymbol{x'} \in \mathcal{X}$

und einem beliebigen Vektor $\boldsymbol{y} \in \mathbb{R}^n$ gilt:

$$\begin{aligned} d_H(\boldsymbol{c}, \boldsymbol{c}') &= |\mathcal{D}_1(\boldsymbol{x}, \boldsymbol{x}')| \\ &= |\mathcal{D}_1(\boldsymbol{x}, \boldsymbol{y}) \cap \mathcal{D}_0(\boldsymbol{x}', \boldsymbol{y})| + |\mathcal{D}_0(\boldsymbol{x}, \boldsymbol{y}) \cap \mathcal{D}_1(\boldsymbol{x}', \boldsymbol{y})|. \end{aligned} \tag{2.44}$$

Offensichtlich gelten folgende Eigenschaften für $\mathcal{D}_0(\boldsymbol{x}, \boldsymbol{y})$ und $\mathcal{D}_1(\boldsymbol{x}, \boldsymbol{y})$:

$$\mathcal{D}_0(\boldsymbol{x}, \boldsymbol{y}) = \overline{\mathcal{D}_1(\boldsymbol{x}, \boldsymbol{y})} \quad \text{bzw.} \quad \mathcal{D}_1(\boldsymbol{x}, \boldsymbol{y}) = \overline{\mathcal{D}_0(\boldsymbol{x}, \boldsymbol{y})}, \tag{2.45a}$$

$$\mathcal{D}_0(\boldsymbol{x}, \boldsymbol{y}) \cap \mathcal{D}_1(\boldsymbol{x}, \boldsymbol{y}) = \emptyset, \tag{2.45b}$$

$$\mathcal{D}_0(\boldsymbol{x}, \boldsymbol{y}) \cup \mathcal{D}_1(\boldsymbol{x}, \boldsymbol{y}) = \{0, 1, \ldots, n-1\}, \tag{2.45c}$$

$$|\mathcal{D}_0(\boldsymbol{x}, \boldsymbol{y})| + |\mathcal{D}_1(\boldsymbol{x}, \boldsymbol{y})| = n, \tag{2.45d}$$

wobei $\overline{(\cdot)}$ für das Komplement einer Menge (Komplementärmenge) steht.

Für zwei beliebige Indexmengen $\mathcal{J}(\boldsymbol{x}, \boldsymbol{y})$ und $\mathcal{J}'(\boldsymbol{x}, \boldsymbol{y})$ gilt:

$$\Big|\mathcal{J}(\boldsymbol{x}, \boldsymbol{y}) \cap \mathcal{J}'(\boldsymbol{x}, \boldsymbol{y})\Big| + \Big|\mathcal{J}(\boldsymbol{x}, \boldsymbol{y}) \cap \overline{\mathcal{J}'(\boldsymbol{x}, \boldsymbol{y})}\Big| = \Big|\mathcal{J}(\boldsymbol{x}, \boldsymbol{y})\Big|. \tag{2.46}$$

2.4.2 Indexmenge der unzuverlässigsten Stellen

Vorausgesetzt werden eine beliebige Indexmenge $\mathcal{J}(\boldsymbol{x}, \boldsymbol{y}) \subseteq \{0, 1, \ldots, n-1\}$ und eine beliebige Zahl $\delta \in \mathbb{Z}$, wobei $\mathbb{Z}$ die Menge der ganzen Zahlen bezeichnet. Mit $\mathcal{J}(\boldsymbol{x}, \boldsymbol{y})^{(\delta)}$ wird die Indexmenge der δ unzuverlässigsten Stellen von $\mathcal{J}(\boldsymbol{x}, \boldsymbol{y})$ bezeichnet, wobei $\mathcal{J}(\boldsymbol{x}, \boldsymbol{y})^{(\delta)} = \mathcal{J}(\boldsymbol{x}, \boldsymbol{y})$, wenn $\delta \geq |\mathcal{J}(\boldsymbol{x}, \boldsymbol{y})|$ und $\mathcal{J}(\boldsymbol{x}, \boldsymbol{y})^{(\delta)} = \emptyset$, wenn $\delta \leq 0$ [KTKL99]. Dabei steht $\emptyset$ für die leere Menge.

Basierend auf einer Idee in [Lie93] kann diese Definition durch die drei folgenden Forderungen mathematisch angegeben werden:

$$\emptyset \subseteq \mathcal{J}(\boldsymbol{x}, \boldsymbol{y})^{(\delta)} \subseteq \mathcal{J}(\boldsymbol{x}, \boldsymbol{y}) \subseteq \{0, 1, \ldots, n-1\}, \tag{2.47a}$$

$$\big|\mathcal{J}(\boldsymbol{x}, \boldsymbol{y})^{(\delta)}\big| = \max\{\min\{\delta, |\mathcal{J}(\boldsymbol{x}, \boldsymbol{y})|\}, 0\}, \tag{2.47b}$$

$$i \in \mathcal{J}(\boldsymbol{x}, \boldsymbol{y})^{(\delta)},\ j \in \mathcal{J}(\boldsymbol{x}, \boldsymbol{y}) \setminus \mathcal{J}(\boldsymbol{x}, \boldsymbol{y})^{(\delta)} \implies \beta_i \leq \beta_j. \tag{2.47c}$$

2.4.3 Gewichtetes Hamminggewicht

Das gewichtete Hamminggewicht $w_\beta(\boldsymbol{y})$ eines Vektors $\boldsymbol{y}$ wird definiert als:

$$w_\beta(\boldsymbol{y}) = \sum_{i \,|\, z_i = -1} \beta_i, \tag{2.48}$$

wobei $\beta_i = |y_i| \in \mathbb{R}_0^+$ die Zuverlässigkeitsinformation jeder Komponente y_i des Vektors $\boldsymbol{y}$ bezeichnet und $z_i = \operatorname{sign}(y_i) \in \{\pm 1\}$ das Vorzeichen jeder Komponente y_i des Vektors $\boldsymbol{y}$.

2.4.4 Gewichteter Hammingabstand

Der gewichtete Hammingabstand $d_\beta(\boldsymbol{x}, \boldsymbol{y})$ zwischen einer Sendefolge $\boldsymbol{x}$ und einem Vektor $\boldsymbol{y}$ wird definiert als:

$$d_\beta(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i \in \mathcal{D}_1(\boldsymbol{x}, \boldsymbol{y})} \beta_i, \tag{2.49}$$

wobei $\beta_i = |y_i| \in \mathbb{R}_0^+$ die Zuverlässigkeitsinformation des Vektors $\boldsymbol{y}$ bezeichnet und $\mathcal{D}_1(\boldsymbol{x}, \boldsymbol{y})$ die Indexmenge gemäß Gleichung (2.42).

2.4.5 Euklidische Norm

Die euklidische Norm $\|\boldsymbol{y}\| \in \mathbb{R}$ eines Vektors $\boldsymbol{y}$ wird definiert als:

$$\|\boldsymbol{y}\| = \sqrt{\sum_i {y_i}^2}. \tag{2.50}$$

Sie wird auch euklidisches Gewicht $w_E(\boldsymbol{y})$ [God91], Betrag, Länge von $\boldsymbol{y}$ [Stö99] oder einfach Norm $\|\boldsymbol{y}\|_2$ [BSMM99] genannt.

2.4.6 Maximumnorm

Die Maximumnorm $\|\boldsymbol{y}\|_\infty \in \mathbb{R}$ eines Vektors $\boldsymbol{y}$ wird definiert als:

$$\|\boldsymbol{y}\|_\infty = \max_{0 \le i < n} \beta_i, \tag{2.51}$$

wobei $\beta_i = |y_i| \in \mathbb{R}_0^+$ die Zuverlässigkeitsinformation ist.

Die Maximumnorm wird ebenfalls als sup–Norm bezeichnet [Bar97].

2.4.7 Einheitsvektor

Einheitsvektor $\boldsymbol{y}^0$ wird ein Vektor der Länge n genannt, dessen euklidische Norm $\|\boldsymbol{y}^0\|$ gleich Eins ist:

$$\boldsymbol{y}^0 = \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|} = ({y^0}_0, \ldots, {y^0}_{n-1}), \tag{2.52a}$$

$${y^0}_i = \frac{y_i}{\|\boldsymbol{y}\|} \in \mathbb{R}, \quad 0 \le i < n, \tag{2.52b}$$

wobei $\|\boldsymbol{y}\|$ nach Gleichung (2.50) angegeben wird:

$$y^0{}_i = \frac{y_i}{\sqrt{\sum_j y_j{}^2}}, \quad 0 \leq i, j < n. \tag{2.52c}$$

Der Einheitsvektor wird auch als normierter Vektor [Stö99] oder als Einsvektor [Bar97] bezeichnet.

2.4.8 Normierter Vektor

Ein normierter Vektor $\tilde{\boldsymbol{y}}$ wird definiert als:

$$\tilde{\boldsymbol{y}} = \frac{\boldsymbol{y}}{\|\boldsymbol{y}\|_\infty} = (\tilde{y}_0, \ldots, \tilde{y}_{n-1}), \tag{2.53a}$$

$$\tilde{y}_i = \frac{y_i}{\|\boldsymbol{y}\|_\infty} \in \mathbb{R}, \quad |\tilde{y}_i| \leq 1, \quad 0 \leq i < n, \tag{2.53b}$$

wobei $\|\boldsymbol{y}\|_\infty$ nach Gleichung (2.51) angegeben wird:

$$\tilde{y}_i = \frac{y_i}{\max\limits_{0 \leq i < n} \beta_i}, \quad 0 \leq i < n. \tag{2.53c}$$

2.4.9 Skalarprodukt

Das Skalarprodukt $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ von zwei Vektoren $\boldsymbol{x}$ und $\boldsymbol{y}$ wird definiert als:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_i x_i \, y_i. \tag{2.54}$$

Es wird ebenfalls Punktprodukt $\boldsymbol{x} \cdot \boldsymbol{y}$ [BSMM99], Innenprodukt [Bar97, Stö99] oder einfach Korrelation zwischen $\boldsymbol{x}$ und $\boldsymbol{y}$ [Fri95] genannt.

In Anhang A.2 wird gezeigt, daß Gleichung (2.54) folgendermaßen umgeformt werden kann:

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_i \beta_i - 2 \, d_\beta(\boldsymbol{x}, \boldsymbol{z}). \tag{2.55}$$

2.4.10 Winkel

Der zwischen den zwei Vektoren $\boldsymbol{y}$ und $\boldsymbol{y}'$ eingeschlossene Winkel $\varphi(\boldsymbol{y}, \boldsymbol{y}')$ wird definiert als:

$$\cos\varphi(\boldsymbol{y}, \boldsymbol{y}') = \frac{\langle \boldsymbol{y}, \boldsymbol{y}' \rangle}{\|\boldsymbol{y}\| \, \|\boldsymbol{y}'\|}, \quad \boldsymbol{y}, \boldsymbol{y}' \neq 0, \quad 0 \leq \varphi(\boldsymbol{y}, \boldsymbol{y}') \leq \pi, \tag{2.56}$$

2.4.11 Euklidischer Abstand

Der euklidische Abstand $d_E(\boldsymbol{x}, \boldsymbol{y})$ zwischen einer Sendefolge $\boldsymbol{x}$ und einem Vektor $\boldsymbol{y}$ wird definiert als:

$$d_E(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\sum_i (x_i - y_i)^2}. \tag{2.57}$$

Er wird auch als euklidische Distanz $\|\boldsymbol{y} - \boldsymbol{x}\|$ bezeichnet [Bos98, God91].

Offensichtlich gilt für den Zusammenhang zwischen euklidischer Norm und euklidischem Abstand:

$$\|\boldsymbol{y}\| = d_E(\boldsymbol{0}, \boldsymbol{y}) \quad \text{mit} \quad \boldsymbol{0} = (0, \ldots, 0). \tag{2.58}$$

Darüber hinaus gilt folgender Zusammenhang zwischen euklidischem Abstand und Hammingabstand für zwei Sendefolgen $\boldsymbol{x}$ und $\boldsymbol{x}'$ und die entsprechenden Codewörter $\boldsymbol{c}$ und $\boldsymbol{c}'$:

$$d_E(\boldsymbol{x}, \boldsymbol{x}') = 2\sqrt{d_H(\boldsymbol{c}, \boldsymbol{c}')}. \tag{2.59}$$

2.4.12 Euklidischer Minimalabstand

Der euklidische Minimalabstand $d_{E\min}$ eines modulierten Codes $\mathcal{X}$ wird definiert als der kleinste euklidische Abstand zwischen allen Sendefolgen aus $\mathcal{X}$:

$$d_{E\min} = \min\{d_E(\boldsymbol{x}, \boldsymbol{x}') \,|\, \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}, \boldsymbol{x} \neq \boldsymbol{x}'\}. \tag{2.60}$$

Nach Gleichung (2.59) gilt

$$d_{E\min} = 2\sqrt{d_{H\min}} \tag{2.61}$$

für den Zusammenhang zwischen euklidischem Minimalabstand $d_{E\min}$ und Minimalabstand $d_{H\min}$.

2.4.13 Quadratischer euklidischer Abstand

Vorausgesetzt wird ein modulierter Code $\mathcal{X}$ der Länge n. Der quadratische euklidische Abstand ${d_E}^2(\boldsymbol{x}, \boldsymbol{y})$ zwischen einer Sendefolge $\boldsymbol{x} \in \mathcal{X}$ und einem Vektor $\boldsymbol{y} \in \mathbb{R}^n$ kann wie folgt umgeformt werden:

$$
\begin{aligned}
{d_E}^2(\boldsymbol{x}, \boldsymbol{y}) &= \sum_{i=0}^{n-1} (x_i - y_i)^2 && \text{(2.62a)} \\
&= \underbrace{\sum_{i=0}^{n-1} {x_i}^2}_{n} - 2 \underbrace{\sum_{i=0}^{n-1} x_i\, y_i}_{\langle \boldsymbol{x}, \boldsymbol{y} \rangle} + \underbrace{\sum_{i=0}^{n-1} {y_i}^2}_{\text{konstant}} . && \text{(2.62b)}
\end{aligned}
$$

Daraus folgt, daß eine Minimierung des quadratischen euklidischen Abstandes ${d_E}^2(\boldsymbol{x}, \boldsymbol{y})$ in bezug auf $\boldsymbol{x}$ bei gegebenem $\boldsymbol{y}$ gleich einer Maximierung des Skalarproduktes $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ ist, weil sowohl der erste Term als auch der letzte Term in Gleichung (2.62b) unabhängig von $\boldsymbol{x}$ sind. Am Ende des nächsten Unterabschnittes wird dieser Ansatz nochmals diskutiert.

2.4.14 Ellipsoidischer Abstand

Vorausgesetzt wird ein Zuverlässigkeitsvektor $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_{n-1})$, $\beta_i = |y_i| \in \mathbb{R}_0^+$, $0 \leq i < n$. Der ellipsoidische Abstand $d_{\mathcal{E}}(\boldsymbol{\beta}, \boldsymbol{v}, \boldsymbol{v'})$ zwischen zwei Vektoren $\boldsymbol{v}$ und $\boldsymbol{v'}$ aus $\mathbb{F}_2^n$ wird definiert als [Dum97b, Dum]:

$$
d_{\mathcal{E}}(\boldsymbol{\beta}, \boldsymbol{v}, \boldsymbol{v'}) = \sum_i \beta_i \, (v_i - v_i')^2, \tag{2.63}
$$

wobei $v_i, v_i' \in \{0, 1\}$ hier als reelle Zahlen behandelt werden.

Mittels Gleichung (2.42) kann diese Definition ebenfalls folgendermaßen angegeben werden:

$$
d_{\mathcal{E}}(\boldsymbol{\beta}, \boldsymbol{v}, \boldsymbol{v'}) = \sum_{i \in \mathcal{D}_1(\boldsymbol{v}, \boldsymbol{v'})} \beta_i, \tag{2.64}
$$

Offensichtlich existiert ein enger Zusammenhang zwischen ellipsoidischem Abstand und gewichtetem Hammingabstand. Wenn $\boldsymbol{v}$ der Sendefolge $\boldsymbol{x}$ zugeordnet wird und $\boldsymbol{v'}$ als Hard–Decision Information von $\boldsymbol{y}$ und $\boldsymbol{\beta}$ als Zuverlässigkeitsinformation von $\boldsymbol{y}$ betrachtet werden, dann ist der ellipsoidische Abstand $d_{\mathcal{E}}(\boldsymbol{\beta}, \boldsymbol{v}, \boldsymbol{v'})$ gleich dem gewichteten Hammingabstand $d_\beta(\boldsymbol{x}, \boldsymbol{y})$ gemäß Gleichung (2.49).

2.4.15 Diskretes Ellipsoid

Ein diskretes Ellipsoid $\mathcal{E}(r, \boldsymbol{\beta}, \boldsymbol{v})$ mit Hammingradius r und Zuverlässigkeitsvektor $\boldsymbol{\beta}$ um einen Vektor $\boldsymbol{v}$ wird definiert als die Menge aller Vektoren $\boldsymbol{v'}$, die einen ellipsoidischen Abstand $d_{\mathcal{E}}$ zu $\boldsymbol{v}$ kleiner oder gleich r aufweisen [Dum]:

$$\mathcal{E}(r, \boldsymbol{\beta}, \boldsymbol{v}) = \{\boldsymbol{v'} \in \{0,1\}^n \mid d_{\mathcal{E}}(\boldsymbol{\beta}, \boldsymbol{v}, \boldsymbol{v'}) \leq r\}. \tag{2.65}$$

Selbstverständlich ist eine Korrekturkugel $\mathcal{S}(r, \boldsymbol{c})$ nach Gleichung (2.15) ein spezieller Fall eines diskreten Ellipsoids $\mathcal{E}(r, \boldsymbol{\beta}, \boldsymbol{v})$ mit $\boldsymbol{v} = \boldsymbol{c}$ und $\beta_i = 1$, $0 \leq i < n$.

Diskrete Ellipsoide spielen eine wesentliche Rolle bei Soft–Decision wegen ihres engen Zusammenhanges mit einer SDMLD, wie in Abschnitt 4.3 deutlich wird.

2.5 Soft–Decision–Maximum–Likelihood–Decodierung (SDMLD)

Normalerweise ist das gesendete Codewort $\boldsymbol{c}$ auf der Empfangsseite nicht bekannt, und die Störsignale des Kanals führen zu Fehlern im Empfangsvektor $\boldsymbol{r}$. Der Decodierer versucht, das gesendete Codewort $\boldsymbol{c}$ richtig zu schätzen unter Kenntnis des Vektors $\boldsymbol{r}$.

Das Ziel einer Decodierung ist also, daß das geschätzte Codewort $\hat{\boldsymbol{c}}$ möglichst häufig mit dem gesendeten Codewort $\boldsymbol{c}$ übereinstimmt. Diese Forderung ist das Kriterium, nach dem der Decodierer konstruiert werden soll.

Als Konzept für eine optimale Decodierung für Codewörter wird das Maximum–Likelihood–Prinzip verwendet [Bos98]. Bei Soft–Decision im euklidischen Vektorraum $\mathbb{R}^n$ kann diese optimale Prozedur wie folgt definiert werden: Vorausgesetzt wird ein modulierter Code $\mathcal{X}$ der Länge n, der sich durch Abbildung eines Codes $\mathcal{C}$ aus $\mathbb{F}_2^n$ in die Menge $\{\pm 1\}^n$ ergibt. Ferner wird zur Herleitung dieser Decodiervorschrift grundsätzlich angenommen, daß alle 2^k geschätzten Sendefolgen $\hat{\boldsymbol{x}}$ mit der gleichen A–priori–Wahrscheinlichkeit $\Pr(\hat{\boldsymbol{x}}) = 2^{-k}$ vom Decodierer abgegeben wurden. Damit ist die wahrscheinlichste Sendefolge $\boldsymbol{x}_{ML}$ diejenige geschätzte Sendefolge $\hat{\boldsymbol{x}} \in \mathcal{X}$, für die $\Pr(\boldsymbol{r} \mid \hat{\boldsymbol{x}})$ maximal wird:

$$\boldsymbol{x}_{ML} = \underset{\hat{\boldsymbol{x}} \in \mathcal{X}}{\arg\max} \Pr(\boldsymbol{r} \mid \hat{\boldsymbol{x}}), \tag{2.66}$$

wobei $\Pr(\boldsymbol{r} \,|\, \hat{\boldsymbol{x}})$ die A–priori–Übergangswahrscheinlichkeit ist, daß $\boldsymbol{r}$ empfangen wurde unter der Voraussetzung, daß $\hat{\boldsymbol{x}}$ gesendet wurde, und $\arg\max f(\varsigma)$ das Argument ς bezeichnet, das eine beliebige Funktion $f(\varsigma)$ maximiert.

Ohne die Annahme gleicher A–priori–Wahrscheinlichkeiten $\Pr(\hat{\boldsymbol{x}})$ ergibt sich jedoch ein anderer Decodierer, nämlich der Maximum–a–posteriori–Decodierer (MAP–Decodierer, *Maximum–a–posteriori Decoder*), der exakt auf die Quellenstatistik angepaßt ist und bei nicht gleichwahrscheinlichen geschätzten Sendefolgen $\hat{\boldsymbol{x}}$ zu einer kleineren Fehlerwahrscheinlichkeit führt als die Fehlerwahrscheinlichkeit eines entsprechenden Maximum–Likelihood–Decodierers (ML–Decodierer, *Maximum–Likelihood Decoder*) [Fri95, Dor97]. Infolge dieser Abhängigkeit von der Quellenstatistik wird der MAP–Decodierer in der vorliegenden Arbeit nicht verwendet.

Die Maximierung von $\Pr(\boldsymbol{r} \,|\, \hat{\boldsymbol{x}})$ gemäß Gleichung (2.66) ist gleichbedeutend mit der Maximierung von $\ln \Pr(\boldsymbol{r} \,|\, \hat{\boldsymbol{x}}) - \ln \Pr(\boldsymbol{r} \,|\, -\hat{\boldsymbol{x}})$:

$$\boldsymbol{x}_{ML} = \underset{\hat{\boldsymbol{x}} \in \mathcal{X}}{\arg\max} \, \{\ln \Pr(\boldsymbol{r} \,|\, \hat{\boldsymbol{x}}) - \ln \Pr(\boldsymbol{r} \,|\, -\hat{\boldsymbol{x}})\}. \tag{2.67}$$

Da der zeitdiskrete Kanal gedächtnislos ist, gilt:

$$\Pr(\boldsymbol{r} \,|\, \hat{\boldsymbol{x}}) = \prod_i \Pr(r_i \,|\, \hat{x}_i). \tag{2.68}$$

Wie in Anhang A.3 bewiesen wird, ist Gleichung (2.67) bei einem AWGN–Kanal äquivalent zu:

$$\boldsymbol{x}_{ML} = \underset{\hat{\boldsymbol{x}} \in \mathcal{X}}{\arg\max} \, \langle \hat{\boldsymbol{x}}, \boldsymbol{y} \rangle. \tag{2.69}$$

Infolgedessen wird als wahrscheinlichste Sendefolge $\boldsymbol{x}_{ML}$ diejenige geschätzte Sendefolge $\hat{\boldsymbol{x}}$ gewählt, für die das Skalarprodukt $\langle \hat{\boldsymbol{x}}, \boldsymbol{y} \rangle$ (d. h. die Korrelation) mit dem Vektor $\boldsymbol{y}$ maximal wird.

Die SDMLD kann offensichtlich nach Gleichung (2.69) folgendermaßen ausgeschrieben werden:

$$\langle \hat{\boldsymbol{x}}, \boldsymbol{y} \rangle \geq \langle \hat{\boldsymbol{x}}', \boldsymbol{y} \rangle, \;\; \forall \; \hat{\boldsymbol{x}}' \in \mathcal{X}. \tag{2.70}$$

Anhang A.4 zeigt, daß diese Ungleichung äquivalent zur folgenden Ungleichung ist:

$$d_E(\hat{\boldsymbol{x}}, \boldsymbol{y}) \leq d_E(\hat{\boldsymbol{x}}', \boldsymbol{y}), \;\; \forall \; \hat{\boldsymbol{x}}' \in \mathcal{X}. \tag{2.71}$$

Somit wird die SDMLD auch durchgeführt, wenn zum Vektor $\boldsymbol{y} \in \mathbb{R}^n$ eine geschätzte Sendefolge $\hat{\boldsymbol{x}} \in \mathcal{X}$ gewählt wird, die von $\boldsymbol{y}$ den minimalen euklidischen Abstand $d_E(\hat{\boldsymbol{x}}, \boldsymbol{y})$ hat.

Im nächsten Abschnitt wird das Konzept von SDMLD durch eine geometrische Interpretation verdeutlicht.

2.6 Voronoi–Regionen eines Codes

Die Voronoi–Regionen sind von grundlegender Bedeutung für die geometrische Analyse der Güte eines Blockcodes bei Soft–Decision–Decodierung. Beispielsweise werden sie für die theoretische Berechnung der Fehlerwahrscheinlichkeit bei einem AWGN–Kanal verwendet [Agr96].

Gemäß Forney [For91] bestimmt die Form der Voronoi–Region fast alle Eigenschaften eines Codes, die für eine Übertragung wichtig sind.

Vorausgesetzt wird ein modulierter $(n, k)_2$-Code $\mathcal{X}$ mit Minimalabstand $d_{E\min}$. Die Voronoi–Region $\mathcal{V}(\boldsymbol{x})$ einer Sendefolge $\boldsymbol{x} \in \mathcal{X}$ wird definiert als die Menge aller Punkte (Vektoren) $\boldsymbol{y} \in \mathbb{R}^n$, die die geringsten quadratischen euklidischen Abstände zu $\boldsymbol{x}$ aufweisen, verglichen mit allen anderen Sendefolgen $\boldsymbol{x'} \in \mathcal{X}$ [Agr96, Agr98]:

$$\mathcal{V}(\boldsymbol{x}) = \{\boldsymbol{y} \in \mathbb{R}^n \mid {d_E}^2(\boldsymbol{y}, \boldsymbol{x}) \leq {d_E}^2(\boldsymbol{y}, \boldsymbol{x'}),\ \forall\ \boldsymbol{x'} \in \mathcal{X}\}. \tag{2.72}$$

Somit kann eine SDMLD durch ihre geometrische Interpretation angegeben werden:

$$\boldsymbol{y} \in \mathcal{V}(\boldsymbol{x}) \Longleftrightarrow \hat{\boldsymbol{x}} = \boldsymbol{x}. \tag{2.73}$$

Zum Vektor $\boldsymbol{y}$ wird hier als geschätzte Sendefolge $\hat{\boldsymbol{x}}$ die Sendefolge $\boldsymbol{x}$ gewählt, falls $\boldsymbol{y}$ innerhalb der Voronoi–Region $\mathcal{V}(\boldsymbol{x})$ der Sendefolge $\boldsymbol{x}$ liegt.

Die Sendefolgen $\boldsymbol{x}$ bilden 2^k Eckpunkte eines n-dimensionalen Hyperwürfels, der um den Koordinatenursprung $\mathbf{0}$ zentriert ist und durch $2\,n$ Hyperebenen $x_i \in \{\pm 1\}$, $0 \leq i < n$ begrenzt wird. Die Voronoi–Region $\mathcal{V}(\boldsymbol{x})$ ist eine Hyperpyramide mit einer Spitze am Ursprung $\mathbf{0}$ und unendlicher Höhe. Ihre entsprechende unendliche Grundfläche ist um die Sendefolge $\boldsymbol{x}$ zentriert [Agr98]. Der dreidimensionale euklidische Hyperwürfel und die entsprechende Voronoi–Region $\mathcal{V}(\boldsymbol{x})$ einer Sendefolge $\boldsymbol{x}$ werden für einen modulierten $(3, 2)_2$-Code $\mathcal{X}$ in Abbildung 2.2 gezeigt.

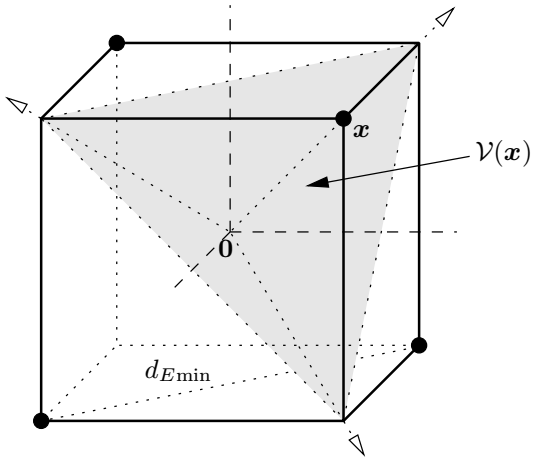

Abbildung 2.2: Dreidimensionaler euklidischer Hyperwürfel und Voronoi–Region $\mathcal{V}(\boldsymbol{x})$.

2.7 Zufallszahlengeneratoren

Die Anwendung von Zufallszahlengeneratoren in Simulationen anstatt echter Zufallsdaten hat im allgemeinen den Vorteil, daß die Ergebnisse reproduziert werden können [Pus96].

Im folgenden werden extrem schnelle Zufallszahlengeneratoren als ein Beitrag dieser Arbeit vorgeschlagen. Dabei werden zunächst gleichverteilte Zufallszahlengeneratoren vorgestellt, die sowohl für den Decodieralgorithmus von Kapitel 4 als auch für die Simulationen in Kapitel 5 nötig sind. Anschließend werden gaußverteilte Zufallszahlengeneratoren betrachtet, die als Störsignal für die Simulationsmethode von Abschnitt 2.8 benutzt werden.

2.7.1 Gleichverteilte Zufallszahlengeneratoren

Nach Park und Miller haben beinahe alle gleichverteilten Zufallszahlengeneratoren, die von Programmiersprachen bzw. Betriebsystemen zur Verfügung gestellt werden, so schlechte statistische Eigenschaften, daß sie die Ergebnisse von sehr langen Simulationen völlig verfälschen können [PM88]. Besonders betroffen in diesem Fall sind alle Arten von gleichverteilten Zufallszahlengeneratoren, die auf linear rückgekoppelten Schieberegistern der Länge 31 beruhen [DB96, Kam96].

Da die gesamte Anzahl von gesendeten Codewörtern in den Simulationen von Kapitel 5 in einzelnen Fällen sogar wesentlich größer als 10 Millionen sein kann, wurde ein besonderer Wert nicht nur auf die statistischen Eigenschaften sondern auch auf die Geschwindigkeit der gleichverteilten Zufallszahlengeneratoren gelegt. Deswegen wurden lediglich linear kongruente Generatoren (*Linear Congruential Generators*) in dieser Arbeit verwendet, die in multiplikative kongruente Generatoren (*Multiplicative Congruential Generators*) und gemischte kongruente Generatoren (*Mixed Congruential Generators*) untergegliedert werden können [PTVF92, Knu98a].

Park und Miller haben hauptsächlich vier verschiedene gleichverteilte Zufallszahlengeneratoren modulo $2^{31} - 1$ als Referenzen vorgeschlagen, die hervorragende statistische Eigenschaften haben [PM88]. Genauso geeignet sind die 410 gleichverteilten Zufallszahlengeneratoren, die Fishman und Moore mühsam untersucht haben [FM86] oder diejenigen, die von L'Ecuyer in seinem Beitrag zu gemeinsamen gleichverteilten Zufallszahlengeneratoren analysiert wurden [L'E88].

In [PTVF92] wurden viele gleichverteilte Zufallszahlengeneratoren von [Knu98a] entnommen und in der Programmiersprache C implementiert. In [Wu97] wurden u. a. alle multiplikativen kongruenten Generatoren modulo $2^{31} - 1$ mit einem Multiplikator der Form $\pm 2^i \pm 2^j$ getestet, wobei $0 < i < j < 31$ ist.

Alle erwähnten gleichverteilten Zufallszahlengeneratoren wurden in der vorliegenden Arbeit mithilfe der Programmiertechniken von Abschnitt 5.1 sorgfältig optimiert und nach ihren Geschwindigkeiten bewertet. Schließlich wurden folgende gleichverteilte Zufallszahlengeneratoren ausgewählt, weil sie einerseits ausgezeichnete statistische Eigenschaften besitzen und andererseits mindestens um den Faktor 15 schneller als die ursprünglichen gleichverteilten Zufallszahlengeneratoren von Park und Miller [PM88] waren:

- Der gleichverteilte Zufallszahlengenerator modulo 2^{32} namens ***ranqd1*** von [PTVF92, S. 284].
- Die gleichverteilten Zufallszahlengeneratoren modulo $2^{31} - 1$ mit Multiplikatoren $\mathbf{(2^{31} - 1) - 2^{16} - 2^{11}}$ und $\mathbf{2^{15} - 2^{10}}$ von [Wu97, S. 260].

Der zusätzliche Vorteil dieser linear kongruenten Generatoren im Vergleich zu den gleichverteilten Zufallszahlengeneratoren auf Basis von linear rückgekoppelten Schieberegistern [DB96, Kam96] ist, daß ebenfalls besonders lange Nullfolgen auftreten können [Pus96].

2.7.2 Gaußverteilte Zufallszahlengeneratoren

Aus demselben im letzten Unterabschnitt angegebenen Grund wurde auch hier großer Wert auf die statistischen Eigenschaften sowie Geschwindigkeiten von gaußverteilten Zufallszahlengeneratoren gelegt.

In [PTVF92, S. 289] wurde die originale Polarmethode (*Polar Method*) von Box et al. [BMM58] in der Programmiersprache C realisiert. Der Vergleich zwischen allen Methoden zur Erzeugung gaußverteilter Zufallszahlen von [Knu98a] wurde bereits in einer früheren Arbeit durchgeführt [Bar91b]. Im Jahre 1992 hat Leva einen schnelleren gaußverteilten Zufallszahlengenerator vorgeschlagen [Lev86]. In einem Beitrag von Eck und Söder wurde ein noch schnellerer gaußverteilter Zufallszahlengenerator basierend auf einer vor den Simulationen gespeicherten Tabelle erörtert [ES96].

Dagegen ist die schnellste in der Literatur gefundene Methode zur Erzeugung gaußverteilter Zufallszahlen die Arbeit von Wallace, in der bestimmte Transformationen in einer orthogonalen, (256,4)-dimensionalen Matrix von gaußschen Zahlen durchgeführt werden [Wal96]. Diese Methode wurde zusätzlich um 24 % mittels der in Abschnitt 5.1 beschriebenen Programmiertechniken optimiert, so daß die Erzeugung von gaußverteilten Zufallszahlen etwa die gleiche Zeit benötigt wie die von gleichverteilten und mindestens um den Faktor 37 schneller ist als die originale Polarmethode von Box et al. [BMM58, PTVF92].

2.8 Simulationsmethode

Vorausgesetzt werden ein modulierter $(n, k)_2$-Code $\mathcal{X}$ und ein digitales Übertragungssystem gemäß Abbildung 2.1. Damit Codes unterschiedlicher Raten R untereinander und mit einer uncodierten Übertragung ($R = 1$) verglichen werden können, wird die Fehlerwahrscheinlichkeit üblicherweise in Abhängigkeit vom SNR pro empfangenem Infobit $E_b \,/\, N_0$ anstelle des SNR pro empfangenem Codebit $E_c \,/\, N_0$ angegeben [Fri95], wobei E_b die mittlere Energie pro empfangenem Infobit bezeichnet. Die Energienormierung auf E_b gewährleistet, daß unabhängig von der Rate R des verwendeten modulierten Codes $\mathcal{X}$ stets gleichviel Energie aufgewendet wird [Bos98].

Zwischen E_c und E_b besteht folgender Zusammenhang:

$$n\,E_c = k\,E_b \quad \Longrightarrow \quad \frac{E_c}{N_0} = R\,\frac{E_b}{N_0}. \tag{2.74}$$

Zur Ermittlung der Fehlerwahrscheinlichkeit existieren verschiedene Verfahren [Söd93]. Im allgemeinen kann die Fehlerwahrscheinlichkeit $\Pr(\hat{\boldsymbol{x}} \neq \boldsymbol{x})$ bei einem gegebenen Wert von $E_b \,/\, N_0$ wie folgt ermittelt werden:

$$\Pr(\hat{\boldsymbol{x}} \neq \boldsymbol{x}) = \int_{\tau} \Pr(\hat{\boldsymbol{x}} \neq \boldsymbol{x} \,|\, \tau)\, p(\tau)\, d\tau. \tag{2.75}$$

Bei der üblichen Monte–Carlo–Methode [BSMM99] werden unwesentliche Ereignisse sehr häufig simuliert, weil die Wahrscheinlichkeit des Auftretens eines Fehlers bei großen SNR sehr gering ist.

Im folgenden wird der quadratische euklidische Abstand ${d_E}^2(\boldsymbol{x}, \boldsymbol{y})$ zwischen einer Sendefolge $\boldsymbol{x} \in \mathcal{X}$ und einem vom Empfangsvektor $\boldsymbol{r}$ abhängigen Vektor $\boldsymbol{y} \in \mathbb{R}^n$ mit $\Delta^2 \in \mathbb{R}$ bezeichnet und als Integrationsvariable $\tau = \Delta^2$ in Gleichung (2.75) verwendet. Diese Simulationsmethode gestattet, daß die wichtigsten Fehlerereignisse stark berücksichtigt werden. Bei der Übertragung über einen AWGN–Kanal ist Δ^2 eine Zufallsvariable, dessen Wahrscheinlichkeitsdichte $p(\tau = \Delta^2)$ vom aktuellen SNR pro empfangenem Infobit $E_b \,/\, N_0$ abhängt und analytisch berechnet werden kann.

Das Ziel in diesem Abschnitt ist, Zufallsvektoren zu erzeugen, die einen bestimmten Betrag Δ besitzen, wobei alle Winkel gleichverteilt sind. Dies wird erreicht, indem ein gaußverteilter Vektor erzeugt wird, dessen Länge in nachhinein auf den gewünschten Wert Δ normiert wird.

Eine Zufallsvariable $\Phi \in \mathbb{R}_0^+$ wird definiert als:

$$\Phi = \sum_{i=0}^{n-1} {\phi_i}^2, \tag{2.76}$$

wobei ϕ_i der Zufallswert einer gaußverteilten Zufallsvariablen mit Erwartungswert $\mu_\phi = 0$ bezeichnet. Dabei werden alle n verschiedenen Zufallswerte $\phi_i \in \mathbb{R}$ mittels des gaußverteilten Zufallszahlengenerators des letzten Unterabschnittes erzeugt.

Im zeitdiskreten Kanal wird zu jeder Komponente $x_i \in \{\pm 1\}$ einer Sendefolge $\boldsymbol{x}$ eine normierte Zufallsvariable addiert:

$$r_i = x_i + \Lambda\, \phi_i, \tag{2.77}$$

wobei der Normierungsfaktor Λ folgendermaßen angegeben wird:

$$\Lambda = \sqrt{\frac{\Delta^2}{\Phi}}. \tag{2.78}$$

Am Ausgang des Decodierers werden sowohl eine simulierte Wortfehlerrate P_{Δ^2} (WER, *Word Error Rate*, auch Blockfehlerrate oder Blockfehlerwahrscheinlichkeit (im asymptotischen Fall) genannt) als auch eine simulierte Bitfehlerrate p_{Δ^2} (BER, *Bit Error Rate*, ebenfalls Bitfehlerwahrscheinlichkeit (im asymptotischen Fall) genannt) gemessen, die vom quadratischen Abstand Δ^2 abhängig sind, der über einen gewissen Bereich variiert wird. Für Beispiele von Kurven von P_{Δ^2} wird auf Abbildungen 5.1 und 5.5 verwiesen.

Für die Berechnung von Gleichung (2.75) wurden Werte P_{Δ^2} bzw. p_{Δ^2} benötigt, die nicht simuliert wurden. Diese Werte wurden im logarithmischen Maß nach den Methoden von [PTVF92] linear interpoliert bzw. extrapoliert.

Bei einem gegebenen $E_b \,/\, N_0$ kann die WER P_σ bzw. BER p_σ aus den Kurven wie folgt berechnet werden:

$$P_\sigma = \int_0^\infty P_{\Delta^2}\ \chi(\Delta^2, n, \sigma)\ d\Delta^2, \tag{2.79a}$$

$$p_\sigma = \int_0^\infty p_{\Delta^2}\ \chi(\Delta^2, n, \sigma)\ d\Delta^2. \tag{2.79b}$$

Dabei ist

$$p(\tau = \Delta^2) = \chi(\varsigma, n, \sigma) = \frac{\varsigma^{\frac{n}{2}-1}\, e^{-\frac{\varsigma}{2\sigma^2}}}{\sigma^n\ 2^{\frac{n}{2}}\ \Gamma\left(\frac{n}{2}\right)}, \quad \varsigma \geq 0 \tag{2.80}$$

die χ^2-Wahrscheinlichkeitsdichte mit dem Freiheitsgrad n [Pro95], wobei

$$\Gamma(\varsigma) = \int_0^\infty e^{-\tau} \tau^{\varsigma-1} d\tau \tag{2.81}$$

die Gamma–Funktion [Bar97, BSMM99, Stö99] und

$$\sigma^2 = \frac{N_0}{2\,E_c} = \frac{1}{2\,R\,10^{0.1\frac{E_b}{N_0}[dB]}} \tag{2.82}$$

die Varianz bezeichnet.

In Kapitel 5 sind Abbildungen 5.7 und 5.6 Beispiele für die Anwendung von Gleichung (2.79). Für die Berechnung dieser Integration wurde ein Romberg–Verfahren [BSMM99] verwendet, das in [PTVF92] in der Programmiersprache C implementiert wurde.

Der wesentliche Vorteil dieser Simulationsmethode im Vergleich zu einer gewöhnlichen Monte–Carlo–Methode [BSMM99] ist, daß sowohl Interpolationen als auch Extrapolationen hervorragende Ergebnisse bezüglich der Genauigkeit ermöglichen, wie in Kapitel 5 gezeigt wird.

2.9 Simulationsschranke für SDMLD

Damit die Güte eines beliebigen Decodieralgorithmus Ψ beurteilt werden kann, sollte seine WER P_Ψ mit der WER P_{MLD} einer SDMLD verglichen werden. Weil die SDMLD üblicherweise ausschließlich für relativ kurze Codes simuliert werden kann, wird eine untere Schranke P_{LB} (LB, *Lower Bound*) für die WER einer SDMLD verwendet.

Die Grundidee für diese untere Schranke P_{LB} ist folgende: Wird von irgendeinem Decodierer irgendeine geschätzte Sendefolge $\hat{\boldsymbol{x}}$ gefunden, die besser ist als die Sendefolge $\boldsymbol{x}$, dann würde auch ein ML–Decodierer einen Fehler begehen. Wird nur diese Fälle als Fehler gezählt, ergibt sich eine untere Simulationsschranke P_{LB} für die WER P_{MLD} einer SDMLD.

Diese untere Simulationsschranke P_{LB} kann folgendermaßen beschrieben werden: Vorausgesetzt werden eine Sendefolge $\boldsymbol{x}$, ein vom Empfangsvektor $\boldsymbol{r}$ abhängiger Vektor $\boldsymbol{y}$ und ein Decodieralgorithmus Ψ, der eine geschätzte Sendefolge $\hat{\boldsymbol{x}}$ an seinem Ausgang liefert. Gemäß [Luc97, Bos98] hat eine SDMLD eine WER P_{MLD}, die immer größer oder gleich ist als die WER P_{LB} eines Decodierers Ψ_{LB}, der unter Kenntnis der Sendefolge $\boldsymbol{x}$, des Vektors $\boldsymbol{y}$ und des Ergebnisses $\hat{\boldsymbol{x}}$ des Decodieralgorithmus Ψ folgende Sendefolge $\hat{\boldsymbol{x}}_{\boldsymbol{LB}}$ an seinem Ausgang liefert:

$$\hat{\boldsymbol{x}} = \boldsymbol{x} \quad \Longrightarrow \quad \hat{\boldsymbol{x}}_{\boldsymbol{LB}} = \hat{\boldsymbol{x}} = \boldsymbol{x} = \boldsymbol{x}_{\boldsymbol{ML}} \tag{2.83a}$$

$$\hat{\boldsymbol{x}} \neq \boldsymbol{x}, \quad \langle \hat{\boldsymbol{x}}, \boldsymbol{y} \rangle > \langle \boldsymbol{x}, \boldsymbol{y} \rangle \quad \Longrightarrow \quad \hat{\boldsymbol{x}}_{\boldsymbol{LB}} = \hat{\boldsymbol{x}} \tag{2.83b}$$

$$\hat{\boldsymbol{x}} \neq \boldsymbol{x}, \quad \langle \hat{\boldsymbol{x}}, \boldsymbol{y} \rangle \leq \langle \boldsymbol{x}, \boldsymbol{y} \rangle \quad \Longrightarrow \quad \hat{\boldsymbol{x}}_{\boldsymbol{LB}} = \boldsymbol{x} \tag{2.83c}$$

Offensichtlich ist der Decodierer Ψ_{LB} lediglich für Simulationszwecke geeignet, weil die Sendefolge $\boldsymbol{x}$ auf der Empfangsseite in der Praxis üblicherweise nicht bekannt ist.

Im Fall (2.83a) findet der Decodieralgorithmus Ψ bzw. der Decodierer Ψ_{LB} die richtige Sendefolge $\boldsymbol{x}$, und eine SDMLD kann selbstverständlich nicht besser sein.

Im Fall (2.83b) ist das Skalarprodukt $\langle \hat{\boldsymbol{x}}, \boldsymbol{y} \rangle$ zwischen $\hat{\boldsymbol{x}}$ und $\boldsymbol{y}$ größer als das Skalarprodukt $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ zwischen $\boldsymbol{x}$ und $\boldsymbol{y}$. So findet eine SDMLD entweder dieselbe geschätzte Sendefolge $\hat{\boldsymbol{x}}$ oder eine andere geschätzte Sendefolge $\hat{\boldsymbol{x}}'$, die sogar ein größeres Skalarprodukt $\langle \hat{\boldsymbol{x}}', \boldsymbol{y} \rangle$ zu $\boldsymbol{y}$ aufweist. In den beiden Fällen begeht eine SDMLD bzw. ein Decodierer Ψ_{LB} einen Blockfehler.

Im Fall (2.83c) ist das Skalarprodukt $\langle \hat{\boldsymbol{x}}, \boldsymbol{y} \rangle$ zwischen $\hat{\boldsymbol{x}}$ und $\boldsymbol{y}$ kleiner oder gleich als das Skalarprodukt $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ zwischen $\boldsymbol{x}$ und $\boldsymbol{y}$. Hier wird angenommen,

daß der Decodierer Ψ_{LB} keinen Fehler begeht, obwohl das Ergebnis einer SDMLD nicht vorhersagbar ist. Damit ist die WER P_{LB} des Decodierers Ψ_{LB} niemals größer als die WER P_{MLD} einer SDMLD.

Durch die verschiedenen Simulationen in Kapitel 5 wurde festgestellt, daß je näher der Decodieralgorithmus Ψ an die SDMLD herankommt, um so dichter und genauer wird die untere Schranke P_{LB}. Führt der Decodieralgorithmus Ψ eine SDMLD durch, stimmt diese untere Schranke P_{LB} genau mit der WER P_{MLD} einer SDMLD überein.

Kapitel 3

Akzeptanzkriterien

Ein Akzeptanzkriterium ist eine Bedingung Ξ, die es erlaubt, daß eine iterative Decodierung möglicherweise früher abgebrochen werden kann. Eine derartige Bedingung wird ebenfalls Entscheidungskriterium [KNIH94], Abbruchkriterium [Nil94, KNH97], Abbruchregel [Bar93, BGW93, BGW97] oder einfach Test [SJ98] genannt.

Falls diese Bedingung Ξ häufig erfüllt ist, besteht ein Vorteil eines solchen Tests darin, daß die durchschnittliche Anzahl von Iterationen sowohl eines optimalen als auch eines suboptimalen Decodieralgorithmus wesentlich reduziert wird.

Gemäß Bossert [Bos98] ist ein Soft–Decision–Maximum–Likelihood–Akzeptanzkriterium eine Bedingung Ξ zwischen einer geschätzten Sendefolge $\hat{\boldsymbol{x}}$ und dem Vektor $\boldsymbol{y}$. Diese Bedingung Ξ gewährleistet, daß

- es sich um die wahrscheinlichste Sendefolge $\boldsymbol{x}_{ML}$ handelt,
- die Bedingung Ξ für keine andere geschätzte Sendefolge $\hat{\boldsymbol{x}}'$ gilt und
- diese wahrscheinlichste Sendefolge $\boldsymbol{x}_{ML}$ von einem noch zu definierenden Decodieralgorithmus Ψ gefunden wird.

Falls Ξ nicht erfüllt wird, dann kann keine Aussage getroffen werden.

Im ersten Abschnitt dieses Kapitels werden einige Akzeptanzkriterien basierend auf einem Codewort für eine SDMLD hergeleitet. Danach werden die Akzeptanzkriterien auf Basis von mehreren Codewörtern behandelt. Die Güte aller Akzeptanzkriterien wird durch Simulationen in Kapitel 5 untersucht. Dort werden die Akzeptanzkriterien zusammen mit den im nächsten

Kapitel vorgestellten Decodierverfahren verwendet, damit die gesamte durchschnittliche Komplexität der Decodierung verringert werden kann.

3.1 Akzeptanzkriterien basierend auf einem Codewort

In den folgenden Unterabschnitten werden fünf verschiedene Akzeptanzkriterien auf Basis eines Codewortes für eine SDMLD hergeleitet. Sie werden in Kapitel 5 miteinander verglichen.

3.1.1 Syndrom–Test

Das einfachste und sicherlich bekannteste Akzeptanzkriterium ist der sogenannte Syndrom–Test [God91], der das bereits in Gleichung (2.7) erwähnte Konzept eines Syndroms verwendet:

$$\boldsymbol{s} = \boldsymbol{v}\,\boldsymbol{H}^{\mathrm{T}}. \tag{2.7}$$

Vorausgesetzt werden ein $(n,k)_2$-Code $\mathcal{C}$ mit einer Prüfmatrix $\boldsymbol{H} \in \mathbb{F}_2^{n-k,n}$ und ein Vektor $\boldsymbol{v} \in \mathbb{F}_2^n$ nach Gleichung (2.37). Wenn die Bedingung

$$\Xi_{\text{Syndrom}}: \quad \boldsymbol{v}\,\boldsymbol{H}^{\mathrm{T}} = \boldsymbol{0} \tag{3.1}$$

für diesen Vektor $\boldsymbol{v}$ gilt, dann ist $\boldsymbol{v}$ das wahrscheinlichste gesendete Codewort $\boldsymbol{c}_{ML}$.

Dieses Akzeptanzkriterium kann gleichfalls wie folgt verstanden werden: Wenn das Syndrom $\boldsymbol{s}$ exakt das Nullwort $\boldsymbol{0}$ ist, dann ist der Vektor $\boldsymbol{v}$ ein Codewort aus $\mathcal{C}$, und zwar das wahrscheinlichst gesendete Codewort $\boldsymbol{c}_{ML}$. Deswegen wird diese Bedingung auch als Codewort–Test bezeichnet [SJ98].

3.1.2 Hyperkugel–Test

Ein anderes Akzeptanzkriterium beruht auf einem Decodierbereich innerhalb einer Hyperkugel [Ric95]. Für einen modulierten $(3,2)_2$-Code $\mathcal{X}$ zeigt Abbildung 3.1 die dreidimensionale Darstellung einer solchen Region $\mathcal{R}_S(\boldsymbol{x})$.

Jede n-dimensionale euklidische Hyperkugel $\mathcal{R}_S(\boldsymbol{x})$ ist durch ihren Mittelpunkt $\boldsymbol{x} \in \mathcal{X}$ und ihren Radius $\rho_S \in \mathbb{R}_0^+$ definiert. Diese Hyperkugeln

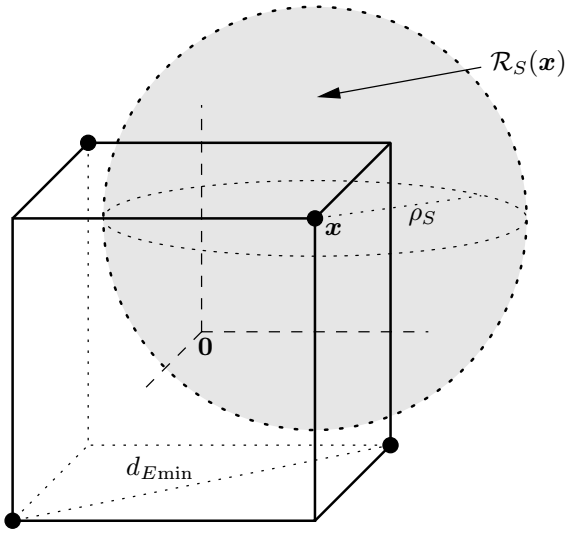

Abbildung 3.1: Dreidimensionale euklidische Hyperkugel $\mathcal{R}_S(\boldsymbol{x})$.

überlappen sich nicht, wenn:

$$\rho_S \leq \frac{d_{E\min}}{2}. \tag{3.2}$$

Zu einem Vektor $\boldsymbol{y}$ wird die Sendefolge $\boldsymbol{x}$ nur dann gewählt, falls $\boldsymbol{y}$ innerhalb der Hyperkugel $\mathcal{R}_S(\boldsymbol{x})$ liegt:

$$\boldsymbol{y} \in \mathcal{R}_S(\boldsymbol{x}) \Longleftrightarrow \hat{\boldsymbol{x}} = \boldsymbol{x}. \tag{3.3}$$

Der Hyperkugel–Test kann folgendermaßen formuliert werden: Vorausgesetzt werden ein modulierter Code $\mathcal{X}$ mit Minimalabstand $d_{H\min}$ bzw. euklidischem Minimalabstand $d_{E\min}$, ein vom Empfangsvektor $\boldsymbol{r}$ abhängiger Vektor $\boldsymbol{y}$ und eine geschätzte Sendefolge $\hat{\boldsymbol{x}} \in \mathcal{X}$. Wird die Bedingung

$$\Xi_{\text{Hyperkugel}}: \quad \langle \boldsymbol{y}^0 - \hat{\boldsymbol{x}}, \boldsymbol{y}^0 - \hat{\boldsymbol{x}} \rangle \leq {\rho_S}^2 \tag{3.4}$$

erfüllt, dann ist $\hat{\boldsymbol{x}}$ die wahrscheinlichste Sendefolge $\boldsymbol{x}_{ML}$. Dabei wird der Einheitsvektor $\boldsymbol{y}^0$ gemäß Gleichung (2.52a) angegeben.

In Anhang A.5 wird gezeigt, daß Bedingung (3.4) wie folgt umformuliert werden kann:

$$\Xi_{\text{Hyperkugel}}: \quad \sum_i (y_i - \|\boldsymbol{y}\|\, \hat{x}_i)^2 \leq \|\boldsymbol{y}\|^2 \, d_{H\min}, \tag{3.5}$$

wobei $\|\boldsymbol{y}\|$ nach Gleichung (2.50) unabhängig von $\hat{\boldsymbol{x}}$ ist und damit als Konstante betrachtet werden kann:

$$\|\boldsymbol{y}\| = \sqrt{\sum_j {y_j}^2}. \tag{2.50}$$

Als Beispiel für die Anwendung des Hyperkugel–Tests wird auf [Ric95] verwiesen.

3.1.3 Hyperkreiskegel–Test

Da die in Abschnitt 2.6 vorgestellte Voronoi–Region $\mathcal{V}(\boldsymbol{x})$ einer Sendefolge $\boldsymbol{x}$ eine konvexe Hyperpyramide mit einer Spitze am Ursprung $\mathbf{0}$ ist, kann – wie in [For66a] vorgeschlagen wurde – der Decodierbereich von einer Hyperkugel zu einem Hyperkreiskegel erweitert werden.

Die dreidimensionale Darstellung einer solchen Region $\mathcal{R}_C(\boldsymbol{x})$ ist für einen modulierten $(3, 2)_2$-Code $\mathcal{X}$ in Abbildung 3.2 veranschaulicht. Der Decodierbereich ist ein n-dimensionaler euklidischer Hyperkreiskegel $\mathcal{R}_C(\boldsymbol{x})$ mit Spitze am Ursprung $\mathbf{0}$ und unendlicher Höhe um die Sendefolge $\boldsymbol{x}$ zentriert.

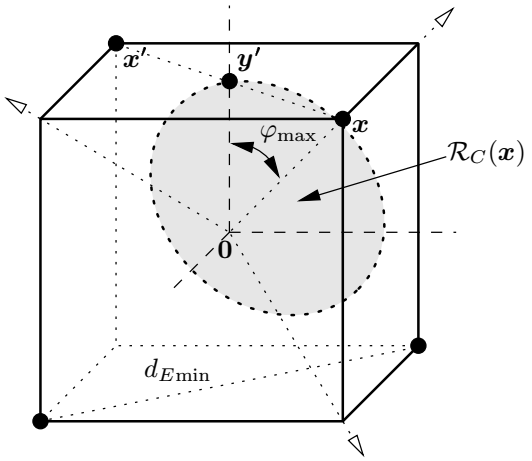

Abbildung 3.2: Dreidimensionaler euklidischer Hyperkreiskegel $\mathcal{R}_C(\boldsymbol{x})$.

Zur Herleitung dieses Akzeptanzkriteriums wird $d_E(\boldsymbol{x}, \boldsymbol{x'}) = d_{E\min}$ angenommen. Damit ist der Hyperkreiskegel durch den Winkel $\varphi_{\max}$ zwischen $\boldsymbol{x}$

und einem im Mittelpunkt zwischen $\boldsymbol{x}$ und $\boldsymbol{x}'$ liegenden Vektor $\boldsymbol{y}' \in \mathbb{R}^n$ definiert. Dieser Winkel wird gemäß Gleichung (2.56) folgendermaßen bestimmt:

$$\cos \varphi_{\max} = \frac{\langle \boldsymbol{x}, \boldsymbol{y}' \rangle}{\|\boldsymbol{x}\| \, \|\boldsymbol{y}'\|}. \tag{3.6}$$

In Anhang A.6 wird gezeigt, daß für einen modulierten $(n,k)_2$-Code $\mathcal{X}$ mit Minimalabstand $d_{H\min}$ aus Gleichung (3.6)

$$\cos \varphi_{\max} = \sqrt{1 - \frac{d_{H\min}}{n}} \tag{3.7}$$

folgt.

Ein Vektor $\boldsymbol{y}$ liegt innerhalb des Hyperkreiskegels $\mathcal{R}_C(\boldsymbol{x})$, falls der Winkel $\varphi(\boldsymbol{x}, \boldsymbol{y})$ zwischen $\boldsymbol{x}$ und $\boldsymbol{y}$ kleiner oder gleich als $\varphi_{\max}$ ist:

$$\varphi(\boldsymbol{x}, \boldsymbol{y}) \leq \varphi_{\max} \Longleftrightarrow \boldsymbol{y} \in \mathcal{R}_C(\boldsymbol{x}). \tag{3.8}$$

Da

$$\varphi(\boldsymbol{x}, \boldsymbol{y}) \leq \varphi_{\max} \Longleftrightarrow \cos \varphi(\boldsymbol{x}, \boldsymbol{y}) \geq \cos \varphi_{\max}, \; \forall \, \varphi(\boldsymbol{x}, \boldsymbol{y}), \varphi_{\max} \in [0, \pi], \tag{3.9}$$

kann der Hyperkreiskegel–Test wie folgt beschrieben werden: Vorausgesetzt werden ein modulierter $(n,k)_2$-Code $\mathcal{X}$ mit Minimalabstand $d_{H\min}$ bzw. euklidischem Minimalabstand $d_{E\min}$, eine geschätzte Sendefolge $\hat{\boldsymbol{x}} \in \mathcal{X}$ und ein vom Empfangsvektor $\boldsymbol{r}$ abhängiger Vektor $\boldsymbol{y}$. Wenn die Bedingung

$$\Xi_{\text{Hyperkreiskegel}}: \quad \frac{\langle \hat{\boldsymbol{x}}, \boldsymbol{y} \rangle}{\|\hat{\boldsymbol{x}}\| \, \|\boldsymbol{y}\|} \geq \sqrt{1 - \frac{d_{H\min}}{n}} \tag{3.10}$$

erfüllt wird, dann ist $\hat{\boldsymbol{x}}$ die wahrscheinlichste Sendefolge $\boldsymbol{x}_{ML}$.

In Anhang A.7 wird diese Bedingung in folgender Form abgewandelt:

$$\Xi_{\text{Hyperkreiskegel}}: \quad \sum_i \hat{x}_i \, y_i \geq \|\boldsymbol{y}\| \sqrt{n - d_{H\min}}. \tag{3.11}$$

Einige Algorithmen, die den Hyperkreiskegel–Test benutzen, werden beispielsweise in [God91, HHC91, Bar93] behandelt.

3.1.4 GMD–Test

Im Jahre 1966 wurde das Konzept einer verallgemeinerten Minimalabstand–Decodierung (GMD, *Generalized Minimum Distance*) von Forney eingeführt [For66b]. Seine Arbeit wurde für q-stufige Symbole in [ES76, YC80] analysiert und erweitert. Der GMD Algorithmus verwendet ein Akzeptanzkriterium, das im folgenden hergeleitet wird.

Vorausgesetzt werden ein $(n,k)_2$-Code $\mathcal{C}$ mit Minimalabstand $d_{H\min}$, ein geschätztes Codewort $\hat{\boldsymbol{c}} \in \mathcal{C}$ bzw. eine geschätzte Sendefolge $\hat{\boldsymbol{x}} \in \mathcal{X}$ und zwei vom Empfangsvektor $\boldsymbol{r}$ abhängige Vektoren $\boldsymbol{v} \in \mathbb{F}_2^n$ nach Gleichung (2.37) und $\boldsymbol{y} \in \mathbb{R}^n$ nach Gleichung (2.36).

Da der kleinste Hammingabstand zwischen zwei Codewörtern aus $\mathcal{C}$ gleich dem Minimalabstand $d_{H\min}$ ist, gilt folgende Ungleichung für höchstens ein geschätztes Codewort $\hat{\boldsymbol{c}}$:

$$d_H(\hat{\boldsymbol{c}}, \boldsymbol{v}) \leq \left\lfloor \frac{d_{H\min} - 1}{2} \right\rfloor . \tag{3.12}$$

Für einen nach Gleichung (2.53a) normierten Vektor $\tilde{\boldsymbol{y}} = (\tilde{y}_0, \ldots, \tilde{y}_{n-1})$, $|\tilde{y}_i| \leq 1$, $0 \leq i < n$ gilt gemäß Gleichung (2.59):

$$d_E{}^2(\hat{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \leq 4\, d_H(\hat{\boldsymbol{c}}, \boldsymbol{v}). \tag{3.13}$$

Damit existiert höchstens eine geschätzte Sendefolge $\hat{\boldsymbol{x}}$, für die

$$d_E{}^2(\hat{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \leq 2\,(d_{H\min} - 1) \tag{3.14}$$

gültig ist.

Des weiteren gilt nach Gleichung (2.62b) für den normierten Vektor $\tilde{\boldsymbol{y}}$:

$$d_E{}^2(\hat{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \leq n - 2\,\langle \hat{\boldsymbol{x}}, \tilde{\boldsymbol{y}} \rangle + n \leq 2\,(d_{H\min} - 1). \tag{3.15}$$

Daraus folgt:

$$\begin{aligned} -2\,\langle \hat{\boldsymbol{x}}, \tilde{\boldsymbol{y}} \rangle + 2\,n &\leq 2\,(d_{H\min} - 1) \\ -\langle \hat{\boldsymbol{x}}, \tilde{\boldsymbol{y}} \rangle &\leq -n + d_{H\min} - 1 \\ \langle \hat{\boldsymbol{x}}, \tilde{\boldsymbol{y}} \rangle &\geq n - d_{H\min} + 1. \end{aligned} \tag{3.16}$$

Schließlich kann das GMD–Akzeptanzkriterium folgendermaßen formuliert werden: Wenn die Bedingung

$$\Xi_{\mathrm{GMD}}\colon \quad \langle \hat{\boldsymbol{x}}, \tilde{\boldsymbol{y}} \rangle > n - d_{H\min} \tag{3.17}$$

erfüllt wird, dann ist $\hat{\boldsymbol{x}}$ die wahrscheinlichste Sendefolge $\boldsymbol{x}_{ML}$.

Unter Anwendung der Gleichungen (2.54) und (2.53c) wird diese Bedingung ausgeschrieben:

$$\Xi_{\mathrm{GMD}}: \quad \sum_i \hat{x}_i \, y_i > (n - d_{H\min}) \max_{0 \le i < n} \beta_i. \tag{3.18}$$

Gemäß [Bla83] definiert diese Akzeptanzbedingung einen Decodierbereich $\mathcal{R}_G(\boldsymbol{x}'')$ einer Sendefolge $\boldsymbol{x}''$, die anhand der Abbildung 3.3 für einen modulierten $(3,2)_2$-Code $\mathcal{X}$ im dreidimensionalen euklidischen Vektorraum $\mathbb{R}^3$ verdeutlicht ist.

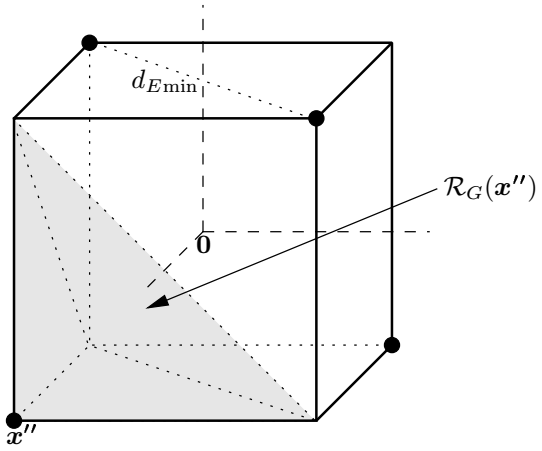

Abbildung 3.3: Dreidimensionale euklidische GMD–Region $\mathcal{R}_G(\boldsymbol{x}'')$.

3.1.5 Akzeptanzkriterium nach Taipale und Pursley

Ein anderes Akzeptanzkriterium wurde von Enns [Enn87] und Taipale und Pursley [TP89, TP91] unabhängig voneinander vorgeschlagen, dessen von den Entscheidungsräumen überdeckte Fläche innerhalb des Hyperwürfels größer als bei dem GMD–Test ist, wie in [HT90, Tol96] ausführlich analysiert wurde.

Angenommen werden ein modulierter $(n,k)_2$-Code $\mathcal{X}$ mit Minimalabstand $d_{H\min}$, ein geschätztes Codewort $\hat{\boldsymbol{c}} \in \mathcal{C}$ bzw. eine geschätzte Sendefolge $\hat{\boldsymbol{x}} \in \mathcal{X}$, ein vom Empfangsvektor $\boldsymbol{r}$ abhängiger Vektor $\boldsymbol{y}$ bzw. $\boldsymbol{v}$ bzw. $\boldsymbol{z}$ und ein Konkurrenz–Vektor $\boldsymbol{v}' \in \mathbb{F}_2^n$, dessen entsprechender Konkurrenz–Vektor

$\boldsymbol{z}' \in \{\pm 1\}^n$ im euklidischen Vektorraum $\mathbb{R}^n$ den geringsten euklidischen Abstand $d_E(\boldsymbol{z}', \boldsymbol{y})$ zu $\boldsymbol{y}$ aufweist unter der Voraussetzung, daß:

$$d_H(\hat{\boldsymbol{c}}, \boldsymbol{v}') = |\mathcal{D}_1(\hat{\boldsymbol{x}}, \boldsymbol{z}')| = d_{H\min}. \tag{3.19}$$

wobei die Indexmenge $\mathcal{D}_1(\hat{\boldsymbol{x}}, \boldsymbol{z}')$ nach Gleichung (2.42) angegeben wird.

Überdies wird zur Herleitung dieses Akzeptanzkriteriums angenommen, daß weniger als $d_{H\min}$ Fehler aufgetreten sind:

$$d_H(\hat{\boldsymbol{c}}, \boldsymbol{v}) = |\mathcal{D}_1(\hat{\boldsymbol{x}}, \boldsymbol{y})| < d_{H\min}. \tag{3.20}$$

Damit der euklidische Abstand $d_E(\boldsymbol{z}', \boldsymbol{y})$ minimal wird, sollen die kleinsten Zuverlässigskeiten β_i genommen werden. Somit wird die Indexmenge der $d_{H\min} - |\mathcal{D}_1(\hat{\boldsymbol{x}}, \boldsymbol{y})|$ unzuverlässigsten Stellen, die nicht zu $\mathcal{D}_1(\hat{\boldsymbol{x}}, \boldsymbol{y})$ gehören, benötigt. Gemäß der Notationen der Unterabschnitten 2.4.1 und 2.4.2 kann diese Indexmenge als $\mathcal{D}_0(\hat{\boldsymbol{x}}, \boldsymbol{y})^{(\delta)}$ geschrieben werden, wobei

$$\delta = d_{H\min} - |\mathcal{D}_1(\hat{\boldsymbol{x}}, \boldsymbol{y})| \tag{3.21}$$

ist.

Aus $\hat{\boldsymbol{x}}$ und $\boldsymbol{z}$ kann der Konkurrenz–Vektor $\boldsymbol{z}'$ durch das Umdrehen der Stellen von $\mathcal{D}_0(\hat{\boldsymbol{x}}, \boldsymbol{y})^{(\delta)}$ konstruiert werden [Nil94]. Daraus folgt:

$$i \in \mathcal{D}_1(\hat{\boldsymbol{x}}, \boldsymbol{y}) \implies \hat{x}_i \neq z_i,\ z_i = z'_i, \tag{3.22a}$$

$$i \in \mathcal{D}_0(\hat{\boldsymbol{x}}, \boldsymbol{y})^{(\delta)} \implies \hat{x}_i = z_i,\ z_i \neq z'_i, \tag{3.22b}$$

$$i \notin \mathcal{D}_1(\hat{\boldsymbol{x}}, \boldsymbol{y}) \cup \mathcal{D}_0(\hat{\boldsymbol{x}}, \boldsymbol{y})^{(\delta)} \implies \hat{x}_i = z_i,\ z_i = z'_i. \tag{3.22c}$$

Damit kann dieses Akzeptanzkriterium wie folgt beschrieben werden: Wenn die Bedingung

$$\Xi_{\mathrm{TP}}: \quad d_E^2(\hat{\boldsymbol{x}}, \boldsymbol{y}) \leq d_E^2(\boldsymbol{z}', \boldsymbol{y}) \tag{3.23}$$

erfüllt wird, die äquivalent zu Ungleichung (2.71) ist, dann ist $\hat{\boldsymbol{x}}$ die wahrscheinlichste Sendefolge $\boldsymbol{x}_{ML}$.

In Anhang A.8 wird bewiesen, daß diese Bedingung folgendermaßen umgeformt werden kann:

$$\Xi_{\mathrm{TP}}: \quad d_\beta(\hat{\boldsymbol{x}}, \boldsymbol{y}) \leq \sum_{i \in \mathcal{D}_0(\hat{\boldsymbol{x}}, \boldsymbol{y})^{(\delta)}} \beta_i, \tag{3.24}$$

wobei $d_\beta(\hat{\boldsymbol{x}}, \boldsymbol{y})$ nach Gleichung (2.49) und δ nach Gleichung (3.21) angegeben wird.

Ist der Konkurrenz–Vektor $\boldsymbol{v}'$ ein Codewort aus $\mathcal{C}$, dann ist entweder $\hat{\boldsymbol{c}}$ oder $\boldsymbol{v}'$ das wahrscheinlichst gesendete Codewort $\boldsymbol{c}_{ML}$ [Nil94].

Dieses Akzeptanzkriterium wird ebenfalls polyedrischer Hyperkreiskegel–Test genannt [SJ98].

In der Praxis benötigen viele Algorithmen [GA88, GA89, God91, Bar93, BGW94, BGW97, BD97] lediglich ein Akzeptanzkriterium, das testen soll, ob der Vektor $\boldsymbol{v}$ innerhalb der Voronoi–Region $\mathcal{V}(\mathbf{0})$ des Nullwortes $\mathbf{0}$ liegt [BGW93]. In diesem Fall wird die Berechnung von Gleichung (3.24) vereinfacht, denn die geschätzte Sendefolge $\hat{\boldsymbol{x}}$ im euklidischen Vektorraum entspricht stets dem Nullwort $\mathbf{0}$ in $\mathbb{F}_2^n$ [Bar93].

3.2 Akzeptanzkriterien basierend auf mehreren Codewörtern

Das im vorherigen Unterabschnitt erörterte Akzeptanzkriterium nach Taipale und Pursley [TP89, TP91] ist die bestmögliche Bedingung Ξ zwischen einer geschätzten Sendefolge $\hat{\boldsymbol{x}}$ und dem Vektor $\boldsymbol{y}$, die hergeleitet werden kann. Somit wäre die Suche nach noch besseren Akzeptanzkriterien längst abgeschlossen. Trotzdem haben Kaneko et al. im Jahre 1994 ein verbessertes Akzeptanzkriterium vorgestellt [KNIH94], das auf der Kenntnis von zwei Codewörtern anstelle von nur einem beruht. Damit werden die Entscheidungsräume um die Codewörter größer als beim Akzeptanzkriterium nach Taipale und Pursley [Bos98].

Kurz danach haben Kasami et al. ein neues Akzeptanzkriterium auf Basis von mehreren Codewörtern vorgeschlagen [KTK$^+$95, KKTL95a, KKTL95b], so daß das Akzeptanzkriterium nach Taipale und Pursley [TP89, TP91] als ein spezieller Fall dieses neuen Akzeptanzkriteriums interpretiert werden kann. Außerdem sind die Entscheidungsräume noch größer [KTK$^+$95] als beim Akzeptanzkriterium nach Kaneko [KNIH94].

Im nächsten Unterabschnitt wird zuerst eine vereinheitlichte Darstellung eingeführt. Es folgt die Herleitung einer allgemeinen unteren Schranke für den minimalen gewichteten Hammingabstand bei SDMLD. Diese untere Schranke ist auch als Akzeptanzkriterium nach Kasami bekannt [Bos98]. In Unterabschnitt 3.2.3 wird ein Algorithmus für dieses Akzeptanzkriterium auf der

Basis von zwei Codewörtern vorgeschlagen. Anschließend wird das Akzeptanzkriterium basierend auf mehr als zwei Codewörtern diskutiert.

3.2.1 Notation

Vorausgesetzt werden ein $(n, k, d_{H\min})_2$-Code $\mathcal{C}$ mit euklidischem Minimalabstand $d_{E\min}$ nach Gleichung (2.61) und Gewichtsprofil $\mathcal{W}$ und ein iterativer, beliebiger Decodierer Ψ, der in jeder Iteration i ein geschätztes Codewort $\hat{\boldsymbol{c}}_i$ von $\mathcal{C}$ bzw. eine geschätzte Sendefolge $\hat{\boldsymbol{x}}_i$ des entsprechenden modulierten Codes $\mathcal{X}$ liefert.

Mit $h \in \mathbb{N}^+$ wird die Anzahl der geschätzten Codewörtern $\hat{\boldsymbol{c}}_1, \hat{\boldsymbol{c}}_2, \ldots, \hat{\boldsymbol{c}}_h \in \mathcal{C}$ bzw. Sendefolgen $\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \ldots, \hat{\boldsymbol{x}}_h \in \mathcal{X}$ bezeichnet, wobei $\mathbb{N}^+$ die Menge der natürlichen Zahlen ohne Null ist. Dabei steht $\hat{\boldsymbol{x}}_h$ für diejenige geschätzte Sendefolge, die in der letzten Iteration von Ψ erzeugt wurde und im Moment getestet wird, $\hat{\boldsymbol{x}}_{h-1}$ für die beste geschätzte Sendefolge, die von der ersten Iteration an bis zur vorletzten erzeugt wurde, usw. und $\hat{\boldsymbol{x}}_1$ für die beste geschätzte Sendefolge, die innerhalb aller Iterationen außer den $h-1$ letzten erzeugt wurde [KKTL95a].

Zur Vereinfachung der Notation wird die Menge der h besten geschätzten Codewörter mit $\mathcal{C}^h$ bezeichnet:

$$\mathcal{C}^h = \{\hat{\boldsymbol{c}}_1, \hat{\boldsymbol{c}}_2, \ldots, \hat{\boldsymbol{c}}_h\}. \tag{3.25}$$

Entsprechend steht $\mathcal{X}^h$ für die Menge der h besten geschätzten Sendefolgen:

$$\mathcal{X}^h = \{\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \ldots, \hat{\boldsymbol{x}}_h\}. \tag{3.26}$$

Darüber hinaus bezeichnet $\hat{\boldsymbol{x}} \in \mathcal{X}$ diejenige geschätzte Sendefolge, die im Moment getestet wird.

Die Menge der 2^h verschiedenen binären Muster der Länge $h \in \mathbb{N}^+$ wird mit $\mathbb{B}^h$ bezeichnet:

$$\mathbb{B}^h = \{\overbrace{\underbrace{0\cdots00}_{h \text{ Stellen}}, 0\cdots01, \ldots, 1\cdots11}^{2^h \text{ binäre Muster}}\}, \quad \left|\mathbb{B}^h\right| = 2^h. \tag{3.27}$$

Selbstverständlich können diese binären Muster als Vektoren $\boldsymbol{v}$ mit Komponenten aus $\mathbb{F}_2$ verstanden werden. In diesem Fall ist $\mathbb{B}^h$ äquivalent zu $\mathbb{F}_2^h$.

Infolgedessen wird ein binäres Muster mit $\boldsymbol{\alpha} = \alpha_1 \cdots \alpha_i \cdots \alpha_h \in \mathbb{B}^h$ bezeichnet, wobei $\alpha_i \in \mathbb{F}_2$, $1 \leq i \leq h$ das Bit an der i-ten Stelle von $\boldsymbol{\alpha}$ bezeichnet.

Eine Indexmenge $\mathcal{D}_{\boldsymbol{\alpha}}$ wird definiert als:

$$\mathcal{D}_{\boldsymbol{\alpha}} = \bigcap_{i=1}^{h} \mathcal{D}_{\alpha_i}(\hat{\boldsymbol{x}}_i, \boldsymbol{y}), \quad \mathcal{D}_{\boldsymbol{\alpha}} \subseteq \{0, 1, \ldots, n-1\}. \tag{3.28}$$

Für zwei verschiedene binäre Muster $\boldsymbol{\alpha}$ und $\boldsymbol{\alpha}'$ sind die entsprechenden Mengen $\mathcal{D}_{\boldsymbol{\alpha}}$ und $\mathcal{D}_{\boldsymbol{\alpha}'}$ disjunkt [YKF98]:

$$\boldsymbol{\alpha}, \boldsymbol{\alpha}' \in \mathbb{B}^h, \boldsymbol{\alpha} \neq \boldsymbol{\alpha}' \quad \Longrightarrow \quad \mathcal{D}_{\boldsymbol{\alpha}} \cap \mathcal{D}_{\boldsymbol{\alpha}'} = \emptyset. \tag{3.29}$$

Wie in Anhang A.9 gezeigt wird, ergibt sich folgende Eigenschaft aus Gleichung (3.28):

$$\sum_{\boldsymbol{\alpha} \in \mathbb{B}^h \,|\, \alpha_i = 0} |\mathcal{D}_{\boldsymbol{\alpha}}| = |\mathcal{D}_0(\hat{\boldsymbol{x}}_i, \boldsymbol{y})|, \quad 1 \leq i \leq h. \tag{3.30a}$$

Entsprechend gilt:

$$\sum_{\boldsymbol{\alpha} \in \mathbb{B}^h \,|\, \alpha_i = 1} |\mathcal{D}_{\boldsymbol{\alpha}}| = |\mathcal{D}_1(\hat{\boldsymbol{x}}_i, \boldsymbol{y})|, \quad 1 \leq i \leq h. \tag{3.30b}$$

Eine Kardinalzahl $m_{\boldsymbol{\alpha}}(\hat{\boldsymbol{x}}, \boldsymbol{y}) \in \mathbb{N}$ wird definiert als:

$$m_{\boldsymbol{\alpha}}(\hat{\boldsymbol{x}}, \boldsymbol{y}) = |\mathcal{D}_{\boldsymbol{\alpha}} \cap \mathcal{D}_1(\hat{\boldsymbol{x}}, \boldsymbol{y})|. \tag{3.31}$$

Aus dieser Definition folgt:

$$\emptyset \subseteq \mathcal{D}_{\boldsymbol{\alpha}} \cap \mathcal{D}_1(\hat{\boldsymbol{x}}, \boldsymbol{y}) \subseteq \mathcal{D}_{\boldsymbol{\alpha}} \quad \Longrightarrow \quad 0 \leq m_{\boldsymbol{\alpha}}(\hat{\boldsymbol{x}}, \boldsymbol{y}) \leq |\mathcal{D}_{\boldsymbol{\alpha}}|. \tag{3.32}$$

In Anhang A.10 wird bewiesen, daß

$$|\mathcal{D}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}_i)| = \sum_{\boldsymbol{\alpha} \in \mathbb{B}^h} (-1)^{\alpha_i} \, m_{\boldsymbol{\alpha}}(\hat{\boldsymbol{x}}, \boldsymbol{y}) + |\mathcal{D}_1(\hat{\boldsymbol{x}}_i, \boldsymbol{y})|, \quad 1 \leq i \leq h. \tag{3.33}$$

3.2.2 Allgemeines Akzeptanzkriterium nach Kasami

Wie in Abschnitt 2.5 gezeigt wurde, wird als wahrscheinlichste Sendefolge $\boldsymbol{x}_{ML}$ diejenige geschätzte Sendefolge $\hat{\boldsymbol{x}}$ gewählt, für die das Skalarprodukt $\langle \hat{\boldsymbol{x}}, \boldsymbol{y} \rangle$ mit dem Vektor $\boldsymbol{y}$ maximal wird. Unter Ausnutzung von Gleichung (2.55) kann das Soft–Decision–Maximum–Likelihood–Akzeptanzkriterium wie folgt umformuliert werden: Wenn die Bedingung

$$\Xi_{\text{SDMLD}}: \quad d_\beta(\hat{\boldsymbol{x}}_h, \boldsymbol{y}) \leq \min_{\hat{\boldsymbol{x}}' \in \mathcal{X} \setminus \{\hat{\boldsymbol{x}}_h\}} d_\beta(\hat{\boldsymbol{x}}', \boldsymbol{y}) \tag{3.34}$$

erfüllt wird, dann ist $\hat{\boldsymbol{x}}_h$ die wahrscheinlichste Sendefolge $\boldsymbol{x}_{ML}$.

Dabei kann die Menge $\mathcal{X} \setminus \{\hat{\boldsymbol{x}}_h\}$ verstanden werden als die Menge derjenigen geschätzten Sendefolgen $\hat{\boldsymbol{x}}'$, die weit entfernt von $\hat{\boldsymbol{x}}_h$ liegen und deswegen mindestens einen euklidischen Abstand $d_{E\min}$ zu $\hat{\boldsymbol{x}}_h$ aufweisen. Gemäß [KTK+95, MLK95a, MLK95b, MLK97] kann die rechte Seite der obigen Ungleichung nicht exakt bestimmt werden, ohne den gewichteten Hammingabstand $d_\beta(\hat{\boldsymbol{x}}', \boldsymbol{y})$ aller geschätzten Sendefolgen $\hat{\boldsymbol{x}}'$ in $\mathcal{X} \setminus \{\hat{\boldsymbol{x}}_h\}$ zu berechnen. Hingegen kann eine dichte untere Schranke für die rechte Seite von Ungleichung (3.34) berechnet werden, nämlich

$$\min_{\hat{\boldsymbol{x}}_h \in \mathcal{T}} d_\beta(\hat{\boldsymbol{x}}_h, \boldsymbol{y}) \leq \min_{\hat{\boldsymbol{x}}' \in \mathcal{X} \setminus \{\hat{\boldsymbol{x}}_h\}} d_\beta(\hat{\boldsymbol{x}}', \boldsymbol{y}), \tag{3.35}$$

wenn

- zunächst eine Menge $\mathcal{T}$ von geschätzten Sendefolgen definiert wird, die weit entfernt von den besten, bereits in den vorherigen Iterationen von Ψ getesteten geschätzten Sendefolgen $\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \ldots, \hat{\boldsymbol{x}}_h \in \mathcal{X}^h$ liegen und
- danach der gewichtete Hammingabstand d_β zu allen geschätzten Sendefolgen in $\mathcal{T}$ minimiert wird.

Diese Menge $\mathcal{T}$ von geschätzten Sendefolgen im euklidischen Vektorraum $\mathbb{R}^n$ kann mittels folgender Gleichung definiert werden [YKF98]:

$$\mathcal{T}(h, \mathcal{W}, \mathcal{X}^h) = \bigcap_{i=1}^{h} \{\hat{\boldsymbol{x}} \in \mathcal{X} \,|\, |\mathcal{D}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}_i)| > w_i\}, \tag{3.36a}$$

$$\mathcal{T}(h, \mathcal{W}, \mathcal{X}^h) \subset \mathcal{X}, \quad h < |\mathcal{W}|, \tag{3.36b}$$

wobei $\mathcal{W}$ das Gewichtsprofil des $(n, k, d_{H\min})_2$-Codes $\mathcal{C}$ nach Gleichung (2.14) bezeichnet.

Da $d_H(\hat{\boldsymbol{c}}, \hat{\boldsymbol{c}}_i) = |\mathcal{D}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}_i)|$ gemäß (2.44) gilt, entspricht $\mathcal{T}(h, \mathcal{W}, \mathcal{X}^h)$ einer Schnittmenge $\mathcal{T}'(h, \mathcal{W}, \mathcal{C}^h)$ aller h Mengen derjenigen geschätzten Codewörter um $\hat{\boldsymbol{c}}_i$ in $\mathbb{F}_2^n$, die einen Hammingabstand d_H zu $\hat{\boldsymbol{c}}_i$ größer als w_i aufweisen:

$$\mathcal{T}'(h, \mathcal{W}, \mathcal{C}^h) = \bigcap_{i=1}^{h} \{\hat{\boldsymbol{c}} \in \mathcal{C} \,|\, d_H(\hat{\boldsymbol{c}}, \hat{\boldsymbol{c}}_i) > w_i\}, \tag{3.37a}$$

$$\mathcal{T}'(h, \mathcal{W}, \mathcal{C}^h) \subset \mathcal{C}, \quad h < |\mathcal{W}|. \tag{3.37b}$$

Nach Definition (3.36) gehört eine geschätzte Sendefolge $\hat{\boldsymbol{x}}$ zur Menge $\mathcal{T}(h, \mathcal{W}, \mathcal{X}^h)$, wenn

$$|\mathcal{D}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}_{\boldsymbol{i}})| > w_i, \quad 1 \leq i \leq h < |\mathcal{W}|. \tag{3.38}$$

Unter Verwendung von Gleichung (3.33) kann die obigen Ungleichung folgendermaßen umgeformt werden [KTK+95, KKTL95a, KKTL95b]:

$$\sum_{\boldsymbol{\alpha} \in \mathbb{B}^h} (-1)^{\alpha_i} \, m_{\boldsymbol{\alpha}}(\hat{\boldsymbol{x}}, \boldsymbol{y}) > \delta_i, \quad 1 \leq i \leq h < |\mathcal{W}|, \tag{3.39}$$

wobei

$$\delta_i = w_i - |\mathcal{D}_1(\hat{\boldsymbol{x}}_{\boldsymbol{i}}, \boldsymbol{y})| \tag{3.40}$$

ist.

Zur Vereinfachung der Notation werden 2^h beliebige Kardinalzahlen $m_{\boldsymbol{\alpha}}$ in einer Menge $\boldsymbol{m}$ zusammengefaßt:

$$\boldsymbol{m} = \{\overbrace{\underbrace{m_{0\cdots00}}_{h \text{ Stellen}}, m_{0\cdots01}, \ldots, m_{\boldsymbol{\alpha}}, \ldots, m_{1\cdots11}}^{2^h \text{ beliebige Kardinalzahlen}}\}, \quad m_{\boldsymbol{\alpha}} \in \mathbb{N}, \quad \boldsymbol{\alpha} \in \mathbb{B}^h. \tag{3.41}$$

Somit wird die Menge derjenigen Mengen $\boldsymbol{m}$ von 2^h Kardinalzahlen $m_{\boldsymbol{\alpha}}(\hat{\boldsymbol{x}}, \boldsymbol{y}) \in \mathbb{N}$, die Gleichung (3.39) erfüllen, mit $\mathbb{M}$ bezeichnet [KTK+95, KKTL95a, KKTL95b, YK97a, YK97b, YKF98]:

$$\mathbb{M} = \left\{ \boldsymbol{m} \;\middle|\; \boldsymbol{m} = \{m_{0\cdots00}(\hat{\boldsymbol{x}}, \boldsymbol{y}), m_{0\cdots01}(\hat{\boldsymbol{x}}, \boldsymbol{y}), \ldots, m_{\boldsymbol{\alpha}}(\hat{\boldsymbol{x}}, \boldsymbol{y}), \ldots, m_{1\cdots11}(\hat{\boldsymbol{x}}, \boldsymbol{y})\}, \sum_{\boldsymbol{\alpha} \in \mathbb{B}^h} (-1)^{\alpha_i} \, m_{\boldsymbol{\alpha}}(\hat{\boldsymbol{x}}, \boldsymbol{y}) > \delta_i, \; 1 \leq i \leq h < |\mathcal{W}| \right\}. \tag{3.42}$$

Basierend auf Gleichungen (3.41) und (3.42) wird eine Menge $\mathbb{M}_{\min}$ minimaler Mengen $\boldsymbol{m} \in \mathbb{M}$ von 2^h Kardinalzahlen $m_{\boldsymbol{\alpha}}(\hat{\boldsymbol{x}}, \boldsymbol{y})$ wie folgt definiert [KTK+95, KKTL95a, KKTL95b, YK97a, YK97b]:

$$\mathbb{M}_{\min} = \{\boldsymbol{m} \,|\, m_{\boldsymbol{\alpha}}(\hat{\boldsymbol{x}},\boldsymbol{y}) < m'_{\boldsymbol{\alpha}}(\hat{\boldsymbol{x}},\boldsymbol{y}), \; \boldsymbol{m}, \boldsymbol{m'} \in \mathbb{M}, \; \boldsymbol{m} \neq \boldsymbol{m'}, \; \boldsymbol{\alpha} \in \mathbb{B}^h\}, \tag{3.43a}$$

$$\emptyset \subseteq \mathbb{M}_{\min} \subset \mathbb{M}. \tag{3.43b}$$

Schließlich kann die untere Schranke für die rechte Seite von Ungleichung (3.34) unter Berücksichtigung der mittels Gleichungen (3.41), (3.43), (3.27), (3.28) und (3.31) eingeführten Notation folgendermaßen berechnet werden [KTK+95, KKTL95a, KKTL95b]:

$$\min_{\hat{\boldsymbol{x}}_h \in \mathcal{T}(h,\mathcal{W},\mathcal{X}^h)} d_\beta(\hat{\boldsymbol{x}}_h, \boldsymbol{y}) = \min_{\boldsymbol{m} \in \mathbb{M}_{\min}} \sum_{i \in \bigcup\limits_{\alpha \in \mathbb{F}^h} \mathcal{D}_\alpha^{(m_\alpha(\hat{\boldsymbol{x}}_h, y))}} \beta_i. \tag{3.44}$$

Infolgedessen kann das allgemeine Akzeptanzkriterium nach Kasami basierend auf mehreren Codewörtern wie folgt formuliert werden: Wird die Bedingung

$$\Xi_{\text{Kasami}}: \quad d_\beta(\hat{\boldsymbol{x}}_h, \boldsymbol{y}) \leq \min_{\boldsymbol{m} \in \mathbb{M}_{\min}} \sum_{i \in \bigcup\limits_{\alpha \in \mathbb{F}^h} \mathcal{D}_\alpha^{(m_\alpha(\hat{\boldsymbol{x}}_h, y))}} \beta_i \tag{3.45}$$

erfüllt, dann ist $\hat{\boldsymbol{x}}$ die wahrscheinlichste Sendefolge $\boldsymbol{x}_{ML}$ [KTK+95, KKTL95a, KKTL95b, YK97a, YK97b, YKF98].

Wie diese Bedingung in der Praxis benutzt werden kann, wird in den nächsten Unterabschnitten diskutiert.

3.2.3 Akzeptanzkriterium basierend auf zwei Codewörtern

Eine ausführliche Herleitung des Akzeptanzkriteriums nach Kasami basierend auf zwei Codewörtern wurde bereits in [KTK+95] durchgeführt und wird hier nicht nochmals wiederholt. Stattdessen werden Implementierungsaspekte durch eine algorithmische Darstellung als Beitrag dieser Arbeit berücksichtigt.

Vorausgesetzt werden ein modulierter Code $\mathcal{X}$ mit Gewichtsprofil $\mathcal{W}$, ein vom Empfangsvektor $\boldsymbol{r}$ abhängiger Vektor $\boldsymbol{y}$ und ein iterativer, beliebiger Decodierer Ψ. Ferner gilt $h = 2$ in diesem Unterabschnitt. Das bedeutet, daß zwei geschätzte Sendefolgen $\hat{\boldsymbol{x}}_{\boldsymbol{1}}$ und $\hat{\boldsymbol{x}}_{\boldsymbol{2}}$ verwendet werden, damit eine untere Schranke für die rechte Seite von Ungleichung (3.34) berechnet wird. Dabei wurde $\hat{\boldsymbol{x}}_{\boldsymbol{2}}$ vom Decodierer Ψ in der letzten Iteration erzeugt und wird im Moment getestet, während $\hat{\boldsymbol{x}}_{\boldsymbol{1}}$ die beste geschätzte Sendefolge innerhalb aller Iterationen außer der letzten bezeichnet.

Dieses Akzeptanzkriterium kann durch folgenden Algorithmus implementiert werden:

- **Schritt 1:** Bilde die Indexmengen $\mathcal{D}_0(\hat{\boldsymbol{x}}_\mathbf{1}, \boldsymbol{y})$, $\mathcal{D}_0(\hat{\boldsymbol{x}}_\mathbf{2}, \boldsymbol{y})$, $\mathcal{D}_1(\hat{\boldsymbol{x}}_\mathbf{1}, \boldsymbol{y})$ und $\mathcal{D}_1(\hat{\boldsymbol{x}}_\mathbf{2}, \boldsymbol{y})$ gemäß Gleichungen (2.41) und (2.42).

- **Schritt 2:** Bestimme $|\mathcal{D}_1(\hat{\boldsymbol{x}}_\mathbf{1}, \boldsymbol{y})|$ und $|\mathcal{D}_1(\hat{\boldsymbol{x}}_\mathbf{2}, \boldsymbol{y})|$.

- **Schritt 3:** Berechne δ_1 und δ_2 mittels Gleichung (3.40):

$$\delta_1 = w_1 - |\mathcal{D}_1(\hat{\boldsymbol{x}}_\mathbf{1}, \boldsymbol{y})|, \tag{3.46a}$$
$$\delta_2 = w_2 - |\mathcal{D}_1(\hat{\boldsymbol{x}}_\mathbf{2}, \boldsymbol{y})|. \tag{3.46b}$$

- **Schritt 4:** Falls $\delta_1 < \delta_2$, vertausche δ_1 mit δ_2 und $\hat{\boldsymbol{x}}_\mathbf{1}$ mit $\hat{\boldsymbol{x}}_\mathbf{2}$.

- **Schritt 5:** Falls $\delta_1 \leq 0$ ist, dann gehe zum Schritt 11.

- **Schritt 6:** Berechne die Indexmengen $\mathcal{D}_{00}$ und $\mathcal{D}_{01}$ mittels Gleichung (3.28):

$$\mathcal{D}_{00} = \mathcal{D}_0(\hat{\boldsymbol{x}}_\mathbf{1}, \boldsymbol{y}) \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_\mathbf{2}, \boldsymbol{y}), \tag{3.47a}$$
$$\mathcal{D}_{01} = \mathcal{D}_0(\hat{\boldsymbol{x}}_\mathbf{1}, \boldsymbol{y}) \cap \mathcal{D}_1(\hat{\boldsymbol{x}}_\mathbf{2}, \boldsymbol{y}). \tag{3.47b}$$

- **Schritt 7:** Aus Gleichungen (3.46) und (3.47) berechne folgende Indexmenge $\mathcal{Q}$ [KTK+95, KKTL95a, KKTL95b, KTKL99]:

$$\mathcal{Q} = \left(\mathcal{D}_{00} \cup \mathcal{D}_{01}^{\left(\left\lfloor \frac{\delta_1 - \delta_2}{2} \right\rfloor\right)} \right)^{(\delta_1)}. \tag{3.48}$$

- **Schritt 8:** Berechne den gewichteten Hammingabstand $d_\beta(\hat{\boldsymbol{x}}_\mathbf{2}, \boldsymbol{y})$ nach Gleichung (2.49):

$$d_\beta(\hat{\boldsymbol{x}}_\mathbf{2}, \boldsymbol{y}) = \sum_{i \in \mathcal{D}_1(\hat{\boldsymbol{x}}_\mathbf{2}, \boldsymbol{y})} \beta_i. \tag{3.49}$$

- **Schritt 9:** Berechne die untere Schranke (entspricht Gleichung (3.44)) für den Fall $h = 2$:

$$\min_{\hat{\boldsymbol{x}}_\mathbf{2} \in \mathcal{T}(2, \mathcal{W}, \mathcal{X}^2)} d_\beta(\hat{\boldsymbol{x}}_\mathbf{2}, \boldsymbol{y}) = \sum_{i \in \mathcal{Q}} \beta_i. \tag{3.50}$$

- **Schritt 10:** Wird die Bedingung

$$\Xi_{\text{Kasami}(\hat{\boldsymbol{x}}_1,\hat{\boldsymbol{x}}_2)}: \quad d_\beta(\hat{\boldsymbol{x}}_2, \boldsymbol{y}) \leq \sum_{i \in \mathcal{Q}} \beta_i \tag{3.51}$$

  erfüllt, dann ist $\hat{\boldsymbol{x}}_2$ die wahrscheinlichste Sendefolge $\boldsymbol{x}_{ML}$. Beende den Algorithmus.

- **Schritt 11:** Es konnte nicht festgestellt werden, ob $\hat{\boldsymbol{x}}_2$ tatsächlich die wahrscheinlichste Sendefolge $\boldsymbol{x}_{ML}$ ist. Beende den Algorithmus.

Speziell für den Fall $\hat{\boldsymbol{x}}_1 = \hat{\boldsymbol{x}}_2 = \hat{\boldsymbol{x}}$ (d. h. $h = 1$) gelten:

$$\delta_1 = \delta = d_{H\min} - |\mathcal{D}_1(\hat{\boldsymbol{x}}, \boldsymbol{y})|, \tag{3.52a}$$

$$\mathcal{D}_{00} = \mathcal{D}_0(\hat{\boldsymbol{x}}, \boldsymbol{y}), \tag{3.52b}$$

$$\mathcal{D}_{01} = \emptyset. \tag{3.52c}$$

$$\mathcal{Q} = \mathcal{D}_0(\hat{\boldsymbol{x}}, \boldsymbol{y})^{(\delta)}. \tag{3.52d}$$

Somit vereinfacht sich die Bedingung $\Xi_{\text{Kasami}(\hat{\boldsymbol{x}}_1,\hat{\boldsymbol{x}}_2)}$ des Akzeptanzkriteriums nach Kasami basierend auf zwei Codewörtern in die Bedingung Ξ_{TP} des Akzeptanzkriteriums nach Taipale und Pursley:

$$\Xi_{\text{TP}}: \quad d_\beta(\hat{\boldsymbol{x}}, \boldsymbol{y}) \leq \sum_{i \in \mathcal{D}_0(\hat{\boldsymbol{x}}, \boldsymbol{y})^{(\delta)}} \beta_i. \tag{3.24}$$

In [KTK+95] wird ebenfalls bewiesen, daß die Entscheidungsräume des hier vorgestellten Akzeptanzkriteriums größer sind als beim Akzeptanzkriterium nach Kaneko [KNIH94].

Als Beispiele für die Verwendung des Akzeptanzkriteriums nach Kasami basierend auf zwei Codewörtern wird auf [KNT+95, KTKL96, KKL98, KTKL99] verwiesen.

3.2.4 Akzeptanzkriterium basierend auf mehr als zwei Codewörtern

In [KTK+95] wird auch ein Akzeptanzkriterium basierend auf drei Codewörtern hergeleitet. Jedoch zeigen die Simulationsergebnisse mit der zweiten Variante des Algorithmus von Chase [Cha72, Bos98], daß die Entscheidungsräume um die Codewörter geringfügig größer werden als beim Akzeptanzkriterium basierend auf zwei Codewörtern, obwohl die Komplexität wesentlich größer wird [KTK+95, KKTL95a, KKTL95b]. Der Begriff Komplexität kann in diesem Fall hauptsächlich als die Anzahl der Operationen mit

Skalaren verstanden werden [YKF98]. Aus diesem Grund wurden Akzeptanzkriterien nach Kasami basierend auf mehr als zwei Codewörtern in den Simulationen von Kapitel 5 nicht implementiert. Für Beispiele von Decodieralgorithmen, die das Akzeptanzkriterium mit $h = 3$ verwenden, wird auf [KNT+95, KTKL96, KKL98, KTKL99] verwiesen.

Während die Komplexität der Akzeptanzkriterien für $1 \leq h \leq 3$ linear mit der Länge n des Codes $\mathcal{C}$ steigt, wächst sie für $h = 4$ quadratisch mit n [YKF98]. Eine ausführliche Komplexitätsanalyse für diesem Fall wird in [KK98a, YKF98] angegeben und eine Methode zur Implementierung dieses Akzeptanzkriteriums wird in [YK97b, YKF98] vorgeschlagen. Sie wird in [KK98b] für einige Decodieralgorithmen erörtert.

Sogar für den Fall $h = 5$ werden in [YK97a, YK97b, YKF98] einige theoretische Überlegungen diskutiert.

3.3 Abschließende Bemerkungen

In diesem Kapitel wurde zunächst das Konzept eines Akzeptanzkriteriums definiert. Danach wurden die fünf wesentlichsten Akzeptanzkriterien basierend auf einem Codewort vorgestellt: Der Syndrom–Test, der Hyperkugel–Test, der Hyperkreiskegel–Test, der GMD-Test und das Akzeptanzkriterium nach Taipale und Pursley.

Das Akzeptanzkriterium nach Dumer [Dum81] wurde nicht erläutert, denn die Größe seiner Entscheidungsräume um die Codewörter liegen zwischen der eines Hyperkreiskegel–Tests und der eines Akzeptanzkriteriums nach Taipale und Pursley. Das Akzeptanzkriterium nach Enns [Enn87] wurde ebenfalls nicht dargestellt, weil es äquivalent zum Akzeptanzkriterium nach Taipale und Pursley ist.

In einem Beitrag von Kabatyanskii wurden der GMD-Test, das Akzeptanzkriterium nach Dumer [Dum81] und das Akzeptanzkriterium nach Enns [Enn87] (bzw. nach Taipale und Pursley) durch Metriken miteinander verglichen, wobei das letzte die größten Entscheidungsräumen besitzt [Kab91].

Schranken für die Wahrscheinlichkeit, daß ein Akzeptanzkriterium eine Decodierung eines vorgegebenen Codes $\mathcal{C}$ nicht abbricht, wurden in [SJ98] für den Syndrom–Test, den Hyperkugel–Test, den Hyperkreiskegel–Test und für das Akzeptanzkriterium nach Taipale und Pursley hergeleitet.

Des weiteren wurde in Unterabschnitt 3.1.5 erklärt, daß das in [GA88, GA89, God91, Bar93, BGW94, BGW97, BD97] verwendete Akzeptanzkri-

terium, das testet, ob ein bestimmter Vektor innerhalb der Voronoi–Region $\mathcal{V}(\mathbf{0})$ des Nullwortes $\mathbf{0}$ liegt, als ein spezieller Fall des Akzeptanzkriterium nach Taipale und Pursley mit geringerer Komplexität verstanden werden kann.

In Abschnitt 3.2 wurden die Akzeptanzkriterien auf Basis von mehreren Codewörtern berücksichtigt. Das Akzeptanzkriterium nach Kaneko [KNIH94] wurde dabei nicht behandelt, weil seine Entscheidungsräume um die Codewörter kleiner sind als die des Akzeptanzkriteriums nach Kasami [KTK+95].

Eine einheitliche Notation wurde in Unterabschnitt 3.2.1 eingeführt, die die Herleitung des Akzeptanzkriteriums nach Kasami im allgemeinen Fall ermöglichte. Danach wurde als Beitrag der vorliegenden Arbeit ein Algorithmus zur Berechnung des Akzeptanzkriteriums nach Kasami basierend auf zwei Codewörtern vorgeschlagen, der gemeinsam mit dem Decodieralgorithmus von Kapitel 5 simuliert wird. Schließlich wurden im letzten Unterabschnitt viele Verweise auf Akzeptanzkriterien basierend auf mehr als zwei Codewörten angegeben.

Überdies existieren auch einige Akzeptanzkriterien, die mit einem spezifischen Decodieralgorithmus arbeiten. Beispielsweise liefert das Akzeptanzkriterium nach Moorthy und Lin zusätzlich eine Region des euklidischen Vektorraumes, in der die wahrscheinlichste Sendefolge $\boldsymbol{x}_{ML}$ gefunden werden kann, falls das Kriterium nicht erfüllt wird [MLK95a, MLK95b, MLK97].

Wie die ursprüngliche Idee von Dorsch [Dor74] arbeiten sowohl der Decodieralgorithmus von Fossorier und Lin [Fos94, FL95, FL96a, FL96b] als auch der von Gazelle und Snyders [GS97] mit einer zuverlässigsten Basis (MRB, *Most Reliable Basis*). Beide Decodieralgorithmen benutzen dasselbe Akzeptanzkriterium, das auf einem einzigen Codewort beruht. Wegen der Anwendung der MRB werden die Entscheidungsräume dagegen größer als beim Akzeptanzkriterium nach Taipale und Pursley.

In [FL97] haben Fossorier und Lin das Akzeptanzkriterium nach Kaneko [KNIH94] mittels der MRB verbessert. Sogar das im letzten Unterabschnitt zitierte Akzeptanzkriterium nach Kasami basierend auf drei Codewörtern [KTK+95, KKTL95a, KKTL95b] kann unter Anwendung der MRB weiter verbessert werden [FKT+97].

Hingegen sind alle Akzeptanzkriterien, die eine MRB verwenden, inkompatibel mit dem in nächsten Kapitel vorgestellten Decodieralgorithmus mittels Informationsmengen, weil sich die MRB in diesem Fall ständig verändert. Deswegen wurden diese Klasse von Akzeptanzkriterien in diesem Kapitel lediglich zitiert und nicht näher untersucht.

Im nächsten Kapitel werden die hier betrachteten Akzeptanzkriterien in einem Decodieralgorithmus verwendet, der geeignet ist, lange Codes mithilfe von verschiedenen Informationsmengen zu decodieren.

Kapitel 4

Decodieralgorithmen mit Informationsmengen

Wie bereits am Anfang dieser Arbeit erwähnt wurde, kann eine SDMLD von langen und komplexen Blockcodes nicht aufwandsgünstig durchgeführt werden, sondern es ist nur eine HDBMD möglich. Damit müßte üblicherweise auf den in Kapitel 1 genannten zusätzlichen Gewinn von 2 bis zu asymptotisch 3 dB durch Anwendung von Soft–Decision verzichtet werden.

In diesem Kapitel wird jedoch eine allgemeine kombinatorische Methode namens Informationsmengen–Decodierung erörtert, die eine geringere Komplexität als die einer üblichen SDMLD hat und asymptotisch dieselbe WER wie die einer üblichen SDMLD erreichen kann. Ähnliche Methoden wurden erstmals für Hard–Decision Kanäle verwendet. Die theoretische Begründung der Gültigkeit solcher Methoden wird im ersten Abschnitt beschrieben. Basierend darauf befaßt sich der zweite Abschnitt mit den Decodierverfahren mit Hilfe von Informationsmengen in Hard–Decision Kanälen. Die Theorie wird dann in Abschnitt 4.3 für Soft-Decision erweitert. Der nächste Abschnitt beschäftigt sich mit der Decodierung mit Hilfe von Informationsmengen bei Soft–Decision. Schließlich wird das Kapitel zusammengefaßt.

4.1 Blockfehlerwahrscheinlichkeit bei Hard–Decision

Im Jahre 1983 hat Evseev bewiesen [Evs83], daß sich die WER P_Ψ eines Decodierers Ψ, der mit einem linearen $(n,k)_2$-Code $\mathcal{C}$ bei Hard–Decision ar-

beitet, höchstens verdoppelt, wenn anstelle der Durchführung einer vollständigen Hard–Decision–Minimalabstand–Decodierung (HDMDD, *Minimum–Distance Hard–Decision Decoding*) lediglich die 2^{n-k} wahrscheinlichsten Fehlervektoren berücksichtigt werden:

$$P_{\Psi} \leq 2\, P_{MDD}, \tag{4.1}$$

wobei P_{MDD} die WER einer HDMDD bezeichnet. In Anhang A.11 wird diese Ungleichung mit der in Kapitel 2 verwendeten Notation hergeleitet.

Gemäß Gleichungen (2.16) und (2.21) liegen die 2^{n-k} wahrscheinlichsten Fehlervektoren innerhalb der Kugel $\mathcal{S}(d_G, \mathbf{0})$ mit Hammingradius d_G, die um das Nullwort $\mathbf{0}$ zentriert ist. Wenn die Länge n des Codes $\mathcal{C}$ gegen Unendlich geht, überdeckt $\mathcal{S}(d_G, \mathbf{0})$ exakt nur die 2^{n-k} wahrscheinlichsten Fehlervektoren. In diesem Fall befaßt sich der Decodierer Ψ zunächst mit keinem Fehler, dann mit allen n Fehlermuster vom Hamminggewicht 1, dann mit allen $\binom{n}{2}$ Fehlermuster vom Hamminggewicht 2, usw. bis schließlich mit allen $\binom{n}{d_G}$ Fehlermuster vom Hamminggewicht d_G.

Wichtig in dieser Arbeit [Evs83] war, daß erheblicher Decodieraufwand bei näherungsweiser HDMDD eingespart wird. Der Beitrag von Evseev war der erste auf einem ganz neuen Forschungsgebiet, nämlich der Suche nach Prozeduren zur näherungsweisen HDMDD mit geringer Komplexität. Einige Varianten dieser Algorithmen sind beispielweise in [Dum89, Kro89, Dum96b, BKv97] dargestellt. Für eine Analyse und einen theoretischen Vergleich dieser Decodierverfahren wird auf [Bar98] verwiesen.

Wenige Jahre später hat Blinovsky gezeigt [Bli87], daß nahezu alle linearen $(n, k)_2$-Codes $\mathcal{C}$ einen Überdeckungsradius $\rho = (1 + o(1))\, d_G$ haben, wenn n gegen Unendlich geht. Dabei steht $o(\cdot)$ für den Landau–Operator „klein o“ [Bar97, BSMM99]. Somit bezeichnet $o(1)$ eine Zahl, die gegen Null geht, wenn die Länge n des Codes $\mathcal{C}$ gegen Unendlich geht. Das bedeutet, daß die Betrachtung von Fehlerereignissen bis zum Hamminggewicht w_H wenig mehr als d_G bei fast allen langen linearen $(n, k)_2$-Codes $\mathcal{C}$ äquivalent zu einer vollständigen HDMDD ist.

Folglich muß ein Decodierer eines langen linearen $(n, k)_2$-Codes $\mathcal{C}$ lediglich etwa die wahrscheinlichsten 2^{n-k} Fehlervektoren berücksichtigen, um nahezu die WER P_{MDD} einer HDMDD zu erreichen. Sogar bei konkreten $(n, k)_2$-Codes $\mathcal{C}$ mit endlicher Länge n liegt P_{Ψ} bei der Decodierung nahe an P_{MDD}, wenn Fehlermuster bis zum Hamminggewicht d_G in Betracht gezogen werden [Bar98].

Erst mit der Arbeit von Dumer im Jahre 1996 wurde das Ergebnis von Evseev sorgfältig interpretiert und durch folgende Ungleichung verallgemeinert [Dum96b]:

$$P_{\Psi} \leq P_{MDD}\left(1 + \frac{2^{n-k}}{I}\right), \quad I \geq 2^{n-k}, \tag{4.2}$$

wobei I die Anzahl von wahrscheinlichsten Fehlervektoren bezeichnet, die betrachtet werden sollen. Selbstverständlich ist Gleichung (4.1) ein Spezialfall der obigen Gleichung für $I = 2^{n-k}$.

4.2 Decodierverfahren mit Informationsmengen bei Hard–Decision

Damit eine Decodierung mit Hilfe von Informationsmengen nach Unterabschnitt 2.2.6 durchgeführt werden kann, soll eine allgemeine Methode zur Auswahl von Informationsmengen $\mathcal{I}$ definiert werden. Dagegen ist es normalerweise zu kompliziert, eine Menge von verschiedenen Informationsmengen $\mathcal{I}$ zu finden, die für alle $(n,k)_2$-Codes $\mathcal{C}$ eine bestimmte Anzahl von Fehlern überdecken. Einige wenige Beispiele dieser Mengen für konkrete Codes werden in [Gor82, Wol83, And92, Bar93, MF93, KF95] behandelt.

Offensichtlich ist ein möglicher Ansatz, immer eine gleichverteilte Auswahl von Stellen zu verwenden, damit die Fehlermuster gleichmäßig überdeckt werden. Um alle Fehlerereignisse bis zum Hamminggewicht d_G berücksichtigt zu können, suchen die meisten Algorithmen, die auf dieser Idee beruhen, ständig eine Menge von k fehlerfreien Stellen des Hard–Decision Vektors $\boldsymbol{v}$. Mit anderen Worten, sie suchen einen Block $\mathcal{B}$ von $n-k$ Stellen, der bis zu d_G Fehler enthält. Formal wird eine $(n, n-k, d_G)$-Überdeckungsmenge gemäß Unterabschnitt 2.2.7 benötigt, so daß alle möglichen Fehlermuster bis zum Hamminggewicht d_G durch mindestens einen Block $\mathcal{B}$ von $n-k$ Stellen überdeckt werden. Deshalb wird dieses Decodierverfahren Decodierung mit Hilfe von Überdeckungsmengen (*Covering–Set*–Decoding) genannt [Kro89, CG90, Bar98].

Nach Gleichungen (2.31) und (2.35) können die untere und die obere Schranke für $|\mathcal{U}(n, n-k, d_G)|$ folgendermaßen geschätzt werden [ES74,

CC81, ASE92, Dum98a, Dum98c]:

$$\frac{\binom{n}{d_G}}{\binom{n-k}{d_G}} \leq |\mathcal{U}(n, n-k, d_G)| \leq \left(\ln \binom{n-k}{d_G} + 1\right) \frac{\binom{n}{d_G}}{\binom{n-k}{d_G}}. \tag{4.3}$$

Gemäß der Definition in [Dum] haben zwei Funktionen $f(n)$ und $g(n)$ die gleiche exponentielle Ordnung

$$f(n) \stackrel{\text{exp}}{=} g(n), \tag{4.4a}$$

wenn

$$\lim_{n\to\infty} \frac{\ln f(n)}{\ln g(n)} = \kappa, \tag{4.4b}$$

wobei κ eine beliebige Konstante ist.

Daraus folgt, daß sowohl die linke als auch die rechte Seite von Ungleichung (4.3) die gleiche exponentielle Ordnung $\binom{n}{d_G} / \binom{n-k}{d_G}$ besitzen [Dum97b, Dum98a, Dum98b, Dum98c], wobei die obere Schranke nach Gleichung (2.35) bei der Annahme einer gleichverteilten Auswahl von Blöcken $\mathcal{B}$ hergeleitet wurde [ES74, ASE92].

Nachdem ν Blöcke $\mathcal{B}$ von $n-k$ Stellen ausgewählt wurden, mit

$$\nu \geq n \ln n \frac{\binom{n}{d_G}}{\binom{n-k}{d_G}}, \tag{4.5}$$

ist auch in [ES74, ASE92] bewiesen, daß alle Fehlervektoren bis zum Hamminggewicht d_G mit einer geringfügigen Wahrscheinlichkeit $\exp(-n \ln n)$ nicht überdeckt werden.

Jeder ausgewählte Block $\mathcal{B}$ von $n-k$ Stellen entspricht einer Menge von k vermutlich fehlerfreien Stellen des Hard–Decision Vektors $\boldsymbol{v}$, die möglicherweise eine Informationsmenge $\mathcal{I}$ bilden können. Dementsprechend wird ein Codewort $\boldsymbol{c}'(\mathcal{I})$ erzeugt. Somit wird dasjenige geschätzte Codewort $\hat{\boldsymbol{c}}$ am Ausgang des Decodierers Ψ geliefert, das den geringsten Hammingabstand d_H zum Hard–Decision–Vektor $\boldsymbol{v}$ aufweist, verglichen mit allen anderen erzeugten $\boldsymbol{c}'(\mathcal{I})$.

Im allgemeinen bilden nicht alle Mengen von k Stellen eine Informationsmenge $\mathcal{I}$. Ausnahmen sind die in Unterabschnitt 2.2.4 beschriebenen MDS-Codes. Es ist hingegen bewiesen, daß eine beliebige Menge von k Stellen asymptotisch für nahezu alle linearen $(n,k)_2$-Codes $\mathcal{C}$ in den meisten Fällen eine Informationsmenge $\mathcal{I}$ bildet [Dum97b, Dum98a]. Die Anzahl ξ von Codewörtern $\boldsymbol{c}$, die an k beliebig ausgewählten Stellen von $\boldsymbol{c}$ übereinstimmen, kann asymptotisch durch folgende Ungleichung bestimmt werden [BKv99]:

$$\operatorname{lb} \xi \leq \sqrt{\operatorname{lb}\binom{n}{k}} < \sqrt{n \, \mathrm{H}_2\left(\frac{k}{n}\right)}, \tag{4.6}$$

wobei

$$\mathrm{H}_2(\varsigma) = -\varsigma \operatorname{lb}\varsigma - (1-\varsigma) \operatorname{lb}(1-\varsigma) = -\operatorname{lb}\left(\varsigma^{\varsigma}(1-\varsigma)^{1-\varsigma}\right) \tag{4.7}$$

die binäre Entropiefunktion [Fri95] und $\operatorname{lb}\varsigma$ den binären (dyadischen) Logarithmus bezeichnet.

Vor kurzem wurde ein neues Decodierverfahren basierend auf dem in Abschnitt 4.1 erwähnten Prinzip vorgeschlagen [BKv99]. Dieses Decodierverfahren namens Supercode–Decodierung stellt die bis jetzt geringste bekannte asymptotische Komplexität einer HDMDD dar. Grundsätzlich ist es eine allgemeine Methode zur Decodierung von Blockcodes und beruht auf den Konzepten sowohl einer Decodierung mit Hilfe von Überdeckungsmengen [Kro89, CG90, Bar98] als auch einer *Split–Syndrome*–Decodierung (etwa Decodierung mit Hilfe von Teilen des Syndroms) [Dum89, Bar98]. Für eine ausführliche Beschreibung dieser Decodiermethoden wird auf [BKv99] verwiesen.

4.3 Blockfehlerwahrscheinlichkeit bei Soft–Decision

Dumer hat in seiner Arbeit [Dum96b] nicht nur den Hard-Decision Fall gemäß Gleichung (4.2) berücksichtigt, sondern ebenfalls die bereits in Abschnitt 4.1 genannten Ergebnisse von Evseev [Evs83] für ganz allgemeine symmetrische Kanäle erweitert, die kontinuierlich oder diskret, additiv oder multiplikativ und gedächtnislos oder gedächtnisbehaftet sein können.

In [Dum96b] wurde bewiesen, daß die WER P_Ψ eines Decodierers Ψ, der mit einem linearen $(n,k)_2$-Code $\mathcal{C}$ in Soft–Decision Kanälen arbeitet, wie folgt berechnet werden kann:

$$P_\Psi \leq P_{MLD}\left(1+\frac{2^{n-k}}{I-2^{n-k}}\right), \quad I > 2^{n-k}. \tag{4.8}$$

Dabei bezeichnet I wieder die Anzahl von wahrscheinlichsten Fehlermustern, die in Betracht gezogen werden sollen.

Somit hat Dumer allgemeine Soft–Decision Decodieralgorithmen mit geringer Komplexität vorgeschlagen [Dum96b], die ähnliche Ansätze verwenden wie die seit dem Beitrag von Evseev [Evs83] bekannten Hard–Decision Decodieralgorithmen [Dum89, Kro89, BKv97, BKv99].

Jedoch ist viel weniger über eine Soft–Decision Decodierung als über eine Hard–Decision Decodierung bekannt. In Hard–Decision Kanälen liegen die 2^{n-k} wahrscheinlichsten Fehlervektoren innerhalb der Kugel $\mathcal{S}(d_G, \mathbf{0})$ mit Hammingradius d_G um das Nullwort $\mathbf{0}$, die völlig unabhängig vom Vektor $\boldsymbol{y}$ ist. Andererseits liegt die wesentliche Schwierigkeit bei Soft–Decision an der Abhängigkeit der Menge der wahrscheinlichsten Fehlervektoren von der Zuverlässigkeitsinformation β_i, $0 \leq i < n$ jeder Komponente y_i des Vektors $\boldsymbol{y} = (y_0, \ldots, y_{n-1})$ [Dum96b]. Im allgemeinen bilden diese wahrscheinlichsten Fehlervektoren in Soft–Decision Kanälen keine Kugel mehr, und ihr Hamminggewicht w_H kann möglicherweise deutlich höher als d_G sein [Dum97b]. In besonderen Fällen sollen Fehlerereignisse bis zum Hamminggewicht $n-k$ in einer SDMLD betrachtet werden [Dum].

Außerdem kann eine gleichverteilte Auswahl von Stellen wie in Abschnitt 4.2 bei Soft–Decision nicht benutzt werden, denn jede Stelle i hat ihre eigene Zuverlässigkeit β_i. Damit besitzen verschiedene Mengen von k Stellen unterschiedliche Wahrscheinlichkeiten. Deswegen soll der Decodierer Ψ die geringste Anzahl von Blöcken $\mathcal{B}$ verwenden, die die Fehlermuster überdecken, unter Berücksichtigung der n Zuverlässigkeiten β_i.

Wie viele Fehlervektoren bei Soft–Decision berücksichtigt werden sollen, damit die WER P_Ψ eines beliebigen Decodierers Ψ nahe an der WER P_{MLD} einer SDMLD liegt, wurde von Dumer in einem anderen Beitrag [Dum96a] ausführlich untersucht. Ausgehend von der Definition eines diskreten Ellipsoids nach Gleichung (2.65) hat er gezeigt, daß die gleiche exponentielle Ordnung von 2^{n-k} wahrscheinlichsten Fehlervektoren gemäß Definition (4.4) genauso wie bei Hard–Decision in Betracht gezogen werden soll, damit der

Decodierer Ψ im asymptotischen Fall die gleiche WER wie die einer SDMLD erreicht [Dum96a].

Das wesentliche Problem ist, eine minimale ellipsoidische Überdeckungsmenge zu finden, die ein beliebiges diskretes Ellipsoid $\mathcal{E}(r, \boldsymbol{\beta}, \mathbf{0})$ von exponentieller Mächtigkeit 2^{n-k} mit der kleinsten Anzahl von Blöcken $\mathcal{B}$ der Länge $n-k$ völlig überdeckt [Dum97b].

Dumer hat sich viele Jahre mit diesem Problem beschäftigt. In einer Reihe von Artikeln [Dum96a, Dum97a, Dum97b, Dum98a, Dum98b, Dum98c, Dum] hat er die asymptotische Lösung aufgezeigt. Er hat bewiesen, daß nur die 2^{n-k} wahrscheinlichsten Fehlervektoren berücksichtigt werden sollen, damit eine Soft–Decision–Decodierung asymptotisch optimal wird. Darüber hinaus hat er vorgeschlagen, wie die dazugehörige Überdeckung erzeugt werden kann [Dum97b, Dum98a, Dum98c].

4.4 Decodierverfahren mit Informationsmengen bei Soft–Decision

In diesem Abschnitt wird zunächst eine allgemeine Methode zur Konstruktion einer minimalen ellipsoidischen Überdeckungsmenge dargestellt. Wie die bestmögliche Informationsmenge aus einer Menge von geordneten Stellen gebildet werden kann, wird im Unterabschnitt 4.4.2 diskutiert. Es folgt zuletzt die Beschreibung eines allgemeinen Decodierverfahrens mit Informationsmengen und Akzeptanzkriterium für Soft–Decision Kanäle.

4.4.1 Erzeugung ellipsoidischer Überdeckungsmengen

Die hier betrachtete Methode beruht hauptsächlich auf den Arbeiten von Dumer [Dum97b, Dum98a, Dum98c]. Der grundsätzliche Unterschied in den nächstfolgenden Gleichungen ist dagegen, daß die von Dumer beschriebene Methode ständig einen Block $\mathcal{B}$ von höchstens $n-k$ Stellen sucht, die möglicherweise eine Prüfmenge $\mathcal{P}$ bilden können, während die hier dargestellte Methode eine Menge von mindestens k Stellen auswählt, die möglicherweise eine Informationsmenge $\mathcal{I}$ bilden können. Wie im nächsten Unterabschnitt deutlich wird, ermöglicht diese Verbesserung einerseits eine einfachere Erzeugung von Informationsmengen $\mathcal{I}$ und andererseits eine effiziente Zuordnung der ausgewählten Stellen i bezüglich ihrer Zuverlässigkeiten β_i.

Vorausgesetzt werden ein $(n,k)_2$-Code $\mathcal{C}$ und ein Vektor $\boldsymbol{y} \in \mathbb{R}^n$, der decodiert werden soll. Im asymptotischen Fall können die 2^{n-k} wahrscheinlichsten Fehlervektoren des diskreten Ellipsoids $\mathcal{E}(r, \boldsymbol{\beta}, \mathbf{0})$ mit der geringsten Anzahl von Versuchen durch folgende Methode überdeckt werden:

- Jede Stelle i des Vektors $\boldsymbol{y}$ wird mit einer Wahrscheinlichkeit p_i ausgewählt, die unabhängig von den anderen Stellen ist.
- Damit durchschnittlich k Stellen ausgewählt werden können, sollen diese Wahrscheinlichkeiten p_i folgende Gleichung erfüllen:

$$\sum_{i=0}^{n-1} p_i = k. \tag{4.9}$$

- Falls eine bestimmte Auswahl weniger als k Stellen enthält, dann wird sie vernachlässigt und eine neue Auswahl wird durchgeführt.
- Damit die Anzahl von notwendigen Schritten zur Erzeugung der ellipsoidischen Überdeckungsmenge minimiert werden kann, soll folgende Gleichung nach dem Parameter $\lambda \in \mathbb{R}^+$ aufgelöst werden [Dum98a, Dum98c]:

$$\sum_{i=0}^{n-1} \mathrm{H}_2(\gamma_i) = k, \tag{4.10a}$$

$$\gamma_i = \frac{1}{1 + 2^{\lambda\,\beta_i}}, \tag{4.10b}$$

  wobei $\mathrm{H}_2(\cdot)$ die binäre Entropiefunktion nach Gleichung (4.7) ist.
- Wenn die Wahrscheinlichkeiten p_i die Ungleichung

$$0 < p_i \leq 1 - 2\,\gamma_i, \quad 0 \leq i < n \tag{4.11}$$

  erfüllen, dann wird das diskrete Ellipsoid $\mathcal{E}(r, \boldsymbol{\beta}, \mathbf{0})$ mit der minimalen Anzahl von Durchläufen des Algorithmus überdeckt.

Offensichtlich werden alle Wahrscheinlichkeiten p_i durch Gleichung (4.9) und Ungleichung (4.11) nicht eindeutig bestimmt. Trotzdem können (4.9)

und (4.11) stets erfüllt werden, denn $\gamma_i < 1\,/\,2$ gilt für $\lambda > 0$. In Abschnitt 5.3 werden die Wahrscheinlichkeiten p_i als Funktionen der oberen Grenze $1 - 2\,\gamma_i$ normiert.

Da jede Stelle i mit Hilfe eines gleichverteilten Zufallszahlengenerators nach Unterabschnitt 2.7.1 mit einer Wahrscheinlichkeit p_i, $0 < p_i < 1$ ausgewählt wird, liefert die ständige Wiederholung der obengenannten Methode jedesmal eine unterschiedliche Menge von Stellen. Somit wird die gewünschte ellipsoidische Überdeckungsmenge zufällig erzeugt.

4.4.2 Bildung von Informationsmengen

In diesem Unterabschnitt wird ein Algorithmus zur Bildung der zuverlässigsten Informationsmenge $\mathcal{I}$ vorgestellt. Er basiert einerseits auf einer gewöhnlichen gaußschen Eliminationsverfahren [Bar97, BSMM99, Stö99] und andererseits auf einer Kombination der in [HH92, HHC91] und [Bar91a, Bar93, BGW97] vorgeschlagenen Methoden. Anschließend wird ein Beispiel mittels einen $(7, 4, 3)_2$-Codes $\mathcal{C}$ angegeben.

Vorausgesetzt werden ein $(n, k)_2$-Code $\mathcal{C}$ mit einer Generatormatrix $\boldsymbol{G}$ gemäß (2.4) und eine geordnete Menge $\mathcal{L}$ von Stellen, die nach der Methode von Unterabschnitt 4.4.1 erzeugt wurde und aus der eine Informationsmenge $\mathcal{I}$ möglicherweise gebildet werden kann. Diese geordnete Menge $\mathcal{L}$ wird folgendermaßen definiert:

$$\mathcal{L} = \{l_0, l_1, \dots\}, \quad \beta_{l_0} \geq \beta_{l_1} \geq \cdots, \quad k \leq |\mathcal{L}| \leq n. \tag{4.12}$$

Dabei bezeichnet l_0 die zuverlässigste Stelle, l_1 die zweitzuverlässigste Stelle, usw.

Eine Prozedur zur Bildung der bestmöglichen, zuverlässigsten Informationsmenge $\mathcal{I}$ aus der geordneten Menge $\mathcal{L}$ kann wie folgt formuliert werden:

- **Schritt 1:** Initialisierungsschritt: Setze den Pivotindex ϱ auf Null.
- **Schritt 2:** Falls das Pivotelement g_{ϱ,l_ϱ} von $\boldsymbol{G}$ ungleich Null ist, dann gehe zum Schritt 7.
- **Schritt 3:** Suche in der Pivotspalte l_ϱ von $\boldsymbol{G}$, unter der Pivotzeile ϱ, ein Element ungleich Null, das das neue Pivotelement sein wird.
- **Schritt 4:** Falls dieses Element gefunden wird, vertausche die ganze Zeile von $\boldsymbol{G}$, in der sich dieses Element befindet, mit der Pivotzeile ϱ und gehe zum Schritt 7.

- **Schritt 5:** Falls das Element ungleich Null nicht gefunden werden kann, dann entferne diese Spalte l_ϱ aus der geordneten Menge $\mathcal{L}$.

- **Schritt 6:** Falls die neue geordnete Menge weniger als k Elemente enthält, dann kann eine Informationsmenge $\mathcal{I}$ aus der ursprünglichen geordneten Menge $\mathcal{L}$ nicht gebildet werden und der Algorithmus wird beendet. Sonst gehe zum Schritt 2 zurück.

- **Schritt 7:** Nach der erfolgreichen Pivotsuche setze alle Elemente der Spalte l_ϱ mittels elementaren Zeilenoperationen in $\boldsymbol{G}$ auf Null, außer dem Pivotelement g_{ϱ,l_ϱ}.

- **Schritt 8:** Erhöhe den Pivotindex ϱ um Eins und Wiederhole die Schritte 2 bis 7 solange $\varrho < k$.

- **Schritt 9:** Beende den Algorithmus. Somit konnte die ursprüngliche Generatormatrix $\boldsymbol{G}$ in eine andere Generatormatrix $\boldsymbol{G}'(\mathcal{I})$ überführt werden, indem die ersten k Stellen der neue geordnete Menge die bestmögliche, zuverlässigste Informationsmenge $\mathcal{I}$ bilden.

Anhand eines Beispiels wird diese Prozedur deutlicher. Derselbe $(7, 4, 3)_2$-Code $\mathcal{C}$ wird verwendet, der im Unterabschnitt 2.2.3 durch die Generatormatrix (2.29) angegeben wird:

$$\boldsymbol{G} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \tag{2.29}$$

Es wird angenommen, daß folgende geordnete Menge $\mathcal{L}$ mittels des Algorithmus von Unterabschnitt 4.4.1 vorher ausgewählt wurde:

$$\mathcal{L} = \{2, 1, 3, 6, 0, 5\}. \tag{4.13}$$

Die bestmögliche, zuverlässigste Informationsmenge $\mathcal{I}$ wird durch folgende Schritte (a)–(v) gebildet:

(a) **Schritt 1:** Pivotindex $\varrho \leftarrow 0$.

(b) **Schritt 2:** Pivotelement $g_{0,2} = 0$.

(c) **Schritt 3:** Neues Pivotelement: $g_{1,2} = 1$.

(d) **Schritt 4:**

$$\boldsymbol{G} \leftarrow \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \tag{4.14}$$

(e) **Schritt 7:**

$$\boldsymbol{G} \leftarrow \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \tag{4.15}$$

(f) **Schritt 8:** Pivotindex $\varrho \leftarrow 1$.

(g) **Schritt 2:** Pivotelement $g_{1,1} = 1$.

(h) **Schritt 7:**

$$\boldsymbol{G} \leftarrow \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \tag{4.16}$$

(i) **Schritt 8:** Pivotindex $\varrho \leftarrow 2$.

(j) **Schritt 2:** Pivotelement $g_{2,3} = 0$.

(k) **Schritt 3:** Neues Pivotelement: $g_{3,3} = 1$.

(l) **Schritt 4:**

$$\boldsymbol{G} \leftarrow \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}. \tag{4.17}$$

(m) **Schritt 7:**

$$\boldsymbol{G} \leftarrow \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}. \tag{4.18}$$

(n) **Schritt 8:** Pivotindex $\varrho \leftarrow 3$.

(o) **Schritt 2:** Pivotelement $g_{3,6} = 0$.

(p) **Schritt 3:** Ein neues Pivotelement kann in der Pivotspalte 6 nicht gefunden werden.

(q) **Schritt 5:**

$$\mathcal{L} \leftarrow \{2, 1, 3, 0, 5\}. \tag{4.19}$$

(r) **Schritt 6:** $|\mathcal{L}| = 5$.

(s) **Schritt 2:** Pivotelement $g_{3,0} = 1$.

(t) **Schritt 7:**

$$\boldsymbol{G} \leftarrow \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}. \tag{4.20}$$

(u) **Schritt 8:** Pivotindex $\varrho \leftarrow 4$.

(v) **Schritt 9:** Der Algorithmus wird beendet, wobei die Generatormatrix $\boldsymbol{G}'(\mathcal{I})$ durch (4.20) und die bestmögliche, zuverlässigste Informationsmenge $\mathcal{I}$ durch die ersten 4 Stellen von (4.19) angegeben ist:

$$\boldsymbol{G}'(\mathcal{I}) = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \tag{4.21}$$

$$\mathcal{I} = \{2, 1, 3, 0\}. \tag{4.22}$$

4.4.3 Allgemeines Decodierverfahren mit Akzeptanzkriterium

Vorausgesetzt werden ein $(n, k)_2$-Code $\mathcal{C}$ mit einer Generatormatrix $\boldsymbol{G}$, ein vom Empfangsvektor $\boldsymbol{r}$ abhängiger Vektor $\boldsymbol{y}$ bzw. $\boldsymbol{v}$ bzw. $\boldsymbol{\beta}$, der mit Soft–Decision decodiert werden soll, und eine Zahl $\mathcal{I}_{\max} \in \mathbb{N}$, die die maximale Anzahl der gebildeten Informationsmengen $\mathcal{I}$ bezeichnet.

Im folgenden bezeichnet die Variable ι die Anzahl der gebildeten Informationsmengen $\mathcal{I}$ und Υ eine Korrelationsschwelle, die die Bewertung verschiedener Iterationsschritte ermöglicht.

Eine allgemeine Prozedur zur Decodierung mit Hilfe von Informationsmengen und eines Akzeptanzkriteriums kann folgendermaßen formuliert werden:

- **Schritt 1:** fakultativ: Wird der Syndrom–Test gemäß Gleichung (3.1) erfüllt, dann setzte das geschätzte Codewort $\hat{\boldsymbol{c}}$ gleich dem Hard–Decision Vektor $\boldsymbol{v}$ und gehe zum Schritt 10.

- **Schritt 2:** Initialisierungsschritt:

  - **Schritt 2.1:** Setze die Variable ι auf Null.
  - **Schritt 2.2:** Setze das geschätzte Codewort $\hat{\boldsymbol{c}}$ gleich dem Nullwort $\mathbf{0}$.
  - **Schritt 2.3:** Setze die Korrelationsschwelle Υ gleich dem Skalarprodukt $\langle \hat{\boldsymbol{x}}, \boldsymbol{y} \rangle$, wobei die geschätzte Sendefolge $\hat{\boldsymbol{x}}$ aus $\hat{\boldsymbol{c}}$ bestimmt wird.
  - **Schritt 2.4:** Berechne die Zuordnung der Komponenten β_i des Zuverlässigkeitsvektors $\boldsymbol{\beta}$.
  - **Schritt 2.5:** Löse Gleichung (4.10) nach dem Parameter λ auf.
  - **Schritt 2.6:** Berechne die Wahrscheinlichkeiten p_i (genaueres siehe Abschnitt 5.3).

- **Schritt 3:** Erzeuge eine zufällige Menge von mindestens k Stellen nach der Methode des Unterabschnittes 4.4.1 mit den Wahrscheinlichkeiten p_i und bilde damit die geordnete Menge $\mathcal{L}$ gemäß Gleichung (4.12).

- **Schritt 4:** Bilde eine Informationsmenge $\mathcal{I}$ und die entsprechende Generatormatrix $\boldsymbol{G}'(\mathcal{I})$ nach der Prozedur des vorherigen Unterabschnittes. Falls eine Informationsmenge $\mathcal{I}$ aus der geordneten Menge $\mathcal{L}$ nicht gebildet werden kann, dann gehe zum Schritt 3 zurück.

- **Schritt 5:** Erhöhe ι um Eins und bilde ein Infowort $\boldsymbol{u}_{[\mathcal{I}]}$, das diejenigen k Komponenten aus dem Hard–Decision Vektor $\boldsymbol{v}$ enthält, die durch $\mathcal{I}$ bestimmt werden. Damit berechne ein Codewort $\boldsymbol{c}'(\mathcal{I})$ mittels einer ähnlichen Codiervorschrift wie Gleichung (2.5): $\boldsymbol{c}'(\mathcal{I}) = \boldsymbol{u}_{[\mathcal{I}]}\,\boldsymbol{G}'(\mathcal{I})$.

- **Schritt 6:** Falls die Korrelationsschwelle $\Upsilon \geq \langle \boldsymbol{x}'(\mathcal{I}), \boldsymbol{y} \rangle$ gilt, wobei $\boldsymbol{x}'(\mathcal{I})$ aus $\boldsymbol{c}'(\mathcal{I})$ bestimmt wird, gehe zum Schritt 9.

- **Schritt 7:** Setze $\hat{\boldsymbol{c}}$ gleich dem Codewort $\boldsymbol{c}'(\mathcal{I})$, das bisher die beste Schätzung des gesendeten Codewortes $\boldsymbol{c}$ ist. Setze die neue Korrelationsschwelle Υ auf $\langle \boldsymbol{x}'(\mathcal{I}), \boldsymbol{y} \rangle$.

- **Schritt 8:** Führe eines der in Kapitel 3 beschriebenen Akzeptanzkriterien zwischen der geschätzten Sendefolge $\hat{\boldsymbol{x}}$ und $\boldsymbol{y}$ durch. Wird dieses Kriterium erfüllt, dann gehe zum Schritt 10.

- **Schritt 9:** Falls $\iota < \mathcal{I}_{\max}$, gehe zum Schritt 3 zurück.

- **Schritt 10:** Beende den Algorithmus. Das geschätzte Codewort $\hat{\boldsymbol{c}}$ ist die beste Schätzung des gesendeten Codewortes $\boldsymbol{c}$.

4.5 Abschließende Bemerkungen

Das Hauptziel dieses Kapitels war die Erläuterung einer Informationsmengen–Decodierung bei sowohl Hard–Decision als auch Soft–Decision.

Wie in den beiden ersten Abschnitten vorgestellt wurde, hat die Arbeit von Evseev [Evs83] eine Reihe von Untersuchungen über Prozeduren zur näherungsweisen HDMDD mit geringer Komplexität ermöglicht. Seitdem wurden viele leistungsfähige allgemeine Decodieralgorithmen in Hard–Decision Kanälen entwickelt.

Obwohl Dumer die Ergebnisse von Evseev für Soft–Decision Kanälen verallgemeinert hat [Dum96b], wurde in Abschnitt 4.3 besonders betont, daß das Kernproblem in diesem Fall durch die Abhängigkeit der Menge der wahrscheinlichsten Fehlervektoren vom Zuverlässigkeitsvektor $\boldsymbol{\beta}$ verursacht wird.

Später hat Dumer in vielen Artikeln gezeigt [Dum96a, Dum97a, Dum97b, Dum98a, Dum98b, Dum98c, Dum], daß die asymptotische Lösung auf dem Konzept eines diskreten Ellipsoids basiert.

Er hat hingegen in seinen Arbeiten lediglich Codes im asymptotischen Fall analysiert. Andererseits ist das wesentliche Ziel der vorliegenden Arbeit die Betrachtung konkreter Codes. Deswegen wurde in Abschnitt 4.4 auf ein Decodierverfahren mit Hilfe von Informationsmengen im Soft–Decision Fall eingegangen. Im ersten Unterabschnitt wurde eine Methode zur Erzeugung einer minimalen ellipsoidischen Überdeckungsmenge vorgeschlagen, die ein beliebiges, diskretes Ellipsoid überdeckt. Der Unterabschnitt 4.4.2 behandelte eine Prozedur zur Bildung der bestmöglichen Informationsmenge, die für die Decodierung gebraucht wird. Das Verständnis dieser Prozedur wurde mittels eines ausführlichen Beispiels vermittelt.

Schließlich enthält der letzte Unterabschnitt ein Decodierverfahren, das für alle linearen Codes $\mathcal{C}$ verwendet werden kann. Wegen der Anwendung der in Kapitel 3 hergeleiteten Akzeptanzkriterien wird dieses Decodierverfahren besonders für konkrete, lange Codes $\mathcal{C}$ sehr leistungsfähig [Bar99], wie im nächsten Kapitel mit Hilfe von Simulationsergebnissen veranschaulicht wird.

Kapitel 5

Simulationsergebnisse

Hauptziel dieses Kapitels ist, die durchgeführten Untersuchungen und wesentlichen Simulationsergebnisse der im vorherigen Kapitel vorgeschlagenen Decodiermethode in Verbindung mit den in Kapitel 3 hergeleiteten Akzeptanzkriterien darzustellen.

Um Simulationen langer und leistungsfähiger Codes $\mathcal{C}$ in endlichen Ausführungszeiten und mit vernünftiger Genauigkeit bei WER in einigen Fällen sogar kleiner als 10^{-5} zu ermöglichen, wurden die Simulationsprogramme mithilfe von bestimmten Programmiertechniken optimiert. Die wichtigsten Techniken werden zunächst in Abschnitt 5.1 kurz zusammengefaßt. Danach wird die praktische Realisierung des Übertragungssystems von Abbildung 2.1 erörtert. Ferner werden einige Implementierungsaspekte genauer betrachtet.

Wie am Ende von Unterabschnitt 4.4.1 kurz darauf hingewiesen wurde, werden alle Wahrscheinlichkeiten p_i durch Gleichung (4.9) und Ungleichung (4.11) nicht eindeutig bestimmt. Einer der wesentlichsten Beiträge dieser Arbeit ist, durch Simulationen zu zeigen, daß verschiedene Methoden zur Berechnung dieser Wahrscheinlichkeiten p_i zu ganz unterschiedlichen WER bzw. Decodierkomplexitäten führen. Diese Methoden, die sogenannten Normierungen, werden in Abschnitt 5.3 eingeführt und durch Simulationen nachgewiesen.

Insbesondere wird eine Referenzkurve für die SDMLD eines $(128, 64, 22)_2$-Codes gezeigt, die erstmals so genau für kleine BER simuliert wurde. Der nächste Abschnitt befaßt sich mit dem Vergleich der Effizienz der in Kapitel 3 hergeleiteten Akzeptanzkriterien. Es folgt die Darstellung einer wesentlichen Verbesserung dieser Akzeptanzkriterien für lange Codes, wie der $(128, 64, 22)_2$-Code.

Im vorletzten Abschnitt werden die vollständigen Simulationsergebnisse eines Codes der Länge $n = 255$ analysiert und mit theoretischen Grenzen wie Kanalkapazität oder *Cutoff–Rate* [CC81, Hub92, Fri95, Pro95] verglichen. Anschließend wird eine kurze Überblick über das ganze Kapitel angegeben.

5.1 Programmieraspekte

Die Simulationen wurden in der Programmiersprache C [KR77] geschrieben und unter den Betriebsystemen UNIX [Per94] bzw. Linux [JT98] auf SUN Workstations bzw. PC Rechnern durchgeführt.

Damit die Decodierung langer Blockcodes mit akzeptablen Ausführungszeiten und genügend Genauigkeit simuliert werden konnte, wurden u. a. folgende Programmiertechniken verwendet:

- Ausnutzung von vielen Kompilationsdirektiven des Übersetzungsprogrammes (*Compiler*),
- Benutzung von Makro–Ersetzungen [KR77] anstelle bestimmter Variablen (z. B. die Parameter des Codes), die während des Durchlaufes einer Simulation konstant bleiben,
- Ersetzung aller zweidimensionalen Felder (Matrizen) durch Zeiger auf Zeilen von Zeigern (*Pointers–to–Row–of–Pointers*) [PTVF92],
- intensive Anwendung von Zeigern (*Pointers*) [KR77] und
- möglichst Verwendung von Ganzzahl–Variablen (*Integer Variables*) anstatt Fließkommazahl–Variablen (*Floating Point Variables*).

Wenn beispielsweise nur ein paar bestimmte Kompilationsdirektiven benutzt werden, ohne eine einzige Zeile im Programm zu ändern, erhöht sich die Geschwindigkeit der Ausführung für manche Übersetzungsprogramme um 100 bis 250 %.

Mit Berücksichtigung aller obengenannten Punkte und einer sorgfältigen Programmierung haben sich die Simulationsdauern um den Faktor 10 (für kurze Codes) bis 90 (für lange Codes) verkürzt.

5.2 Implementierung

Im folgenden wird die Implementierung sowohl des in Abschnitt 2.3 vorgestellten Übertragungssystems als auch des in Unterabschnitt 4.4.3 beschriebenen Decodieralgorithmus ausführlich erklärt.

Die Informationsquelle wird mit dem gleichverteilten Zufallszahlengenerator von Unterabschnitt 2.7.1 realisiert. Bei den Simulationen werden alle Infowörter $\boldsymbol{u}$ bzw. Codewörter $\boldsymbol{c}$ bzw. Sendefolge $\boldsymbol{x}$ zufällig und gleichwahrscheinlich erzeugt.

Für einen angegebenen $(n, k)_2$-Code $\mathcal{C}$ wird eine systematische Codierung mittels einer Generatormatrix $\boldsymbol{G}$ in der kanonischen Staffelform gemäß Gleichung (2.28) durchgeführt.

Der zeitdiskrete Kanal verwendet eine antipodische Signalisierung mit BPSK als Modulationsverfahren, einen mithilfe eines gaußverteilten Zufallszahlengenerators nach Unterabschnitt 2.7.2 erzeugten AWGN–Kanal und einen kohärenten Empfänger (d. h. eine ideale Träger- und Phasensynchronisation) mit Soft–Decision Demodulation.

Üblicherweise werden Simulationen mit einer begrenzten Anzahl von Codewörtern durchgeführt. Während in [GS97] 35000 Codewörter $\boldsymbol{c}$ für jedes SNR gesendet wurden, ist die Anzahl der Codewörter für jedes SNR in [KNIH94] gleich 100000. Diese Anzahl braucht jedoch für alle simulierten SNR nicht unbedingt konstant zu bleiben. Beispielsweise wurden 35000 Codewörter für SNR zwischen 3 und 4 dB und 500000 Codewörter für SNR zwischen 5 und 8 dB in [SWHM98] gesendet.

Der Nachteil einer begrenzten Anzahl von gesendeten Codewörtern ist, daß die Anzahl von Wortfehlern im zeitdiskreten Kanal variiert. Einerseits werden zu viele Codewörter für kleine SNR gesendet, so daß die Simulationen unnötigerweise lange dauern. Andererseits werden die Ergebnisse für große SNR wesentlich ungenauer als für kleine. Falls weniger als 10 Wortfehler im zeitdiskreten Kanal aufgetreten sind, werden die Simulationen so unpräzis, daß ihre Ergebnisse nutzlos werden. Deswegen zeigen die Abbildungen in den obengenannten Arbeiten keine WER wesentlich kleiner als 10^{-3}.

Aus diesen Gründen werden die Simulationen dieser Arbeit mit der Simulationsmethode von Abschnitt 2.8 und einer unbegrenzten maximalen Anzahl von Codewörtern durchgeführt. Sie werden für jeden quadratischen Abstand Δ^2 erst abgebrochen, wenn die Anzahl von Wortfehlern größer als 100 wird und mindestens 1000 Codewörtern gesendet werden. Diese minimale Anzahl von Codewörtern gewährleistet, daß die Ergebnisse für große quadratische

Abstände Δ^2 bzw. kleine SNR ohne großen Aufwand ziemlich präzis bleiben.

Da die vollständige Simulation einer einzigen Kurve durch die ständige Verkleinerung des quadratischen Abstandes Δ^2 durchgeführt wird, bis die gemessene WER am Ausgang des Decodierers in einigen Fällen sogar weniger als 10^{-5} beträgt, wird die gesamte Anzahl von gesendeten Codewörtern manchmal wesentlich größer als 10 Millionen.

Im Decodieralgorithmus des Unterabschnittes 4.4.3 wird die Zuordnung der Komponenten β_i des Zuverlässigkeitsvektors $\boldsymbol{\beta}$ in Schritt 2.4 durch einen *Quicksort*–Algorithmus [Knu98b] gemäß [PTVF92, VTPF92] implementiert, obwohl ein *Heapsort*–Algorithmus [PTVF92, VTPF92, Knu98b] wegen seiner ähnlichen Komplexität für die Länge n der in dieser Arbeit berücksichtigten Codes auch getestet wurde und anwendbar sein könnte.

Das Auflösen der nichtlinearen Gleichung (4.10) nach dem Parameter λ in Schritt 2.5 ist lediglich mittels numerischen Verfahren durchführbar. Mögliche Iterationsverfahren sind die sogenannten Bisektions–, Sekanten–, Regula falsi– und Newton–Verfahren [PTVF92, VTPF92, BSMM99, Stö99], die für Gleichungen mit transzendenten Termen geeignet sind. Die in den Simulationen verwendete Methode wurde aus [PTVF92, VTPF92] entnommen und basiert auf der Arbeit von Brent [Bre73].

Für die Berechnung der Wahrscheinlichkeiten p_i in Schritt 2.6 wird eine der im nächsten Abschnitt vorgeschlagenen Normierungen verwendet.

Wie bereits in Unterabschnitt 4.4.1 erwähnt wurde, wird die Menge von mindestens k Stellen in Schritt 3 mit Hilfe eines gleichverteilten Zufallszahlengenerators nach Unterabschnitt 2.7.1 erzeugt.

Für jeden Durchlauf einer Simulation werden eins der in Kapitel 3 hergeleiteten Akzeptanzkriterien und ein bestimmter Wert für die maximale Anzahl $\mathcal{I}_{\max}$ der gebildeten Informationsmengen $\mathcal{I}$ benutzt. Des weiteren wird eine untere Simulationsschranke P_{LB} für die WER einer SDMLD gemäß der Methode von Abschnitt 2.9 berechnet.

5.3 Normierungen

In seinen Arbeiten [Dum97b, Dum98a, Dum98c] hat Dumer nicht darauf hingewiesen, wie die Wahrscheinlichkeiten p_i von Unterabschnitt 4.4.1 berechnet werden sollen, weil er nur eine allgemeine Methode zur Konstruktion einer minimalen ellipsoidischen Überdeckungsmenge im asymptotischen Fall vorgeschlagen hat.

Als wesentlicher Beitrag der vorliegenden Arbeit wurde durch Simulationen festgestellt, daß verschiedene Methoden zur Berechnung dieser Wahrscheinlichkeiten p_i zu ganz unterschiedlichen Ergebnissen führen können, obwohl alle erzeugten ellipsoidischen Überdeckungsmengen noch minimal bleiben.

Im folgenden werden drei mögliche Methoden dargestellt, in denen die Wahrscheinlichkeiten p_i als Funktionen der entsprechenden γ_i nach Gleichung (4.10b) so normiert werden, daß sie sowohl Gleichung (4.9) als auch Ungleichung (4.11) erfüllen. Daß bei all diesen drei Methoden die obere Grenze von p_i in Ungleichung (4.11) eingehalten ist, wurde bei den Simulationen der verschiedenen Codes jeweils im Einzelnen überprüft.

- Lineare Normierung:

$$p_i = \eta_1(1 - 2\,\gamma_i), \tag{5.1}$$

  wobei der lineare Normierungsfaktor η_1 Gleichung (4.9) erfüllen soll:

$$\sum_{i=0}^{n-1} \eta_1(1 - 2\,\gamma_i) = k. \tag{5.2}$$

  Daraus folgt:

$$\eta_1 = \frac{k}{\sum\limits_{i=0}^{n-1}(1 - 2\,\gamma_i)} = \frac{k}{n - 2\sum\limits_{i=0}^{n-1}\gamma_i}. \tag{5.3}$$

- Quadratische Normierung:

$$p_i = \eta_2(1 - 2\,\gamma_i)^2. \tag{5.4}$$

  Gleichung (4.9) soll ebenfalls vom quadratischen Normierungsfaktor η_2 erfüllt werden:

$$\sum_{i=0}^{n-1} \eta_2(1 - 2\,\gamma_i)^2 = k. \tag{5.5}$$

  Daraus folgt:

$$\eta_2 = \frac{k}{\sum\limits_{i=0}^{n-1}(1 - 2\,\gamma_i)^2}. \tag{5.6}$$

- Kubische Normierung:

$$p_i = \eta_3 (1 - 2\,\gamma_i)^3. \tag{5.7}$$

Aus Gleichung (4.9) folgt der kubische Normierungsfaktor η_3:

$$\eta_3 = \frac{k}{\sum_{i=0}^{n-1} (1 - 2\,\gamma_i)^3}. \tag{5.8}$$

Der Zusammenhang zwischen den obengenannten Normierungen und der WER wurde durch Simulationen untersucht. Der ausgewählte Code $\mathcal{C}$ ist ein erweiterter Bose–Ray-Chaudhuri–Hocquenghem–Code (eBCH-Code, *extended Bose–Ray-Chaudhuri–Hocquenghem–Code*) der Länge $n = 128$, Dimension $k = 64$ und Minimalabstand $d_{H\min} = 22$. Da lediglich binäre Codes in dieser Arbeit betrachtet werden, wurde auf eine formale Definition von Bose–Ray-Chaudhuri–Hocquenghem–Codes (BCH-Codes) in Kapitel 2 verzichtet. Für eine umfassende Darstellung dieser Codes wird auf [MS83, Fri95, Bos98] verwiesen.

Gemäß [BV93] ist die obere Schranke für den Minimalabstand $d_{H\min}$ eines linearen $(127, 64)_2$-Codes $\mathcal{C}$ gleich 21. Das bedeutet, daß kein anderer linearer Code mit den Parametern $n = 127$ und $k = 64$ existiert, der einen größeren Minimalabstand $d_{H\min}$ aufweist als den des $(127, 64, 21)_2$-BCH-Codes. Somit ist der entsprechende $(128, 64, 22)_2$-eBCH-Code der beste Code mit den Parametern $n = 128$ und $k = 64$, falls es keinen anderen, noch zu entdeckenden Code gibt, der ein besseres Gewichtsprofil $\mathcal{W}$ besitzt als das dieses leistungsfähigen eBCH-Codes.

Weiterhin wurde der $(128, 64, 22)_2$-eBCH-Code ebenfalls sehr häufig für Simulationen ausgewählt [Dor74, HHC91, HH92, KNIH94, FL95, FL96b, GS97, KNH97, Han98, SWHM98, Bar99], weil Codes mit Rate $R = 1\,/\,2$ den schwierigsten Fall bezüglich der Decodierkomplexität darstellen.

Die maximale Anzahl $\mathcal{I}_{\max}$ der gebildeten Informationsmengen wurde für jede verschiedene Normierung u. a. auf 50, 500 und 5000 begrenzt. Zusätzlich zu diesen 9 Kurven wurde die untere Simulationsschranke P_{LB} für SDMLD von Abschnitt 2.9 mit $\mathcal{I}_{\max} = 5000$ und einer quadratischen Normierung berechnet. In Abbildung 5.1 wird die Darstellung der WER in Abhängigkeit vom quadratischen Abstand Δ^2 veranschaulicht. Abbildung 5.2 zeigt die entsprechende BER in Abhängigkeit von SNR pro empfangenem Infobit $E_b\,/\,N_0$.

Um Vergleiche mit der Fehlerrate P_u einer uncodierten Übertragung zu ermöglichen, wurde diese Kurve in Abbildung 5.2 auch dargestellt und wie folgt berechnet [Fri95, Pro95, DB96, Kam96]:

$$P_u = \frac{1}{2}\operatorname{erfc}\Big(10^{0.05\frac{E_b}{N_0}[dB]}\Big). \tag{5.9}$$

Dabei steht

$$\operatorname{erfc}(\varsigma) = \frac{2}{\sqrt{\pi}}\int_{\varsigma}^{\infty} e^{-\tau^2} d\tau \tag{5.10}$$

für die komplementäre Fehlerfunktion [DB96, Kam96] (*Complementary Error Function* [Pro95], ebenfalls konjugierte Fehlerfunktion [Stö99] genannt).

Wie bereits in Abschnitt 2.8 erwähnt wurde, sind die Simulationen für kleine quadratische Abstände Δ^2 (entsprechen großen SNR) mit erheblichem Aufwand verbunden. Deswegen wurden nicht alle Kurven von Abbildung 5.1 bis zu einer WER kleiner als 10^{-5} durchgeführt. Stattdessen wurden die fehlenden Werte, die zur Berechnung der Kurven von Abbildung 5.2 benötigt werden, mithilfe der entsprechenden zwei untersten Punkte in Abbildung 5.1 linear extrapoliert. Damit sind die Kurven in Abbildung 5.2 eigentlich obere Schranken, die unter Umständen ungenau für große $E_b \,/\, N_0$ werden können, wie beispielsweise für $\mathcal{I}_{\max} = 500$ und eine kubische Normierung deutlich zu erkennen ist. Diese Ungenauigkeiten der Simulationsmethode sollen mit einer möglichen Eigenschaft des $(128, 64, 22)_2$-eBCH-Codes oder des Decodieralgorithmus nicht verwechselt werden.

Aus den beiden Abbildungen ergibt sich folgende Überlegungen:

- Obschon eine lineare Normierung nach Gleichung (5.1) zunächst ganz logisch erscheint, ist sie keinesfalls die beste Auswahl.

- Wenn eine kleine Decodierkomplexität (beispielsweise mit $\mathcal{I}_{\max} = 50$) bei einer BER größer als 10^{-5} gewünscht wird, sollte eine kubische Normierung gemäß Gleichung (5.7) verwendet werden. Abbildung 5.2 zeigt deutlich, daß eine kleinere BER am Ausgang des Decodierers gemessen wird, wenn eine kubische Normierung anstelle einer quadratischen oder linearen Normierung benutzt wird, obwohl diese BER wesentlich schlechter ist als die für eine größere Decodierkomplexität.

- Für mittlere Decodierkomplexität (z. B. mit $\mathcal{I}_{\max} = 500$) ist die kubische Normierung nur besser bei großem Δ^2 bzw. kleinem SNR. Bessere

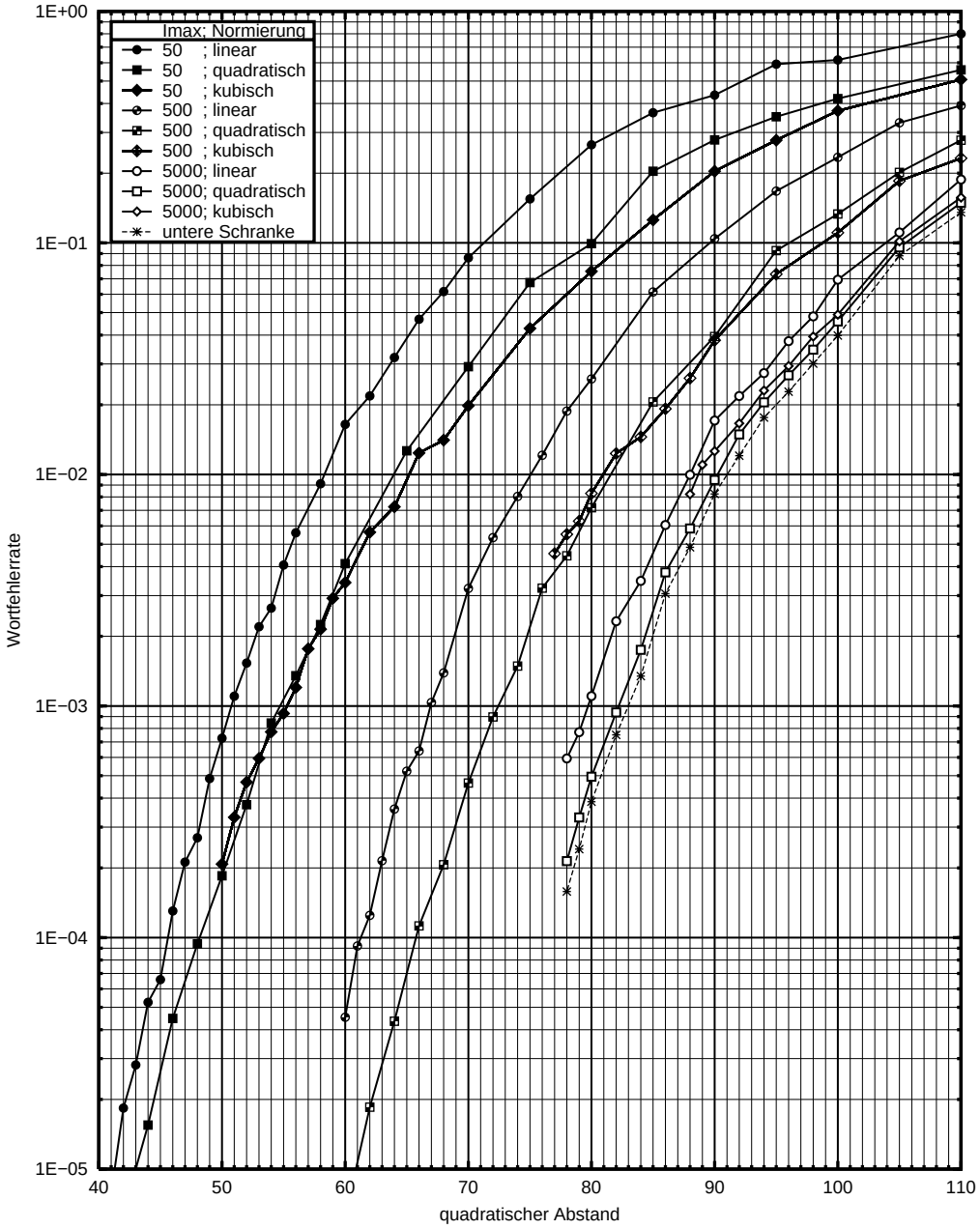

Abbildung 5.1: WER in Abhängigkeit von Δ^2 bei $\mathcal{I}_{\max} = 50$, 500 und 5000 und verschiedenen Normierungen für den $(128, 64, 22)_2$-eBCH-Code.

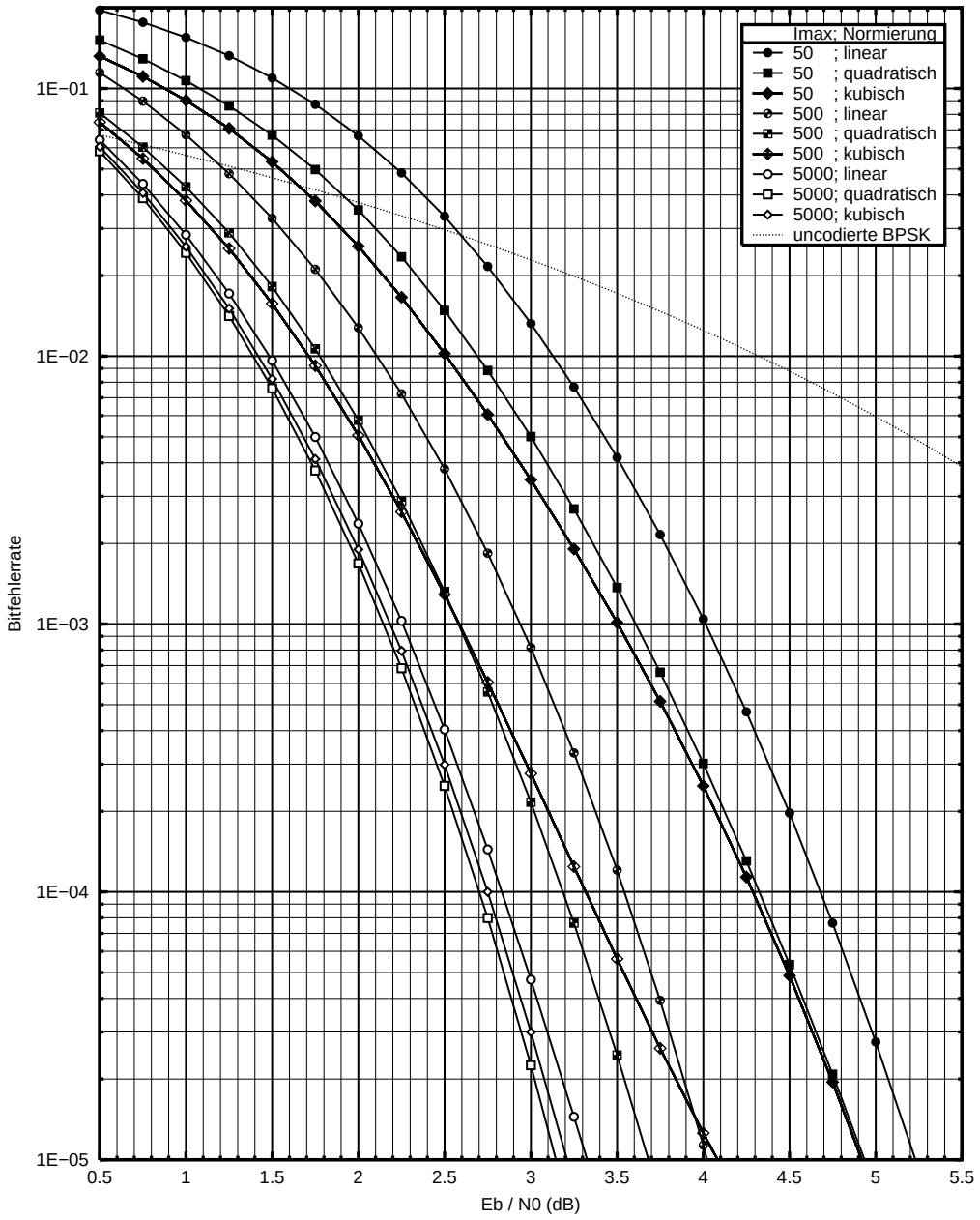

Abbildung 5.2: BER in Abhängigkeit von SNR bei $\mathcal{I}_{\max} = 50$, 500 und 5000 und verschiedenen Normierungen für den $(128, 64, 22)_2$-eBCH-Code.

Ergebnisse werden erreicht, falls eine quadratische Normierung angewendet wird. Beispielsweise beträgt der Decodierungsgewinn bei einer BER von 10^{-5} etwa 1.24 dB, wenn $\mathcal{I}_{\max} = 500$ anstatt von $\mathcal{I}_{\max} = 50$ benutzt wird.

- Eine sehr leistungsfähige Decodierung wird durchgeführt, wenn eine große maximale Anzahl der gebildeten Informationsmengen ($\mathcal{I}_{\max} = 5000$) mit einer quadratischen Normierung nach Gleichung (5.4) erlaubt wird.

Obgleich nur eine untere Schranke für SDMLD in Abbildung 5.1 dargestellt ist, liegt das Simulationsergebnis für $\mathcal{I}_{\max} = 5000$ und quadratische Normierung sehr dicht bei dieser Schranke. Da die tatsächliche SDMLD möglicherweise etwa in der Mitte dieser beiden Kurven liegt, führt der Decodieralgorithmus in diesem Fall praktisch eine reine SDMLD. Aus diesen Gründen stellt die Kurve für $\mathcal{I}_{\max} = 5000$ und quadratische Normierung in Abbildung 5.2 eine Referenzkurve für die SDMLD eines $(128, 64, 22)_2$-eBCH-Codes dar, die erstmals so präzis für kleine BER simuliert wurde.

5.4 Effizienz verschiedener Akzeptanzkriterien

Alle Simulationen dieser Arbeit wurden mit dem Syndrom–Test von Unterabschnitt 3.1.1 begonnen, obwohl dieser Test bei den simulierten quadratischen Abständen Δ^2 für lange Codes $\mathcal{C}$ nur selten zum Abbruch führt.

Damit die Effizienz der in Kapitel 3 eingeführten Akzeptanzkriterien hinsichtlich des durchschnittlichen Prozentsatzes der Abbrüche miteinander verglichen werden konnten, wurde in einer ersten Untersuchung derselbe $(7, 4, 3)_2$-Code $\mathcal{C}$ von Unterabschnitt 4.4.2 simuliert. Es wurde folgendes festgestellt:

- Selbst wenn der Hyperkugel–Test von Unterabschnitt 3.1.2 sehr leicht implementiert werden kann, ist er kein gutes Akzeptanzkriterium und wäre deswegen völlig sinnlos für große Codes $\mathcal{C}$.

- Trotz der etwas besseren Ergebnisse unter Anwendung des GMD–Tests von Unterabschnitt 3.1.4 anstelle des Hyperkugel–Tests sollte er für große Codes $\mathcal{C}$ auch nicht benutzt werden.

- Für den simulierten Code $\mathcal{C}$ zeigte sich der Hyperkreiskegel–Test von Unterabschnitt 3.1.3 deutlich besser als die beiden vorherigen Akzeptanzkriterien. Dieses Akzeptanzkriterium ist eine gute Auswahl für den Fall, daß eine kleine Komplexität der Implementierung einer Abbruchregel gewünscht wird, denn es verwendet keinerlei Zuordnung der Komponenten β_i des Zuverlässigkeitsvektors $\boldsymbol{\beta}$.

- Unter Verwendung des Akzeptanzkriteriums nach Taipale und Pursley von Unterabschnitt 3.1.5 konnte die durchschnittliche Anzahl der gebildeten Informationsmengen wesentlich verringert werden. Wie bereits in Abschnitt 3.2 erwähnt wurde, ist dies die bestmögliche Bedingung Ξ basierend auf einem einzigen Codewort für eine SDMLD.

- Eine geringere durchschnittliche Anzahl der gebildeten Informationsmengen kann erreicht werden, falls das Akzeptanzkriterium nach Kasami basierend auf zwei Codewörtern von Unterabschnitt 3.2.3 implementiert wird. Dieses Akzeptanzkriterium ist besonders attraktiv für den Fall, in dem die Komplexität der Abbruchregel keine entscheidende Rolle spielt, nämlich für Codes $\mathcal{C}$ größerer Länge n.

Um festzustellen, ob die obengenannten Schlußfolgerungen nicht unbedingt von den Eigenschaften der Akzeptanzkriterien, sondern eventuell nur von den Eigenschaften des $(7, 4, 3)_2$-Codes $\mathcal{C}$ abhängig waren, wurde u. a. auch der $(24, 12, 8)_2$-Golay-Code [MS83, Bos98] simuliert. Wie erwartet, war der durchschnittliche Prozentsatz der Abbrüche für den Hyperkugel–Test und für den GMD–Test (nach der Verwendung des Syndrom–Tests) so niedrig, daß er für die simulierten quadratischen Abstände Δ^2 praktisch gleich Null war. Aus diesem Grund wurden lediglich der Hyperkreiskegel–Test, das Akzeptanzkriterium nach Taipale und Pursley sowie das Akzeptanzkriterium nach Kasami in Abbildung 5.3 dargestellt.

Es ist ersichtlich, daß das Akzeptanzkriterium nach Kasami basierend auf zwei Codewörtern zu besseren Ergebnissen führt als die des besten Akzeptanzkriteriums basierend auf einem einzigen Codewort, nämlich des Akzeptanzkriteriums nach Taipale und Pursley. Dagegen lohnt sich die Implementierung von Akzeptanzkriterien basierend auf mehr als zwei Codewörtern kaum, weil die Komplexität beträchtlich steigt, während sich der durchschnittliche Prozentsatz der Abbrüche lediglich geringfügig ändert [KTK$^+$95, KKTL95a, KKTL95b], wie in Unterabschnitt 3.2.4 besprochen wurde.

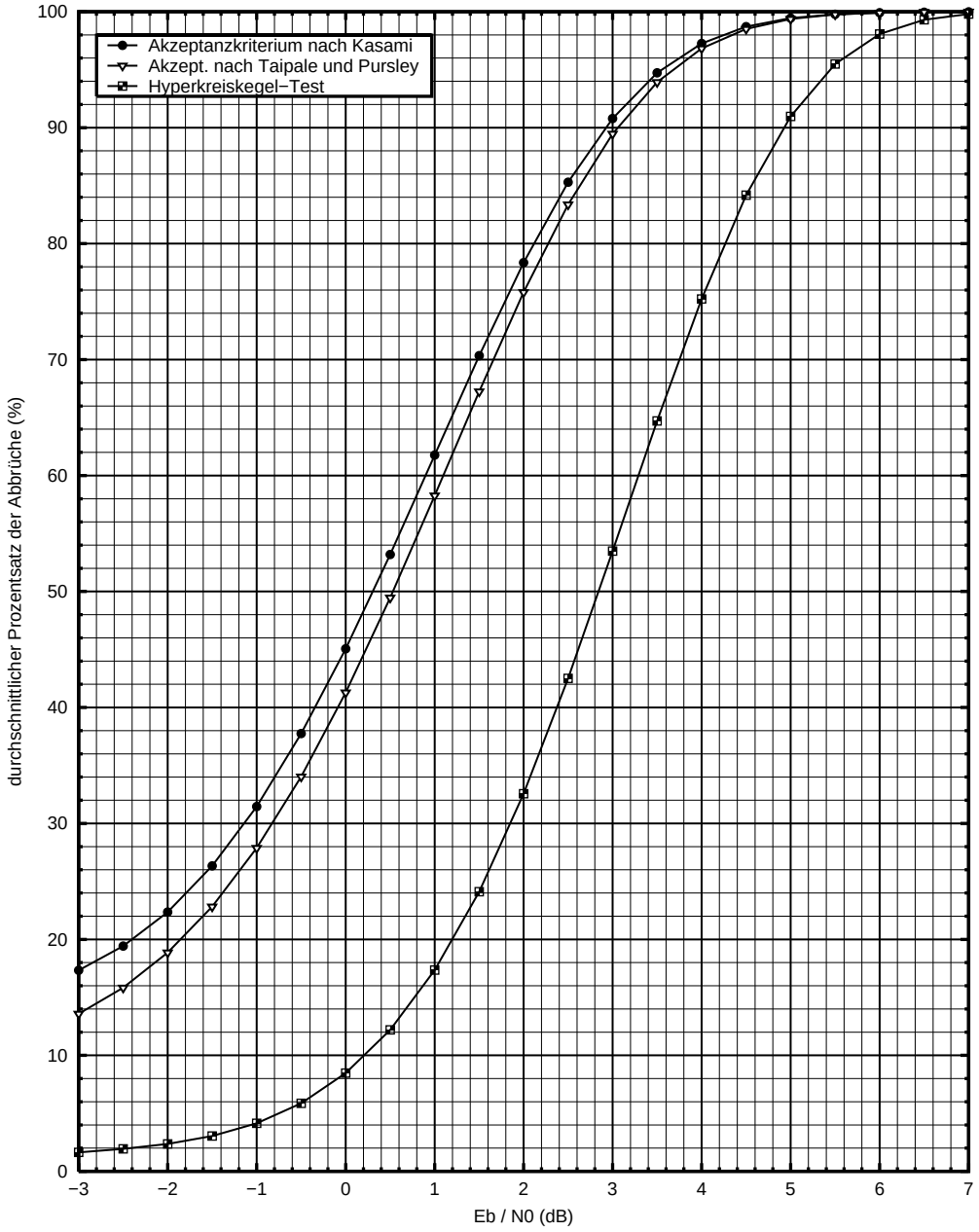

Abbildung 5.3: Durchschnittlicher Prozentsatz der Abbrüche in Abhängigkeit von SNR bei verschiedenen Akzeptanzkriterien für den $(24, 12, 8)_2$-Golay-Code.

5.5 Suboptimale Akzeptanzkriterien bei langen Codes

Offensichtlich werden alle in Kapitel 3 erläuterten Soft–Decision–Maximum–Likelihood–Akzeptanzkriterien strenger, wenn die Länge n des verwendeten Codes $\mathcal{C}$ allmählich wächst. Besonders bei langen Codes, die häufig nur wenige minimalgewichtige Codewörter $\boldsymbol{c} \in \mathcal{C}$ verglichen mit der gesamten Anzahl von Codewörtern des Codes $\mathcal{C}$ besitzen, ist es daher sehr unwahrscheinlich, daß eine geschätzte Sendefolge $\hat{\boldsymbol{x}} \in \mathcal{X}$ existiert, die die Soft–Decision–Maximum–Likelihood–Akzeptanzkriterien erfüllen könnte. In diesen Fällen wird der durchschnittliche Prozentsatz der Abbrüche besonders bei großem quadratischen Abstand Δ^2 bzw. kleinem SNR sehr gering.

Deswegen wurde zusätzlich ein Akzeptanzkriterium für lange Codes untersucht, bei dem bereits dann abgebrochen wird, wenn anstelle von Bedingung $\Xi_{\text{Kasami}(\hat{\boldsymbol{x}}_1,\hat{\boldsymbol{x}}_2)}$ (Gleichung (3.51) des Akzeptanzkriteriums basierend auf zwei Codewörtern) folgende Bedingung erfüllt wird:

$$\Xi_\epsilon: \quad \epsilon \, d_\beta(\hat{\boldsymbol{x}}_2, \boldsymbol{y}) \leq \sum_{i \in \mathcal{Q}} \beta_i, \tag{5.11}$$

wobei $\epsilon \leq 1$, $\epsilon \in \mathbb{R}^+$ ist und die anderen Parameter bereits in Unterabschnitt 3.2.3 erklärt wurden. Selbstverständlich wird keine SDMLD mehr durchgeführt, falls $\epsilon < 1$ ausgewählt wird. Hingegen wurde durch verschiedene Simulationen festgestellt, daß die Auswahl eines $\epsilon < 1$ bei langen Codes $\mathcal{C}$ eine beträchtliche Erhöhung des durchschnittlichen Prozentsatzes der Abbrüche ermöglicht, ohne die Leistungsfähigkeit des Decodierers bezüglich der WER bzw. BER sichtbar zu beeinträchtigen. Überdies war dieses Verhalten nahezu unabhängig von der maximalen Anzahl $\mathcal{I}_{\max}$ der gebildeten Informationsmengen.

Beispielsweise wurde der $(128, 64, 22)_2$-eBCH-Code von Abschnitt 5.3 mit quadratischer Normierung und $\mathcal{I}_{\max} = 50$ für immer kleinere wachsende Werte von ϵ simuliert, bis sich die WER von der WER für $\epsilon = 1$ unterscheidet. Der kleinste Wert von ϵ, für den noch alle Punkte der Kurven identisch geblieben sind, war $\epsilon = 0.2$. Mit anderen Worten, alle Punkte von Abbildung 5.1 wurden mindestens zweimal (mit $\epsilon = 1$ und 0.2) simuliert, um festzustellen, ob tatsächlich die WER (mit elfstelliger Genauigkeit) identisch war. In Abbildung 5.4 ist deutlich zu erkennen, daß der erreichte Gewinn mit der Verwendung von $\epsilon = 0.2$ anstatt von $\epsilon = 1$ innerhalb eines großen Bereiches etwa 1.5 dB beträgt. Deswegen soll in diesem Fall stark betont werden,

daß dieses suboptimale Akzeptanzkriterium ebenfalls eine SDMLD bei einer WER größer als 10^{-5} unter der Genauigkeit der Simulationen durchgeführt. Der Gewinn könnte beträchtlich vergrößert werden, falls eine winzige Erhöhung der WER (in der Praxis nahezu Null) tolerierbar wäre, was in der vorliegenden Arbeit nicht genauer untersucht wurde.

5.6 Simulationsergebnisse eines längeren Codes

Um sowohl die hervorragende Leistungsfähigkeit des Decodieralgorithmus von Unterabschnitt 4.4.3 als auch die außerordentliche Geschwindigkeit der in Abschnitt 2.8 vorgeschlagenen Simulationsmethode unter Beweis zu stellen, wurde ein $(255, 123)_2$-BCH-Code simuliert. Der Grund für die Auswahl dieses Codes ist, daß er eine Rate $R = 0.48$ und beinahe doppelte Länge n besitzt als der $(128, 64, 22)_2$-eBCH-Code von Abschnitt 5.3. Dies ermöglicht, daß die beiden Codes direkt miteinander verglichen werden können.

Der $(255, 123)_2$-BCH-Code hat einen sogenannten geplanten Minimalabstand [MS83] (*Designed Distance*, auch Entwurfsdistanz [Fri95] oder geplante Mindestdistanz [Bos98] genannt) von 39 und daher einen (zur Zeit noch unbekannten) wirklichen Minimalabstand $d_{H\min}$ von mindestens 39. Da die Kenntnis des Minimalabstandes $d_{H\min}$ im Decodieralgorithmus nicht verwendet wird, sondern sie lediglich für die Berechnung der Akzeptanzkriterien von Kapitel 3 notwendig ist, spielt der exakte Wert von $d_{H\min}$ eine untergeordnete Rolle, weil das Akzeptanzkriterium des vorherigen Abschnittes als Abbruchregel in diesem Fall benutzt wurde.

In allen Simulationen dieses Abschnittes wurde die beste der drei Normierungen von Abschnitt 5.3 angewendet, nämlich die quadratische Normierung und ϵ konnte bis auf dem Wert 0.1 verkleinert werden, ohne die Fehlerwahrscheinlichkeit gegenüber der eines Soft–Decision–Maximum–Likelihood–Akzeptanzkriteriums zu verschlechtern. Außerdem wurde die maximale Anzahl $\mathcal{I}_{\max}$ der gebildeten Informationsmengen u. a. auf 100, 500, 5000, 50000 und 500000 begrenzt.

Abbildung 5.5 zeigt die WER in Abhängigkeit vom quadratischen Abstand Δ^2 zusammen mit der unteren Simulationsschranke P_{LB} für die SDMLD von Abschnitt 2.9. Diese untere Schranke wurde aus den Simulationsergebnissen für den Fall $\mathcal{I}_{\max} = 500000$ berechnet. Deswegen liegt sie nicht sehr

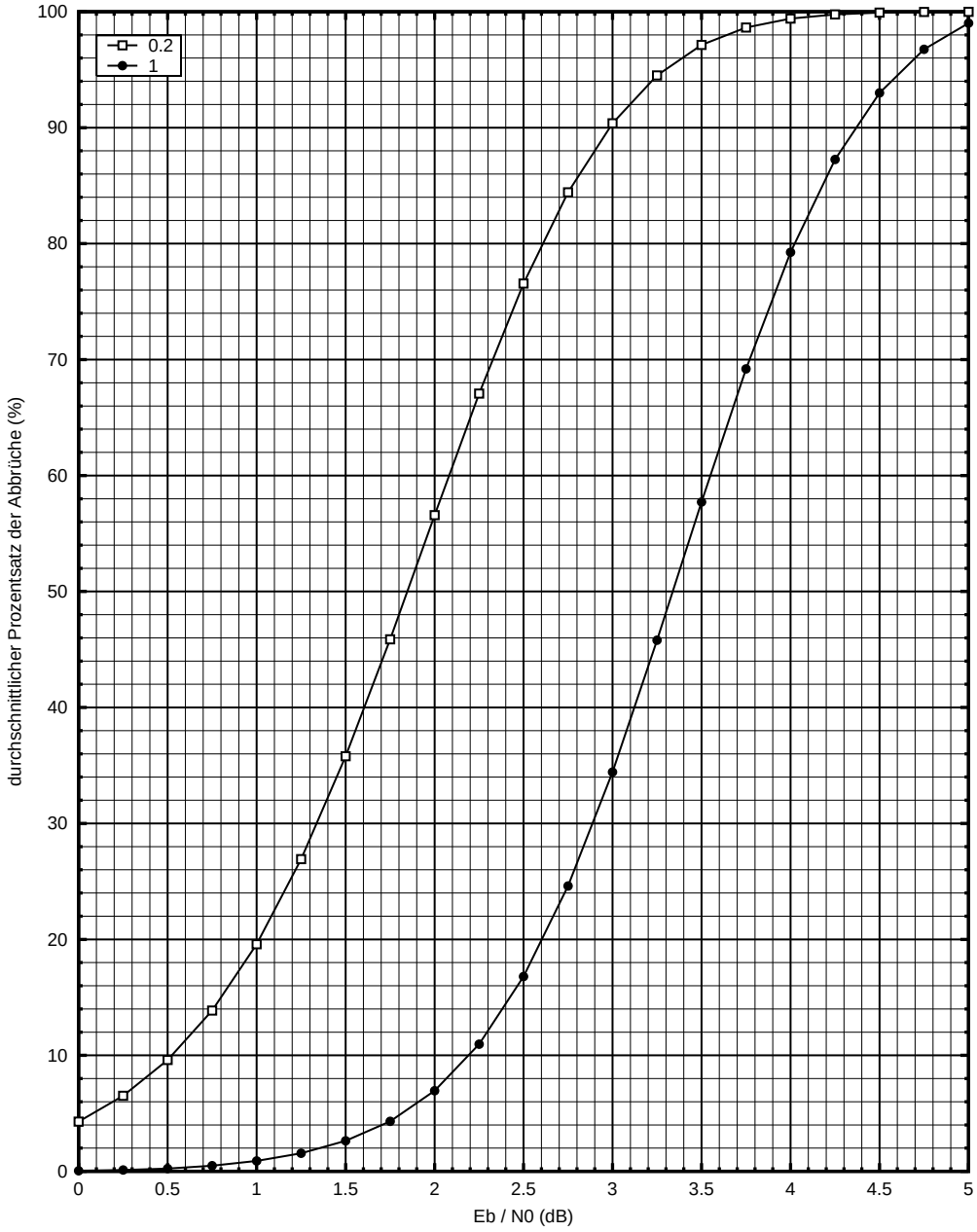

Abbildung 5.4: Durchschnittlicher Prozentsatz der Abbrüche in Abhängigkeit von SNR bei quadratischer Normierung, $\mathcal{I}_{\max} = 50$ und $\epsilon = 1$ und 0.2 für den $(128, 64, 22)_2$-eBCH-Code.

dicht an der tatsächlichen SDMLD.

Basierend auf den gesamten Simulationsergebnissen wurden die entsprechenden Kurven von BER in Abhängigkeit von SNR berechnet und in Abbildung 5.6 dargestellt. Zwecks einer Referenz wurde ebenfalls die Fehlerrate P_u einer uncodierten Übertragung nach Gleichung (5.9) hinzugefügt.

Aus Abbildung 5.5 ist ersichtlich, daß selbst mit $\mathcal{I}_{\max} = 500000$ noch keine SDMLD erreicht wird. Diese Zahl erscheint eigentlich nicht zu groß verglichen mit der ungeheuren Anzahl von mehr als 10^{37} Codewörtern des $(255, 123)_2$-BCH-Codes. Trotzdem beträgt der Gewinn in diesem Fall bei einer BER von 10^{-5} (Abbildung 5.6) im Vergleich mit einer SDMLD eines $(128, 64, 22)_2$-eBCH-Codes (Abbildung 5.2) mindestens 0.55 dB. Möglicherweise ist die Anwendung der in dieser Arbeit vorgeschlagenen Decodiermethode mit $\mathcal{I}_{\max} = 500000$ weniger als 0.2 dB von einer SDMLD für den $(255, 123)_2$-BCH-Code entfernt.

In Abbildung 5.7 wird die WER in Abhängigkeit von SNR veranschaulicht. Wie üblich ist die Fehlerrate P_u einer uncodierten Übertragung angegeben. Zum Zwecke von Vergleichen wurde auch die Wortfehlerrate P_{BMD} einer HDBMD–Decodierung dargestellt, die durch folgende Gleichung bestimmt werden kann:

$$P_{BMD} = 1 - \sum_{i=0}^{t} \binom{n}{i} {p_c}^i (1 - p_c)^{n-i}, \tag{5.12}$$

wobei der Packungsradius t gemäß Gleichung (2.17) und die codierte Fehlerwahrscheinlichkeit p_c folgendermaßen berechnet wird:

$$p_c = \frac{1}{2} \operatorname{erfc}\Big(\sqrt{R}\, 10^{0.05 \frac{E_b}{N_0}[dB]}\Big). \tag{5.13}$$

Aus Abbildung 5.7 folgt, daß eine (Soft-Decision) Decodierung mittels des Algorithmus von Abschnitt 4.4.3 sogar mit nur $\mathcal{I}_{\max} = 100$ bei einer WER von 10^{-5} bereits mindestens 0.75 dB besser ist als die beste (Hard–Decision) algebraische Decodierung für den $(255, 123)_2$-BCH-Code. Offensichtlich wächst dieser Gewinn hinsichtlich einer HDBMD–Decodierung wesentlich, falls eine größere maximale Anzahl der gebildeten Informationsmengen $\mathcal{I}_{\max}$ erlaubt wird.

Zusätzlich wurde die WER P_{COR} basierend auf der *Computational Cutoff–Rate* R_0 [CC81, Hub92, Fri95, Pro95] ebenfalls in Abbildung 5.7 hinzugefügt. Diese Kurve wurde wie folgt berechnet [Fri95]:

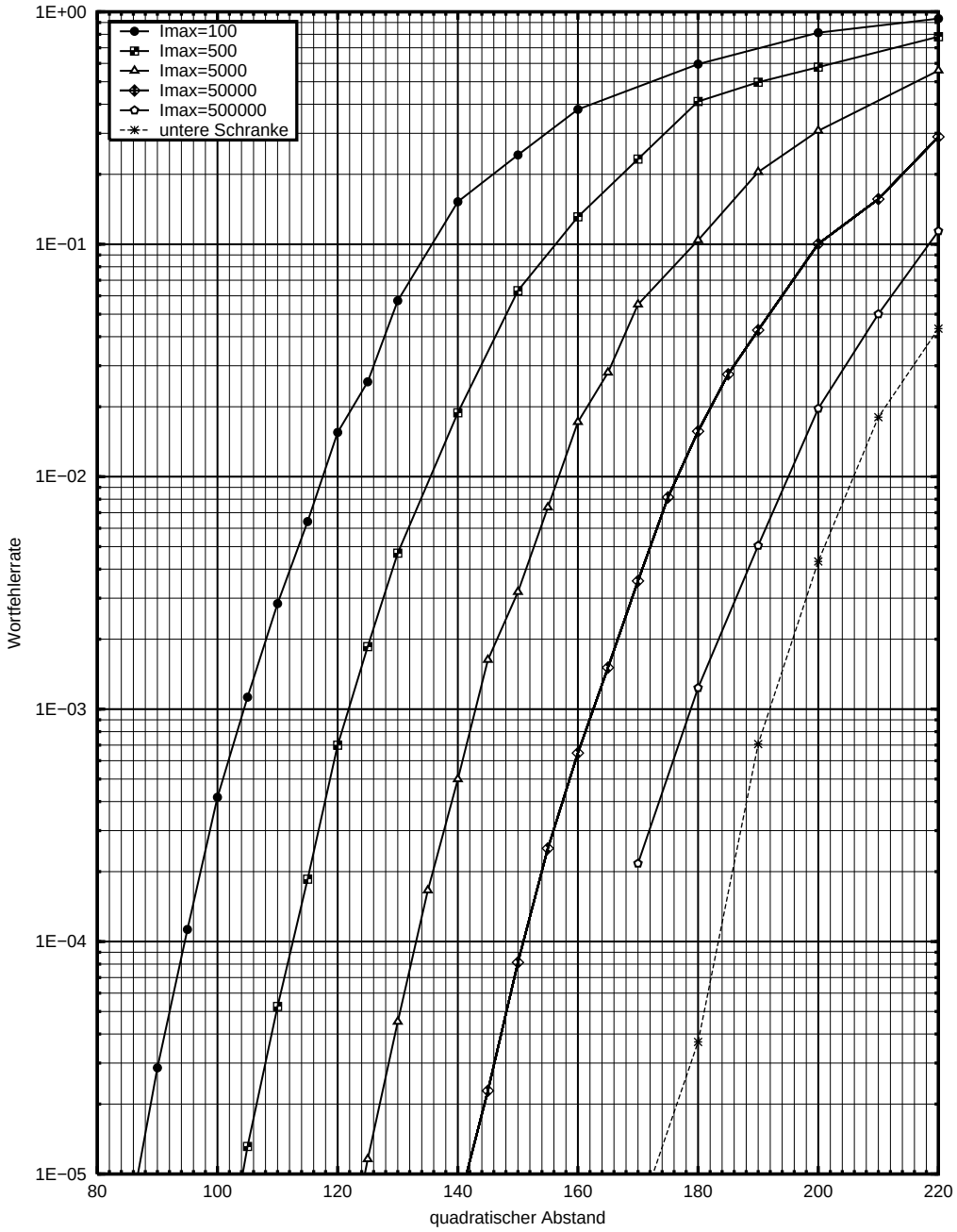

Abbildung 5.5: WER in Abhängigkeit von Δ^2 für den $(255, 123)_2$-BCH-Code.

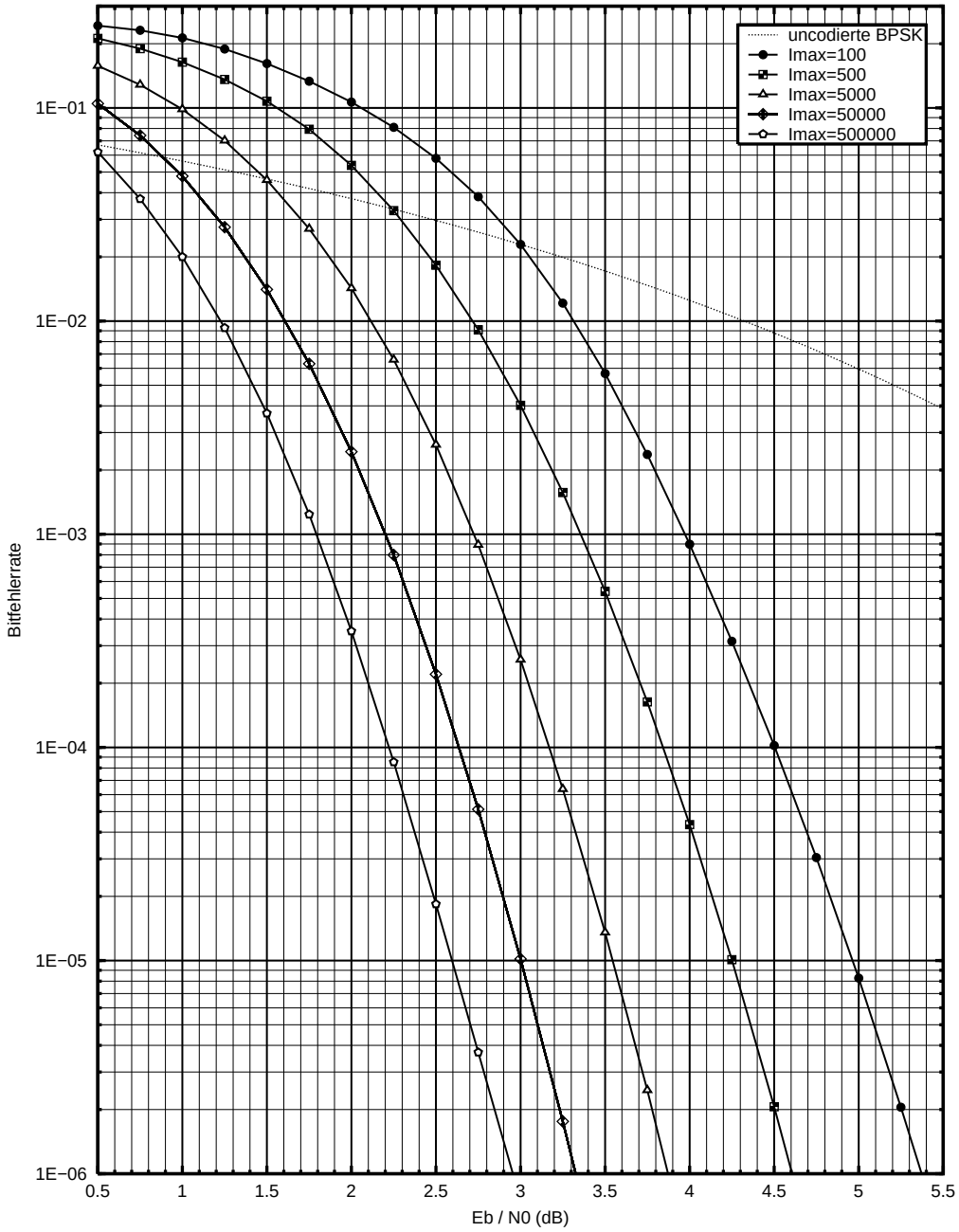

Abbildung 5.6: BER in Abhängigkeit von SNR für den $(255, 123)_2$-BCH-Code.

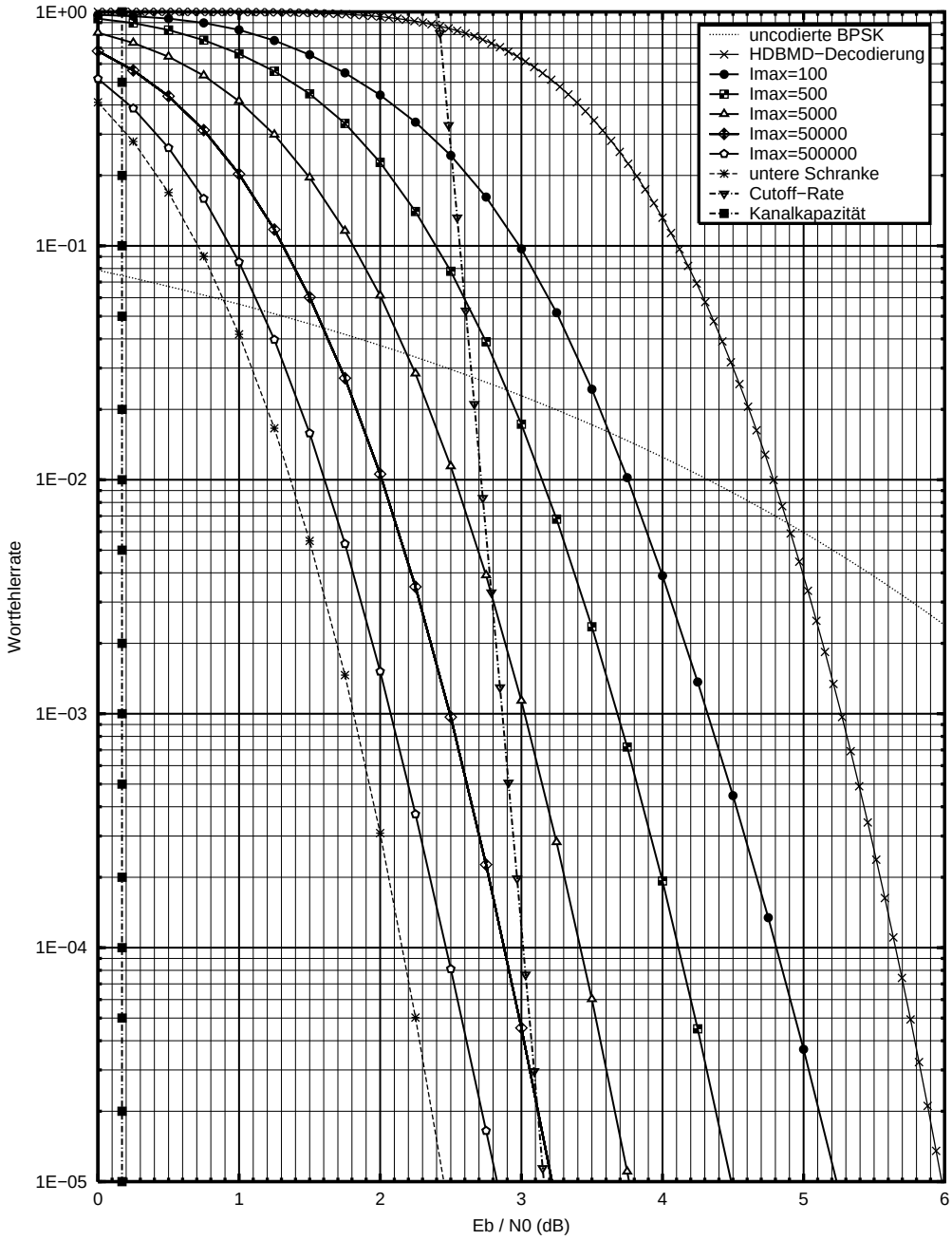

Abbildung 5.7: WER in Abhängigkeit von SNR für den $(255, 123)_2$-BCH-Code.

$$P_{COR} < 2^{-n\,(R_0 - R)}. \tag{5.14}$$

Dabei ist

$$R_0 = 1 - \mathrm{lb}\left(1 + e^{-R\,10^{0.1\frac{E_b}{N_0}[dB]}}\right) \tag{5.15}$$

die *Computational Cutoff–Rate* R_0 (auch R_0–Kriterium [Hub92] oder R_0–Wert [Fri95] genannt) für einen AWGN–Kanal. Dieser Wert ist in der Praxis von großer Bedeutung, weil er mit realisierbaren Decodierverfahren und vernünftigem Aufwand erreichbar ist [Fri95]. Beispielsweise ist in Abbildung 5.7 leicht zu erkennen, daß der Decodieralgorithmus mit $\mathcal{I}_{\max} = 500000$ über die *Cutoff–Rate* hinausgeht.

Die Kapazitätsgrenze bei dieser Coderate, binären Sendesignalen und einem AWGN–Kanal liegt bei $E_b \,/\, N_0 = 0.17$. Sie ist eine theoretische Grenze, von der die praktisch angewendeten Decodierverfahren deutlich entfernt sind [Fri95]. Trotzdem ist die SDMLD eines $(255, 123)_2$-BCH-Codes bei einer WER von 10^{-5} wahrscheinlich nicht weiter als 2.5 dB von dieser theoretischen Grenze entfernt, was vermutlich auch für $\mathcal{I}_{\max} = 1000000$ in diesem Fall erreicht wird.

Abbildung 5.8 zeigt die durchschnittliche Anzahl der gebildeten Informationsmengen in Abhängigkeit von SNR. Offensichtlich gehen alle Kurven für verschiedene $\mathcal{I}_{\max}$ bei großen SNR asymptotisch zusammen, weil dann der Syndrom–Test von Unterabschnitt 3.1.1, der zunächst ausgeführt wird, eine entscheidende Rolle spielt.

Für lange Codes $\mathcal{C}$ kann die gesamte Komplexität des Decodieralgorithmus bei einem gegebenen SNR geschätzt werden als die Komplexität zur Bildung einer Informationsmenge multipliziert mit der durchschnittlichen Anzahl der gebildeten Informationsmengen, die direkt aus Abbildung 5.8 abgelesen werden kann. Dabei entspricht die Komplexität zur Bildung einer Informationsmenge etwa der eines gewöhnlichen gaußschen Eliminationsverfahrens [Bar97, BSMM99, Stö99]. Asymptotisch beträgt diese Komplexität höchstens $k^2\,n$ elementare Operationen [Pet67, FL95].

Gemäß [Bar98] kann die Geschwindigkeit einer Decodierung mithilfe von Informationsmengen für lange Codes ($n > 1000$) beträchtlich erhöht werden, wenn berücksichtigt wird, daß aufeinanderfolgende Informationsmengen eine große Schnittmenge haben, die zur Vereinfachung der Berechnung von $\boldsymbol{G}'(\mathcal{I})$ in Gleichung (2.30) bzw. (4.21) benutzt werden kann.

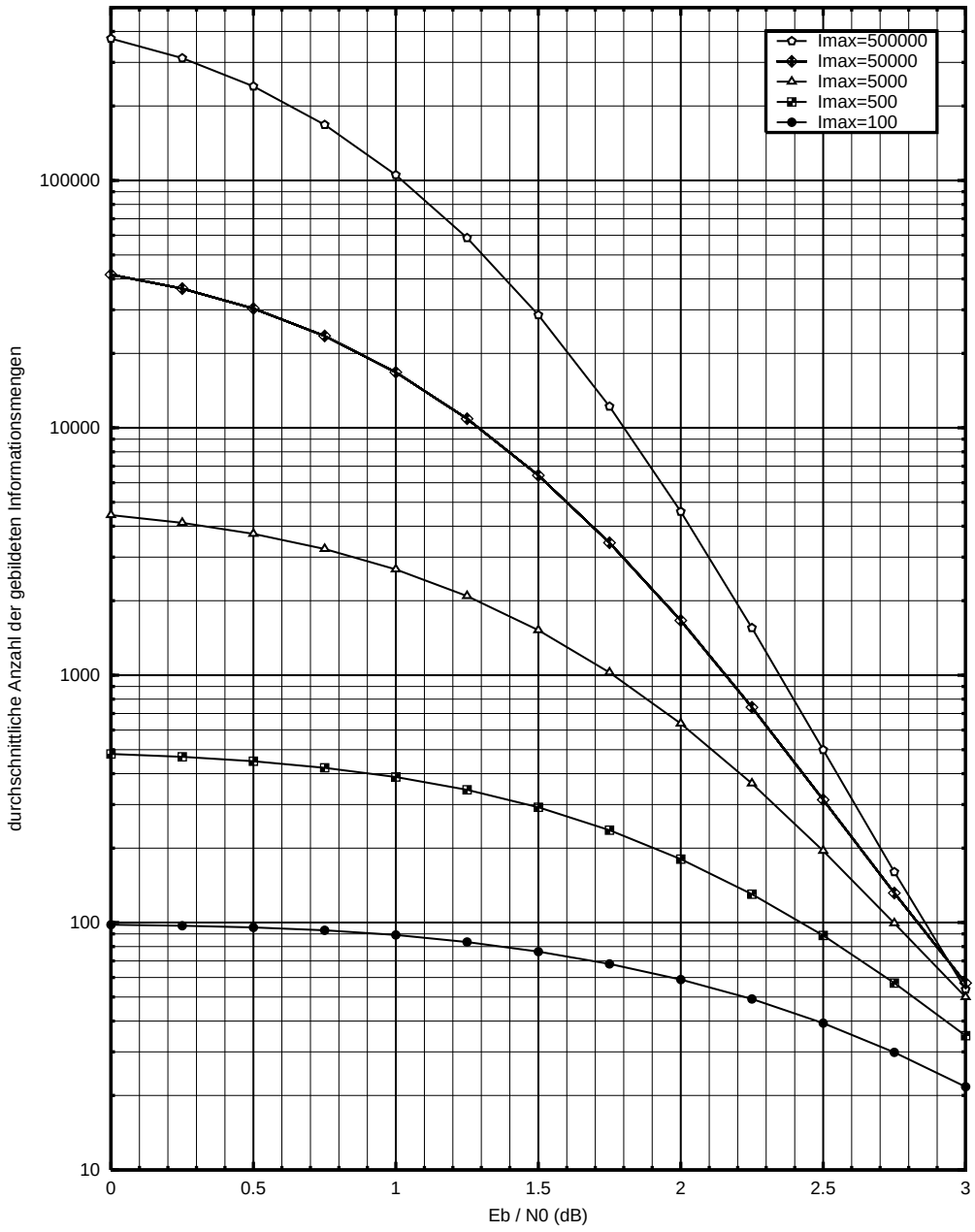

Abbildung 5.8: Durchschnittliche Anzahl der gebildeten Informationsmengen in Abhängigkeit von SNR für den $(255, 123)_2$-BCH-Code.

5.7 Abschließende Bemerkungen

In diesem Kapitel wurden die wesentlichsten Simulationsergebnisse dargestellt. Einige Techniken zur Optimierung von Programmen wurden im ersten Abschnitt diskutiert. Sie waren entscheidend zur beträchtlichen Verminderung der Ausführungszeiten der Simulationen, die in einigen Fällen sogar über 150 Stunden gedauert haben.

Im zweiten Abschnitt wurde auf die Implementierungsaspekte eingegangen. Es wurde erläutert, wie sowohl das Übertragungssystem als auch der Decodieralgorithmus realisiert wurden. Dadurch, daß solange Codewörter gesendet wurden, bis eine bestimmte Anzahl von Decodierfehlern auftratt, wurde gewährleistete, daß alle Simulationsergebnisse sogar bei einer WER kleiner als 10^{-5} große Genauigkeit hatten. Somit wurden Referenzkurven für den $(128, 64, 22)_2$-eBCH-Code bzw. den $(255, 123)_2$-BCH-Code in den nächsten Abschnitten erzeugt, die eventuelle Vergleiche mit der Güte anderer (suboptimalen) Decodieralgorithmen ermöglichen.

In Abschnitt 5.3 wurden drei verschiedene Normierungen vorgestellt, die in Schritt 2.6 des allgemeinen Decodieralgorithmus von Unterabschnitt 4.4.3 verwendet werden können, nämlich eine lineare, eine quadratische und eine kubische Normierung. Es wurde durch Simulationen festgestellt, daß eine quadratische Normierung nach Gleichung (5.4) die beste der drei Normierungen für Codes $\mathcal{C}$ großer Länge n ist.

Danach wurden die Akzeptanzkriterien von Kapitel 3 miteinander verglichen. Hauptsächlich wurden folgende Schlußfolgerungen gezogen: Da der Hyperkreiskegel–Test gemäß Gleichung (3.11) keinerlei Information über der Zuordnung der Komponenten β_i benutzt, ist er eine gute Auswahl, wenn eine kleine Komplexität einer Abbruchregel gewünscht wird. Das Akzeptanzkriterium nach Taipale und Pursley nach Gleichung (3.24) stellte sich als die bestmögliche Bedingung Ξ auf Basis eines einzigen geschätzten Codewortes $\hat{\boldsymbol{c}}$ heraus. Sind zwei geschätzte Codewörter $\hat{\boldsymbol{c}}_\mathbf{1}$ und $\hat{\boldsymbol{c}}_\mathbf{2}$ als Referenz verfügbar, dann erweist sich das Akzeptanzkriterium nach Kasami von Unterabschnitt 3.2.3 als beste Lösung für Codes $\mathcal{C}$ großer Länge n.

Im nächsten Abschnitt wurde ein suboptimales Akzeptanzkriterium eingeführt, das eine beträchtliche Erhöhung des durchschnittlichen Prozentsatzes der Abbrüche ermöglichte, ohne die Leistungsfähigkeit des Decodierers bezüglich der WER bzw. BER sichtbar zu beeinträchtigen. Es wurde durch verschiedene Simulationen beobachtet, daß die Verringerung von ϵ in Gleichung (5.11) die Korrekturfähigkeit kürzerer Codes $\mathcal{C}$ stärker beeinflußt als

die von längeren. Obwohl $\epsilon = 0.2$ für den $(128, 64, 22)_2$-eBCH-Code bzw. $\epsilon = 0.1$ für den $(255, 123)_2$-BCH-Code ausgewählt wurde, wurde noch eine SDMLD bei einer WER größer als 10^{-5} unter der Genauigkeit der Simulationen durchgeführt. Würde eine winzige Erhöhung der WER erlaubt, so könnte der durchschnittliche Prozentsatz der Abbrüche beträchtlich steigen.

Anschließend wurden die Simulationsergebnisse für den $(255, 123)_2$-BCH-Code dargestellt. Einerseits hat die Berechnung der WER einer HDBMD–Decodierung gezeigt, daß der Decodieralgorithmus von Abschnitt 4.4.3 mit einer kleinen Komplexität ($\mathcal{I}_{\max} = 100$) wesentlich leistungsfähiger ist als ein algebraischer Decodierer. Andererseits konnte die Decodiermethode mit einer großen Komplexität ($\mathcal{I}_{\max} = 500000$) sogar die R_0–Werte überschreiten. Aus den Kurven von Abbildung 5.7 wurde vermutet, daß die Decodierung eines $(255, 123)_2$-BCH-Codes mit $\mathcal{I}_{\max} = 1000000$ möglicherweise etwa 2.5 dB weit von der Kanalkapazität entfernt wäre. In der letzten Abbildung wurde die durchschnittliche Anzahl der gebildeten Informationsmengen in Abhängigkeit von SNR in Abbildung 5.8 betrachtet. Aus diesem Abbildung kann die Komplexität für lange Codes $\mathcal{C}$ berechnet werden.

Kapitel 6

Zusammenfassung

Motivation der vorliegenden Arbeit war, einen aufwandsgünstigen Soft–Decision Decodieralgorithmus zu entwickeln, der einen beliebigen linearen langen Blockcode weit über den halben Minimalabstand hinaus decodieren kann. Dazu wurde diese Dissertation folgendermaßen eingeteilt:

- Zunächst wurden hauptsächlich Grundbegriffe und Definitionen in Kapitel 2 vorgestellt, die dem Verständnis der ganzen Arbeit dienten. Anschließend wurden sowohl Zufallszahlengeneratoren als auch eine Methode zur Simulation langer Codes betrachtet.

- In Kapitel 3 wurden unterschiedliche Soft–Decision–Maximum–Likelihood–Akzeptanzkriterien hergeleitet, die es erlauben, daß eine iterative Decodierung möglicherweise früher abgebrochen werden kann. Somit wird die durchschnittliche Komplexität einer Decodierung wesentlich reduziert.

- Weiterhin wurden Decodiermethoden zuerst für Hard–Decision und dann für Soft–Decision Kanäle in Kapitel 4 behandelt. Dabei wurde ein Decodierverfahren mithilfe von Informationsmengen und Akzeptanzkriterien vorgeschlagen, das besonders für die Decodierung langer Blockcodes angewendet werden kann.

- Schließlich wurden Simulationsergebnisse in Kapitel 5 präsentiert. Dabei wurden drei verschiedene Normierungen sowie die Akzeptanzkriterien von Kapitel 3 genauer untersucht und ein suboptimales Akzeptanzkriterium für lange Codes vorgeschlagen. Danach wurden Simulationsergebnisse eines langen Codes dargestellt.

6.1 Ergebnisse und Beiträge der Arbeit

In den Kapiteln wurden viele theoretische Überlegungen beschrieben und Untersuchungen durchgeführt. Daraus folgen die wichtigsten Ergebnisse, Vorschläge und Beiträge der Arbeit:

(a) In Abschnitt 2.7 wurden extrem schnelle Zufallszahlengeneratoren implementiert, die sowohl für die Decodiermethode als auch für die Simulationen notwendig sind. Beispielsweise wird die Menge von mindestens k Stellen in Schritt 3 der Decodiermethode von Unterabschnitt 4.4.3 mit Hilfe eines gleichverteilten Zufallszahlengenerators erzeugt.

(b) Die Simulationsmethode von Abschnitt 2.8 ermöglichte in Kapitel 5 präzise Simulationen langer Blockcodes bei kleiner Fehlerwahrscheinlichkeit mit akzeptablen Ausführungszeiten.

(c) In Unterabschnitt 3.2.3 wurde eine algorithmische Darstellung der Methode zur Berechnung des effizienten Akzeptanzkriteriums nach Kasami basierend auf zwei Codewörtern präsentiert, die gemeinsam mit dem Decodieralgorithmus in Kapitel 5 simuliert wurde.

(d) In Kapitel 4 wurde als einer der wesentlichsten Beiträge dieser Arbeit gezeigt, wie der probabilistische Decodieralgorithmus von Dumer nicht nur im asymptotischen Fall, sondern auch für konkrete Codes verwendet werden kann. Dabei wurde ein Decodieralgorithmus für solche Codes endlicher Länge in den drei Unterabschnitten des Abschnittes 4.4 ausführlich erklärt.

(e) Ein weiterer wesentlicher Beitrag der vorliegenden Arbeit ist, mittels der Simulationen von Abschnitt 5.3 zu beweisen, daß verschiedene Methoden zur Berechnung der Wahrscheinlichkeiten p_i in Gleichung (4.9) bzw. Ungleichung (4.11) zu ganz unterschiedlichen Ergebnissen bezüglich der WER und Komplexität der Decodiermethode führen können. Die Simulationen zeigen, daß eine quadratische Normierung nach Gleichung (5.4) die geeigneteste Normierung für lange Codes ist.

(f) Dank der hervorragenden Leistungsfähigkeit des Decodieralgorithmus von Unterabschnitt 4.4.3 sowie der außerordentlichen Geschwindigkeit der in Abschnitt 2.8 vorgeschlagenen Simulationsmethode wurde eine

Referenzkurve für die SDMLD eines $(128, 64, 22)_2$-eBCH-Codes in Abschnitt 5.3 dargestellt, die erstmals so präzis für kleine BER simuliert wurde.

(g) Darüber hinaus wurde in Abschnitt 5.5 besonders betont, daß die Anwendung eines suboptimalen Akzeptanzkriteriums bei langen Codes eine erhebliche Verminderung der durchschnittlichen Decodierkomplexität ermöglicht, ohne die Leistungsfähigkeit des Decodierers bezüglich der WER bzw. BER sichtbar zu beeinträchtigen.

(h) Abschließend wurden Simulationen mit einem $(255, 123)_2$-BCH-Code in Abschnitt 5.6 durchgeführt. Durch diese Simulationsergebnisse konnte festgestellt werden, daß die Leistungsfähigkeit dieses Codes die eines zufälligen Codes gemäß der *Computational Cutoff–Rate* R_0 übersteigt, falls eine genügend große Komplexität tolerierbar ist. Die erzeugten Referenzkurven für den $(255, 123)_2$-BCH-Code ermöglichen eventuelle Vergleiche mit der Güte anderer (suboptimalen) Decodieralgorithmen.

6.2 Offene Fragen und weiterführende Untersuchungen

Im folgenden werden sowohl interessante Verbesserungen und Erweiterungen des entwickelten Decodierverfahrens als auch weiterführende Forschungsarbeiten vorgeschlagen:

- Wird ein Akzeptanzkriterium basierend auf drei Codewörtern anstatt auf zwei benutzt, werden die Entscheidungsräume um die Codewörter lediglich geringfügig größer, obwohl die Komplexität der Implementierung erheblich steigt. Dies wurde in [KTK$^+$95, KKTL95a, KKTL95b] mit der zweiten Variante des Algorithmus von Chase [Cha72, Bos98] simuliert. Deshalb sollten neue Simulationen mit den Akzeptanzkriterien basierend auf mehreren Codewörtern [KKTL95a, KKTL95b, KNT$^+$95, KTK$^+$95, KTKL96, YK97a, YK97b, KKL98, YKF98, KTKL99] durchgeführt werden, damit sichergestellt ist, ob dies ebenfalls für den Decodieralgorithmus von Unterabschnitt 4.4.3 gilt.

- Eine interessante und völlig neue Forschungsrichtung wäre die Herleitung von Akzeptanzkriterien basierend auf mehreren Codewörtern im

nicht binären Fall. Als Ausgangspunkt für diese Erweiterungen dient der Beitrag von Kaneko [KNIH94], in dem ein Akzeptanzkriterium auch für q-stufige Codes behandelt wurde.

- Falls q-stufige MDS–Codes, wie beispielsweise RS–Codes, mit einer Verallgemeinerung des Decodierverfahrens von Unterabschnitt 4.4.3 verwendet werden, vereinfacht sich Schritt 4 in Unterabschnitt 4.4.3 wesentlich, weil alle beliebig ausgewählten Blöcke von k Stellen in einem MDS–Code immer eine Informationsmenge $\mathcal{I}$ bilden. Somit ist der Algorithmus von Unterabschnitt 4.4.2 nicht mehr nötig und die gesamte Komplexität des Decodieralgorithmus wird sich bestimmt sehr stark verringern.

- Eine Suche nach möglichen besseren Normierungen wird erforderlich. Wie in Abschnitt 5.3 gesehen, könnte eine kleine Verbesserung in diese Richtung die Leistungsfähigkeit des Decodieralgorithmus beträchtlich erhöhen.

- Wie am Ende von Abschnitt 5.5 besprochen, könnte der Gewinn bei der Verwendung eines suboptimalen Akzeptanzkriteriums erheblich vergrößert werden, falls eine Erhöhung der WER tolerierbar wäre. Dies sollte zukünftig genauer untersucht werden.

- Schließlich steigt die Decodierkomplexität bei Erhöhung der Rate R des Codes. Bei Anwendung hochratiger Codes ($1/2 \leq R < 1$) sollte der duale Decodieralgorithmus offensichtlich benutzt werden, indem anstelle der Generatormatrix $\boldsymbol{G}$ die Prüfmatrix $\boldsymbol{H}$ und anstelle von Informationsmengen $\mathcal{I}$ Prüfmengen $\mathcal{P}$ verwendet werden.

Anhang A

Herleitungen

A.1 Herleitung von Gleichung (2.44)

Da

$$\operatorname{sign}(x_i) \neq \operatorname{sign}(x_i') \Longrightarrow \begin{cases} \operatorname{sign}(x_i) \neq \operatorname{sign}(y_i) \text{ und } \operatorname{sign}(x_i') = \operatorname{sign}(y_i) \\ \qquad\qquad\qquad\qquad \text{oder} \\ \operatorname{sign}(x_i) = \operatorname{sign}(y_i) \text{ und } \operatorname{sign}(x_i') \neq \operatorname{sign}(y_i) \end{cases} \tag{A.1}$$

für $0 \leq i < n$ gültig ist, kann der Hammingabstand $d_H(\boldsymbol{c}, \boldsymbol{c'})$ bzw. die Mächtigkeit (Kardinalzahl) der Indexmenge $\mathcal{D}_1(\boldsymbol{x}, \boldsymbol{x'})$ unter Ausnutzung der Gleichungen (2.41) und (2.42) wie folgt umgeformt werden:

$$\begin{aligned} d_H(\boldsymbol{c}, \boldsymbol{c'}) &= |\mathcal{D}_1(\boldsymbol{x}, \boldsymbol{x'})| \\ &= |\{i \,|\, \operatorname{sign}(x_i) \neq \operatorname{sign}(x_i'),\ 0 \leq i < n\}| \\ &= |(\{i \,|\, \operatorname{sign}(x_i) \neq \operatorname{sign}(y_i),\ 0 \leq i < n\} \cap \\ &\quad \{i \,|\, \operatorname{sign}(x_i') = \operatorname{sign}(y_i),\ 0 \leq i < n\}) \cup \\ &\quad (\{i \,|\, \operatorname{sign}(x_i) = \operatorname{sign}(y_i),\ 0 \leq i < n\} \cap \\ &\quad \{i \,|\, \operatorname{sign}(x_i') \neq \operatorname{sign}(y_i),\ 0 \leq i < n\})| \\ &= |\{i \,|\, \operatorname{sign}(x_i) \neq \operatorname{sign}(y_i),\ 0 \leq i < n\} \cap \\ &\quad \{i \,|\, \operatorname{sign}(x_i') = \operatorname{sign}(y_i),\ 0 \leq i < n\}| + \\ &\quad |\{i \,|\, \operatorname{sign}(x_i) = \operatorname{sign}(y_i),\ 0 \leq i < n\} \cap \\ &\quad \{i \,|\, \operatorname{sign}(x_i') \neq \operatorname{sign}(y_i),\ 0 \leq i < n\}| \\ &= |\mathcal{D}_1(\boldsymbol{x}, \boldsymbol{y}) \cap \mathcal{D}_0(\boldsymbol{x'}, \boldsymbol{y})| + |\mathcal{D}_0(\boldsymbol{x}, \boldsymbol{y}) \cap \mathcal{D}_1(\boldsymbol{x'}, \boldsymbol{y})|. \end{aligned} \tag{2.44}$$

A.2 Herleitung von Gleichung (2.55)

Aus Gleichung (2.54) wird Gleichung (2.55) hergeleitet unter Verwendung der Gleichungen (2.39a), (2.41), (2.42) und (2.49):

$$
\begin{aligned}
\langle \boldsymbol{x}, \boldsymbol{y} \rangle &= \sum_i x_i \, y_i && (2.54)\\
&= \sum_i x_i \, z_i \, \beta_i \\
&= \sum_{i \in \mathcal{D}_0(\boldsymbol{x},\boldsymbol{z})} \beta_i - \sum_{i \in \mathcal{D}_1(\boldsymbol{x},\boldsymbol{z})} \beta_i \\
&= \left(\sum_i \beta_i - \sum_{i \in \mathcal{D}_1(\boldsymbol{x},\boldsymbol{z})} \beta_i \right) - \sum_{i \in \mathcal{D}_1(\boldsymbol{x},\boldsymbol{z})} \beta_i \\
&= \sum_i \beta_i - 2 \sum_{i \in \mathcal{D}_1(\boldsymbol{x},\boldsymbol{z})} \beta_i \\
&= \sum_i \beta_i - 2 \, d_\beta(\boldsymbol{x}, \boldsymbol{z}). && (2.55)
\end{aligned}
$$

A.3 Herleitung von Gleichung (2.69)

Unter Berücksichtigung der Gleichungen (2.68), (2.36) und (2.54) wird bewiesen, daß Gleichung (2.67) äquivalent zu Gleichung (2.69) ist:

$$
\begin{aligned}
\boldsymbol{x}_{ML} &= \arg\max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \ln \Pr(\boldsymbol{r} \,|\, \hat{\boldsymbol{x}}) - \ln \Pr(\boldsymbol{r} \,|\, -\hat{\boldsymbol{x}}) \right\} && (2.67)\\
&= \arg\max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \ln \prod_i \Pr(r_i \,|\, \hat{x}_i) - \ln \prod_i \Pr(r_i \,|\, -\hat{x}_i) \right\} \\
&= \arg\max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \sum_i \ln \Pr(r_i \,|\, \hat{x}_i) - \sum_i \ln \Pr(r_i \,|\, -\hat{x}_i) \right\} \\
&= \arg\max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \sum_i \ln \frac{\Pr(r_i \,|\, \hat{x}_i)}{\Pr(r_i \,|\, -\hat{x}_i)} \right\} \\
&= \arg\max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \sum_i \ln \left(\frac{\Pr(r_i \,|\, 1)}{\Pr(r_i \,|\, -1)} \right)^{\hat{x}_i} \right\}
\end{aligned}
$$

$$
\begin{aligned}
&= \arg\max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \sum_i \hat{x}_i \, \ln \frac{\Pr(r_i \,|\, 1)}{\Pr(r_i \,|\, -1)} \right\} \\
&= \arg\max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \sum_i \hat{x}_i \, \frac{y_i}{\kappa} \right\} \\
&= \arg\max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \left\{ \frac{1}{\kappa} \sum_i \hat{x}_i \, y_i \right\} \\
&= \arg\max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \sum_i \hat{x}_i \, y_i \\
&= \arg\max_{\hat{\boldsymbol{x}} \in \mathcal{X}} \langle \hat{\boldsymbol{x}}, \boldsymbol{y} \rangle. \qquad (2.69)
\end{aligned}
$$

A.4 Herleitung von Ungleichung (2.71)

Gleichung (2.70) kann unter Anwendung der Gleichungen (2.54) und (2.57) folgendermaßen umformuliert werden:

$$
\begin{aligned}
\langle \hat{\boldsymbol{x}}, \boldsymbol{y} \rangle &\geq \langle \hat{\boldsymbol{x}}', \boldsymbol{y} \rangle, && \forall\ \hat{\boldsymbol{x}}' \in \mathcal{X} \qquad (2.70) \\
\sum_i \hat{x}_i \, y_i &\geq \sum_i \hat{x}'_i \, y_i, && \forall\ \hat{\boldsymbol{x}}' \in \mathcal{X} \\
\sum_i (-2 \, \hat{x}_i \, y_i) &\leq \sum_i (-2 \, \hat{x}'_i \, y_i), && \forall\ \hat{\boldsymbol{x}}' \in \mathcal{X} \\
\sum_i \left(1 - 2 \, \hat{x}_i \, y_i + {y_i}^2\right) &\leq \sum_i \left(1 - 2 \, \hat{x}'_i \, y_i + {y_i}^2\right), && \forall\ \hat{\boldsymbol{x}}' \in \mathcal{X} \\
\sum_i \left({\hat{x}_i}^2 - 2 \, \hat{x}_i \, y_i + {y_i}^2\right) &\leq \sum_i \left({\hat{x}'_i}^2 - 2 \, \hat{x}'_i \, y_i + {y_i}^2\right), && \forall\ \hat{\boldsymbol{x}}' \in \mathcal{X} \\
\sum_i (\hat{x}_i - y_i)^2 &\leq \sum_i (\hat{x}'_i - y_i)^2, && \forall\ \hat{\boldsymbol{x}}' \in \mathcal{X} \\
\sqrt{\sum_i (\hat{x}_i - y_i)^2} &\leq \sqrt{\sum_i (\hat{x}'_i - y_i)^2}, && \forall\ \hat{\boldsymbol{x}}' \in \mathcal{X} \\
d_E(\hat{\boldsymbol{x}}, \boldsymbol{y}) &\leq d_E(\hat{\boldsymbol{x}}', \boldsymbol{y}), && \forall\ \hat{\boldsymbol{x}}' \in \mathcal{X}. \qquad (2.71)
\end{aligned}
$$

A.5 Herleitung von Bedingung (3.5)

Unter Ausnutzung der Gleichungen (2.54), (2.52b) und (2.61) und von Ungleichung (3.2) kann Bedingung (3.4) in Bedingung (3.5) umgewandelt werden:

$$\Xi_{\text{Hyperkugel}}: \qquad \langle \boldsymbol{y}^0 - \hat{\boldsymbol{x}}, \boldsymbol{y}^0 - \hat{\boldsymbol{x}} \rangle \leq {\rho_S}^2 \tag{3.4}$$

$$\Xi_{\text{Hyperkugel}}: \qquad \sum_i \left({y^0}_i - \hat{x}_i \right)^2 \leq \left(\frac{d_{E\min}}{2} \right)^2$$

$$\Xi_{\text{Hyperkugel}}: \qquad \sum_i \left(\frac{y_i}{\|\boldsymbol{y}\|} - \hat{x}_i \right)^2 \leq \left(\frac{2\sqrt{d_{H\min}}}{2} \right)^2$$

$$\Xi_{\text{Hyperkugel}}: \qquad \frac{1}{\|\boldsymbol{y}\|^2} \sum_i (y_i - \|\boldsymbol{y}\| \, \hat{x}_i)^2 \leq \left(\frac{2\sqrt{d_{H\min}}}{2} \right)^2$$

$$\Xi_{\text{Hyperkugel}}: \qquad \sum_i (y_i - \|\boldsymbol{y}\| \, \hat{x}_i)^2 \leq \|\boldsymbol{y}\|^2 \, d_{H\min}. \tag{3.5}$$

A.6 Herleitung von Gleichung (3.7)

Gleichung (3.6) wird in Gleichung (3.7) unter Verwendung der Gleichungen (2.54) und (2.50) wie folgt vereinfacht:

$$\begin{aligned} \cos\varphi_{\max} &= \frac{\langle \boldsymbol{x}, \boldsymbol{y'} \rangle}{\|\boldsymbol{x}\| \, \|\boldsymbol{y'}\|} \\ &= \frac{\sum_{i=0}^{n-1} x_i \, y'_i}{\sqrt{\sum_{i=0}^{n-1} {x_i}^2} \sqrt{\sum_{i=0}^{n-1} {y'_i}^2}}. \end{aligned} \tag{3.6}$$

Da genau $d_{H\min}$ Komponenten des Vektors $\boldsymbol{y'}$ Null sind:

$$\cos\varphi_{\max} = \frac{\sum_{i=0}^{n-d_{H\min}-1} 1}{\sqrt{\sum_{i=0}^{n-1} 1} \sqrt{\sum_{i=0}^{n-d_{H\min}-1} 1}}$$

$$
\begin{aligned}
&= \frac{n - d_{H\min}}{\sqrt{n}\,\sqrt{n - d_{H\min}}} \\
&= \sqrt{\frac{(n - d_{H\min})^2}{n\,(n - d_{H\min})}} \\
&= \sqrt{\frac{n - d_{H\min}}{n}} \\
&= \sqrt{1 - \frac{d_{H\min}}{n}}. \qquad (3.7)
\end{aligned}
$$

A.7 Herleitung von Bedingung (3.11)

Zu Gleichung (3.11) kann Gleichung (3.10) umgeformt werden unter Berücksichtigung der Gleichungen (2.54) und (2.50):

$$
\begin{aligned}
&\Xi_{\text{Hyperkreiskegel}}: \quad \frac{\langle \hat{\boldsymbol{x}}, \boldsymbol{y} \rangle}{\|\hat{\boldsymbol{x}}\|\,\|\boldsymbol{y}\|} \geq \sqrt{1 - \frac{d_{H\min}}{n}} \qquad (3.10) \\
&\Xi_{\text{Hyperkreiskegel}}: \quad \langle \hat{\boldsymbol{x}}, \boldsymbol{y} \rangle \geq \|\boldsymbol{y}\|\,\|\hat{\boldsymbol{x}}\| \sqrt{\frac{n - d_{H\min}}{n}} \\
&\Xi_{\text{Hyperkreiskegel}}: \quad \sum_i \hat{x}_i\, y_i \geq \|\boldsymbol{y}\| \sqrt{\sum_{i=0}^{n-1} \hat{x}_i^2}\, \frac{\sqrt{n - d_{H\min}}}{\sqrt{n}} \\
&\Xi_{\text{Hyperkreiskegel}}: \quad \sum_i \hat{x}_i\, y_i \geq \|\boldsymbol{y}\| \sqrt{n}\, \frac{\sqrt{n - d_{H\min}}}{\sqrt{n}} \\
&\Xi_{\text{Hyperkreiskegel}}: \quad \sum_i \hat{x}_i\, y_i \geq \|\boldsymbol{y}\| \sqrt{n - d_{H\min}}. \qquad (3.11)
\end{aligned}
$$

A.8 Herleitung von Bedingung (3.24)

Bedingung (3.24) wird aus Bedingung (3.23) hergeleitet unter Anwendung der Gleichungen (2.62a), (2.39a), (3.22) und (2.49):

$$
\begin{aligned}
&\Xi_{\text{TP}}: \quad {d_E}^2(\hat{\boldsymbol{x}}, \boldsymbol{y}) \leq {d_E}^2(\boldsymbol{z}', \boldsymbol{y}) \qquad (3.23) \\
&\Xi_{\text{TP}}: \quad \sum_i (\hat{x}_i - y_i)^2 \leq \sum_i (z'_i - y_i)^2
\end{aligned}
$$

$$\Xi_{\mathrm{TP}}: \quad \sum_i \left(\hat{x}_i{}^2 - 2\,\hat{x}_i\,y_i + y_i{}^2\right) \leq \sum_i \left(z'_i{}^2 - 2\,z'_i\,y_i + y_i{}^2\right)$$

$$\Xi_{\mathrm{TP}}: \quad \sum_i 1 - 2\sum_i \hat{x}_i\,y_i + \sum_i y_i{}^2 \leq \sum_i 1 - 2\sum_i z'_i\,y_i + \sum_i y_i{}^2$$

$$\Xi_{\mathrm{TP}}: \quad -\sum_i \hat{x}_i\,z_i\,\beta_i \leq -\sum_i z'_i\,z_i\,\beta_i$$

$$\Xi_{\mathrm{TP}}: \quad -\sum_{\substack{i\in\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y})\cup\\ \mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y})^{(\delta)}}} \hat{x}_i\,z_i\,\beta_i - \sum_{\substack{i\notin\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y})\cup\\ \mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y})^{(\delta)}}} \hat{x}_i\,z_i\,\beta_i \leq -\sum_{\substack{i\in\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y})\cup\\ \mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y})^{(\delta)}}} z'_i\,z_i\,\beta_i - \sum_{\substack{i\notin\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y})\cup\\ \mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y})^{(\delta)}}} z'_i\,z_i\,\beta_i$$

$$\Xi_{\mathrm{TP}}: \quad -\sum_{\substack{i\in\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y})\cup\\ \mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y})^{(\delta)}}} \hat{x}_i\,z_i\,\beta_i - \sum_{\substack{i\notin\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y})\cup\\ \mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y})^{(\delta)}}} \beta_i \leq -\sum_{\substack{i\in\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y})\cup\\ \mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y})^{(\delta)}}} z'_i\,z_i\,\beta_i - \sum_{\substack{i\notin\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y})\cup\\ \mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y})^{(\delta)}}} \beta_i$$

$$\Xi_{\mathrm{TP}}: \quad -\sum_{i\in\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y})} \hat{x}_i\,z_i\,\beta_i - \sum_{i\in\mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y})^{(\delta)}} \hat{x}_i\,z_i\,\beta_i \leq -\sum_{i\in\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y})} z'_i\,z_i\,\beta_i - \sum_{i\in\mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y})^{(\delta)}} z'_i\,z_i\,\beta_i$$

$$\Xi_{\mathrm{TP}}: \quad \sum_{i\in\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y})} \beta_i - \sum_{i\in\mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y})^{(\delta)}} \beta_i \leq -\sum_{i\in\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y})} \beta_i + \sum_{i\in\mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y})^{(\delta)}} \beta_i$$

$$\Xi_{\mathrm{TP}}: \quad 2\sum_{i\in\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y})} \beta_i \leq 2\sum_{i\in\mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y})^{(\delta)}} \beta_i$$

$$\Xi_{\mathrm{TP}}: \quad d_\beta(\hat{\boldsymbol{x}},\boldsymbol{y}) \leq \sum_{i\in\mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y})^{(\delta)}} \beta_i. \tag{3.24}$$

A.9 Herleitung von Gleichung (3.30a)

Die linke Seite von Gleichung (3.30a) kann folgendermaßen geschrieben werden:

$$\begin{aligned}\sum_{\boldsymbol{\alpha}\in\mathbb{B}^h\,|\,\alpha_i=0} |\mathcal{D}_{\boldsymbol{\alpha}}| &= \sum_{\alpha_1\cdots\alpha_{i-1}0\,\alpha_{i+1}\cdots\alpha_{h-1}\alpha_h} |\mathcal{D}_{\boldsymbol{\alpha}}| \\ &= |\mathcal{D}_{0\cdots000\cdots00}| + |\mathcal{D}_{0\cdots000\cdots01}| + \cdots \\ &\quad + |\mathcal{D}_{1\cdots101\cdots10}| + |\mathcal{D}_{1\cdots101\cdots11}|, \quad 1\leq i\leq h.\end{aligned}$$

Wird das Bit α_i an der i-ten Stelle eines binären Musters $\boldsymbol{\alpha}\in\mathbb{B}^h$ mit $\mathrm{bit}(i,\boldsymbol{\alpha})$ bezeichnet, folgt aus Gleichung (3.28):

$$\sum_{\boldsymbol{\alpha}\in\mathbb{B}^h\,|\,\alpha_i=0} |\mathcal{D}_{\boldsymbol{\alpha}}| = \left|\bigcap_{j=1}^{h} \mathcal{D}_{\mathrm{bit}(j,0\cdots000\cdots00)}(\hat{\boldsymbol{x}}_j,\boldsymbol{y})\right| + \left|\bigcap_{j=1}^{h} \mathcal{D}_{\mathrm{bit}(j,0\cdots000\cdots01)}(\hat{\boldsymbol{x}}_j,\boldsymbol{y})\right| + \cdots$$

$$
\begin{aligned}
&+ \left| \bigcap_{j=1}^{h} \mathcal{D}_{\mathrm{bit}(j,1\cdots101\cdots10)}(\hat{\boldsymbol{x}}_{\boldsymbol{j}},\boldsymbol{y}) \right| + \left| \bigcap_{j=1}^{h} \mathcal{D}_{\mathrm{bit}(j,1\cdots101\cdots11)}(\hat{\boldsymbol{x}}_{\boldsymbol{j}},\boldsymbol{y}) \right| \\
&= |\mathcal{D}_0(\hat{\boldsymbol{x}}_{\mathbf{1}},\boldsymbol{y}) \cap \cdots \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_{\boldsymbol{i-1}},\boldsymbol{y}) \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_{i},\boldsymbol{y}) \cap \\
&\quad \mathcal{D}_0(\hat{\boldsymbol{x}}_{\boldsymbol{i+1}},\boldsymbol{y}) \cap \cdots \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_{\boldsymbol{h-1}},\boldsymbol{y}) \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_{h},\boldsymbol{y})| + \\
&\quad |\mathcal{D}_0(\hat{\boldsymbol{x}}_{\mathbf{1}},\boldsymbol{y}) \cap \cdots \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_{\boldsymbol{i-1}},\boldsymbol{y}) \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_{i},\boldsymbol{y}) \cap \\
&\quad \mathcal{D}_0(\hat{\boldsymbol{x}}_{\boldsymbol{i+1}},\boldsymbol{y}) \cap \cdots \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_{\boldsymbol{h-1}},\boldsymbol{y}) \cap \mathcal{D}_1(\hat{\boldsymbol{x}}_{h},\boldsymbol{y})| + \cdots \\
&\quad + |\mathcal{D}_1(\hat{\boldsymbol{x}}_{\mathbf{1}},\boldsymbol{y}) \cap \cdots \cap \mathcal{D}_1(\hat{\boldsymbol{x}}_{\boldsymbol{i-1}},\boldsymbol{y}) \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_{i},\boldsymbol{y}) \cap \\
&\qquad \mathcal{D}_1(\hat{\boldsymbol{x}}_{\boldsymbol{i+1}},\boldsymbol{y}) \cap \cdots \cap \mathcal{D}_1(\hat{\boldsymbol{x}}_{\boldsymbol{h-1}},\boldsymbol{y}) \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_{h},\boldsymbol{y})| + \\
&\qquad |\mathcal{D}_1(\hat{\boldsymbol{x}}_{\mathbf{1}},\boldsymbol{y}) \cap \cdots \cap \mathcal{D}_1(\hat{\boldsymbol{x}}_{\boldsymbol{i-1}},\boldsymbol{y}) \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_{i},\boldsymbol{y}) \cap \\
&\qquad \mathcal{D}_1(\hat{\boldsymbol{x}}_{\boldsymbol{i+1}},\boldsymbol{y}) \cap \cdots \cap \mathcal{D}_1(\hat{\boldsymbol{x}}_{\boldsymbol{h-1}},\boldsymbol{y}) \cap \mathcal{D}_1(\hat{\boldsymbol{x}}_{h},\boldsymbol{y})|, \quad 1 \leq i \leq h.
\end{aligned}
$$

Wenn Gleichung (2.46) mit $\mathcal{J}'(\boldsymbol{x},\boldsymbol{y}) = \mathcal{D}_0(\hat{\boldsymbol{x}}_h,\boldsymbol{y})$ verwendet wird, vereinfacht sich die rechte Seite in:

$$
\begin{aligned}
\sum_{\alpha\in\mathbb{B}^h \,|\, \alpha_i=0} |\mathcal{D}_\alpha| = {} & |\mathcal{D}_0(\hat{\boldsymbol{x}}_{\mathbf{1}},\boldsymbol{y}) \cap \cdots \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_{\boldsymbol{i-1}},\boldsymbol{y}) \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_{i},\boldsymbol{y}) \cap \\
& \mathcal{D}_0(\hat{\boldsymbol{x}}_{\boldsymbol{i+1}},\boldsymbol{y}) \cap \cdots \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_{\boldsymbol{h-1}},\boldsymbol{y})| + \cdots \\
& + |\mathcal{D}_1(\hat{\boldsymbol{x}}_{\mathbf{1}},\boldsymbol{y}) \cap \cdots \cap \mathcal{D}_1(\hat{\boldsymbol{x}}_{\boldsymbol{i-1}},\boldsymbol{y}) \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_{i},\boldsymbol{y}) \cap \\
& \quad \mathcal{D}_1(\hat{\boldsymbol{x}}_{\boldsymbol{i+1}},\boldsymbol{y}) \cap \cdots \cap \mathcal{D}_1(\hat{\boldsymbol{x}}_{\boldsymbol{h-1}},\boldsymbol{y})|, \quad 1 \leq i \leq h.
\end{aligned}
$$

Wird dieselbe Gleichung rekursiv für alle Terme außer $\mathcal{D}_0(\hat{\boldsymbol{x}}_i, \boldsymbol{y})$ benutzt, ergibt sich schließlich Gleichung (3.30a):

$$
\sum_{\alpha\in\mathbb{B}^h \,|\, \alpha_i=0} |\mathcal{D}_\alpha| = |\mathcal{D}_0(\hat{\boldsymbol{x}}_i, \boldsymbol{y})|, \quad 1 \leq i \leq h. \tag{3.30a}
$$

A.10 Herleitung von Gleichung (3.33)

Zunächst kann Indexmenge $|\mathcal{D}_1(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}_i)|$ gemäß Gleichung (2.44) wie folgt geschrieben werden:

$$
\begin{aligned}
|\mathcal{D}_1(\hat{\boldsymbol{x}},\hat{\boldsymbol{x}}_i)| &= |\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y}) \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_i,\boldsymbol{y})| + |\mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y}) \cap \mathcal{D}_1(\hat{\boldsymbol{x}}_i,\boldsymbol{y})| \\
&= |\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y}) \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_i,\boldsymbol{y})| + (-|\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y}) \cap \mathcal{D}_1(\hat{\boldsymbol{x}}_i,\boldsymbol{y})| + \\
&\quad |\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y}) \cap \mathcal{D}_1(\hat{\boldsymbol{x}}_i,\boldsymbol{y})|) + |\mathcal{D}_0(\hat{\boldsymbol{x}},\boldsymbol{y}) \cap \mathcal{D}_1(\hat{\boldsymbol{x}}_i,\boldsymbol{y})|, \quad 1 \leq i \leq h.
\end{aligned}
$$

Wird Gleichung (2.46) mit $\mathcal{J}'(\boldsymbol{x},\boldsymbol{y}) = \mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y})$ benutzt, vereinfacht sich die rechte Seite der obigen Gleichung in:

$$|\mathcal{D}_1(\hat{\boldsymbol{x}},\hat{\boldsymbol{x}}_i)| = |\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y}) \cap \mathcal{D}_0(\hat{\boldsymbol{x}}_i,\boldsymbol{y})| - |\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y}) \cap \mathcal{D}_1(\hat{\boldsymbol{x}}_i,\boldsymbol{y})| + |\mathcal{D}_1(\hat{\boldsymbol{x}}_i,\boldsymbol{y})|, \quad 1 \leq i \leq h.$$

Unter Ausnutzung der Gleichungen (3.30a), (3.30b) und (3.31) ergibt sich letztlich Gleichung (3.33):

$$\begin{aligned} |\mathcal{D}_1(\hat{\boldsymbol{x}},\hat{\boldsymbol{x}}_i)| &= \sum_{\boldsymbol{\alpha}\in\mathbb{B}^h \,|\, \alpha_i=0} |\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y}) \cap \mathcal{D}_{\boldsymbol{\alpha}}| - \sum_{\boldsymbol{\alpha}\in\mathbb{B}^h \,|\, \alpha_i=1} |\mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y}) \cap \mathcal{D}_{\boldsymbol{\alpha}}| + |\mathcal{D}_1(\hat{\boldsymbol{x}}_i,\boldsymbol{y})| \\ &= \sum_{\boldsymbol{\alpha}\in\mathbb{B}^h} (-1)^{\alpha_i} \, |\mathcal{D}_{\boldsymbol{\alpha}} \cap \mathcal{D}_1(\hat{\boldsymbol{x}},\boldsymbol{y})| + |\mathcal{D}_1(\hat{\boldsymbol{x}}_i,\boldsymbol{y})| \\ &= \sum_{\boldsymbol{\alpha}\in\mathbb{B}^h} (-1)^{\alpha_i} \, m_{\boldsymbol{\alpha}}(\hat{\boldsymbol{x}},\boldsymbol{y}) + |\mathcal{D}_1(\hat{\boldsymbol{x}}_i,\boldsymbol{y})|, \quad 1 \leq i \leq h. \end{aligned} \tag{3.33}$$

A.11 Herleitung von Ungleichung (4.1)

Vorausgesetzt werden ein $(n,k)_2$-Code $\mathcal{C}$ und seine 2^{n-k} disjunkten Nebenklassen $\mathcal{N}_0, \mathcal{N}_1, \ldots, \mathcal{N}_{2^{n-k}-1}$. Nach Abschnitt 2.1.22 ist ein Nebenklassenführer definiert als der wahrscheinlichste Fehlervektor $\boldsymbol{v}_i$ in einer Nebenklasse $\mathcal{N}_i$:

$$\boldsymbol{v}_i \in \mathcal{N}_i, \quad \Pr(\boldsymbol{v}_i) \geq \Pr(\boldsymbol{v}) \implies w_H(\boldsymbol{v}_i) \leq w_H(\boldsymbol{v}), \quad \forall\, \boldsymbol{v} \in \mathcal{N}_i. \tag{A.2}$$

So bilden die Nebenklassenführer eine Menge $\boldsymbol{\mathcal{M}} = \{\boldsymbol{v}_0, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_{2^{n-k}-1}\} \subset \mathbb{F}_2^n$ von 2^{n-k} Fehlervektoren, die in einer HDMDD ausgewählt werden.

Sei $\boldsymbol{\mathcal{A}}$ die Menge der 2^{n-k} wahrscheinlichsten Fehlervektoren in der ganzen Menge $\mathbb{F}_2^n$:

$$\begin{gathered} \boldsymbol{\mathcal{A}} = \{\boldsymbol{a_0}, \boldsymbol{a_1}, \ldots, \boldsymbol{a_{2^{n-k}-1}}\} \subset \mathbb{F}_2^n, \quad |\boldsymbol{\mathcal{A}}| = 2^{n-k}, \\ \Pr(\boldsymbol{a}_j) \geq \Pr(\boldsymbol{a}_l) \implies w_H(\boldsymbol{a}_j) \leq w_H(\boldsymbol{a}_l), \quad \forall\, \boldsymbol{a}_j \in \boldsymbol{\mathcal{A}}, \quad \forall\, \boldsymbol{a}_l \notin \boldsymbol{\mathcal{A}}. \end{gathered} \tag{A.3}$$

In der HDMDD wird jedes Codewort $\boldsymbol{c} \in \mathcal{C}$ falsch geschätzt, falls der Fehlervektor außerhalb der Menge $\boldsymbol{\mathcal{M}}$ liegt:

$$P_{MDD} = \Pr\{\mathbb{F}_2^n \setminus \boldsymbol{\mathcal{M}}\}, \tag{A.4}$$

wobei $\Pr\{\cdot\}$ die gesamte Wahrscheinlichkeit einer Menge bezeichnet.

Wird jedoch ein Decodierer Ψ verwendet, der lediglich die 2^{n-k} wahrscheinlichsten Fehlervektoren $\boldsymbol{a}$ berücksichtigt, schätzt er jedes Codewort $\boldsymbol{c} \in \mathcal{C}$ falsch, nicht nur wenn der Fehlervektor außerhalb der Menge $\boldsymbol{\mathcal{M}}$ liegt, sondern auch wenn der Fehlervektor in der Menge $\boldsymbol{\mathcal{M}} \setminus \boldsymbol{\mathcal{A}}$ liegt:

$$P_{\Psi} = P_{MDD} + \Pr\{\boldsymbol{\mathcal{M}} \setminus \boldsymbol{\mathcal{A}}\}. \tag{A.5}$$

Nach der obigen Definitionen der Mengen $\boldsymbol{\mathcal{M}}$ und $\boldsymbol{\mathcal{A}}$ gilt

$$\Pr\{\boldsymbol{\mathcal{A}}\} \geq \Pr\{\boldsymbol{\mathcal{M}}\}. \tag{A.6}$$

Daraus folgt:

$$\Pr\{\boldsymbol{\mathcal{M}} \setminus \boldsymbol{\mathcal{A}}\} \leq \Pr\{\mathbb{F}_2^n \setminus \boldsymbol{\mathcal{A}}\} \leq \Pr\{\mathbb{F}_2^n \setminus \boldsymbol{\mathcal{M}}\} = P_{MDD}. \tag{A.7}$$

Wenn diese Ungleichung in Gleichung (A.5) eingesetzt wird, wird schließlich Gleichung (4.1) bewiesen:

$$P_{\Psi} \leq 2\, P_{MDD}. \tag{4.1}$$

Literaturverzeichnis

[Agr96] Agrell, Erik: *Voronoi regions for binary linear block codes.* IEEE Transactions on Information Theory, 42(1): 310–316, January 1996.

[Agr98] Agrell, Erik: *On the Voronoi neighbor ratio for binary linear block codes.* IEEE Transactions on Information Theory, 44(7): 3064–3082, November 1998.

[And92] Anderson, Jeffrey L.: *On minimal decoding sets for the extended binary Golay code.* IEEE Transactions on Information Theory, 38(5): 1560–1561, September 1992.

[ASE92] Alon, Noga, Spencer, Joel H., and Erdös, Paul: *The Probabilistic Method.* Wiley Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, New York; Chichester; Brisbane; Toronto; Singapore, 1992.

[Bar91a] Barros, Dulte José de: *Estudos dos Algoritmos de Decodificação com Conjuntos de Informação.* Diplomarbeit, Centro Federal de Educação Tecnológica do Paraná, Curitiba, Paraná, Brasilien, Januar 1991. (auf Portugiesisch).

[Bar91b] Barros, Dulte José de: *Geração Eficiente de uma Variável Aleatória Gaussiana.* Tecnologia & Humanismo, 6(8): 11–13, Mai 1991. (auf Portugiesisch).

[Bar93] Barros, Dulte José de: *Um Novo Enfoque para a Decodificação com Conjuntos de Informação.* Master's Thesis, Centro Federal de Educação Tecnológica do Paraná, Curitiba, Paraná, Brasilien, Januar 1993. (auf Portugiesisch).

[Bar97] BARTSCH, HANS–JOCHEN: *Taschenbuch mathematischer Formeln*. Fachbuchverlag, Leipzig, 17. Auflage, 1997.

[Bar98] BARG, ALEXANDER M.: *Complexity issues in coding theory*. In PLESS, VERA S., HUFFMAN, W. C., and BRUALDI, RICHARD A. (editors): *Handbook of Coding Theory*, volume 1, pages 649–754. Elsevier / North–Holland, Amsterdam, The Netherlands; New York; Oxford, 1998.

[Bar99] BARROS, DULTE JOSÉ DE: *Soft–Decision Decodierung mit Informationsmengen*. In: *KiK' 99 – Workshop Kommunikation im Kleinwalsertal*, Hirschegg, Kleinwalsertal, Österreich, Juni 1999.

[BD97] BARROS, DULTE JOSÉ DE and DORSCH, BERNHARD G.: *An efficient information set algorithm for soft–decision decoding of linear block codes*. In *Proceedings of the 1997 IEEE International Symposium on Information Theory – ISIT 1997*, page 302, Ulm, Germany, June 1997.

[Ber74] BERLEKAMP, ELWYN R. (editor): *Key Papers in The Development of Coding Theory*. IEEE Press, New York, 1974.

[BGW93] BARROS, DULTE JOSÉ DE, GODOY, JR., WALTER, and WILLE, EMÍLIO CARLOS GOMES: *Um novo limiar de parada na decodificação*. In *11º Simpósio Brasileiro de Telecomunicações*, pages 489–492, Natal, Brazil, August 1993.

[BGW94] BARROS, DULTE JOSÉ DE, GODOY, JR., WALTER, and WILLE, EMÍLIO CARLOS GOMES: *The real time – modified information set decoding algorithm*. In *Proceedings of the S-BT/IEEE International Telecommunications Symposium – ITS 94*, pages 93–96, Rio de Janeiro, Brazil, August 1994.

[BGW97] BARROS, DULTE JOSÉ DE, GODOY, JR., WALTER, and WILLE, EMÍLIO CARLOS GOMES: *A new approach to the information set decoding algorithm*. Computer Communications, 20(4): 302–308, June 1997.

[BKv97] BARG, ALEXANDER M., KROUK, EVGUENI AVRAMOVICH, and VAN TILBORG, HENK C. A.: *The complexity of hard–decision decoding of linear codes*. In *Proceedings of the 1997*

IEEE International Symposium on Information Theory, page 331, Ulm, Germany, June 1997.

[BKv99] BARG, ALEXANDER M., KROUK, EVGUENI AVRAMOVICH, and VAN TILBORG, HENK C. A.: *On the complexity of minimum–distance decoding of long linear codes.* IEEE Transactions on Information Theory, 45(5): 1392–1405, July 1999.

[Bla83] BLAHUT, RICHARD E.: *Theory and Practice of Error Control Codes.* Addison–Wesley, Reading, Massachusetts; Harlow, England; Menlo Park, California; Berkeley, California; Don Mills, Ontario; Sydney; Bonn; Amsterdam; Tokyo; Mexico City, 1983.

[Bli87] BLINOVSKY, VOLODIA M.: *Lower asymptotic bound on the number of linear codewords in a sphere of given radius in* F_q^n. Problemy Peredachi Informatsii, 23(2): 50–53 (Russian Original) and 130–132 (English Translation), April–June 1987.

[BMM58] BOX, G. E. P., MULLER, M. E., and MARSAGLIA, G.: *The polar method.* Annals of Mathematics Statistics, 29: 610–611, 1958.

[Bos98] BOSSERT, MARTIN: *Kanalcodierung.* Buchreihe Informationstechnik. B. G. Teubner, Stuttgart, Zweite Auflage, 1998.

[Bre73] BRENT, R. P.: *Algorithms for Minimization without Derivatives.* Prentice–Hall, New Jersey; London; Sydney; Rio de Janeiro; Toronto; New Delhi; Tokyo; Singapore; Wellinton, 1973.

[BSMM99] BRONSTEIN, I. N., SEMENDJAJEW, K. A., MUSIOL, G. und MÜHLIG, H.: *Taschenbuch der Mathematik.* Harri Deutsch, Frankfurt am Main; Thun, 4. Auflage, 1999.

[BV93] BROUWER, ANDRIES E. and VERHOEFF, TOM: *An updated table of minimum–distance bounds binary for linear codes.* IEEE Transactions on Information Theory, 39(2): 662–677, March 1993.

[CC81] CLARK, JR., GEORGE C. and CAIN, J. BIBB: *Error–Correction Coding for Digital Communications.* Plenum Press, New York; London, 1981.

[CG81] CHAN, AGNES HUI and GAMES, RICHARD A.: (n, k, t)-*covering systems and error–trapping decoding*. IEEE Transactions on Information Theory, 27(5): 643–646, September 1981.

[CG90] COFFEY, JOHN T. and GOODMAN, RODNEY M.: *The complexity of information set decoding*. IEEE Transactions on Information Theory, 36(5): 1031–1037, September 1990.

[Cha72] CHASE, DAVID: *A class of algorithms for decoding block codes with channel measurement information*. IEEE Transactions on Information Theory, 18(1): 170–182, January 1972.

[CS99] CONWAY, JOHN HORTON and SLOANE, NEIL JAMES ALEXANDER: *Sphere Packings, Lattices and Groups*, volume 290 of *Grundlehren der mathematischen Wissenschaften – A Series of Comprehensive Studies in Mathematics*. Springer, New York; Berlin; Heidelberg; Barcelona; Hongkong; London; Milan; Paris; Singapore; Tokio, third edition, 1999.

[DB96] DAVID, KLAUS und BENKNER, THORSTEN: *Digitale Mobilfunksysteme*. Buchreihe Informationstechnik. B. G. Teubner, Stuttgart, 1996.

[Dor74] DORSCH, BERNHARD G.: *A decoding algorithm for binary block codes and J-ary output channels*. IEEE Transactions on Information Theory, 20(3): 391–394, May 1974.

[Dor97] DORSCH, BERNHARD G.: *Codierung zur Fehlerkorrektur*. Vorlesungsskript, Technische Universität Darmstadt, 1997.

[Dum] DUMER, ILYA I.: *Ellipsoidal lists and maximum–likelihood decoding*. IEEE Transactions on Information Theory, (submitted for publication).

[Dum81] DUMER, ILYA I.: *Choice of the methods of concatenated decoding depending on noise value*. In *Proceedings of the Third All–Union Conference on Coding Theory and Information Transmission*, volume 2, pages 61–65, 1981.

[Dum89] DUMER, ILYA I.: *Two decoding algorithms for linear codes.* Problemy Peredachi Informatsii, 25(1): 24–32 (Russian Original) and 17–23 (English Translation), January–March 1989.

[Dum96a] DUMER, ILYA I.: *Covering lists in maximum–likelihood decoding.* In *Proceedings of the Thirty-Fourth Annual Allerton Conference on Communication, Control, and Computing*, pages 683–692, Monticello, USA, October 1996.

[Dum96b] DUMER, ILYA I.: *Suboptimal decoding of linear codes: Partition technique.* IEEE Transactions on Information Theory, 42(6): 1971–1986, November 1996.

[Dum97a] DUMER, ILYA I.: *Maximum–likelihood decoding with reduced complexity.* In *Proceedings of the 1997 IEEE International Symposium on Information Theory*, page 396, Ulm, Germany, June 1997.

[Dum97b] DUMER, ILYA I.: *Ellipsoids in Hamming spaces: Isoperimetry, random search, and maximum–likelihood decoding.* In *Proceedings of the Thirty-Fifth Annual Allerton Conference on Communication, Control, and Computing*, pages 976–985, Monticello, USA, October 1997.

[Dum98a] DUMER, ILYA I.: *Information–set soft–decision decoding.* In *Proceedings of the 1998 Information Theory Workshop*, page 77, Killarney, Ireland, June 1998.

[Dum98b] DUMER, ILYA I.: *Ellipsoidal coverings and error-free search in maximum–likelihood decoding.* In *Proceedings of the 1998 IEEE International Symposium on Information Theory*, page 367, Cambridge, Massachusetts, August 1998.

[Dum98c] DUMER, ILYA I.: *Soft–decision decoding via sphero–ellipsoidal coverings.* In *Proceedings of the Sixth International Workshop on Algebraic and Combinatorial Coding Theory*, pages 98–102, Pskov, Russia, September 1998.

[Enn87] ENNS, V. I.: *New bounds of decoding domain for certain methods of error correction with soft decision.* In *Proceedings of the*

Third Joint Soviet–Swedish International Workshop on Information Theory: Convolutional Codes; Multi–User Communication, pages 347–350, Sochi, USSR, May 1987.

[ES74] ERDÖS, PAUL and SPENCER, JOEL H.: *Probabilistic Methods in Combinatorics*. Akadémiai Kiadó, Budapest, 1974.

[ES76] EINARSSON, GÖRAN and SUNDBERG, CARL–ERIK W.: *A note on soft decision decoding with successive erasures*. IEEE Transactions on Information Theory, 22(1): 88–96, January 1976.

[ES96] ECK, PETER and SÖDER, GÜNTER: *Tabulated inversion, a fast method for white gaussian noise simulation*. AEÜ (Archiv für Elektronik und Übertragungstechnik) International Journal of Electronics and Communications, 50(1): 41–48, January 1996.

[Evs83] EVSEEV, G. S.: *On the complexity of decoding of linear block codes*. Problemy Peredachi Informatsii, 19(1): 3–8 (Russian Original) and 1–6 (English Translation), January–March 1983.

[FKT+97] FOSSORIER, MARC P. C., KOUMOTO, TAKUYA, TAKATA, TOYOO, KASAMI, TADAO, and LIN, SHU: *The least stringent sufficient condition on the optimality of a suboptimally decoded codeword using the most reliable basis*. In *Proceedings of the 1997 IEEE International Symposium on Information Theory – ISIT 1997*, page 430, Ulm, Germany, June 1997.

[FL95] FOSSORIER, MARC P. C. and LIN, SHU: *Soft–decision decoding of linear block codes based on ordered statistics*. IEEE Transactions on Information Theory, 41(5): 1379–1396, September 1995.

[FL96a] FOSSORIER, MARC P. C. and LIN, SHU: *Correction to "Soft–decision decoding of linear block codes based on ordered statistics"*. IEEE Transactions on Information Theory, 42(1): 328, January 1996.

[FL96b] FOSSORIER, MARC P. C. and LIN, SHU: *Computationally efficient soft–decision decoding of linear block codes based on ordered statistics*. IEEE Transactions on Information Theory, 42(3): 738–750, May 1996.

[FL97] FOSSORIER, MARC P. C. and LIN, SHU: *Complementary reliability–based decodings of binary linear block codes.* IEEE Transactions on Information Theory, 43(5): 1667–1672, September 1997.

[FM86] FISHMAN, G. S. and MOORE III, L. R.: *An exhaustive analysis of multiplicative congruential random number generators with modulus* $2^{31} - 1$. SIAM Journal of the Science Statistical Computing, 7(1): 24–45, January 1986.

[For66a] FORNEY, JR., G. DAVID: *Concatenated Codes*, volume 37 of *M. I. T. Research Monograph.* M. I. T. Press, Cambridge, Massachusetts, 1966.

[For66b] FORNEY, JR., G. DAVID: *Generalized minimum distance decoding.* IEEE Transactions on Information Theory, 12(2): 125–131, April 1966.

[For91] FORNEY, JR., G. DAVID: *Geometrically uniform codes.* IEEE Transactions on Information Theory, 37(5): 1241–1260, September 1991.

[Fos94] FOSSORIER, MARC P. C.: *Soft–Decision Decoding of Linear Block Codes Based on Ordered Statistics.* PhD thesis, University of Hawaii at Manoa, Honolulu, USA, December 1994.

[Fri95] FRIEDRICHS, BERND: *Kanalcodierung: Grundlagen und Anwendungen in modernen Kommunikationssystemen.* Buchreihe Information und Kommunikation. Springer, Berlin; Heidelberg; New York; Barcelona; Budapest; Hongkong; London; Mailand; Paris; Santa Clara; Singapur; Tokio, 1995.

[FV96] FORNEY, JR., G. DAVID and VARDY, ALEXANDER: *Generalized minimum distance decoding of euclidean–space codes and lattices.* IEEE Transactions on Information Theory, 42(6): 1992–2026, November 1996.

[GA88] GODOY, JR., WALTER and ARANTES, DALTON SOARES: *Sub–optimum soft–decision decoding of block codes using the zero–neighbors algorithm.* In *Proceedings of the 1988 IEEE International Symposium on Information Theory*, Kobe, Japan, 1988.

[GA89] GODOY, JR., WALTER and ARANTES, DALTON SOARES: *An algorithm for soft–decision minimum weight decoding of block codes.* In *Proceedings of the International Symposium on Signals, Systems and Electronics – ISSSE 89*, pages 18–20, Erlangen, Germany, September 1989.

[Gil52] GILBERT, EDGAR N.: *A comparison of signalling alphabets.* The Bell System Technical Journal, 31: 504–522, May 1952. (In [Ber74]).

[God91] GODOY, JR., WALTER: *Esquemas de Modulação Codificada com Códigos de Bloco.* Editora CEFET–PR, Curitiba, Paraná, Brasilien, 1991. (auf Portugiesisch).

[Gor82] GORDON, DANIEL M.: *Minimal permutation sets for decoding the binary Golay code.* IEEE Transactions on Information Theory, 28(3): 541–543, May 1982.

[GS97] GAZELLE, DAVID and SNYDERS, JAKOV: *Reliability–based code–search algorithms for maximum–likelihood decoding of block codes.* IEEE Transactions on Information Theory, 43(1): 239–249, January 1997.

[Han98] HAN, YUNGHSIANG S.: *A new treatment of priority–first search maximum–likelihood soft–decision decoding of linear block codes.* IEEE Transactions on Information Theory, 44(7): 3091–3096, November 1998.

[HH92] HAN, YUNGHSIANG S. and HARTMANN, CARLOS R. P.: *Designing efficient maximum–likelihood soft–decision decoding algorithms for linear block codes using algorithm A*.* Technical Report SU–CIS–92–10, School of Computer and Information Science, Syracuse University, New York, June 1992.

[HHC91] HAN, YUNGHSIANG S., HARTMANN, CARLOS R. P., and CHEN, CHIH–CHIEH J.: *Efficient maximum–likelihood soft–decision decoding of linear block codes using algorithm A*.* Technical Report SU–CIS–91–42, School of Computer and Information Science, Syracuse University, New York, December 1991.

[HT90] HOLLMANN, HENK D. L. and TOLHUIZEN, LUDOVICUS MARINUS GERARDUS MARIA: *Improved conditions for generalized minimum distance decoding, with applications to the decoding of product codes and generalized concatenated codes.* Technical report, Philips Research Laboratories, Einhoven, The Netherlands, November 1990.

[Hub92] HUBER, JOHANNES: *Trelliscodierung: Grundlagen und Anwendungen in der digitalen Übertragungstechnik.* Buchreihe Nachrichtentechnik. Springer, Berlin; Heidelberg; New York; Barcelona; Budapest; Hongkong; London; Mailand; Paris; Santa Clara; Singapur; Tokio, 1992.

[JT98] JOHNSON, MICHAEL K. und TROAN, ERIK W.: *Anwendungen entwickeln unter Linux.* Addison–Wesley, Reading, Massachusetts; Harlow, England; Menlo Park, California; Berkeley, California; Don Mills, Ontario; Sydney; Bonn; Amsterdam; Tokyo; Mexico City, 1998.

[Kab91] KABATYANSKII, GRISCHA A.: *About metrics and decoding domains of Forney's algorithm.* In *Proceedings of the Fifth Joint Soviet–Swedish International Workshop on Information Theory: Convolutional Codes; Multi–User Communication*, pages 81–85, Moscow, USSR, January 1991.

[Kam96] KAMMEYER, KARL DIRK: *Nachrichtenübertragung.* Buchreihe Informationstechnik. B. G. Teubner, Stuttgart, Zweite Auflage, 1996.

[KF95] KROUK, EVGUENI AVRAMOVICH and FEDORENKO, SERGEI VALENTINOVICH: *Decoding by generalized information sets.* Problemy Peredachi Informatsii, 31(2): 54–61 (Russian Original) and 143–149 (English Translation), April–June 1995.

[KK98a] KASAMI, TADAO and KOUMOTO, TAKUYA: *Computation complexity for computing sufficient conditions on the optimality of a decoded codeword.* Technical Report NAIST–IS–TR98008, Nara Institute of Science and Technology – Graduate School of Information Science, Japan, July 1998.

[KK98b] KOUMOTO, TAKUYA and KASAMI, TADAO: *Analysis and improvement on GMD like decoding algorithms.* Technical Report IT98–10, pages 53–58, Institute of Electronics, Information and Communication Engineers, Japan, 1998.

[KKL98] KOUMOTO, TAKUYA, KASAMI, TADAO, and LIN, SHU: *A sufficient condition for ruding out some useless test error patterns in Chase–type decoding algorithms.* IEICE Transactions on Fundamental of Electronics, Communications and Computer Sciences, E81–A(2): 321–326, 1998.

[KKTL95a] KASAMI, TADAO, KOUMOTO, TAKUYA, TAKATA, TOYOO, and LIN, SHU: *The effectiveness of the least stringent sufficient condition on the optimality of decoded codewords.* In *Proceedings of the Third International Symposium on Communication Theory and Applications*, pages 324–333, Ambleside, UK, July 1995.

[KKTL95b] KASAMI, TADAO, KOUMOTO, TAKUYA, TAKATA, TOYOO, and LIN, SHU: *The least stringent sufficient conditions on the optimality of decoded codewords.* In *Proceedings of the 1995 IEEE International Symposium on Information Theory*, page 470, Whistler, Canada, September 1995.

[KNH97] KANEKO, TOSHIMITSU, NISHIJIMA, TOSHIHISA, and HIRASAWA, SHIGEICHI: *An improvement of soft–decision maximum–likelihood decoding algorithm using hard–decision bounded–distance decoding.* IEEE Transactions on Information Theory, 43(4): 1314–1319, July 1997.

[KNIH94] KANEKO, TOSHIMITSU, NISHIJIMA, TOSHIHISA, INAZUMI, HIROSHIGE, and HIRASAWA, SHIGEICHI: *An efficient maximum–likelihood decoding algorithm for linear block codes with algebraic decoder.* IEEE Transactions on Information Theory, 40(2): 320–327, March 1994.

[KNT+95] KOUMOTO, TAKUYA, NAGANO, HIDEYUKI, TAKATA, TOYOO, FUJIWARA, TORU, KASAMI, TADAO, and LIN, SHU: *A new iterative soft–decision decoding algorithm.* Technical Report IT95–28, pages 19–24, Institute of Electronics, Information and Communication Engineers, Japan, 1995.

[Knu98a] KNUTH, DONALD ERVIN: *The Art of Computer Programming*, volume 2: Seminumerical Algorithms. Addison–Wesley, Reading, Massachusetts; Harlow, England; Menlo Park, California; Berkeley, California; Don Mills, Ontario; Sydney; Bonn; Amsterdam; Tokyo; Mexico City, third edition, 1998.

[Knu98b] KNUTH, DONALD ERVIN: *The Art of Computer Programming*, volume 3: Sorting and Searching. Addison–Wesley, Reading, Massachusetts; Harlow, England; Menlo Park, California; Berkeley, California; Don Mills, Ontario; Sydney; Bonn; Amsterdam; Tokyo; Mexico City, second edition, 1998.

[KR77] KERNIGHAN, BRIAN W. and RITCHIE, DENIS M.: *The C Programming Language*. Prentice–Hall, New Jersey; London; Sydney; Rio de Janeiro; Toronto; New Delhi; Tokyo; Singapore; Wellinton, 1977.

[Kro89] KROUK, EVGUENI AVRAMOVICH: *Decoding complexity bound for linear block codes*. Problemy Peredachi Informatsii, 25(3): 103–107 (Russian Original) and 251–254 (English Translation), July–September 1989.

[KTK+95] KASAMI, TADAO, TAKATA, TOYOO, KOUMOTO, TAKUYA, FUJIWARA, TORU, YAMAMOTO, HIROSHI, and LIN, SHU: *The least stringent sufficient condition on optimality of a suboptimal decoded codewords*. Technical Report IT94–82, pages 19–24, Institute of Electronics, Information and Communication Engineers, Japan, January 1995.

[KTKL96] KOUMOTO, TAKUYA, TAKATA, TOYOO, KASAMI, TADAO, and LIN, SHU: *An iterative soft–decision decoding algorithm*. In *Proceedings of the International Symposium on Information Theory and its Applications*, pages 806–810, Victoria, Canada, September 1996.

[KTKL99] KOUMOTO, TAKUYA, TAKATA, TOYOO, KASAMI, TADAO, and LIN, SHU: *A low–weight trellis–based iterative soft–decision decoding algorithm for binary linear block codes*. IEEE Transactions on Information Theory, 45(2): 731–741, March 1999.

[LC83] LIN, SHU and COSTELLO, JR., DANIEL J.: *Error Control Coding: Fundamentals and Applications.* Prentice–Hall Series in Computer Applications in Electrical Engineering. Prentice–Hall, New Jersey; London; Sydney; Rio de Janeiro; Toronto; New Delhi; Tokyo; Singapore; Wellinton, 1983.

[L'E88] L'ECUYER, PIERRE: *Efficient and portable combined random number generators.* Communications of the ACM, 31(6): 742–749 and 774, June 1988.

[Lev86] LEVA, JOSEPH L.: *A fast normal random number generator.* ACM Transactions on Mathematical Software, 18(4): 449–455, December 1986.

[Lie93] LIESENFELD, BERNHARD: *Codekonstruktionen mit modifizierter mehrdimensionaler DFT.* Fortschritt–Berichte VDI, Reihe 10: Informatik/Kommunikationstechnik, Nummer 258, VDI–Verlag, Düsseldorf, 1993.

[Luc97] LUCAS, RAINER: *On iterative soft–decision decoding of linear binary block codes.* Fortschritt–Berichte VDI, Reihe 10: Informatik/Kommunikationstechnik, Nummer 511, VDI–Verlag, Düsseldorf, 1997.

[MF93] MIRONCHIKOV, E. T. and FEDORENKO, SERGEI VALENTINOVICH: *Generalized information–set decoding of (L, g)-codes.* Problemy Peredachi Informatsii, 29(4): 94–98 (Russian Original) and 381–384 (English Translation), October–December 1993.

[ML85] MICHELSON, ARNOLD M. and LEVESQUE, ALLEN H.: *Error–Control Techniques for Digital Communication.* John Wiley & Sons, New York; Chichester; Brisbane; Toronto; Singapore, 1985.

[MLK95a] MOORTHY, HARY THIRU, LIN, SHU, and KASAMI, TADAO: *An efficient soft–decision decoding scheme for binary linear block codes.* In *Proceedings of the Third International Symposium on Communication Theory and Applications*, pages 4–11, Ambleside, UK, July 1995.

[MLK95b] MOORTHY, HARY THIRU, LIN, SHU, and KASAMI, TADAO: *Soft–decision decoding of binary linear block codes based on an iterative search algorithm.* In *Proceedings of the 1995 IEEE International Symposium on Information Theory*, page 474, Whistler, Canada, September 1995.

[MLK97] MOORTHY, HARY THIRU, LIN, SHU, and KASAMI, TADAO: *Soft–decision decoding of binary linear block codes based on an iterative search algorithm.* IEEE Transactions on Information Theory, 43(3): 1030–1040, May 1997.

[MS83] MACWILLIAMS, FLORENCE JESSIE and SLOANE, NEIL JAMES ALEXANDER: *The Theory of Error–Correcting Codes.* Elsevier / North–Holland, Amsterdam, The Netherlands; New York; Oxford, third edition, 1983.

[Nil94] NILSSON, JAN: *On Hard and Soft Decoding of Block Codes.* PhD thesis 333, Linköping University, Linköping, Sweden, April 1994.

[Per94] PERLMAN, G.: *Unix for Software Developers.* Prentice–Hall, New Jersey; London; Sydney; Rio de Janeiro; Toronto; New Delhi; Tokyo; Singapore; Wellinton, 1994.

[Pet67] PETERSON, W. WESLEY: *Prüfbare und korrigierbare Codes.* R. Oldenbourg, München; Wien, 1967.

[PM88] PARK, STEPHEN K. and MILLER, KEITH W.: *Random number generators: Good ones are hard to find.* Communications of the ACM, 31(10): 1192–1201, October 1988.

[Pro95] PROAKIS, JOHN G.: *Digital Communications.* McGraw–Hill Series in Electrical and Computer Engineering. McGraw–Hill, New York; St. Louis; San Francisco; Auckland; Bogotá; Caracas; Lisbon; London; Madrid; Mexico City; Milan; Montreal; New Delhi; San Juan; Singapore; Sydney; Tokyo; Toronto, third edition, 1995.

[PTVF92] PRESS, WILLIAM H., TEUKOLSKY, SAUL A., VETTERLING, WILLIAM T., and FLANNERY, BRIAN P.: *Numerical Recipes*

in C: The Art of Scientific Computing. Cambridge University, Cambridge; New York; Melbourne, second edition, 1992.

[Pus96] PUSCH, WOLFGANG: *Kanalangepaßte Trelliscodes mit Runlength–Beschränkung*. Doktorarbeit, Technische Universität Wien, Österreich, 1996.

[Ric95] RICE, MICHAEL: *A geometric approach to incomplete soft decision block decoding*. IEEE Transactions on Communications, 43(2/3/4): 1383–1391, February/March/April 1995.

[Sch64] SCHÖNHEIM, J.: *On coverings*. Pacific Journal of Mathematics, 14: 1401–1411, 1964.

[Sha48a] SHANNON, CLAUDE ELWOOD: *A mathematical theory of communication*. The Bell System Technical Journal, 27(3): 379–423, July 1948. (In [SW93]).

[Sha48b] SHANNON, CLAUDE ELWOOD: *A mathematical theory of communication*. The Bell System Technical Journal, 27(4): 623–656, October 1948. (In [SW93]).

[SJ98] SWASZEK, PETER F. and JONES, WILLIAM: *How often is hard–decision decoding enough?* IEEE Transactions on Information Theory, 44(3): 1187–1193, May 1998.

[Söd93] SÖDER, GÜNTER: *Modellierung, Simulation und Optimierung von Nachrichtensystemen*. Buchreihe Nachrichtentechnik. Springer, Berlin; Heidelberg; New York; Barcelona; Budapest; Hongkong; London; Mailand; Paris; Santa Clara; Singapur; Tokio, 1993.

[Stö99] STÖCKER, HORST: *Taschenbuch mathematischer Formeln und moderner Verfahren*. Harri Deutsch, Frankfurt am Main; Thun, 4. Auflage, 1999.

[SW93] SLOANE, NEIL JAMES ALEXANDER and WYNER, AARON D. (editors): *Claude Elwood Shannon: Collected Papers*. IEEE Press, New York, 1993.

[SWHM98] SHIH, CHING-CHENG, WULFF, CHRISTOPHER R., HARTMANN, CARLOS R. P., and MOHAN, CHILUKURI K.: *Efficient heuristic search algorithms for soft–decision decoding of linear block codes.* IEEE Transactions on Information Theory, 44(7): 3023–3038, November 1998.

[TH93] TZSCHACH, HANS und HASSLINGER, GERHARD: *Codes für den störungssicheren Datentransfer.* R. Oldenbourg, München; Wien, 1993.

[Tol96] TOLHUIZEN, LUDOVICUS MARINUS GERARDUS MARIA: *Cooperating Error–Correcting Codes and their Decoding.* PhD thesis, Eindhoven University of Technology, Einhoven, The Netherlands, June 1996.

[TP89] TAIPALE, DANA J. and PURSLEY, MICHAEL B.: *New results on soft–decision decoding of block codes.* In *Proceedings of the IEEE Military Communications Conference*, pages 546–550, Boston, USA, October 1989.

[TP91] TAIPALE, DANA J. and PURSLEY, MICHAEL B.: *An improvement to generalized–minimum–distance decoding.* IEEE Transactions on Information Theory, 37(1): 167–172, January 1991.

[VTPF92] VETTERLING, WILLIAM T., TEUKOLSKY, SAUL A., PRESS, WILLIAM H., and FLANNERY, BRIAN P.: *Numerical Recipes: Example Book (C).* Cambridge University, Cambridge; New York; Melbourne, second edition, 1992.

[Wal96] WALLACE, C. S.: *Fast pseudorandom generators for normal and exponential variates.* ACM Transactions on Mathematical Software, 22(1): 119–127, March 1996.

[Wol83] WOLFMANN, JACQUES: *A permutation decoding of the* $(24, 12, 8)$ *Golay code.* IEEE Transactions on Information Theory, 29(5): 748–750, September 1983.

[Wu97] WU, PEI-CHI: *Multiplicative, congruential random–number generators with multiplier* $\pm 2^{k1} \pm 2^{k2}$ *and modulus* $2^p - 1$. ACM Transactions on Mathematical Software, 23(2): 255–265, June 1997.

[YC80] YU, CHRISTOPHER CHI-HSUN and COSTELLO, JR., DANIEL J.: *Generalized minimum distance decoding algorithms for Q ary output channels.* IEEE Transactions on Information Theory, 26(2): 238–243, March 1980.

[YK97a] YUANSHENG, TANG and KASAMI, TADAO: *One sufficient termination condition for some iterative soft–decision algorithms decoding binary linear block codes.* Technical Report NAIST–IS–TR97012, Nara Institute of Science and Technology – Graduate School of Information Science, Japan, June 1997.

[YK97b] YUANSHENG, TANG and KASAMI, TADAO: *On a testing condition on the optimality of a decoded codeword of binary block codes.* In *Proceedings of the 20th Symposium on Information Theory and its Applications*, pages 321–324, Matsuyama, Japan, December 1997.

[YKF98] YUANSHENG, TANG, KASAMI, TADAO, and FUJIWARA, TORU: *An optimality testing algorithm for a decoded codeword of binary block codes.* Technical Report NAIST–IS–TR98012, Nara Institute of Science and Technology – Graduate School of Information Science, Japan, October 1998.

Lebenslauf

Persönliche Daten:

Name:	Barros, Dulte José de
geboren:	28.2.1965 in Curitiba / Paraná / Brasilien
Staatsangehörigkeit:	brasilianisch
Familienstand:	verheiratet seit 9/1995

Ausbildung:

1980–1983	Technischer Kurs an der CEFET–PR in Curitiba (9.–12. Schuljahr) Abschluß: Elektroniktechniker
1983	Hochschulaufnahmeprüfung: bestanden
1983–1989	Studium der Ingenieurwissenschaft an der Technischen Universität CEFET–PR in Curitiba Abschluß: Elektroingenieur mit Schwerpunkten in Elektronik und Nachrichtentechnik
1990–1993	*Master* in Telematik an der Technischen Universität CEFET–PR in Curitiba Abschluß: *Master of Electrical Engineering in Telematic* Prädikat: *cum laude*
1990–1991	Stipendium vom CNPq (eine brasilianische Organisation wie der DAAD) für die Erlangung des *Master* Abschlusses
1991–1992	Stipendium vom CPqD (eine andere brasilianische Organisation wie der DAAD) für die Entwicklung eines Computerprogramms zur Simulation digitaler Mobilfunksysteme

seit 12/1995	Stipendium vom DAAD für einen viermonatigen Deutschkurs in Mannheim und für die Arbeit an einer Dissertation
seit 4/1996	Arbeit an der Dissertation in Elektrotechnik / Nachrichtentechnik am Institut für Nachrichtentechnik der Technischen Universität Darmstadt unter Betreuung von Prof. Dr.–Ing. Bernhard G. Dorsch

Auszeichnungen:

1979	Bester Schüler im 8. Schuljahr an der Schule *Colégio Estadual do Paraná* in Curitiba mit einer Gesamtnote: 99,0 (von 0 bis 100)
1980–1983	Zweitbester Schüler aller Fachrichtungen am technischen Kurs am CEFET–PR in Curitiba mit einer Gesamtnote: 94,6 (von 0 bis 100)
1983–1989	Bester Student aller Fachrichtungen an der Technischen Universität CEFET–PR in Curitiba mit einer Gesamtnote: 90,7 (von 0 bis 100)

Berufstätigkeit:

2/1984–5/1984	Elektroniktechniker bei SCHAUSE S. A. in Curitiba
2/1985–3/1990	Elektroniktechniker und Ingenieur bei TELEPAR S. A. in Curitiba
seit 10/1991	Hochschullehrer an der Technischen Universität CEFET–PR in Curitiba
9/1993–2/1994	Stipendium vom DAAD für eine Dozententätigkeit an der Fachhochschule München auf dem Gebiet der digitalen Signalverarbeitung

Sprachkenntnisse:

Portugiesisch:	Muttersprache
Deutsch:	PNDS (Prüfung zum Nachweis deutscher Sprachkenntnisse)
Englisch:	Mittelstufe
Spanisch:	Mittelstufe
Französisch:	Mittelstufe
Italienisch:	Grundstufe