

# Kanalkodierung

Leistungsnachweis WS13/14

Aufgabe 1

## **Vergleich der Leistungsfähigkeit serieller und paralleler Kodeverkettungen am Beispiel des (7,4,3)-HAMMING-Kodes**

Julia Grabinsky

Julia.Grabinsky@mailbox.tu-dresden.de

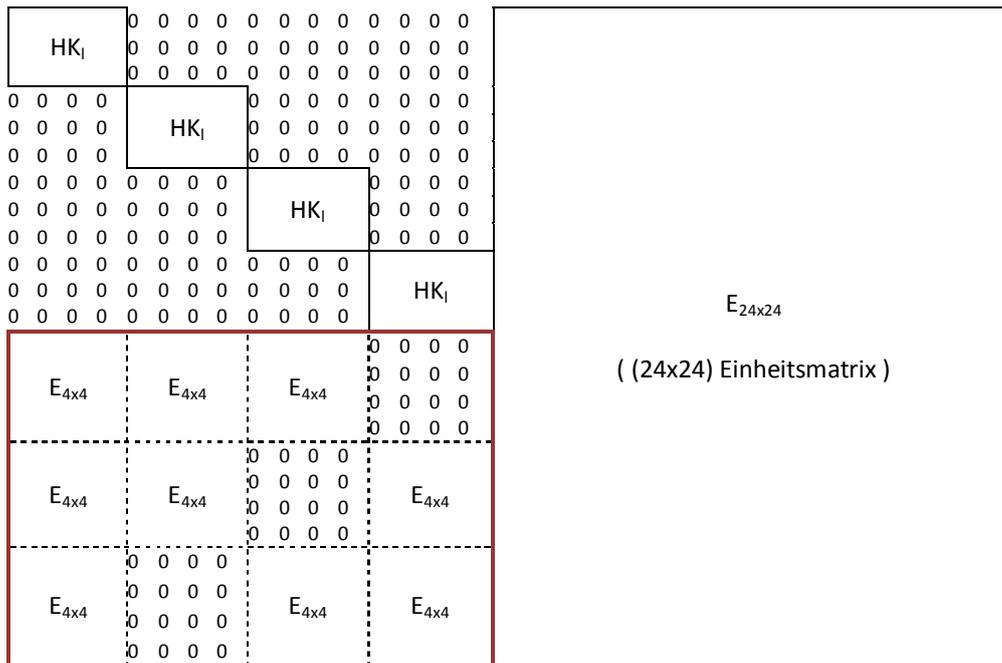
Matrikelnr.: 3118283

## Inhaltsverzeichnis

|  |   |
|--|---|
| 1. Aufgabe .....   | 3 |
| 2. Kontrollmatrizen der Kodeverkettungen .....                             | 3 |
| 2.1 Kontrollmatrix der parallelen Verkettung des (7,4,3)Hammingkodes ..... | 3 |
| 2.2 Kontrollmatrix der seriellen Verkettung des (7,4,3)Hammingkodes .....  | 4 |
| 3. Programm .....  | 6 |
| 3.1. Implementierung.....  | 6 |
| 3.2 Ausführung.....  | 7 |
| 4. Ergebnisse.....   | 8 |



Diese Kontrollmatrix lässt sich folgendermaßen schematisch darstellen (bzw. aufbauen):



**Abb. 2:** Schematische Darstellung der Kontrollmatrix der parallelen Verkettung.

Die rechte Teilmatrix der Kontrollmatrix ist die Einheitsmatrix der Dimension  $k_p = 24$ . Die linke Teilmatrix, deren Spalten den Informationsstellen des Kanalkodewortes der Verkettung entsprechen, setzt sich, schematisch betrachtet, oben aus einer Diagonalmatrix zusammen, auf deren Hauptdiagonale  $l_1$ -mal die Teilkontrollmatrix  $HK_1$  (der Informationsstellen) steht. Die untere Teilmatrix spiegelt die Matrix  $HK_1$  selbst wider, deren Elemente allerdings statt „0“ und „1“ die  $(l_2 \times l_2)$  Nullmatrix und  $(l_2 \times l_2)$  Einheitsmatrix sind ( $l_2 = 4$ ).

Nach diesem Schema lassen sich die Kontrollmatrizen paralleler Verkettungen beliebiger Hammingcodes analog aufbauen.

## 2.2 Kontrollmatrix der seriellen Verkettung des (7,4,3)Hammingcodes

Die Kontrollmatrix der seriellen Kodeverkettung ist eine  $(33 \times 49)$  Matrix. Im Vergleich zur Kontrollmatrix der parallelen Verkettung werden in die Kontrollmatrix der seriellen Verkettung zusätzlich die Kontrollgleichungen der redundanten Stellen (des äußeren Codes) über redundante Stellen (des inneren Codes) eingefügt. Es kommen somit, im Vergleich zur parallelen Verkettung,  $(k_1 \cdot k_2)$  Zeilen und Spalten hinzu. Diese Zeilen sind in der linken Teilkontrollmatrix bzgl. der Informationsstellen ausschließlich mit „0“ besetzt. In der Teilkontrollmatrix bzgl. der Redundanzstellen findet sich die Kontrollmatrix des äußeren Codes in den entsprechenden Zeilen derart wieder, dass sie sich in die Hauptdiagonale aus den Werten „1“ einfügt. Diese rechte Teilmatrix ist dadurch keine Einheitsmatrix mehr, sondern eine sehr dünn besetzte untere Dreiecksmatrix.

Die Kontrollmatrix hat folgende (binären) Einträge (sh. Abb. 3):



### 3. Programm

#### 3.1. Implementierung

Für den Vergleich der Leistungsfähigkeit serieller und paralleler Verkettung habe ich die iterative LDPC-Dekodierung unter Verwendung des sub-optimalen min-sum Algorithmus' (*decode\_min\_sum\_Algo.m*) in MATLAB implementiert.

Das Programm erzeugt zufällige Quellenkodewörter [qWort]<sup>1</sup>, die kodiert [a] und moduliert [a\_mod] werden. Pro Quellenkodewort werden mehrere Übertragungen simuliert, indem auf die modulierte Kanalkodefolge [a\_mod] eine, nach Standardnormalverteilung mit  $\mu = 0$  und

$\sigma^2 = 1/(2 \cdot R \cdot 10^{(E_b/N_0/10)})$ , zufällige Fehlerfolge addiert wird. Die somit entstandene „Empfangsfolge“ [b] wird iterativ mithilfe des min-sum Algorithmus' dekodiert. Dabei werden höchstens sechs Iterationsschritte ausgeführt (festgelegt durch [MaxDekodIter]). Die verschiedenen Dekodierergebnisse – korrekte Rekonstruktion, Falschkorrektur und Dekodierversagen – werden zum einen bzgl. der aufgetretenen Fehleranzahl (absolute Fehleranzahl und Bit-Error-Rate) und zum anderen bzgl. verschiedener  $E_b/N_0$ -Werte in Balkendiagrammen dargestellt.

Diese Simulation der Übertragung und Dekodierung soll solange durchgeführt werden bis bei einem weiteren Durchlauf mit gleicher Anzahl an Gesamtübertragungen keine signifikanten Abweichungen mehr auftreten, d. h. die Dekodierergebnisse (dividiert durch Anzahl der Übertragungen) pro  $E_b/N_0$ -Wert  $\epsilon$ -nahe beieinander liegen. Trifft dies nicht zu, wird die Summe dieser beiden Durchläufe mit dem darauf folgenden Durchlauf der verdoppelten Anzahl an Dekodierungen verglichen. Diese Verdopplung wird in dem Programm durch die Verdopplung der Anzahl der Quellenkodewortzeugungen realisiert.

Das Programm wird unabhängig voneinander für die parallele ([parall]=1) oder die serielle ([parall]=0) Kodeverkettung ausgeführt.

Getroffene Festlegungen (für die hier aufgezeigten Vergleiche):

- maximale Anzahl an Dekodierungsiterationen: [MaxDekodIter] = 6
- $\epsilon$ : [eps] = 0.005 (d.h. die Dekodierergebnisse zweier Durchläufe sollen jeweils höchstens 0.5% voneinander abweichen)
- $E_b/N_0 \in \{1.5, 2.0, 2.5, 3.0\}$
- pro Quellenkodewort (QKW) und pro  $E_b/N_0$ -Wert finden 60 „Übertragungen“ (Ü.) und Dekodierungen statt (somit 240 Dekodierungen pro QKW)
- 1. und 2. Durchlauf erzeugen je 20 QKW und simulieren somit zusammen 9600 Ü.

Die Matrix [fehler] (Tabelle 1) speichert die Dekodierergebnisse bzgl. der absoluten Fehleranzahl (d. h. bzgl. des Gewichtes des Fehlerwortes) und ist somit wesentlicher Teil der Grundlage der Auswertung. Die Fehleranzahl ist dabei implizit durch den Spaltenindex gegeben. Ein Fehlergewicht von „0“ oder größer als  $n/3$  wird nicht für die Auswertung bzgl. der Fehleranzahl berücksichtigt, geht aber in die Dekodierergebnisse bzgl. der  $E_b/N_0$ -Werte ein (sh. [SNR] in Tabelle 2).

| Index |                                   | Fehleranzahl |   |   |     |                       |
|-------|-----------------------------------|--------------|---|---|-----|-----------------------|
|       |                                   | 1            | 2 | 3 | ... | $\lfloor n/3 \rfloor$ |
| 1     | Anzahl an Dekodierungen           | „[fehler]“   |   |   |     |                       |
| 2     | Anzahl korrekter Rekonstruktionen |              |   |   |     |                       |
| 3     | Anzahl der Falschkorrekturen      |              |   |   |     |                       |
| 4     | Anzahl des Dekodierversagens      |              |   |   |     |                       |

**Tabelle 1:** Aufbau der Dekodierergebnis-Matrix [fehler] bzgl. der absoluten Fehleranzahl.

<sup>1</sup> In eckigen Klammern steht der entsprechende Variablenname, wie er im Programm verwendet wird.

Die Matrix [SNR] (Tabelle 2) speichert die Dekodierergebnisse bzgl. vier verschiedener  $E_b/N_0$ -Werte und ist somit ein weiterer wesentlicher Teil der Grundlage der Auswertung.

| Index                               | 1       | 2   | 3   | 4   |
|-------------------------------------|---------|-----|-----|-----|
| 1 Anzahl an Dekodierungen           | „[SNR]“ |     |     |     |
| 2 Anzahl korrekter Rekonstruktionen |         |     |     |     |
| 3 Anzahl der Falschkorrekturen      |         |     |     |     |
| 4 Anzahl des Dekodierversagens      |         |     |     |     |
| 5 $E_b/N_0$ -Wert                   | 1.5     | 2.0 | 2.5 | 3.0 |

**Tabelle 2:** Aufbau der Dekodierergebnis-Matrix [SNR] bzgl. verschiedener  $E_b/N_0$ -Werte.

### 3.2 Ausführung

Das auszuführende Hauptprogramm ist [**Kodeverkett\_Kod\_Dekod.m**], in welchem die Variable [parall] auf „1“ (bzw. „0“) für die Ausführung einer parallelen (bzw. seriellen) Verkettung gesetzt werden muss.

Um nicht sehr lange auf ein Ergebnis warten zu müssen, ist es ratsam die Signifikanzschwelle [eps] zu erhöhen (sh. nächster Absatz). Die *Speicherung* sowohl der Grafiken als auch der Dekodierergebnis-Matrizen habe ich auskommentiert.

Bei beiden Ausführungen, sowohl der parallelen als auch der seriellen Verkettung, habe ich (aus zeitlichen Gründen) das Programm abgebrochen bevor die Signifikanzschwelle von 0.5 % für alle Dekodierergebnisse erreicht wurde. Bei vorangegangenen Probeläufen war diese Schwelle [eps] deutlich höher gesetzt und führte oftmals dazu, dass nur die minimale Anzahl an Durchläufen ausgeführt wurde, welche allerdings bereits die Tendenz der im Folgenden dargestellten Ergebnisse aufzeigte.

Die Ausführung der parallelen Verkettung wurde nach 87963 Dekodierungen abgebrochen. Die Auswertung der letzten zwei gleichmächtigen Durchläufe (knapp 39000 Dekod.) ergibt, dass die Abweichungen der korrekten Rekonstruktionen zwar über 0.5%, aber unter 1% liegen ([eps]=0.01 hätte zum Abbruch geführt). Die Werte der Falschkorrektur unterliegen der signifikanten Abweichung. (Das Dekodierergebnis „Dekodierversagen“ ergibt sich aus den Werten zur korrekten Rekonstruktion und zur Falschkorrektur, und wird somit hier nicht berücksichtigt.)

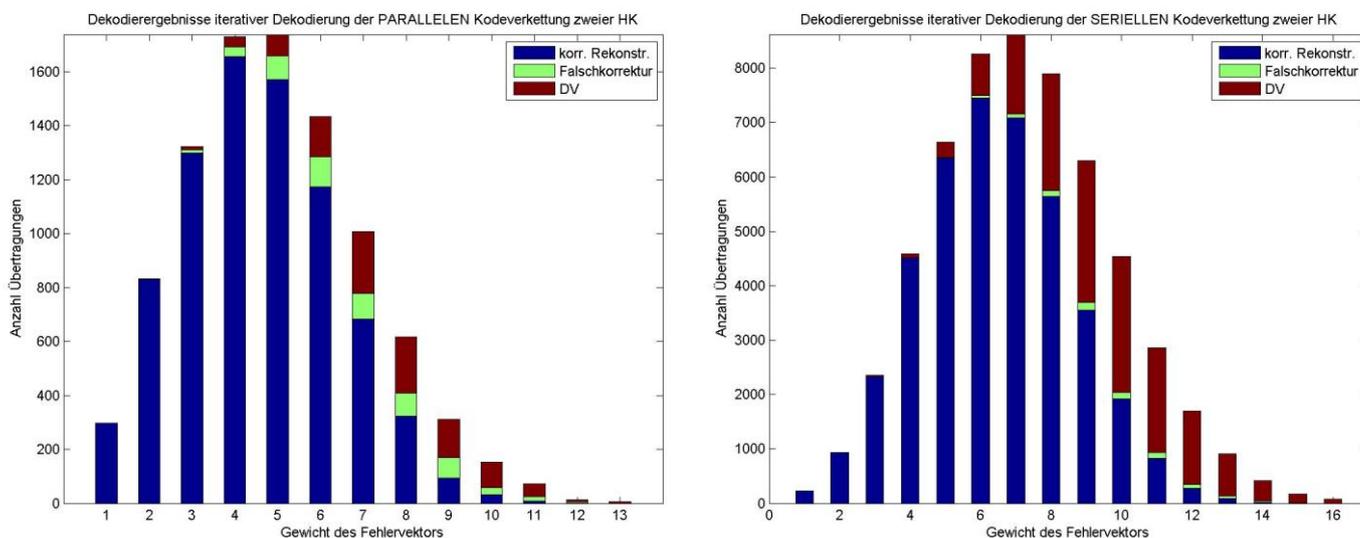
| prozentuale Abweichungen der Dekodierergebnisse der <i>parallelen</i> Verkettung |             |             |             |             |
|--|-------------|-------------|-------------|-------------|
|  | $E_b/N_0 =$ |             |             |             |
|  | 1.5         | 2.0         | 2.5         | 3.0         |
| korrekte Rekonstruktion  | 0.27        | <b>0.85</b> | <b>0.72</b> | <b>0.84</b> |
| Falschkorrektur  | 0.36        | 0.19        | 0.23        | 0.28        |

Die Ausführung der seriellen Verkettung wurde bereits nach 56559 Dekodierungen abgebrochen. Die Auswertung der letzten zwei gleichmächtigen Durchläufe (ca. 19200 Dekod.) ergibt, dass die Abweichungen der korrekten Rekonstruktionen sowohl über 0.5% als auch, für  $E_b/N_0 \in \{1.5, 2.0\}$ , über 1% liegen. Hier hätte erst [eps]=0.017 zum Abbruch geführt.

| prozentuale Abweichungen der Dekodierergebnisse der <i>seriellen</i> Verkettung |             |             |      |             |
|---|-------------|-------------|------|-------------|
|   | $E_b/N_0 =$ |             |      |             |
|   | 1.5         | 2.0         | 2.5  | 3.0         |
| korrekte Rekonstruktion   | <b>1.65</b> | <b>1.45</b> | 0.17 | <b>0.62</b> |
| Falschkorrektur   | 0.06        | 0.33        | 0.08 | 0.21        |

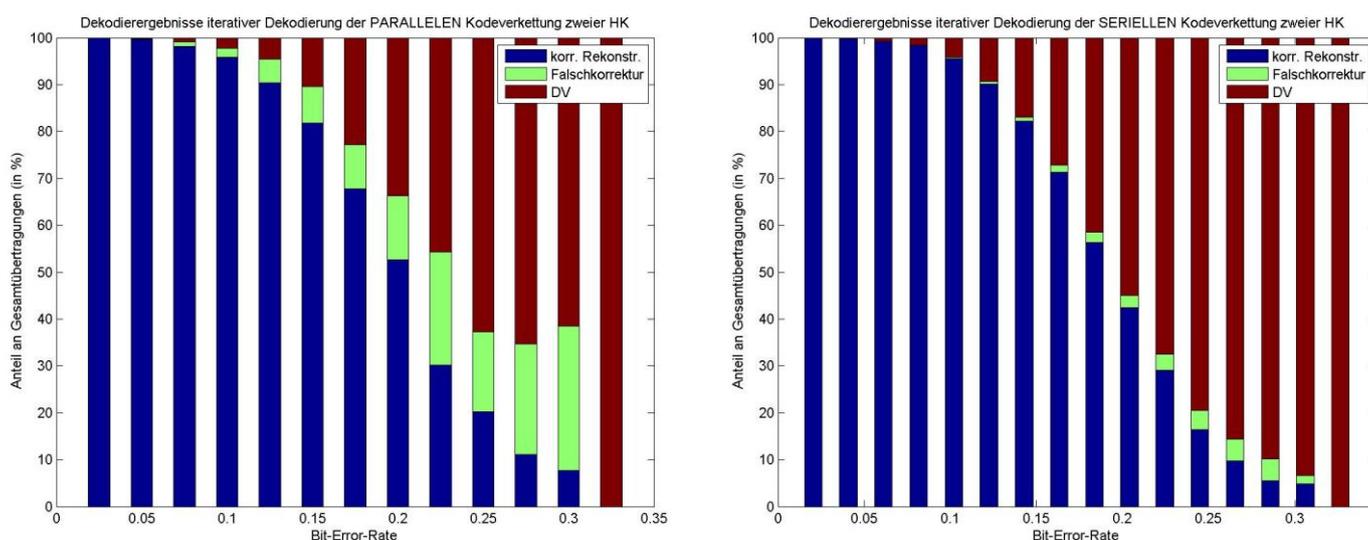
## 4. Ergebnisse

Die folgenden Balkendiagramme zeigen die Leistungsfähigkeit der parallelen und seriellen Verkettung anhand der Dekodierergebnisse zu unterschiedlichen Bezugsgrößen.



**Abb. 5:** Tatsächliche Anzahl der Dekodierergebnisse bzgl. der absoluten Fehleranzahl pro Kanalkodewort bei paralleler (links) und serieller (rechts) Verkettung.

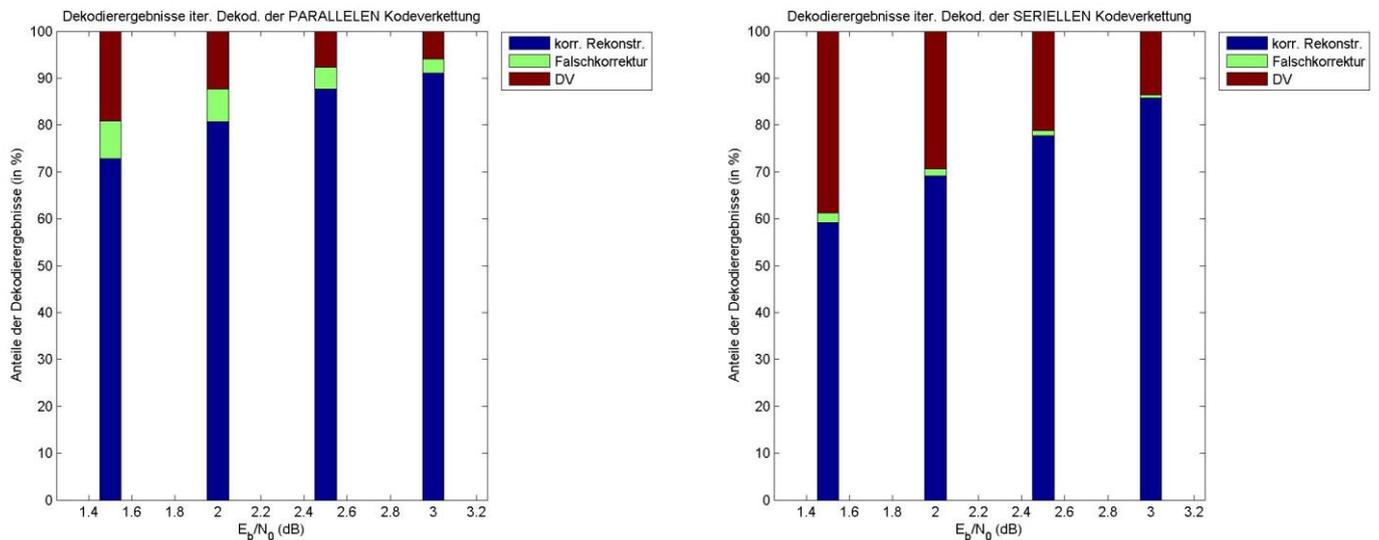
Abb. 5 zeigt die tatsächliche Anzahl der jeweiligen Dekodierergebnisse bzgl. der absoluten Fehleranzahl pro Kanalkodewort bei der Übertragung. Es ist zu beachten, dass ein Kanalkodewort der seriellen Verkettung durchschnittlich mehr Fehlerbits enthält als ein Kanalkodewort der parallelen Verkettung bei Verwendung derselben Kanalbedingungen. Deshalb stellt Abb. 6 den Anteil der Dekodierergebnisse der Gesamtübertragungen in Bezug zur Bit-Error-Rate (BER) dar. Dabei ist wiederum zu beachten, dass insbesondere einer hohen BER eine geringere Anzahl an Dekodierungen zugrunde liegt als einer mittleren BER. Dennoch wird in diesen beiden Abbildungen deutlich, dass bei paralleler Verkettung deutlich mehr Falschkorrekturen auftreten als bei der seriellen Verkettung, bei



**Abb. 6:** Prozentualer Anteil der Dekodierergebnisse an der Gesamtanzahl der Dekodierungen bzgl. der Bit-Error-Rate bei paralleler (links) und serieller (rechts) Verkettung.

welcher die Falschkorrektur durch die Bildung redundanter Stellen über redundante Stellen stärker verhindert wird. Allerdings trägt es in einem solchen Fall wahrscheinlich nicht zur korrekten Rekonstruktion bei, sondern führt zum Dekodierversagen.

Betrachtet man die Anteile der Dekodierergebnisse zusammengefasst bzgl. eines  $E_b/N_0$ -Wertes (sh. Abb. 7), so wird deutlich, dass die parallele Verkettung zwar mehr Falschkorrekturen hervorruft, aber auch einen höheren Anteil an korrekten Rekonstruktionen aufweist.



**Abb. 7:** Prozentualer Anteil der Dekodierergebnisse bzgl. der vier  $E_b/N_0$ -Werte aus {1.5, 2.0, 2.5, 3.0} bei paralleler (links) und serieller (rechts) Verkettung.

Somit vermindert die Bildung redundanter Stellen über redundante Stellen, wie bei der seriellen Verkettung, das unerwünschte Auftreten von Falschkorrekturen. Die Ausführung der parallelen Verkettung erzielt jedoch mit weniger Redundanz pro Kanalkodewort einen höheren Anteil an korrekten Rekonstruktionen.