Codierungsverfahren in der Rechnerkommunikation: Spezifikationen und Leistungsanalysen

Diplomarbeit von
Jens Hofmann
bei Prof. Dr. Dieter Baum,
Universität Trier, Fachbereich IV

Trier, im September 1994

Vorwort

Diese Diplomarbeit befaßt sich mit Verfahren zur Fehlerkontrolle in Datenübertragungssystemen. Ein typisches Anwendungsbeispiel stellt die Satellitenkommunikation dar. Das Interesse des Autors an diesem Themengebiet wurde durch die Vorlesungen „Lokale Rechnernetze“ von Prof. Dr. Baum an der Universität Trier und „Coding Theory“ von Dr. Slater an der University of Bristol geweckt.

Die Grundlage für die ursprüngliche Gliederung bildete eine im Januar 1993 erschienene Arbeit von V. K. Oduol und S. D. Morgera mit dem Titel „Performance Evaluation of the Generalized Type-II Hybrid ARQ Scheme with Noisy Feedback on Markov Channels“ ([15]). Hiervon ausgehend sollten ARQ- und Hybrid-ARQ-Systeme mit gestörtem Rückkopplungskanal untersucht und mit den entsprechenden Systemen mit ungestörtem Rückkopplungskanal verglichen werden. Es stellte sich jedoch bald heraus, daß bereits die Analyse der letztgenannten Übertragungssysteme mehr als genügend Stoff für eine Diplomarbeit bietet. Aus diesem Grund wurde auch auf die Behandlung von Faltungscodes verzichtet.

Das in dieser Arbeit betrachtete stark abstrahierte Kommunikationsmodell wird im ersten Kapitel definiert. Außerdem werden in diesem Kapitel die später zu analysierenden ARQ- und Hybrid-ARQ-Systeme kurz vorgestellt.

Das zweite Kapitel enthält eine kurze Einführung in die Theorie einiger Blockcodes, die in ARQ- und Hybrid-ARQ-Systemen verwendet werden können. Hier muß natürlich eine Beschränkung auf die — nach Meinung des Autors — wesentlichsten Ergebnisse erfolgen. Eine ausführliche Behandlung der Codes würde den Rahmen dieser Diplomarbeit sprengen.

In den Kapiteln 3 bis 6 werden schließlich das ARQ- und verschiedene Hybrid-ARQ-Systeme spezifiziert und analysiert. Zu den jeweiligen Analysen gehört auch ein Vergleich mit den anderen hier betrachteten Übertragungssystemen.

Die Frage nach der praktischen Relevanz der Ergebnisse bleibt unbeantwortet, sie muß den Technikern überlassen werden. Der Autor hofft jedoch, daß seine Arbeit sowohl dem Theoretiker als auch dem Praktiker interessante Erkenntnisse vermittelt.

Trier, im September 1994
Jens Hofmann

Inhaltsverzeichnis

Symbolverzeichnis **v**

1 Einleitung **1**
1.1 Ein Modell eines Kommunikationssystems 1
1.2 ARQ-Systeme 2
1.3 FEC-Systeme 3
1.4 Hybrid-ARQ-Systeme 3
1.4.1 Klassifizierung von Hybrid-ARQ-Systemen 3
1.4.2 Codeverkettung 4

2 Spezielle Codes **8**
2.1 Zyklische Codes 8
2.2 BCH–Codes 11
2.2.1 Binäre BCH–Codes 13
2.2.2 Codierung von BCH–Codes 14
2.2.3 Decodierung von BCH–Codes 15
2.2.4 Verkürzte und erweiterte BCH–Codes 18
2.3 Reed–Solomon–Codes 18
2.3.1 Binäre Codes aus RS–Codes 20
2.3.2 Codierung von RS–Codes 21
2.3.3 Decodierung von RS–Codes 22
2.4 Polynomiale Codes 23
2.4.1 Codierung und Decodierung polynomialer Codes 23
2.4.2 Fehlererkennung 24
2.4.3 Standardisierte polynomiale Codes 26
2.5 Hamming–Codes 26
2.5.1 Zyklische Hamming–Codes 28
2.5.2 Simplex–Codes (duale Hamming–Codes) 29
2.5.3 Gewichtszeiger der binären Hamming–Codes 29
2.5.4 Codierung von Hamming–Codes 30
2.5.5 Decodierung von Hamming–Codes 31
2.5.6 Verkürzte und erweiterte Hamming–Codes 31

2.6 Krishna–Morgera–Codes . . . 32
2.6.1 Berechnung von Bilinearformen und Konstruktion der KM–Codes . . . 32
2.6.2 Bestimmung der Generatormatrix . . . 35
2.6.3 Codierung von KM–Codes . . . 42
2.6.4 Decodierung von KM–Codes . . . 43
2.6.5 Partitionierung der Generatormatrix . . . 46

3 ARQ-Systeme 48
3.1 Zuverlässigkeit . . . 48
3.2 Durchsatz . . . 50
3.2.1 Selective-Repeat-ARQ . . . 51
3.2.2 Go-Back-M-ARQ . . . 51

4 Typ-I Hybrid-ARQ-Systeme 52
4.1 Zuverlässigkeit . . . 52
4.1.1 Perfekte Codes als innere Codes . . . 54
4.2 Durchsatz . . . 57
4.3 Vergleich mit ARQ-Systemen: Die Kløve–Miller–Bedingung . . . 58
4.3.1 Resümee . . . 64

5 Typ-II Hybrid-ARQ-Systeme 65
5.1 Spezifikation . . . 65
5.2 Leistungsanalyse . . . 66
5.2.1 Notation . . . 68
5.2.2 Zuverlässigkeit . . . 69
5.2.3 Durchsatz . . . 71
5.3 Vergleich mit ARQ- und Typ-I Hybrid-ARQ-Systemen . . . 72
5.3.1 Vergleich mit ARQ-Systemen . . . 74
5.3.2 Vergleich mit Typ-I Hybrid-ARQ-Systemen . . . 77
5.3.3 Resümee . . . 79

6 Verallgemeinerte Typ-II Hybrid-ARQ-Systeme 80
6.1 Spezifikation . . . 80
6.2 Leistungsanalyse . . . 81
6.2.1 Notation . . . 81
6.2.2 Zuverlässigkeit . . . 82
6.2.3 Durchsatz . . . 87
6.3 Vergleich mit ARQ- und Typ-I Hybrid-ARQ-Systemen . . . 90
6.3.1 Vergleich mit ARQ-Systemen . . . 90
6.3.2 Vergleich mit Typ-I Hybrid-ARQ-Systemen . . . 96
6.3.3 Resümee . . . 99
6.4 Vergleich zweier GH-ARQ-Systeme verschiedener Tiefe . . . 100
6.4.1 Resümee . . . 102

Abschließende Betrachtung **103**

Literaturverzeichnis **104**

Symbolverzeichnis

Wir bezeichnen in dieser Arbeit Codes mit kalligraphischen Großbuchstaben. Des weiteren bezeichnen wir Matrizen mit fettgedruckten Großbuchstaben, Spaltenvektoren mit fettgedruckten Kleinbuchstaben und Zeilenvektoren (Wörter) mit kursiven Kleinbuchstaben sowie einem Querstrich. Matrix- bzw. Vektortransposition kennzeichnen wir durch ein hochgestelltes „T“.

Wir benutzen folgende Abkürzungen und Bezeichnungen:

ACK	:	positive acknowledgement,
$d(\bar{x}, \bar{y})$	:	Hamming–Abstand der Wörter $\bar{x}$ und $\bar{y}$,
$\deg(g(x))$	:	Grad des Polynoms $g(x)$,
GBN	:	Go-Back-Protokoll mit Rückgrifftiefe N,
$GF(q)$	:	Galois–Körper der Ordnung q (q sei eine Primzahlpotenz),
$GF(q)^*$	:	Multiplikative Gruppe von $GF(q)$,
$GF(q)[x]$	:	Ring der Polynome über $GF(q)$,
GH-ARQ	:	Verallgemeinertes Typ-II Hybrid-ARQ-System,
HI-ARQ	:	Typ-I Hybrid-ARQ-System,
HII-ARQ	:	Typ-II Hybrid-ARQ-System,
lcm	:	kleinstes gemeinsames Vielfaches,
$\mathbb{N}$	:	Menge der natürlichen Zahlen (ohne 0),
NACK	:	negative acknowledgement,
(n, k)-Code	:	linearer Blockcode der Länge n und Dimension k,
(n, k, d)-Code	:	(n, k)-Code mit Minimalabstand d,
$[n, M, d]$-Code	:	Blockcode der Länge n mit M-Codewörtern und Minimalabstand d,
$\bar{0}$	:	Nullwort (Zeilenvektor),
$\mathbf{0}$	:	Nullvektor (Spaltenvektor),
$\mathbf{O}$	:	Nullmatrix,
Prob(A)	:	Wahrscheinlichkeit für das Eintreten des Ereignisses A (in einem geeigneten Wahrscheinlichkeitsraum),
$\mathbb{R}$	:	Menge der reellen Zahlen,
SR	:	Selective-Repeat-Protokoll,
$w(\bar{x})$	:	Hamming–Gewicht des Wortes $\bar{x}$.

Weitere Abkürzungen und Bezeichnungen, die nur in einzelnen Kapiteln oder Abschnitten Verwendung finden, werden dort definiert.

Kapitel 1

Einleitung

1.1 Ein Modell eines Kommunikationssystems

Wir wollen in dieser Arbeit ein stark abstrahiertes Kommunikationssystem betrachten: Ein Sender überträgt ein Informationswort über einen fehleranfälligen Kanal an einen Empfänger. Um Übertragungsfehler erkennen und korrigieren zu können, wird das Informationswort vor der Einspeisung in den Kanal unter Verwendung eines geeigneten Codes codiert und das so erhaltene Codewort eventuell noch in einzelne Blöcke aufgeteilt. Diese werden nach der Übertragung wieder decodiert.

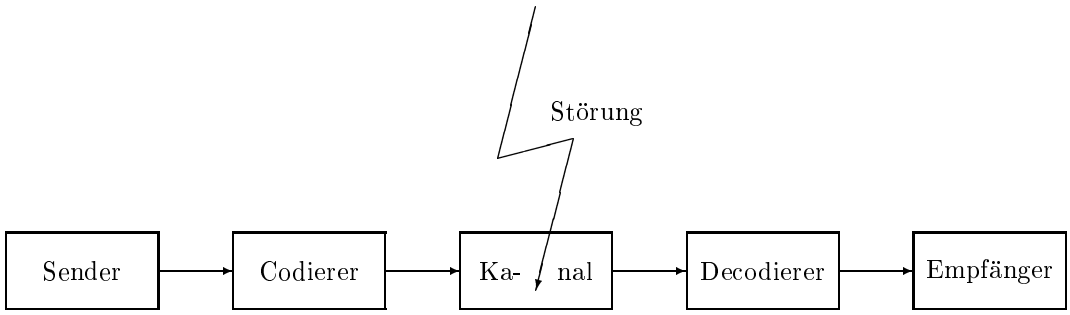

Abb. 1.1: Modell eines Kommunikationssystems

Für dieses System sollen folgende Annahmen gelten:

- Die Übertragung vom Sender zum Codierer und vom Decodierer zum Empfänger erfolge fehlerfrei.
- Der Kanal sei ein gedächtnisloser binärer symmetrischer Kanal (BSC = Binary Symmetric Channel) mit Bitfehlerwahrscheinlichkeit $\varepsilon \in (0, \frac{1}{2}]$.

- Die Codierung diene nur der Fehlerkontrolle.
- Jedes Informationswort sei ein Wort der Länge k mit Symbolen aus $GF(2^h)$[1] und kann als Binärwort der Länge hk betrachtet werden.

In den folgenden Abschnitten werden die wichtigsten Fehlerkontrollsysteme (bzw. deren Protokolle) kurz vorgestellt. Eine detailliertere Betrachtung findet man z. B. in dem Buch von Bhargava e. a. ([2, S. 29ff und S. 323ff]). Eine genaue Analyse erfolgt in den späteren Kapiteln.

1.2 ARQ-Systeme

Ein ARQ-System (ARQ = Automatic Repeat reQuest) basiert auf der Benutzung eines fehlererkennenden Codes. Der Decodierer überprüft, ob der empfangene Block Fehler enthält, und teilt dies dem Sender über einen Rückkopplungskanal (Feedback-Kanal) mit.

Falls keine Fehler entdeckt wurden, sendet der Decodierer eine positive Bestätigung (ACK = positive ACKnowledgement). Die Übertragung dieses Blocks ist damit beendet.

Wurden jedoch Fehler entdeckt, so sendet der Decodierer eine negative Bestätigung (NACK = Negative ACKnowledgement). In diesem Fall wird der Block erneut übertragen. Es gibt verschiedene Protokolle für die Durchführung der Reübertragung, die beiden wichtigsten sind das Go–Back N ARQ und das Selective–Repeat ARQ.

Beim *Go-Back-N-ARQ* (auch *Continuous-ARQ*) werden kontinuierlich Datenblöcke übertragen. Sobald der Sender ein NACK empfängt, wird die Übertragung der N zuletzt gesendeten Blöcke wiederholt. Hierbei ist N die maximale Anzahl von Blöcken, die zwischen dem Beginn des Sendens eines Blocks und dem Erhalt der zugehörigen Bestätigung übertragen werden können. Es gibt Variationen dieses Protokolls, bei denen nur der fehlerhafte Block und alle nachfolgenden wiederholt werden, dies können auch weniger als N sein. Hierfür ist es natürlich erforderlich, den fehlerhaften Block identifizieren zu können. Man könnte N hier als durchschnittliche Anzahl zu wiederholender Blöcke interpretieren, es dürfte dann aber schwierig sein, N exakt zu bestimmen. Wir werden in dieser Arbeit nur das „reine“ Go-Back-N-Protokoll betrachten, das stets N Blöcke wiederholt.

Das *Selective-Repeat-ARQ* verhält sich zunächst genauso wie das Go-Back-N-ARQ, jedoch wird im Falle eines Fehlers nur der fehlerhafte Block erneut übertragen. Dies erfordert, daß die Bestätigung des Empfängers die Identifizierung des entsprechenden Blocks ermöglicht.

Die Beschreibung dieser Protokolle zeigt bereits einen Nachteil der ARQ-Systeme: Auf der Seite des Senders muß genügend Speicherplatz vorhanden sein, um alle Blöcke, deren Übertragung noch nicht beendet ist, speichern zu können. Beim Selective-Repeat-ARQ muß zusätzlich Speicherplatz auf der Seite des Empfängers bereitgestellt werden, weil die einzelnen Blöcke eventuell nicht in der richtigen Reihenfolge eintreffen. Ein weiterer Nachteil der ARQ-Systeme ist der zusätzliche Aufwand für den Rückkopplungskanal.

[1] $GF(2^h)$ bezeichne den Galois-Körper der Ordnung 2^h.

Die wichtigsten Vorteile der ARQ-Systeme sind die hohe erreichbare Zuverlässigkeit, d. h. die geringe Wahrscheinlichkeit unentdeckter Fehler, sowie die einfache technische Realisierung des Codierers und des Decodierers ([2, S. 31]).

Der Durchsatz eines ARQ-Systems, d. h. das durchschnittliche Verhältnis erfolgreich übertragener Bits zu insgesamt übertragenen Bits, hängt von der Anzahl benötigter Reübertragungen ab. Diese wird durch die Kanalqualität und das verwendete Protokoll bestimmt.

1.3 FEC-Systeme

Ein FEC-System (FEC = Forward-acting Error Control) basiert auf der Benutzung eines fehlerkorrigierenden Codes. Der Decodierer versucht, aus der empfangenen Nachricht den ursprünglichen Block zu rekonstruieren und gibt das Ergebnis an den Empfänger weiter.

Offensichtlich ist die Zuverlässigkeit dieses Verfahrens hauptsächlich von der Kanalqualität und der Güte des verwendeten Codes abhängig. Gegenüber einem ARQ- bietet ein FEC-System die Vorteile, weder einen Rückkopplungskanal noch einen Speicher zu benötigen und eine konstante Durchsatzrate aufzuweisen. Die Durchsatzrate wird jedoch sehr klein, wenn Codes mit hoher Redundanz verwendet werden, um eine geringe Fehlerwahrscheinlichkeit zu erhalten. Ein FEC-System bietet auch keine Möglichkeit, auf Veränderungen der Kanalqualität zu reagieren.

1.4 Hybrid-ARQ-Systeme

Die Vorteile von ARQ- und FEC-Systemen können in einem Hybrid-ARQ-System verbunden werden. Dieses besteht aus einem ARQ-System, dessen Übertragungskanal ein FEC-System ist. In diesem Zusammenhang wird das innere FEC-System auch als *Superkanal* bezeichnet. Entsprechend nennt man die Kombination aus ARQ- und FEC-Codierer *Supercodierer* und die aus FEC- und ARQ-Decodierer *Superdecodierer* (s. Abb. 1.2).

Hybrid-ARQ-Systeme bieten geringe Fehlerwahrscheinlichkeiten bei hohen Durchsätzen. Daher sind sie z. B. für die Satellitenkommunikation gut geeignet ([2, S. 33]).

Bei der Konstruktion von Hybrid-ARQ-Systemen ist zu beachten, daß der Superkanal andere Eigenschaften als der ursprüngliche Kanal hat. Durch falsches Korrigieren fehlerhaft übertragener Blöcke entstehen zusätzliche Fehler, die auch in Bündeln (*Error Bursts*) auftreten können. Das ARQ-System muß in der Lage sein, auch solche Fehler zu erkennen. Dies kann durch Verwendung geeigneter Codes oder durch Verflechtung geschehen.

1.4.1 Klassifizierung von Hybrid-ARQ-Systemen

Die bisher beschriebenen Hybrid-ARQ-Systeme lassen sich anhand des verwendeten FEC-Systems in verschiedene Typen aufteilen (s. [8, S. 94ff], [9], [12, S. 183ff]).

Wenn das FEC-System stets den gleichen fehlerkorrigierenden Code verwendet, spricht man von einem *Typ-I Hybrid-ARQ-System* (HI-ARQ). Dies ist besonders geeignet für Kanäle, deren Charakteristik als nahezu konstant vorausgesetzt werden kann.

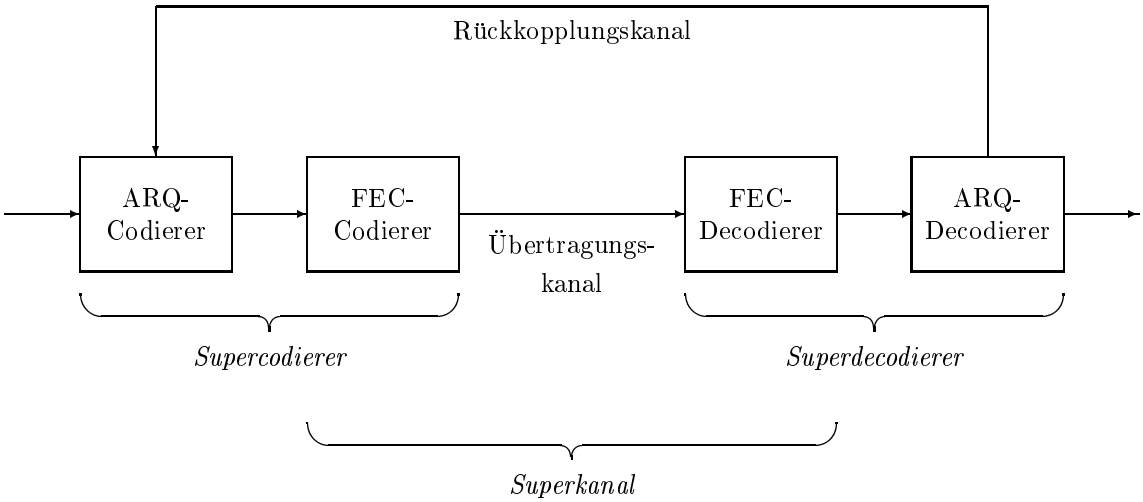

Abb. 1.2: Hybrid-ARQ-System
(Quelle: [2, S. 343], [16, S. 239])

Für Kanäle mit variierendem Störverhalten sind *Typ-II Hybrid-ARQ-Systeme* (HII-ARQ) besonders geeignet. Falls bei der Übertragung eines Blocks Fehler auftreten, wird nicht der gleiche Block erneut übertragen, sondern ein Block, der aus Redundanzbits des fehlerkorrigierenden Codes besteht. Zur Decodierung werden dann die Redundanzbits und der ursprüngliche Block verwendet. Wenn bei aufeinanderfolgenden Reübertragungen verschiedene Blöcke von Redundanzbits übertragen werden, spricht man von einem *verallgemeinerten Typ-II Hybrid-ARQ-System* (GH-ARQ = Generalized type-II Hybrid ARQ scheme). Typ-II Hybrid-ARQ-Systeme passen sich quasi der aktuellen Kanalqualität an, man nennt sie daher auch *adaptiv*.

1.4.2 Codeverkettung

Aus codierungstheoretischer Sicht können wir ein Hybrid-ARQ-System als Verkettung von Codes interpretieren. In dieser Arbeit werden wir zwei Varianten der Codeverkettung betrachten, die wir hier als „*gewöhnliche Codeverkettung*“ und „*Codeverkettung mit Partitionierung*“ bezeichnen wollen. In der Literatur wird im allgemeinen nur die Bezeichnung „Codeverkettung“ verwendet (s. z. B. [6], [7], [17, S. 380f]). Eine weitere Variante, die wir hier jedoch nicht betrachten werden, wird von Bossert ([4, S. 175ff]) „verallgemeinerte Codeverkettung“ genannt.

Gewöhnliche Codeverkettung

Es seien $\mathcal{C}_2$ ein fehlererkennender, linearer (l, k, d_2)-Code über $GF(2)$ und $\mathcal{C}_1$ ein fehlerkorrigierender, linearer (n, l, d_1)-Code über $GF(2)$. Man bezeichnet $\mathcal{C}_2$ als *äußeren* und $\mathcal{C}_1$ als *inneren* Code.

Ein binäres Informationswort der Länge k wird zuerst mittels $\mathcal{C}_2$ codiert und anschließend in den $\mathcal{C}_1$-Codierer eingegeben. Das so erhaltene Codewort des verketteten Codes ist also ein $\mathcal{C}_1$-Codewort (s. Abb. 1.3).

Die Decodierung erfolgt in der umgekehrten Reihenfolge wie die Codierung.

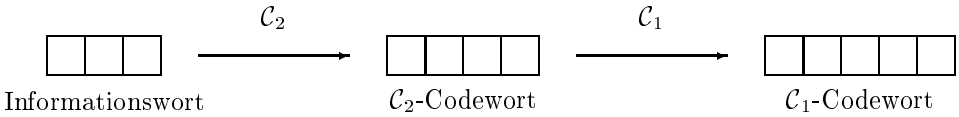

Abb. 1.3: Gewöhnliche Codeverkettung (hier: $k = 3$, $l = 4$, $n = 5$)

Das folgende Lemma gibt uns eine Charakterisierung des so definierten Codes.

Lemma 1.1 *Es seien $\mathcal{C}_2$ ein linearer (l, k, d_2)-Code über $GF(2)$ und $\mathcal{C}_1$ ein linearer (n, l, d_1)-Code über $GF(2)$. Für den durch gewöhnliche Codeverkettung von $\mathcal{C}_1$ und $\mathcal{C}_2$ definierten linearen (n, k, d)-Code $\mathcal{C}$ über $GF(2)$ gilt:*

a) $d \geq d_1$.

b) Falls $\mathbf{G}_1$ und $\mathbf{G}_2$ Generatormatrizen von $\mathcal{C}_1$ und $\mathcal{C}_2$ sind, ist $\mathbf{G} := \mathbf{G}_2\mathbf{G}_1$ eine Generatormatrix von $\mathcal{C}$.

Beweis: Zu (a): $\mathcal{C}$ ist ein Teilcode von $\mathcal{C}_1$, daher gilt $d \geq d_1$.

Zu (b): Es seien $\bar{a} \in (GF(2))^k$ ein Informationswort und $\bar{c}$ das aus $\bar{a}$ gebildete $\mathcal{C}$-Codewort, dann gilt $\bar{c} = \bar{a}\mathbf{G}_2\mathbf{G}_1$.
Insbesondere folgt, daß $\mathcal{C}$ ein linearer Code ist. q.e.d.

Das Prinzip der gewöhnlichen Codeverkettung findet sowohl in Typ-I- als auch in Typ-II Hybrid-ARQ-Systemen Anwendung.

Codeverkettung mit Partitionierung

Es seien $\mathcal{C}_2$ ein fehlererkennender, linearer (n_2, k_2, d_2)-Code über $GF(2^{k_1})$ und $\mathcal{C}_1$ ein fehlerkorrigierender, linearer (n_1, k_1, d_1)-Code über $GF(2)$. Man bezeichnet wieder $\mathcal{C}_2$ als *äußeren* und $\mathcal{C}_1$ als *inneren* Code.

Ein binäres Informationswort $\bar{a}$ der Länge $k_1 k_2$ kann als Wort der Länge k_2 mit Symbolen aus $GF(2^{k_1})$ betrachtet werden: $\bar{a} = (\bar{a}_1, \ldots, \bar{a}_{k_2})$ mit $\bar{a}_i \in GF(2^{k_1})$, $i = 1, \ldots, k_2$.

Der $\mathcal{C}_2$-Codierer erstellt aus $\bar{a}$ ein Codewort $\bar{c}$ der Länge n_2 über $GF(2^{k_1})$: $\bar{c} = (\bar{c}_1, \ldots, \bar{c}_{n_2})$ mit $\bar{c}_i \in GF(2^{k_1})$, $i = 1, \ldots, n_2$.

Die k_1-Tupel $\bar{c}_1, \ldots, \bar{c}_{n_2}$ bezeichnet man auch als *Segmente* ([6]). Sie werden nun sukzessive in den $\mathcal{C}_1$-Codierer eingegeben. Auf diese Weise erhält man n_2 Codewörter $\bar{d}_1, \ldots, \bar{d}_{n_2}$, die sogenannten *Rahmen* ([6]). Diese können zu einem Wort $\bar{d} = (\bar{d}_1, \ldots, \bar{d}_{n_2})$ der Länge $n_1 n_2$, dem sogenannten *Block* ([6]), zusammengesetzt werden (s. Abb. 1.4).

Die Decodierung erfolgt in der umgekehrten Reihenfolge wie die Codierung.

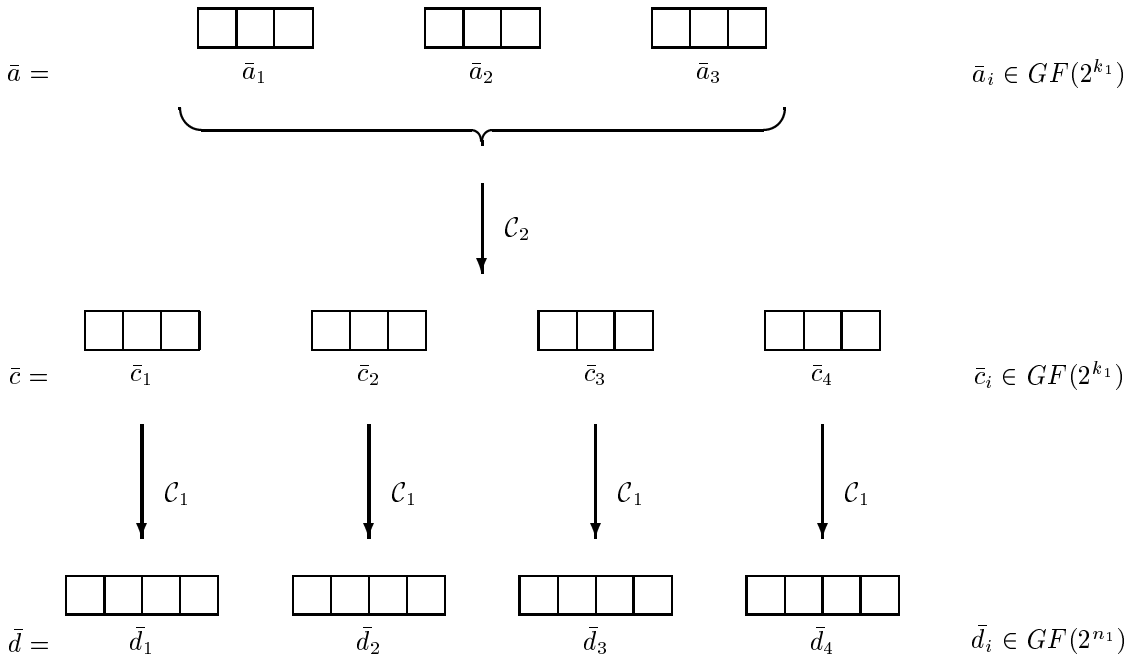

Abb. 1.4: Codeverkettung mit Partitionierung (hier: $k_1 = k_2 = 3$, $n_1 = n_2 = 4$)
(Quelle: [17, S. 381])

Das folgende Lemma gibt uns eine Charakterisierung des so definierten Codes (s. [17, S. 381] für die Aussagen (a) und (b)).

Lemma 1.2 *Es seien $\mathcal{C}_2$ ein linearer (n_2, k_2, d_2)-Code über $GF(2^{k_1})$ und $\mathcal{C}_1$ ein linearer (n_1, k_1, d_1)-Code über $GF(2)$. Für den durch Codeverkettung von $\mathcal{C}_1$ und $\mathcal{C}_2$ mit Partitionierung definierten linearen (n, k, d)-Code $\mathcal{C}$ über $GF(2)$ gilt:*

a) $n = n_1 n_2$, $k = k_1 k_2$,

b) $d \geq d_1 d_2$,

c) Falls $\mathbf{G}_1$ *und* $\mathbf{G}_2$ *Generatormatrizen von* $\mathcal{C}_1$ *und* $\mathcal{C}_2$ *sind, ist*

$$\mathbf{G} := \mathbf{G}_2 \cdot (\mathbf{I}_{n_2} \otimes \mathbf{G}_1) = \mathbf{G}_2 \cdot \begin{pmatrix} \mathbf{G}_1 & \mathbf{O} & \ldots\ldots & \mathbf{O} \\ \mathbf{O} & \mathbf{G}_1 & \mathbf{O} \cdots & \mathbf{O} \\ \ldots\ldots\ldots\ldots\ldots\ldots & & & \\ \mathbf{O} & \ldots\ldots & \mathbf{O} & \mathbf{G}_1 \end{pmatrix} ,$$

wobei $\mathbf{I}_{n_2}$ *die* $(n_2 \times n_2)$*-Einheitsmatrix,* $\mathbf{O}$ *die* $(k_1 \times n_1)$*-Nullmatrix und* $\otimes$ *das Kronecker–Produkt bezeichnen, eine Generatormatrix von* $\mathcal{C}$*.*

Beweis: Zu (a): Die Aussage folgt unmittelbar aus der obigen Beschreibung der Codeverkettung mit Partitionierung.

Zu (b): Es sei $\bar{a} \neq \bar{0}$, wobei $\bar{0}$ den Nullvektor (bzw. das Nullwort) bezeichne.
Das $\mathcal{C}_2$-Codewort $\bar{c}$ enthält dann mindestens d_2 Symbole $\bar{c}_{i_1}, \ldots, \bar{c}_{i_\delta}$ $(\delta \geq d_2)$ aus $GF(2^{k_1})$ ungleich 0 . Daher enthält jedes $\mathcal{C}_1$-Codewort $\bar{d}_{i_\nu}$, $\nu = 1, \ldots, \delta$ wenigstens d_1 Einsen.
Also hat $\bar{d} = (\bar{d}_1, \ldots, \bar{d}_{n_2})$ mindestens das Hamming-Gewicht $d_1 d_2$.

Zu (c): Aus obiger Beschreibung der Codeverkettung mit Partitionierung folgt:
$\bar{c} = \bar{a}\mathbf{G}_2$ und $\bar{d}_i = \bar{c}_i \mathbf{G}_1$, $i = 1, \ldots, n_2$.
Also gilt: $\bar{d} = (\bar{d}_1, \ldots, \bar{d}_{n_2}) = (\bar{c}_1 \mathbf{G}_1, \ldots, \bar{c}_{n_2} \mathbf{G}_1) = \bar{c}(\mathbf{I}_{n_2} \otimes \mathbf{G}_1) = \bar{a}\mathbf{G}_2(\mathbf{I}_{n_2} \otimes \mathbf{G}_1)$.
Insbesondere folgt, daß $\mathcal{C}$ ein linearer Code ist. q.e.d.

In dieser Arbeit werden wir das Prinzip der Codeverkettung mit Partitionierung nicht verwenden. Wir werden uns aber einer Analysemethode bedienen, die von Kasami e. a. für ein Typ-I Hybrid-ARQ-System mit Codeverkettung mit Partitionierung vorgeschlagen wurde ([6]).

Kapitel 2

Spezielle Codes

In diesem Kapitel werden wir einige spezielle Codes betrachten, die in ARQ- und Hybrid-ARQ-Systemen Verwendung finden. Wir beschränken uns hier auf Reed–Solomon–Codes und polynomiale Codes für die Fehlererkennung sowie BCH–, Hamming– und KM–Codes für die Fehlerkorrektur. Diese Auswahl beruht im wesentlichen auf den Empfehlungen von Ananasso e. a. ([1]), Bhargava e. a. ([2, S. 474ff]), Krishna und Morgera ([8, S. 104ff], [12, S. 188ff]), Nussbaumer ([13]) sowie Wu e. a. ([20]).

Zur Vorbereitung beschäftigen wir uns zunächst kurz mit der Theorie der zyklischen Codes.

2.1 Zyklische Codes

In diesem Abschnitt werden wir stets lineare Codes der Länge n über $GF(2^h)$ betrachten, wobei n eine ungerade natürliche Zahl sei. Eine Verallgemeinerung auf lineare Codes über $GF(q)$ ist problemlos möglich, falls n und q relativ prim sind (s. [10, S. 75ff], [17, S. 320ff]).

Definition 2.1 (Zyklische Codes) *Ein linearer Code $\mathcal{C}$ der Länge n über $GF(2^h)$ heißt* zyklischer Code, *falls aus $(c_1, c_2, \ldots, c_n) \in \mathcal{C}$ stets $(c_n, c_1, \ldots, c_{n-1}) \in \mathcal{C}$ folgt.*

Im folgenden bezeichne $GF(2^h)[x]$ den Ring der Polynome über $GF(2^h)$. Bekanntlich kann ein Wort $\bar{c} = (c_0, c_1, \ldots, c_{n-1}) \in (GF(2^h))^n$ als Polynom vom Grad $n-1$ in $GF(2^h)[x]$ interpretiert werden. Formal geschieht dies mittels einer monomorphen Abbildung

$$\begin{array}{rccc} \phi : & (GF(2^h))^n & \longrightarrow & GF(2^h)[x] \\ & (c_0, c_1, \ldots, c_{n-1}) & \longmapsto & \displaystyle\sum_{i=0}^{n-1} c_i x^i \ . \end{array}$$

Daher können wir einen linearen Code $\mathcal{C}$ der Länge n über $GF(2^h)$ als Teilmenge von

$$R_n \quad := \quad \frac{GF(2^h)[x]}{x^n - 1}$$

betrachten.

Satz 2.2 *Ein linearer Code $\mathcal{C}$ der Länge n über $GF(2^h)$ ist genau dann zyklisch, wenn $\mathcal{C}$ ein Ideal in R_n ist.*

Quelle: van Lint ([10, S. 76]).

Wir werden sehen, daß jedes Ideal in R_n ein Hauptideal ist, also von einem Polynom $g(x) \in R_n$ erzeugt wird. Es bezeichne $\langle g(x) \rangle$ das von $g(x)$ erzeugte Ideal in R_n, d. h.

$$\langle g(x) \rangle \quad := \quad \{f(x)g(x) \bmod (x^n - 1) \ : \ f(x) \in R_n\} \ .$$

Satz 2.3 *Es sei $\mathcal{C}$ ein zyklischer Code der Länge n über $GF(2^h)$, dann gilt:*

a) Es gibt ein eindeutig bestimmtes Polynom $g(x)$ mit Führungskoeffizient 1 und minimalem Grad in $\mathcal{C}$. Dieses Polynom $g(x)$ erzeugt $\mathcal{C}$ und heißt Generatorpolynom *von $\mathcal{C}$.*

b) Das Generatorpolynom $g(x)$ teilt $x^n - 1$.

c) Falls $g(x)$ den Grad $n - k$ hat, dann hat $\mathcal{C}$ die Dimension k und es gilt

$$\mathcal{C} \ = \ \langle g(x) \rangle \ = \ \{c(x) = a(x)g(x) \ : \ \deg(a(x)) < k\} \ .$$

d) Falls $g(x) = \sum_{i=0}^{n-k} g_i x^i$ ist, dann ist $g_0 \neq 0$ und $\mathcal{C}$ besitzt die Generatormatrix

$$\mathbf{G} := \begin{pmatrix} g_0 & g_1 & \cdots\cdots & g_{n-k} & 0 & \cdots\cdots\cdots & 0 \\ 0 & g_0 & g_1 & \cdots\cdots\cdots & g_{n-k} & 0 & \cdots\cdots\cdots \\ 0 & 0 & g_0 & g_1 & \cdots\cdots\cdots & g_{n-k} & 0 & \cdots \\ \vdots & & \ddots & \ddots & \ddots & & \ddots & \ddots \\ 0 & \cdots\cdots & 0 & g_0 & g_1 & \cdots\cdots\cdots & g_{n-k} \end{pmatrix} \in (GF(2^h))^{k \times n} \ .$$

Quelle: Roman ([17, S. 321f]).

Um zyklische Codes konstruieren zu können, werden wir das folgende Ergebnis benötigen:

Satz 2.4 *Ein Polynom $p(x) \in R_n$ mit Führungskoeffizient 1 ist genau dann das Generatorpolynom eines zyklischen Codes der Länge n über $GF(2^h)$, wenn $p(x)$ das Polynom $x^n - 1$ teilt.*

Quelle: Roman ([17, S. 323]).

Für das Generatorpolynom $g(x)$ eines zyklischen Codes $\mathcal{C}$ gilt also $x^n - 1 = g(x)h(x)$ in $GF(2^h)[x]$. Das Polynom $h(x)$ hat den Grad $k = n - \deg(g(x))$ und wird als *Kontrollpolynom* (oder *Prüfpolynom*) von $\mathcal{C}$ bezeichnet. Die Begründung hierfür liefert der nächste Satz.

Satz 2.5 *Es sei $h(x)$ das Kontrollpolynom eines zyklischen Codes $\mathcal{C}$ der Länge n und Dimension k über $GF(2^h)$, dann gilt:*

a) $\mathcal{C} = \{c(x) \in R_n : c(x)h(x) \equiv 0 \bmod (x^n - 1)\}$.

b) Falls $h(x) = \sum_{i=0}^{k} h_i x^i$ *ist, dann ist*

$$\mathbf{H} := \begin{pmatrix} h_k & h_{k-1} & \ldots\ldots\ldots & h_0 & 0 & \ldots\ldots & 0 \\ 0 & h_k & h_{k-1} & \ldots\ldots\ldots & h_0 & 0 & \ldots\ldots \\ 0 & 0 & h_k & h_{k-1} & \ldots\ldots\ldots & h_0 & 0 & \ldots \\ \vdots & & \ddots & \ddots & \ddots & & \ddots & \ddots \\ 0 & \ldots\ldots\ldots & 0 & h_k & h_{k-1} & \ldots\ldots & h_0 \end{pmatrix} \in (GF(2^h))^{(n-k)\times n}$$

eine Kontrollmatrix von $\mathcal{C}$.

c) Der zu $\mathcal{C}$ duale Code $\mathcal{C}^\perp$ ist ein zyklischer Code der Dimension $n-k$ mit Generatorpolynom

$$h^\perp(x) := h_0^{-1} x^k h(x^{-1}) = h_0^{-1}(h_0 x^k + h_1 x^{k-1} + \ldots + h_k) .$$

Das Polynom $x^k h(x^{-1})$ ist das zu $h(x)$ reziproke Polynom[1]*.*

Quelle: Roman ([17, S. 325]).

Wir können zyklische Codes auch mit Hilfe der Nullstellen des Generatorpolynoms beschreiben. Nach Satz 2.4 ist jede Nullstelle des Generatorpolynoms auch eine Nullstelle von $x^n - 1$, also eine n-te Einheitswurzel.

Satz 2.6 *Es sei $g(x) = \prod_{i=1}^{t} q_i(x) \in R_n$ ein Produkt irreduzibler Faktoren $q_1(x), \ldots, q_t(x)$ von $x^n - 1$, und $\alpha_1, \ldots, \alpha_u$ seien die Wurzeln von $g(x)$ im Zerfällungskörper von $x^n - 1$ über $GF(2^h)$, dann gilt:*

$$\langle g(x) \rangle = \{f(x) \in R_n : f(\alpha_i) = 0 \quad \forall\, i \in \{1, \ldots, u\}\} .$$

Es ist sogar ausreichend, nur eine Wurzel jedes irreduziblen Faktors von $g(x)$ zu betrachten: Für $i = 1, \ldots, t$ sei β_i eine Wurzel von $q_i(x)$, dann ist

$$\langle g(x) \rangle = \{f(x) \in R_n : f(\beta_i) = 0 \quad \forall\, i \in \{1, \ldots, t\}\} .$$

Quelle: Roman ([17, S. 327]).

Definition 2.7 *Die Wurzeln des Generatorpolynoms eines zyklischen Codes heißen die* Nullen *(oder* Wurzeln*) des Codes, alle anderen n-ten Einheitswurzeln werden* Nichtnullen *des Codes genannt.*

[1] Es sei $p(x)$ ein Polynom vom Grad d. Das Polynom $\hat{p}(x) := x^d p(x^{-1})$ heißt *zu $p(x)$ reziprokes Polynom*. Falls $p(x) = \sum_{i=0}^{d} p_i x^i$ ist, gilt $\hat{p}(x) = \sum_{i=0}^{d} p_{d-i} x^i$ (s. [17, S. 307]).

Bemerkung: Die Kenntnis der Nullen eines zyklischen Codes ermöglicht die Konstruktion des Generatorpolynoms und einer Kontrollmatrix.

Falls $\alpha_1, \ldots, \alpha_u$ Wurzeln von $x^n - 1$ sind, ist das Generatorpolynom des Codes

$$\mathcal{C} = \{ f(x) \in R_n \ : \ f(\alpha_i) = 0 \quad \forall\, i \in \{1, \ldots, u\} \}$$

das kleinste gemeinsame Vielfache der Minimalpolynome von $\alpha_1, \ldots, \alpha_u$.

Wenn wir annehmen, daß $\alpha_1, \ldots, \alpha_u$ in dem Erweiterungskörper $GF((2^h)^d)$ liegen, können wir jede der Potenzen α_i^j $(i = 1, \ldots, u,\ j = 0, \ldots, n-1)$ als Spaltenvektor $[\alpha_i^j]$ der Länge d über $GF(2^h)$ schreiben. Es sei $f(x) = \sum_{j=0}^{n-1} f_j x^j$ ein Polynom in R_n, dann ist

$$f(\alpha_i) \ = \ \sum_{j=0}^{n-1} f_j [\alpha_i^j] \ = \ \left[\sum_{j=0}^{n-1} f_j \alpha_i^j \right] \ , \quad i = 1, \ldots, u \ .$$

Deshalb gilt $f(\alpha_i) = 0$ für alle $i = 1, \ldots, u$ genau dann, wenn $\bar{f}\mathbf{H}^T = \bar{0}$ ist, wobei $\bar{f} := (f_0, \ldots, f_{n-1})$ sei und

$$\mathbf{H} \ := \ \begin{pmatrix} [1] & [\alpha_1] & [\alpha_1^2] & \ldots & [\alpha_1^{n-1}] \\ [1] & [\alpha_2] & [\alpha_2^2] & \ldots & [\alpha_2^{n-1}] \\ \vdots & \vdots & \vdots & & \vdots \\ [1] & [\alpha_u] & [\alpha_u^2] & \ldots & [\alpha_u^{n-1}] \end{pmatrix} .$$

Wenn wir alle linear abhängigen Zeilen von $\mathbf{H}$ streichen, erhalten wir eine Kontrollmatrix des Codes $\mathcal{C}$.

Quelle: Roman ([17, S.327f]).

Aufbauend auf diesen Grundlagen können wir nun für unsere Anwendungen interessante zyklische Codes betrachten.

2.2 BCH–Codes

Eine wichtige Klasse zyklischer Codes wurde unabhängig voneinander 1960 von R. C. Bose und D. K. Ray-Chaudhuri und 1959 von A. Hocquenghem entdeckt. Diese Codes werden deshalb *BCH–Codes* genannt. Ihre große Bedeutung rührt daher, daß sie alle Fehlermuster bis zu einer gewissen Fehlerzahl korrigieren können. Außerdem ist der Decodieralgorithmus einfach und mit einem vertretbaren Materialaufwand leicht zu implementieren (s. [16, S. 167]).

Wir werden in diesem Abschnitt wieder Codes der Länge n über $GF(2^h)$ betrachten, wobei n ungerade sei. Eine Verallgemeinerung auf Codes über $GF(q)$ ist möglich, falls n und q relativ prim sind (s. z. B. [17, S. 353ff]).

Definition 2.8 (BCH–Codes) *Es seien ω eine primitive n-te Einheitswurzel*[2] *über $GF(2^h)$ und $b \geq 0$ sowie $1 \leq \delta \leq n$. Ferner sei $g(x) \in GF(2^h)[x]$ das Polynom mit Führungskoeffizient 1 und minimalem Grad, so daß*

$$g(\omega^b) \;=\; g(\omega^{b+1}) \;=\; \ldots \;=\; g(\omega^{b+\delta-2}) \;=\; 0$$

gilt. Der zyklische Code $\mathcal{B}_h(n, \delta, \omega, b)$ der Länge n über $GF(2^h)$ mit Generatorpolynom $g(x)$ heißt BCH–Code mit Designabstand δ.
Falls $b = 1$ ist, heißt der Code $\mathcal{B}_h(n, \delta, \omega) := \mathcal{B}_h(n, \delta, \omega, 1)$ BCH–Code im engeren Sinne.
Falls ω ein primitives Element von $GF((2^h)^s)$ für ein $s \geq 1$ ist, also $n = (2^h)^s - 1$ gilt, heißt $\mathcal{B}_h(n, \delta, \omega, b)$ primitiver BCH–Code.

Quelle: Roman ([17, S. 354f]).

Bemerkungen:

a) BCH–Codes sind lineare Codes.

b) Das Generatorpolynom $g(x)$ ist das kleinste gemeinsame Vielfache der Minimalpolynome von $\omega^b, \omega^{b+1}, \ldots, \omega^{b+\delta-2}$. Nach Satz 2.4 ist $\mathcal{B}_h(n, \delta, \omega, b)$ ein zyklischer Code.

c) Aus der Definition der BCH–Codes und Satz 2.6 folgt

$$\mathcal{B}_h(n, \delta, \omega, b) \;=\; \{c(x) \in R_n \;:\; c(\omega^b) = c(\omega^{b+1}) = \ldots = c(\omega^{b+\delta-2}) = 0\}\;.$$

Der Begriff „Designabstand“ suggeriert, daß es sich hierbei auch um den Minimalabstand handelt. Der folgende Satz zeigt jedoch, daß der Designabstand nur eine untere Schranke für den Minimalabstand darstellt (s. [10, S. 85f]):

Satz 2.9 (BCH–Schranke) *Der Minimalabstand eines BCH–Codes $\mathcal{B}_h(n, \delta, \omega, b)$ beträgt mindestens δ.*

Beweis: Wie wir in der Bemerkung zu Definition 2.7 gesehen haben, erhalten wir aus der Matrix

$$\mathbf{H} \;:=\; \begin{pmatrix} [1] & [\omega^b] & [\omega^{2b}] & \ldots & [\omega^{(n-1)b}] \\ [1] & [\omega^{b+1}] & [\omega^{2(b+1)}] & \ldots & [\omega^{(n-1)(b+1)}] \\ \vdots & \vdots & \vdots & & \vdots \\ [1] & [\omega^{b+\delta-2}] & [\omega^{2(b+\delta-2)}] & \ldots & [\omega^{(n-1)(b+\delta-2)}] \end{pmatrix}$$

durch Streichen der linear abhängigen Zeilen eine Kontrollmatrix von $\mathcal{B}_h(n, \delta, \omega, b)$. Wir betrachten nun $\delta - 1$ beliebige Spalten von $\mathbf{H}$, deren oberste Einträge $\omega^{i_1 b}, \omega^{i_2 b}, \ldots, \omega^{i_{\delta-1} b}$ seien. Diese

[2] Eine n-te Einheitswurzel der Ordnung n heißt *primitive n-te Einheitswurzel*, d. h. es gilt $\omega^n = 1$ und $\omega^m \neq 1$ für $0 < m < n$ (s. [10, S. 9], [17, S. 308]).

Spalten definieren eine quadratische Teilmatrix $\mathbf{H}'$ von $\mathbf{H}$, für deren Determinante gilt:

$$\begin{aligned}
\det \mathbf{H}' &= \begin{vmatrix} \omega^{i_1 b} & \omega^{i_2 b} & \dots & \omega^{i_{\delta-1} b} \\ \omega^{i_1 (b+1)} & \omega^{i_2 (b+1)} & \dots & \omega^{i_{\delta-1} (b+1)} \\ \vdots & \vdots & & \vdots \\ \omega^{i_1 (b+\delta-2)} & \omega^{i_2 (b+\delta-2)} & \dots & \omega^{i_{\delta-1} (b+\delta-2)} \end{vmatrix} \\
&= \omega^{(i_1+i_2+\dots+i_{\delta-1})b} \cdot \begin{vmatrix} 1 & 1 & \dots & 1 \\ \omega^{i_1} & \omega^{i_2} & \dots & \omega^{i_{\delta-1}} \\ \vdots & \vdots & & \vdots \\ \omega^{i_1 (\delta-2)} & \omega^{i_2 (\delta-2)} & \dots & \omega^{i_{\delta-1} (\delta-2)} \end{vmatrix} \\
&= \omega^{(i_1+i_2+\dots+i_{\delta-1})b} \cdot \prod_{1 \le \nu < \mu \le \delta-1} (\omega^{i_\mu} - \omega^{i_\nu})
\end{aligned}$$

(Vandermonde'sche Matrix, s. [5, S. 180]).

Da ω eine primitive n-te Einheitswurzel ist, folgt $\det \mathbf{H}' \neq 0$. Also sind jeweils $\delta - 1$ Spalten von $\mathbf{H}$ linear unabhängig. Dies impliziert, daß der Minimalabstand von $\mathcal{B}_h(n, \delta, \omega, b)$ mindestens δ beträgt (s. [5, S. 227], [17, S. 201]). q.e.d.

Es gibt eine Vielzahl weiterer Ergebnisse über den Minimalabstand eines BCH–Codes, eine Auswahl findet man z. B. in den Büchern von MacWilliams e. a. ([11, S. 259ff]) und Roman ([17, S. 360ff]).

Eine ebenfalls wichtige Größe ist die Dimension eines BCH–Codes. Der folgende Satz gibt uns eine Abschätzung hierfür (s. [17, S. 355]).

Satz 2.10 *Es sei s die Ordnung von 2^h modulo n, d. h. $s = \min\{t \in \mathbb{N} : (2^h)^t \equiv 1 \bmod n\}$, dann gilt für die Dimension eines BCH–Codes $\mathcal{B}_h(n, \delta, \omega, b)$:*

$$\dim\left(\mathcal{B}_h(n, \delta, \omega, b)\right) \ \ge \ n - s(\delta - 1) \, .$$

Beweis: Der Zerfällungskörper von $x^n - 1$ hat den Grad s über $GF(2^h)$ (s. [17, S. 309]), folglich hat für $i \in \{0, \dots, \delta - 2\}$ das Minimalpolynom $m_i(x)$ von ω^{b+i} über $GF(2^h)$ höchstens den Grad s. Das impliziert, daß das Generatorpolynom $g(x) = \mathrm{lcm}\{m_0(x), m_1(x), \dots, m_{\delta-2}(x)\}$ von $\mathcal{B}_h(n, \delta, \omega, b)$ höchstens den Grad $(\delta - 1)s$ hat. Also gilt für den BCH–Code $\mathcal{B}_h(n, \delta, \omega, b)$:
$\dim\left(\mathcal{B}_h(n, \delta, \omega, b)\right) \ = \ n - \deg(g(x)) \ \ge \ n - (\delta - 1)s \, .$ q.e.d.

Weitere Ergebnisse über die Dimension eines BCH–Codes findet man z. B. in dem Buch von MacWilliams e. a. ([11, S. 262ff]).

2.2.1 Binäre BCH–Codes

Wenn wir BCH–Codes als innere Codes eines Hybrid-ARQ-Systems verwenden wollen, werden wir im allgemeinen binäre BCH–Codes im engeren Sinne wählen. Um deren Generatorpolynome zu konstruieren, benötigen wir nur die Minimalpolynome der ungeradzahligen Potenzen der primitiven n-ten Einheitswurzel ω (s. [17, S. 356]):

Satz 2.11 *Es sei $g(x)$ das Generatorpolynom eines binären BCH–Codes im engeren Sinne $\mathcal{B}_1(n,\delta,\omega)$. Ferner sei $m_i(x)$ das Minimalpolynom über $GF(2)$ von ω^i, $i = 1,\ldots,\delta-1$. Dann gilt:*

$$g(x) \;=\; \begin{cases} \operatorname{lcm}\{m_1(x), m_3(x), m_5(x), \ldots, m_{\delta-2}(x)\} & \textit{falls } \delta \textit{ ungerade ist,} \\ \operatorname{lcm}\{m_1(x), m_3(x), m_5(x), \ldots, m_{\delta-1}(x)\} & \textit{falls } \delta \textit{ gerade ist.} \end{cases}$$

Beweis: Für $i = 1,\ldots,\delta-1$ gilt (s. [17, S. 298f]):

$$m_i(x) \;=\; \prod_{\nu=0}^{t-1}(x-\omega^{2^\nu i}) \;=\; (x-\omega^i)(x-\omega^{2i})(x-\omega^{4i})\cdot\ldots\cdot(x-\omega^{2^{t-1}i})\;,$$

wobei t die kleinste natürliche Zahl sei, für die $\omega^{2^t i} = \omega^i$, also $2^t \equiv 1 \bmod n$, gilt. Dies impliziert für $i = 1,\ldots,\lfloor\frac{\delta-1}{2}\rfloor$:

$$m_{2i}(x) \;=\; \prod_{\nu=0}^{t-1}(x-\omega^{2^\nu\cdot 2i}) \;=\; \prod_{\nu=1}^{t}(x-\omega^{2^\nu i}) \;=\; \prod_{\nu=0}^{t-1}(x-\omega^{2^\nu i}) \;=\; m_i(x)\;.$$

Aus $g(x) = \operatorname{lcm}\{m_1(x), m_2(x), \ldots, m_{\delta-1}(x)\}$ folgt die Behauptung. q.e.d.

Bei der Betrachtung binärer Codes im engeren Sinne können wir uns also auf ungerade Designabstände konzentrieren. Der obige Satz ermöglicht auch eine genauere Abschätzung für die Dimension dieser Codes (s. [17, S. 356]):

Korollar 2.12 *Für einen binären BCH–Code im engeren Sinne $\mathcal{B}_1(n,\delta,\omega)$ mit ungeradem Designabstand $\delta = 2\gamma+1$ gilt:*

$$\dim\left(\mathcal{B}_1(n,\delta,\omega)\right) \;\geq\; n - s\gamma\;,$$

wobei s die Ordnung von 2 modulo n sei, d. h. $s = \min\{t \in \mathbb{N} \;:\; 2^t \equiv 1 \bmod n\}$.

Beweis: Die Aussage folgt aus Satz 2.11 und dem Beweis zu Satz 2.10. □

2.2.2 Codierung von BCH–Codes

Für die Codierung von BCH–Codes (oder allgemein von zyklischen Codes) gibt es im wesentlichen zwei Verfahren, ein systematisches und ein nichtsystematisches (s. [17, S. 344f]).

Es sei $\mathcal{B}_h(n,\delta,\omega,b)$ ein BCH–Code der Dimension k mit Generatorpolynom $g(x)$.

Nichtsystematische Codierung

Aus einem Informationswort $(a_0, a_1, \ldots, a_{k-1})$ wird das Polynom $a(x) = \sum_{i=0}^{k-1} a_i x^i$ gebildet. Die Multiplikation mit dem Generatorpolynom $g(x)$ liefert das Codepolynom
$c(x) = a(x)g(x) \in \mathcal{B}_h(n,\delta,\omega,b)$.

Systematische Codierung

Wie bei der nichtsystematischen Codierung wird aus einem Informationswort $(a_0, a_1, \ldots, a_{k-1})$ ein Polynom gebildet, diesmal jedoch

$$\tilde{a}(x) \;:=\; a_0 x^{n-1} + a_1 x^{n-2} + \ldots + a_{k-1} x^{n-k} \;.$$

Dieses wird durch das Generatorpolynom $g(x)$ dividiert: $\tilde{a}(x) \;=\; q(x)g(x) + r(x)$, wobei $\deg(r(x)) < n-k$ ist. Das Codepolynom ist dann

$$c(x) \;:=\; \tilde{a}(x) - r(x) \;=\; q(x)g(x) \in \mathcal{B}_h(n, \delta, \omega, b) \;.$$

Weil die Polynome $\tilde{a}(x)$ und $r(x)$ keine Terme mit gleichem Grad haben, ist die Codierung systematisch. Wenn wir die Koeffizienten von $c(x)$ vom höchsten zum niedrigsten Grad lesen (also $c(x) = c_{n-1}x^{n-1} + c_{n-2}x^{n-2} + \ldots + c_0$), sind die ersten k Positionen die Informationssymbole und die übrigen $n-k$ sind Kontrollsymbole.

Darstellungen der Schieberegister–Schaltungen für dieses Codierverfahren findet man z. B. in den Büchern von Bhargava e. a. ([2, S. 462ff]) und Rhee ([16, S. 100ff]).

2.2.3 Decodierung von BCH–Codes

Es gibt verschiedene Verfahren für die praktische Realisierung von BCH–Decodierern, eine Übersicht enthält z. B. das Buch von Rhee ([16, S. 178–219]). Wu e. a. formulieren in ihrer Arbeit einige Kriterien für die Wahl eines für eine spezielle Anwendung geeigneten Decodierers ([20, S. 733]). Sie nennen unter anderem die Abwägung von Durchsatz oder Fehlerwahrscheinlichkeit versus technischem Aufwand sowie von Komplexität des Codierers versus Komplexität des Decodierers.

Die allen Verfahren zugrundeliegende Theorie werden wir im folgenden kurz erläutern (s. [5, S. 307ff], [10, S. 92f], [17, S. 363ff]). Wir betrachten einen BCH–Code im engeren Sinne $\mathcal{B}_h(n, \delta, \omega)$ mit ungeradem Designabstand $\delta = 2t+1$ und einer primitiven n-ten Einheitswurzel $\omega \in GF((2^h)^m)$. Eine Verallgemeinerung auf BCH–Codes im weiteren Sinne, also $\mathcal{B}_h(n, \delta, \omega, b)$ mit $b > 1$, ist möglich.

Angenommen, das Codepolynom $c(x)$ wurde gesendet, und empfangen wurde das Polynom $r(x) = c(x) + e(x)$, wobei $e(x) = \sum_{i=0}^{n-1} e_i x^i$ ein Fehlermuster darstelle. Es seien $E := \{i \,:\, e_i \neq 0\}$ die Menge der fehlerhaften Positionen und $\vartheta := \sharp E$ die Anzahl der Fehler.
Das *Fehlerstellenpolynom* (oder auch *Lokatorpolynom*) $\ell(x)$ sei definiert durch

$$\ell(x) \;:=\; \prod_{j \in E} (1 - \omega^j x) \;.$$

Ferner sei das *Begleitpolynom* $w(x)$ definiert durch

$$w(x) \;:=\; \sum_{j \in E} e_j \, \omega^j x \prod_{l \in E \setminus \{j\}} (1 - \omega^l x) \;.$$

Wir nehmen an, daß höchstens $t := \lfloor \frac{\delta-1}{2} \rfloor$ Fehler[3] aufgetreten sind, also $\vartheta \leq t = \lfloor \frac{\delta-1}{2} \rfloor$ ist. Anderenfalls erheben wir keinen Anspruch, die Fehler korrigieren zu können. Folglich ist $\mathcal{B}_h(n, \delta, \omega)$ ein t-fehlerkorrigierender Code, unabhängig von dem wirklichen Minimalabstand.

Die Kenntnis der Polynome $\ell(x)$ und $w(x)$ ermöglicht uns die Korrektur aller $\vartheta \leq t$ Fehler. Falls $\mathcal{B}_h(n, \delta, \omega)$ ein binärer Code ist, also $h = 1$, genügt sogar die Kenntnis von $\ell(x)$, denn in diesem Fall gilt $e_i = 1$ für alle $i \in E$.

Satz 2.13 *Mit obigen Bezeichnungen gilt:*

a) $\ell(\omega^{-i}) = 0 \iff i \in E$,

b) $e_i = -\dfrac{w(\omega^{-i})\,\omega^i}{\ell'(\omega^{-i})}$ *für* $i \in E$, *wobei* $\ell'(x)$ *die formale Ableitung von* $\ell(x)$ *sei.*

Beweis: Zu (a): Es gelte $\ell(\omega^{-i}) = 0$, also $\omega^j \omega^{-i} = 1$ für ein $j \in E$. Dies ist jedoch nur für $j = i$ möglich, also folgt $i \in E$.
Es sei nun $i \in E$, dann gilt offensichtlich $\ell(\omega^{-i}) = 0$.

Zu (b): Es sei $i \in E$, dann folgt

$$w(\omega^{-i}) \;=\; \sum_{j \in E} e_j\, \omega^j \omega^{-i} \prod_{l \in E \setminus \{j\}} (1 - \omega^l \omega^{-i}) \;=\; e_i \prod_{l \in E \setminus \{i\}} (1 - \omega^l \omega^{-i})\,.$$

Nach der Produktregel für die formale Ableitung gilt ferner:

$$\ell'(x) \;=\; \sum_{j \in E} (-\omega^j) \prod_{l \in E \setminus \{j\}} (1 - \omega^l x)\,,$$

also

$$\ell'(\omega^{-i}) \;=\; -\sum_{j \in E} \omega^j \prod_{l \in E \setminus \{j\}} (1 - \omega^l \omega^{-i}) \;=\; -\omega^i \prod_{l \in E \setminus \{i\}} (1 - \omega^l \omega^{-i})\,.$$

Hieraus folgt die Aussage (b). q.e.d.

Wir müssen also die Polynome $\ell(x)$ und $w(x)$ (falls $h > 1$ ist) bestimmen. Mit den Rechenregeln für formale Potenzreihen über $GF((2^h)^m)$ erhalten wir:

$$\begin{aligned} \frac{w(x)}{\ell(x)} \;&=\; \sum_{j \in E} \frac{e_j \omega^j x}{1 - \omega^j x} \;=\; \sum_{j \in E} e_j\, \omega^j x \sum_{l=0}^{\infty} (\omega^j x)^l \\ &=\; \sum_{l=1}^{\infty} x^l \sum_{j \in E} e_j\, \omega^{lj} \;=\; \sum_{l=1}^{\infty} x^l\, e(\omega^l)\,. \end{aligned}$$

Für $l = 1, \ldots, 2t$ gilt aber $e(\omega^l) = r(\omega^l) - c(\omega^l) = r(\omega^l)$, so daß wir die ersten $2t$ Koeffizienten der formalen Potenzreihe $\sum\limits_{l=1}^{\infty} x^l e(\omega^l)$ kennen. Die Größen $r(\omega^l)$, $l = 1, \ldots, 2t$ werden *Syndrome*

[3] Das Symbol $\lfloor \cdot \rfloor$ bezeichne die untere Gaußklammer.

genannt. Sie ermöglichen die Fehlererkennung, denn es gilt $r(x) \in \mathcal{B}_h(n, \delta, \omega)$ genau dann, wenn alle Syndrome gleich 0 sind (vgl. Abschnitt 2.3.3).

Wir suchen also Polynome $\ell(x)$ und $w(x)$ mit minimalem Grad (Maximum–Likelihood–Decodierung), die die Bedingungen

a) $\dfrac{w(x)}{\ell(x)} \equiv \sum\limits_{l=1}^{2t} x^l \, r(\omega^l) \bmod x^{2t+1}$ und

b) $\deg(w(x)) \leq \deg(\ell(x))$

erfüllen.

Es seien $u_l := r(\omega^l) = e(\omega^l)$ für $l = 1, \ldots, 2t$ und $\ell(x) = \sum\limits_{j=0}^{t} \sigma_j x^j$ (hierbei sei $\sigma_j = 0$ für $j = \vartheta + 1, \ldots, t$), dann folgt:

$$w(x) \equiv \left(\sum_{l=1}^{2t} u_l \, x^l \right) \left(\sum_{j=0}^{t} \sigma_j \, x^j \right) \equiv \sum_{k=1}^{2t} x^k \sum_{\substack{l \in \{1,\ldots,2t\}, j \in \{0,\ldots,t\} \\ l+j=k}} u_l \sigma_j \mod x^{2t+1} . \tag{2.1}$$

Wir wissen, daß $\sigma_0 = 1$ ist. Aus $\deg(w(x)) \leq deg(\ell(x)) = \vartheta \leq t$ ergibt sich

$$\sum_{\substack{l \in \{1,\ldots,2t\}, j \in \{0,\ldots,t\} \\ l+j=k}} u_l \sigma_j = 0 \quad \text{für } k = t+1, \ldots, 2t . \tag{2.2}$$

Dies ist ein System von t Gleichungen für die Unbekannten $\sigma_1, \ldots, \sigma_t$. Aus der Herleitung folgt, daß dieses Gleichungssystem lösbar ist.

Es sei $\tilde{\ell}(x) := \sum\limits_{j=0}^{t} \tilde{\sigma}_j x^j$ eine Lösung von (2.2) mit $\tilde{\sigma}_0 = 1$, dann gilt für $k = t+1, \ldots, 2t$:

$$0 = \sum_{\substack{l \in \{1,\ldots,2t\}, j \in \{0,\ldots,t\} \\ l+j=k}} u_l \tilde{\sigma}_j = \sum_{\substack{l \in \{1,\ldots,2t\}, j \in \{0,\ldots,t\} \\ l+j=k}} \sum_{i \in E} e_i \, \omega^{il} \tilde{\sigma}_j = \sum_{i \in E} e_i \, \omega^{ik} \, \tilde{\ell}(\omega^{-i}) . \tag{2.3}$$

Dies ist ein lineares Gleichungssystem für $e_i \tilde{\ell}(\omega^{-i})$, $i \in E$, mit Koeffizienten ω^{ik}, $i \in E$ und $k = t+1, \ldots, 2t$. Wenn wir nur die ersten ϑ Zeilen betrachten, ist die Koeffizientenmatrix dieses Teilsystems eine Vandermonde'sche Matrix (vgl. Beweis zu Satz 2.9). Das Gleichungssystem (2.3) ist also nur trivial lösbar, d. h. es gilt $e_i \tilde{\ell}(\omega^{-i}) = 0$ für alle $i \in E$, so daß $\tilde{\ell}(x)$ nach Satz 2.13(a) ein Vielfaches von $\ell(x)$ sein muß. Da beide Polynome den Grad ϑ haben und $\sigma_0 = \tilde{\sigma}_0 = 1$ gilt, ist $\tilde{\ell}(x) = \ell(x)$. Wir erhalten das Fehlerstellenpolynom $\ell(x)$ also als Lösung des linearen Gleichungssystems (2.2).

Es sei $w(x) = \sum\limits_{k=1}^{t} w_k x^k$, dann ergibt sich aus (2.1) durch Koeffizientenvergleich:

$$w_k = \sum_{\substack{l \in \{1,\ldots,2t\}, j \in \{0,\ldots,t\} \\ l+j=k}} u_l \sigma_j , \quad k = 1, \ldots, t .$$

Wir kennen jetzt also auch das Begleitpolynom $w(x)$.

Auf diese Weise können wir bis zu $t = \lfloor \frac{\delta-1}{2} \rfloor$ Fehler korrigieren. Diese Anzahl t ist nur von dem Designabstand abhängig, nicht von dem wirklichen Minimalabstand.

2.2.4 Verkürzte und erweiterte BCH–Codes

Häufig werden nur Teilcodes von BCH–Codes verwendet; hierbei benutzt man nur die Codewörter, die zu Informationsworten gehören, deren letzten Positionen 0 sind (s. [4, S. 100]).

Definition 2.14 (Verkürzte BCH–Codes) *Es sei $\mathcal{B}_h(n, \delta, \omega, b)$ ein BCH–Code der Länge n und Dimension k mit Generatorpolynom $g(x)$, also*

$$\mathcal{B}_h(n, \delta, \omega, b) \;=\; \{a(x)g(x) \;:\; \deg(a(x)) < k\} \; .$$

Für $k^* < k$ ist der verkürzte BCH–Code $\mathcal{B}_h^*(n, \delta, \omega, b)$ *definiert durch*

$$\mathcal{B}_h^*(n, \delta, \omega, b) \;:=\; \{a(x)g(x) \;:\; \deg(a(x)) < k^*\} \; .$$

Die Decodierung verkürzter BCH–Codes kann in der gleichen Weise wie die nichtverkürzter BCH–Codes erfolgen.

Außerdem besteht noch die Möglichkeit, binäre BCH-Codes zu erweitern. Dies geschieht durch das Anhängen eines zusätzlichen Paritätskontrollbits an jedes Codewort (s. [4, S. 100]).

Definition 2.15 (Erweiterte BCH–Codes) *Es sei $\mathcal{B}_1(n, \delta, \omega, b)$ ein binärer BCH–Code der Länge n. Der* erweiterte BCH–Code $\hat{\mathcal{B}}_1(n, \delta, \omega, b)$ *der Länge $n+1$ ist definiert durch*

$$\hat{\mathcal{B}}_1(n, \delta, \omega, b) \;:=\; \{(c_0, c_1, \ldots, c_n) \;:\; (c_0, \ldots, c_{n-1}) \in \mathcal{B}_1(n, \delta, \omega, b) \,,\; c_n = \sum_{i=0}^{n-1} c_i\} \; .$$

Erweiterte BCH–Codes sind zwar linear, aber nicht zyklisch (s. [4, S. 100]).

2.3 Reed–Solomon–Codes

Eine wichtige Gruppe innerhalb der BCH–Codes stellen die *Reed–Solomon–Codes (RS–Codes)* dar. Sie werden in Kommunikationssystemen, aber auch in Datenspeichern eingesetzt ([16, S. 219]). In dieser Arbeit werden wir RS–Codes als fehlererkennende Codes in ARQ- und Hybrid-ARQ-Systemen verwenden. Wir werden sehen, daß sie MDS–Codes sind.

Definition 2.16 (Reed–Solomon–Codes) *Es seien $b \geq 0$ und $1 \leq \delta \leq n$. Ferner sei ω eine primitive n-te Einheitswurzel über $GF(2^h)$, $h > 1$. Ein 2^h-närer* Reed–Solomon–Code (RS–Code) *ist ein BCH–Code $\mathcal{B}_h(n, \delta, \omega, b)$ der Länge $n = 2^h - 1$.*

Bezeichnung: $\mathcal{S}_h(\delta, \omega, b) := \mathcal{B}_h(n, \delta, \omega, b)$ mit $n = 2^h - 1$.

Wegen $n = 2^h - 1$ ist jede primitive n-te Einheitswurzel ω ein primitives Element von $GF(2^h)$. Für das Generatorpolynom eines RS–Codes erhalten wir deshalb (s. [17, S. 369]):

Satz 2.17 *Das Generatorpolynom $g(x)$ eines RS–Codes $\mathcal{S}_h(\delta, \omega, b)$ ist gegeben durch*

$$g(x) \;=\; \prod_{i=0}^{\delta-2}(x - \omega^{b+i}) \;=\; (x-\omega^b)(x-\omega^{b+1})\cdot\ldots\cdot(x-\omega^{b+\delta-2})\,.$$

Beweis: Die Wurzeln des Polynoms $x^n - 1$ sind die Elemente von $GF(2^h)^* := GF(2^h) \setminus \{0\}$, also gilt

$$x^n - 1 \;=\; \prod_{\alpha\in GF(2^h)^*}(x-\alpha) \;=\; \prod_{i=0}^{n-1}(x-\omega^i)\,.$$

Das Minimalpolynom von ω^i, $i = 0,\ldots,n-1$, ist daher $m_i(x) = x - \omega^i$. Hieraus folgt die Behauptung. q.e.d.

Während wir für die Dimension eines BCH–Codes nur eine Abschätzung angeben konnten, ergibt sich aus der obigen Darstellung des Generatorpolynoms auch die Dimension eines RS–Codes (s. [17, S. 370]):

Korollar 2.18 *Die Dimension eines RS–Codes $\mathcal{S}_h(\delta, \omega, b)$ beträgt $n - \delta + 1$.*

Beweis: Nach Satz 2.17 hat das Generatorpolynom $g(x)$ von $\mathcal{S}_h(\delta, \omega, b)$ den Grad $\delta - 1$. Dies impliziert $\dim(\mathcal{S}_h(\delta, \omega, b)) = n - \deg(g(x)) = n - \delta + 1$. q.e.d.

Ein großer Vorteil der RS–Codes gegenüber den restlichen BCH–Codes liegt darin, daß sie MDS–Codes sind.

Satz 2.19 (Singleton–Schranke) *Für einen linearen (n, k, d)-Code gilt $k \le n - d + 1$.*

Definition 2.20 (MDS–Codes) *Ein linearer (n, k, d)-Code, der die Singleton–Schranke mit Gleichheit erfüllt, d. h. für den $k = n - d + 1$ gilt, heißt* MDS–Code (Maximum Distance Separabel).

Quelle: van Lint ([10, S. 61]), Bossert ([4, S. 116]).

Bei gegebener Länge und Dimension haben MDS–Codes also den größten möglichen Minimalabstand. Für RS–Codes gilt (s. [17, S. 370]):

Satz 2.21 *RS–Codes sind MDS–Codes. Der Minimalabstand eines RS–Codes ist gleich dem Designabstand.*

Beweis: Es sei $\mathcal{S}_h(\delta, \omega, b)$ ein RS–Code mit Dimension $k = n - \delta + 1$ (s. Kor. 2.18). Für den Minimalabstand d von $\mathcal{S}_h(\delta, \omega, b)$ folgt aus der BCH–Schranke (Satz 2.9) und der Singleton–Schranke (Satz 2.19): $d \ge \delta = n - k + 1 \ge d$, also $d = \delta = n - k + 1$. q.e.d.

MDS–Codes besitzen eine Reihe schöner Eigenschaften, eine Übersicht findet man z. B. in dem Buch von Roman ([17, S. 235ff]). Für uns ist es wichtig, daß die Gewichtsverteilung der MDS–Codes bekannt ist.

Definition 2.22 *Es sei $\mathcal{C}$ ein Code der Länge n. Für $i = 0, 1, \ldots, n$ sei A_i die Anzahl der $\mathcal{C}$-Codewörter mit Hamming–Gewicht i. Die Menge $\{A_0, A_1, \ldots, A_n\}$ heißt* Gewichtsverteilung *von $\mathcal{C}$.*

Quelle: Roman ([17, S. 216]).

Satz 2.23 *Es sei $\{A_0, \ldots, A_n\}$ die Gewichtsverteilung eines RS–Codes $\mathcal{S}_h(\delta, \omega, b)$, dann gilt für $i = \delta, \ldots, n$:*

$$\begin{aligned} A_i &= \binom{n}{i} \sum_{j=0}^{i-\delta} (-1)^j \binom{i}{j} (2^{h(i-\delta+1-j)} - 1) \\ &= \binom{n}{i} (2^h - 1) \sum_{j=0}^{i-\delta} (-1)^j \binom{i-1}{j} 2^{h(i-\delta-j)} \, . \end{aligned}$$

Beweis und Quelle: Wegen Satz 2.21 folgt die Behauptung aus dem entsprechenden Satz für MDS–Codes, s. [17, S. 240]. □

Wir werden auch die Gewichtsverteilung des zu einem RS–Code dualen Codes benötigen. Dies stellt jedoch kein Problem dar, weil es sich hierbei wieder um einen RS–Code handelt (s. [17, S. 371]).

2.3.1 Binäre Codes aus RS–Codes

Für einige Anwendungen ist ein binärer Code besser geeignet als ein 2^h-närer ($h > 1$). Es ist aber nicht schwierig, einen (n, k)-RS–Code $\mathcal{S} := \mathcal{S}_h(\delta, \omega, b)$ in einen binären (hn, hk)-Code umzuwandeln (s. [17, S. 372ff]).

Den Körper $GF(2^h)$ können wir als h-dimensionalen Vektorraum über $GF(2)$ betrachten. Es sei $\{\alpha_1, \ldots, \alpha_h\}$ eine Basis von $GF(2^h)$ über $GF(2)$, dann können wir jedem Vektor $\bar{a} = \sum_{i=1}^{h} a_i \alpha_i$ das Wort $(a_1, \ldots, a_h)$ der Länge h zuordnen. Diese Zuordnung ist eineindeutig. Ein Codewort $\bar{c} = (\bar{c}_0, \ldots, \bar{c}_{n-1}) \in \mathcal{S}$ mit $\bar{c}_i = \sum_{j=1}^{h} c_{ij} \alpha_j$ für $i = 0, \ldots, n-1$ kann auf diese Weise als Binärwort

$$(c_{01}, \ldots, c_{0h}, c_{11}, \ldots, c_{1h}, \ldots, c_{n-1,1}, \ldots, c_{n-1,h})$$

der Länge hn dargestellt werden. So erhalten wir den Binärcode

$$\begin{aligned} \vec{\mathcal{S}} := \Big\{ &(c_{01}, \ldots, c_{0h}, c_{11}, \ldots, c_{1h}, \ldots, c_{n-1,1}, \ldots, c_{n-1,h}) \; : \\ &(\bar{c}_0, \ldots, \bar{c}_{n-1}) \in \mathcal{S} \text{ mit } \bar{c}_i = \sum_{j=1}^{h} c_{ij} \alpha_j \, , \; i = 0, \ldots, n-1 \Big\} \, . \end{aligned}$$

Über diesen Code können wir folgende Aussage machen (s. [17, S. 372]):

Satz 2.24 *Es seien $\mathcal{S} = \mathcal{S}_h(\delta, \omega, b)$ ein (n,k)-RS–Code und $\vec{\mathcal{S}}$ der gemäß obiger Konstruktion bezüglich der Basis $\{\alpha_1, \ldots, \alpha_h\}$ gebildete $(\vec{n}, \vec{k}, \vec{d})$-Binärcode, dann gilt:*

$$\vec{n} = hn\,, \quad \vec{k} = hk \quad \textit{und} \quad \vec{d} \geq \delta\,.$$

Beweis: Aus der Konstruktion des Codes $\vec{\mathcal{S}}$ folgt $\vec{n} = hn$ und $\vec{k} = hk$.

Nach Satz 2.21 hat $\mathcal{S}$ den Minimalabstand δ, also enthält jedes $\mathcal{S}$-Codewort außer dem Nullwort mindestens δ Symbole ungleich 0. Jedes aus einem dieser Symbole gebildete Binärwort der Länge h enthält mindestens eine 1. Der Code $\vec{\mathcal{S}}$ ist linear, deshalb folgt $\vec{d} \geq \delta$. q.e.d.

Bemerkungen:

a) Der Code $\vec{\mathcal{S}}$ ist linear, im allgemeinen aber nicht zyklisch (s. [17, S. 372f]).

b) Der Minimalabstand des Codes $\vec{\mathcal{S}}$ hängt von der Wahl der Basis $\{\alpha_1, \ldots, \alpha_h\}$ ab (s. [17, S. 373]).

Wir können den Code $\vec{\mathcal{S}}$ erweitern, indem wir an jedes Binärwort der Länge h, das einem Symbol aus $GF(2^h)$ zugeordnet wurde, ein Paritätskontrollbit anhängen (s.[17, S. 372f]). So erhalten wir den Code

$$\begin{aligned}\mathcal{S}^{\uparrow} \;:=\; \Big\{ & (c_{01}, \ldots, c_{0h}, x_0, c_{11}, \ldots, c_{1h}, x_1, \ldots, c_{n-1,1}, \ldots, c_{n-1,h}, x_{n-1}) \;: \\ & (c_{01}, \ldots, c_{0h}, \ldots, c_{n-1,1}, \ldots, c_{n-1,h}) \in \vec{\mathcal{S}}\,,\; x_i = \sum_{j=1}^{h} c_{ij}\,,\; i = 0, \ldots, n-1 \Big\}\,.\end{aligned}$$

Der Code $\mathcal{S}^{\uparrow}$ ist ein Vektorraum der Dimension hk über $GF(2)$ und hat die Länge $h(n+1)$ sowie mindestens den Minimalabstand 2δ. Letzteres folgt aus Satz 2.24 und der Tatsache, daß jeder Block $(c_{i1}, \ldots, c_{ih}, x_i) \neq \bar{0}$ mindestens zwei Einsen enthält (s. [17, S. 372f]).

Erkennung und Korrektur von Fehlerbündeln

Es sei $\mathcal{S} = \mathcal{S}_h(\delta, \omega, b)$ ein RS–Code mit ungeradem Minimalabstand $\delta = 2t + 1$. Wir haben in Abschnitt 2.2.3 gesehen, daß wir bis zu t Fehler korrigieren können. Der Binärcode $\vec{\mathcal{S}}$ kann daher alle Fehlermuster korrigieren, die höchstens t Blöcke der Länge h, die einem Symbol aus $GF(2^h)$ zugeordnet wurden, betreffen. Insbesondere können Fehlerbündel bis zur Länge $h(t-1)+1$ korrigiert werden (s. [17, S. 374]).

Analog können Fehlerbündel bis zur Länge $h(\delta - 2) + 1$ erkannt werden, dies geschieht z. B. durch Berechnung des Syndroms (vgl. Abschnitt 2.2.3).

2.3.2 Codierung von RS–Codes

RS–Codes sind spezielle BCH–Codes, daher können wir für die Codierung von RS–Codes die in Abschnitt 2.2.2 beschriebenen Verfahren verwenden. Eine Darstellung der Schieberegister–Schaltungen für die systematische Codierung findet man z. B. in dem Buch von Rhee ([16, S. 222ff]).

Ein weiteres Verfahren zur Codierung von RS–Codes im engeren Sinne stammt von Reed und Solomon (s. [17, S. 377f]). Es sei $\mathcal{S}_h(\delta, \omega)$ ein RS–Code im engeren Sinne mit der Dimension $k = n - \delta + 1$ (s. Kor. 2.18). Ein Informationswort $\bar{a} = (a_0, \ldots, a_{k-1}) \in (GF(2^h))^k$ können wir wieder als Polynom $a(x) = \sum_{j=0}^{k-1} a_j x^j$ auffassen. Nun sei $c(x) := \sum_{j=0}^{n-1} a(\omega^j) x^j$. Wir werden sehen, daß $c(x)$ ein Codepolynom in $\mathcal{S}_h(\delta, \omega)$ ist. Wenn wir $a_j := 0$ setzen für $j = k, \ldots, n-1$, kann man zeigen (s. [17, S. 337]), daß für $i = 0, \ldots, n-1$ gilt:

$$a_i \;=\; \frac{1}{n} \sum_{j=0}^{n-1} a(\omega^j) \omega^{-ij} \;=\; \frac{1}{n}\, c(\omega^{-i}) \;=\; \frac{1}{n}\, c(\omega^{n-i}) \;.$$

Für $j = 0, \ldots, \delta - 1$ erhalten wir dann wegen $n - j \geq n - \delta + 1 = k$:

$$c(\omega^j) \;=\; n a_{n-j} \;=\; 0 \;.$$

Also gilt $c(x) \in \mathcal{S}_h(\delta, \omega)$, so daß wir $a(x)$ in $c(x)$ codieren können.

Das Informationswort $\bar{a}$ kann aus dem Codepolynom $c(x)$ durch Auswertung von $c(x)$ an $\omega^\delta, \ldots, \omega^n$ (es gilt $n = \delta + k - 1$) rekonstruiert werden.

2.3.3 Decodierung von RS–Codes

Für die Fehlerkorrektur kann das Decodierverfahren aus Abschnitt 2.2.3 verwendet werden. Einige weitere Verfahren, die zum Teil auf der diskreten Fourier–Transformation bzw. Mattson–Solomon–Polynomen basieren, werden von Blahut ([3]) und Bossert ([4]) vorgestellt. Wir gehen hierauf jedoch nicht näher ein.

In dieser Arbeit werden wir RS–Codes stets als fehlererkennende Codes einsetzen. Die Decodierung beschränkt sich dann auf die Berechnung des Syndroms und gegebenenfalls die Rekonstruktion des übertragenen Informationswortes (s. Abschnitt 2.3.2).

Definition 2.25 *Es sei $\mathcal{C}$ ein linearer (n,k)-Code über $GF(2^h)$ mit Kontrollmatrix $\mathbf{H}$, dann ist für jedes $\bar{x} \in (GF(2^h))^n$ das* Syndrom *$\bar{s}$ von $\bar{x}$ definiert durch $\bar{s} := \bar{x}\mathbf{H}^T$.*

Bemerkung: Es sei $\bar{x} \in (GF(2^h))^n$ und $\bar{s} := \bar{x}\mathbf{H}^T$ das Syndrom von $\bar{x}$. Aus der Definition der Kontrollmatrix folgt, daß $\bar{s} = \bar{0}$ genau dann gilt, wenn $\bar{x} \in \mathcal{C}$ ist.

Quelle: van Lint ([10, S. 34]).

Wir können also durch Berechnung des Syndroms entscheiden, ob ein empfangenes Wort ein Codewort ist.

Es sei $\mathcal{S} := \mathcal{S}_h(\delta, \omega, b)$ ein RS–Code. Angenommen, das Codepolynom $c(x)$ wurde gesendet, und empfangen wurde das Polynom $r(x) = c(x) + e(x)$, wobei $e(x)$ ein Fehlermuster darstelle.

Das Syndrom $s(x)$ von $r(x)$ können wir wie folgt berechnen (s. [2, S. 468f]):
Nach der Bemerkung zu Definition 2.7 ist

$$\mathbf{H} := \begin{pmatrix} 1 & \omega^b & \omega^{2b} & \dots & \omega^{(n-1)b} \\ 1 & \omega^{b+1} & \omega^{2(b+1)} & \dots & \omega^{(n-1)(b+1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{b+\delta-2} & \omega^{2(b+\delta-2)} & \dots & \omega^{(n-1)(b+\delta-2)} \end{pmatrix}$$

eine Kontrollmatrix von $\mathcal{S}$. So erhalten wir

$$\begin{aligned} \bar{s} \;=\; \bar{r}\mathbf{H}^T \;&=\; \left(\sum_{j=0}^{n-1} r_j \omega^{jb}, \sum_{j=0}^{n-1} r_j \omega^{j(b+1)}, \dots, \sum_{j=0}^{n-1} r_j \omega^{j(b+\delta-2)} \right) \\ &=\; (r(\omega^b), r(\omega^{b+1}), \dots, r(\omega^{b+\delta-2}))\,, \end{aligned}$$

wobei $r(x) = \sum_{j=0}^{n-1} r_j x^j$, $\bar{r} = (r_0, \dots, r_{n-1})$, $s(x) = \sum_{j=0}^{\delta-2} s_j x^j$ und $\bar{s} = (s_0, \dots, s_{\delta-2})$ seien. Also ist $s_j = r(\omega^{b+j}) = e(\omega^{b+j})$ für $j = 0, \dots, \delta-2$ und es gilt $\bar{r} \in \mathcal{S}$ genau dann, wenn $\bar{s} = \bar{0}$ ist.

Eine Darstellung der Schieberegister–Schaltungen für die Berechnung des Syndroms findet man z. B. in dem Buch von Rhee ([16, S. 232]).

2.4 Polynomiale Codes

Polynomiale Codes sind spezielle systematische lineare Codes über $GF(2)$. Sie werden zur Fehlererkennung eingesetzt und sind gut geeignet, um Fehlerbündel zu erkennen. Ihre Codierer und Decodierer sind einfach zu realisieren (s. [13, S. 165]).

Wie in den vorherigen Abschnitten identifizieren wir Wörter der Länge n mit Polynomen vom Grad $n-1$. Polynomiale Codes werden — wie der Name schon suggeriert — von einem Polynom erzeugt, wir definieren sie folgendermaßen (s. [13, S. 166]):

Definition 2.26 (Polynomiale Codes) *Ein* polynomialer Code $\mathcal{C}$ *ist ein binärer systematischer linearer Code, dessen Codepolynome Vielfache eines gegebenen Polynoms $g(x)$ sind. Das Polynom $g(x)$ heißt* Generatorpolynom *von $\mathcal{C}$.*

Insbesondere sind binäre zyklische Codes auch polynomiale Codes.

2.4.1 Codierung und Decodierung polynomialer Codes

Es sei $\mathcal{C}$ ein polynomialer (n,k)-Code über $GF(2)$ mit Generatorpolynom $g(x)$.

Ein Informationswort $\bar{a}$ der Länge k wird als Polynom vom Grad $k-1$ dargestellt. Das Informationspolynom $a(x)$ wird mit x^{n-k} multipliziert und das Produkt durch $g(x)$ dividiert, so erhalten wir

$$x^{n-k}a(x) \;=\; q(x)g(x) + r(x)$$

mit $\deg(r(x)) < \deg(g(x)) = n-k$.

Das zu $\bar{a}$ gehörige Codepolynom ist (s. [13, S. 166])

$$c(x) \;:=\; x^{n-k}a(x) - r(x) \;=\; q(x)g(x)\;.$$

Diese Codierung ist systematisch, denn die Koeffizienten $c_{n-k}, \ldots, c_{n-1}$ von $c(x) = \sum_{i=0}^{n-1} c_i x^i$ stellen das Informationswort $\bar{a}$ dar.

Bei der Decodierung muß also nur die Fehlererkennung durchgeführt werden. Dies geschieht durch Berechnung des Syndroms (s. [13, S. 167f]). Jedes Codepolynom ist ein Vielfaches des Generatorpolynoms $g(x)$ und jedes Vielfache von $g(x)$ mit Grad kleiner als n ist ein Codepolynom.

Angenommen, das Codepolynom $c(x) = q(x)g(x)$ wurde gesendet, und empfangen wurde das Polynom $t(x) = c(x) + e(x)$, wobei $e(x)$ ein Fehlermuster darstelle. Das Syndrom $s(x)$ ist dann gegeben durch

$$s(x) \;\equiv\; t(x) \;\equiv\; e(x) \bmod g(x)\;.$$

Daher gilt $s(x) \equiv 0$ genau dann, wenn $e(x)$ ein Codepolynom ist. Ein Übertragungsfehler wird also genau dann erkannt, wenn $e(x)$ kein Codepolynom ist.

Darstellungen der Schieberegister–Schaltungen für die Codierung und Decodierung polynomialer Codes findet man z. B. in dem Buch von Nussbaumer ([13, S. 168ff]).

2.4.2 Fehlererkennung

Der oben beschriebene Decodieralgorithmus ermöglicht die Erkennung verschiedener zufälliger Fehlermuster und Fehlerbündel. Wir fassen dies in dem folgenden Satz zusammen (s. [13, S. 173f]):

Satz 2.27 *Es sei $\mathcal{C}$ ein polynomialer (n,k)-Code mit Generatorpolynom $g(x) = \sum_{i=0}^{n-k} g_i x^i$, und es seien $g_0 = g_{n-k} = 1$. Bei Verwendung des obigen Decodieralgorithmusses gilt:*

a) Jeder Einzelbitfehler wird erkannt.

b) Jeder Doppelbitfehler wird erkannt, falls $g(x)$ keines der Polynome $x^u + 1$, $u = 1, \ldots, n-1$, teilt.

c) Jedes Fehlermuster mit einer ungeraden Anzahl von Fehlern wird erkannt, falls $g(x)$ den Faktor $x + 1$ enthält.

d) Jedes Fehlerbündel, das höchstens die Länge $n - k$ hat, wird erkannt.

e) Ein Fehlerbündel der Länge b wird mit Wahrscheinlichkeit $1 - (\frac{1}{2})^{n-k-1}$ erkannt, falls $b = n - k + 1$ ist, und mit Wahrscheinlichkeit $1 - (\frac{1}{2})^{n-k}$, falls $b > n - k + 1$ ist.

Beweis: Es sei $e(x)$ ein Fehlerpolynom. Ein Übertragungsfehler wird genau dann entdeckt, wenn $e(x)$ kein Vielfaches von $g(x)$ ist (s. Abschnitt 2.4.1).

Zu (a): Es gelte $e(x) = x^i$ für ein $i \in \{0, \ldots, n-1\}$. Wegen $g_0 = g_{n-k} = 1$ kann $e(x)$ kein Vielfaches von $g(x)$ sein.

Zu (b): Es gelte $e(x) = x^i + x^j$ mit $i, j \in \{0, \ldots, n-1\}$ und $i < j$, dann ist $e(x) = x^i(x^{j-i}+1)$ mit $j - i \in \{1, \ldots, n-1\}$. Wenn wir annehmen, daß $e(x) = q(x)g(x)$ sei, folgt aus $g_0 = 1$, daß $q(x) = x^i q_1(x)$ sein muß. Dies impliziert aber $x^{j-i} + 1 = q_1(x)g(x)$ im Widerspruch zur Voraussetzung. Also kann $e(x)$ kein Vielfaches von $g(x)$ sein.

Zu (c): Eine ungerade Anzahl von Fehlern impliziert $e(1) = 1$. Falls $g(x)$ den Faktor $x + 1$ enthält, gilt $g(1) = 0$. Also ist $e(x)$ kein Vielfaches von $g(x)$.

Zu (d) und (e): Ein Fehlerbündel der Länge b entspricht

$$\begin{aligned} e(x) &= x^i + e_{i+1}x^{i+1} + \ldots + e_{i+b-2}x^{i+b-2} + x^{i+b-1} \\ &= x^i(1 + e_{i+1}x + \ldots + e_{i+b-2}x^{b-2} + x^{b-1}) \quad =: \quad x^i\tilde{e}(x)\,, \end{aligned}$$

wobei $i \in \{0, \ldots, n-b\}$ sei.

Falls $b \leq n - k$ ist, ist $\tilde{e}(x)$ wegen $\deg(g(x)) = n - k > b - 1 = \deg(\tilde{e}(x))$ kein Vielfaches von $g(x)$. Wie im Beweis zu (b) folgt, daß $e(x)$ kein Vielfaches von $g(x)$ ist. Hieraus erhalten wir die Aussage (d).

Es sei nun $b \geq n - k + 1$. Wir nehmen an, daß $\tilde{e}(x) = q(x)g(x)$ gelte, wobei $q(x)$ ein Polynom vom Grad $b - 1 - (n - k)$ sei. Außerdem muß der nullte Koeffizient von $q(x)$ gleich 1 sein (dies folgt aus $g_0 = e_i = 1$).

Falls $b = n - k + 1$ ist, hat $q(x)$ den Grad 0, so daß $\tilde{e}(x) = g(x)$ sein muß. Es gibt 2^{n-k-1} Möglichkeiten, die Koeffizienten $e_{i+1}, \ldots, e_{i+n-k-1}$ von $\tilde{e}(x)$ zu wählen. Die Wahrscheinlichkeit für das Eintreten des Ereignisses „$\tilde{e}(x) = g(x)$“ beträgt deshalb $2^{-(n-k-1)}$.

Falls $b > n - k + 1$ ist, gibt es $2^{b-(n-k)-2}$ Möglichkeiten, die restlichen Koeffizienten von $q(x)$ zu wählen. Jede dieser Kombinationen liefert ein Polynom $q_i(x)$, und es gilt $\tilde{e}(x) = q_i(x)g(x)$ mit Wahrscheinlichkeit $2^{-(b-2)}$ (analog zum Fall $b = n - k + 1$), $i = 1, \ldots, 2^{b-(n-k)-2}$. Deshalb gilt $\tilde{e}(x) = q(x)g(x)$ mit Wahrscheinlichkeit $2^{b-(n-k)-2} \cdot 2^{-(b-2)} = 2^{-(n-k)}$. Insgesamt erhalten wir die Aussage (e). q.e.d.

Bemerkung: Im Beweis zu (e) setzen wir voraus, daß die Wahrscheinlichkeit für das Auftreten eines Fehlers in einem Fehlerbündel $\frac{1}{2}$ beträgt. Diese Annahme ist bei der Analyse polynomialer Codes üblich. Eine neuere Arbeit, in der hierauf verzichtet und ein neues Modell zur Beschreibung von Fehlerbündeln vorgestellt wird, stammt von Wolf und Chun ([19]). Sie analysieren binäre zyklische Codes mit Generatorpolynomen der Form $g(x) = (1 + x)p(x)$, wobei $p(x)$ ein primitives Polynom sei, und kommen zu dem Ergebnis, daß die Wahrscheinlichkeit für die Erkennung von Fehlerbündeln von der Wahl des Polynoms $p(x)$ abhängt. Dies gilt sogar für Codes mit der gleichen Anzahl von Kontrollbits. Wir gehen hierauf nicht näher ein, da dies den Rahmen dieser kurzen Einführung in die Theorie der polynomialen Codes sprengen würde.

2.4.3 Standardisierte polynomiale Codes

Verschiedene polynomiale Codes wurden standardisiert, wir nennen kurz drei Beispiele (s. [13, S. 174f]), ohne hierauf näher einzugehen.

Bezeichnung:	Generatorpolynom:
CCITT V.41	$x^{16} + x^{12} + x^5 + 1$
CRC-16	$x^{16} + x^{15} + x^2 + 1$
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$

Für eine detaillierte Betrachtung sei auf das Buch von Nussbaumer verwiesen ([13, S. 174f]).

2.5 Hamming–Codes

Eine wichtige Klasse fehlerkorrigierender Codes bilden die Hamming–Codes. Sie wurden 1949 von Marcel Golay und unabhängig davon 1950 von Richard W. Hamming entdeckt ([17, S. 253]). Die zugrundeliegenden geometrischen Überlegungen soll R. W. Hamming 1948 während einer Autofahrt angestellt haben ([5, S. 28]). Die Hamming–Codes sind 1-fehlerkorrigierende, perfekte Codes und können sehr einfach decodiert werden.

Bekanntlich ist der Minimalabstand eines linearen Codes gleich der minimalen Anzahl linear abhängiger Spalten der Kontrollmatrix (s. [5, S. 227], [17, S. 201]). Ein 1-fehlerkorrigierender Code hat den Minimalabstand $d = 3$ oder $d = 4$; um einen „guten“ Code[4] konstruieren zu können, wählen wir $d = 3$. Dies führt zu der Definition der Hamming–Codes (s. [10, S. 36], [17, S. 254]):

Definition 2.28 (Hamming–Codes) *Es seien q eine Primzahlpotenz , $r \in \mathbb{N}$ und $n := \frac{q^r-1}{q-1}$. Der $(n, n-r)$-*Hamming–Code $\mathcal{H}_q(r)$ der Ordnung r *über $GF(q)$ ist ein Code, dessen Kontrollmatrix aus Spalten besteht, die über $GF(q)$ paarweise linear unabhängig sind.*

Bemerkungen:

a) Wir unterscheiden hier nicht zwischen äquivalenten Codes, d. h. zwischen Codes, deren Kontrollmatrizen sich nur in der Reihenfolge ihrer Spalten unterscheiden.

b) Die Spalten der Kontrollmatrix eines Hamming–Codes $\mathcal{H}_q(r)$ bilden eine maximale Menge paarweise linear unabhängiger Vektoren aus $(GF(q))^r$ (s. [10, S. 36]), denn es gibt genau $n = \frac{q^r-1}{q-1}$ solcher Vektoren (s. [17, S. 254]). Dies impliziert, daß die Kontrollmatrix drei linear abhängige Spalten enthält, denn die Summe zweier Spalten ist entweder selbst eine Spalte oder das skalare Vielfache einer Spalte. Der Hamming–Code $\mathcal{H}_q(r)$ hat also den Minimalabstand 3.

[4] Wir werden bald sehen, inwiefern die Hamming–Codes „gute“ Codes sind.

Hamming–Codes sind perfekte Codes, d. h. sie erfüllen die Hamming–Schranke mit Gleichheit.

Satz 2.29 (Hamming–Schranke) *Für einen $[n, M, d]$-Code[5] über $GF(q)$ gilt*

$$M \leq \frac{q^n}{\sum_{i=0}^{t} \binom{n}{i}(q-1)^i}, \quad \textit{wobei } t := \left\lfloor \tfrac{d-1}{2} \right\rfloor \textit{ sei.}$$

Bemerkung: In der englischsprachigen Literatur wird die Hamming–Schranke auch „Sphere–Packing Bound“ genannt.

Quelle: Roman ([17, S. 171]).

Korollar 2.30 *Für einen linearen (n, k, d)-Code über $GF(q)$ gilt:*

$$q^{n-k} \geq \sum_{i=0}^{t} \binom{n}{i}(q-1)^i, \quad \textit{wobei } t := \left\lfloor \tfrac{d-1}{2} \right\rfloor \textit{ sei.}$$

Beweis: Ein linearer (n, k, d)-Code hat die Größe $M = q^k$, die Behauptung folgt deshalb aus Satz 2.29. □

Definition 2.31 (Perfekte Codes) *Es sei $\mathcal{C}$ ein Code der Länge n über $GF(q)$.*

a) *Der* Packradius *(engl.:* packing radius*) $pr(\mathcal{C})$ ist die größte Zahl $r \in \mathbb{N}$, für die die Kugeln mit Radius r um alle $\mathcal{C}$-Codewörter paarweise disjunkt sind.*

b) *Der* Überdeckungsradius *(engl.:* covering radius*) $cr(\mathcal{C})$ ist die kleinste Zahl $s \in \mathbb{N}$, für die die Kugeln mit Radius s um alle $\mathcal{C}$-Codewörter den Raum $(GF(q))^n$ überdecken.*

c) *Falls $pr(\mathcal{C}) = cr(\mathcal{C})$ gilt, heißt $\mathcal{C}$* perfekter Code.

Satz 2.32 *Es sei $\mathcal{C}$ ein $[n, M, d]$-Code über $GF(q)$.*

a) *Der Code $\mathcal{C}$ ist genau dann perfekt, wenn $d = 2t + 1$ für ein $t \in \mathbb{N}$ ist und*

$$M = \frac{q^n}{\sum_{i=0}^{t} \binom{n}{i}(q-1)^i}$$

gilt.

b) *Der Code $\mathcal{C}$ ist genau dann perfekt, wenn $d = 2t + 1$ für ein $t \in \mathbb{N}$ ist und jedes Wort aus $(GF(q))^n$ den Hamming–Abstand kleiner oder gleich t zu genau einem Codewort aus $\mathcal{C}$ hat.*

[5] Ein $[n, M, d]$-Code besitzt M Codewörter der Länge n und den Minimalabstand d.

Quellen: van Lint ([10, S. 32]), Roman ([17, S. 138f]).

Für Hamming–Codes gilt (s. [10, S. 36f], [17, S. 254]):

Satz 2.33 *Jeder Hamming–Code ist ein perfekter Code.*

Beweis: Für einen (n, k, d)-Hamming–Code $\mathcal{H}_q(r)$ gilt $n = \frac{q^r-1}{q-1}$, $k = n - r$ und $d = 3$. Es sei $\bar{x} \in (GF(q))^n$, dann enthält die Kugel mit Radius 1 um $\bar{x}$ genau $1+n(q-1) = q^r = q^{n-k}$ Wörter. Deshalb enthält die Vereinigung der q^k Kugeln mit Radius 1 um die Codewörter von $\mathcal{H}_q(r)$ alle $q^k \cdot q^{n-k} = q^n$ Wörter von $(GF(q))^n$. Wegen $d = 3$ sind diese Kugeln paarweise disjunkt. Also gilt $pr(\mathcal{H}_q(r)) = cr(\mathcal{H}_q(r)) = 1$, so daß der Code $\mathcal{H}_q(r)$ perfekt ist. q.e.d.

Wir werden Hamming–Codes als innere fehlerkorrigierende Codes in Hybrid-ARQ-Systemen verwenden, deshalb betrachten wir im folgenden nur noch binäre Hamming–Codes.

2.5.1 Zyklische Hamming–Codes

Wir können binäre Hamming–Codes auch als zyklische Codes darstellen (s. [17, S. 329]):

Satz 2.34 *Der binäre Hamming–Code $\mathcal{H}_2(r)$ der Ordnung r ist äquivalent zu einem zyklischen Code.*

Beweis: Der Hamming–Code $\mathcal{H}_2(r)$ hat die Länge $n = 2^r - 1$ und die Dimension $k = 2^r - 1 - r$. Die Spalten seiner Kontrollmatrix sind die $2^r - 1$ verschiedenen Vektoren der Länge r über $GF(2)$ außer dem Nullvektor. Es sei ω eine primitive n-te Einheitswurzel über $GF(2)$. Die Ordnung von 2 modulo n beträgt r, d. h. $r = \min\{t \in \mathbb{N} : 2^t \equiv 1 \bmod n\}$. Also ist $GF(2^r)$ der Zerfällungskörper von $x^n - 1$ über $GF(2)$ (s. [17, S. 309], vgl. auch Satz 2.10). Als primitive n-te Einheitswurzel hat ω die Ordnung $n = 2^r - 1$ und ist deshalb ein primitives Element von $GF(2^r)$. Für ein Element $\omega^i \in GF(2^r)$, $i \in \{0, \ldots, n-1\}$, bezeichne $[\omega^i]$ den zugehörigen Spaltenvektor der Länge r über $GF(2)$. Die Matrix

$$\mathbf{H} := \left(\begin{array}{cccc} [\omega^0] & [\omega^1] & \ldots & [\omega^{n-1}] \end{array} \right)$$

ist dann die Kontrollmatrix eines binären Hamming–Codes $\mathcal{H}_2(r)$ der Ordnung r. Aus Satz 2.6 und der Bemerkung zu Definition 2.7 folgt, daß $\mathbf{H}$ die Kontrollmatrix eines zyklischen Codes ist, dessen Generatorpolynom das Minimalpolynom von ω ist. Dieser Code ist äquivalent zu $\mathcal{H}_2(r)$.

q.e.d.

2.5.2 Simplex–Codes (duale Hamming–Codes)

Es sei $\mathcal{H}_2(r)$ ein binärer $(2^r-1, 2^r-1-r)$-Hamming–Code. Der zu $\mathcal{H}_2(r)$ duale $(2^r-1, r)$-Code heißt *Simplex–Code* und wird mit Σ_r bezeichnet (s. [17, S. 256]).

Satz 2.35 *Für den Simplex–Code Σ_r gilt:*

a) Jedes Codewort außer dem Nullwort hat das Hamming–Gewicht 2^{r-1}, also ist Σ_r ein $(2^r-1, r, 2^{r-1})$-Code.

b) Der Hamming–Abstand zweier beliebiger Σ_r-Codewörter beträgt 2^{r-1}.

c) Der Simplex–Code Σ_r ist zyklisch.

Quelle: Roman ([17, S. 257]).

Die Aussage (b) erklärt, warum diese Codes Simplex–Codes heißen: Wenn wir uns die Codewörter als Punkte im (2^r-1)-dimensionalen Euklidischen Raum vorstellen, bilden ihre Verbindungslinien einen regulären Simplex (s. [17, S. 257f]).

Für uns sind die Simplex–Codes interessant, weil wir mit ihrer Hilfe den Gewichtszeiger der binären Hamming–Codes bestimmen können.

2.5.3 Gewichtszeiger der binären Hamming–Codes

Der Gewichtszeiger eines linearen Codes ist die erzeugende Funktion der Gewichtsverteilung.

Definition 2.36 *Es sei $\mathcal{C}$ ein linearer (n,k)-Code mit Gewichtsverteilung $\{A_0, \ldots, A_n\}$. Der* Gewichtszeiger *$W_{\mathcal{C}}(x)$ von $\mathcal{C}$ ist definiert durch*

$$W_{\mathcal{C}}(x) := \sum_{i=0}^{n} A_i x^i \qquad (x \in \mathbb{R}).$$

Quelle: Heise e. a. ([5, S. 204f]).

Der Satz 2.35 liefert uns den Gewichtszeiger der Simplex–Codes. Hieraus können wir mit Hilfe der MacWilliams–Identitäten den Gewichtszeiger der binären Hamming–Codes bestimmen.

Satz 2.37 (MacWilliams–Identitäten) *Es sei $\mathcal{C}$ ein linearer (n,k)-Code über $GF(2)$ mit Gewichtsverteilung $\{A_0, \ldots, A_n\}$. Ferner sei $\{B_0, \ldots, B_n\}$ die Gewichtsverteilung des zu $\mathcal{C}$ dualen Codes, dann gilt:*

$$\sum_{i=0}^{n} A_i \, x^{n-i} \, y^i \;=\; \frac{1}{2^{n-k}} \sum_{i=0}^{n} B_i \, (x+y)^{n-i} (x-y)^i \qquad (x, y \in \mathbb{R}).$$

Quelle: MacWilliams e. a. ([11, S. 127f]).

So erhalten wir den folgenden Satz (s. [16, S. 63]):

Satz 2.38 *Es sei $\mathcal{H}_2(r)$ ein binärer Hamming–Code der Länge $n = 2^r - 1$. Der Gewichtszeiger $W_{\mathcal{H}_2(r)}(x)$ von $\mathcal{H}_2(r)$ ist gegeben durch*

$$W_{\mathcal{H}_2(r)}(x) \;=\; \frac{1}{n+1}\left((1+x)^n + n(1+x)^{\frac{n-1}{2}}(1-x)^{\frac{n+1}{2}}\right) \qquad (x \in \mathbb{R}).$$

Beweis: Es sei Σ_r der zu $\mathcal{H}_2(r)$ duale Simplex–Code. Ferner seien $\{A_0, \ldots, A_n\}$ die Gewichtsverteilung von $\mathcal{H}_2(r)$ und $\{B_0, \ldots, B_n\}$ die von Σ_r. Für den Gewichtszeiger $W_{\Sigma_r}(x)$ von Σ_r gilt nach Satz 2.35(a):

$$W_{\Sigma_r}(x) \;=\; B_0 + B_{2^{r-1}}x^{2^{r-1}} \;=\; 1 + (2^r - 1)x^{2^{r-1}}\,.$$

Aus den MacWilliams–Identitäten folgt deshalb:

$$\begin{aligned} W_{\mathcal{H}_2(r)}(x) &= \sum_{i=0}^{n} A_i x^i \\ &= \frac{1}{2^r}\left(B_0(1+x)^n + B_{2^{r-1}}(1+x)^{n-2^{r-1}}(1-x)^{2^{r-1}}\right) \\ &= \frac{1}{n+1}\left((1+x)^n + n(1+x)^{\frac{n-1}{2}}(1-x)^{\frac{n+1}{2}}\right)\,. \end{aligned}$$

q.e.d.

Wir können also durch Berechnung des Polynoms

$$\frac{1}{n+1}\left((1+x)^n + n(1+x)^{\frac{n-1}{2}}(1-x)^{\frac{n+1}{2}}\right)$$

die Gewichtsverteilung eines binären Hamming–Codes der Länge n bestimmen.

2.5.4 Codierung von Hamming–Codes

Wir kennen bisher nur die Kontrollmatrix $\mathbf{H}$ eines binären Hamming–Codes $\mathcal{H}_2(r)$ der Ordnung r. Mit ihrer Hilfe können wir aber auch eine Generatormatrix konstruieren. Falls $\mathbf{H}$ in der Form $\mathbf{H} = [\mathbf{A} \mid \mathbf{I}_r]$ vorliegt, wobei $\mathbf{I}_r$ die $(r \times r)$-Einheitsmatrix bezeichne und $\mathbf{A}$ eine $(r \times (2^r - 1 - r))$-Matrix sei, ist die Generatormatrix $\mathbf{G}$ gegeben durch $\mathbf{G} = [\mathbf{I}_{2^r-1-r} \mid -\mathbf{A}^T]$, wobei $\mathbf{I}_{2^r-1-r}$ die $((2^r - 1 - r) \times (2^r - 1 - r))$-Einheitsmatrix bezeichne (s. [10, S. 34]). Es gilt nämlich

$$\mathbf{G}\mathbf{H}^T \;=\; \left[\mathbf{I}_{2^r-1-r} \mid -\mathbf{A}^T\right]\begin{bmatrix}\mathbf{A}^T \\ \mathbf{I}_r\end{bmatrix} \;=\; \mathbf{A}^T - \mathbf{A}^T \;=\; \mathbf{O}\,.$$

Falls die Matrix $\mathbf{H}$ in einer anderen Form vorliegt, kann sie durch Spaltenvertauschungen in die Form $\mathbf{H} = [\mathbf{A} \mid \mathbf{I}_r]$ gebracht werden (s. [10, S. 33], [17, S. 198]). Wenn wir nun die Matrix $\mathbf{G}$ wie

oben konstruieren und anschließend die gleichen Spaltenvertauschungen durchführen, erhalten wir eine Generatormatrix von $\mathcal{H}_2(r)$.

Ein Informationswort $\bar{a} \in (GF(2))^k$ wird codiert durch Multiplikation mit der Generatormatrix $\mathbf{G}$, das zugehörige Codewort ist also $\bar{c} = \bar{a}\mathbf{G}$.

Die Darstellung einer Schieberegister–Schaltung für die systematische Codierung mit einer Generatormatrix $\mathbf{G} = [\mathbf{I}_{2^r-1-r} \mid -\mathbf{A}^T]$ findet man z. B. in dem Buch von Rhee ([16, S. 42]).

2.5.5 Decodierung von Hamming–Codes

Hamming–Codes sind 1-fehlerkorrigierend. Wir betrachten ein Verfahren, um einen Fehler korrigieren zu können (s. [17, S. 254f]).

Es sei $\mathcal{H}_2(r)$ ein binärer Hamming–Code der Ordnung r mit Kontrollmatrix $\mathbf{H}$. Die $2^r - 1$ Spalten von $\mathbf{H}$ sind die Binärdarstellungen der Zahlen von 1 bis $2^r - 1$. Bei der Definition der Hamming–Codes haben wir nicht zwischen äquivalenten Codes unterschieden; wir können deshalb annehmen, daß die Spalten von $\mathbf{H}$ so angeordnet sind, daß jede Spalte (von oben nach unten gelesen) die Binärdarstellung ihrer Spaltennummer ist (s. [17, S. 254f]).

Angenommen, das Codewort $\bar{c} \in \mathcal{H}_2(r)$ wurde gesendet, und empfangen wurde das Wort $\bar{w} = \bar{c} + \bar{e}$, wobei $\bar{e}$ ein Fehlermuster darstelle. Falls ein Fehler in der i-ten Position aufgetreten ist ($i \in \{1, \ldots, 2^r - 1\}$), ist $\bar{e}$ der i-te Einheitsvektor $\bar{e}_i$. Die Berechnung des Syndroms $\bar{s}$ von $\bar{w}$ (s. Def. 2.25) liefert $\bar{s} = \bar{w}\mathbf{H}^T = \bar{e}_i\mathbf{H}^T$. Also ist $\bar{s}$ die i-te Spalte von $\mathbf{H}$ und daher die Binärdarstellung der Fehlerposition i. Damit können wir das gesendete Codewort $\bar{c} = \bar{w} + \bar{e}_i$ erhalten. Die Kenntnis der Informationsstellen (s. Abschnitt 2.5.4) ermöglicht uns die Rekonstruktion des in $\bar{c}$ codierten Informationswortes.

Spezielle Decodierverfahren für zyklische Hamming–Codes findet man z. B. in dem Buch von Rhee ([16, S. 116ff]).

2.5.6 Verkürzte und erweiterte Hamming–Codes

Ein binärer Hamming–Code $\mathcal{H}_2(r)$ der Ordnung r kann durch das Anhängen eines zusätzlichen Paritätskontrollbits erweitert werden. Der so definierte $(\hat{n}, \hat{k}, \hat{d})$-Code $\hat{\mathcal{H}}_2(r)$ hat die Parameter $\hat{n} = 2^r$, $\hat{k} = 2^r - 1 - r$ und $\hat{d} = 4$ (s. [17, S. 155]). Der Code $\hat{\mathcal{H}}_2(r)$ hat einen geraden Minimalabstand und ist deshalb nicht perfekt (s. Satz 2.32). Er ist ebenfalls 1-fehlerkorrigierend.

Das Verkürzen von Codes ist eine gebräuchliche Technik, um aus einem gegebenen Code einen neuen zu konstruieren. Weil sie häufig auf Hamming–Codes angewendet wird, wollen wir sie kurz erläutern (s. [17, S. 157]): Es sei $\mathcal{C}$ ein Code der Länge n. Wir wählen eine Position $i \in \{1, \ldots, n\}$ und behalten nur die Codewörter, die an der i-ten Position ein bestimmtes Symbol haben, z. B. 0. Alle anderen Codewörter werden aus $\mathcal{C}$ entfernt, bei den verbleibenden wird die i-te Stelle gestrichen. So erhalten wir einen Code $\mathcal{C}'$ der Länge $n-1$. Offensichtlich haben $\mathcal{C}$ und $\mathcal{C}'$ den gleichen Minimalabstand.

Falls $\mathcal{C}$ linear ist und ein Symbol ungleich 0 entfernt wurde, ist $\mathcal{C}'$ nicht linear (s. [10, S. 47]), denn das Nullwort ist dann kein $\mathcal{C}'$-Codewort. Außerdem ist $\mathcal{C}'$ im allgemeinen nicht mehr zyklisch, wenn $\mathcal{C}$ zyklisch ist (s. [16, S. 110]).

2.6 Krishna–Morgera–Codes

In diesem Abschnitt betrachten wir eine Klasse von Codes, deren Generatormatrix wir bei der Berechnung von Bilinearformen erhalten. Diese Codes wurden von Krishna und Morgera entdeckt und werden von ihnen „Krishna–Morgera–Codes“ („KM–Codes“) genannt (s. [8], [9], [12]). Wir werden diese Bezeichnung übernehmen, obwohl sie in den zu Rate gezogenen Lehrbüchern anderer Autoren nicht zu finden ist.

KM–Codes sind besonders für den Einsatz als innere, fehlerkorrigierende Codes in verallgemeinerten Typ-II Hybrid-ARQ-Systemen geeignet. Wir werden hier nur binäre KM–Codes betrachten, eine Verallgemeinerung ist jedoch möglich (s.[8], [12]).

2.6.1 Berechnung von Bilinearformen und Konstruktion der KM–Codes

Im folgenden seien $F := GF(2)$, I eine Menge von Unbekannten (Variablen) und $F(I)$ der kleinste kommutative Ring, der $F \cup I$ enthält ([8, S. 11f]). Ferner sei Ψ ein System von k Bilinearformen über F und I, d. h. (s. [12, S. 12]) $\Psi = \{\psi_1, \ldots, \psi_k\}$ und

$$\psi_t \;=\; \sum_{i=1}^{r} \sum_{j=1}^{s} \alpha_{tij} x_i y_j \,, \quad t = 1, \ldots, k \,,$$

mit $\alpha_{tij} \in F$ und $x_i, y_j \in I$ $(t = 1, \ldots, k,\ i = 1, \ldots, r,\ j = 1, \ldots, s)$. In Matrixform können wir Ψ folgendermaßen darstellen (s. [12, S. 12]):

$$\boldsymbol{\psi} = \mathbf{X}\mathbf{y} \,,$$

wobei

$$\boldsymbol{\psi} := \begin{pmatrix} \psi_1 \\ \vdots \\ \psi_k \end{pmatrix} , \quad \mathbf{X} := \begin{pmatrix} \sum\limits_{i=1}^{r} \alpha_{1i1} x_i & \ldots & \sum\limits_{i=1}^{r} \alpha_{1is} x_i \\ \vdots & & \vdots \\ \sum\limits_{i=1}^{r} \alpha_{ki1} x_i & \ldots & \sum\limits_{i=1}^{r} \alpha_{kis} x_i \end{pmatrix} , \quad \mathbf{y} := \begin{pmatrix} y_1 \\ \vdots \\ y_s \end{pmatrix} \tag{2.4}$$

seien. Außerdem sei $\mathbf{x} := (x_1, \ldots, x_r)^T$, dann gilt (s. [12, S. 13]):

Satz 2.39 *Wir können $\boldsymbol{\psi} = \mathbf{X}\mathbf{y}$ mit Hilfe des bilinearen Algorithmusses*

$$\boldsymbol{\psi} \;=\; \mathbf{C}(\mathbf{A}\mathbf{x} \bullet \mathbf{B}\mathbf{y}) \;=:\; \mathbf{C}\mathbf{m}$$

berechnen. Hierbei seien $\mathbf{A}$ eine $(n \times r)$-, $\mathbf{B}$ eine $(n \times s)$- und $\mathbf{C}$ eine $(k \times n)$-Matrix mit Einträgen aus F für ein geeignetes $n \in \mathbb{N}$. Ferner bezeichne $\bullet$ die komponentenweise Multiplikation von Vektoren.

Beweis: Es seien $\mathbf{A} = (a_{ij})_{\substack{i=1,\dots,n\\ j=1,\dots,r}}$, $\mathbf{B} = (b_{ij})_{\substack{i=1,\dots,n\\ j=1,\dots,s}}$ und $\mathbf{C} = (c_{ij})_{\substack{i=1,\dots,k\\ j=1,\dots,n}}$, dann gilt:

$$\begin{aligned}
\mathbf{Cm} &= \mathbf{C}(\mathbf{Ax} \bullet \mathbf{By}) \\
&= \mathbf{C} \begin{pmatrix} \sum_{i=1}^{r}\sum_{j=1}^{s} a_{1i}b_{1j}x_iy_j \\ \vdots \\ \sum_{i=1}^{r}\sum_{j=1}^{s} a_{ni}b_{nj}x_iy_j \end{pmatrix} \\
&= \begin{pmatrix} \sum_{l=1}^{n}\sum_{i=1}^{r}\sum_{j=1}^{s} c_{1l}a_{li}b_{lj}x_iy_j \\ \vdots \\ \sum_{l=1}^{n}\sum_{i=1}^{r}\sum_{j=1}^{s} c_{kl}a_{li}b_{lj}x_iy_j \end{pmatrix} \quad \stackrel{!}{=} \quad \boldsymbol{\psi} \quad = \quad \begin{pmatrix} \sum_{i=1}^{r}\sum_{j=1}^{s} \alpha_{1ij}x_iy_j \\ \vdots \\ \sum_{i=1}^{r}\sum_{j=1}^{s} \alpha_{kij}x_iy_j \end{pmatrix} .
\end{aligned}$$

Hieraus ergibt sich das Gleichungssystem

$$\alpha_{tij} = \sum_{l=1}^{n} c_{tl}a_{li}b_{lj} , \quad t = 1,\dots,k , \; i = 1,\dots,r , \; j = 1,\dots,s . \tag{2.5}$$

Dies sind krs Gleichungen für die $n(k+r+s)$ Unbekannten c_{tl} , a_{li} und b_{lj}. Wir müssen n so groß wählen, daß dieses Gleichungssystem lösbar ist. q.e.d.

Eine untere Schranke für n erhalten wir durch die Betrachtung der minimalen Anzahl μ von Multiplikationen, die zur Berechnung von $\boldsymbol{\psi}$ benötigt werden. Multiplikationen mit Elementen aus F werden hierbei nicht mitgezählt (s. [12, S. 12]). Der Algorithmus $\boldsymbol{\psi} = \mathbf{C}(\mathbf{Ax} \bullet \mathbf{By}) = \mathbf{Cm}$ benötigt in diesem Sinne n Multiplikationen, sie treten bei der Berechnung von $\mathbf{m} = \mathbf{Ax} \bullet \mathbf{By}$ auf (s. [12, S. 13]). Es muß also $n \geq \mu$ gelten. Die unteren Schranken für μ stellen daher auch untere Schranken für n dar.

Definition 2.40 *Es seien F^m und $F^m(I)$ die m-dimensionalen Vektorräume über F bzw. $F(I)$.*

a) Eine Menge $\{\mathbf{v}_1,\dots,\mathbf{v}_n\}$ von Vektoren aus $F^m(I)$ ist linear unabhängig modulo F^m*, wenn für $a_1,\dots,a_n \in F$ aus $\sum_{i=1}^{n} a_i\mathbf{v}_i \in F^m$ stets $a_i = 0$ für alle $i = 1,\dots,n$ folgt.*

b) Es sei $\mathbf{X}$ eine $(k \times s)$-Matrix mit Einträgen aus $F(I)$. Der Zeilenrang modulo F^s *von $\mathbf{X}$ ist die Anzahl der modulo F^s linear unabhängigen Zeilen von $\mathbf{X}$. Der* Spaltenrang modulo F^k *von $\mathbf{X}$ ist die Anzahl der modulo F^k linear unabhängigen Spalten von $\mathbf{X}$.*

Bezeichnung: $\rho_r(\mathbf{X})$:= Zeilenrang modulo F^s von $\mathbf{X}$,
$\rho_c(\mathbf{X})$:= Spaltenrang modulo F^k von $\mathbf{X}$.

Satz 2.41 *Für die minimale Anzahl μ von Multiplikationen, die zur Berechnung von $\boldsymbol{\psi} = \mathbf{Xy}$ benötigt werden, gilt:*

a) $\mu \geq \max\{\rho_r(\mathbf{X}), \rho_c(\mathbf{X})\}$,

b) Falls $\mathbf{X}$ *eine* $(\alpha \times \beta)$*-Untermatrix* $\mathbf{W}$ *hat, so daß aus* $\mathbf{u}^T\mathbf{Wv} \in F$ *für* $\mathbf{u} \in F^\alpha$ *und* $\mathbf{v} \in F^\beta$ *stets* $\mathbf{u} = \mathbf{0}$ *oder* $\mathbf{v} = \mathbf{0}$ *folgt, dann ist* $\mu \geq \alpha + \beta - 1$.

Quelle: Morgera e. a. ([12, S. 13f]).

Von besonderem Interesse ist für uns der Fall $\rho_r(\mathbf{X}) = k$. Wegen $\boldsymbol{\psi} = \mathbf{Xy} = \mathbf{C}(\mathbf{Ax} \bullet \mathbf{By})$ hat dann auch $\mathbf{C}$ den Rang k und kann daher als Generatormatrix eines (n, k)-Codes verwendet werden (nach Satz 2.41(a) gilt $n \geq k$). Über den Minimalabstand dieses Codes können wir folgendes aussagen (s. [12, S. 18f]):

Satz 2.42 *Die Matrix* $\mathbf{C}$ *erzeugt einen linearen* (n, k, d)*-Code* $\mathcal{C}$ *über* $F = GF(2)$ *mit Minimalabstand*

$$d \geq \delta := \min\{\, \rho_c(\mathbf{a}^T\mathbf{X}) \,:\, \mathbf{a} \in F^k \,,\, \mathbf{a} \neq \mathbf{0} \,\} \,.$$

Die Größe δ *heißt* Designabstand *von* $\mathcal{C}$.

Beweis: Es sei $\mathbf{a} \in F^k \setminus \{\mathbf{0}\}$, das zugehörige Codewort ist dann $\mathbf{c}^T := \mathbf{a}^T\mathbf{C} \neq \mathbf{0}^T$.
Wir betrachten nun die Bilinearform $\mathbf{a}^T\mathbf{Xy}$. Für sie gilt:

$$\mathbf{a}^T\mathbf{Xy} \;=\; \mathbf{a}^T\mathbf{C}(\mathbf{Ax} \bullet \mathbf{By}) \;=\; \mathbf{c}^T\mathbf{m} \,.$$

Jede Komponente von $\mathbf{m}$ entspricht einer Multiplikation im oben genannten Sinne. Dies impliziert, daß $\mathbf{c}$ mindestens so viele Einträge ungleich 0 haben muß, wie Multiplikationen nötig sind, um $\mathbf{a}^T\mathbf{Xy}$ berechnen zu können (sonst hätten wir einen Algorithmus gefunden, der weniger Multiplikationen benötigt). Aus Satz 2.41(a) folgt $w(\mathbf{c}) \geq \rho_c(\mathbf{a}^T\mathbf{X})$. q.e.d.

Eine spezielle Bilinearform

Wir betrachten folgendes System von Bilinearformen (s. [12, S. 19f]):

$$\boldsymbol{\psi} \;=\; \mathbf{Xy} \;:=\; \begin{pmatrix} x_0 & x_1 & x_2 & \dots & x_{\delta-1} \\ x_1 & x_2 & x_3 & \dots & x_\delta \\ \vdots & \vdots & \vdots & & \vdots \\ x_{k-1} & x_k & x_{k+1} & \dots & x_{k+\delta-2} \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{\delta-1} \end{pmatrix} . \tag{2.6}$$

Offensichtlich gilt $\rho_r(\mathbf{X}) = k$.
Es seien $\mathbf{a} = (a_0, \dots, a_{k-1})^T \in F^k \setminus \{\mathbf{0}\}$ und $\mathbf{b} = (b_0, \dots, b_{\delta-1})^T \in F^\delta$, dann folgt

$$\mathbf{a}^T\mathbf{X} \;=\; \left(\sum_{i=0}^{k-1} a_i x_i \,,\; \sum_{i=0}^{k-1} a_i x_{i+1} \,, \dots \,,\; \sum_{i=0}^{k-1} a_i x_{i+\delta-1} \right)$$

und

$$\sum_{j=0}^{\delta-1} b_j(\mathbf{a}^T\mathbf{X})_j \;=\; \sum_{j=0}^{\delta-1}\sum_{i=0}^{k-1} a_i b_j x_{i+j} \;=\; \sum_{\substack{i=0\\ a_i=1}}^{k-1}\sum_{\substack{j=0\\ b_j=1}}^{\delta-1} x_{i+j}\;.$$

Daher ist $\sum_{j=0}^{\delta-1} b_j(\mathbf{a}^T\mathbf{X})_j = \mathbf{a}^T\mathbf{X}\mathbf{b}$ nur dann in F, wenn $\mathbf{b} = \mathbf{0}$ ist. Nach Def. 2.40 hat $\mathbf{a}^T\mathbf{X}$ den Spaltenrang modulo F $\rho_c(\mathbf{a}^T\mathbf{X}) = \delta$. Außerdem gilt $\mathbf{a}^T\mathbf{X}\mathbf{b} \neq 0$ für alle $\mathbf{a} \in F^k \setminus \{\mathbf{0}\}$ und $\mathbf{b} \in F^\delta \setminus \{\mathbf{0}\}$, so daß nach Satz 2.41(b) mindestens $k + \delta - 1$ Multiplikationen nötig sind, um $\boldsymbol{\psi} = \mathbf{X}\mathbf{y}$ zu berechnen. Für einen Algorithmus $\boldsymbol{\psi} = \mathbf{X}\mathbf{y} = \mathbf{C}(\mathbf{A}\mathbf{x} \bullet \mathbf{B}\mathbf{y})$ mit einer $(k \times n)$-Matrix $\mathbf{C}$ gilt deshalb $n \geq k + \delta - 1$.

Der von $\mathbf{C}$ erzeugte (n, k, d)-Code $\mathcal{C}$ heißt *Krishna–Morgera–Code (KM–Code)*. Nach Satz 2.42 gilt $d \geq \delta$. Falls $n = k + d - 1$ ist, ist $\mathcal{C}$ ein MDS–Code (s. Def. 2.20). Dies ist insbesondere erfüllt, wenn $n = k + \delta - 1$ ist, denn dann folgt aus der Singleton–Schranke (Satz 2.19) $\delta \leq d \leq n - k + 1 = \delta$.

Wir stehen nun vor der Frage, wie wir einen Algorithmus $\boldsymbol{\psi} = \mathbf{X}\mathbf{y} = \mathbf{C}(\mathbf{A}\mathbf{x} \bullet \mathbf{B}\mathbf{y})$ entwickeln und die uns interessierende Matrix $\mathbf{C}$ bestimmen können. Hierfür betrachten wir ein Verfahren, das auf dem Prinzip der aperiodischen (nichtzyklischen) Faltung basiert (s. [8, S. 16ff], [12, S. 20ff]). Morgera und Krishna stellen in ihrem Buch ([12, S. 20ff]) noch ein Verfahren vor, das auf der periodischen (zyklischen) Faltung basiert; wir gehen hierauf jedoch nicht näher ein, da dieses für unsere Anwendungen — auch nach Aussage von Morgera und Krishna ([12, S. 20]) — weniger interessant zu sein scheint.

2.6.2 Bestimmung der Generatormatrix

Wir betrachten weiterhin das System von Bilinearformen Ψ mit $\boldsymbol{\psi} = \mathbf{X}\mathbf{y} = \mathbf{C}(\mathbf{A}\mathbf{x} \bullet \mathbf{B}\mathbf{y})$. Es seien $\mathbf{A}_i^T \in F^r$ und $\mathbf{B}_j^T \in F^s$ die i-te bzw. j-te Zeile von $\mathbf{A}$ bzw. $\mathbf{B}$ $(i, j = 1, \ldots, n)$, dann können wir Ψ auch so darstellen (s. [12, S. 16]):

$$\psi_t \;=\; \mathbf{x}^T\mathbf{G}_t\mathbf{y}\;, \quad t = 1, \ldots, k,$$

wobei $\mathbf{G}_t$ folgende $(r \times s)$-Matrix sei:

$$\mathbf{G}_t \;:=\; \sum_{l=1}^{n} c_{tl}\mathbf{A}_l\mathbf{B}_l^T\;, \quad t = 1, \ldots, k.$$

Für die Matrizen $\mathbf{G}_1, \ldots, \mathbf{G}_k$ gilt (s. [12, S. 17]):

Lemma 2.43 *Die Matrix $\mathbf{G}_t$ besitzt die Einträge α_{tij} $(t = 1, \ldots, k,\ i = 1, \ldots, r,\ j = 1, \ldots, s)$.*

Beweis: Die Matrix $\mathbf{A}_l\mathbf{B}_l^T$ hat die Einträge $(\mathbf{A}_l\mathbf{B}_l^T)_{ij} = a_{li}b_{lj}$. Mit (2.5) folgt:

$$(\mathbf{G}_t)_{ij} \;=\; \sum_{l=1}^{n} c_{tl}a_{li}b_{lj} \;=\; \alpha_{tij} \quad (t = 1, \ldots, k,\ i = 1, \ldots, r,\ j = 1, \ldots, s).$$

q.e.d.

Es sei nun $\mathbf{z} = (z_1, \ldots, z_k)^T \in I^k$ ein weiterer Vektor bestehend aus k Variablen, die unabhängig von $\mathbf{x} = (x_1, \ldots, x_r)^T \in I^r$ und $\mathbf{y} = (y_1, \ldots, y_s)^T \in I^s$ gewählt wurden. Wir betrachten die Trilinearform (s. [12, S. 16f])

$$\tau \;:=\; \mathbf{z}^T\mathbf{X}\mathbf{y} \;=\; \sum_{t=1}^{k} z_t\psi_t \;=\; \sum_{t=1}^{k}\sum_{i=1}^{r}\sum_{j=1}^{s}\alpha_{tij}x_iy_jz_t\,.$$

Wenn wir die Matrix

$$\mathbf{Z} \;:=\; \sum_{t=1}^{k} z_t\mathbf{G}_t \;=\; \sum_{t=1}^{k}\sum_{l=1}^{n} z_t c_{tl}\mathbf{A}_l\mathbf{B}_l^T \tag{2.7}$$

definieren, folgt aus Lemma 2.43:

$$\mathbf{z}^T\mathbf{X}\mathbf{y} \;=\; \sum_{t=1}^{k}\sum_{i=1}^{r}\sum_{j=1}^{s}\alpha_{tij}x_iy_jz_t \;=\; \sum_{t=1}^{k} z_t\mathbf{x}^T\mathbf{G}_t\mathbf{y} \;=\; \mathbf{x}^T\mathbf{Z}\mathbf{y}\,. \tag{2.8}$$

Das System von Bilinearformen Φ, das durch $\boldsymbol{\phi} = \mathbf{Z}\mathbf{y}$ gegeben ist, heißt zu Ψ *duales System* (s. [12, S. 17]).

Aus dem Algorithmus $\boldsymbol{\psi} = \mathbf{C}(\mathbf{A}\mathbf{x} \bullet \mathbf{B}\mathbf{y})$ können wir einen Algorithmus zur Berechnung von Φ erhalten. Es sei $\mathbf{D}_a(\mathbf{x})$ die $(n \times n)$-Diagonalmatrix mit den Einträgen $\sum_{j=1}^{r} a_{ij}x_j$ $(i = 1, \ldots, n)$, dann folgt (s. [12, S. 15]):

$$\boldsymbol{\psi} \;=\; \mathbf{X}\mathbf{y} = \mathbf{C}(\mathbf{A}\mathbf{x} \bullet \mathbf{B}\mathbf{y}) \;=\; \mathbf{C}\mathbf{D}_a(\mathbf{x})\mathbf{B}\mathbf{y}\,.$$

Weil dies für alle $\mathbf{y} \in I^s$ gelten soll, ergibt sich $\mathbf{X} = \mathbf{C}\mathbf{D}_a(\mathbf{x})\mathbf{B}$.

Außerdem sei $\mathbf{D}_c(\mathbf{z})$ die $(n \times n)$-Diagonalmatrix mit den Einträgen $\sum_{t=1}^{k} c_{tj}z_t$ $(j = 1, \ldots, n)$, dann folgt aus (2.7): $\mathbf{Z} = \mathbf{A}^T\mathbf{D}_c(\mathbf{z})\mathbf{B}$. Ein bilinearer Algorithmus zur Berechnung von Φ ist daher gegeben durch (s. [12, S. 18]):

$$\boldsymbol{\phi} \;=\; \mathbf{Z}\mathbf{y} \;=\; \mathbf{A}^T(\mathbf{C}^T\mathbf{z} \bullet \mathbf{B}\mathbf{y})\,.$$

Der obige Algorithmus $\boldsymbol{\phi} = \mathbf{A}^T(\mathbf{C}^T\mathbf{z} \bullet \mathbf{B}\mathbf{y})$ wird als *A-dual* (manchmal auch *P-dual*) zu $\boldsymbol{\psi} = \mathbf{C}(\mathbf{A}\mathbf{x} \bullet \mathbf{B}\mathbf{y})$ bezeichnet (s. [12, S. 18]). Für die Anzahl der benötigten Multiplikationen gilt der folgende Satz:

Satz 2.44 *Es gibt ein Verfahren zur Berechnung von* $\mathbf{C}(\mathbf{A}\mathbf{x} \bullet \mathbf{B}\mathbf{y})$*, das* n *Multiplikationen benötigt, genau dann, wenn es ein Verfahren zur Berechnung von* $\mathbf{A}^T(\mathbf{C}^T\mathbf{z} \bullet \mathbf{B}\mathbf{y})$ *gibt, das* n *Multiplikationen benötigt.*

Quelle: Morgera e. a. ([12, S. 18]).

Wir können die Generatormatrix $\mathbf{C}$ also auch bestimmen, indem wir $\boldsymbol{\phi} = \mathbf{A}^T(\mathbf{C}^T\mathbf{z} \bullet \mathbf{B}\mathbf{y})$ berechnen. Das System von Bilinearformen Φ kann auch mittels einer aperiodischen Faltung

beschrieben werden. Wir betrachten erneut das System von Bilinearformen $\boldsymbol{\psi} = \mathbf{Xy}$ aus (2.6), für die Trilinearform τ gilt dann mit einem Vektor $\mathbf{z} = (z_0, \ldots, z_{k-1})^T \in I^k$ (s. [12, S. 22]):

$$\tau \;=\; \mathbf{z}^T\mathbf{Xy} \;=\; \sum_{t=0}^{k-1} z_t \sum_{j=0}^{\delta-1} x_{t+j}y_j \;=\; \sum_{t=0}^{k+\delta-2} x_t \sum_{j=0}^{t} z_j y_{t-j}\ ,$$

wobei $z_j = 0$ für $j > k-1$ und $y_j = 0$ für $j > \delta - 1$ seien. Mit (2.8) erhalten wir

$$\tau \;=\; \mathbf{z}^T\mathbf{Xy} = \mathbf{x}^T\mathbf{Zy} \;=\; \mathbf{x}^T\boldsymbol{\phi}\,.$$

Hieraus folgt

$$\phi_t \;=\; \sum_{j=0}^{t} z_j y_{t-j}\,, \quad t = 0, \ldots, k+\delta-2\ .$$

Dies entspricht der aperiodischen Faltung zweier Folgen $z_0, z_1, \ldots, z_{k-1}$ und $y_0, y_1, \ldots, y_{\delta-1}$ (s. [12, S. 2]). Es seien $z(u) := \sum_{t=0}^{k-1} z_t u^t$ und $y(u) := \sum_{t=0}^{\delta-1} y_t u^t$ die erzeugenden Polynome von $\mathbf{z}$ und $\mathbf{y}$, dann gilt (s. [12, S. 23]):

$$\phi(u) \;:=\; \sum_{t=0}^{k+\delta-2} \phi_t u^t \;=\; z(u)y(u)\ .$$

Das Polynom $\phi(u) = z(u)y(u)$ hat den Grad $N - 1 := k + \delta - 2$. Wir können $\phi(u)$ daher auch folgendermaßen berechnen (s. [12, S. 67]):

$$\phi(u) \;\equiv\; z(u)y(u) \bmod p(u)\ ,$$

wobei $p(u)$ ein Polynom vom Grad $\deg(p(u)) \geq N$ sei. Dies führt zu einem Algorithmus, den wir zunächst kurz vorstellen und dann auf unser Problem anwenden wollen. Für eine detailliertere Betrachtung sei auf das Buch von Morgera e. a. ([12, S. 67ff]) verwiesen, der Algorithmus heißt dort „Aperiodic Convolution with Wraparound“.

Satz 2.45 *Das Produkt $\phi(u) = z(u)y(u)$ zweier Polynome $z(u)$ und $y(u)$ vom Grad $N-1$ (bzw. $\phi(u) \equiv z(u)y(u) \bmod p(u)$ mit $\deg(p(u)) = D := 2N-1$) kann ermittelt werden durch Berechnung der $D - s$ Koeffizienten ($s \in \mathbb{N}, s < D$) von*

$$\phi_I(u) \;\equiv\; z(u)y(u) \bmod (u^{D-s} - 1)$$

und der ersten und letzten $\frac{s}{2}$ Koeffizienten von $\phi(u)$, falls s gerade ist, bzw.
der ersten (letzten) $\frac{s-1}{2}$ und letzten (ersten) $\frac{s+1}{2}$ Koeffizienten von $\phi(u)$, falls s ungerade ist.
Zwischen den Koeffizienten $\phi_0^I, \ldots, \phi_{D-s-1}^I$ von $\phi_I(u)$ und $\phi_0, \ldots, \phi_{D-1}$ von $\phi(u)$ besteht die Beziehung

$$\phi_t^I \;=\; \begin{cases} \phi_t + \phi_{t+D-s} & \text{für } 0 \leq t \leq s-1, \\ \phi_t & \text{für } s \leq t < D-s. \end{cases} \tag{2.9}$$

Es seien $\hat{z}(u) = u^{N-1}z(u^{-1})$ *und* $\hat{y}(u) = u^{N-1}y(u^{-1})$ *die zu* $z(u)$ *bzw.* $y(u)$ *reziproken Polynome*[6]*. Die* s *Koeffizienten von*

$$\hat{\phi}(u) \equiv \hat{z}(u)\hat{y}(u) \bmod u^s$$

stellen die Werte dar, die benötigt werden, um $\phi(u)$ *aus* $\phi_I(u)$ *rekonstruieren zu können.*

Quelle: Morgera e. a. ([12, S. 67f]).

Bemerkungen:

a) Für das Polynom $\hat{\phi}(u)$ gilt

$$\hat{\phi}(u) = \hat{z}(u)\hat{y}(u) = u^{2N-2}z(u^{-1})y(u^{-1}) = u^{D-1}\phi(u^{-1}),$$

so daß $\hat{\phi}(u)$ das zu $\phi(u)$ reziproke Polynom ist. Es seien $\hat{\phi}_0, \ldots, \hat{\phi}_{s-1}$ die ersten s Koeffizienten von $\hat{\phi}(u)$, dann folgt:

$$\hat{\phi}_0 = \phi_{D-1}, \quad \hat{\phi}_1 = \phi_{D-2}, \ \ldots, \quad \hat{\phi}_{s-1} = \phi_{D-s}. \tag{2.10}$$

Die Beziehungen (2.9) und (2.10) ermöglichen uns also die Berechnung von $\phi(u)$.

b) Der Parameter s sollte so gewählt werden, daß möglichst wenige Multiplikationen benötigt werden. Nach Aussage von Krishna und Morgera ist dies nicht unbedingt einfach; sie empfehlen, beginnend bei 1 nur die ersten möglichen Werte von s zu untersuchen ([12, S. 68]).

Wir können die Berechnung von $\phi_I(u)$ vereinfachen, indem wir ein Polynom $p(u)$ verwenden, das in paarweise teilerfremde Faktoren mit möglichst niedrigen Graden zerfällt. Hierfür benötigen wir das folgende wichtige Ergebnis aus der Algebra.

Satz 2.46 (Chinesischer Restsatz für Polynome) *Das Polynom* $p(u) = \prod_{i=1}^{L} p_i(u)$ *sei das Produkt von* L *paarweise teilerfremden Polynomen* $p_1(u), \ldots, p_L(u)$*. Im Ring der Polynome modulo* $p(u)$ *kann jedes Polynom* $y(u)$ *in der Form*

$$y(u) \equiv \sum_{i=1}^{L} s_i(u)y_i(u) \bmod p(u)$$

dargestellt werden, wobei

$$y_i(u) \equiv y(u) \bmod p_i(u)$$

sei und die Polynome $s_i(u)$ *die Kongruenzen*

$$s_i(u) \equiv \begin{cases} 0 \bmod p_j(u) & \forall\, j \in \{1, \ldots, L\} \setminus \{i\} \\ 1 \bmod p_i(u) \end{cases}$$

[6] Es sei $a(u) = \sum_{i=0}^{n} a_i u^i$ ein Polynom vom Grad n, das zu $a(u)$ reziproke Polynom ist $\hat{a}(u) = u^n a(u^{-1}) = \sum_{i=0}^{n} a_{n-i}u^i$ (s. [17, S. 307]; vgl. auch Satz 2.5).

erfüllen ($i = 1, \ldots, L$). Die Polynome $s_i(u)$ haben die Gestalt

$$s_i(u) \;=\; r_i(u) \prod_{\substack{j=1 \\ j \neq i}}^{L} p_j(u) \;,$$

wobei die Polynome $r_i(u)$ durch die Kongruenz

$$r_i(u) \prod_{\substack{j=1 \\ j \neq i}}^{L} p_j(u) \;\equiv\; 1 \bmod p_i(u)$$

bestimmt sind ($i = 1, \ldots, L$). Diese Darstellung von $y(u)$ ist eindeutig.

Quellen: Krishna ([8, S. 18f]), Nussbaumer ([14, S. 26f]), Blahut ([3, S. 77f]).

Die Sätze 2.45 und 2.46 ermöglichen uns die Berechnung von $\phi(u) = z(u)y(u)$. Wenn wir das System von Bilinearformen Φ wieder in der Form $\boldsymbol{\phi} = \mathbf{A}^T(\mathbf{C}^T\mathbf{z} \bullet \mathbf{B}\mathbf{y})$ darstellen, erhalten wir die Matrix $\mathbf{C}$. Zur Verdeutlichung betrachten wir ein Beispiel aus dem Buch von Morgera e. a. ([12, S. 118ff]).

Beispiel

Es seien $z(u) = \sum_{i=0}^{3} \zeta_i u^i$ und $y(u) = \sum_{i=0}^{2} \eta_i u^i$, also $k = 4$ und $\delta = 3$. Wir wählen $s = 1$ und $p(u) = u(u^2+1)(u^2+u+1)$, um $\phi(u) = z(u)y(u)$ zu berechnen. Das Polynom p(u) können wir in die paarweise teilerfremden Faktoren $p_1(u) = u$, $p_2(u) = u^2 + 1$ und $p_3(u) = u^2 + u + 1$ zerlegen.

Durch Polynomdivision erhalten wir für $z_i(u) \equiv z(u) \bmod p_i(u)$, $y_i(u) \equiv y(u) \bmod p_i(u)$ und $\phi_i(u) \equiv \phi(u) \bmod p_i(u)$ $(i = 1, 2, 3)$:

a) $z_1(u) = \zeta_0$,
$y_1(u) = \eta_0$ und
$\phi_1(u) = \zeta_0\eta_0 =: m_0$,

b) $z_2(u) = (\zeta_0 + \zeta_2) + (\zeta_1 + \zeta_3)u$,
$y_2(u) = (\eta_0 + \eta_2) + \eta_1 u$ und
$\phi_2(u) = (m_1 + m_3) + (m_1 + m_2 + m_3)u$,
mit $m_1 := (\zeta_0 + \zeta_2)(\eta_0 + \eta_2)$,
$m_2 := (\zeta_0 + \zeta_1 + \zeta_2 + \zeta_3)(\eta_0 + \eta_1 + \eta_2)$,
$m_3 := (\zeta_1 + \zeta_3)\eta_1$,

c) $z_3(u) = (\zeta_0 + \zeta_2 + \zeta_3) + (\zeta_1 + \zeta_2)u$,
$y_3(u) = (\eta_0 + \eta_2) + (\eta_1 + \eta_2)u$ und

$$\begin{aligned}&\phi_3(u) = (m_4 + m_6) + (m_4 + m_5)u,\\ &\text{mit}\quad m_4 := (\zeta_0 + \zeta_2 + \zeta_3)(\eta_0 + \eta_2),\\ &\qquad m_5 := (\zeta_0 + \zeta_1 + \zeta_3)(\eta_0 + \eta_1),\\ &\qquad m_6 := (\zeta_1 + \zeta_2)(\eta_1 + \eta_2).\end{aligned}$$

Nach dem Chinesischen Restsatz (Satz 2.46) gilt

$$\phi(u) \;\equiv\; \sum_{i=1}^{3} s_i(u)\phi_i(u) \bmod p(u)\;,$$

wobei die Polynome $s_i(u)$ durch

$$s_i(u) \;=\; r_i(u) \prod_{\substack{j=1\\ j\neq i}}^{3} p_j(u)$$

gegeben sind und die Polynome $r_i(u)$ mit dem Euklidischen Algorithmus (s. [3, S. 76f], [14, S. 5ff]) als Lösung der Kongruenzen

$$r_i(u) \prod_{\substack{j=1\\ j\neq i}}^{3} p_j(u) \;\equiv\; 1 \bmod p_i(u)$$

bestimmt werden können ($i = 1, 2, 3$).

Der Fall $i = 3$ ist gut geeignet, um dieses Verfahren zu erläutern; wir behandeln ihn deshalb ausführlich: Gesucht wird die Lösung $(r_3(u), q_3(u))$ der Gleichung

$$r_3(u)p_1(u)p_2(u) + q_3(u)p_3(u) \;=\; 1\;.$$

Durch Polynomdivision von $p_1(u)p_2(u) = u^3 + u$ durch $p_3(u) = u^2 + u + 1$ erhalten wir:

$$(u^3 + u) \;=\; (u^2 + u + 1)(u + 1) + (u + 1)\;.$$

Der erste Schritt des Euklidischen Algorithmusses zur Bestimmung des größten gemeinsamen Teilers von $p_1(u)p_2(u)$ und $p_3(u)$ lautet deshalb:

$$\begin{pmatrix} 0 & 1\\ 1 & u+1 \end{pmatrix} \begin{pmatrix} u^3 + u\\ u^2 + u + 1 \end{pmatrix} \;=\; \begin{pmatrix} u^2 + u + 1\\ u + 1 \end{pmatrix}\;.$$

Wir erhalten wieder durch Polynomdivision:

$$(u^2 + u + 1) \;=\; (u + 1)u + 1\;.$$

Der zweite Schritt des Euklidischen Algorithmusses ist daher:

$$\begin{pmatrix} 0 & 1\\ 1 & u \end{pmatrix} \begin{pmatrix} u^2 + u + 1\\ u + 1 \end{pmatrix} \;=\; \begin{pmatrix} u + 1\\ 1 \end{pmatrix}\;.$$

Dies impliziert:

$$\begin{aligned} 1 &= 1 \cdot (u^2+u+1) + u(u+1) \\ &= 1 \cdot (u^2+u+1) + u[1 \cdot (u^3+u) + (u+1)(u^2+u+1)] \\ &= u(u^3+u) + (1+u+u^2)(u^2+u+1) \ . \end{aligned}$$

Also ist $r_3(u) = u$ und folglich $s_3(u) = u(u^3+u) = u^4+u^2$.
Analog erhalten wir $s_1(u) = u^4+u^3+u+1$ und $s_2(u) = u^3+u^2+u$.
Es sei $\sum_{i=0}^{4} \phi'_i u^i \equiv \phi(u) \bmod p(u)$, dann gilt:

$$\begin{aligned} \phi'_0 &= m_0 \ , \\ \phi'_1 &= m_0 + m_1 + m_3 + m_4 + m_5 \ , \\ \phi'_2 &= m_2 + m_5 + m_6 \ , \\ \phi'_3 &= m_0 + m_2 + m_4 + m_5 \ , \\ \phi'_4 &= m_0 + m_1 + m_2 + m_3 + m_5 + m_6 \ . \end{aligned}$$

Um $\phi(u)$ mit dem Verfahren aus Satz 2.45 berechnen zu können, benötigen wir noch den Koeffizienten von $\hat{\phi}(u) \equiv \hat{z}(u)\hat{y}(u) \bmod u$. Aus der Definition von $\hat{z}(u)$ und $\hat{y}(u)$ folgt, daß dieser gleich $m_7 := \zeta_3\eta_2$ ist. Daher gilt nach Satz 2.45:

$$\phi(u) \ = \ m_7 p(u) + \sum_{i=0}^{4} \phi'_i u^i \ .$$

Wegen $p(u) = u^5+u^4+u^2+u$ erhalten wir für die Koeffizienten $\phi_0, \ldots, \phi_5$ von $\phi(u)$:

$$\begin{array}{lllllllll} \phi_0 = \phi'_0 & = m_0 & & & & & & & \\ \phi_1 = \phi'_1 + m_7 & = m_0 & +m_1 & & +m_3 & +m_4 & +m_5 & & +m_7 \\ \phi_2 = \phi'_2 + m_7 & = & & m_2 & & & +m_5 & +m_6 & +m_7 \\ \phi_3 = \phi'_3 & = m_0 & & +m_2 & & +m_4 & +m_5 & & \\ \phi_4 = \phi'_4 + m_7 & = m_0 & +m_1 & +m_2 & +m_3 & & +m_5 & +m_6 & +m_7 \\ \phi_5 = m_7 & = & & & & & & & m_7 \ . \end{array}$$

Aus der Definition der m_i, $i = 0, \ldots, 7$, folgt die Darstellung von $\boldsymbol{\phi}$ in der Form
$\boldsymbol{\phi} = \mathbf{A}^T(\mathbf{C}^T\mathbf{z} \bullet \mathbf{B}\mathbf{y})$:

$$
\begin{aligned}
\boldsymbol{\phi} &= \mathbf{A}^T(\mathbf{C}^T\mathbf{z} \bullet \mathbf{B}\mathbf{y}) \\
&= \begin{bmatrix} 1&0&0&0&0&0&0&0 \\ 1&1&0&1&1&1&0&1 \\ 0&0&1&0&0&1&1&1 \\ 1&0&1&0&1&1&0&0 \\ 1&1&1&1&0&1&1&1 \\ 0&0&0&0&0&0&0&1 \end{bmatrix} \left(\begin{bmatrix} 1&0&0&0 \\ 1&0&1&0 \\ 1&1&1&1 \\ 0&1&0&1 \\ 1&0&1&1 \\ 1&1&0&1 \\ 0&1&1&0 \\ 0&0&0&1 \end{bmatrix} \begin{bmatrix} \zeta_0 \\ \zeta_1 \\ \zeta_2 \\ \zeta_3 \end{bmatrix} \bullet \begin{bmatrix} 1&0&0 \\ 1&0&1 \\ 1&1&1 \\ 0&1&0 \\ 1&0&1 \\ 1&1&0 \\ 0&1&1 \\ 0&0&1 \end{bmatrix} \begin{bmatrix} \eta_0 \\ \eta_1 \\ \eta_2 \end{bmatrix} \right) .
\end{aligned}
$$

Die Generatormatrix des zugehörigen KM–Codes ist daher gegeben durch

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

Der so definierte lineare Code $\mathcal{C}$ hat die Länge $n = 8$, die Dimension $k = 4$ und mindestens den Minimalabstand $\delta = 3$. Wenn wir das Informationswort $\bar{z} := (1, 1, 1, 0)$ betrachten, so stellen wir fest, daß das zugehörige Codewort $\bar{z}\mathbf{C} = (1, 0, 1, 1, 0, 0, 0, 0)$ das Hamming–Gewicht 3 hat. Der Minimalabstand d des Codes $\mathcal{C}$ beträgt deshalb 3 und ist gleich dem Designabstand δ. Bei gegebener Länge $n = 8$ und gegebenem Minimalabstand $d = 3$ hat $\mathcal{C}$ die größte mögliche Dimension. Aus der Hamming–Schranke (Satz 2.29) folgt nämlich für einen binären linearen $(8, \tilde{k}, 3)$-Code:

$$
2^{\tilde{k}} \leq \frac{2^8}{\binom{8}{0} + \binom{8}{1}} = \frac{2^8}{9} < 1,8 \cdot 2^4 .
$$

Dies impliziert $\tilde{k} \leq 4$. Wir stellen aber auch fest, daß $\mathcal{C}$ nicht perfekt ist (s. Satz 2.32). Außerdem gilt $k = 4 < n - d + 1 = 6$, so daß $\mathcal{C}$ auch kein MDS–Code ist (s. Def. 2.20).

2.6.3 Codierung von KM–Codes

Im letzten Abschnitt haben wir gesehen, wie wir die Generatormatrix $\mathbf{C}$ eines KM–Codes bestimmen und ein Informationswort $(\zeta_0, \ldots, \zeta_{k-1})$ codieren können. Wir wollen dies kurz zusammenfassen (s. [12, S. 132]).

Es sei $p(u) = \prod_{i=1}^{L} p_i(u)$ das Produkt paarweise teilerfremder Polynome $p_1(u), \ldots, p_L(u)$. Ferner sei das Polynom $y(u) = \sum_{i=0}^{\delta-1} \eta_i u^i$ gegeben, und es seien $y_i(u) \equiv y(u) \bmod p_i(u)$ für $i = 1, \ldots, L$. Das Polynom $z(u) = \sum_{i=0}^{k-1} \zeta_i u^i$ stelle ein Informationswort $\bar{z} = (\zeta_0, \ldots, \zeta_{k-1})$ dar. Die Codierung von $z(u)$ erfolgt dann in drei Schritten:

a) Berechne $z_i(u) \equiv z(u) \bmod p_i(u)$ für $i = 1, \ldots, L$.

b) Bilde geeignete Linearkombinationen der Koeffizienten der Polynome $z_i(u)$, um die Produkte $z_i(u)y_i(u) \bmod p_i(u)$ berechnen zu können $(i = 1, \ldots, L)$.

c) Bilde geeignete Linearkombinationen der Koeffizienten des Polynoms $\hat{z}(u)$, um das Produkt $\hat{z}(u)\hat{y}(u) \bmod u^s$ berechnen zu können, wobei $s := k + \delta - 1 - \deg(p(u))$ sei.

Die Schritte (b) und (c) entsprechen der Multiplikation des Informationswortes $\bar{z}$ mit der Generatormatrix $\mathbf{C}$.

Eine Darstellung der Schieberegister–Schaltungen für dieses Codierverfahren findet man in dem Buch von Morgera e. a. ([12, S. 132ff]).

2.6.4 Decodierung von KM–Codes

KM–Codes eignen sich zur Fehlererkennung und zur Fehlerkorrektur. Wir werden sie als innere, fehlerkorrigierende Codes in verallgemeinerten Typ-II Hybrid-ARQ-Systemen verwenden, deshalb betrachten wir hier nur die Decodierung zur Fehlerkorrektur. Eine Beschreibung des Verfahrens zur Fehlererkennung findet man in dem Buch von Morgera e. a. ([12, S. 134ff]). Bei der folgenden Erläuterung der Fehlerkorrektur orientieren wir uns ebenfalls an diesem Buch ([12, S. 144ff]). Wir verwenden die Bezeichnungen aus den Abschnitten 2.6.2 und 2.6.3.

Das Polynom $p(u)$ zerfällt in L paarweise teilerfremde Polynome $p_1(u), \ldots, p_L(u)$. Wir haben in Abschnitt 2.6.2 gesehen, daß die Generatormatrix $\mathbf{C}$ aus $L+1$ Blöcken $\mathbf{C}_1, \ldots, \mathbf{C}_{L+1}$ besteht. Die ersten L Blöcke entsprechen der Berechnung von

$$\phi_i(u) \equiv z_i(u)y_i(u) \bmod p_i(u), \quad i = 1, \ldots, L.$$

Der letzte Block $\mathbf{C}_{L+1}$ entspricht der Berechnung von

$$\hat{\phi}(u) \equiv \hat{z}(u)\hat{y}(u) \bmod u^s .$$

Es seien $\sigma_i := \deg(p_i(u))$ für $i = 1, \ldots, L$ und $\sigma_{L+1} := s$. Wir nehmen an, daß $k \geq \sigma_i$ und $\delta \geq \sigma_i$ für alle $i = 1, \ldots, L+1$ gilt, dies stellt keine wesentliche Einschränkung dar. Ferner bezeichne M_i die Anzahl der Spalten des Blockes $\mathbf{C}_i$, $i = 1, \ldots, L+1$.

Weiterhin nehmen wir an, daß die Berechnung von $\phi_i \equiv z_i(u)y_i(u) \bmod p_i(u)$ für $i = 1, \ldots, L$ bzw. von $\hat{\phi}(u) \equiv \hat{z}(u)\hat{y}(u) \bmod u^s$ in zwei Schritten erfolgt:

a) $\phi'_i(u) := z_i(u)y_i(u)$ bzw. $\hat{\phi}'(u) := \hat{z}(u)\hat{y}(u)$,

b) $\phi_i(u) \equiv \phi'_i(u) \bmod p_i(u)$ bzw. $\hat{\phi}(u) \equiv \hat{\phi}'(u) \bmod u^s$.

Der Schritt (a) liefert uns die Generatormatrix eines $(M_i, \sigma_i, \sigma_i)$-Codes (vgl. Abschnitt 2.6.2), diese wird im Schritt (b) nicht mehr verändert (es sei $\boldsymbol{\phi}_i = \mathbf{A}_i^T(\mathbf{C}_i^T\mathbf{z} \bullet \mathbf{B}_i\mathbf{y})$, dann wird in (b) nur $\mathbf{A}_i$ bestimmt). Falls $p_i(u)$ nicht irreduzibel ist, kann dieses Verfahren eine Erhöhung der Komplexität M_i bewirken. Nach Aussage von Morgera und Krishna gilt dies jedoch nicht für $\sigma_i \leq 2$; für $\sigma_i = 3$ oder $\sigma_i = 4$ wird M_i nur um 1 erhöht ([12, S. 145f]).

Ein empfangenes Wort können wir in $L+1$ Blöcke aufteilen, die zu den Matrizen $\mathbf{C}_1, \ldots, \mathbf{C}_{L+1}$ gehören, und jeden dieser Blöcke einzeln decodieren. Die entsprechenden Codes sind linear

und haben nur kleine Dimensionen, so daß wir hier eine einfache Maximum–Likelihood– oder Majority–Logic–Decodierung durchführen können. Im Falle eines Decodierversagens wird der betroffene Block ausgesondert. Wenn der i-te Block höchstens $\lfloor \frac{\sigma_i - 1}{2} \rfloor$ Fehler enthält, liefert die Decodierung das Polynom $z_i(u)$ für $i = 1, \ldots, L$ bzw. $\hat{z}(u) \bmod u^s$ für $i = L+1$.

Es sei α_i die Anzahl der Fehler im i-ten Block, $i = 1, \ldots, L+1$. Wir suchen ein Verfahren, um $z(u)$ rekonstruieren zu können, falls insgesamt weniger als $\lfloor \frac{\delta - 1}{2} \rfloor$ Fehler aufgetreten sind, d. h. falls

$$\sum_{i=1}^{L+1} \alpha_i \ \leq \ \left\lfloor \frac{\delta - 1}{2} \right\rfloor$$

ist. Hierzu dient uns der folgende Satz (s. [12, S. 147]):

Satz 2.47 *Falls die Anzahl der Fehler in dem empfangenen Wort höchstens $\lfloor \frac{\delta-1}{2} \rfloor$ beträgt und jeder der $L+1$ Blöcke wie oben beschrieben decodiert wurde, gibt es mindestens eine Menge von fehlerfrei decodierten Blöcken mit Indizes $\{i_1, \ldots, i_\kappa\}$, für die $\sum_{\nu=1}^{\kappa} \sigma_{i_\nu} \geq k$ gilt.*

Beweis: Es seien $i_1, \ldots, i_f$ die Indizes der falsch decodierten oder ausgesonderten Blöcke, d. h. es gilt $\alpha_{i_\nu} > \lfloor \frac{\sigma_{i_\nu} - 1}{2} \rfloor$ für $\nu = 1, \ldots, f$. Es sind also mindestens $\sum_{\nu=1}^{f} (\lfloor \frac{\sigma_{i_\nu} - 1}{2} \rfloor + 1)$ Fehler aufgetreten. Es gelte

$$\left\lfloor \frac{\delta - 1}{2} \right\rfloor \ \geq \ \sum_{\nu=1}^{f} \left(\left\lfloor \frac{\sigma_{i_\nu} - 1}{2} \right\rfloor + 1 \right) \ .$$

Falls δ gerade ist, impliziert dies

$$\frac{\delta - 2}{2} \ \geq \ \sum_{\nu=1}^{f} \left(\left\lfloor \frac{\sigma_{i_\nu} - 1}{2} \right\rfloor + 1 \right) \ \geq \ \sum_{\nu=1}^{f} \left(\frac{\sigma_{i_\nu} - 2}{2} + 1 \right) \ = \ \sum_{\nu=1}^{f} \frac{\sigma_{i_\nu}}{2} \ ,$$

und analog, falls δ ungerade ist:

$$\frac{\delta - 1}{2} \ \geq \ \sum_{\nu=1}^{f} \frac{\sigma_{i_\nu}}{2} \ .$$

Also gilt in beiden Fällen $\delta > \sum_{\nu=1}^{f} \sigma_{i_\nu}$. Wegen $\sum_{j=1}^{L+1} \sigma_j = \deg(p(u)) + s = k + \delta - 1$ folgt

$$\sum_{\substack{j=1 \\ j \neq i_1, \ldots, i_f}}^{L+1} \sigma_j \ = \ k + \delta - 1 - \sum_{\nu=1}^{f} \sigma_{i_\nu} \ \geq \ k \ .$$

Hieraus erhalten wir die Behauptung. q.e.d.

Bemerkung: Der Beweis zeigt, daß auch mehr als $\lfloor \frac{\delta-1}{2} \rfloor$ Fehler auftreten dürfen, es muß nur

$$\left\lfloor \frac{\delta - 1}{2} \right\rfloor \ \geq \ \sum_{\nu=1}^{f} \left(\left\lfloor \frac{\sigma_{i_\nu} - 1}{2} \right\rfloor + 1 \right)$$

gelten.

Es sei nun $\pi := \{1, 2, \ldots, L+1\}$. Wir bilden Teilmengen $\pi_1, \pi_2, \ldots$ von π so, daß jede Menge π_l minimal ist bezüglich der Eigenschaft $\sum_{j \in \pi_l} \sigma_j \geq k$.

Der Chinesische Restsatz (Satz 2.46) ermöglicht uns die Konstruktion eines Polynoms $z'_1(u)$ vom Grad $k_1 - 1 := \sum_{j \in \pi_1} \sigma_j - 1$ aus den Polynomen $z_i(u),\ i \in \pi_1$. Wenn wir annehmen, daß $z'_1(u) = \sum_{\nu=0}^{k_1-1} z'_{1\nu} u^\nu$ das übertragene Informationswort wäre, dann müßte $z'_{1\nu} = 0$ für $\nu = k, \ldots, k_1 - 1$ gelten. In diesem Fall ist $z'_1(u)$ ein Kandidat für das zu rekonstruierende Informationswort. Analoges gilt für die übrigen Mengen $\pi_2, \pi_3, \ldots$.

Es seien $z'_1(u), z'_2(u), \ldots$ die Kandidaten für das zu rekonstruierende Informationswort. Falls höchstens $\lfloor \frac{\delta-1}{2} \rfloor$ Fehler aufgetreten sind, sichert uns der Satz 2.47 die Existenz mindestens eines Kandidaten. Außerdem ist ein Kandidat $z'_i(u)$ genau dann das übertragene Informationswort, wenn sich das zugehörige Codewort in höchstens $\lfloor \frac{\delta-1}{2} \rfloor$ Stellen von dem empfangenen Wort unterscheidet. So erhalten wir den folgenden Algorithmus (s. [12, S. 148f]):

a) Partitioniere das empfangene Wort entsprechend den Blöcken $\mathbf{C}_1, \ldots, \mathbf{C}_{L+1}$ von $\mathbf{C}$.

b) Decodiere jeden Block einzeln.

c) Sondere die Blöcke aus, bei denen ein Decodierversagen auftrat.

d) Bilde die Kandidaten für das zu rekonstruierende Informationswort.

e) Bilde die Codewörter zu den Kandidaten aus Schritt (d).

f) Betrachte den Kandidaten als übertragenes Informationswort, dessen Codewort in höchstens $\lfloor \frac{\delta-1}{2} \rfloor$ Stellen von dem empfangenen Wort abweicht.

Falls mehr als $\lfloor \frac{\delta-1}{2} \rfloor$ Fehler aufgetreten sind, können wir eventuell auch mindestens einen Kandidaten für das zu rekonstruierende Informationswort erhalten (s. Bemerkung zu Satz 2.47). Im Sinne einer Maximum–Likelihood–Decodierung sollte dann im Schritt (f) ein Kandidat gewählt werden, dessen Codewort den minimalen Hamming–Abstand zu dem empfangenen Wort hat.

Eine Darstellung der Schieberegister–Schaltungen für dieses Decodierverfahren findet man in dem Buch von Morgera e. a. ([12, S. 152]).

2.6.5 Partitionierung der Generatormatrix

Wir wollen KM–Codes als innere Codes in verallgemeinerten Typ-II Hybrid-ARQ-Systemen einsetzen. Hierfür müssen wir die Generatormatrix eines (n, k, d)-KM–Codes, wobei $n = mk$ für ein $m \in \mathbb{N}$ sei, in m invertierbare $(k \times k)$-Matrizen partitionieren.

In den Abschnitten 2.6.2 und 2.6.4 haben wir bereits gesehen, daß wir die Generatormatrix $\mathbf{C}$ in $L+1$ Blöcke $\mathbf{C}_1, \ldots, \mathbf{C}_{L+1}$ aufteilen können, die jeweils einer Berechnung der Form $\phi_i(u) \equiv z_i(u) y_i(u) \bmod p_i(u)$ für $i = 1, \ldots, L$ bzw. $\hat{\phi}(u) \equiv \hat{z}(u)\hat{y}(u) \bmod u^s$ entsprechen. Es seien wieder $\sigma_i := \deg(p_i(u))$ für $i = 1, \ldots, L$ und $\sigma_{L+1} := s$, ferner sei M_i die Anzahl der Spalten des Blockes $\mathbf{C}_i$, $i = 1, \ldots, L+1$. Außerdem nehmen wir wieder an, daß $k \geq \sigma_i$ und $\delta \geq \sigma_i$ für alle $i = 1, \ldots, L+1$ gilt. Der Block $\mathbf{C}_i$, $i \in \{1, \ldots, L+1\}$ entspricht dann einer Berechnung, deren Ergebnis ein Polynom vom Grad $\sigma_i - 1$ ist. Deshalb enthält er σ_i linear unabhängige Spalten und die übrigen $M_i - \sigma_i$ Spalten sind Linearkombinationen hiervon (s. [12, S. 137]). Des weiteren können beliebige Spalten, die zu verschiedenen Blöcken gehören, linear unabhängig sein. Bei geeigneter Wahl der Polynome $p_1(u), \ldots, p_L(u)$ kann daher durch Umordnung der Spalten eine Partitionierung der Generatormatrix in invertierbare Untermatrizen erreicht werden. Leider nennen Krishna und Morgera in ihren Büchern ([8], [12]) keine Kriterien, um hierfür geeignete Polynome zu wählen.

Uns interessiert jetzt noch die Frage, was passiert, wenn wir einen Block $\mathbf{C}_i$ aus der Matrix $\mathbf{C}$ entfernen. Die so erhaltene Matrix $\mathbf{C}'$ hat dann die Größe $(k \times n')$ mit $n' = n - M_i$ und ist ebenfalls die Generatormatrix eines KM–Codes. Der Designabstand dieses Codes beträgt $\delta' = \delta - \sigma_i$, denn es muß

$$k + \delta' - 1 \;=\; \sum_{\substack{j=1 \\ j \neq i}}^{L+1} \sigma_j \;=\; k + \delta - 1 - \sigma_i$$

gelten (s. [12, S. 155f]). Bei der Decodierung bleiben die Schritte (a), (b) und (c) unverändert. Im Schritt (d) werden die Mengen π_l ausgelassen, die i enthalten. Im Schritt (e) erfolgt die Codierung mittels der Matrix $\mathbf{C}'$ und schließlich wird in (f) ein Codewort gewählt, das in höchstens $\lfloor \frac{\delta'-1}{2} \rfloor$ Stellen von dem empfangenen Wort abweicht (s. [12, S. 157]).

Wir können also durch die Wahl geeigneter Polynome $p_1(u), \ldots, p_L(u)$ und des Parameters s — und somit der Blöcke $\mathbf{C}_1, \ldots, \mathbf{C}_{L+1}$ — einen KM–Code $\mathcal{C}$ erhalten, dessen Generatormatrix nach einer Umordnung der Spalten in mehrere invertierbare Untermatrizen $\mathbf{D}_1, \ldots, \mathbf{D}_m$ aufgeteilt werden kann. Es sei $\mathcal{C}_i$ der von der Matrix $[\mathbf{D}_1 \mid \ldots \mid \mathbf{D}_i]$ erzeugte KM–Code, $i = 2, \ldots, m$, dann können wir für die Decodierung von $\mathcal{C}_i$ den $\mathcal{C}$-Decodierer verwenden. Diese Eigenschaft werden wir bei der Konstruktion der verallgemeinerten Typ-II Hybrid-ARQ-Systeme benötigen.

Beispiel

Zur Verdeutlichung wollen wir noch ein Beispiel betrachten (s. [8, S. 110ff]). Die Generatormatrix des $(12, 4, 5)$-KM–Codes ist gegeben durch

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Hierbei sind $p(u) = p_1(u)p_2(u)p_3(u) = u^2(u^2+1)(u^2+u+1)$ und $s = 2$.
Der Berechnung von $z_1(u)y_1(u) \bmod p_1(u)$ entspricht der Block

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

wobei die erste Spalte der Berechnung von $z_1(u)y_1(u) \bmod u$ entspricht. Ferner gehört der Block

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

zur Berechnung von $\hat{z}(u)\hat{y}(u) \bmod u^s$, wobei die erste Spalte zur Berechnung von $\hat{z}(u)\hat{y}(u) \bmod u$ gehört.

Wenn wir nun aus diesen beiden Blöcken jeweils die letzten zwei Spalten streichen, erhalten wir die Generatormatrix des durch $\tilde{p}(u) = u(u^2+1)(u^2+u+1)$ und $s = 1$ gegebenen $(8, 4, 3)$-KM–Codes (vgl. Abschnitt 2.6.2). Durch Spaltenvertauschungen bekommen wir als neue Generatormatrix des $(12, 4, 5)$-KM–Codes:

$$\left[\begin{array}{cccc|cccc|cccc} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right].$$

Die drei (4×4)-Untermatrizen sind invertierbar und die ersten beiden bilden eine Generatormatrix des $(8, 4, 3)$-KM–Codes. æ

Kapitel 3

ARQ-Systeme

In diesem Kapitel werden wir die Zuverlässigkeit und den Durchsatz von ARQ-Systemen untersuchen. Als Maß für die Zuverlässigkeit verwenden wir die *Restfehlerwahrscheinlichkeit*, d. h. die Wahrscheinlichkeit dafür, daß ein Block, der an den Empfänger übergeben wird, Fehler enthält.

Wir betrachten ein ARQ-System, für das zusätzlich zu den in Abschnitt 1.1 genannten die folgenden Annahmen gelten sollen:

- Der Übertragungskanal sei ein gedächtnisloser binärer symmetrischer Kanal mit Bitfehlerwahrscheinlichkeit $\varepsilon \in (0, \frac{1}{2}]$.
- Der Rückkopplungskanal sei ungestört.
- Der Sender und der Empfänger (bei Verwendung des Selective-Repeat-Protokolls) verfügen über unendlich große Speicher.
- Das ARQ-System verwende einen fehlererkennenden linearen (m, k, d)-Code $\mathcal{C}$ über $GF(2)$.

3.1 Zuverlässigkeit

Zunächst bestimmen wir die Decodierfehlerwahrscheinlichkeit des ARQ-Systems, d. h. die Wahrscheinlichkeit dafür, daß ein Übertragungsfehler nicht erkannt wird.

Angenommen, das Codewort $\bar{c} \in \mathcal{C}$ wurde gesendet, und empfangen wurde $\bar{r} = \bar{c} + \bar{e}$, wobei $\bar{e}$ ein Fehlermuster darstelle. Falls $\bar{e}$ nicht das Nullwort ist, ist ein Übertragungsfehler aufgetreten. Dieser wird genau dann entdeckt, wenn $\bar{r}$ kein Codewort ist, z. B. durch Berechnung des Syndroms[1]. Der Code $\mathcal{C}$ ist linear, daher gilt $\bar{r} \in \mathcal{C}$ genau dann, wenn $\bar{e} = \bar{r} - \bar{c} \in \mathcal{C}$ ist. Ein Decodierfehler tritt also genau dann auf, wenn $\bar{e}$ ein $\mathcal{C}$-Codewort ungleich dem Nullwort ist.

Bezeichnen P_c , P_e bzw. P_d die Wahrscheinlichkeit für richtige Decodierung, falsche Decodierung bzw. für das Entdecken eines Übertragungsfehlers, so gilt der folgende Satz (s. [4, S. 26] oder [16, S. 61] für Aussage (b)):

[1] Es sei $\mathbf{H}$ eine Kontrollmatrix von $\mathcal{C}$, dann ist das Syndrom $\bar{s}$ von $\bar{r}$ definiert durch $\bar{s} := \bar{r}\mathbf{H}^T = \bar{e}\mathbf{H}^T$, und es gilt $\bar{r} \in \mathcal{C}$ genau dann, wenn $\bar{s} = \bar{0}$ ist (s. [2, S. 324] oder [16, S. 51ff], vgl. Abschnitt 2.3.3).

Satz 3.1 *Es sei* $\{A_0, \ldots, A_m\}$ *die Gewichtsverteilung von* $\mathcal{C}$*, dann gilt:*

a) $P_c = (1-\varepsilon)^m$,

b) $P_e = \sum_{i=d}^{m} A_i \, \varepsilon^i \, (1-\varepsilon)^{m-i}$,

c) $P_d = \sum_{i=1}^{m} \left(\binom{m}{i} - A_i \right) \varepsilon^i \, (1-\varepsilon)^{m-i}$.

Beweis: Die Aussage (a) folgt direkt aus der Gedächtnislosigkeit des Übertragungskanals. Ein Fehlermuster $\bar{e}$ tritt mit der Wahrscheinlichkeit $\varepsilon^{w(\bar{e})}(1-\varepsilon)^{m-w(\bar{e})}$ auf und wird genau dann nicht erkannt, wenn $\bar{e} \in \mathcal{C} \setminus \{\bar{0}\}$ gilt. Wegen $A_i = 0$ für $i = 1, \ldots, d-1$ folgt:

$$P_e = \sum_{\bar{e} \in \mathcal{C} \setminus \{\bar{0}\}} \varepsilon^{w(\bar{e})}(1-\varepsilon)^{m-w(\bar{e})} = \sum_{i=d}^{m} A_i \, \varepsilon^i \, (1-\varepsilon)^{m-i} .$$

Daher gilt:

$$P_d = 1 - P_c - P_e = 1 - \sum_{i=0}^{m} A_i \, \varepsilon^i \, (1-\varepsilon)^{m-i} = \sum_{i=1}^{m} \left(\binom{m}{i} - A_i \right) \varepsilon^i \, (1-\varepsilon)^{m-i} .$$

q.e.d.

Bemerkung: Der in Aussage (c) auftretende Term $\binom{m}{i} - A_i$ ist die Anzahl der Binärwörter der Länge m mit Hamming–Gewicht i, die nicht zu $\mathcal{C}$ gehören.

Für kleine Bitfehlerraten verhält sich das ARQ-System gutartig:

Korollar 3.2 *Mit obigen Bezeichnungen gilt:*

$$\lim_{\varepsilon \to 0} P_c(\varepsilon) = 1 , \quad \lim_{\varepsilon \to 0} P_e(\varepsilon) = 0 \quad \textit{und} \quad \lim_{\varepsilon \to 0} P_d(\varepsilon) = 0 .$$

Beweis: Die Behauptung folgt unmittelbar aus Satz 3.1. □

Es ist oft sehr aufwendig, die Gewichtsverteilung von $\mathcal{C}$ zu bestimmen. Wir können die Decodierfehlerwahrscheinlichkeit P_e jedoch auch mit Hilfe der Gewichtsverteilung $\{B_0, \ldots, B_m\}$ des zu $\mathcal{C}$ dualen Codes $\mathcal{C}^\perp$ berechnen. Dies kann effizienter sein, falls m kleiner als $2k$ ist, weil der Code $\mathcal{C}$ 2^k Codewörter enthält, $\mathcal{C}^\perp$ jedoch nur 2^{m-k}. Für die Decodierfehlerwahrscheinlichkeit P_e gilt dann (s. [16, S. 62]):

Satz 3.3 *Mit obigen Bezeichnungen gilt:*

$$P_e = \frac{1}{2^{m-k}} \sum_{i=0}^{m} B_i \, (1-2\varepsilon)^i - (1-\varepsilon)^m .$$

Beweis: Mit den MacWilliams–Identitäten (Satz 2.37) erhalten wir aus Satz 3.1(b):

$$\begin{aligned} P_e &= \sum_{i=0}^{m} A_i \varepsilon^i (1-\varepsilon)^{m-i} - (1-\varepsilon)^m \\ &= \frac{1}{2^{m-k}} \sum_{i=0}^{m} B_i (1-2\varepsilon)^i - (1-\varepsilon)^m . \end{aligned}$$

q.e.d.

Die Kenntnis der Decodierfehlerwahrscheinlichkeit P_e ist noch nicht ausreichend, um die Zuverlässigkeit eines Übertragungssystems beurteilen zu können. Wir müssen berücksichtigen, daß ein Block eventuell erst nach mehreren Reübertragungen an den Empfänger weitergegeben wird. Wir betrachten deshalb die Restfehlerwahrscheinlichkeit P_R (s. [18, S. 602]):

Satz 3.4 *Für die Restfehlerwahrscheinlichkeit P_R eines ARQ-Systems gilt:*

$$P_R = \frac{P_e}{1-P_d} = \frac{P_e}{P_c + P_e} .$$

Beweis: Ein Block wird genau dann fehlerhaft an den Empfänger übergeben, wenn ein Übertragungsfehler nicht erkannt wurde. Dies kann bei der ersten Übertragung oder einer Reübertragung geschehen. Der Übertragungskanal ist gedächtnislos, daher gilt:

$$P_R = \sum_{i=0}^{\infty} {P_d}^i P_e = P_e \frac{1}{1-P_d} .$$

q.e.d.

Bemerkung: Die Restfehlerwahrscheinlichkeit ist der relative Anteil der fehlerhaft an den Empfänger übergegebenen Blöcke an allen an den Empfänger übergegebenen Blöcken.

3.2 Durchsatz

Die Informationsrate R des Codes $\mathcal{C}$ beträgt $\frac{k}{m}$ (s. [2, S. 312]). Die mittlere Anzahl V der Übertragungen eines Blockes und die Durchsatzrate

$$\begin{aligned} \eta &= \frac{\text{durchschnittliche Anzahl erfolgreich übertragener Informationsbits pro Zeiteinheit}}{\text{Anzahl insgesamt übertragener Bits pro Zeiteinheit}} \\ &= \frac{1}{V} \cdot R \end{aligned}$$

sind abhängig von dem zugrundeliegenden Protokoll. Wir werden hier das Selective-Repeat-ARQ und das Go-Back-M-ARQ untersuchen, wobei wir stets annehmen, daß sowohl beim Sender als auch beim Empfänger (bei Verwendung des Selective-Repeat-Protokolls) unendlich große Speicher zur Verfügung stehen.

3.2.1 Selective-Repeat-ARQ

Eine Übertragung muß genau dann wiederholt werden, wenn bei der Decodierung dieses Blockes ein Fehler entdeckt wurde. Die Durchsatzrate η_{SR} beträgt daher (s. [2, S. 334], [6, S. 486]):

$$\eta_{\mathrm{SR}} \;=\; (1-P_d)\frac{k}{m}\,. \tag{3.1}$$

Beweis:

$$V_{\mathrm{SR}} \;=\; \sum_{i=0}^{\infty}(i+1){P_d}^i(1-P_d) \;=\; (1-P_d)\frac{1}{(1-P_d)^2} \;=\; \frac{1}{1-P_d}\,.$$

q.e.d.

3.2.2 Go-Back-M-ARQ

Bei jeder Reübertragung werden zusätzlich $(M-1)$ Blöcke erneut übertragen. Als Durchsatzrate $\eta_{\mathrm{GB}M}$ erhalten wir deshalb (s. [2, S. 330], [6, S. 486]):

$$\eta_{\mathrm{GB}M} \;=\; \frac{1-P_d}{1+(M-1)P_d}\cdot\frac{k}{m}\,. \tag{3.2}$$

Beweis:

$$\begin{aligned} V_{\mathrm{GB}M} \;&=\; \sum_{i=0}^{\infty}(i+1+i(M-1)){P_d}^i\,(1-P_d) \;=\; (1-P_d)\sum_{i=0}^{\infty}(Mi+1){P_d}^i \\ &=\; (1-P_d)\left(M\frac{P_d}{(1-P_d)^2}+\frac{1}{1-P_d}\right) \;=\; \frac{1+(M-1)P_d}{1-P_d}\,. \end{aligned}$$

q.e.d.

Kapitel 4

Typ-I Hybrid-ARQ-Systeme

In diesem Kapitel werden wir die Zuverlässigkeit und den Durchsatz von Typ-I Hybrid-ARQ-Systemen (HI-ARQ) mit gewöhnlicher Codeverkettung (s. Abschnitt 1.4.2) untersuchen. Als Maß für die Zuverlässigkeit verwenden wir wieder die Restfehlerwahrscheinlichkeit (s. Kap. 3).

Wir betrachten ein HI-ARQ-System, für das zusätzlich zu den in Abschnitt 1.1 genannten die folgenden Annahmen gelten sollen:

- Der Übertragungskanal sei ein gedächtnisloser binärer symmetrischer Kanal mit Bitfehlerwahrscheinlichkeit $\varepsilon \in (0, \frac{1}{2}]$.
- Der Rückkopplungskanal sei ungestört.
- Der Sender und der Empfänger (bei Verwendung des Selective-Repeat-Protokolls) verfügen über unendlich große Speicher.
- Der äußere Code $\mathcal{C}_2$ sei ein fehlererkennender linearer (l, k, d_2)-Code über $GF(2)$.
- Der innere Code $\mathcal{C}_1$ sei ein fehlerkorrigierender linearer (n, l, d_1)-Code über $GF(2)$.

Nach Lemma 1.1 ist der durch gewöhnliche Codeverkettung von $\mathcal{C}_1$ und $\mathcal{C}_2$ definierte Code $\mathcal{C}$ ein linearer (n, k, d)-Code.

4.1 Zuverlässigkeit

Als Decodieralgorithmus für den fehlerkorrigierenden Code $\mathcal{C}_1$ soll ein Maximum–Likelihood–Verfahren verwendet werden, d. h. ein empfangenes Wort wird in ein $\mathcal{C}_1$-Codewort mit minimalem Hamming–Abstand decodiert. Falls es mehrere solcher $\mathcal{C}_1$-Codewörter gibt, wird ein beliebiges davon gewählt (s. [4, S. 24], [10, S. 26]). Außerdem werden die Redundanzbits des Codes $\mathcal{C}_1$ entfernt.

Für ein $\mathcal{C}_1$-Codewort $\bar{c}_1$ sei

$$Q_{\bar{c}_1} := \{\bar{x} \in (GF(2))^n \; : \; d(\bar{x}, \bar{c}_1) \leq d(\bar{x}, \bar{d}_1) \quad \forall\, \bar{d}_1 \in \mathcal{C}_1\} \tag{4.1}$$

die Menge der Wörter, die in $\bar{c}_1$ decodiert werden können.

Für ein Wort $\bar{x}$ sei

$$a(\bar{x}) := \sharp\{\bar{c}_1 \in \mathcal{C}_1 \; : \; d(\bar{x}, \bar{c}_1) \leq d(\bar{x}, \bar{d}_1) \quad \forall\, \bar{d}_1 \in \mathcal{C}_1\} \tag{4.2}$$

die Anzahl der $\mathcal{C}_1$-Codewörter mit minimalem Hamming–Abstand zu $\bar{x}$.

Angenommen, das $\mathcal{C}$-Codewort $\bar{c}$ wurde gesendet, und empfangen wurde das Wort $\bar{r} = \bar{c} + \bar{e}$, wobei $\bar{e}$ ein Fehlermuster darstelle. Für die Wahrscheinlichkeit $p\,(\bar{d}, \bar{c})$, daß $\bar{r}$ in ein $\mathcal{C}_1$-Codewort $\bar{d}$ decodiert wird, folgt aus der obigen Erklärung des Decodierverfahrens (s. [7, S. 513]):

$$p\,(\bar{d}, \bar{c}) \; = \; \sum_{\bar{x} \in Q_{\bar{d}}} \frac{1}{a(\bar{x})}\, \varepsilon^{d(\bar{c},\bar{x})}\, (1-\varepsilon)^{n-d(\bar{c},\bar{x})} \; .$$

Die Decodierung ist also mit der Wahrscheinlichkeit $p\,(\bar{c}, \bar{c})$ richtig.

Wie wir in Abschnitt 3.1 gesehen haben, wird ein Decodierfehler des inneren FEC-Systems von dem äußeren ARQ-System genau dann nicht erkannt, wenn $\bar{r}$ in ein $\mathcal{C}$-Codewort $\bar{d}$ ungleich $\bar{c}$ decodiert wird.

Bezeichnen wieder P_c, P_e bzw. P_d die Wahrscheinlichkeit für richtige Decodierung, falsche Decodierung bzw. für das Entdecken eines Übertragungsfehlers, so erhalten wir den folgenden Satz (s. [7, S. 513] für Aussage (b)):

Satz 4.1 *Für das HI-ARQ-System gilt mit obigen Bezeichnungen:*

a) $$P_c \;=\; \sum_{\bar{x} \in Q_{\bar{0}}} \frac{1}{a(\bar{x})}\, \varepsilon^{w(\bar{x})}\, (1-\varepsilon)^{n-w(\bar{x})} \; ,$$

b) $$P_e \;=\; \sum_{\bar{d} \in \mathcal{C} \setminus \{\bar{0}\}} \; \sum_{\bar{x} \in Q_{\bar{d}}} \frac{1}{a(\bar{x})}\, \varepsilon^{w(\bar{x})}\, (1-\varepsilon)^{n-w(\bar{x})} \; ,$$

c) $$P_d \;=\; 1 - \sum_{\bar{d} \in \mathcal{C}} \; \sum_{\bar{x} \in Q_{\bar{d}}} \frac{1}{a(\bar{x})}\, \varepsilon^{w(\bar{x})}\, (1-\varepsilon)^{n-w(\bar{x})} \; .$$

Beweis: Wegen der Linearität von $\mathcal{C}$ und der Translations–Invarianz des Hamming–Abstands[1] können wir ohne Beschränkung der Allgemeinheit annehmen, daß das Nullwort $\bar{0}$ übertragen

[1] Der Hamming–Abstand d ist auf $(GF(2))^n$ translations–invariant, d. h. für je drei Wörter $\bar{x}, \bar{y}, \bar{z} \in (GF(2))^n$ gilt stets $d(\bar{x}, \bar{y}) = d(\bar{x} + \bar{z}, \bar{y} + \bar{z})$ (Quelle: [5, S. 204]).

wurde. Aus den obigen Überlegungen folgt:

$$\begin{aligned} P_c &= p(\bar{0},\bar{0}) = \sum_{\bar{x}\in Q_{\bar{0}}} \frac{1}{a(\bar{x})}\,\varepsilon^{w(\bar{x})}\,(1-\varepsilon)^{n-w(\bar{x})}\,, \\ P_e &= \sum_{\bar{d}\in\mathcal{C}\setminus\{\bar{0}\}} p(\bar{d},\bar{0}) = \sum_{\bar{d}\in\mathcal{C}\setminus\{\bar{0}\}}\sum_{\bar{x}\in Q_{\bar{d}}} \frac{1}{a(\bar{x})}\,\varepsilon^{w(\bar{x})}\,(1-\varepsilon)^{n-w(\bar{x})}\,, \\ P_d &= 1-P_c-P_e = 1-\sum_{\bar{d}\in\mathcal{C}}\sum_{\bar{x}\in Q_{\bar{d}}} \frac{1}{a(\bar{x})}\,\varepsilon^{w(\bar{x})}\,(1-\varepsilon)^{n-w(\bar{x})}\,. \end{aligned}$$

q.e.d.

Für kleine Bitfehlerraten verhält sich das HI-ARQ-System gutartig:

Korollar 4.2 *Mit obigen Bezeichnungen gilt:*

$$\lim_{\varepsilon\to 0} P_c(\varepsilon) = 1\,, \quad \lim_{\varepsilon\to 0} P_e(\varepsilon) = 0 \quad \textit{und} \quad \lim_{\varepsilon\to 0} P_d(\varepsilon) = 0\,.$$

Beweis: Die Behauptung folgt unmittelbar aus Satz 4.1. □

Für die Restfehlerwahrscheinlichkeit P_R erhalten wir wie in Abschnitt 3.1, Satz 3.4:

$$P_R = \frac{P_e}{1-P_d} = \frac{P_e}{P_c+P_e}\,. \tag{4.3}$$

4.1.1 Perfekte Codes als innere Codes

Falls $\mathcal{C}_1$ ein perfekter Code ist (z. B. ein Hamming–Code), gilt $d_1 = 2t_1+1$ für ein $t_1 \in \mathbb{N}$ und aus Satz 2.32 folgt:

$$Q_{\bar{c}_1} = \{\bar{x}\in (GF(2))^n \,:\, d(\bar{x},\bar{c}_1)\le t_1\} \qquad \forall\,\bar{c}_1\in\mathcal{C}_1\,, \tag{4.4}$$

$$a(\bar{x}) = 1 \qquad \forall\,\bar{x}\in(GF(2))^n\,. \tag{4.5}$$

Es sei $\{A_0,\ldots,A_n\}$ die Gewichtsverteilung von $\mathcal{C}$. Für ein festes $\mathcal{C}$-Codewort $\bar{c}$ mit Hamming–Gewicht i sei

$$W_{j,s}^{(i)} := \sharp\{\bar{x}\in(GF(2))^n \,:\, w(\bar{x})=j\,,\, d(\bar{x},\bar{c})=s\} \tag{4.6}$$

die Anzahl der Wörter mit Hamming–Gewicht j und Hamming–Abstand s von $\bar{c}$ (s. [6, S. 482]). Die Wohldefiniertheit der $W_{j,s}^{(i)}$ folgt aus der Linearität von $\mathcal{C}$ und der Translations–Invarianz des Hamming–Abstands; das folgende Lemma zeigt uns, wie sie berechnet werden können (s. [6, S. 482] für Aussage (b)):

Lemma 4.3 *Mit obigen Bezeichnungen gilt:*

a) $$W_{j,s}^{(0)} = \begin{cases} \binom{n}{j} & \textit{falls} \quad j=s \\ 0 & \textit{falls} \quad j\neq s \end{cases}\,,$$

$$b) \quad W_{j,s}^{(i)} \;=\; \begin{cases} \binom{n-i}{\frac{1}{2}(j+s-i)}\binom{i}{\frac{1}{2}(j+i-s)} & \textit{falls} \quad j+s-i \textit{ gerade ist} \\ 0 & \textit{falls} \quad j+s-i \textit{ ungerade ist} \end{cases} \qquad (i \in \mathbb{N}).$$

Beweis: Die Aussage (a) folgt direkt aus der Definition der $W_{j,s}^{(i)}$.
Zu (b): Es seien $\bar{a} = (a_1, \ldots, a_n)$, $\bar{b} = (b_1, \ldots, b_n) \in (GF(2))^n$ mit $w(\bar{a}) = i$ und $w(\bar{b}) = j$. Ferner seien $q := \sharp\{\nu : a_\nu = 0,\, b_\nu = 1\}$ und $r := \sharp\{\nu : a_\nu = 1,\, b_\nu = 0\}$, dann folgt

$$r \;=\; w(\bar{a}) - \sharp\{\nu : a_\nu = 1,\, b_\nu = 1\} \;=\; i - (j - q)\,.$$

Wegen $d(\bar{a}, \bar{b}) = q + r = 2q + i - j$ gilt $d(\bar{a}, \bar{b}) = s$ genau dann, wenn $q = \frac{1}{2}(s - i + j)$ ist. Damit erhalten wir $W_{j,s}^{(i)} = 0$, falls $\frac{1}{2}(s - i + j) \notin \mathbb{N}$, und $W_{j,s}^{(i)} = \binom{n-i}{q}\binom{i}{j-q}$ sonst. q.e.d.

Lemma 4.4 *Die erzeugende Funktion der $W_{j,s}^{(i)}$ ist gegeben durch:*

$$\sum_{j=0}^{n}\sum_{s=0}^{n} W_{j,s}^{(i)}\, x^j\, y^s \;=\; (1+xy)^{n-i}(x+y)^i\,, \quad x, y \in \mathbb{R}\,, \quad i = 0, \ldots, n\,.$$

Quelle: Kasami e. a. ([6, S. 482]) verweisen auf MacWilliams (1963).

Mit (4.4) und (4.5) folgt aus Satz 4.1 (analog zu [6, S. 482f]):

$$P_c \;=\; \sum_{i=0}^{t_1} \binom{n}{i} \varepsilon^i\, (1-\varepsilon)^{n-i}\,, \tag{4.7}$$

$$P_e \;=\; \sum_{i=d_1}^{n} A_i \sum_{s=0}^{t_1}\sum_{j=0}^{n} W_{j,s}^{(i)}\, \varepsilon^j\, (1-\varepsilon)^{n-j}\,. \tag{4.8}$$

Wir können die Decodierfehlerwahrscheinlichkeit P_e auch wieder mit Hilfe der Gewichtsverteilung $\{B_0, \ldots, B_n\}$ des zu $\mathcal{C}$ dualen Codes $\mathcal{C}^\perp$ berechnen. Hierfür benötigen wir die Definition der Krawtchouk–Polynome.

Definition 4.5 (Krawtchouk–Polynome) *Für $s \in \mathbb{N}_0$ und $n, q \in \mathbb{N}$ sei das* Krawtchouk–Polynom $K_s(x; n, q)$ *definiert durch*

$$K_s(x; n, q) \;:=\; \sum_{j=0}^{s} (-1)^j \binom{x}{j}\binom{n-x}{s-j}(q-1)^{s-j} \qquad (x \in \mathbb{R}).$$

Dabei sei

$$\binom{x}{j} \;:=\; \begin{cases} \dfrac{x(x-1)\cdot\ldots\cdot(x-j+1)}{j!} & \textit{falls } j \in \mathbb{N}\,, \\ 1 & \textit{falls } j = 0\,, \\ 0 & \textit{sonst.} \end{cases}$$

Bemerkung: Im Fall $q = 2$ gilt:

$$K_s(x;n) \;:=\; K_s(x;n,2) \;=\; \sum_{j=0}^{s}(-1)^j \binom{x}{j}\binom{n-x}{s-j} \qquad (x \in \mathbb{R}).$$

Quellen: Van Lint ([10, S. 15]), MacWilliams e. a. ([11, S. 130]).

Lemma 4.6 *Es seien $x \in \mathbb{R}$ und $n, q, t \in \mathbb{N}$, dann gilt:*

a) $$\sum_{j=0}^{t} K_j(x;n,q) \;=\; K_t(x-1;n-1,q)\,,$$

b) $$(1+x)^{n-i}(1-x)^i \;=\; \sum_{j=0}^{n} K_j(i;n)\, x^j \quad \textit{für } i \in \mathbb{N}_0\,,\ i \le n \textit{ und } q = 2\,.$$

Quellen: Van Lint ([10, S. 17]) für Aussage (a) und MacWilliams e. a. ([11, S. 130]) für (b).

Analog zu einem Ergebnis von Kasami e. a. für Typ-I Hybrid-ARQ-Systeme mit Codeverkettung mit Partitionierung ([6]) erhalten wir den folgenden Satz:

Satz 4.7 *Falls $\mathcal{C}_1$ ein perfekter Code ist, gilt mit obigen Bezeichnungen:*

a) $$P_c \;=\; \sum_{i=0}^{t_1} \binom{n}{i} \varepsilon^i\,(1-\varepsilon)^{n-i}\,,$$

b) $$P_c + P_e \;=\; \frac{1}{2^{n-k}} \sum_{i=0}^{n} B_i\,(1-2\varepsilon)^i\, K_{t_1}(i-1;n-1)\,.$$

Beweis: Die Aussage (a) haben wir bereits gezeigt (s. (4.7)).
Wegen $A_0 = 1$ und $A_i = 0$ für $i = 1, \ldots, d-1$ gilt nach (4.7), (4.8) und Lemma 4.3(a):

$$P_c + P_e = \sum_{i=0}^{n} A_i \sum_{s=0}^{t_1}\sum_{j=0}^{n} W_{j,s}^{(i)} \varepsilon^j\,(1-\varepsilon)^{n-j}\,. \tag{4.9}$$

Aus Lemma 4.4 und den MacWilliams–Identitäten (Satz 2.37) folgt für $x, y \in \mathbb{R}$:

$$\begin{aligned}
\sum_{i=0}^{n} A_i \sum_{j=0}^{n}\sum_{s=0}^{n} W_{j,s}^{(i)}\, x^j\, y^s \;&=\; \sum_{i=0}^{n} A_i\,(1+xy)^{n-i}(x+y)^i \\
&=\; \frac{1}{2^{n-k}} \sum_{i=0}^{n} B_i\,(1+xy+x+y)^{n-i}(1+xy-x-y)^i \\
&=\; \frac{1}{2^{n-k}} \sum_{i=0}^{n} B_i\,(1+x)^{n-i}(1-x)^i(1+y)^{n-i}(1-y)^i\,.
\end{aligned}$$

Mit Lemma 4.6(b) ergibt sich:

$$\sum_{i=0}^{n} A_i \sum_{j=0}^{n} \sum_{s=0}^{n} W_{j,s}^{(i)}\, x^j\, y^s \;=\; \frac{1}{2^{n-k}} \sum_{i=0}^{n} B_i\,(1+x)^{n-i}(1-x)^i \sum_{j=0}^{n} K_j(i;n)\, y^j \;.$$

Durch Koeffizientenvergleich erhalten wir:

$$\sum_{i=0}^{n} A_i \sum_{j=0}^{n} \sum_{s=0}^{t_1} W_{j,s}^{(i)}\, x^j\, y^s \quad = \quad \frac{1}{2^{n-k}} \sum_{i=0}^{n} B_i\,(1+x)^{n-i}(1-x)^i \sum_{j=0}^{t_1} K_j(i;n)\, y^j \;.$$

Setzen wir nun $x := \dfrac{\varepsilon}{1-\varepsilon}$ und $y := 1$, so folgt mit Lemma 4.6(a):

$$\sum_{i=0}^{n} A_i \sum_{j=0}^{n} \sum_{s=0}^{t_1} W_{j,s}^{(i)} \left(\frac{\varepsilon}{1-\varepsilon}\right)^j \quad = \quad \frac{1}{2^{n-k}} \sum_{i=0}^{n} B_i \left(\frac{1}{1-\varepsilon}\right)^{n-i} \left(\frac{1-2\varepsilon}{1-\varepsilon}\right)^i K_{t_1}(i-1;n-1) \;.$$

Multiplizieren wir diese Gleichung mit $(1-\varepsilon)^n$, so erhalten wir mit (4.9) die Aussage (b). q.e.d.

4.2 Durchsatz

Wie wir bei der Einführung der Hybrid-ARQ-Systeme in Abschnitt 1.4 gesehen haben, verhält sich ein HI-ARQ- wie ein normales ARQ-System mit verbessertem Übertragungskanal. Daher können wir hier die entsprechenden Ergebnisse für ARQ-Systeme (s. Abschnitt 3.2) anwenden, wobei wir jedoch beachten müssen, daß die Wahrscheinlichkeit für eine Reübertragung von dem jeweiligen Übertragungssystem abhängt. Wir nehmen wieder an, daß beim Sender und beim Empfänger (bei Verwendung des Selective-Repeat-Protokolls) unendlich große Speicher zur Verfügung stehen.

Wir erhalten also für die Durchsatzrate η bei Verwendung des Selective-Repeat-Protokolls

$$\eta_{\mathrm{SR}} \;=\; (1-P_d)\,\frac{k}{n} \tag{4.10}$$

und bei Verwendung des Go-Back-N-Protokolls

$$\eta_{\mathrm{GB}N} \;=\; \frac{1-P_d}{1+(N-1)P_d} \cdot \frac{k}{n} \;. \tag{4.11}$$

4.3 Vergleich mit ARQ-Systemen: Die Kløve–Miller–Bedingung

Ein HI-ARQ-System ist technisch aufwendiger als ein einfaches ARQ-System. Es stellt sich die Frage, unter welchen Voraussetzungen dieser zusätzliche Aufwand gerechtfertigt ist. Wir werden deshalb im folgenden die Zuverlässigkeit eines ARQ- und eines HI-ARQ-Systems mit gewöhnlicher Codeverkettung vergleichen. Dabei werden wir uns auf „kleine“ Bitfehlerwahrscheinlichkeiten konzentrieren.

Es sei $\mathcal{C}_2$ ein fehlererkennender linearer (l, k, d_2)-Code über GF(2) mit Gewichtsverteilung $\{A_0^{(2)}, \ldots, A_l^{(2)}\}$. Nach Satz 3.1 gilt für ein ARQ-System, das $\mathcal{C}_2$ verwendet:

$$P_c^{\mathrm{ARQ}}(\varepsilon) = (1-\varepsilon)^l \,, \tag{4.12}$$

$$P_e^{\mathrm{ARQ}}(\varepsilon) = \sum_{i=d_2}^{l} A_i^{(2)}\, \varepsilon^i\, (1-\varepsilon)^{l-i} \,, \tag{4.13}$$

$$P_d^{\mathrm{ARQ}}(\varepsilon) = 1 - P_c^{\mathrm{ARQ}}(\varepsilon) - P_e^{\mathrm{ARQ}}(\varepsilon) \,. \tag{4.14}$$

Ferner seien $\mathcal{C}_1$ ein fehlerkorrigierender linearer (n, l, d_1)-Code über $GF(2)$ und $\mathcal{C}$ der durch gewöhnliche Codeverkettung von $\mathcal{C}_1$ und $\mathcal{C}_2$ definierte lineare (n, k, d)-Code, dann gilt nach Satz 4.1 für ein HI-ARQ-System, das $\mathcal{C}$ verwendet:

$$P_c^{\mathrm{HI}}(\varepsilon) = \sum_{\bar{x} \in Q_{\bar{0}}} \frac{1}{a(\bar{x})} \varepsilon^{w(\bar{x})}\, (1-\varepsilon)^{n-w(\bar{x})} \,, \tag{4.15}$$

$$P_e^{\mathrm{HI}}(\varepsilon) = \sum_{\bar{d} \in \mathcal{C} \setminus \{\bar{0}\}} \sum_{\bar{x} \in Q_{\bar{d}}} \frac{1}{a(\bar{x})} \varepsilon^{w(\bar{x})}\, (1-\varepsilon)^{n-w(\bar{x})} \,, \tag{4.16}$$

$$P_d^{\mathrm{HI}}(\varepsilon) = 1 - P_c^{\mathrm{HI}}(\varepsilon) - P_e^{\mathrm{HI}}(\varepsilon) \,. \tag{4.17}$$

Zum Vergleich der Decodierfehlerwahrscheinlichkeiten betrachten wir den Quotienten

$$R_e(\varepsilon) := \frac{P_e^{\mathrm{HI}}(\varepsilon)}{P_e^{\mathrm{ARQ}}(\varepsilon)} \,.$$

Wir suchen ein Kriterium für die Wahl von $\mathcal{C}_1$ und $\mathcal{C}_2$, um zu erreichen, daß $R_e(\varepsilon)$ für hinreichend kleine ε „klein“ wird. Wir werden sehen, daß unter gewissen Voraussetzungen $\lim_{\varepsilon \to 0} R_e(\varepsilon) = 0$ gilt.

Falls die Bitfehlerwahrscheinlichkeit $\frac{1}{2}$ beträgt, ist die Decodierfehlerwahrscheinlichkeit des ARQ- gleich der des HI-ARQ-Systems (s. [7, S. 513]). In diesem Fall hat das ARQ-System aber eine höhere Durchsatzrate als das HI-ARQ-System. Bei der Verwendung des Go–Back–Protokolls ist zu beachten, daß die Rückgrifftiefe L (ARQ) bzw. N (HI-ARQ) von der Länge des zu übertragenden Codewortes und der Komplexität des Decodierverfahrens abhängt. Das bedeutet, daß hier $L \leq N$ gilt.

Satz 4.8 *Mit obigen Bezeichnungen gilt:*

a) $P_c^{\mathrm{ARQ}}\left(\frac{1}{2}\right) = P_c^{\mathrm{HI}}\left(\frac{1}{2}\right)$, $P_e^{\mathrm{ARQ}}\left(\frac{1}{2}\right) = P_e^{\mathrm{HI}}\left(\frac{1}{2}\right)$ *und* $P_d^{\mathrm{ARQ}}\left(\frac{1}{2}\right) = P_d^{\mathrm{HI}}\left(\frac{1}{2}\right)$,

b) $\eta_{\mathrm{SR}}^{\mathrm{ARQ}}\left(\frac{1}{2}\right) > \eta_{\mathrm{SR}}^{\mathrm{HI}}\left(\frac{1}{2}\right)$ *und* $\eta_{\mathrm{GBL}}^{\mathrm{ARQ}}\left(\frac{1}{2}\right) > \eta_{\mathrm{GB}N}^{\mathrm{HI}}\left(\frac{1}{2}\right)$ *, falls* $N \geq L$ *ist.*

Beweis: Nach (4.12) und (4.13) gilt $P_c^{\mathrm{ARQ}}\left(\frac{1}{2}\right) = \left(\frac{1}{2}\right)^l$ und $P_e^{\mathrm{ARQ}}\left(\frac{1}{2}\right) = \left(\frac{1}{2}\right)^l (2^k - 1)$. Andererseits ist wegen der Linearität von $\mathcal{C}_1$ und der Translations–Invarianz des Hamming–Abstands die Größe $\sum_{\bar{x}\in Q_{\bar{c}_1}} \frac{1}{a(\bar{x})}$ unabhängig von dem speziell gewählten $\bar{c}_1 \in \mathcal{C}_1$. Daher gilt:

$$2^l \sum_{\bar{x}\in Q_{\bar{c}_1}} \frac{1}{a(\bar{x})} = \sum_{\bar{c}_1\in\mathcal{C}_1} \sum_{\bar{x}\in Q_{\bar{c}_1}} \frac{1}{a(\bar{x})} = 2^n .$$

Dies impliziert:

$$\sum_{\bar{x}\in Q_{\bar{c}_1}} \frac{1}{a(\bar{x})} = 2^{n-l} \qquad \forall\, \bar{c}_1 \in \mathcal{C}_1 .$$

Mit (4.15) und (4.16) folgt:

$$P_c^{\mathrm{HI}}\left(\tfrac{1}{2}\right) = \left(\frac{1}{2}\right)^n \sum_{\bar{x}\in Q_{\bar{0}}} \frac{1}{a(\bar{x})} = \left(\frac{1}{2}\right)^l = P_c^{\mathrm{ARQ}}\left(\tfrac{1}{2}\right)$$

und

$$P_e^{\mathrm{HI}}\left(\tfrac{1}{2}\right) = \left(\frac{1}{2}\right)^n \sum_{\bar{d}\in\mathcal{C}\setminus\{\bar{0}\}} \sum_{\bar{x}\in Q_{\bar{d}}} \frac{1}{a(\bar{x})} = \left(\frac{1}{2}\right)^n \left(2^k - 1\right) 2^{n-l} = \left(\frac{1}{2}\right)^l \left(2^k - 1\right) = P_e^{\mathrm{ARQ}}\left(\tfrac{1}{2}\right) .$$

Also gilt auch $P_d^{\mathrm{ARQ}}\left(\frac{1}{2}\right) = P_d^{\mathrm{HI}}\left(\frac{1}{2}\right)$.
$\mathcal{C}_1$ ist ein fehlerkorrigierender (n, l)-Code, daher gilt $n > l$. Mit den Ergebnissen aus den Abschnitten 3.2 und 4.2 folgt:

$$\frac{\eta_{\mathrm{SR}}^{\mathrm{ARQ}}\left(\frac{1}{2}\right)}{\eta_{\mathrm{SR}}^{\mathrm{HI}}\left(\frac{1}{2}\right)} = \frac{\frac{k}{l}}{\frac{k}{n}} = \frac{n}{l} > 1$$

und

$$\frac{\eta_{\mathrm{GBL}}^{\mathrm{ARQ}}\left(\frac{1}{2}\right)}{\eta_{\mathrm{GB}N}^{\mathrm{HI}}\left(\frac{1}{2}\right)} = \frac{\frac{k}{l}\cdot\frac{1-P_d^{\mathrm{ARQ}}\left(\frac{1}{2}\right)}{1+(L-1)P_d^{\mathrm{ARQ}}\left(\frac{1}{2}\right)}}{\frac{k}{n}\cdot\frac{1-P_d^{\mathrm{HI}}\left(\frac{1}{2}\right)}{1+(N-1)P_d^{\mathrm{HI}}\left(\frac{1}{2}\right)}} = \frac{n}{l}\cdot\frac{1+(N-1)P_d^{\mathrm{HI}}\left(\frac{1}{2}\right)}{1+(L-1)P_d^{\mathrm{ARQ}}\left(\frac{1}{2}\right)} > 1 .$$

q.e.d.

Eine Verringerung der Decodierfehlerwahrscheinlichkeit können wir also nur für Bitfehlerwahrscheinlichkeiten $\varepsilon < \frac{1}{2}$ erreichen.

Eine wichtige Rolle spielt hierbei das minimale Gewicht Δ eines Wortes, das bei Maximum–Likelihood–Decodierung in ein $\mathcal{C}$-Codewort ungleich dem Nullwort decodiert werden kann:

$$\Delta \;:=\; \min\{w(\bar{x}) \;:\; \bar{x} \in Q_{\bar{c}} \text{ für ein } \bar{c} \in \mathcal{C} \setminus \{\bar{0}\}\}\,.$$

Ferner bezeichne ρ_1 den Überdeckungsradius von $\mathcal{C}_1$, d. h. (s. [10, S. 32], vgl. auch Def. 2.31):

$$\rho_1 \;=\; \max_{\bar{x}\in(GF(2))^n} \min_{\bar{c}_1\in\mathcal{C}_1} d(\bar{x},\bar{c}_1)\,.$$

Die Größe Δ läßt sich wie folgt abschätzen ([7, S. 512ff]):

Lemma 4.9 *Mit obigen Bezeichnungen gilt:*

a) $\Delta \;\geq\; d-\rho_1\,,$

b) $\Delta \;\geq\; \left\lceil\frac{d}{2}\right\rceil \;\geq\; \left\lceil\frac{d_1}{2}\right\rceil\,,$
wobei $\lceil\cdot\rceil$ die obere Gaußklammer bezeichne,

c) $\Delta \;\leq\; d-\left\lfloor\frac{d_1}{2}\right\rfloor\,,$
wobei $\lfloor\cdot\rfloor$ die untere Gaußklammer bezeichne.

Beweis: Zu (a): Es sei $\bar{x} \in Q_{\bar{c}}$ für ein $\bar{c} \in \mathcal{C}\setminus\{\bar{0}\}$, dann gilt $d(\bar{x},\bar{c}) \leq d(\bar{x},\bar{d}_1)$ für alle $\bar{d}_1 \in \mathcal{C}_1$. Dies impliziert: $\rho_1 \geq d(\bar{x},\bar{c})$. Mit der Dreiecksungleichung erhalten wir

$$w(\bar{x})+\rho_1 \;\geq\; d(\bar{0},\bar{x})+d(\bar{x},\bar{c}) \;\geq\; d(\bar{0},\bar{c}) \;\geq\; d\,.$$

Also folgt $\Delta \;\geq\; d-\rho_1$.

Zu (b): Es sei $\bar{y} \in (GF(2))^n$ mit $w(\bar{y}) < \frac{d}{2}$, dann gilt für alle $\bar{c} \in \mathcal{C}\setminus\{\bar{0}\}$:

$$d(\bar{y},\bar{0}) \;<\; \tfrac{1}{2}d(\bar{c},\bar{0}) \;\leq\; \tfrac{1}{2}[d(\bar{c},\bar{y})+d(\bar{y},\bar{0})]\,.$$

Hieraus folgt $d(\bar{y},\bar{0}) < d(\bar{y},\bar{c})$ für alle $\bar{c} \in \mathcal{C}\setminus\{\bar{0}\}$. Weil $\bar{0}$ ein $\mathcal{C}$-Codewort ist, gilt $\bar{y} \notin Q_{\bar{c}}$ für alle $\bar{c} \in \mathcal{C}\setminus\{\bar{0}\}$. Es sei nun $\bar{x} \in Q_{\bar{c}}$ für ein $\bar{c} \in \mathcal{C}\setminus\{0\}$, dann muß $w(\bar{x}) \geq \frac{d}{2}$ gelten. Wegen $\Delta \in I\!N$ folgt $\Delta \;\geq\; \left\lceil\frac{d}{2}\right\rceil$. Nach Lemma 1.1 gilt ferner $d \geq d_1$.

Zu (c): Es sei $\bar{c} \in \mathcal{C}\setminus\{\bar{0}\}$ mit $w(\bar{c}) = d$. Wegen $d \geq d_1$ enthält $\bar{c}$ mindestens d_1 Einsen. Wenn wir in $\bar{c}$ $\lfloor\frac{d_1}{2}\rfloor$ Einsen durch Nullen ersetzen, erhalten wir ein Wort $\bar{y}$ mit $d(\bar{y},\bar{c}) \leq \frac{d_1}{2}$. Daher gilt $\bar{y} \in Q_{\bar{c}}$ und es folgt $\Delta \leq w(\bar{y}) = d-\lfloor\frac{d_1}{2}\rfloor$. q.e.d.

Wir können nun eine Aussage über das Verhalten von $R_e(\varepsilon)$ für „kleine“ ε machen, dieses Ergebnis stammt aus einer Arbeit von Kløve und Miller ([7]). Die daraus abgeleitete hinreichende Bedingung, um $\lim_{\varepsilon\to 0} R_e(\varepsilon) = 0$ zu erhalten ([7, S. 512]), wird in der Literatur als „Kløve–Miller–Bedingung“ bezeichnet ([15]). Sie stellt ein einfach zu überprüfendes Kriterium für die Wahl der Codes zur Konstruktion eines HI-ARQ-Systems mit gewöhnlicher Codeverkettung dar.

Satz 4.10 *Mit obigen Bezeichnungen gilt:*

a) $\Delta > d_2$ *impliziert* $\lim\limits_{\varepsilon\to 0} R_e(\varepsilon) = 0$,

b) $\Delta = d_2$ *impliziert* $0 < \lim\limits_{\varepsilon\to 0} R_e(\varepsilon) < \infty$,

c) $\Delta < d_2$ *impliziert* $\lim\limits_{\varepsilon\to 0} R_e(\varepsilon) = \infty$.

Beweis: Für $i = 1, \ldots, n$ sei $\alpha_i := \sum\limits_{\bar{d}\in\mathcal{C}\setminus\{\bar{0}\}} \sum\limits_{\substack{\bar{x}\in Q_{\bar{d}} \\ w(\bar{x})=i}} \frac{1}{a(\bar{x})}$.

Nach Definition von Δ gilt $\alpha_i = 0$ für $i = 1, \ldots, \Delta - 1$ und mit (4.16) folgt:

$$P_e^{\mathrm{HI}}(\varepsilon) \quad = \quad \sum_{i=\Delta}^{n} \alpha_i \, \varepsilon^i \, (1-\varepsilon)^{n-i} \, .$$

Ferner gilt $\alpha_\Delta > 0$ und $A_{d_2}^{(2)} > 0$. Mit (4.13) erhalten wir deshalb:

$$R_e(\varepsilon) \quad = \quad \frac{\sum\limits_{i=\Delta}^{n} \alpha_i \, \varepsilon^i \, (1-\varepsilon)^{n-i}}{\sum\limits_{j=d_2}^{l} A_j^{(2)} \, \varepsilon^j \, (1-\varepsilon)^{l-j}} \quad = \quad \varepsilon^{\Delta-d_2} \cdot \frac{\sum\limits_{i=0}^{n-\Delta} \alpha_{i+\Delta} \, \varepsilon^i \, (1-\varepsilon)^{n-\Delta-i}}{\sum\limits_{j=0}^{l-d_2} A_{j+d_2}^{(2)} \, \varepsilon^j \, (1-\varepsilon)^{l-d_2-j}} \quad . \qquad (4.18)$$

Hieraus folgen die Aussagen (a), (b) und (c). q.e.d.

Korollar 4.11 (Kløve–Miller–Bedingung) *Die Bedingung* $d_1 > 2d_2$ *impliziert*

$$\lim_{\varepsilon\to 0} R_e(\varepsilon) = 0 \, .$$

Beweis: Die Behauptung folgt unmittelbar aus Lemma 4.9(b) und Satz 4.10(a). □

Wir zeigen nun, daß die Verringerung der Decodierfehlerwahrscheinlichkeit des HI-ARQ-Systems gegenüber dem ARQ-System auch die Verringerung der Restfehlerwahrscheinlichkeit impliziert.

Lemma 4.12 *Mit obigen Bezeichnungen gilt:*

$$\lim_{\varepsilon\to 0} \frac{P_c^{\mathrm{HI}}(\varepsilon)}{P_c^{\mathrm{ARQ}}(\varepsilon)} \; = \; 1 \qquad \textit{und} \qquad \lim_{\varepsilon\to 0} \frac{1 - P_d^{\mathrm{HI}}(\varepsilon)}{1 - P_d^{\mathrm{ARQ}}(\varepsilon)} \; = \; 1 \, .$$

Beweis: Die Behauptung folgt unmittelbar aus den Korollaren 3.2 und 4.2. □

Satz 4.13 *Im Fall $\Delta > d_2$ (oder $d_1 > 2d_2$) gilt für die Restfehlerwahrscheinlichkeiten P_R^{ARQ} des ARQ-Systems und P_R^{HI} des HI-ARQ-Systems:*

$$\lim_{\varepsilon \to 0} \frac{P_R^{\mathrm{HI}}(\varepsilon)}{P_R^{\mathrm{ARQ}}(\varepsilon)} \;=\; 0\,.$$

Beweis: Aus Satz 4.10(a) (bzw. Kor. 4.11) und Lemma 4.12 folgt:

$$\frac{P_R^{\mathrm{HI}}(\varepsilon)}{P_R^{\mathrm{ARQ}}(\varepsilon)} \;=\; \frac{P_e^{\mathrm{HI}}(\varepsilon)}{P_e^{\mathrm{ARQ}}(\varepsilon)} \cdot \frac{1 - P_d^{\mathrm{ARQ}}(\varepsilon)}{1 - P_d^{\mathrm{HI}}(\varepsilon)} \quad \stackrel{\varepsilon \to 0}{\longrightarrow} \quad 0\,.$$

q.e.d.

Ein HI-ARQ-System bietet also unter gewissen Voraussetzungen eine kleinere Restfehlerwahrscheinlichkeit als ein einfaches ARQ-System. Dem steht jedoch eine Verringerung des Durchsatzes bei kleinen Bitfehlerraten gegenüber. Wir betrachten zunächst die Wahrscheinlichkeiten für die Wiederholung einer Übertragung. In Analogie zu Satz 4.10 erhalten wir folgendes Ergebnis:

Satz 4.14 *Aus $\Delta > d_2$ folgt*

$$\lim_{\varepsilon \to 0} \frac{P_d^{\mathrm{HI}}(\varepsilon)}{P_d^{\mathrm{ARQ}}(\varepsilon)} \;=\; 0\,.$$

Beweis: Für $i = 0, \ldots, n$ sei $\beta_i := \sum\limits_{\bar{d} \in \mathcal{C}} \sum\limits_{\substack{\bar{x} \in Q_{\bar{d}} \\ w(\bar{x}) = i}} \frac{1}{a(\bar{x})}$.

Aus $\Delta > d_2 \geq 1$ folgt $\beta_0 = 1$ und $\beta_1 = \binom{n}{1} = n$. So ergibt sich mit (4.17):

$$\begin{aligned} P_d^{\mathrm{HI}}(\varepsilon) &= 1 - \sum_{\bar{d} \in \mathcal{C}} \sum_{\bar{x} \in Q_{\bar{d}}} \frac{1}{a(\bar{x})}\, \varepsilon^{w(\bar{x})}\, (1-\varepsilon)^{n - w(\bar{x})} \\ &= 1 - \sum_{i=0}^{n} \beta_i\, \varepsilon^i\, (1-\varepsilon)^{n-i} \quad = \quad \sum_{i=2}^{n} \left(\binom{n}{i} - \beta_i \right) \varepsilon^i\, (1-\varepsilon)^{n-i}\,. \end{aligned}$$

$\mathcal{C}_1$ ist ein fehlerkorrigierender Code, daher gilt $d_1 > 1$ und somit $A_1^{(2)} = 0$, also $\binom{l}{1} - A_1^{(2)} = l > 0$. Für kleine ε erhalten wir deshalb mit der Landau'schen Notation:

$$\frac{P_d^{\mathrm{HI}}(\varepsilon)}{P_d^{\mathrm{ARQ}}(\varepsilon)} \quad = \quad \frac{\sum\limits_{i=2}^{n} \left(\binom{n}{i} - \beta_i \right) \varepsilon^i\, (1-\varepsilon)^{n-i}}{\sum\limits_{i=1}^{l} \left(\binom{l}{i} - A_i^{(2)} \right) \varepsilon^i\, (1-\varepsilon)^{l-i}} \quad = \quad O(\varepsilon)\,.$$

q.e.d.

Bemerkung: Wir werden diesen Satz stets unter der Voraussetzung $\Delta > d_2$ anwenden, der Beweis zeigt jedoch, daß es genügt, $\Delta \geq 2$ zu fordern.

Die Informationsrate des HI-ARQ-Systems ist geringer als die des ARQ-Systems. Außerdem hatten wir uns bereits überlegt, daß bei der Verwendung des Go-Back-Protokolls die Rückgrifftiefe des HI-ARQ- größer als die des ARQ-Systems sein kann. Dies führt zu:

Satz 4.15 *Für die Durchsatzraten* η^{ARQ} *und* η^{HI} *gilt*

a) bei Verwendung des Selective-Repeat-Protokolls:

$$\lim_{\varepsilon\to 0}\frac{\eta^{\mathrm{HI}}_{\mathrm{SR}}(\varepsilon)}{\eta^{\mathrm{ARQ}}_{\mathrm{SR}}(\varepsilon)} \;=\; \frac{l}{n}$$

und falls $\Delta > d_2$ *ist:*

$$\eta^{\mathrm{HI}}_{\mathrm{SR}}(\varepsilon) \;\geq\; \tfrac{l}{n}\cdot\eta^{\mathrm{ARQ}}_{\mathrm{SR}}(\varepsilon) \quad \textit{für hinreichend kleine } \varepsilon\,,$$

b) bei Verwendung des Go-Back-Protokolls:

$$\lim_{\varepsilon\to 0}\frac{\eta^{\mathrm{HI}}_{\mathrm{GB}N}(\varepsilon)}{\eta^{\mathrm{ARQ}}_{\mathrm{GB}L}(\varepsilon)} \;=\; \frac{l}{n}$$

und

$$\eta^{\mathrm{HI}}_{\mathrm{GB}N}(\varepsilon) \;\geq\; \tfrac{l}{n}\cdot\eta^{\mathrm{ARQ}}_{\mathrm{GB}L}(\varepsilon)\,, \quad \textit{falls} \quad N \leq L\,\frac{P_d^{\mathrm{ARQ}}(\varepsilon)(1-P_d^{\mathrm{HI}}(\varepsilon))}{P_d^{\mathrm{HI}}(\varepsilon)(1-P_d^{\mathrm{ARQ}}(\varepsilon))} \quad \textit{ist.}$$

Beweis: Mit den Ergebnissen aus den Abschnitten 3.2 und 4.2 sowie Lemma 4.12 folgt:

$$\begin{aligned}
\frac{\eta^{\mathrm{HI}}_{\mathrm{SR}}(\varepsilon)}{\eta^{\mathrm{ARQ}}_{\mathrm{SR}}(\varepsilon)} &= \frac{1-P_d^{\mathrm{HI}}(\varepsilon)}{1-P_d^{\mathrm{ARQ}}(\varepsilon)}\cdot\frac{l}{n} \;\overset{\varepsilon\to 0}{\longrightarrow}\; \frac{l}{n} \quad \text{und}\\
\frac{\eta^{\mathrm{HI}}_{\mathrm{GB}N}(\varepsilon)}{\eta^{\mathrm{ARQ}}_{\mathrm{GB}L}(\varepsilon)} &= \frac{1-P_d^{\mathrm{HI}}(\varepsilon)}{1-P_d^{\mathrm{ARQ}}(\varepsilon)}\cdot\frac{1+(L-1)P_d^{\mathrm{ARQ}}(\varepsilon)}{1+(N-1)P_d^{\mathrm{HI}}(\varepsilon)}\cdot\frac{l}{n} \;\overset{\varepsilon\to 0}{\longrightarrow}\; \frac{l}{n}\,.
\end{aligned}$$

Nach Satz 4.14 gilt für hinreichend kleine ε:

$$\eta^{\mathrm{HI}}_{\mathrm{SR}}(\varepsilon) \;=\; (1-P_d^{\mathrm{HI}}(\varepsilon))\,\tfrac{k}{n} \;\geq\; (1-P_d^{\mathrm{ARQ}}(\varepsilon))\,\tfrac{l}{n}\cdot\tfrac{k}{l} \;=\; \tfrac{l}{n}\,\eta^{\mathrm{ARQ}}_{\mathrm{SR}}(\varepsilon)\,.$$

Mit elementaren Umformungen erhalten wir:

$$\frac{1-P_d^{\mathrm{HI}}(\varepsilon)}{1-P_d^{\mathrm{ARQ}}(\varepsilon)}\cdot\frac{1+(L-1)P_d^{\mathrm{ARQ}}(\varepsilon)}{1+(N-1)P_d^{\mathrm{HI}}(\varepsilon)} \;\geq\; 1 \;\Longleftrightarrow\; N \;\leq\; L\,\frac{P_d^{\mathrm{ARQ}}(\varepsilon)(1-P_d^{\mathrm{HI}}(\varepsilon))}{P_d^{\mathrm{HI}}(\varepsilon)(1-P_d^{\mathrm{ARQ}}(\varepsilon))}\,.$$

Dies impliziert

$$\frac{\eta^{\mathrm{HI}}_{\mathrm{GB}N}(\varepsilon)}{\eta^{\mathrm{ARQ}}_{\mathrm{GB}L}(\varepsilon)} \;\geq\; \frac{l}{n} \;\Longleftrightarrow\; N \;\leq\; L\,\frac{P_d^{\mathrm{ARQ}}(\varepsilon)(1-P_d^{\mathrm{HI}}(\varepsilon))}{P_d^{\mathrm{HI}}(\varepsilon)(1-P_d^{\mathrm{ARQ}}(\varepsilon))}\,.$$

q.e.d.

Bemerkungen:

a) Im Fall $\Delta > d_2$ folgt aus Satz 4.14

$$\frac{P_d^{\mathrm{ARQ}}(\varepsilon)(1-P_d^{\mathrm{HI}}(\varepsilon))}{P_d^{\mathrm{HI}}(\varepsilon)(1-P_d^{\mathrm{ARQ}}(\varepsilon))} \geq 1 \quad \text{und} \quad \lim_{\varepsilon\to 0} \frac{P_d^{\mathrm{ARQ}}(\varepsilon)(1-P_d^{\mathrm{HI}}(\varepsilon))}{P_d^{\mathrm{HI}}(\varepsilon)(1-P_d^{\mathrm{ARQ}}(\varepsilon))} = \infty \,.$$

Dies bedeutet, daß bei hinreichend kleinen Bitfehlerraten ε (abhängig von L und N)

$$\eta_{\mathrm{GB}N}^{\mathrm{HI}}(\varepsilon) \;\geq\; \tfrac{l}{n} \cdot \eta_{\mathrm{GB}L}^{\mathrm{ARQ}}(\varepsilon)$$

gilt.

b) $\mathcal{C}_1$ ist ein fehlerkorrigierender (n,l)-Code, daher gilt $n > l$. Bei kleinen Bitfehlerwahrscheinlichkeiten ist der Durchsatz des HI-ARQ-Systems also geringer als der des ARQ-Systems. Die Durchsatzrate des HI-ARQ-Systems bleibt jedoch größer als die mit der Informationsrate des inneren Codes gewichtete Durchsatzrate des ARQ-Systems. Bei Verwendung des Go-Back-Protokolls gilt dies nur, falls die Differenz der Rückgrifftiefen hinreichend klein ist.

4.3.1 Resümee

Das HI-ARQ-System ist technisch bedeutend aufwendiger als das einfache ARQ-System. Dafür bewirkt es unter gewissen Voraussetzungen eine Verringerung der Restfehlerwahrscheinlichkeit, insbesondere bei kleinen Bitfehlerraten. Ein hinreichendes Kriterium hierfür stellt die Kløve–Miller–Bedingung dar. Weiterhin gewährleistet die Kløve–Miller–Bedingung, daß die Verschlechterung des Durchsatzes durch die Informationsrate des inneren, fehlerkorrigierenden Codes beschränkt wird.

Die Verwendung des HI-ARQ-Systems erscheint also insbesondere dann sinnvoll, wenn die Minimierung der Restfehlerwahrscheinlichkeit angestrebt wird, auch wenn damit eine eventuelle Verschlechterung des Durchsatzes verbunden ist.

Kapitel 5

Typ-II Hybrid-ARQ-Systeme

In diesem Kapitel werden wir die Zuverlässigkeit und den Durchsatz eines Typ-II Hybrid-ARQ-Systems (HII-ARQ) untersuchen.

Das hier beschriebene und analysierte Übertragungssystem ist eine Weiterentwicklung des HI-ARQ-Systems und wurde von Wang und Lin vorgeschlagen ([18]), aufbauend auf Ergebnissen von Lin und Yu. Die zugrundeliegende Idee ist die abwechselnde Übertragung von Datenblöcken und Blöcken, die aus Redundanzbits des inneren, fehlerkorrigierenden Codes bestehen. Eine Verallgemeinerung dieses Systems werden wir im nächsten Kapitel betrachten.

5.1 Spezifikation

Es seien $\mathcal{C}_2$ ein linearer fehlererkennender (n, k, d_2)-Code über $GF(2)$ und $\mathcal{C}_1$ ein linearer fehlerkorrigierender $(2n, n, d_1)$-Code über $GF(2)$. Der Code $\mathcal{C}_1$ sei systematisch[1] und invertierbar[2]. Die Generatormatrix $\mathbf{G}_1$ von $\mathcal{C}_1$ hat daher die Gestalt

$$\mathbf{G_1} = \left[\mathbf{I}_n \mid \tilde{\mathbf{G}}_1\right] ,$$

wobei $\mathbf{I}_n$ die $(n \times n)$-Einheitsmatrix bezeichne und $\tilde{\mathbf{G}}_1$ eine nicht-singuläre $(n \times n)$-Matrix über $GF(2)$ sei. Ferner bezeichne $\tilde{\mathcal{C}_1}$ den durch $\tilde{\mathbf{G}}_1$ definierten (n, n)-Code über $GF(2)$.

Ein Informationswort $\bar{a}$ der Länge k wird nach dem Prinzip der gewöhnlichen Codeverkettung (s. Abschnitt 1.4.2) in ein $\mathcal{C}_1$-Codewort $(\bar{c}, p(\bar{c}))$ codiert, wobei $\bar{c}$ ein $\mathcal{C}_2$-Codewort und $p(\bar{c}) = \bar{c}\tilde{G}_1$ die zugehörigen Redundanzbits seien. Der so definierte Code werde mit $\mathcal{C}$ bezeichnet.

Zunächst wird nur der Block $\bar{c}$ übertragen und die Fehlererkennung basierend auf dem Code $\mathcal{C}_2$ durchgeführt. Wenn keine Fehler entdeckt werden, wird ein ACK an den Sender geschickt, die Übertragung des Informationswortes $\bar{a}$ ist damit beendet (wir nehmen hier an, daß der Rückkopplungskanal ungestört sei). Anderenfalls wird der empfangene Block $\hat{c}$ gespeichert und ein NACK an den Sender geschickt. Dieser überträgt daraufhin den Block $p(\bar{c})$. Hierbei handelt es sich um ein $\tilde{\mathcal{C}_1}$-Codewort, der empfangene Block $\hat{p}(\bar{c})$ wird deshalb invertiert und vom

[1] *Systematisch* bedeutet hier, daß die ersten n Bits eines $\mathcal{C}_1$-Codewortes Informationsbits sind (s. [5, S. 36]).

[2] *Invertierbar* bedeutet hier, daß aus den n Redundanzbits eines $\mathcal{C}_1$-Codewortes das codierte Informationswort eindeutig rekonstruiert werden kann ([12, S. 185]).

$\mathcal{C}_2$-Decodierer auf Übertragungsfehler untersucht. Falls dabei Fehler entdeckt werden, wird das Wort $(\hat{c}, \hat{p}(\bar{c}))$ einer auf dem Code $\mathcal{C}_1$ basierenden Fehlerkorrektur unterzogen. Das so erhaltene Wort c^* wird nun wiederum auf Fehler untersucht. Sollte eine erneute Reübertragung notwendig sein, wird der Block $\hat{c}$ aus dem Speicher des Empfängers gelöscht und statt dessen $\hat{p}(\bar{c})$ gespeichert. Der Sender überträgt als nächstes wieder den Block $\bar{c}$. Auf der Seite des Empfängers werden Fehlererkennung und gegebenenfalls Fehlerkorrektur (mit $\hat{p}(\bar{c})$) durchgeführt. Dieses Verfahren wird wiederholt, bis keine Fehler mehr entdeckt werden (s. Abb. 5.1).

Es sei $\tilde{\mathcal{C}}$ der durch gewöhnliche Codeverkettung von $\tilde{\mathcal{C}}_1$ und $\mathcal{C}_2$ definierte (n,k)-Code. Wir wollen voraussetzen, daß $\mathcal{C}_2$ und $\tilde{\mathcal{C}}$ äquivalente Codes[3] sind. In diesem Fall sind nämlich die Decodierfehlerwahrscheinlichkeiten der Codes $\mathcal{C}_2$ und $\tilde{\mathcal{C}}$ gleich, sie entsprechen der eines einfachen ARQ-Systems (s. Satz 3.1).

Wang und Lin schlagen folgende Wahl der Codes $\mathcal{C}_1$ und $\mathcal{C}_2$ vor ([18, S. 595]): Es seien $\mathcal{C}_1$ und $\mathcal{C}_2$ zyklische oder gekürzte zyklische Codes mit Generatorpolynomen $g_1(x)$ und $g_2(x)$, wobei $g_2(x)$ ein Faktor von $g_1(x)$ sei. Ein $\mathcal{C}_2$-Codewort $c_2(x)$ (in Polynomdarstellung) wird in $x^n c_2(x) + p(x)$ codiert, wobei $p(x)$ ein $\tilde{\mathcal{C}}$-Codewort (in Polynomdarstellung) ist. Außerdem gilt $p(x) \equiv x^n c_2(x) \bmod g_1(x)$, denn $p(x)$ ist der bei der Division von $x^n c_2(x)$ durch $g_1(x)$ verbleibende Rest (vgl. Abschnitt 2.2.2). Nun sind aber $g_1(x)$ und $c_2(x)$ durch $g_2(x)$ teilbar, also auch $p(x)$. Daher entspricht $p(x)$ einem $\mathcal{C}_2$-Codewort. Folglich sind $\mathcal{C}_2$ und $\tilde{\mathcal{C}}$ identisch und somit insbesondere äquivalent. In diesem Fall ist es natürlich nicht nötig, ein $\tilde{\mathcal{C}}$-Codewort vor der Fehlererkennung zu invertieren.

5.2 Leistungsanalyse

Für das oben beschriebene HII-ARQ-System sollen zusätzlich zu den in Abschnitt 1.1 genannten die folgenden Annahmen gelten:

- Der Übertragungskanal sei ein gedächtnisloser binärer symmetrischer Kanal mit Bitfehlerwahrscheinlichkeit $\varepsilon \in (0, \frac{1}{2}]$.
- Der Rückkopplungskanal sei ungestört.
- Der Sender und der Empfänger (bei Verwendung des Selective-Repeat-Protokolls) verfügen über unendlich große Speicher.
- Die Codes $\mathcal{C}_1$ und $\mathcal{C}_2$ seien so gewählt, daß $\mathcal{C}_2$ und $\tilde{\mathcal{C}}$ äquivalent sind.

[3] Zwei lineare (n,k)-Codes $\mathcal{D}_1$ und $\mathcal{D}_2$ heißen *äquivalent*, falls es eine Permutation σ von $\{1, 2, \ldots, n\}$ mit $\mathcal{D}_2 = \{(x_{\sigma(1)}, \ldots, x_{\sigma(n)}) : (x_1, \ldots, x_n) \in \mathcal{D}_1\}$ gibt (s. [5, S. 37], [10, S. 33]).

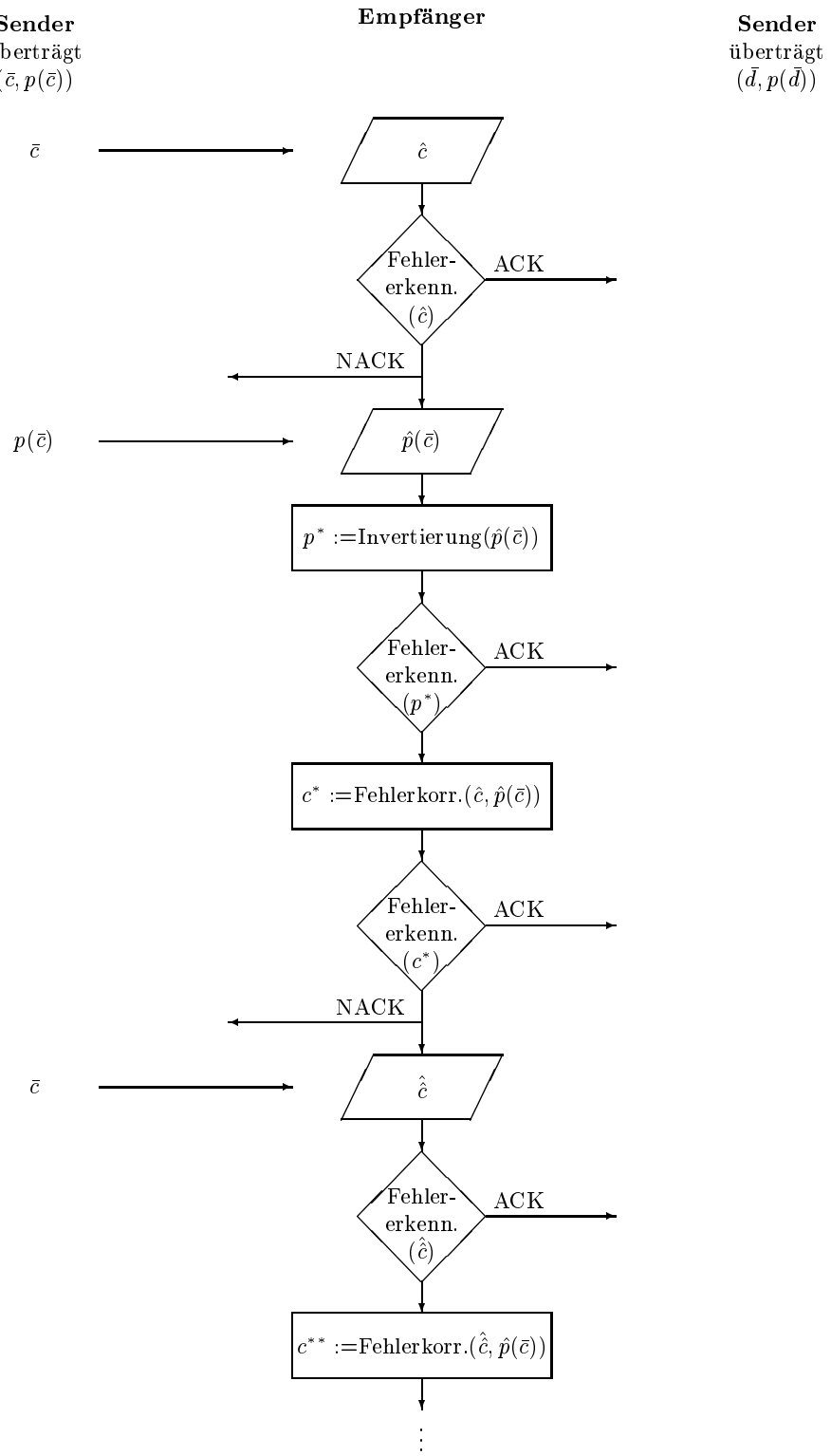

Abb. 5.1: Übertragungsprozedur des HII-ARQ-Systems

5.2.1 Notation

Entsprechend der von Wang und Lin ([18]) benutzten Notation werden wir die mit der Fehlererkennung und -korrektur verbundenen Ereignisse wie folgt bezeichnen:

A_0^c, A_0^d, A_0^e	:	bei der ersten Übertragung treten keine Fehler (A_0^c), erkennbare Fehler (A_0^d) bzw. nichterkennbare Fehler (A_0^e) auf,
B_i^c, B_i^d, B_i^e	:	bei der i-ten Reübertragung treten keine Fehler (B_i^c), erkennbare Fehler (B_i^d) bzw. nichterkennbare Fehler (B_i^e) auf ($i \in I\!N$),
G_i^c, G_i^d, G_i^e	:	das nach der i-ten Fehlerkorrektur erhaltene Wort enthält keine Fehler (G_i^c), erkennbare Fehler (G_i^d) bzw. nichterkennbare Fehler (G_i^e) ($i \in I\!N$),
E_i^c, E_i^e	:	nach der i-ten Reübertragung wird das Informationswort richtig (E_i^c) bzw. fehlerbehaftet (E_i^e) an den Empfänger übergeben ($i \in I\!N$),
E_i^d	:	nach der i-ten Reübertragung wird das Informationswort nicht an den Empfänger übergeben ($i \in I\!N$).

Zwischen diesen Ereignissen bestehen die Zusammenhänge:

$$E_i^c = B_i^c + (B_i^d \cap G_i^c)\,, \quad E_i^d = B_i^d \cap G_i^d \quad \text{und} \quad E_i^e = B_i^e + (B_i^d \cap G_i^e) \quad (i \in I\!N)\,,$$

wobei + die Vereinigung disjunkter Ereignisse bezeichne.

Wegen der Äquivalenz der Codes $\mathcal{C}_2$ und $\tilde{\mathcal{C}}$ können wir die folgenden Wahrscheinlichkeiten[4] definieren (hierbei seien $B_0^c := A_0^c$, $B_0^d := A_0^d$ und $B_0^e := A_0^e$):

$$\begin{aligned}
P_c^{(2)} &:= \text{Prob}(A_0^c) = \text{Prob}(B_i^c) \quad \forall\, i \in I\!N\,,\\
P_d^{(2)} &:= \text{Prob}(A_0^d) = \text{Prob}(B_i^d) \quad \forall\, i \in I\!N\,,\\
P_e^{(2)} &:= \text{Prob}(A_0^e) = \text{Prob}(B_i^e) \quad \forall\, i \in I\!N\,,\\
P_c^{(1)} &:= \text{Prob}(B_{i-1}^d \cap B_i^d \cap G_i^c) \quad \forall\, i \in I\!N\,,\\
P_d^{(1)} &:= \text{Prob}(B_{i-1}^d \cap B_i^d \cap G_i^d) \quad \forall\, i \in I\!N\,,\\
P_e^{(1)} &:= \text{Prob}(B_{i-1}^d \cap B_i^d \cap G_i^e) \quad \forall\, i \in I\!N\,.
\end{aligned}$$

Die Wahrscheinlichkeiten $P_c^{(2)}$, $P_d^{(2)}$ und $P_e^{(2)}$ entsprechen denen eines ARQ-Systems und sind durch Satz 3.1 gegeben. Eine wichtige Beziehung zwischen diesen Wahrscheinlichkeiten liefert das folgende Lemma.

Lemma 5.1 *Mit obigen Bezeichnungen gilt:*

$$0 \quad \leq {P_d^{(2)}}^2 - P_c^{(1)} - P_e^{(1)} \quad < \quad 1\,.$$

Beweis: Wegen der Gedächtnislosigkeit des Übertragungskanals sind die Ereignisse B_{i-1}^d und B_i^d stochastisch unabhängig ($i \in I\!N$). Dies impliziert:

$$P_c^{(1)} + P_e^{(1)} \;=\; \text{Prob}(B_{i-1}^d \cap B_i^d \cap (G_i^c + G_i^e)) \;\leq\; \text{Prob}(B_{i-1}^d \cap B_i^d) \;=\; {P_d^{(2)}}^2\,.$$

Außerdem ist ${P_d^{(2)}}^2 - P_c^{(1)} - P_e^{(1)} \leq P_d^{(2)} < 1$ nach Satz 3.1. q.e.d.

[4] Es sei ein geeigneter Wahrscheinlichkeitsraum zugrundegelegt.

5.2.2 Zuverlässigkeit

Als Maß für die Zuverlässigkeit des hier betrachteten HII-ARQ-Systems dient uns wieder die Restfehlerwahrscheinlichkeit P_R (s. Kap. 3). Hierfür gilt mit obigen Bezeichnungen offensichtlich (s. [18, S. 602]):

$$P_R \quad = \quad \mathrm{Prob}(A_0^e) + \sum_{i=1}^{\infty} \mathrm{Prob}(A_0^d \cap E_1^d \cap \ldots \cap E_{i-1}^d \cap E_i^e) \; . \tag{5.1}$$

Leider können die Wahrscheinlichkeiten $\mathrm{Prob}(A_0^d \cap E_1^d \cap \ldots \cap E_{i-1}^d \cap E_i^e)$ im allgemeinen nur schwer bestimmt werden, denn die hier auftretenden Ereignisse sind nicht stochastisch unabhängig.

Wir können die Restfehlerwahrscheinlichkeit nach oben abschätzen, indem wir ein vereinfachtes Übertragungssystem $\widehat{\mathrm{HII}}$-ARQ betrachten. Wir nehmen an, daß das $\widehat{\mathrm{HII}}$-ARQ-System die Fehlerkorrektur nur nach jeder zweiten Übertragung durchführt (s. [18, S. 596]). Dieses System hat eine größere Restfehlerwahrscheinlichkeit $\hat{P_R}$ als unser HII-ARQ-System, es gilt nämlich:

$$\begin{aligned} \hat{P_R} \quad &= \quad \mathrm{Prob}(A_0^e) + \sum_{i=1}^{\infty} \mathrm{Prob}(A_0^d \cap E_1^d \cap B_2^d \cap E_3^d \cap \ldots \cap B_{2i-2}^d \cap E_{2i-1}^e) \\ &\qquad + \sum_{i=1}^{\infty} \mathrm{Prob}(A_0^d \cap E_1^d \cap B_2^d \cap E_3^d \cap \ldots \cap E_{2i-1}^d \cap B_{2i}^e) \\ &\geq \quad P_R \; . \end{aligned}$$

So erhalten wir folgenden Satz:

Satz 5.2 *Für die Restfehlerwahrscheinlichkeit P_R des HII-ARQ-Systems gilt:*

$$P_R \; \leq \; \frac{P_e^{(2)} \left(1 + P_d^{(2)}\right) + P_e^{(1)}}{1 - \left({P_d^{(2)}}^2 - P_c^{(1)} - P_e^{(1)}\right)} \; .$$

Beweis: Dieser Satz ist ein Spezialfall des Satzes 6.1 bzw. des Korollars 6.2 ($m = 2$). □

Bemerkungen:

a) Der obige Bruch ist die Restfehlerwahrscheinlichkeit des $\widehat{\mathrm{HII}}$-ARQ-Systems. Im Zähler steht die Wahrscheinlichkeit dafür, daß bei einer der beiden ersten Übertragungen oder bei der Fehlerkorrektur ein nichterkennbarer Fehler auftritt. Im Beweis zu Satz 6.1, Beziehung (6.7), werden wir sehen, daß ${P_d^{(2)}}^2 - P_c^{(1)} - P_e^{(1)} = \mathrm{Prob}(B_{j-1}^d \cap E_j^d)$ für ungerade $j \in I\!N$ gilt. Also steht im Nenner die Wahrscheinlichkeit dafür, daß spätestens die zweite Übertragung erfolgreich ist.

b) Wang und Lin erhalten in ihrer Arbeit durch eine gröbere Abschätzung und unter der Annahme, daß $\mathrm{Prob}(G_i^e \mid A_0^d \cap B_1^d \cap \ldots \cap B_i^d) \leq P_e^{(2)}$ gilt ([18, S. 602]):

$$P_R \leq \left(1 + {P_d^{(2)}}^2\right) \frac{P_e^{(2)}}{P_c^{(2)} + P_e^{(2)}} \; .$$

In diesem Fall gilt aber $P_e^{(1)} \leq P_e^{(2)} {P_d^{(2)}}^2$ und deshalb

$$\begin{aligned}\frac{P_e^{(2)}(1+P_d^{(2)})+P_e^{(1)}}{1-({P_d^{(2)}}^2-P_c^{(1)}-P_e^{(1)})} &\leq \frac{P_e^{(2)}(1+P_d^{(2)}+{P_d^{(2)}}^2)}{1-{P_d^{(2)}}^2} \\ &\leq \frac{P_e^{(2)}(1+P_d^{(2)})(1+{P_d^{(2)}}^2)}{(1-P_d^{(2)})(1+P_d^{(2)})} = \frac{P_e^{(2)}(1+{P_d^{(2)}}^2)}{P_c^{(2)}+P_e^{(2)}} .\end{aligned}$$

c) Das Decodierverfahren des fehlerkorrigierenden Codes $\mathcal{C}_1$ sei ein Maximum–Likelihood–Verfahren (s. Abschnitt 4.1). Wie in Abschnitt 4.1 seien

$$Q_{\bar{c}_1} := \left\{ \bar{x} \in (GF(2))^{2n} \;:\; d(\bar{x},\bar{c}_1) \leq d(\bar{x},\bar{d}_1) \quad \forall\, \bar{d}_1 \in \mathcal{C}_1 \right\}, \quad \bar{c}_1 \in \mathcal{C}_1$$

und

$$a(\bar{x}) := \sharp\left\{ \bar{c}_1 \in \mathcal{C}_1 \;:\; d(\bar{x},\bar{c}_1) \leq d(\bar{x},\bar{d}_1) \quad \forall\, \bar{d}_1 \in \mathcal{C}_1 \right\}, \quad \bar{x} \in (GF(2))^{2n} .$$

Nach Satz 4.1 gilt dann für $i \in \mathbb{N}$:

$$\begin{aligned}\mathrm{Prob}(G_i^c) &= \sum_{\bar{x}\in Q_{\bar{0}}} \frac{1}{a(\bar{x})}\, \varepsilon^{w(\bar{x})}\, (1-\varepsilon)^{2n-w(\bar{x})} , \\ \mathrm{Prob}(G_i^e) &= \sum_{\bar{d}\in\mathcal{C}\setminus\{\bar{0}\}} \sum_{\bar{x}\in Q_{\bar{d}}} \frac{1}{a(\bar{x})}\, \varepsilon^{w(\bar{x})}\, (1-\varepsilon)^{2n-w(\bar{x})} .\end{aligned}$$

Damit erhalten wir für $P_c^{(1)}$ und $P_e^{(1)}$:

$$\begin{aligned}P_c^{(1)} &= \mathrm{Prob}(B_{i-1}^d \cap B_i^d \cap G_i^c) = \sum_{\substack{\bar{x}=(\bar{x}_1,\bar{x}_2)\in Q_{\bar{0}} \\ \bar{x}_1,\bar{x}_2\notin\mathcal{C}_2}} \frac{1}{a(\bar{x})}\, \varepsilon^{w(\bar{x})}\, (1-\varepsilon)^{2n-w(\bar{x})} , \\ P_e^{(1)} &= \mathrm{Prob}(B_{i-1}^d \cap B_i^d \cap G_i^e) = \sum_{\bar{d}\in\mathcal{C}\setminus\{\bar{0}\}} \sum_{\substack{\bar{x}=(\bar{x}_1,\bar{x}_2)\in Q_{\bar{d}} \\ \bar{x}_1,\bar{x}_2\notin\mathcal{C}_2}} \frac{1}{a(\bar{x})}\, \varepsilon^{w(\bar{x})}\, (1-\varepsilon)^{2n-w(\bar{x})} .\end{aligned}$$

d) Ein ARQ-System entspricht einem HII-ARQ-System ohne Fehlerkorrektur (d. h. wir wählen $\tilde{\mathbf{G}}_1 = \mathbf{I}_n$). In diesem Fall gilt $P_c^{(1)} = \mathrm{Prob}(B_{i-1}^d \cap B_i^d \cap G_i^c) = \mathrm{Prob}(\emptyset) = 0$ und analog $P_e^{(1)} = 0$, so daß wir das gleiche Ergebnis wie in Satz 3.4 erhalten. Unsere Analyse ist hier sogar exakt.

Bei der Analyse der Zuverlässigkeit eines HI-ARQ-Systems betrachteten wir den Fall, daß der innere Code perfekt ist. Dies ist hier nicht möglich, denn es gibt keinen perfekten linearen $(2n, n)$-Code über $GF(2)$ (s. [4, S. 21], [17, S. 154]).

5.2.3 Durchsatz

Wir nehmen wieder an, daß beim Sender und beim Empfänger (bei Verwendung des Selective-Repeat-Protokolls) unendlich große Speicher zur Verfügung stehen.

Es werden immer nur $\mathcal{C}_2$- oder $\tilde{\mathcal{C}}$-Codewörter übertragen. Die Informationsrate beider Codes beträgt $R = \frac{k}{n}$. Für die Durchsatzrate η gilt dann $\eta = V^{-1}R$, wobei V die mittlere Anzahl der Übertragungen eines Informationswortes bezeichne.

Bei Verwendung des Selective-Repeat-Protokolls gilt dann mit obigen Bezeichnungen (vgl. [18, S. 596]):

$$V_{\mathrm{SR}} \;=\; \mathrm{Prob}(A_0^c + A_0^e) + \sum_{i=1}^{\infty}(i+1)\,\mathrm{Prob}(A_0^d \cap E_1^d \cap \ldots \cap E_{i-1}^d \cap (E_i^c + E_i^e)) \;. \tag{5.2}$$

Analog gilt bei der Verwendung des Go-Back-N-Protokolls:

$$V_{\mathrm{GBN}} \;=\; \mathrm{Prob}(A_0^c + A_0^e) + \sum_{i=1}^{\infty}(iN+1)\,\mathrm{Prob}(A_0^d \cap E_1^d \cap \ldots \cap E_{i-1}^d \cap (E_i^c + E_i^e)) \;. \tag{5.3}$$

Wie bei der Analyse der Zuverlässigkeit dieses HII-ARQ-Systems werden wir wieder das vereinfachte Übertragungssystem $\widehat{\mathrm{HII}}$-ARQ betrachten und so eine untere Schranke für die Durchsatzraten erhalten.

Satz 5.3 *Für die Durchsatzrate η des HII-ARQ-Systems gilt*

a) bei Verwendung des Selective-Repeat-Protokolls:

$$\eta_{\mathrm{SR}} \;\geq\; \frac{1 - \left(P_d^{(2)^2} - P_c^{(1)} - P_e^{(1)}\right)}{1 + P_d^{(2)}} \cdot \frac{k}{n} \;,$$

b) bei Verwendung des Go-Back-N-Protokolls:

$$\eta_{\mathrm{GBN}} \;\geq\; \frac{1 - \left(P_d^{(2)^2} - P_c^{(1)} - P_e^{(1)}\right)}{1 + NP_d^{(2)} + (N-1)\left(P_d^{(2)^2} - P_c^{(1)} - P_e^{(1)}\right)} \cdot \frac{k}{n} \;.$$

Beweis: Dieser Satz ist ein Spezialfall des Satzes 6.3 ($m = 2$). □

Bemerkungen:

a) Die obigen Brüche sind die Durchsatzraten des $\widehat{\mathrm{HII}}$-ARQ-Systems. Im Beweis zu Satz 6.1, Beziehung (6.7), werden wir sehen, daß $P_d^{(2)^2} - P_c^{(1)} - P_e^{(1)} = \mathrm{Prob}(B_{j-1}^d \cap E_j^d)$ für ungerade $j \in \mathbb{N}$ gilt. Also steht im Zähler die Wahrscheinlichkeit dafür, daß spätestens die zweite Übertragung erfolgreich ist. Diese wird im Nenner mit der entsprechenden Anzahl

übertragener Blöcke gewichtet. Beim Selective-Repeat-Protokoll sind dies einfach $1+P_d^{(2)}$. Beim Go-Back-N-Protokoll sind dies $1+NP_d^{(2)}$ und die Anzahl zusätzlich zu wiederholender Blöcke, falls die ersten beiden Übertragungen nicht erfolgreich sind, nämlich $(N-1)\text{Prob}(B_{j-1}^d \cap E_j^d)$.

b) Wang und Lin analysieren in ihrer Arbeit nur Systeme mit Selective-Repeat-Protokoll. Sie erhalten unter der Annahme, daß $P_e^{(1)} \approx 0$ und $P_e^{(2)} \approx 0$ gilt ([18, S. 604f]):

$$\eta_{\text{SR}} \geq \frac{2P_c^{(2)} + (1-P_c^{(2)})q_1 - P_c^{(2)}\left[P_c^{(2)} + (1-P_c^{(2)})q_1\right]}{2-P_c^{(2)}} \cdot \frac{k}{n} \quad \text{mit} \quad q_1 = \frac{P_c^{(1)}}{{P_d^{(2)}}^2}\,.$$

Unter obiger Annahme gilt $P_c^{(2)} \approx 1-P_d^{(2)}$ und $P_c^{(1)} \approx P_c^{(1)} + P_e^{(1)}$, so daß unser Ergebnis mit dem von Wang und Lin übereinstimmt.
Wang und Lin wählen den fehlererkennenden Code $\mathcal{C}_2$ so, daß $P_e^{(2)} \leq (1-(1-\varepsilon)^k)2^{-(n-k)}$ gilt ([18, S. 594]). Dies begründet die obige Annahme. Jedoch sind bisher nur sehr wenige Codes bekannt, die diese Schranke erfüllen (s. [12, S. 196]).

5.3 Vergleich mit ARQ- und Typ-I Hybrid-ARQ-Systemen

Wir wollen unser HII-ARQ-System mit einem einfachen ARQ- und einem HI-ARQ-System, die ebenfalls die Codes $\mathcal{C}_1$ (nur HI-ARQ) und $\mathcal{C}_2$ verwenden, vergleichen. Es stellt sich natürlich die Frage, ob die Kombination der Codes $\mathcal{C}_1$ und $\mathcal{C}_2$ in einem HI-ARQ-System sinnvoll ist, schließlich wurde $\mathcal{C}_1$ speziell für ein HII-ARQ-System gewählt. Theoretisch ist eine solche Kombination möglich, so daß wir sie hier auch betrachten wollen. Ein weiterer Grund für diesen Vergleich liegt in der Struktur des HII-ARQ-Systems: Es enthält nämlich ein einfaches ARQ- und zwei derartige HI-ARQ-Systeme (eines davon tritt nach den ungeradzahligen Reübertragungen in Aktion, das andere nach den geradzahligen). Dies zeigt sich auch in der folgenden Abschätzung der Restfehlerwahrscheinlichkeit des HII-ARQ-Systems.

Satz 5.4 *Für die Restfehlerwahrscheinlichkeiten P_R^{ARQ}, P_R^{HI} und P_R^{HII} gilt:*

$$P_R^{\text{HII}} \leq P_R^{\text{ARQ}} + 2\,P_R^{\text{HI}}\,.$$

Beweis: Wegen der Äquivalenz der Codes $\mathcal{C}_2$ und $\tilde{\mathcal{C}}$ gilt mit der Notation aus Abschnitt 5.2.1:

$$P_R^{\text{ARQ}} = \text{Prob}(A_0^e) + \sum_{i=1}^{\infty} \text{Prob}(A_0^d \cap B_1^d \cap \ldots \cap B_{i-1}^d \cap B_i^e)$$

und

$$P_R^{\text{HI}} = \sum_{i=0}^{\infty} \text{Prob}(G_1^d \cap G_3^d \cap \ldots \cap G_{2i-1}^d \cap G_{2i+1}^e) = \sum_{i=1}^{\infty} \text{Prob}(G_2^d \cap G_4^d \cap \ldots \cap G_{2(i-1)}^d \cap G_{2i}^e)\,.$$

Aus (5.1) folgt deshalb:

$$\begin{aligned}
P_R^{\mathrm{HII}} &= \mathrm{Prob}(A_0^e) + \sum_{i=1}^{\infty} \mathrm{Prob}(A_0^d \cap E_1^d \cap \ldots \cap E_{i-1}^d \cap E_i^e) \\
&= \mathrm{Prob}(A_0^e) + \sum_{i=1}^{\infty} \mathrm{Prob}(A_0^d \cap B_1^d \cap G_1^d \cap \ldots \cap B_{i-1}^d \cap G_{i-1}^d \cap B_i^e) \\
&\qquad + \sum_{i=1}^{\infty} \mathrm{Prob}(A_0^d \cap B_1^d \cap G_1^d \cap \ldots \cap B_{i-1}^d \cap G_{i-1}^d \cap B_i^d \cap G_i^e) \\
&\leq \mathrm{Prob}(A_0^e) + \sum_{i=1}^{\infty} \mathrm{Prob}(A_0^d \cap B_1^d \cap B_2^d \cap \ldots \cap B_{i-1}^d \cap B_i^e) \\
&\quad + \sum_{i=0}^{\infty} \mathrm{Prob}(G_1^d \cap G_3^d \cap \ldots \cap G_{2i-1}^d \cap G_{2i+1}^e) \\
&\quad + \sum_{i=1}^{\infty} \mathrm{Prob}(G_2^d \cap G_4^d \cap \ldots \cap G_{2(i-1)}^d \cap G_{2i}^e) \\
&= P_R^{\mathrm{ARQ}} + 2P_R^{\mathrm{HI}} \,.
\end{aligned}$$

q.e.d.

Wir nehmen für den Rest dieses Kapitels an, daß als Decodierverfahren des fehlerkorrigierenden Codes $\mathcal{C}_1$ ein Maximum–Likelihood–Verfahren verwendet werde.

Wegen der Äquivalenz der Codes $\mathcal{C}_2$ und $\tilde{\mathcal{C}}$ hat $\tilde{\mathcal{C}}$ ebenfalls den Minimalabstand d_2. Für den durch gewöhnliche Codeverkettung von $\mathcal{C}_1$ und $\mathcal{C}_2$ definierten $(2n, k, d)$-Code $\mathcal{C}$ gilt daher $d \geq 2d_2$.

Es sei $P_e' := \mathrm{Prob}(G_i^e)$ für $i \in \mathbb{N}$, dies entspricht der Decodierfehlerwahrscheinlichkeit eines mit $\mathcal{C}$ gebildeten HI-ARQ-Systems (s. Abschnitt 4.1). Wie in Abschnitt 4.3 sei

$$\Delta := \min\{w(\bar{x}) \,:\, \bar{x} \in Q_{\bar{c}} \text{ für ein } \bar{c} \in \mathcal{C} \setminus \{\bar{0}\}\} \,.$$

Im Fall $d > 2d_2$ folgt aus Lemma 4.9(b): $\Delta > d_2$. Satz 4.10 liefert deshalb:

$$0 \;\leq\; \lim_{\varepsilon \to 0} \frac{P_e^{(1)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} \;\leq\; \lim_{\varepsilon \to 0} \frac{P_e'(\varepsilon)}{P_e^{(2)}(\varepsilon)} \;=\; 0 \,.$$

Beim Vergleich der Restfehlerwahrscheinlichkeiten werden wir annehmen, daß $\Delta > d_2$ gilt. Dies ist z. B. dann der Fall, wenn $\mathcal{C}_1$ und $\mathcal{C}_2$ die Kløve–Miller–Bedingung $d_1 > 2d_2$ erfüllen (s. Kor. 4.11). Für hinreichend kleine ε erhalten wir dann $P_e^{(1)}(\varepsilon) \leq P_e'(\varepsilon) < P_e^{(2)}(\varepsilon)$.

5.3.1 Vergleich mit ARQ-Systemen

Wir wollen unser HII-ARQ-System zunächst weiter mit dem ARQ-System vergleichen. Im Fall kleiner Bitfehlerwahrscheinlichkeiten besteht zwischen den Restfehlerwahrscheinkeiten der beiden Übertragungssysteme die folgende Beziehung:

Satz 5.5 *Falls $\Delta > d_2$ gilt, folgt für die Restfehlerwahrscheinlichkeiten P_R^{ARQ} und P_R^{HII}:*

$$\lim_{\varepsilon \to 0} \frac{P_R^{\mathrm{HII}}(\varepsilon)}{P_R^{\mathrm{ARQ}}(\varepsilon)} \;=\; 1\,.$$

Beweis: Wegen $\Delta > d_2$ gilt: $\lim\limits_{\varepsilon \to 0} \dfrac{P_e^{(1)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} = 0\,.$
Aus den Sätzen 3.4 und 5.2 folgt deshalb:

$$\begin{aligned}
\frac{P_R^{\mathrm{HII}}(\varepsilon)}{P_R^{\mathrm{ARQ}}(\varepsilon)} &\leq \frac{P_e^{(2)}(\varepsilon)[1+P_d^{(2)}(\varepsilon)] + P_e^{(1)}(\varepsilon)}{1 - \left[(P_d^{(2)}(\varepsilon))^2 - P_c^{(1)}(\varepsilon) - P_e^{(1)}(\varepsilon)\right]} \cdot \frac{1-P_d^{(2)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} \\
&\leq \frac{P_e^{(2)}(\varepsilon)[1+P_d^{(2)}(\varepsilon)] + P_e^{(1)}(\varepsilon)}{1 - (P_d^{(2)}(\varepsilon))^2} \cdot \frac{1-P_d^{(2)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} \\
&= 1 + \frac{P_e^{(1)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} \cdot \frac{1}{1+P_d^{(2)}(\varepsilon)} \quad \stackrel{\varepsilon \to 0}{\longrightarrow} \quad 1\,.
\end{aligned}$$

Aus (5.1) folgt $P_R^{\mathrm{HII}} \geq \mathrm{Prob}(A_0^e) = P_e^{(2)}$. Hiermit erhalten wir

$$\frac{P_R^{\mathrm{HII}}(\varepsilon)}{P_R^{\mathrm{ARQ}}(\varepsilon)} \;\geq\; P_e^{(2)}(\varepsilon)\,\frac{1-P_d^{(2)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} \;=\; 1 - P_d^{(2)}(\varepsilon) \quad \stackrel{\varepsilon \to 0}{\longrightarrow} \quad 1\,.$$

q.e.d.

Bemerkung: Wang und Lin erhalten in ihrer Arbeit unter der Annahme, daß $\mathrm{Prob}(G_i^e \mid A_0^d \cap B_1^d \cap \ldots \cap B_i^d) \leq P_e^{(2)}$, also $P_e^{(1)} \leq P_e^{(2)} P_d^{(2)^2}$ gilt ([18, S. 602]):

$$P_R^{\mathrm{HII}} \leq \left(1 + P_d^{(2)^2}\right) P_R^{\mathrm{ARQ}}\,.$$

Unter der obigen Annahme gilt dies auch für unser Ergebnis (vgl. Bemerkung (b) zu Satz 5.2).

Das HII-ARQ-System erreicht also die gleiche Fehlerordnung wie ein einfaches ARQ-System. Wir können aber unter gewissen Voraussetzungen den Durchsatz erhöhen. Bei der Verwendung des Selective-Repeat-Protokolls gilt dies in jedem Fall. Für die Analyse des Go-Back-N-Protokolls müssen wir die Rückgrifftiefe N näher betrachten (vgl. Abschnitt 1.2). Beim einfachen ARQ-System ist der Decodieraufwand geringer als beim HII-ARQ-System. Wir müssen das Go-Back-N-HII-ARQ-System deshalb mit einem Go-Back-M-ARQ-System vergleichen, wobei im allgemeinen $M \neq N$ sei.

Satz 5.6 *Für die Durchsatzraten η^{ARQ} und η^{HII} gilt*

a) bei Verwendung des Selective-Repeat-Protokolls:

$$\eta_{\mathrm{SR}}^{\mathrm{HII}} \geq \eta_{\mathrm{SR}}^{\mathrm{ARQ}} ,$$

b) bei Verwendung des Go-Back-Protokolls:

$$\eta_{\mathrm{GB}N}^{\mathrm{HII}} \geq \eta_{\mathrm{GB}M}^{\mathrm{ARQ}} , \quad \textit{falls} \quad N \leq M \, \frac{1 - \left(P_d^{(2)^2} - P_c^{(1)} - P_e^{(1)}\right)}{1 - P_d^{(2)^2} - \left(1 - P_d^{(2)}\right) \frac{P_c^{(1)} + P_e^{(1)}}{P_d^{(2)}}} \quad \textit{ist.}$$

Beweis: Zu (a): Nach (3.1) und Satz 5.3(a) gilt:

$$\begin{aligned} \eta_{\mathrm{SR}}^{\mathrm{HII}} &\geq \frac{1 - P_d^{(2)^2} + P_c^{(1)} + P_e^{(1)}}{1 + P_d^{(2)}} \cdot \frac{k}{n} = \left(\frac{1 - P_d^{(2)^2}}{1 + P_d^{(2)}} + \frac{P_c^{(1)} + P_e^{(1)}}{1 + P_d^{(2)}} \right) \cdot \frac{k}{n} \\ &= \left(1 + \frac{P_c^{(1)} + P_e^{(1)}}{1 - P_d^{(2)^2}} \right) (1 - P_d^{(2)}) \frac{k}{n} = \left(1 + \frac{P_c^{(1)} + P_e^{(1)}}{1 - P_d^{(2)^2}} \right) \eta_{\mathrm{SR}}^{\mathrm{ARQ}} \\ &\geq \eta_{\mathrm{SR}}^{\mathrm{ARQ}} . \end{aligned}$$

Zu (b): Nach (3.2) und Satz 5.3(b) gilt:

$$\begin{aligned} \eta_{\mathrm{GB}N}^{\mathrm{HII}} &\geq \frac{1 - P_d^{(2)^2} + P_c^{(1)} + P_e^{(1)}}{1 + NP_d^{(2)} + (N-1)\left(P_d^{(2)^2} - P_c^{(1)} - P_e^{(1)}\right)} \cdot \frac{k}{n} , \\ \eta_{\mathrm{GB}M}^{\mathrm{ARQ}} &= \frac{1 - P_d^{(2)}}{1 + (M-1)P_d^{(2)}} \cdot \frac{k}{n} . \end{aligned}$$

Mit elementaren Umformungen erhalten wir:

$$\frac{1 - P_d^{(2)^2} + P_c^{(1)} + P_e^{(1)}}{1 + NP_d^{(2)} + (N-1)\left(P_d^{(2)^2} - P_c^{(1)} - P_e^{(1)}\right)} \cdot \frac{1 + (M-1)P_d^{(2)}}{1 - P_d^{(2)}} \geq 1$$

$$\iff N \leq M \, \frac{1 - P_d^{(2)^2} + P_c^{(1)} + P_e^{(1)}}{1 - P_d^{(2)^2} - \left(1 - P_d^{(2)}\right) \frac{P_c^{(1)} + P_e^{(1)}}{P_d^{(2)}}} =: M\vartheta .$$

Also impliziert $N \leq M\vartheta$ auch $\eta_{\mathrm{GB}N}^{\mathrm{HII}} \geq \eta_{\mathrm{GB}M}^{\mathrm{ARQ}}$. q.e.d.

Bemerkungen:

a) Wang und Lin erhalten in ihrer Analyse unter der Annahme, daß $P_e^{(1)} \approx 0$ und $P_e^{(2)} \approx 0$ gilt (vgl. Bem. (b) zu Satz 5.3) ([18, S. 596f]): $\eta_{\mathrm{SR}}^{\mathrm{HII}} \geq G_H \, \eta_{\mathrm{SR}}^{\mathrm{ARQ}}$, wobei

$$G_H \;=\; \frac{1 + (1 - P_c^{(2)})\Big[P_c^{(2)} + (1 - P_c^{(2)})q_1\Big]/P_c^{(2)}}{2 - P_c^{(2)}} \quad \text{und} \quad q_1 \;=\; \frac{P_c^{(1)}}{{P_d^{(2)}}^2} \quad \text{seien.}$$

Auch hier gilt natürlich $G_H \geq 1$. Wang und Lin bezeichnen G_H als „Durchsatz–Gewinn–Faktor“ (throughput gain factor).
Aus den obigen Annahmen folgt $1 - P_c^{(2)} \approx P_d^{(2)}$ und daher

$$G_H \;\approx\; 1 \,+\, \frac{P_c^{(1)}}{1 - {P_d^{(2)}}^2} \;\approx\; 1 \,+\, \frac{P_c^{(1)} + P_e^{(1)}}{1 - {P_d^{(2)}}^2}\,,$$

so daß unser Ergebnis mit dem von Wang und Lin übereinstimmt.

b) Der Beweis der Aussage (a) zeigt, daß

$$\eta_{\mathrm{SR}}^{\mathrm{HII}} \;\geq\; \left(1 + \frac{P_c^{(1)} + P_e^{(1)}}{1 - {P_d^{(2)}}^2}\right) \eta_{\mathrm{SR}}^{\mathrm{ARQ}}$$

ist. Nach Definition von $P_c^{(1)}$ und $P_e^{(1)}$ gilt $P_c^{(1)} \leq {P_d^{(2)}}^2$ und $P_e^{(1)} \leq {P_d^{(2)}}^2$. Wegen $\lim\limits_{\varepsilon \to 0} P_d^{(2)}(\varepsilon) = 0$ folgt

$$\lim_{\varepsilon \to 0} \left(1 + \frac{P_c^{(1)}(\varepsilon) + P_e^{(1)}(\varepsilon)}{1 - \left(P_d^{(2)}(\varepsilon)\right)^2}\right) \;=\; 1\,.$$

Dies bedeutet, daß das Selective-Repeat-HII-ARQ-System bei kleinen Bitfehlerwahrscheinlichkeiten nur eine geringe Verbesserung des Durchsatzes gegenüber dem einfachen Selective-Repeat-ARQ-System bewirkt.

c) Bei Verwendung des Go-Back-Protokolls gilt für ϑ (s. Beweis der Aussage (b)):

$$\vartheta \;=\; \frac{1 - {P_d^{(2)}}^2 + P_c^{(1)} + P_e^{(1)}}{1 - {P_d^{(2)}}^2 - \left(1 - P_d^{(2)}\right)\frac{P_c^{(1)} + P_e^{(1)}}{P_d^{(2)}}} \;\geq\; 1\,.$$

Ferner folgt aus Lemma 5.1 und Korollar 3.2:

$$0 \;\leq\; \lim_{\varepsilon \to 0} \frac{P_c^{(1)}(\varepsilon) + P_e^{(1)}(\varepsilon)}{P_d^{(2)}(\varepsilon)} \;\leq\; \lim_{\varepsilon \to 0} \frac{\left(P_d^{(2)}(\varepsilon)\right)^2}{P_d^{(2)}(\varepsilon)} \;=\; 0\,.$$

Hieraus erhalten wir mit Korollar 3.2: $\lim\limits_{\varepsilon \to 0} \vartheta(\varepsilon) = 1\,.$
Das HII-ARQ-System bietet hier also nur bei großen Bitfehlerwahrscheinlichkeiten oder sehr geringer Differenz der Rückgrifftiefen Vorteile gegenüber dem ARQ-System.

5.3.2 Vergleich mit Typ-I Hybrid-ARQ-Systemen

Wir wollen unser HII-ARQ-System nun auch noch mit dem HI-ARQ-System vergleichen. Mit den Bezeichnungen aus Abschnitt 5.2.1 seien

$$\begin{aligned} P'_c &:= \text{Prob}(G_i^c)\,, \quad i \in I\!N\,, \\ P'_e &:= \text{Prob}(G_i^e)\,, \quad i \in I\!N\,, \\ P'_d &:= \text{Prob}(G_i^d)\,, \quad i \in I\!N\,. \end{aligned}$$

Dies entspricht den durch Satz 4.1 gegebenen Wahrscheinlichkeiten. Im Fall $\Delta > d_2$ gilt nach Satz 4.10 und Satz 4.14 für hinreichend kleine ε:

$$P'_e(\varepsilon) < P_e^{(2)}(\varepsilon) \quad \text{und} \quad P'_d(\varepsilon) < P_d^{(2)}(\varepsilon)\,.$$

Wir haben bereits gesehen, daß das HII-ARQ-System die gleiche Fehlerordnung wie das einfache ARQ-System erreicht, falls $\Delta > d_2$ gilt (Satz 5.5). Unter dieser Voraussetzung erreicht das HI-ARQ-System jedoch eine bessere Fehlerordnung als das einfache ARQ-System (Satz 4.10). Eine entsprechende Aussage können wir auch für das HII-ARQ-System zeigen:

Satz 5.7 *Falls $\Delta > d_2$ gilt, folgt für die Restfehlerwahrscheinlichkeiten P_R^{HI} und P_R^{HII}:*

$$\lim_{\varepsilon \to 0} \frac{P_R^{\text{HI}}(\varepsilon)}{P_R^{\text{HII}}(\varepsilon)} \;=\; 0\,.$$

Beweis: Wegen $\Delta > d_2$ gilt: $\lim\limits_{\varepsilon \to 0} \dfrac{P'_e(\varepsilon)}{P_e^{(2)}(\varepsilon)} = 0\,.$

Aus (5.1) folgt $P_R^{\text{HII}} \geq \text{Prob}(A_0^e) = P_e^{(2)}$. Hiermit erhalten wir

$$0 \;\leq\; \frac{P_R^{\text{HI}}(\varepsilon)}{P_R^{\text{HII}}(\varepsilon)} \;\leq\; \frac{P'_e(\varepsilon)}{1 - P'_d(\varepsilon)} \cdot \frac{1}{P_e^{(2)}(\varepsilon)} \;=\; \frac{P'_e(\varepsilon)}{P_e^{(2)}(\varepsilon)} \cdot \frac{1}{1 - P'_d(\varepsilon)} \quad \stackrel{\varepsilon \to 0}{\longrightarrow} \quad 0\,.$$

q.e.d.

Der schlechteren Restfehlerwahrscheinlichkeit steht aber eine deutliche Verbesserung des Durchsatzes gegenüber. Beim Vergleich der Durchsatzraten der Go-Back-Systeme ist allerdings zu beachten, daß sich die Rückgrifftiefe des HII-ARQ-Systems auf Blöcke der Länge n bezieht, die des HI-ARQ-Systems jedoch auf Blöcke der Länge $2n$.

Satz 5.8 *Für die Durchsatzraten η^{HI} und η^{HII} gilt*

a) bei Verwendung des Selective-Repeat-Protokolls:

$$\eta_{\text{SR}}^{\text{HII}} \;\geq\; 2\,\frac{1 - P_d^{(2)}}{1 - P'_d}\,\eta_{\text{SR}}^{\text{HI}}\,,$$

b) bei Verwendung des Go-Back-Protokolls:

$$\lim_{\varepsilon \to 0} \frac{\eta_{\text{GB}N}^{\text{HII}}(\varepsilon)}{\eta_{\text{GB}N'}^{\text{HI}}(\varepsilon)} \;\geq\; 2\,.$$

Beweis: Zu (a): Nach (4.10) und Satz 5.3(a) gilt:

$$\begin{aligned}\frac{\eta_{\mathrm{SR}}^{\mathrm{HII}}}{\eta_{\mathrm{SR}}^{\mathrm{HI}}} &\geq \frac{1-{P_d^{(2)}}^2+P_c^{(1)}+P_e^{(1)}}{1+P_d^{(2)}}\cdot\frac{k}{n}\cdot\frac{1}{1-P_d'}\cdot\frac{2n}{k}\\ &= 2\,\frac{1-P_d^{(2)}}{1-P_d'}\left(1+\frac{P_c^{(1)}+P_e^{(1)}}{1-{P_d^{(2)}}^2}\right) \geq 2\,\frac{1-P_d^{(2)}}{1-P_d'}\,.\end{aligned}$$

Zu (b): Nach (4.11) und Satz 5.3(b) gilt:

$$\begin{aligned}&\frac{\eta_{\mathrm{GBN}}^{\mathrm{HII}}(\varepsilon)}{\eta_{\mathrm{GBN'}}^{\mathrm{HI}}(\varepsilon)}\\ &\geq \frac{1-\left(P_d^{(2)}(\varepsilon)\right)^2+P_c^{(1)}(\varepsilon)+P_e^{(1)}(\varepsilon)}{1+NP_d^{(2)}(\varepsilon)+(N-1)\left[\left(P_d^{(2)}(\varepsilon)\right)^2-P_c^{(1)}(\varepsilon)-P_e^{(1)}(\varepsilon)\right]}\cdot\frac{k}{n}\cdot\frac{1+(N'-1)P_d'(\varepsilon)}{1-P_d'(\varepsilon)}\cdot\frac{2n}{k}\\ &\geq 2\cdot\frac{1-\left(P_d^{(2)}(\varepsilon)\right)^2+P_c^{(1)}(\varepsilon)+P_e^{(1)}(\varepsilon)}{\left(1+P_d^{(2)}(\varepsilon)\right)\left(1+(N-1)P_d^{(2)}(\varepsilon)\right)}\cdot\frac{1+(N'-1)P_d'(\varepsilon)}{1-P_d'(\varepsilon)}\\ &= 2\cdot\left(\frac{1-P_d^{(2)}(\varepsilon)}{1-P_d'(\varepsilon)}+\frac{P_c^{(1)}(\varepsilon)+P_e^{(1)}(\varepsilon)}{\left(1+P_d^{(2)}(\varepsilon)\right)\left(1-P_d'(\varepsilon)\right)}\right)\cdot\frac{1+(N'-1)P_d'(\varepsilon)}{1+(N-1)P_d^{(2)}(\varepsilon)}\\ &\stackrel{\varepsilon\to 0}{\longrightarrow} 2\end{aligned}$$

(aus Lemma 5.1 folgt $0\leq P_c^{(1)}(\varepsilon)+P_e^{(1)}(\varepsilon)\leq\left(P_d^{(2)}(\varepsilon)\right)^2\to 0$ für $\varepsilon\to 0$).

q.e.d.

Bemerkungen:

a) Der Beweis zeigt, daß die Verbesserung des Durchsatzes um den Faktor 2 durch die geringe Informationsrate des Codes $\mathcal{C}$ (kleiner als $\frac{1}{2}$) verursacht wird.

b) Nach Satz 4.14 gilt (unter der üblichen Voraussetzung $\Delta > d_2$) $P_d'(\varepsilon) < P_d^{(2)}(\varepsilon)$ für hinreichend kleine ε. Hieraus erhalten wir für diese ε:

$$\phi(\varepsilon) := \frac{1-P_d^{(2)}(\varepsilon)}{1-P_d'(\varepsilon)} \leq 1\,.$$

Es gilt aber $\lim\limits_{\varepsilon\to 0}\phi(\varepsilon)=1$. Dies impliziert:

$$\lim_{\varepsilon\to 0}\frac{\eta_{\mathrm{SR}}^{\mathrm{HII}}(\varepsilon)}{\eta_{\mathrm{SR}}^{\mathrm{HI}}(\varepsilon)} \geq 2\,.$$

c) Die Aussage (b) liefert uns leider nur Informationen für den Fall sehr kleiner Bitfehlerwahrscheinlichkeiten; dies gilt insbesondere, wenn die Differenz der Rückgrifftiefen groß

ist. Wir können jedoch kein interessanteres Ergebnis erhalten. Betrachten wir wieder

$$\frac{\eta_{\mathrm{GB}N}^{\mathrm{HII}}(\varepsilon)}{\eta_{\mathrm{GB}N'}^{\mathrm{HI}}(\varepsilon)} \stackrel{!}{\geq} 1\,,$$

so ergibt sich durch elementare Umformungen:

$$\frac{1-{P_d^{(2)}}^2+P_c^{(1)}+P_e^{(1)}}{1+NP_d^{(2)}+(N-1)\left({P_d^{(2)}}^2-P_c^{(1)}-P_e^{(1)}\right)}\cdot\frac{1+(N'-1)P_d'}{1-P_d'} \quad\geq\quad \frac{1}{2}$$

$$\Longleftrightarrow \quad N' \quad\geq\quad \frac{N}{2}\cdot\frac{1-P_d'}{P_d'}\cdot\frac{P_d^{(2)}+{P_d^{(2)}}^2-P_c^{(1)}-P_e^{(1)}}{1-{P_d^{(2)}}^2+P_c^{(1)}+P_e^{(1)}}-\frac{1-P_d'}{2P_d'} \quad=:\quad \frac{N}{2}\,\sigma-\tau\,.$$

Nun folgt aber aus Lemma 5.1 und Satz 4.14 (unter der üblichen Voraussetzung $\Delta > d_2$):

$$\lim_{\varepsilon\to 0}\sigma(\varepsilon)=\infty \quad\text{und}\quad \lim_{\varepsilon\to 0}\tau(\varepsilon)=\infty\,.$$

Wir können deshalb keine Aussage machen, für welche N, N' und ε die Bedingung $N' \geq \frac{N}{2}\,\sigma(\varepsilon)-\tau(\varepsilon)$ gilt.

5.3.3 Resümee

Das hier betrachtete HII-ARQ-System erreicht unter gewissen Voraussetzungen die gleiche Fehlerordnung wie das einfache ARQ-System und zeichnet sich zusätzlich durch eine höhere Durchsatzrate aus. Letzteres gilt insbesondere im Fall größerer Bitfehlerwahrscheinlichkeiten. Diesem Vorteil steht jedoch der hohe technische Aufwand gegenüber.

Im Vergleich zum HI-ARQ- schneidet das HII-ARQ-System nicht so gut ab. Es erreicht nur eine deutlich schlechtere Fehlerordnung. Allerdings ist bei kleinen Bitfehlerwahrscheinlichkeiten eine Verbesserung des Durchsatzes möglich. Der technische Aufwand des HII-ARQ- ist nur geringfügig höher als der des HI-ARQ-Systems.

Insgesamt gesehen scheint die Verwendung eines HII-ARQ-Systems insbesondere dann sinnvoll zu sein, wenn bei variierendem Störverhalten des Übertragungskanals hohe Durchsatzraten benötigt werden. Hier verbindet das HII-ARQ-System die Vorteile des ARQ- mit denen des HI-ARQ-Systems. Bei der Wahl der Codes ist wie beim HI-ARQ-System die Einhaltung der Kløve–Miller–Bedingung zu beachten. æ

Kapitel 6

Verallgemeinerte Typ-II Hybrid-ARQ-Systeme

Das im vorherigen Kapitel betrachtete HII-ARQ-System wurde von Krishna und Morgera verallgemeinert (s. [8, S. 104ff], [9], [12, S. 188ff], [15]).

6.1 Spezifikation

Wenn bei der zweiten Reübertragung des HII-ARQ-Systems statt des Datenblockes ein weiterer Block mit Redundanzbits des inneren, fehlerkorrigierenden Codes übertragen wird, kann dieser zusammen mit den beiden vorherigen Blöcken decodiert werden. Auf diese Weise können beliebig viele verschiedene Blöcke übertragen werden, bevor der erste Block wiederholt wird. Ein derartiges Übertragungssystem wird als *verallgemeinertes Typ-II Hybrid-ARQ-System (GH-ARQ)* bezeichnet ([8, S. 104], [9], [12, S. 188]). Die Anzahl der zu übertragenden Blöcke vor der ersten Wiederholung eines Blockes wird *Tiefe* des GH-ARQ-Systems bzw. des inneren Codes genannt (s. z. B. [12, S. 188]).

Die Übertragungsprozedur des GH-ARQ-Systems entspricht der des HII-ARQ-Systems.

Es sei $\mathcal{C}_2$ ein linearer fehlererkennender (n, k, d_2)-Code über $GF(2)$ mit Generatormatrix $\mathbf{G}_2$. Ferner sei $\mathcal{C}_1$ ein linearer fehlerkorrigierender (mn, n, d_1)-Code über $GF(2)$, die Generatormatrix $\mathbf{G}_1$ von $\mathcal{C}_1$ habe die Gestalt

$$\mathbf{G}_1 \;=\; \left[\tilde{\mathbf{G}}_0 \mid \tilde{\mathbf{G}}_1 \mid \ldots \mid \tilde{\mathbf{G}}_{m-1}\right] \;,$$

wobei jede der Matrizen $\tilde{\mathbf{G}}_i$ $(i = 0, \ldots, m-1)$ eine nicht-singuläre $(n \times n)$-Matrix über $GF(2)$ sei. Der Code $\mathcal{C}_1$ hat also die Tiefe m.

Für $i = 1, \ldots, m-1$ bezeichne $\mathcal{C}_1^{(i)}$ den durch

$$\mathbf{G}_1^{(i)} \;:=\; \left[\tilde{\mathbf{G}}_0 \mid \tilde{\mathbf{G}}_1 \mid \ldots \mid \tilde{\mathbf{G}}_i\right]$$

definierten $((i+1)n, n, \delta_i)$-Code über $GF(2)$. Wegen der Nicht-Singularität der Matrizen $\tilde{\mathbf{G}}_0, \ldots, \tilde{\mathbf{G}}_{m-1}$ gilt $\delta_i < \delta_j$ für alle $1 \leq i < j \leq m-1$.

Es ist nicht unbedingt notwendig, die Invertierbarkeit der Matrizen $\tilde{\mathbf{G}}_1, \ldots, \tilde{\mathbf{G}}_{m-1}$ vorauszusetzen, lediglich $\tilde{\mathbf{G}}_0$ muß invertierbar sein (s. [12, S. 189]). Falls eine Matrix $\tilde{\mathbf{G}}_i$, $i \in \{1, \ldots, m-1\}$, singulär ist, kann nach der i-ten Reübertragung keine Fehlererkennung durchgeführt werden, das empfangene Wort wird dann zusammen mit den vorherigen decodiert.

Die Matrizen $\tilde{\mathbf{G}}_0, \ldots, \tilde{\mathbf{G}}_{m-1}$ sollten aber stets so gewählt werden, daß $\delta_i < \delta_j$ für alle $1 \leq i < j \leq m-1$ gilt.

Aus technischen Gründen ist es sinnvoll, den Code $\mathcal{C}_1$ so zu wählen, daß für $\mathcal{C}_1^{(1)}, \ldots, \mathcal{C}_1^{(m-1)}$ jeweils derselbe Decodierer verwendet werden kann (s. [12, S. 190]). Auf der Seite des Empfängers werden dann nur dieser Decodierer und die (höchstens) m Invertierer benötigt.

Die bisher an den inneren fehlerkorrigierenden Code gestellten Anforderungen schränken dessen Wahl stark ein. Sie werden aber z. B. von den Krishna–Morgera–Codes (KM–Codes, s. Abschnitt 2.6) erfüllt.

6.2 Leistungsanalyse

Für das oben beschriebene GH-ARQ-System sollen zusätzlich zu den in Abschnitt 1.1 genannten die folgenden Annahmen gelten:

- Der Übertragungskanal sei ein gedächtnisloser binärer symmetrischer Kanal mit Bitfehlerwahrscheinlichkeit $\varepsilon \in (0, \frac{1}{2}]$.
- Der Rückkopplungskanal sei ungestört.
- Der Sender und der Empfänger (bei Verwendung des Selective-Repeat-Protokolls) verfügen über unendlich große Speicher.
- Die Matrizen $\tilde{\mathbf{G}}_0, \ldots, \tilde{\mathbf{G}}_{m-1}$ seien nicht-singulär.

6.2.1 Notation

Wir verwenden im folgenden für die mit der Fehlererkennung und -korrektur verbundenen Ereignisse die gleiche Notation wie bei der Analyse des HII-ARQ-Systems (vgl. Abschnitt 5.2.1). Außerdem sei wieder

$$P_c^{(2)} := \text{Prob}(A_0^c) = \text{Prob}(B_\nu^c) \quad \forall\, \nu \in \mathbb{N}\,.$$

Wir setzen hier jedoch nicht die Äquivalenz der durch $\mathbf{G}_2\tilde{\mathbf{G}}_i$, $i = 0, \ldots, m-1$, definierten verketteten Codes voraus, daher können wir verschiedene Decodierfehlerwahrscheinlichkeiten beobachten. Für $i = 0, \ldots, m-1$ seien (mit $B_0^c := A_0^c$, $B_0^d := A_0^d$ und $B_0^e := A_0^e$)

$$\begin{aligned} P_e^{(2,i)} &:= \text{Prob}(B_i^e) = \text{Prob}(B_{\nu m+i}^e) \quad \forall\, \nu \in \mathbb{N}\,, \\ P_d^{(2,i)} &:= \text{Prob}(B_i^d) = \text{Prob}(B_{\nu m+i}^d) \quad \forall\, \nu \in \mathbb{N}\,. \end{aligned}$$

Zur Vereinfachung werden wir oft nur die Maxima und Minima dieser Wahrscheinlichkeiten betrachten, es seien

$$\begin{aligned} P_e^{(2,\max)} &:= \max_{i=0,\ldots,m-1} P_e^{(2,i)} , \\ P_e^{(2,\min)} &:= \min_{i=0,\ldots,m-1} P_e^{(2,i)} , \\ P_d^{(2,\max)} &:= \max_{i=0,\ldots,m-1} P_d^{(2,i)} \quad = \quad 1 - P_c^{(2)} - P_e^{(2,\min)} , \\ P_d^{(2,\min)} &:= \min_{i=0,\ldots,m-1} P_d^{(2,i)} \quad = \quad 1 - P_c^{(2)} - P_e^{(2,\max)} . \end{aligned}$$

Wegen $\mathcal{C}_1^{(m-1)} = \mathcal{C}_1$ seien ferner

$$\begin{aligned} P_c^{(1)} &:= \mathrm{Prob}(A_0^d \cap B_1^d \cap \ldots \cap B_{m-1}^d \cap G_{m-1}^c) \\ &= \mathrm{Prob}(B_{\nu m}^d \cap \ldots \cap B_{(\nu+1)m-1}^d \cap G_{(\nu+1)m-1}^c) \quad \forall\, \nu \in I\!N , \\ P_e^{(1)} &:= \mathrm{Prob}(A_0^d \cap B_1^d \cap \ldots \cap B_{m-1}^d \cap G_{m-1}^e) \\ &= \mathrm{Prob}(B_{\nu m}^d \cap \ldots \cap B_{(\nu+1)m-1}^d \cap G_{(\nu+1)m-1}^e) \quad \forall\, \nu \in I\!N . \end{aligned}$$

6.2.2 Zuverlässigkeit

Als Maß für die Zuverlässigkeit des GH-ARQ-Systems dient uns wieder die Restfehlerwahrscheinlichkeit P_R (s. Kap. 3). Hierfür gilt offensichtlich (s. [12, S. 225]):

$$P_R \;=\; \mathrm{Prob}(A_0^e) + \sum_{i=1}^{\infty} \mathrm{Prob}(A_0^d \cap E_1^d \cap \ldots \cap E_{i-1}^d \cap E_i^e) . \tag{6.1}$$

Wie bei der Analyse des HII-ARQ-Systems haben wir auch hier das Problem, daß die Wahrscheinlichkeiten $\mathrm{Prob}(A_0^d \cap E_1^d \cap \ldots \cap E_{i-1}^d \cap E_i^e)$ im allgemeinen nur schwer bestimmt werden können. Deshalb schätzen wir die Restfehlerwahrscheinlichkeit ab, indem wir ein vereinfachtes Übertragungssystem $\widehat{\mathrm{GH}}$-ARQ betrachten (vgl. Abschnitt 5.2.2). Wir nehmen an, daß das $\widehat{\mathrm{GH}}$-ARQ-System die Fehlerkorrektur nur nach jeder m-ten Übertragung durchführt (vgl. [12, S. 228]). So erhalten wir den folgenden Satz (der Beweis erfolgt zum Teil in Anlehnung an [18, S. 604f]):

Satz 6.1 *Für die Restfehlerwahrscheinlichkeit P_R des GH-ARQ-Systems gilt (mit der Vereinbarung $\prod_{\mu=0}^{-1} \alpha_\mu := 1$):*

$$P_R \quad \leq \quad \frac{\sum_{k=0}^{m-1} \Big(\prod_{\mu=0}^{k-1} P_d^{(2,\mu)} \Big) P_e^{(2,k)} + P_e^{(1)}}{1 - \Big(\prod_{\mu=0}^{m-1} P_d^{(2,\mu)} - P_c^{(1)} - P_e^{(1)} \Big)} .$$

Beweis: Die Restfehlerwahrscheinlichkeit $\hat{P}_R$ des oben beschriebenen $\widehat{\text{GH}}$-ARQ-Systems ist gegeben durch

$$\hat{P}_R \;=\; \text{Prob}(A_0^e) + \sum_{i=1}^{\infty} \text{Prob}(A_0^d \cap \hat{E}_1^d \cap \ldots \cap \hat{E}_{i-1}^d \cap \hat{E}_i^e)\,, \tag{6.2}$$

$$\begin{array}{lll}
\text{mit} & \hat{E}_j^c := E_j^c = B_j^c + (B_j^d \cap G_j^c)\,, & \\
& \hat{E}_j^d := E_j^d = B_j^d \cap G_j^d\,, & \\
& \hat{E}_j^e := E_j^e = B_j^e + (B_j^d \cap G_j^e)\,, & \text{falls } j = \nu m - 1 \text{ für ein } \nu \in \mathbb{N} \\
\text{und} & \hat{E}_j^c := B_j^c\,, & \\
& \hat{E}_j^d := B_j^d\,, & \\
& \hat{E}_j^e := B_j^e & \text{sonst.}
\end{array}$$

Der Übertragungskanal ist gedächtnislos, deshalb sind die Ereignisse F_i, $i \in \mathbb{N}_0$, mit $F_0 \in \{A_0^c, A_0^d, A_0^e\}$ und $F_i \in \{B_i^c, B_i^d, B_i^e\}$ für $i \in \mathbb{N}$ stochastisch unabhängig. Aus der Konstruktion des $\widehat{\text{GH}}$-ARQ-Systems folgt außerdem für $i = \nu m + k$, $\nu \in \mathbb{N}_0$ und $0 \le k < m$, die stochastische Unabhängigkeit des Ereignisses $\hat{E}_i^e$ von $A_0^d, \hat{E}_1^d, \ldots, \hat{E}_{\nu m-1}^d$.

Für $\nu \in \mathbb{N}_0$ und $k \in \{0, \ldots, m-1\}$ gilt daher mit den Vereinbarungen $\hat{E}_0^d := A_0^d$ und $\prod_{\mu=0}^{-1} \alpha_\mu := 1$

$$\begin{aligned}
&\text{Prob}(A_0^d \cap \hat{E}_1^d \cap \ldots \cap \hat{E}_{i-1}^d \cap \hat{E}_i^e) \\
&= \left(\prod_{\mu=0}^{\nu-1} \text{Prob}(\hat{E}_{\mu m}^d \cap \ldots \cap \hat{E}_{(\mu+1)m-1}^d) \right) \cdot \text{Prob}(\hat{E}_{\nu m}^d \cap \ldots \cap \hat{E}_{\nu m+k-1}^d \cap \hat{E}_{\nu m+k}^e)\,.
\end{aligned}$$

Für $0 \le k < m-1$ erhalten wir

$$\begin{aligned}
&\text{Prob}(\hat{E}_{\nu m}^d \cap \ldots \cap \hat{E}_{\nu m+k-1}^d \cap \hat{E}_{\nu m+k}^e) \\
&\quad = \text{Prob}(B_{\nu m}^d \cap \ldots \cap B_{\nu m+k-1}^d \cap B_{\nu m+k}^e) \;=\; \left(\prod_{\mu=0}^{k-1} P_d^{(2,\mu)} \right) P_e^{(2,k)}
\end{aligned} \tag{6.3}$$

und für $k = m-1$

$$\begin{aligned}
&\text{Prob}(\hat{E}_{\nu m}^d \cap \ldots \cap \hat{E}_{(\nu+1)m-2}^d \cap \hat{E}_{(\nu+1)m-1}^e) \\
&\quad = \text{Prob}(B_{\nu m}^d \cap \ldots \cap B_{(\nu+1)m-2}^d \cap [B_{(\nu+1)m-1}^e + (B_{(\nu+1)m-1}^d \cap G_{(\nu+1)m-1}^e)]\,) \\
&\quad = \text{Prob}(B_{\nu m}^d \cap \ldots \cap B_{(\nu+1)m-2}^d \cap B_{(\nu+1)m-1}^e) \\
&\qquad + \text{Prob}(B_{\nu m}^d \cap \ldots \cap B_{(\nu+1)m-2}^d \cap B_{(\nu+1)m-1}^d \cap G_{(\nu+1)m-1}^e) \\
&\quad = \left(\prod_{\mu=0}^{m-2} P_d^{(2,\mu)} \right) P_e^{(2,m-1)} \;+\; P_e^{(1)}\,.
\end{aligned} \tag{6.4}$$

Analog erhalten wir für $0 \leq k < m-1$

$$\text{Prob}(\hat{E}^d_{\nu m} \cap \ldots \cap \hat{E}^d_{\nu m+k-1} \cap \hat{E}^c_{\nu m+k}) \;=\; \left(\prod_{\mu=0}^{k-1} P_d^{(2,\mu)}\right) P_c^{(2)} \tag{6.5}$$

und für $k = m-1$

$$\text{Prob}(\hat{E}^d_{\nu m} \cap \ldots \cap \hat{E}^d_{(\nu+1)m-2} \cap \hat{E}^c_{(\nu+1)m-1}) \;=\; \left(\prod_{\mu=0}^{m-2} P_d^{(2,\mu)}\right) P_c^{(2)} \;+\; P_c^{(1)} \;. \tag{6.6}$$

Hieraus ergibt sich für $j \in \mathbb{N}_0$:

$$\begin{aligned}
&\text{Prob}(\hat{E}^d_{jm} \cap \ldots \cap \hat{E}^d_{(j+1)m-1}) \qquad (6.7)\\
&= \quad \text{Prob}(B^d_{jm} \cap \ldots \cap B^d_{(j+1)m-2}) \cdot \text{Prob}(\hat{E}^d_{(j+1)m-1} \mid B^d_{jm} \cap \ldots \cap B^d_{(j+1)m-2})\\
&= \quad \left(\prod_{\mu=0}^{m-2} P_d^{(2,\mu)}\right) \left[1 - \text{Prob}(\hat{E}^c_{(j+1)m-1} + \hat{E}^e_{(j+1)m-1} \mid B^d_{jm} \cap \ldots \cap B_{(j+1)m-2})\right]\\
&\overset{(6.4),(6.6)}{=} \quad \left(\prod_{\mu=0}^{m-2} P_d^{(2,\mu)}\right) \left[\, 1 \;-\; \frac{\left(\prod_{\mu=0}^{m-2} P_d^{(2,\mu)}\right)\left(P_c^{(2)} + P_e^{(2,m-1)}\right) + P_c^{(1)} + P_e^{(1)}}{\prod_{\mu=0}^{m-2} P_d^{(2,\mu)}} \,\right]\\
&= \quad \prod_{\mu=0}^{m-1} P_d^{(2,\mu)} \;-\; P_c^{(1)} \;-\; P_e^{(1)} \;.
\end{aligned}$$

Insgesamt folgt für $i \in \mathbb{N}$:

$$\begin{aligned}
&\text{Prob}(A^d_0 \cap \hat{E}^d_1 \cap \ldots \cap \hat{E}^d_{i-1} \cap \hat{E}^e_i)\\
&= \begin{cases}
\left(\prod_{\mu=0}^{m-1} P_d^{(2,\mu)} - P_c^{(1)} - P_e^{(1)}\right)^{\nu} \cdot \left[\left(\prod_{\mu=0}^{m-2} P_d^{(2,\mu)}\right) P_e^{(2,m-1)} + P_e^{(1)}\right] & \\
\qquad\qquad \text{falls } i = (\nu+1)m-1 \text{ für ein } \nu \in \mathbb{N}_0\,, & \\
\left(\prod_{\mu=0}^{m-1} P_d^{(2,\mu)} - P_c^{(1)} - P_e^{(1)}\right)^{\nu} \cdot \left(\prod_{\mu=0}^{k-1} P_d^{(2,\mu)}\right) P_e^{(2,k)} & \\
\qquad \text{falls } i = \nu m + k \text{ für ein } \nu \in \mathbb{N}_0 \text{ und } k \in \{0,\ldots,m-1\}\,. &
\end{cases}
\end{aligned}$$

Wegen der absoluten Konvergenz der geometrischen Reihe in $[0,1)$ gilt also:

$$\begin{aligned}
\hat{P}_R &= \sum_{k=0}^{m-2}\sum_{\nu=0}^{\infty} \mathrm{Prob}(A_0^d \cap \hat{E}_1^d \cap \ldots \cap \hat{E}_{\nu m+k-1}^d \cap \hat{E}_{\nu m+k}^e) \\
&\quad + \sum_{\nu=0}^{\infty} \mathrm{Prob}(A_0^d \cap \hat{E}_1^d \cap \ldots \cap \hat{E}_{(\nu+1)m-2}^d \cap \hat{E}_{(\nu+1)m-1}^e) \\
&= \sum_{k=0}^{m-2}\sum_{\nu=0}^{\infty} \left(\prod_{\mu=0}^{m-1} P_d^{(2,\mu)} - P_c^{(1)} - P_e^{(1)}\right)^{\nu} \left(\prod_{\mu=0}^{k-1} P_d^{(2,\mu)}\right) P_e^{(2,k)} \\
&\quad + \sum_{\nu=0}^{\infty} \left(\prod_{\mu=0}^{m-1} P_d^{(2,\mu)} - P_c^{(1)} - P_e^{(1)}\right)^{\nu} \left[\left(\prod_{\mu=0}^{m-2} P_d^{(2,\mu)}\right) P_e^{(2,m-1)} + P_e^{(1)}\right] \\
&= \frac{\sum_{k=0}^{m-1}\left(\prod_{\mu=0}^{k-1} P_d^{(2,\mu)}\right) P_e^{(2,k)} + P_e^{(1)}}{1 - \left(\prod_{\mu=0}^{m-1} P_d^{(2,\mu)} - P_c^{(1)} - P_e^{(1)}\right)} \quad .
\end{aligned}$$

Aus (6.1) und (6.2) folgt $\hat{P}_R \geq P_R$. q.e.d.

Korollar 6.2 *Für die Restfehlerwahrscheinlichkeit P_R des GH-ARQ-Systems gilt:*

$$P_R \quad \leq \quad \frac{P_e^{(2,\max)} \frac{1-{P_d^{(2,\max)}}^m}{1-P_d^{(2,\max)}} + P_e^{(1)}}{1 - \left({P_d^{(2,\max)}}^m - P_c^{(1)} - P_e^{(1)}\right)} \, .$$

Beweis: Die Funktion $x \mapsto \frac{1}{1-x}$ ist monoton wachsend in $[0,1)$, daher ist

$$\left[1 - \left(\prod_{\mu=0}^{m-1} P_d^{(2,\mu)} - P_c^{(1)} - P_e^{(1)}\right)\right]^{-1} \leq \left[1 - \left({P_d^{(2,\max)}}^m - P_c^{(1)} - P_e^{(1)}\right)\right]^{-1} .$$

Außerdem gilt:

$$\sum_{k=0}^{m-1}\left(\prod_{\mu=0}^{k-1} P_d^{(2,\mu)}\right) P_e^{(2,k)} \leq P_e^{(2,\max)} \sum_{k=0}^{m-1} {P_d^{(2,\max)}}^k = P_e^{(2,\max)} \frac{1-{P_d^{(2,\max)}}^m}{1-P_d^{(2,\max)}} \, .$$

Mit Satz 6.1 erhalten wir die Behauptung. q.e.d.

Bemerkungen:

a) Krishna und Morgera orientieren sich bei ihrer Analyse ([8, S. 131 und S. 164ff], [12, S. 199f und S. 225ff]) an der Arbeit von Wang und Lin ([18]). Sie erhalten unter der Annahme, daß $\text{Prob}(G_i^e \mid A_0^d \cap B_1^d \cap \ldots \cap B_i^d) \leq P_e^{(2,\max)}$ gilt:

$$P_R \;\leq\; \left(1 + P_d^{(2,\max)^2}\right) \frac{P_e^{(2,\max)}}{1 - P_d^{(2,\max)}} \;.$$

In diesem Fall gilt aber $P_e^{(1)} \leq P_e^{(2,\max)} P_d^{(2,\max)^m} \leq P_e^{(2,\max)} P_d^{(2,\max)^2}$ und deshalb:

$$\begin{aligned}
\frac{P_e^{(2,\max)} \frac{1-P_d^{(2,\max)^m}}{1-P_d^{(2,\max)}} + P_e^{(1)}}{1 - \left(P_d^{(2,\max)^m} - P_c^{(1)} - P_e^{(1)}\right)} \;&\leq\; \frac{P_e^{(2,\max)} \frac{1-P_d^{(2,\max)^m}}{1-P_d^{(2,\max)}} + P_e^{(2,\max)} P_d^{(2,\max)^2}}{1 - P_d^{(2,\max)^m}} \\
&=\; P_e^{(2,\max)} \left(\frac{1}{1 - P_d^{(2,\max)}} + \frac{P_d^{(2,\max)^2}}{1 - P_d^{(2,\max)^m}} \right) \\
&\leq\; P_e^{(2,\max)} \frac{1 + P_d^{(2,\max)^2}}{1 - P_d^{(2,\max)}} \;.
\end{aligned}$$

b) Das Decodierverfahren des fehlerkorrigierenden Codes $\mathcal{C}_1$ sei ein Maximum–Likelihood–Verfahren. Dies gilt z. B. für KM-Codes (s. Abschnitt 2.6.4). Wie in Abschnitt 4.1 seien

$$Q_{\bar{c}_1} \;:=\; \{\bar{x} \in (GF(2))^{mn} \;:\; d(\bar{x}, \bar{c}_1) \leq d(\bar{x}, \bar{d}_1) \quad \forall\, \bar{d}_1 \in \mathcal{C}_1\} \;, \quad \bar{c}_1 \in \mathcal{C}_1$$

und

$$a(\bar{x}) \;:=\; \sharp\,\{\bar{c}_1 \in \mathcal{C}_1 \;:\; d(\bar{x}, \bar{c}_1) \leq d(\bar{x}, \bar{d}_1) \quad \forall\, \bar{d}_1 \in \mathcal{C}_1\} \;, \quad \bar{x} \in (GF(2))^{mn} \;.$$

Es sei $\mathcal{C}$ der durch gewöhnliche Codeverkettung von $\mathcal{C}_2$ und $\mathcal{C}_1$ definierte Code. Wie in Bemerkung (c) zu Satz 5.2 erhalten wir dann:

$$\begin{aligned}
P_c^{(1)} \;&=\; \sum_{\substack{\bar{x}=(\bar{x}_0,\ldots,\bar{x}_{m-1}) \in Q_{\bar{0}} \\ \bar{x}_0,\ldots,\bar{x}_{m-1} \notin \mathcal{C}_2}} \frac{1}{a(\bar{x})}\, \varepsilon^{w(\bar{x})}\, (1-\varepsilon)^{mn - w(\bar{x})} \;, \\
P_e^{(1)} \;&=\; \sum_{\bar{d} \in \mathcal{C} \setminus \{\bar{0}\}} \; \sum_{\substack{\bar{x}=(\bar{x}_0,\ldots,\bar{x}_{m-1}) \in Q_{\bar{d}} \\ \bar{x}_0,\ldots,\bar{x}_{m-1} \notin \mathcal{C}_2}} \frac{1}{a(\bar{x})}\, \varepsilon^{w(\bar{x})}\, (1-\varepsilon)^{mn - w(\bar{x})} \;.
\end{aligned}$$

6.2.3 Durchsatz

Wir nehmen wieder an, daß beim Sender und beim Empfänger (bei Verwendung des Selective-Repeat-Protokolls) unendlich große Speicher zur Verfügung stehen. Analog zu der Analyse des HII-ARQ-Systems erhalten wir für die mittlere Anzahl V der Übertragungen eines Informationswortes bei Verwendung des Selective-Repeat-Protokolls ([12, S. 201])

$$V_{\mathrm{SR}} \;=\; \mathrm{Prob}(A_0^c + A_0^e) + \sum_{i=1}^{\infty} (i+1)\,\mathrm{Prob}(A_0^d \cap E_1^d \cap \ldots \cap E_{i-1}^d \cap (E_i^c + E_i^e)) \tag{6.8}$$

und bei Verwendung des Go-Back-N-Protokolls

$$V_{\mathrm{GBN}} \;=\; \mathrm{Prob}(A_0^c + A_0^e) + \sum_{i=1}^{\infty} (iN+1)\,\mathrm{Prob}(A_0^d \cap E_1^d \cap \ldots \cap E_{i-1}^d \cap (E_i^c + E_i^e))\;. \tag{6.9}$$

Wir werden im folgenden wieder das Übertragungssystem $\widehat{\mathrm{GH}}$-ARQ betrachten und so eine untere Schranke für die Durchsatzrate erhalten.

Satz 6.3 *Für die Durchsatzrate η des GH-ARQ-Systems gilt*

a) bei Verwendung des Selective-Repeat-Protokolls:

$$\eta_{\mathrm{SR}} \;\geq\; \frac{1-P_d^{(2,\max)}}{1-P_d^{(2,\min)}} \cdot \frac{1-\left({P_d^{(2,\max)}}^m - P_c^{(1)} - P_e^{(1)}\right)}{\frac{1-{P_d^{(2,\max)}}^m}{1-P_d^{(2,\max)}}} \cdot \frac{k}{n}\;,$$

b) bei Verwendung des Go-Back-N-Protokolls:

$$\eta_{\mathrm{GBN}} \;\geq\; \frac{1-P_d^{(2,\max)}}{1-P_d^{(2,\min)}} \cdot \frac{1-\left({P_d^{(2,\max)}}^m - P_c^{(1)} - P_e^{(1)}\right)}{\frac{1-{P_d^{(2,\max)}}^m}{1-P_d^{(2,\max)}}\left[1+(N-1)P_d^{(2,\max)}\right] - (N-1)\left(P_c^{(1)} + P_e^{(1)}\right)} \cdot \frac{k}{n}\;.$$

Beweis: Nach (6.3) bis (6.7) gilt mit den dortigen Bezeichnungen und Vereinbarungen:

$$\mathrm{Prob}(A_0^d \cap \hat{E}_1^d \cap \ldots \cap \hat{E}_{i-1}^d \cap (\hat{E}_i^c + \hat{E}_i^e))$$

$$= \begin{cases} \left(\prod_{\mu=0}^{m-1} P_d^{(2,\mu)} - P_c^{(1)} - P_e^{(1)}\right)^{\nu} \cdot \left[\left(\prod_{\mu=0}^{m-2} P_d^{(2,\mu)}\right)\left(P_c^{(2)} + P_e^{(2,m-1)}\right) + \left(P_c^{(1)} + P_e^{(1)}\right)\right] \\ \qquad\qquad \text{falls } i = (\nu+1)m-1 \text{ für ein } \nu \in \mathbb{N}_0\,, \\ \left(\prod_{\mu=0}^{m-1} P_d^{(2,\mu)} - P_c^{(1)} - P_e^{(1)}\right)^{\nu} \cdot \left(\prod_{\mu=0}^{k-1} P_d^{(2,\mu)}\right)\left(P_c^{(2)} + P_e^{(2,k)}\right) \\ \qquad\qquad \text{falls } i = \nu m + k \text{ für ein } \nu \in \mathbb{N}_0 \text{ und } k \in \{0,\ldots,m-1\} \end{cases}$$

$$\leq \begin{cases} \left(P_d^{(2,\max)^m} - P_c^{(1)} - P_e^{(1)}\right)^\nu \left[P_d^{(2,\max)^{m-1}} \left(P_c^{(2)} + P_e^{(2,\max)}\right) + \left(P_c^{(1)} + P_e^{(1)}\right)\right] \\ \qquad\qquad \text{falls } i = (\nu+1)m - 1 \text{ für ein } \nu \in \mathbb{N}_0\,, \\ \left(P_d^{(2,\max)^m} - P_c^{(1)} - P_e^{(1)}\right)^\nu P_d^{(2,\max)^k} \left(P_c^{(2)} + P_e^{(2,\max)}\right) \\ \qquad\qquad \text{falls } i = \nu m + k \text{ für ein } \nu \in \mathbb{N}_0 \text{ und } k \in \{0,\ldots,m-1\}\,. \end{cases}$$

Außerdem gilt

$$1 - P_d^{(2,\max)} \;=\; P_c^{(2)} + P_e^{(2,\min)} \;\leq\; P_c^{(2)} + P_e^{(2,\max)}\,,$$

also

$$\frac{1 - P_d^{(2,\max)}}{P_c^{(2)} + P_e^{(2,\max)}} \;\leq\; 1\,. \tag{6.10}$$

Zu (a): Zur Vereinfachung seien $\rho^{(1)} := P_c^{(1)} + P_e^{(1)}$ und $\rho^{(2)} := P_c^{(2)} + P_e^{(2,\max)} = 1 - P_d^{(2,\min)}$. Aus (6.8) folgt (wegen der absoluten Konvergenz der geometrischen Reihe):

$$\begin{aligned} V_{\mathrm{SR}} \quad &\leq \quad \sum_{k=0}^{m-2} \sum_{\nu=0}^{\infty} (\nu m + k + 1)\, \mathrm{Prob}(A_0^d \cap \hat{E}_1^d \cap \ldots \cap \hat{E}_{\nu m+k-1}^d \cap (\hat{E}_{\nu m+k}^c + \hat{E}_{\nu m+k}^e)) \\ &\qquad + \sum_{\nu=0}^{\infty} (\nu+1)m\, \mathrm{Prob}(A_0^d \cap \hat{E}_1^d \cap \ldots \cap \hat{E}_{(\nu+1)m-2}^d \cap (\hat{E}_{(\nu+1)m-1}^c + \hat{E}_{(\nu+1)m-1}^e)) \\ &\leq \quad \rho^{(2)} \sum_{k=0}^{m-2} P_d^{(2,\max)^k} \sum_{\nu=0}^{\infty} (\nu m + k + 1) \left(P_d^{(2,\max)^m} - \rho^{(1)}\right)^\nu \\ &\qquad + m \left(P_d^{(2,\max)^{m-1}} \rho^{(2)} + \rho^{(1)}\right) \sum_{\nu=0}^{\infty} (\nu+1) \left(P_d^{(2,\max)^m} - \rho^{(1)}\right)^\nu \\ &= \quad \rho^{(2)} \sum_{k=0}^{m-2} P_d^{(2,\max)^k} \left[\frac{m(P_d^{(2,\max)^m} - \rho^{(1)})}{\left(1 - P_d^{(2,\max)^m} + \rho^{(1)}\right)^2} + \frac{k+1}{1 - P_d^{(2,\max)^m} + \rho^{(1)}}\right] \\ &\qquad + \frac{m\left(P_d^{(2,\max)^{m-1}} \rho^{(2)} + \rho^{(1)}\right)}{\left(1 - P_d^{(2,\max)^m} + \rho^{(1)}\right)^2} \\ &= \quad \frac{\rho^{(2)}}{\left(1 - P_d^{(2,\max)^m} + \rho^{(1)}\right)^2} \left[m \left(P_d^{(2,\max)^m} - \rho^{(1)}\right) \frac{1 - P_d^{(2,\max)^{m-1}}}{1 - P_d^{(2,\max)}}\right. \\ &\qquad\qquad + \left(1 - P_d^{(2,\max)^m} + \rho^{(1)}\right) \frac{1 - mP_d^{(2,\max)^{m-1}} + (m-1)P_d^{(2,\max)^m}}{\left(1 - P_d^{(2,\max)}\right)^2} \\ &\qquad\qquad \left. + m \left(P_d^{(2,\max)^{m-1}} + \frac{\rho^{(1)}}{\rho^{(2)}}\right)\right] \end{aligned}$$

$$
\begin{aligned}
&\overset{(6.10)}{\leq} \quad \frac{\rho^{(2)}}{\left(1-P_d^{(2,\max)}\right)^2\left(1-P_d^{(2,\max)^m}+\rho^{(1)}\right)^2} \cdot \\
&\quad \cdot \Big[m\left(P_d^{(2,\max)^m}-\rho^{(1)}\right)\left(1-P_d^{(2,\max)^{m-1}}\right)\left(1-P_d^{(2,\max)}\right) \\
&\qquad + \left(1-P_d^{(2,\max)^m}+\rho^{(1)}\right)\left(1-mP_d^{(2,\max)^{m-1}}+(m-1)P_d^{(2,\max)^m}\right) \\
&\qquad + m\left(1-P_d^{(2,\max)}\right)\left(P_d^{(2,\max)^{m-1}}\left(1-P_d^{(2,\max)}\right)+\rho^{(1)}\right)\Big] \\
&= \quad \frac{\rho^{(2)}}{1-P_d^{(2,\max)}} \cdot \frac{\frac{1-P_d^{(2,\max)^m}}{1-P_d^{(2,\max)}}}{1-P_d^{(2,\max)^m}+\rho^{(1)}} \quad .
\end{aligned}
$$

Mit $\eta_{\mathrm{SR}} = V_{\mathrm{SR}}^{-1}\frac{k}{n}$ erhalten wir die Aussage (a).

Zu (b): Analog zum Beweis von (a) erhalten wir aus (6.9) unter Verwendung von (6.10):

$$
V_{\mathrm{GB}N} \;\leq\; \frac{\rho^{(2)}}{1-P_d^{(2,\max)}} \cdot \frac{\left[1+(N-1)P_d^{(2,\max)}\right]\cdot\frac{1-P_d^{(2,\max)^m}}{1-P_d^{(2,\max)}} \;-\; (N-1)\rho^{(1)}}{1-P_d^{(2,\max)^m}+\rho^{(1)}} \quad .
$$

Wegen $\eta_{\mathrm{GB}N} = V_{\mathrm{GB}N}^{-1}\frac{k}{n}$ folgt hieraus die Aussage (b). q.e.d.

Bemerkungen:

a) Falls für $i = 0,\ldots,m-1$ die durch $\mathbf{G}_2\tilde{\mathbf{G}}_i$ definierten Codes äquivalent sind, gilt $P_d^{(2,i)} = P_d^{(2,\max)} = P_d^{(2,\min)}$ für alle $i \in \{0,\ldots,m-1\}$. Dies impliziert

$$
\frac{1-P_d^{(2,\max)}}{1-P_d^{(2,\min)}} \quad = \quad 1\,.
$$

b) Krishna und Morgera analysieren in ihren Arbeiten nur Systeme mit Selective-Repeat-Protokoll und $m = 3$ ([8], [9], [12]). Sie erhalten unter der Annahme $P_e^{(2,\max)} \approx 0$ und $P_e^{(1)} \approx 0$ ([8, S. 161ff], [9], [12, S. 228ff][1]):

$$
\text{(i)} \quad V_{\mathrm{SR}} \;\leq\; \frac{2-P_c^{(2)}}{P_c^{(2)}+(1-P_c^{(2)})\left(P_c^{(2)}+(1-P_c^{(2)})q_1\right)} \quad \text{und}
$$

$$
\text{(ii)} \quad V_{\mathrm{SR}} \;\leq\; \frac{P_c^{(2)}(1+2a_1)+a_2(2+a_1)+3a_3}{(1-a_1)^2}\,,
$$

[1] In [9] und [12] enthält die Definition von P_t einen Druckfehler, es muß heißen (s. [8]): $P_t = \underline{P_c} + (1-P_c)q_1$.

$$
\begin{aligned}
\text{mit} \quad a_1 &:= (1-P_c^{(2)})^2(1-P_c^{(2)}-(1-P_c^{(2)})q_2)\,, \\
a_2 &:= (1-P_c^{(2)})P_c^{(2)}\,, \\
a_3 &:= (1-P_c^{(2)})^2\,(P_c^{(2)}+(1-P_c^{(2)})q_2)\,, \\
q_1 &:= \mathrm{Prob}(G_1^c \mid A_0^d \cap B_1^d)\,, \\
q_2 &:= \mathrm{Prob}(G_2^c \mid A_0^d \cap B_1^d \cap B_2^d) \;=\; \frac{P_c^{(1)}}{P_d^{(2,0)}P_d^{(2,1)}P_d^{(2,2)}}\,.
\end{aligned}
$$

Unter obiger Annahme gilt $P_d^{(2,\mu)} \approx 1-P_c^{(2)}$ für $\mu = 0,1,2$, daher kann man durch eine einfache Rechnung zeigen, daß unser Ergebnis mit (ii) übereinstimmt. Das Resultat (i) beruht auf der Betrachtung einer anderen Vereinfachung des GH-ARQ-Systems, es kann nicht mit unserem verglichen werden.

6.3 Vergleich mit ARQ- und Typ-I Hybrid-ARQ-Systemen

Wir wollen unser GH-ARQ-System nun auch mit einem einfachen ARQ- und einem HI-ARQ-System vergleichen. Hierfür betrachten wir ein ARQ-System, das den fehlererkennenden Code $\mathcal{C}_2$ des GH-ARQ-Systems verwende. Der Code $\mathcal{C}_2$ habe die Decodierfehlerwahrscheinlichkeit $P_e^{(2)}$ (vgl. Satz 3.1(b)), ferner sei $P_d^{(2)} := 1-P_c^{(2)}-P_e^{(2)}$. Wir müssen beachten, daß im allgemeinen $P_e^{(2)} \neq P_e^{(2,i)}$ und $P_d^{(2)} \neq P_d^{(2,i)}$ für $i = 0,\ldots,m-1$ gilt.

Weiterhin betrachten wir ein HI-ARQ-System, das mit den Codes $\mathcal{C}_2$ und $\mathcal{C}_1$ des GH-ARQ-Systems gebildet wird.

In diesem Abschnitt nehmen wir an, daß als Decodierverfahren des fehlerkorrigierenden Codes $\mathcal{C}_1$ ein Maximum–Likelihood–Verfahren verwendet werde. Es sei $P_e' := \mathrm{Prob}(G_{m-1}^e)$, dies entspricht der Decodierfehlerwahrscheinlichkeit eines mit $\mathcal{C}_2$ und $\mathcal{C}_1$ gebildeten HI-ARQ-Systems. Wie in Abschnitt 4.3 sei

$$\Delta \;:=\; \min\{w(\bar{x}) \;:\; \bar{x} \in Q_{\bar{c}} \text{ für ein } \bar{c} \in \mathcal{C} \setminus \{\bar{0}\}\}\,,$$

wobei $\mathcal{C}$ der durch gewöhnliche Codeverkettung von $\mathcal{C}_2$ und $\mathcal{C}_1$ definierte Code sei. Im Fall $\Delta > d_2$ gilt nach Satz 4.10:

$$0 \;\le\; \lim_{\varepsilon\to 0}\frac{P_e^{(1)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} \;\le\; \lim_{\varepsilon\to 0}\frac{P_e'(\varepsilon)}{P_e^{(2)}(\varepsilon)} \;=\; 0\,.$$

Wir werden im folgenden annehmen, daß $\Delta > d_2$ gilt. Dies ist z. B. dann der Fall, wenn $\mathcal{C}_1$ und $\mathcal{C}_2$ die Kløve–Miller–Bedingung $d_1 > 2d_2$ erfüllen (s. Kor. 4.11).

6.3.1 Vergleich mit ARQ-Systemen

Zunächst vergleichen wir unser GH-ARQ- mit dem oben vorgestellten ARQ-System. Im Gegensatz zu dem in Abschnitt 5.3 durchgeführten Vergleich des HII-ARQ- mit dem ARQ-System,

besitzen die beiden hier betrachteten Übertragungssysteme unterschiedliche Decodierfehlerwahrscheinlichkeiten. Deren Verhältnis wird sich als entscheidend für die folgenden Ergebnisse erweisen. Es wird bestimmt durch die Minimalabstände der beteiligten Codes.

Lemma 6.4 *Für $i = 0, \ldots, m-1$ sei $\tilde{d}_i$ der Minimalabstand des von $\mathbf{G}_2\tilde{\mathbf{G}}_i$ erzeugten Codes. Ferner seien $\tilde{d}_{\min} := \min\{\tilde{d}_i : i = 0, \ldots, m-1\}$ und $\tilde{d}_{\max} := \max\{\tilde{d}_i : i = 0, \ldots, m-1\}$, dann gilt:*

$$
\begin{array}{lllll}
a) & (i) & d_2 > \tilde{d}_{\min} & \textit{impliziert} & \lim\limits_{\varepsilon\to 0} \dfrac{P_e^{(2,\max)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} = \infty \,, \\
 & (ii) & d_2 = \tilde{d}_{\min} & \textit{impliziert} & 0 < \lim\limits_{\varepsilon\to 0} \dfrac{P_e^{(2,\max)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} < \infty \,, \\
 & (iii) & d_2 < \tilde{d}_{\min} & \textit{impliziert} & \lim\limits_{\varepsilon\to 0} \dfrac{P_e^{(2,\max)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} = 0 \,, \\
b) & (i) & d_2 > \tilde{d}_{\max} & \textit{impliziert} & \lim\limits_{\varepsilon\to 0} \dfrac{P_e^{(2,\min)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} = \infty \,, \\
 & (ii) & d_2 = \tilde{d}_{\max} & \textit{impliziert} & 0 < \lim\limits_{\varepsilon\to 0} \dfrac{P_e^{(2,\min)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} < \infty \,, \\
 & (iii) & d_2 < \tilde{d}_{\max} & \textit{impliziert} & \lim\limits_{\varepsilon\to 0} \dfrac{P_e^{(2,\min)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} = 0 \,.
\end{array}
$$

Beweis: Für $i = 0, \ldots, m-1$ sei $\{\tilde{A}_0^{(i)}, \ldots, \tilde{A}_n^{(i)}\}$ die Gewichtsverteilung des von $\mathbf{G}_2\tilde{\mathbf{G}}_i$ definierten Codes. Nach Satz 3.1(b) gilt dann für $i, j \in \{0, \ldots, m-1\}$:

$$
\begin{aligned}
\lim_{\varepsilon\to 0} \frac{P_e^{(2,i)}(\varepsilon)}{P_e^{(2,j)}(\varepsilon)} &= \lim_{\varepsilon\to 0} \frac{\sum\limits_{\nu=\tilde{d}_i}^{n} \tilde{A}_\nu^{(i)}\, \varepsilon^\nu\, (1-\varepsilon)^{n-\nu}}{\sum\limits_{\nu=\tilde{d}_j}^{n} \tilde{A}_\nu^{(j)}\, \varepsilon^\nu\, (1-\varepsilon)^{n-\nu}} \\
&= \lim_{\varepsilon\to 0} \left(\varepsilon^{\tilde{d}_i - \tilde{d}_j}\, \frac{\sum\limits_{\nu=0}^{n-\tilde{d}_i} \tilde{A}_{\nu+\tilde{d}_i}^{(i)}\, \varepsilon^\nu\, (1-\varepsilon)^{n-\nu-\tilde{d}_i}}{\sum\limits_{\nu=0}^{n-\tilde{d}_j} \tilde{A}_{\nu+\tilde{d}_j}^{(j)}\, \varepsilon^\nu\, (1-\varepsilon)^{n-\nu-\tilde{d}_j}} \right) \\
&= \begin{cases} 0 & \text{falls} \quad \tilde{d}_i > \tilde{d}_j \,, \\ \tilde{A}_{\tilde{d}_i}^{(i)} / \tilde{A}_{\tilde{d}_j}^{(j)} & \text{falls} \quad \tilde{d}_i = \tilde{d}_j \,, \\ \infty & \text{falls} \quad \tilde{d}_i < \tilde{d}_j \,. \end{cases}
\end{aligned}
$$

Also ist $P_e^{(2,i)}(\varepsilon) > P_e^{(2,j)}(\varepsilon)$ für hinreichend kleine ε, falls $\tilde{d}_i < \tilde{d}_j$ ist. Deshalb sind $P_e^{(2,\max)}(\varepsilon)$ und $P_e^{(2,\min)}(\varepsilon)$ für kleine ε die Decodierfehlerwahrscheinlichkeiten eines von $\mathbf{G}_2\tilde{\mathbf{G}}_l$ erzeugten

Codes mit Minimalabstand $\tilde{d}_l = \tilde{d}_{\min}$ bzw. $\tilde{d}_l = \tilde{d}_{\max}$. Mit dem gleichen Argument wie oben erhalten wir die Behauptung. q.e.d.

Wir können nun die Restfehlerwahrscheinlichkeiten der betrachteten Systeme vergleichen.

Satz 6.5 *Falls $\Delta > d_2$ gilt, folgt für die Restfehlerwahrscheinlichkeiten P_R^{ARQ} und P_R^{GH}:*

$$\lim_{\varepsilon\to 0} \frac{P_e^{(2,\min)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} \;\le\; \lim_{\varepsilon\to 0} \frac{P_R^{\mathrm{GH}}(\varepsilon)}{P_R^{\mathrm{ARQ}}(\varepsilon)} \;\le\; \lim_{\varepsilon\to 0} \frac{P_e^{(2,\max)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} \,.$$

Beweis: Nach Satz 3.4 und Korollar 6.2 gilt:

$$\begin{aligned}
\frac{P_R^{\mathrm{GH}}}{P_R^{\mathrm{ARQ}}} &\le \frac{P_e^{(2,\max)}\,\frac{1-{P_d^{(2,\max)}}^m}{1-P_d^{(2,\max)}} + P_e^{(1)}}{1-{P_d^{(2,\max)}}^m + P_c^{(1)} + P_e^{(1)}} \cdot \frac{1-P_d^{(2)}}{P_e^{(2)}} \\
&\le \frac{P_e^{(2,\max)}\,\frac{1-{P_d^{(2,\max)}}^m}{1-P_d^{(2,\max)}} + P_e^{(1)}}{1-{P_d^{(2,\max)}}^m} \cdot \frac{1-P_d^{(2)}}{P_e^{(2)}} \\
&= \frac{P_e^{(2,\max)}}{P_e^{(2)}} \cdot \frac{1-P_d^{(2)}}{1-P_d^{(2,\max)}} + \frac{P_e^{(1)}}{P_e^{(2)}} \cdot \frac{1-P_d^{(2)}}{1-{P_d^{(2,\max)}}^m} \,.
\end{aligned}$$

Aus $\Delta > d_2$ folgt $\lim\limits_{\varepsilon\to 0} \frac{P_e^{(1)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} = 0$.

Ferner gilt nach Kor. 3.2: $\lim\limits_{\varepsilon\to 0} P_d^{(2)}(\varepsilon) = \lim\limits_{\varepsilon\to 0} P_d^{(2,\max)}(\varepsilon) = 0$.

So erhalten wir

$$\lim_{\varepsilon\to 0} \frac{P_R^{\mathrm{GH}}(\varepsilon)}{P_R^{\mathrm{ARQ}}(\varepsilon)} \;\le\; \lim_{\varepsilon\to 0} \frac{P_e^{(2,\max)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} \,.$$

Weiterhin folgt aus (6.1): $P_R^{\mathrm{GH}} \ge \mathrm{Prob}(A_0^e) \ge P_e^{(2,\min)}$. Dies impliziert:

$$\lim_{\varepsilon\to 0} \frac{P_R^{\mathrm{GH}}(\varepsilon)}{P_R^{\mathrm{ARQ}}(\varepsilon)} \;\ge\; \lim_{\varepsilon\to 0} \frac{P_e^{(2,\min)}(\varepsilon)(1-P_d^{(2)}(\varepsilon))}{P_e^{(2)}(\varepsilon)} \;=\; \lim_{\varepsilon\to 0} \frac{P_e^{(2,\min)}(\varepsilon)}{P_e^{(2)}(\varepsilon)} \,.$$

q.e.d.

Bemerkungen:

a) Falls $P_e^{(2,i)} = P_e^{(2)}$ für alle $i = 0, \ldots, m-1$ gilt, folgt

$$\lim_{\varepsilon\to 0} \frac{P_R^{\mathrm{GH}}(\varepsilon)}{P_R^{\mathrm{ARQ}}(\varepsilon)} \;=\; 1 \,.$$

Dieses Ergebnis erhalten auch Morgera und Krishna (s. [12, S. 200], vgl. auch Bem. (a) zu Satz 6.1).

b) Der Satz zeigt, daß bei der Wahl der Matrizen $\tilde{\mathbf{G}}_0, \ldots, \tilde{\mathbf{G}}_{m-1}$ die Minimalabstände der verketteten Codes beachtet werden müssen. Gilt nämlich $\tilde{d}_i < d_2$ für alle $i = 0, \ldots, m-1$, so folgt aus Lemma 6.4(b)(i):

$$\lim_{\varepsilon \to 0} \frac{P_R^{\mathrm{GH}}(\varepsilon)}{P_R^{\mathrm{ARQ}}(\varepsilon)} \;=\; \infty \,.$$

Der zusätzliche Aufwand für das GH-ARQ-System bewirkt dann also eine drastische Verschlechterung der Restfehlerwahrscheinlichkeit im Vergleich zu einem einfachen ARQ-System. Umgekehrt können wir aber auch eine spürbare Verbesserung der Restfehlerwahrscheinlichkeit erreichen, wenn $d_2 < \tilde{d}_i$ für alle $i = 0, \ldots, m-1$ gilt. In diesem Fall folgt nämlich aus Lemma 6.4(a)(iii):

$$\lim_{\varepsilon \to 0} \frac{P_R^{\mathrm{GH}}(\varepsilon)}{P_R^{\mathrm{ARQ}}(\varepsilon)} \;=\; 0 \,.$$

Diese Beobachtungen sind nicht überraschend, analoges gilt für ARQ-Systeme, die Codes gleicher Länge und Dimension verwenden (vgl. Beweis zu Lemma 6.4).

Wir wollen nun noch untersuchen, ob das GH-ARQ-System eine Verbesserung des Durchsatzes gegenüber dem betrachteten ARQ-System bewirkt.

Satz 6.6 *Für die Durchsatzraten η^{ARQ} und η^{GH} gilt*

a) bei Verwendung des Selective-Repeat-Protokolls:

$$\eta_{\mathrm{SR}}^{\mathrm{GH}} \;\geq\; \frac{\left(1 - P_d^{(2,\max)}\right)^2}{\left(1 - P_d^{(2,\min)}\right)\left(1 - P_d^{(2)}\right)}\, \eta_{\mathrm{SR}}^{\mathrm{ARQ}} \,,$$

b) bei Verwendung des Go-Back-Protokolls:

$$\eta_{\mathrm{GB}N}^{\mathrm{GH}} \;\geq\; \frac{\left(1 - P_d^{(2,\max)}\right)^2}{\left(1 - P_d^{(2,\min)}\right)\left(1 - P_d^{(2)}\right)}\, \eta_{\mathrm{GB}M}^{\mathrm{ARQ}} \,,$$

$$\textit{falls} \quad N \;\leq\; M\, \frac{P_d^{(2)}}{P_d^{(2,\max)}} \cdot \frac{1 - \left({P_d^{(2,\max)}}^m - P_c^{(1)} - P_e^{(1)}\right)}{1 - {P_d^{(2,\max)}}^m - \left(1 - P_d^{(2,\max)}\right)\frac{P_c^{(1)} + P_e^{(1)}}{P_d^{(2,\max)}}}$$

$$+ \left(1 - \frac{P_d^{(2)}}{P_d^{(2,\max)}}\right) \cdot \frac{1 - \left({P_d^{(2,\max)}}^m - P_c^{(1)} - P_e^{(1)}\right)}{1 - {P_d^{(2,\max)}}^m - \left(1 - P_d^{(2,\max)}\right)\frac{P_c^{(1)} + P_e^{(1)}}{P_d^{(2,\max)}}} \qquad \textit{ist.}$$

Beweis: Zu (a): Nach (3.1) und Satz 6.3(a) gilt:

$$\begin{aligned}
\frac{\eta_{\mathrm{SR}}^{\mathrm{GH}}}{\eta_{\mathrm{SR}}^{\mathrm{ARQ}}} &\geq \frac{1-P_d^{(2,\max)}}{1-P_d^{(2,\min)}} \cdot \frac{1-{P_d^{(2,\max)}}^m + P_c^{(1)} + P_e^{(1)}}{\frac{1-{P_d^{(2,\max)}}^m}{1-P_d^{(2,\max)}}} \cdot \frac{1}{1-P_d^{(2)}} \\
&= \frac{\left(1-P_d^{(2,\max)}\right)^2}{\left(1-P_d^{(2,\min)}\right)\left(1-P_d^{(2)}\right)} \left(1 + \frac{P_c^{(1)} + P_e^{(1)}}{1-{P_d^{(2,\max)}}^m}\right) \\
&\geq \frac{\left(1-P_d^{(2,\max)}\right)^2}{\left(1-P_d^{(2,\min)}\right)\left(1-P_d^{(2)}\right)} .
\end{aligned}$$

Zu (b): Nach (3.2) und Satz 6.3(b) gilt:

$$\begin{aligned}
\eta_{\mathrm{GB}N}^{\mathrm{GH}} &\geq \frac{1-P_d^{(2,\max)}}{1-P_d^{(2,\min)}} \cdot \frac{1-{P_d^{(2,\max)}}^m + P_c^{(1)} + P_e^{(1)}}{\frac{1-{P_d^{(2,\max)}}^m}{1-P_d^{(2,\max)}}\left[1+(N-1)P_d^{(2,\max)}\right] - (N-1)(P_c^{(1)}+P_e^{(1)})} \cdot \frac{k}{n} , \\
\eta_{\mathrm{GB}M}^{\mathrm{ARQ}} &= \frac{1-P_d^{(2)}}{1+(M-1)P_d^{(2)}} \cdot \frac{k}{n} .
\end{aligned}$$

Mit elementaren Umformungen erhalten wir:

$$\begin{aligned}
&\frac{1-P_d^{(2,\max)}}{1-P_d^{(2,\min)}} \cdot \frac{1-{P_d^{(2,\max)}}^m + P_c^{(1)} + P_e^{(1)}}{\frac{1-{P_d^{(2,\max)}}^m}{1-P_d^{(2,\max)}}\left[1+(N-1)P_d^{(2,\max)}\right] - (N-1)(P_c^{(1)}+P_e^{(1)})} \cdot \frac{1+(M-1)P_d^{(2)}}{1-P_d^{(2)}} \\
&\qquad\qquad \geq \frac{\left(1-P_d^{(2,\max)}\right)^2}{\left(1-P_d^{(2,\min)}\right)\left(1-P_d^{(2)}\right)} \\
\iff \quad N &\leq M \frac{P_d^{(2)}}{P_d^{(2,\max)}} \cdot \frac{1-{P_d^{(2,\max)}}^m + P_c^{(1)} + P_e^{(1)}}{1-{P_d^{(2,\max)}}^m - (1-P_d^{(2,\max)})\frac{P_c^{(1)}+P_e^{(1)}}{P_d^{(2,\max)}}} \\
&\quad + \frac{P_d^{(2,\max)} - P_d^{(2)}}{P_d^{(2,\max)}} \cdot \frac{1-{P_d^{(2,\max)}}^m + P_c^{(1)} + P_e^{(1)}}{1-{P_d^{(2,\max)}}^m - (1-P_d^{(2,\max)})\frac{P_c^{(1)}+P_e^{(1)}}{P_d^{(2,\max)}}} \\
&\quad =: M\sigma + \tau .
\end{aligned}$$

Also impliziert $N \leq M\sigma + \tau$ auch

$$\eta_{\mathrm{GB}N}^{\mathrm{GH}} \geq \frac{\left(1-P_d^{(2,\max)}\right)^2}{\left(1-P_d^{(2,\min)}\right)\left(1-P_d^{(2)}\right)} \eta_{\mathrm{GB}M}^{\mathrm{ARQ}} .$$

q.e.d.

Bemerkungen:

a) Der in beiden Aussagen auftretende Term

$$\theta := \frac{\left(1-P_d^{(2,\max)}\right)^2}{\left(1-P_d^{(2,\min)}\right)\left(1-P_d^{(2)}\right)} = \frac{\left(P_c^{(2)}+P_e^{(2,\min)}\right)^2}{\left(P_c^{(2)}+P_e^{(2,\max)}\right)\left(P_c^{(2)}+P_e^{(2)}\right)}$$

zeigt wieder die Bedeutung der unterschiedlichen Decodierfehlerwahrscheinlichkeiten. Falls $P_e^{(2,i)} = P_e^{(2)}$ für alle $i = 0, \ldots, m-1$ gilt, folgt $\theta = 1$.
Es gilt aber stets $\lim\limits_{\varepsilon \to 0} \theta(\varepsilon) = 1$.

b) Der Beweis der Aussage (a) zeigt, daß bei Verwendung des Selective-Repeat-Protokolls

$$\eta_{\mathrm{SR}}^{\mathrm{GH}} \geq \frac{\left(1-P_d^{(2,\max)}\right)^2}{\left(1-P_d^{(2,\min)}\right)\left(1-P_d^{(2)}\right)} \left(1+\frac{P_c^{(1)}+P_e^{(1)}}{1-P_d^{(2,\max)\,m}}\right) \eta_{\mathrm{SR}}^{\mathrm{ARQ}}$$

ist. Es gilt jedoch

$$\lim_{\varepsilon \to 0} \left\{ \frac{\left(1-P_d^{(2,\max)}(\varepsilon)\right)^2}{\left(1-P_d^{(2,\min)}(\varepsilon)\right)\left(1-P_d^{(2)}(\varepsilon)\right)} \left(1+\frac{P_c^{(1)}(\varepsilon)+P_e^{(1)}(\varepsilon)}{1-\left(P_d^{(2,\max)}(\varepsilon)\right)^m}\right) \right\} = 1 .$$

Dies bedeutet, daß das Selective-Repeat GH-ARQ-System bei kleinen Bitfehlerraten nur eine geringe Verbesserung des Durchsatzes gegenüber dem einfachen Selective-Repeat ARQ-System bewirkt.

c) Die Bedingung $N \leq M\sigma + \tau$ aus Aussage (b) (vgl. Beweis zu (b)) zeigt wieder die Bedeutung der unterschiedlichen Decodierfehlerwahrscheinlichkeiten.
Zunächst folgt aus (6.7):

$$0 \leq \lim_{\varepsilon \to 0} \frac{P_c^{(1)}(\varepsilon)+P_e^{(1)}(\varepsilon)}{P_d^{(2,\max)}(\varepsilon)} \leq \lim_{\varepsilon \to 0} \frac{\left(P_d^{(2,\max)}(\varepsilon)\right)^m}{P_d^{(2,\max)}(\varepsilon)} = 0 .$$

So erhalten wir:

$$\begin{aligned} \lim_{\varepsilon \to 0} \sigma(\varepsilon) &= \lim_{\varepsilon \to 0} \left\{ \frac{P_d^{(2)}(\varepsilon)}{P_d^{(2,\max)}(\varepsilon)} \cdot \frac{1-\left(P_d^{(2,\max)}(\varepsilon)\right)^m + P_c^{(1)}(\varepsilon) + P_e^{(1)}(\varepsilon)}{1-\left(P_d^{(2,\max)}(\varepsilon)\right)^m - \left(1-P_d^{(2,\max)}(\varepsilon)\right)\frac{P_c^{(1)}(\varepsilon)+P_e^{(1)}(\varepsilon)}{P_d^{(2,\max)}(\varepsilon)}} \right\} \\ &= \lim_{\varepsilon \to 0} \frac{P_d^{(2)}(\varepsilon)}{P_d^{(2,\max)}(\varepsilon)} \end{aligned}$$

und

$$\lim_{\varepsilon\to 0}\tau(\varepsilon) \;=\; \lim_{\varepsilon\to 0}\left\{\left(1-\frac{P_d^{(2)}(\varepsilon)}{P_d^{(2,\max)}(\varepsilon)}\right)\cdot\frac{1-\left(P_d^{(2,\max)}(\varepsilon)\right)^m+P_c^{(1)}(\varepsilon)+P_e^{(1)}(\varepsilon)}{1-\left(P_d^{(2,\max)}(\varepsilon)\right)^m-\left(1-P_d^{(2,\max)}(\varepsilon)\right)\frac{P_c^{(1)}(\varepsilon)+P_e^{(1)}(\varepsilon)}{P_d^{(2,\max)}(\varepsilon)}}\right\}$$

$$= \;1-\lim_{\varepsilon\to 0}\frac{P_d^{(2)}(\varepsilon)}{P_d^{(2,\max)}(\varepsilon)}\,.$$

Falls $P_e^{(2,i)}=P_e^{(2)}$ für alle $i=0,\ldots,m-1$ gilt, folgt $\lim\limits_{\varepsilon\to 0}\sigma(\varepsilon)=1$ und $\lim\limits_{\varepsilon\to 0}\tau(\varepsilon)=0$.
In diesem Fall gilt sogar

$$\sigma \;=\; \frac{1-{P_d^{(2,\max)}}^m+P_c^{(1)}+P_e^{(1)}}{1-{P_d^{(2,\max)}}^m-(1-P_d^{(2,\max)})\frac{P_c^{(1)}+P_e^{(1)}}{P_d^{(2,\max)}}} \;\geq\; 1 \qquad \text{und} \qquad \tau \;=\; 0\,.$$

Das GH-ARQ-System bietet hier also nur bei großen Bitfehlerwahrscheinlichkeiten oder sehr geringer Differenz der Rückgrifftiefen Vorteile gegenüber dem einfachen ARQ-System.

6.3.2 Vergleich mit Typ-I Hybrid-ARQ-Systemen

Wir wollen unser GH-ARQ-System nun auch noch mit dem HI-ARQ-System vergleichen. Mit den Bezeichnungen aus den Abschnitten 5.2.1 und 6.2.1 seien

$$\begin{aligned} P_c' &:= \text{Prob}(G_{m-1}^c)\,,\\ P_e' &:= \text{Prob}(G_{m-1}^e)\,,\\ P_d' &:= \text{Prob}(G_{m-1}^d)\,. \end{aligned}$$

Dies entspricht den durch Satz 4.1 gegebenen Wahrscheinlichkeiten. Wir haben bereits gesehen, daß im Fall $\Delta > d_2$ für die Decodierfehlerwahrscheinlichkeiten P_e' und $P_e^{(2)}$

$$\lim_{\varepsilon\to 0}\frac{P_e'(\varepsilon)}{P_e^{(2)}(\varepsilon)} \;=\; 0$$

gilt.

Aus Satz 6.5 folgt, daß das GH-ARQ-System unter gewissen Voraussetzungen die gleiche Fehlerordnung wie das einfache ARQ-System erreicht. Unter diesen Voraussetzungen besitzt das HI-ARQ-System eine bessere Fehlerordnung.

Satz 6.7 *Falls $\Delta > d_2$ gilt, folgt für die Restfehlerwahrscheinlichkeiten P_R^{HI} und P_R^{GH}:*

$$\lim_{\varepsilon\to 0}\frac{P_R^{\mathrm{HI}}(\varepsilon)}{P_R^{\mathrm{GH}}(\varepsilon)} \;\leq\; \lim_{\varepsilon\to 0}\left(\frac{P_e'(\varepsilon)}{P_e^{(2)}(\varepsilon)}\cdot\frac{P_e^{(2)}(\varepsilon)}{P_e^{(2,\min)}(\varepsilon)}\right)\,.$$

Ferner seien $\tilde{d}_i$ *der Minimalabstand des von* $\mathbf{G}_2\tilde{\mathbf{G}}_i$ *erzeugten Codes,* $i = 0, \ldots, m-1$, *und* $\tilde{d}_{\max} := \max\{\tilde{d}_i \,:\, i = 0, \ldots, m-1\}$. *Gilt zusätzlich* $\tilde{d}_{\max} \leq d_2$, *so folgt*

$$\lim_{\varepsilon \to 0} \frac{P_R^{\mathrm{HI}}(\varepsilon)}{P_R^{\mathrm{GH}}(\varepsilon)} \;=\; 0\,.$$

Beweis: Aus (6.1) folgt $P_R^{\mathrm{GH}} \geq \mathrm{Prob}(A_0^e) \geq P_e^{(2,\min)}$.
Mit (4.3) erhalten wir:

$$\begin{aligned}
\lim_{\varepsilon \to 0} \frac{P_R^{\mathrm{HI}}(\varepsilon)}{P_R^{\mathrm{GH}}(\varepsilon)} \;&\leq\; \lim_{\varepsilon \to 0} \left(\frac{P_e'(\varepsilon)}{1 - P_d'(\varepsilon)} \cdot \frac{1}{P_e^{(2,\min)}(\varepsilon)} \right) \\
&=\; \lim_{\varepsilon \to 0} \frac{P_e'(\varepsilon)}{P_e^{(2,\min)}(\varepsilon)} \\
&=\; \lim_{\varepsilon \to 0} \left(\frac{P_e'(\varepsilon)}{P_e^{(2)}(\varepsilon)} \cdot \frac{P_e^{(2)}(\varepsilon)}{P_e^{(2,\min)}(\varepsilon)} \right)\,.
\end{aligned}$$

Wegen $\Delta > d_2$ gilt $\lim\limits_{\varepsilon \to 0} \dfrac{P_e'(\varepsilon)}{P_e^{(2)}(\varepsilon)} \;=\; 0\,.$
Ferner ergibt sich im Fall $\tilde{d}_{\max} \leq d_2$ aus Lemma 6.4(b):

$$0 \;\leq\; \lim_{\varepsilon \to 0} \frac{P_e^{(2)}(\varepsilon)}{P_e^{(2,\min)}(\varepsilon)} \;<\; \infty\,,$$

so daß wir in diesem Fall

$$\lim_{\varepsilon \to 0} \frac{P_R^{\mathrm{HI}}(\varepsilon)}{P_R^{\mathrm{GH}}(\varepsilon)} \;=\; 0$$

erhalten. q.e.d.

Bemerkung: Wenn $\tilde{d}_{\max} > d_2$ ist, gilt nach Lemma 6.4(b)(iii):

$$\lim_{\varepsilon \to 0} \frac{P_e^{(2)}(\varepsilon)}{P_e^{(2,\min)}(\varepsilon)} \;=\; \infty\,.$$

In diesem Fall können wir also keine Aussage über $\lim\limits_{\varepsilon \to 0} \frac{P_R^{\mathrm{HI}}(\varepsilon)}{P_R^{\mathrm{GH}}(\varepsilon)}$ machen.

Wie schon bei der Analyse des HII-ARQ-Systems können wir auch hier feststellen, daß der schlechteren Restfehlerwahrscheinlichkeit ein höherer Durchsatz gegenübersteht. Beim Vergleich der Go-Back-Systeme müssen wir wieder beachten, daß sich die Rückgrifftiefe des GH-ARQ-Systems auf Blöcke der Länge n bezieht, die des ARQ-Systems jedoch auf Blöcke der Länge mn.

Satz 6.8 *Für die Durchsatzraten η^{HI} und η^{GH} gilt*

a) bei Verwendung des Selective-Repeat-Protokolls:

$$\eta_{\mathrm{SR}}^{\mathrm{GH}} \;\geq\; m\,\frac{\left(1-P_d^{(2,\max)}\right)^2}{\left(1-P_d^{(2,\min)}\right)\left(1-P_d'\right)}\,\eta_{\mathrm{SR}}^{\mathrm{HI}}\,,$$

b) bei Verwendung des Go-Back-Protokolls:

$$\lim_{\varepsilon\to 0}\frac{\eta_{\mathrm{GB}N}^{\mathrm{GH}}(\varepsilon)}{\eta_{\mathrm{GB}N'}^{\mathrm{HI}}(\varepsilon)} \;\geq\; m\,.$$

Beweis: Zu (a): Nach (4.10) und Satz 6.3(a) gilt:

$$\begin{aligned}
\frac{\eta_{\mathrm{SR}}^{\mathrm{GH}}}{\eta_{\mathrm{SR}}^{\mathrm{HI}}} &\geq \frac{1-P_d^{(2,\max)}}{1-P_d^{(2,\min)}}\cdot\frac{1-P_d^{(2,\max)^m}+P_c^{(1)}+P_e^{(1)}}{\frac{1-P_d^{(2,\max)^m}}{1-P_d^{(2,\max)}}}\cdot\frac{k}{n}\cdot\frac{1}{1-P_d'}\cdot\frac{mn}{k}\\
&= m\,\frac{\left(1-P_d^{(2,\max)}\right)^2}{\left(1-P_d^{(2,\min)}\right)\left(1-P_d'\right)}\left(1+\frac{P_c^{(1)}+P_e^{(1)}}{1-P_d^{(2,\max)^m}}\right)\\
&\geq m\,\frac{\left(1-P_d^{(2,\max)}\right)^2}{\left(1-P_d^{(2,\min)}\right)\left(1-P_d'\right)}\,.
\end{aligned}$$

Zu (b): Nach (4.11) und Satz 6.3(b) gilt mit $\rho^{(1)} := P_c^{(1)} + P_e^{(1)}$:

$$\begin{aligned}
&\frac{\eta_{\mathrm{GB}N}^{\mathrm{GH}}(\varepsilon)}{\eta_{\mathrm{GB}N'}^{\mathrm{HI}}(\varepsilon)}\\
&\geq \frac{1-P_d^{(2,\max)}(\varepsilon)}{1-P_d^{(2,\min)}(\varepsilon)}\cdot\frac{1-\left(P_d^{(2,\max)}(\varepsilon)\right)^m+\rho^{(1)}(\varepsilon)}{\frac{1-\left(P_d^{(2,\max)}(\varepsilon)\right)^m}{1-P_d^{(2,\max)}(\varepsilon)}\left[1+(N-1)P_d^{(2,\max)}(\varepsilon)\right]-(N-1)\rho^{(1)}(\varepsilon)}\cdot\frac{k}{n}\cdot\\
&\quad\cdot\frac{1+(N'-1)P_d'(\varepsilon)}{1-P_d'(\varepsilon)}\cdot\frac{mn}{k}\\
&\geq m\,\frac{\left(1-P_d^{(2,\max)}(\varepsilon)\right)^2}{\left(1-P_d^{(2,\min)}(\varepsilon)\right)\left(1-P_d'(\varepsilon)\right)}\cdot\frac{1-\left(P_d^{(2,\max)}(\varepsilon)\right)^m+\rho^{(1)}(\varepsilon)}{1-\left(P_d^{(2,\max)}(\varepsilon)\right)^m}\cdot\frac{1+(N'-1)P_d'(\varepsilon)}{1+(N-1)P_d^{(2,\max)}(\varepsilon)}\\
&\overset{\varepsilon\to 0}{\longrightarrow} m\,.
\end{aligned}$$

q.e.d.

Bemerkungen:

a) Der Beweis zeigt, daß die Verbesserung des Durchsatzes um den Faktor m durch die geringe Informationsrate des inneren Codes $\mathcal{C}_1$ verursacht wird, diese beträgt $\frac{1}{m}$.

b) Falls $P_d^{(2,\max)} \geq P_d^{(2)}$ ist, gilt nach Satz 4.14 (unter der üblichen Voraussetzung $\Delta > d_2$) $P'_d(\varepsilon) < P_d^{(2,\max)}(\varepsilon)$ für hinreichend kleine ε. Hieraus folgt für diese ε:

$$\frac{\left(1 - P_d^{(2,\max)}(\varepsilon)\right)^2}{\left(1 - P_d^{(2,\min)}(\varepsilon)\right)\left(1 - P'_d(\varepsilon)\right)} \leq 1 .$$

Es gilt aber stets

$$\lim_{\varepsilon \to 0} \frac{\left(1 - P_d^{(2,\max)}(\varepsilon)\right)^2}{\left(1 - P_d^{(2,\min)}(\varepsilon)\right)\left(1 - P'_d(\varepsilon)\right)} = 1$$

und deshalb

$$\lim_{\varepsilon \to 0} \frac{\eta_{\mathrm{SR}}^{\mathrm{GH}}(\varepsilon)}{\eta_{\mathrm{SR}}^{\mathrm{HI}}(\varepsilon)} \geq m .$$

c) Beim Vergleich der Go-Back-Systeme können wir leider kein interessanteres Ergebnis erhalten, dies war auch schon bei der Analyse des HII-ARQ-Systems nicht möglich (s. Bem. (c) zu Satz 5.8).

6.3.3 Resümee

Das GH-ARQ-System ist eine Verallgemeinerung des HII-ARQ-Systems, der Vergleich mit den anderen Übertragungssystemen führt im wesentlichen zu den gleichen Ergebnissen wie in Abschnitt 5.3.

Entscheidend ist hier jedoch die Wahl der Matrizen $\tilde{\mathbf{G}}_0, \ldots, \tilde{\mathbf{G}}_{m-1}$, weil diese die Decodierfehlerwahrscheinlichkeit der Fehlererkennung beeinflussen. Sie sollten so gewählt werden, daß der Minimalabstand des äußeren Codes $\mathcal{C}_2$ nicht größer als die Minimalabstände der von $\mathbf{G}_2\tilde{\mathbf{G}}_i$ erzeugten verketteten Codes ist, $i = 0, \ldots, m-1$. Außerdem ist wieder die Einhaltung der Kløve–Miller–Bedingung zu beachten.

Wie auch das HII-ARQ- scheint das GH-ARQ-System besonders geeignet zu sein, um bei variierendem Störverhalten des Übertragungskanals hohe Durchsatzraten zu erreichen.

6.4 Vergleich zweier GH-ARQ-Systeme verschiedener Tiefe

In diesem Abschnitt wollen wir zwei GH-ARQ-Systeme S_1 und S_2 der Tiefe $m_1 := m$ bzw. $m_2 := m+1$ vergleichen. Das System S_1 kann auch ein HII-ARQ-System sein, dann wäre $m = 2$.

Wir nehmen an, daß beide Übertragungssysteme den gleichen äußeren Code $\mathcal{C}_2$ verwenden. Ferner sei $\mathcal{C}_1$ der innere Code von S_2, dieser habe die Tiefe $m_2 = m+1$. Der innere Code von S_1 sei der Teilcode $\mathcal{C}_1^{(m-1)}$ der Tiefe $m_1 = m$ von $\mathcal{C}_1$. Es seien $\mathbf{G}_2$ die Generatormatrix von $\mathcal{C}_2$ und

$$\mathbf{G}_1 \;=\; \left[\tilde{\mathbf{G}}_0 \mid \ldots \mid \tilde{\mathbf{G}}_{m-1} \mid \tilde{\mathbf{G}}_m\right]$$

die von $\mathcal{C}_1$. Die Generatormatrix von $\mathcal{C}_1^{(m-1)}$ ist dann

$$\mathbf{G}_1^{(m-1)} \;=\; \left[\tilde{\mathbf{G}}_0 \mid \ldots \mid \tilde{\mathbf{G}}_{m-1}\right] .$$

Ferner seien $\tilde{d}_i$ der Minimalabstand des von $\mathbf{G}_2\tilde{\mathbf{G}}_i$ erzeugten Codes, $i = 0, \ldots, m$, und

$$\begin{aligned}
\tilde{d}_{\min}^{(1)} &:= \min\{\tilde{d}_i \,:\, i = 0, \ldots, m-1\}, \\
\tilde{d}_{\max}^{(1)} &:= \max\{\tilde{d}_i \,:\, i = 0, \ldots, m-1\}, \\
\tilde{d}_{\min}^{(2)} &:= \min\{\tilde{d}_i \,:\, i = 0, \ldots, m\}, \\
\tilde{d}_{\max}^{(2)} &:= \max\{\tilde{d}_i \,:\, i = 0, \ldots, m\}.
\end{aligned}$$

Aus dem Beweis des Lemmas 6.4 folgt, daß S_1 und S_2 die gleiche Fehlerordnung erreichen, falls $\tilde{d}_{\min}^{(1)} \le \tilde{d}_m$ gilt. Anderenfalls bewirkt die Decodierfehlerwahrscheinlichkeit des von $\mathbf{G}_2\tilde{\mathbf{G}}_m$ erzeugten Codes eine Verschlechterung der Restfehlerwahrscheinlichkeit von S_2 gegenüber S_1.

Wir nehmen im folgenden an, daß $\tilde{d}_{\min}^{(1)} \le \tilde{d}_m \le \tilde{d}_{\max}^{(1)}$ gelte, dies impliziert $\tilde{d}_{\min}^{(1)} = \tilde{d}_{\min}^{(2)}$ und $\tilde{d}_{\max}^{(1)} = \tilde{d}_{\max}^{(2)}$. Ferner sei d_2 der Minimalabstand des Codes $\mathcal{C}_2$, und es gelte $\tilde{d}_{\min}^{(2)} \le d_2 \le \tilde{d}_{\max}^{(2)}$. Wir definieren:

$$\begin{aligned}
P_{e,1}^{(2,\min)} &:= \min\left\{P_e^{(2,i)} \,:\, i = 0, \ldots, m-1\right\}, \\
P_{e,1}^{(2,\max)} &:= \max\left\{P_e^{(2,i)} \,:\, i = 0, \ldots, m-1\right\}, \\
P_{e,2}^{(2,\min)} &:= \min\left\{P_e^{(2,i)} \,:\, i = 0, \ldots, m\right\}, \\
P_{e,2}^{(2,\max)} &:= \max\left\{P_e^{(2,i)} \,:\, i = 0, \ldots, m\right\}.
\end{aligned}$$

Aus Lemma 6.4 und Satz 6.5 folgt für die Restfehlerwahrscheinlichkeiten $P_R^{S_1}$ und $P_R^{S_2}$:

$$\lim_{\varepsilon\to 0} \frac{P_R^{S_1}(\varepsilon)}{P_R^{S_2}(\varepsilon)} \;=\; \lim_{\varepsilon\to 0} \left(\frac{P_R^{S_1}(\varepsilon)}{P_R^{\mathrm{ARQ}}(\varepsilon)} \cdot \frac{P_R^{\mathrm{ARQ}}(\varepsilon)}{P_R^{S_2}(\varepsilon)}\right) \;\le\; \left(\lim_{\varepsilon\to 0} \frac{P_{e,1}^{(2,\max)}(\varepsilon)}{P_e^{(2)}(\varepsilon)}\right) \left(\lim_{\varepsilon\to 0} \frac{P_e^{(2)}(\varepsilon)}{P_{e,2}^{(2,\min)}(\varepsilon)}\right)$$

und

$$\lim_{\varepsilon\to 0} \frac{P_R^{S_1}(\varepsilon)}{P_R^{S_2}(\varepsilon)} \;=\; \lim_{\varepsilon\to 0} \left(\frac{P_R^{S_1}(\varepsilon)}{P_R^{\mathrm{ARQ}}(\varepsilon)} \cdot \frac{P_R^{\mathrm{ARQ}}(\varepsilon)}{P_R^{S_2}(\varepsilon)}\right) \;\ge\; \left(\lim_{\varepsilon\to 0} \frac{P_{e,1}^{(2,\min)}(\varepsilon)}{P_e^{(2)}(\varepsilon)}\right) \left(\lim_{\varepsilon\to 0} \frac{P_e^{(2)}(\varepsilon)}{P_{e,2}^{(2,\max)}(\varepsilon)}\right) .$$

Interessant ist dieses Ergebnis jedoch nur, falls $d_2 = \tilde{d}_{\min}^{(2)} = \tilde{d}_{\max}^{(2)}$ gilt, denn dann ergibt sich

$$0 \;<\; \lim_{\varepsilon\to 0} \frac{P_{e,1}^{(2,\min)}(\varepsilon)}{P_{e,2}^{(2,\max)}(\varepsilon)} \;\le\; \lim_{\varepsilon\to 0} \frac{P_R^{\mathrm{S}_1}(\varepsilon)}{P_R^{\mathrm{S}_2}(\varepsilon)} \;\le\; \lim_{\varepsilon\to 0} \frac{P_{e,1}^{(2,\max)}(\varepsilon)}{P_{e,2}^{(2,\min)}(\varepsilon)} \;<\; \infty\,.$$

Anderenfalls erhalten wir

$$0 \;\le\; \lim_{\varepsilon\to 0} \frac{P_R^{\mathrm{S}_1}(\varepsilon)}{P_R^{\mathrm{S}_2}(\varepsilon)} \;\le\; \infty\,.$$

Bei geeigneter Wahl der Matrix $\tilde{\mathbf{G}}_m$ erreichen S_1 und S_2 also die gleiche Fehlerordnung.

Wir wollen nun noch die Durchsätze dieser beiden Übertragungssysteme vergleichen. Nach (6.8) und (6.9) gilt für die mittlere Anzahl der Übertragungen:

$$\begin{aligned} V_{\mathrm{SR}}^{(\nu)} &= \mathrm{Prob}(A_0^c + A_0^e) + \sum_{i=1}^{\infty}(i+1)\mathrm{Prob}(A_0^d \cap E_1^{d,\nu} \cap \ldots \cap E_{i-1}^{d,\nu} \cap (E_i^{c,\nu} + E_i^{e,\nu}))\,,\ \nu = 1,\,2\,,\\ V_{\mathrm{GBN}_\nu}^{(\nu)} &= \mathrm{Prob}(A_0^c + A_0^e) + \sum_{i=1}^{\infty}(iN_\nu+1)\mathrm{Prob}(A_0^d \cap E_1^{d,\nu} \cap \ldots \cap E_{i-1}^{d,\nu} \cap (E_i^{c,\nu} + E_i^{e,\nu}))\,,\ \nu = 1,\,2\,, \end{aligned}$$

wobei wir die mit der Fehlererkennung und -korrektur verbundenen Ereignisse $E_j^{c,\nu}$, $E_j^{e,\nu}$, $E_j^{d,\nu}$, $j \in \mathbb{N}$, mit den Indizes $\nu = 1$ bzw. $\nu = 2$ entsprechend den Systemen S_1 und S_2 versehen haben. Die Wahrscheinlichkeit dafür, daß ein Wort spätestens nach der l-ten Übertragung, $l \in \mathbb{N}$, vom Empfänger angenommen wird, beträgt

$$\Pi_l^{(\nu)} \;:=\; \mathrm{Prob}(A_0^c + A_0^e) + \sum_{i=1}^{l-1}\mathrm{Prob}(A_0^d \cap E_1^{d,\nu} \cap \ldots \cap E_{i-1}^{d,\nu} \cap (E_i^{c,\nu} + E_i^{e,\nu}))\,,\ \nu = 1,\,2\,.$$

Aus der Konstruktion von S_1 und S_2 folgt

$$E_j^{c,1} = E_j^{c,2}\,,\quad E_j^{e,1} = E_j^{e,2} \;\text{ und }\; E_j^{d,1} = E_j^{d,2} \quad \text{für } j = 1,\ldots,m-1\,.$$

Es gilt also $\Pi_l^{(1)} = \Pi_l^{(2)}$ für alle $l = 1,\ldots,m$ (s. [12, S. 202ff]). Bei Verwendung des Selective-Repeat-Protokolls kann S_2 deshalb nur dann einen höheren Durchsatz als S_1 erreichen, wenn $\Pi_{m+1}^{(2)} > \Pi_{m+1}^{(1)}$ gilt (s. [12, S. 204]). Dies ist äquivalent zu

$$\mathrm{Prob}(A_0^d \cap E_1^{d,2} \cap \ldots \cap E_{m-1}^{d,2} \cap (E_m^{c,2} + E_m^{e,2})) \;>\; \mathrm{Prob}(A_0^d \cap E_1^{d,1} \cap \ldots \cap E_{m-1}^{d,1} \cap (E_m^{c,1} + E_m^{e,1}))\,. \quad (6.11)$$

Wenn wir wieder annehmen, daß $\tilde{d}_{\min}^{(1)} \le \tilde{d}_m \le \tilde{d}_{\max}^{(1)}$ gilt, tritt der Unterschied zwischen den beiden Systemen erstmals bei der Fehlerkorrektur nach der m-ten Reübertragung deutlich zutage. Die Bedingung (6.11) ist sicher erfüllt, wenn der Code $\mathcal{C}_1$ einen größeren Minimalabstand als $\mathcal{C}_1^{(m-1)}$ hat. Dies hatten wir vorausgesetzt (vgl. Abschnitte 6.1 und 6.2). Das System S_2 erreicht also bei Verwendung des Selective-Repeat-Protokolls einen höheren Durchsatz als S_1. Die tatsächliche Zunahme des Durchsatzes hängt — außer von der Bitfehlerwahrscheinlichkeit ε —

von der Differenz der Wahrscheinlichkeiten $\mathrm{Prob}(A_0^d \cap E_1^{d,\nu} \cap \ldots \cap E_{m-1}^{d,\nu} \cap (E_m^{c,\nu} + E_m^{e,\nu}))$, $\nu = 1, 2$, ab.

Bei Verwendung des Go-Back-Protokolls gilt grundsätzlich das gleiche wie beim Selective-Repeat-Protokoll, wir müssen hier jedoch noch die Auswirkungen unterschiedlicher Rückgrifftiefen N_1 und N_2 berücksichtigen. Wenn $N_2 \leq N_1$ ist, erreicht S_2 einen höheren Durchsatz als S_1. Anderenfalls muß die Wahrscheinlichkeit $\mathrm{Prob}(A_0^d \cap E_1^{d,2} \cap \ldots \cap E_{m-1}^{d,2} \cap (E_m^{c,2} + E_m^{e,2}))$ groß genug sein, um die zusätzlichen Übertragungen, die durch die größere Rückgrifftiefe verursacht werden, ausgleichen zu können. Dies muß bei der Wahl der Matrix $\tilde{\mathbf{G}}_m$ berücksichtigt werden.

6.4.1 Resümee

Wir haben gesehen, daß zwei vergleichbare GH-ARQ-Systeme verschiedener Tiefe — bzw. ein HII-ARQ- und ein GH-ARQ-System — unter gewissen Voraussetzungen die gleiche Fehlerordnung erreichen. Eine größere Tiefe bewirkt dann bei Verwendung des Selective-Repeat-Protokolls eine Verbesserung des Durchsatzes. Beim Go-Back-Protokoll gilt dies nur mit Einschränkungen.

Morgera und Krishna empfehlen, bei Verwendung des Selective-Repeat-Protokolls die Tiefe m so zu wählen, daß $\Pi_m(\varepsilon_{\max}) \approx 1$ für die größte zu erwartende Bitfehlerwahrscheinlichkeit $\varepsilon_{\max}$ gilt. Die Durchsatzrate dieses GH-ARQ-Systems beträgt dann im schlechtesten Fall $\frac{1}{m}$ (s. [12, S. 204]). æ

Abschließende Betrachtung

In dieser Arbeit haben wir das ARQ- und verschiedene Hybrid-ARQ-Systeme analysiert und verglichen. Wir konnten dabei feststellen, daß jedes dieser Übertragungssysteme seine eigenen, spezifischen Vor- und Nachteile hat. Insbesondere gibt es für jedes der betrachteten Verfahren eine geeignete Anwendung.

Bei der Analyse der Restfehlerwahrscheinlichkeit des HII-ARQ- und des GH-ARQ-Systems wurden leichte Verbesserungen gegenüber den in der herangezogenen Literatur genannten Ergebnissen erreicht. Die Analyse des Durchsatzes dieser beiden Übertragungssysteme bei Verwendung des Go-Back-Protokolls scheint neu zu sein. Das gilt auch für die entsprechenden Vergleiche mit dem ARQ- und dem HI-ARQ-System.

Offen bleibt die Frage, inwieweit sich die bei den Vergleichen der Übertragungssysteme erhaltenen Ergebnisse für gedächtnislose binäre symmetrische Kanäle mit kleinen Bitfehlerwahrscheinlichkeiten auch auf andere Übertragungskanäle anwenden lassen. Auch eine numerische Auswertung der Ergebnisse anhand geeigneter Beispiele wäre sicher eine interessante Aufgabe.

Literaturverzeichnis

[1] F. Ananasso, R. Crescimbeni, G. Gallinaro, S. Tirró, „Digital Transmission“, in: *Satellite Communication Systems Design*, herausgegeben von Sebastiano Tirró. New York: Plenum, 1993.

[2] Vijay K. Bhargava, David Haccoun, Robert Matyas, Peter P. Nuspl, *Digital Communications by Satellite*. New York: Wiley, 1981.

[3] Richard E. Blahut, *Algebraic Methods for Signal Processing and Communications Coding*. New York: Springer, 1991.

[4] Martin Bossert, *Kanalcodierung*. Stuttgart: Teubner, 1992.

[5] Werner Heise, Pasquale Quattrocchi, *Informations- und Codierungstheorie*. Berlin: Springer, 1983.

[6] Tadao Kasami, Tohru Fujiwara, Shu Lin, „A Concatenated Coding Scheme for Error Control“, *IEEE Trans. Commun.*, Vol. COM-34, S. 481-488, 1986.

[7] Torleiv Kløve, Michael Miller, „The Detection of Errors After Error-Correction Decoding“, *IEEE Trans. Commun.*, Vol. COM-32, S. 511-517, 1984.

[8] Hari Krishna, *Computational Complexity of Bilinear Forms*. Berlin: Springer, 1987.

[9] Hari Krishna, Salvatore D. Morgera, „A New Error Control Scheme for Hybrid ARQ Systems“, *IEEE Trans. Commun.*, Vol. COM-35, S. 981-989, 1987.

[10] Jacobus H. van Lint, *Introduction to Coding Theory*. Berlin: Springer, 2. Auflage, 1992.

[11] Florence J. MacWilliams, Neil J. A. Sloane, *The Theory of Error-Correcting Codes*, Band I und II. Amsterdam: North Holland, 1977.

[12] Salvatore D. Morgera, Hari Krishna, *Digital Signal Processing: Applications to Communications and Algebraic Coding Theories*. San Diego: Academic, 1988.

[13] Henri J. Nussbaumer, *Computer Communication Systems, Volume 1: Data Circuits, Error Detection, Data Links*. Chichester: Wiley, 1990.

[14] Henri J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*. Berlin: Springer, 2. Auflage, 1982.

[15] Vitalice K. Oduol, Salvatore D. Morgera, „Performance Evaluation of the Generalized Type-II Hybrid ARQ Scheme with Noisy Feedback on Markov Channels“, *IEEE Trans. Commun.*, Vol. COM-41, S. 32-40, 1993.

[16] Man Young Rhee, *Error-Correcting Coding Theory*. New York: McGraw-Hill, 1989.

[17] Steven Roman, *Coding and Information Theory*. New York: Springer, 1992.

[18] Yu-Ming Wang, Shu Lin, „A Modified Selective-Repeat Type-II Hybrid ARQ System and Its Performance Analysis“, *IEEE Trans. Commun.*, Vol. COM-31, S. 593-607, 1983.

[19] Jack K. Wolf, Dexter Chun, „The Single Burst Error Detection Performance of Binary Cyclic Codes“, *IEEE Trans. Commun.*, Vol. COM-42, S. 11-13, 1994.

[20] William W. Wu, David Haccoun, Robert Peile, Yasuo Hirata, „Coding for Satellite Communication“, *IEEE Journal on Sel. Areas in Commun.*, Vol. SAC-5, S. 724-748, 1987.