

Bild 9.40: Bitfehlerraten der (verallgemeinert) verketteten Codes.

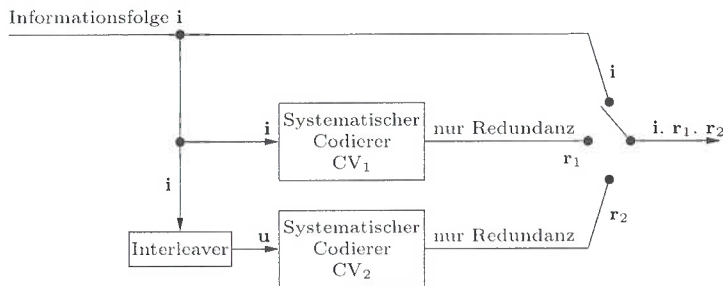


Bild 9.41: Codierschema für Turbo-Codes.

S95, BM96, RW95, WH95] und viele andere. Hauptsächlich handelt es sich da-
 i um Arbeiten zur iterativen Decodierung mittels s/s-MAP-Algorithmen ent-
 reichend Abschnitt 7.4.1. Bild 9.41 zeigt schematisch die Codierung eines Turbo-
 codes. Veranschaulicht man sich diese Art der Verkettung an Blockcodes, so er-
 nnt man, daß es sich dabei um „schlechte“ Codes handelt. Wie in Bild 9.42
 rgestellt, ist die garantierte Mindestdistanz des verketteten Codes d nur bei
 einen Mindestdistanzen der Komponentencodes d_1 und d_2 annähernd so gut wie
 r Produktcodes.

e Arbeiten zu den Turbo-Codes haben dann auch dieses Phänomen der schlech-
 n Mindestdistanz gezeigt, das sich durch ein „Abknicken“ der Bitfehlerratenkurve
 i Werten zwischen 10^{-4} und 10^{-8} hin zu der asymptotisch erwarteten Bitfeh-
 rate zeigt. Es ist bekannt, daß die Bitfehlerrate bei sehr gutem SNR einzig von
 r Mindestdistanz des Codes bestimmt wird. Dieses Abknicken wurde inzwischen
 ch in [Svi95] bzw. [BM96] theoretisch abgeleitet. Es blieb und bleibt ein rätsel-
 ftes Phänomen, warum diese schlechten Codes (aus Sicht der Mindestdistanz)
 i kleinem SNR durch die iterative Decodierung so hervorragend decodiert wer-



Bild 9.42: Resultierende Distanz bei serieller und paralleler Verkettung.

den können.

Die Decodierung wird entsprechend Abschnitt 7.4.1 durch s/s-MAP-Algorithmen
 durchgeführt, die zunächst den einen Komponentencode decodieren und damit die
 empfangenen Werte der Informationssymbole entsprechend modifizieren. Danach
 wird der zweite Komponentencode decodiert und wieder die Werte der Informati-
 on modifiziert, usw. Prinzipiell besteht jedoch kein Unterschied zu der iterativen
 Decodierung, die wir in Abschnitt 7.4.1 beschrieben haben. Es gibt jedoch zahl-
 reiche Varianten, wie und wann die empfangenen Werte der Informationsstellen
 modifiziert werden und ob die empfangenen Werte der Redundanzstellen ebenfalls
 modifiziert werden oder nicht. Des weiteren gibt es Varianten zu der Anzahl der
 Iterationen und zu den verwendeten Faltungscodes.

Dies führt uns zu dem ersten ungelösten Problem, das sowohl für Turbo-Codes,
 Produktcodes, GC-Codes, ja für lange Codes generell existiert, nämlich die Deco-
 dierung bei kleinem SNR. Da es sich in der Regel bei verketteten Codes um sehr
 lange Codes handelt, kann keine ML-Decodierung durchgeführt werden. Darüber
 hinaus ist nicht einmal bekannt, ob eine ML-Decodierung in diesem Fall sinnvoll
 ist, oder ob nicht eine s/s-MAP-Decodierung bessere Ergebnisse liefert, geschweige
 denn wie weit entfernt man noch von einer ML-Decodierung ist.

Speziell für eine iterative Decodierung ist für fast alle Fälle ungeklärt, ob und
 wohin die Iterationen konvergieren. Und für GC-Codes kommt zusätzlich das Pro-
 blem hinzu, daß nicht nur zwischen innerem und äußerem Code, sondern auch
 noch zwischen den Stufen der Paare von inneren und äußeren Codes iteriert wer-
 den kann. Somit ist auch die Decodierung von GC-Codes, die auf Faltungscodes
 basieren, ein ungelöstes Problem.

Für die Codekonstruktion ist eine ungelöste Frage welchen Code ein verwendeter
 Interleaver verkettet mit einem Code ergibt. Des weiteren eröffnet die Partitio-
 nierung von Faltungscodes so viele Möglichkeiten von Codekonstruktionen, unter
 denen sich noch „Juwelen“ befinden könnten.

9.4 GC-Codes mit Block- und Faltungscodes

Bisher haben wir GC-Codes erörtert, bei denen innere und äußere Codes jeweils
 nur Block- oder nur Faltungscodes waren. Selbstverständlich lassen sich auch GC-