

Fehlerschutz-Codierung 2: Faltungs-Code

Inhaltsverzeichnis

1 Faltungs-Codierung	1
1.1 Eigenschaften des Faltungs-Coders	2
1.1.1 Impulsantwort	2
1.1.2 Code-Baum und Trellis	3
1.1.3 Zustands-Diagramm	4
1.1.4 Weg durch den Trellis	5
2 Viterbi Decodierung von Faltungs-Codes	5
2.1 Decodierung hart entschiedener Bits	5
2.2 Distanz-Eigenschaft der Faltungs-Codes	7
2.2.1 Beispiele guter Faltungs-Codes	8
2.2.2 Katastrophen-Codes	8
2.3 Soft Decision Viterbi-Decodierung	8
2.3.1 Soft Decision	8
2.3.2 Soft Decision bei Viterbi	9
2.4 Punktierte Codes	9
2.4.1 Erasure	9
2.4.2 Punktiertes Code bei DVB-S	10
3 Verkettete Codes	11
3.1 Faltungs-Interleaver	11

Abbildungsverzeichnis

1.1 Faltungs-Coder mit $R_c = 1/2$ und Constraint Length 7	1
1.2 Faltungs-Coder mit $R_c = 1/2$ und Constraint Length 3 in unterschiedlicher Darstellung; links: $M = 2$ Register, rechts: $K = 3$ Register	2
1.3 Code-Baum des Faltungs-Coders mit $R_c = 1/2$ und Constraint Length 3	3
1.4 Trellis des Faltungs-Coders mit $R_c = 1/2$ und Constraint Length 3. Markierung an den Wegen: Daten-Bit / Code-Bits (gestrichelt: 0, durchgezogen: 1; gleichartige Wege in gleicher Farbe)	4
1.5 Trellis-Segment und Zustands-Diagramm des Faltungs-Coders mit $R_c = 1/2$ und Constraint Length $K = 3$	4
1.6 Pfad $c = [00\ 11\ 10\ 10\ 00\ 01\ 00\ 10\ 10\ 11]$ durch den Trellis für den Eingangs-Vektor $m = [01101011.00]$	5
2.1 Trellis für die empfangenen Daten $r = [00\ 11\ 10\ 10\ 00\ 11\ 00\ 10\ 10\ 11]$. An den Pfaden: Unterschiede zwischen empfangenen und nominellen Code-Bits. An den Knoten: Minimum der bis dahin aufgetretenen Unterschiede (Metrik). Es werden die Pfade weitergeführt (dick gezeichnet), die die kleinste Metrik haben (<i>survivor</i>). Entscheidung über empfangene Bits nachdem der Survivor feststeht. Decodierte Daten: $m' = [01101011.00]$; Der Fehler wurde korrigiert.	6
2.2 Trellis mit Distanzen vom Null-Pfad. Die kleinste Distanz ist $d_{\text{free}} = 5$.	7
2.3 Zustands-Diagramm (aufgeschnitten) mit Distanzen vom Null-Pfad	7
2.4 Harte und 3 Bit Soft Entscheidung bei einer BPSK Übertragung	9
2.5 Harte und weiche Entscheidung der Dibits	9
2.6 Beginn des Trellis: Encoder und Viterbi-Decoder	9
2.7 Faltungs-Coder bei DVB-S	10
2.8 Punktierungs-Schema bei DVB-S	10
3.1 Übertragungssystem mit Code-Verkettung	11
3.2 Faltungs-Interleaver und -Deinterleaver	11

Fehlerschutz-Codierung 2: Faltungs-Code

Der Name „Faltungs-Code“ betont die Ähnlichkeit zwischen der Struktur des Faltungs-Coders und der eines FIR-Filters. Bei einem Filter ist bekanntlich die Verknüpfung zwischen Ein- und Ausgangsgröße im Zeitbereich durch eine Faltung $u_a(t) = u_e(t) * h(t)$ gegeben. Ein FIR-Filter läßt sich als Schiebe-Register realisieren, bei dem nach jedem einzelnen Register ein Abgriff besteht, der jeweils mit einem Koeffizienten h_0, h_1, \dots, h_N bewertet wird. Alle bewerteten Abgriffe werden aufsummiert und bilden das Ausgangssignal. Die Koeffizienten werden aus der Impulsantwort $h(t)$ gewonnen.

Eine ähnliche Struktur weist auch ein Faltungs-Encoder auf, wobei mindestens zwei unterschiedliche Bewertungen realisiert werden. Aufgrund der Ähnlichkeit mit einem FIR-Filter ist bei einem Faltungs-Code keine Strukturierung der Daten in Blöcke erforderlich. Ganz entsprechend zu einem FIR-Filter hat ein Faltungs-Coder ein „Gedächtnis“, das mit Constraint Länge bezeichnet wird, was bedeutet, daß das aktuelle Ausgangssignal (Code-Bit) von der Vergangenheit des Eingangssignals abhängt. Gleiche Abschnitte von Eingangs-Daten führen deshalb i.a. auf unterschiedliche Ausgangs-Daten bei einem Faltungs-Encoder.

1 Faltungs-Codierung

Da der Faltungs-Encoder Bits verarbeitet, besteht das Schiebe-Register (bei einer diskreten Realisierung) aus Flip-Flops. Daher werden die Additionen Modulo 2 ausgeführt, was EXOR Verknüpfungen entspricht. Diese werden mit \oplus gekennzeichnet. Eine Bewertung der Abgriffe kann hier nur noch mit 1 oder 0 erfolgen. Die Abgriffe sind so gewählt, daß sich eine möglichst große Distanz zwischen den Code-Worten ergibt. Da hierfür keine geschlossene Theorie vorliegt, wurden sie experimentell am Computer ermittelt. Bild 1.1 ist ein Beispiel für einen Faltungs-Coder mit guten Eigenschaften, wie er z.B. bei DAB und DVB-T zum Einsatz kommt.

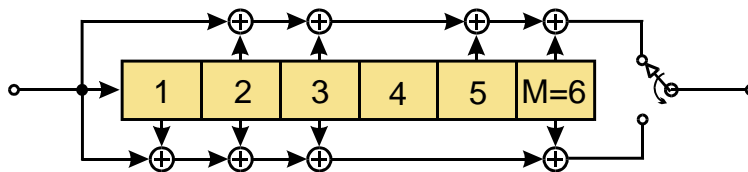


Bild 1.1: Faltungs-Coder mit $R_c = 1/2$ und Constraint Length 7

Für den Faltungs-Coder Bild 1.1 gilt:

- Jedes einlaufende Bit führt zu je 1 Bit an den beiden Ausgängen, die innerhalb der Bit-Dauer T_b auf den Ausgang gegeben werden. Daher entstehen pro Eingangs-Bit 2 Ausgangs-Bits. Die Code-Rate ist deshalb $R_c = 1/2$.
- Das Register hat $M = 6$ Stufen. Die Ausgangs-Bits werden daher von insgesamt $K = M + 1 = 7$ Bits beeinflusst. Das Gedächtnis (Constraint Length) des Faltungs-Encoders ist daher $K = M + 1 = 7$.
- Der erzeugte Faltungs-Code ist nicht systematisch, da er die Datenfolge des Eingangs nicht (in unveränderter Form) enthält. Bei der Faltungs-Codierung werden nicht systematische Codes bevorzugt.
- Jeder Pfad i , der den Eingang mit dem Ausgang verknüpft, kann durch einen Vektor der Impulsantwort h_i beschrieben werden. (1: oben, 2: unten)

$$\mathbf{h}_1 = 1011011, \quad \mathbf{h}_2 = 1111001 \quad (1.1)$$

- Die Verzögerung T_b pro Register-Stufe wird als $z^{-1} = D$ bezeichnet. Dies entspricht einer Darstellung im Bild-Bereich. Damit ergibt sich das Generator-Polynom $g_i(D)$ für jeden Zweig des Faltungs-Coders.

$$g_1(D) = 1 + D^2 + D^3 + D^5 + D^6 \quad \text{Generator-Polynom oberer Zweig} \quad (1.2)$$

$$g_2(D) = 1 + D + D^2 + D^3 + D^6 \quad \text{Generator-Polynom unterer Zweig} \quad (1.3)$$

- Der Faltungs-Coder wird auch durch das entsprechende binäre oder octale Wort charakterisiert.

$$(1 + D^2 + D^3 + D^5 + D^6; 1 + D + D^2 + D^3 + D^6) \equiv (1011011; 1111001) \equiv (133; 171)_{\text{oct}} \quad (1.4)$$

1.1 Eigenschaften des Faltungs-Coders

Die Funktionsweise der Faltungs-Codierung wird anhand eines Faltungs-Coders mit $n = 2$ Zweigen (Code-Rate $R_c = 1/2$) und der Constraint Länge $K = M + 1 = 3$ untersucht. Die Struktur dieses Coders ist in Bild 1.2 dargestellt. Beide Blockschaltbilder sind identisch. Die linke Darstellung zeigt die notwendige Länge M des Registers, während die rechte die Einfluß-Länge K betont.

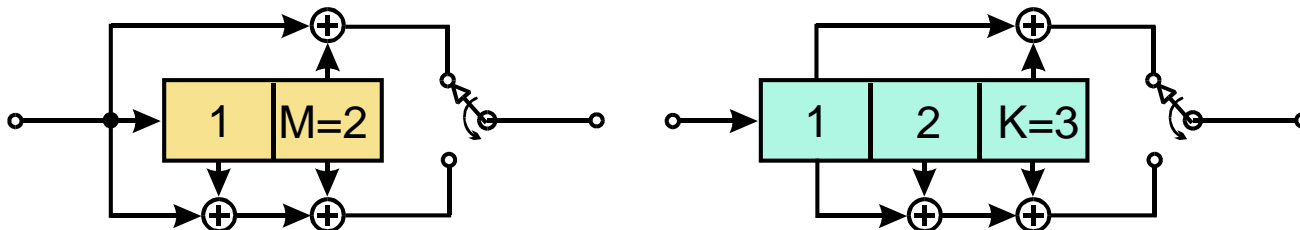


Bild 1.2: Faltungs-Coder mit $R_c = 1/2$ und Constraint Length 3 in unterschiedlicher Darstellung; links: $M = 2$ Register, rechts: $K = 3$ Register

Die Eigenschaften dieses Faltungs-Coders können wie folgt beschrieben werden:

$$(1 + D^2; 1 + D + D^2) \equiv (101; 111) \equiv (5; 7)_{\text{oct}} \tag{1.5}$$

1.1.1 Impulsantwort

Der Anfangs-Zustand (Inhalt der Register) ist dabei $M = [00]$ bzw. $K = [000]$ (per Definition). Die Impuls-Antwort ist die Antwort des Faltungs-Coders auf ein „1“ Bit, das durch sein Register durchgeschoben wird. Wie aus Bild 1.2 erkennbar ist, trägt das „1“ Bit bereits schon zur Impuls-Antwort bei, bevor es im 1. M -Register gespeichert ist. Wir erhalten daher als Ausgangsgrößen h_1 (oben) und h_2 (unten) in Abhängigkeit vom K -Register:

K Register	h_1	h_2	
100	1	1	Eingangs-Vektor : 1 0 0 Impuls-Antwort : 11 01 11
010	0	1	
001	1	1	

Als Eingangs-Datenfolge werde $m = [01101011.00]$ angenommen ($L = 8$ Nutz-Bits und 2 Tail-Bits). Der Ausgangs-Vektor für diese Datenfolge kann (wegen der Linearität) als (modulo 2) Summe der einzelnen Impuls-Antworten dargestellt werden. Dies wird für die ersten 3 Bits aus $m = [01101011.00]$ gezeigt.

Eingangs-Vektor	Ausgangs-Vektor
0	00 00 00
1	11 01 11
1	11 01 11
?	?? ?? ??
Summe \oplus :	00 11 10 10 $\times\times$ $\times\times$

Die Datenfolge $m = [01101011.00]$ kann (im Bild-Bereich) als **Polynom** dargestellt werden.

$$m(D) = D + D^2 + D^4 + D^6 + D^7 \tag{1.6}$$

Im Bild-Bereich ist eine Multiplikation (modulo 2) auszuführen, wenn im Zeitbereich eine Faltung erforderlich ist. Daher erhält man folgende Ausgangs-Polynome:

$$\begin{aligned} c_1(D) &= g_1(D) \otimes m(D) = (1 + D^2) \otimes (D + D^2 + D^4 + D^6 + D^7) \\ &= D + D^2 + D^3 + D^7 + D^8 + D^9 \end{aligned} \tag{1.7}$$

$$\begin{aligned} c_2(D) &= g_2(D) \otimes m(D) = (1 + D + D^2) \otimes (D + D^2 + D^4 + D^6 + D^7) \\ &= D + D^5 + D^9 \end{aligned} \tag{1.8}$$

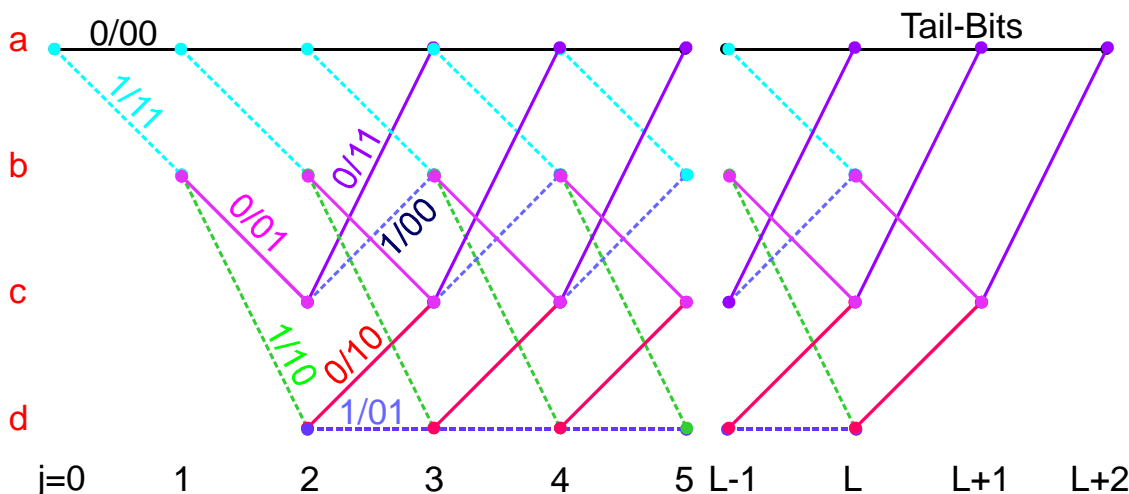


Bild 1.4: Trellis des Faltungscoders mit $R_c = 1/2$ und Constraint Length 3. Markierung an den Wegen: Daten-Bit / Code-Bits (gestrichelt: 0, durchgezogen: 1; gleichartige Wege in gleicher Farbe)

Ein Trellis ist informativer als ein Code-Baum. Der Trellis zeigt eindeutig, daß es sich bei einem Faltungscoder um eine Finite-State Maschine handelt. Die Zustände sind dabei die im Schiebe-Register gespeicherten Nachrichten-Bits. Bei dem Beispiel mit $M = 2$ Registern können sich daher $2^M = 4$ Zustände ergeben.

1.1.3 Zustands-Diagramm

Die 4 Zustände des Faltungscoders (Bild 1.2) sieht man auch an einem einzelnen Segment des Trellis und an einem Zustands-Diagramm, Bild 1.5.

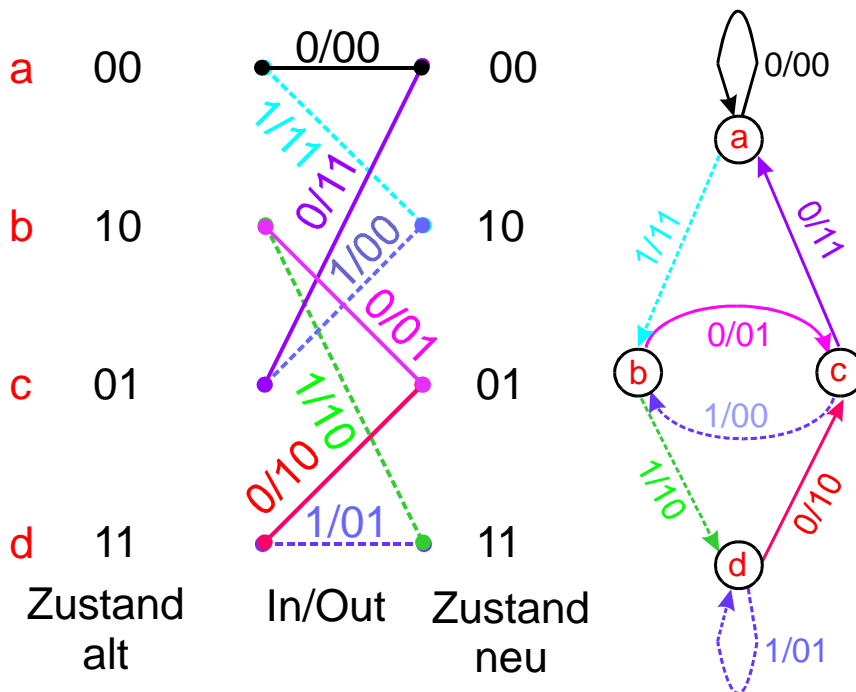


Bild 1.5: Trellis-Segment und Zustands-Diagramm des Faltungscoders mit $R_c = 1/2$ und Constraint Length $K = 3$

Es ist nicht möglich, von jedem beliebigen Zustand aus einen anderen beliebigen Zustand zu erreichen. Wenn der Zustand z.B. $a = 00$ ist, kann nur $b = 10$ oder wieder $a = 00$ erreicht werden, denn das nächste Datenbit, das eingelesen wird, kann nur 1 oder 0 sein.

Die einzigen Zustände, zwischen denen direkt gewechselt werden kann, sind $b = 10$ und $c = 01$. Dafür können diese aber nicht beibehalten werden.

1.1.4 Weg durch den Trellis

Trellis-Segment und Zustands-Diagramm beschreiben den Code vollständig. Der komplette Trellis ist dagegen günstiger, wenn die Codierung einer Daten-Folge untersucht werden soll. Bild 1.6 stellt den Weg $c = [00\ 11\ 10\ 10\ 00\ 01\ 00\ 10\ 10\ 11]$ durch den Trellis dar, wenn der Eingangs-Vektor $m = [01101011.00]$ codiert wird.

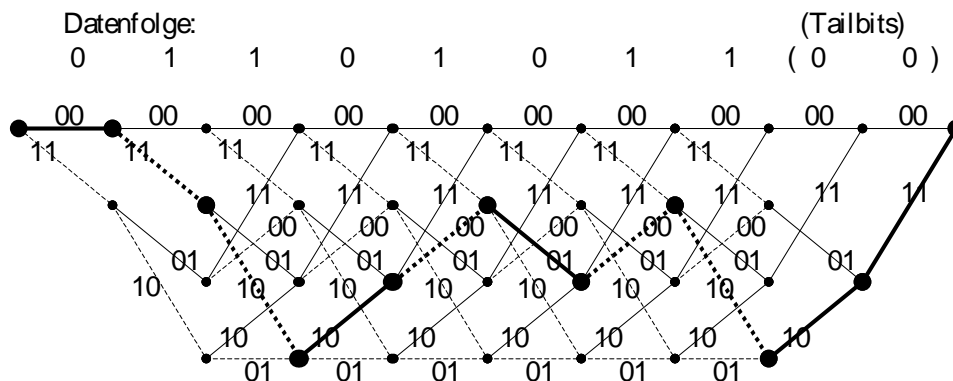


Bild 1.6: Pfad $c = [00\ 11\ 10\ 10\ 00\ 01\ 00\ 10\ 10\ 11]$ durch den Trellis für den Eingangs-Vektor $m = [01101011.00]$

Dieser Weg durch den Trellis ist der Schlüssel für die Decodierung des Faltungs-Codes. Die zugehörige Decodierungs-Strategie ist der Viterbi-Algorithmus.

2 Viterbi Decodierung von Faltungs-Codes

2.1 Decodierung hart entschiedener Bits

Zur einfacheren Erklärung des Viterbi-Algorithmus seien folgende Annahmen gemacht.

- Zu Beginn sei das Register des Faltungs-Encoders $M = [00]$ (D.h., er befindet sich im Zustand a).
- Der Viterbi-Algorithmus ist darauf synchronisiert.
- Der Entscheider liefert 0 bzw. 1 (harte Entscheidungen).
- Die empfangene Datenfolge sei $r = [00\ 11\ 10\ 10\ 00\ 11\ 00\ 10\ 10\ 11]$ und ist damit fehlerhaft gegenüber c .

Da der Viterbi-Decoder die Codier-Vorschrift des Faltungs-Coders kennt, ist auch bekannt, was vom Zustand a) aus möglich ist, nämlich a) (wenn 00 empfangen wird) und b) (wenn 11 empfangen wird).

Nun wird den empfangenen Daten beim Viterbi-Algorithmus aber grundsätzlich mißtraut, da ja auf dem Übertragungsweg Fehler entstanden sein können. Im Empfänger wird daher auch ein Trellis gebildet, bei dem an jedem Teil-Pfad vermerkt ist, um wieviel er sich von dem unterscheidet, was aufgrund der empfangenen Daten sein müßte, vergleiche hierzu Bild 1.4.

Als erste Daten werden 00 empfangen. Der Weg von a) nach a) unterscheidet sich um 0, während sich der Weg von a) nach b) um 2 unterscheidet, da für diesen Weg eine 11 hätte empfangen werden müssen.

Der Viterbi-Algorithmus trifft jedoch in diesem Stadium noch keine Entscheidung. Vielmehr wird diese so lange hinausgeschoben, bis sich zu einem späteren Zeitpunkt $t \geq 3$ verschiedene Pfade wieder treffen. Jetzt wird die Summe aller (bisherigen) Unterschiede (Metrik) der Pfade mit einander verglichen, Bild 2.1.

Es wird immer derjenige Pfad für den wahrscheinlichsten erklärt (maximum Likelihood), der die **kleinste Metrik (Distanz)** bis dahin hat. Der Wert der kleinsten Metrik ist an den Knoten mit großen (roten) Zahlen vermerkt, während an den Pfaden deren jeweilige Pfad-Metrik steht. Der „überlebende“ Pfad (*survivor*) wird weiter betrachtet und der andere wird eliminiert. Jetzt wird die Entscheidung getroffen über das Bit, das über den Verzweigungs-Punkt der beiden Pfade entscheidet.

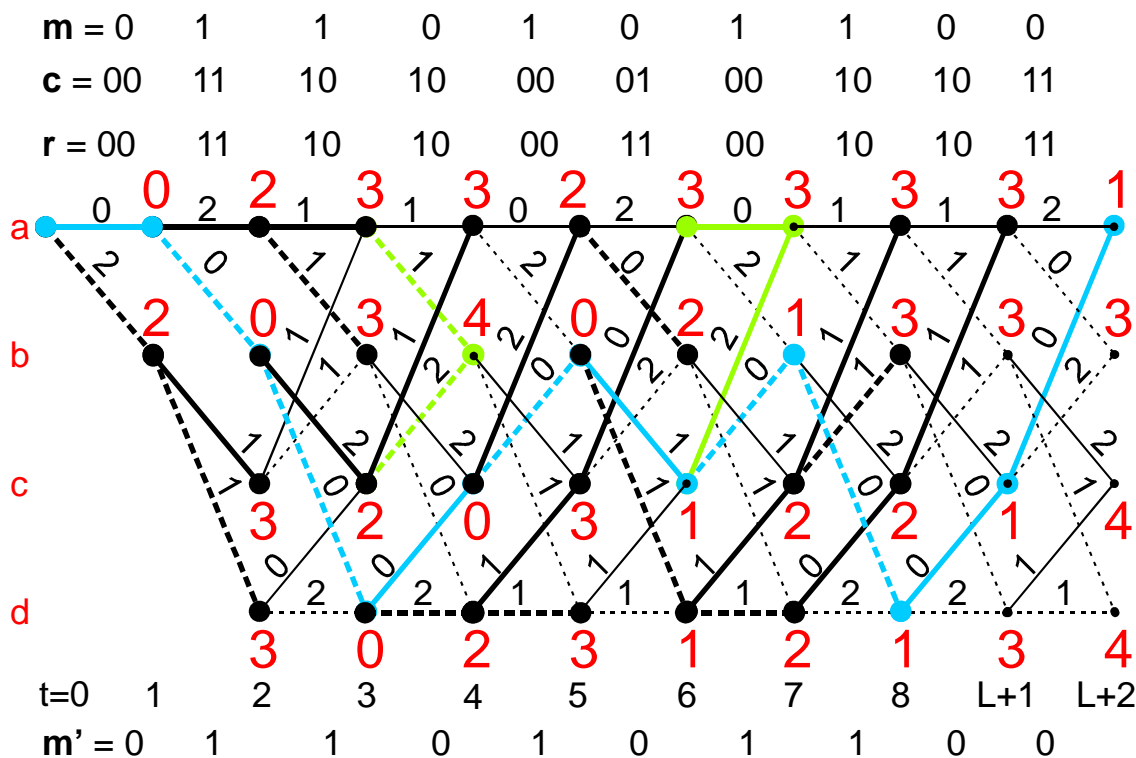


Bild 2.1: Trellis für die empfangenen Daten $r = [00\ 11\ 10\ 10\ 00\ 11\ 00\ 10\ 10\ 11]$. An den Pfaden: Unterschiede zwischen empfangenen und nominellen Code-Bits. An den Knoten: Minimum der bis dahin aufgetretenen Unterschiede (Metrik). Es werden die Pfade weitergeführt (dick gezeichnet), die die kleinste Metrik haben (*survivor*). Entscheidung über empfangene Bits nachdem der Survivor feststeht. Decodierte Daten: $m' = [01101011.00]$: Der Fehler wurde korrigiert.

Der erste Entscheidungs-Zeitpunkt ist bei $t = 3$. Hier erkennt man, daß die empfangene 00 richtig sein muß, da für die Annahme 11 dies zu einer höheren Metrik führt. Der betreffende Weg wird deshalb bei $t = 2$ abgebrochen und der andere mit der kleineren Metrik überlebt (*Survivor*). Damit kann nun bei $t = 3$ das erste decodierte Bit als 0 entschieden werden.

So lange über den Survivor noch keine Entscheidung getroffen werden kann, werden die Pfade dick gezeichnet. Während dieser Zeit muß der Viterbi-Algorithmus die zugehörigen Informationen bereithalten, nach der Entscheidung werden sie gelöscht. Damit wird vermieden, daß der Speicherbedarf exponentiell mit der Länge des Informations-Signals wächst. Der Viterbi-Algorithmus muß also hier zu jedem Zeitpunkt $2^M = 4$ Pfade und deren Verlauf berücksichtigen. Allgemein gilt für die Viterbi-Decodierung, daß eine Entscheidung bis zu der 4- bis 5-fachen der Constraint Länge K andauern kann.

Zum Zeitpunkt $t = 4$ gibt es 2 Wege (grün gezeichnet), die auf die gleiche Metrik 4 im Knoten b) führen. In diesem Fall kann nach dem Zufallsprinzip einer davon als Survivor erklärt werden. Im aktuellen Beispiel ist es gleichgültig, welcher davon als Survivor erklärt wird, denn später überlebt der Pfad sowieso nicht. Diese Konstellation wiederholt sich entsprechend bei $t = 7$ im Knoten a). Es muß aber nicht zwangsläufig zum Absterben eines derartig ausgewürfelten Weges kommen, wenn diese Situation aufgrund von Übertragungs-Fehlern der Code-Bits zustande gekommen ist. Bis der Viterbi-Decoder dann wieder in den richtigen Pfad eingemündet ist, entstehen dadurch eine Reihe von Fehlern (Bündel-Fehler).

Zum Zeitpunkt $t = 5$ kann über das 2. decodierte Bit entschieden werden. Da der zugehörige Weg gestrichelt gezeichnet ist, ist es somit eine „1“.

Zu den Zeitpunkten $L+1$ und $L+2$ werden die Tail-Bits ausgewertet. Diese sind 0 und somit dem Decoder bekannt. Der überlebende Pfad durch den Trellis ist durch seine blaue Farbe hervorgehoben.

Es zeigt sich, daß der Übertragungsfehler vom Decoder korrigiert wurde. Damit ist $m' = m$.

2.2 Distanz-Eigenschaft der Faltungen-Codes

Faltungen-Codes sind lineare Codes. Daher kann die minimale Distanz als Abstand vom $[00000 \dots]$ Pfad gefunden werden. Es werden dann Nullen gesendet und die Wege im Trellis verfolgt, die beginnend mit 0 von dort abbiegen um anschließend wieder zum $[00000 \dots]$ Pfad zurückzukehren, Bilder 2.2 und 2.3. Die dabei minimal auftretende Distanz wird freie Distanz d_{free} genannt (im Unterschied zur Hamming-Distanz d_H bei Block-Codes).

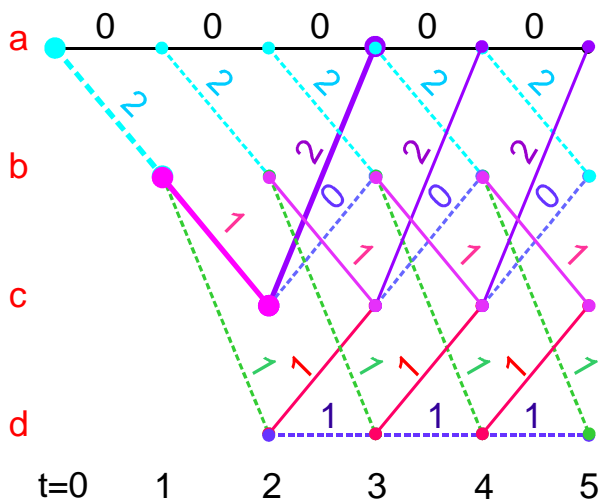


Bild 2.2: Trellis mit Distanzen vom Null-Pfad. Die kleinste Distanz ist $d_{\text{free}} = 5$.

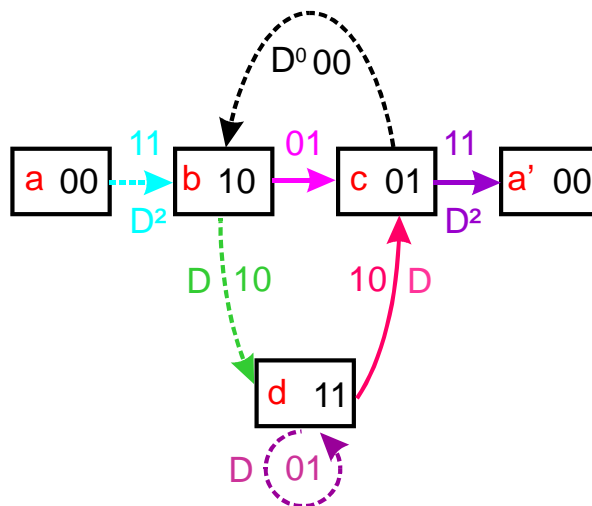


Bild 2.3: Zustands-Diagramm (aufgeschnitten) mit Distanzen vom Null-Pfad

Wird also bei ausschließlich gesendeten Nullen infolge von Übertragungs-Fehlern der $[00000 \dots]$ Pfad verlassen und anschließend wieder dorthin zurückgekehrt, hat der Viterbi-Decoder falsch decodiert. Dadurch entstehen bei den decodierten Daten i.a. Bündelfehler (Fehler-Bursts).

Die Fähigkeit eines Faltungen-Codes zur Korrektur von t Fehlern beträgt ($\lfloor x \rfloor$: größter Integer-Wert $\leq x$):

$$t = \left\lfloor \frac{d_{\text{free}} - 1}{2} \right\rfloor \tag{2.1}$$

Diese Anzahl von Fehlern darf demnach innerhalb der Zeitspanne höchstens auftreten, die der Viterbi-Decoder zur Entscheidung benötigt, also innerhalb der 4- bis 5-fachen der Constraint Länge K .

Der Trellis Bild 2.2 zeigt, daß der Viterbi-Decoder im Fehler-Fall seine Fehler nur gemäß den zulässigen Übergängen machen kann. Würde also bei der Übertragung von Nullen Fehler dergestalt entstehen, daß dem Pfad mit der kleinsten freien Distanz d_{free} gefolgt wird (dick gezeichnet in Bild 2.2), so ergäbe die decodierte Datenfolge hierfür 100 und damit einen Fehler.

Das aufgeschnittene Zustands-Diagramm (Bild 2.3) zeigt eine weitere Möglichkeit die freie Distanz zu bestimmen. Hier wird wieder die Darstellung im Bild-Bereich verwendet. Die Rückführschleife im Knoten a) kann dabei weggelassen werden, da sie nichts zur Distanz beiträgt. Damit ergeben sich folgende Zustands-Gleichungen für die Zustände $X_a, X_b, X_c, X_d, X_{a'}$:

$$X_b = D^2 X_a + X_c \tag{2.2}$$

$$X_c = D X_b + D X_d \tag{2.3}$$

$$X_d = D X_b + D X_d \tag{2.4}$$

$$X_{a'} = D^2 X_c \tag{2.5}$$

Diese lassen sich in einander einsetzen und nach der Übertragungsfunktion $T(D) = X_a'/X_a$ auflösen.

$$T(D) = \frac{D^5}{1-2D} \quad (2.6)$$

$$= D^5 + 2D^6 + 4D^7 + \dots + 2^k D^{k+2} + \dots \quad (2.7)$$

Gleichung (2.7) besagt, daß es einen Weg mit Abstand 5 (Minimalabstand d_{free}) gibt, 2 mit Abstand 6, 4 mit Abstand 7 usw. Diese Wege findet man leicht im Trellis Bild 2.3.

2.2.1 Beispiele guter Faltungscodes

In Ermanglung einer geschlossenen Theorie wurden Faltungscodes durch Computer-Simulationen gefunden. Als Kriterium dient die freie Distanz d_{free} zwischen zwei beliebigen Code-Worten. Für die Constraint-Längen $K = 3$ bis $K = 9$ sind folgende Generatoren für $R = 1/2$ optimal. Die Abgriffe können jeweils auch rechts-links gespiegelt genommen werden.

K	Code-Vektoren	d_{free}	Anwendung
3	(101, 111)	5	
4	(1011, 1111)	6	
5	(10111, 11001)	7	GSM
6	(101111, 110101)	8	
7	(1001111, 1101101)	10	DAB, DVB
8	(10011111, 11100101)	10	
9	(100011101, 110101111)	12	
7	(1001111, 1010111, 1101101)	15	($R = 1/3$)

2.2.2 Katastrophen-Codes

Ein katastrophaler Fehler ist dann gegeben, wenn eine endliche Anzahl Fehler in Code-Symbolen auf eine unendliche Anzahl falsch dekodierter Daten-Bits führt. Bei einem Faltungscod mit $K = 3$ tritt dies z.B. auf, wenn die Code-Vektoren zu (101, 110) gewählt werden. Die Generator-Polynome lauten dann:

$$g_1(D) = 1 + D^2; \quad g_2(D) = 1 + D \quad (2.8)$$

Charakteristisch für Katastrophen-Codes ist, daß die Generator-Polynome einen gemeinsamen Faktor haben, hier $1 + D$, da $(1 + D) \otimes (1 + D) = 1 + D^2$ ergibt.

2.3 Soft Decision Viterbi-Decodierung

Die Berechnung der Metrik der Viterbi-Decodierung, siehe Bild 2.1, geht prinzipiell in gleicher Weise vor sich, wenn der Entscheider die Bits nicht „hart“ entscheidet, sondern Zwischenwerte liefert, aus denen hervorgeht, wie „sicher“ eine Entscheidung ist. Hierdurch werden Informationen über den Zustand des Übertragungskanals ausgewertet.

2.3.1 Soft Decision

Eine harte Entscheidung liefert am Beispiel einer BPSK eine „0“ oder „1“ bzw. „-1“ oder „1“ und damit eine Vorzeichen-Entscheidung. Eine weiche Entscheidung (*soft decision*) liefert außer dem Vorzeichen zusätzlich eine Information, wie weit der entschiedene Wert von der Schwelle entfernt liegt. Im Falle einer Störung auf dem Übertragungsweg durch Gauß-förmiges weißes Rauschen ergeben sich Verhältnisse gemäß Bild 2.4, wenn mit 3 Bit aufgelöst wird.

Eine 3 Bit Auflösung bringt eine Verbesserung im Signal / Geräusch Verhältnis von 2 dB. Eine ∞ feine Auflösung ergäbe 2.2 dB. Daher geht man nicht über 3 Bit hinaus. Für die Berechnung des Viterbi-Algorithmus sind nun „real“ statt „integer“ Zahlen zu verwenden.

In der Darstellung Bild 2.4 kann das 1. Bit auch als Vorzeichen-Bit angesehen werden.

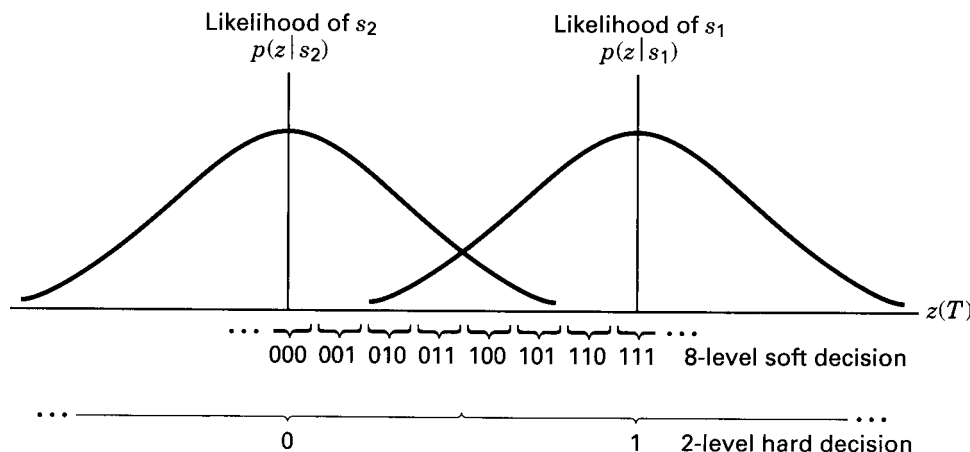


Bild 2.4: Harte und 3 Bit Soft Entscheidung bei einer BPSK Übertragung

2.3.2 Soft Decision bei Viterbi

Bei einem Faltungs-Code der Rate $R_c = 1/2$ werden pro Zeiteinheit 2 Symbole vom Demodulator an den Viterbi-Decoder geliefert. Bei einer harten Entscheidung sind das 00, 01, 11, 10. Diese können als Ecken eines Quadrates dargestellt werden, Bild 2.5.

Bei einer weichen Entscheidung mit 3 Bit ergeben sich damit insgesamt 64 Punkte in der Fläche des Quadrates. Zur Vereinfachung der Berechnung der Metrik werden diese Punkte in beiden Richtungen von 0 bis 7 numeriert.

Es wird nun angenommen, daß das erste Bit der Nachricht eine 1 ist und damit das gesendete Dibit 1,1 (resp. 7,7) sein soll, aber die empfangenen Symbole die Werte 5,4 haben. Im Viterbi-Algorithmus wird nun berechnet, wie weit der Empfangspunkt von den nominellen Sollpunkten (0,0 bzw. 7,7) entfernt liegt. Diese Werte werden dann der Metrik für den Viterbi-Decoder zugrunde gelegt, Bild 2.6.

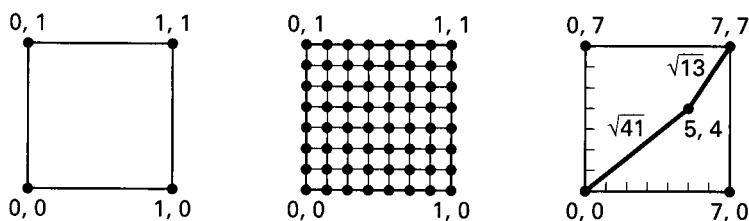


Bild 2.5: Harte und weiche Entscheidung der Dibits

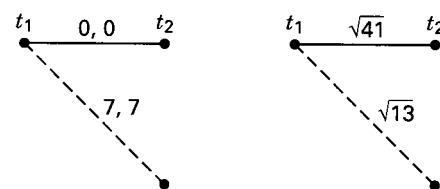


Bild 2.6: Beginn des Trellis: Encoder und Viterbi-Decoder

Damit keine Wurzeln berechnet werden müssen, können in der Metrik auch die Quadrate der Entfernungen verwendet werden. Insgesamt sieht man, daß bei der Viterbi-Decodierung eine Soft Decision nur einen geringen Mehraufwand bedeutet.

2.4 Punktierte Codes

2.4.1 Erasure

Die Soft Decision kann noch insofern verbessert werden, daß abhängig von der Größe der Störung im Kanal entschieden wird, ob überhaupt noch ein empfangbares Symbol vorliegt. In einem solchen Fall kann der Empfänger ein Flag setzen, das dem Decoder angibt, daß das betroffene Symbol „ausradiert“ (*erased*) wurde. Der Decoder behandelt eine Erasure ε als „ich weiß nicht“, was günstiger auswertbar ist als eine Fehlentscheidung. Die Anzahl der korrigierbaren Erasures ε beträgt:

$$\varepsilon \leq d_{\text{free}} + 1 \tag{2.9}$$

Treten t Fehler und ε Erasures gemeinsam auf, gilt die Bedingung:

$$d_{\text{free}} \geq 2t + \varepsilon + 1 \tag{2.10}$$

2.4.2 Punktierter Code bei DVB-S

Die Punktierung beruht auf der Korrekturmöglichkeit von Erasures. Um Daten-Rate zu sparen, werden bei der Punktierung sendeseitig nach einem vorgegebenen Schema Bits „gelöscht“ bzw. erst gar nicht übertragen. Dem Empfänger sind diese Erasure-Positionen bekannt und er kann dort die entsprechenden Flags setzen.

Bei DVB-S (DVB Satellitenstandard) wird ein Faltungscodierer mit Constraint Länge $K = 7$ verwendet, Bild 2.7.¹

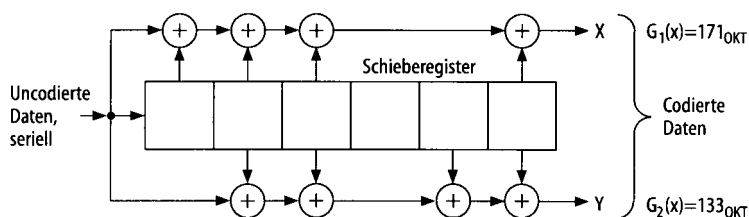


Bild 2.7: Faltungscodierer bei DVB-S

Je nach Modus (d.h. Kanaleigenschaften) kann mit Code-Rate $R_c = 1/2$ (schlechtester Kanal, ohne Punktierung) bis $R_c = 7/8$ (bester Kanal) übertragen werden. Welche Bits bei der Punktierung weggelassen werden und wie diese für die Übertragung umsortiert werden, ist in Bild 2.8 dargestellt.

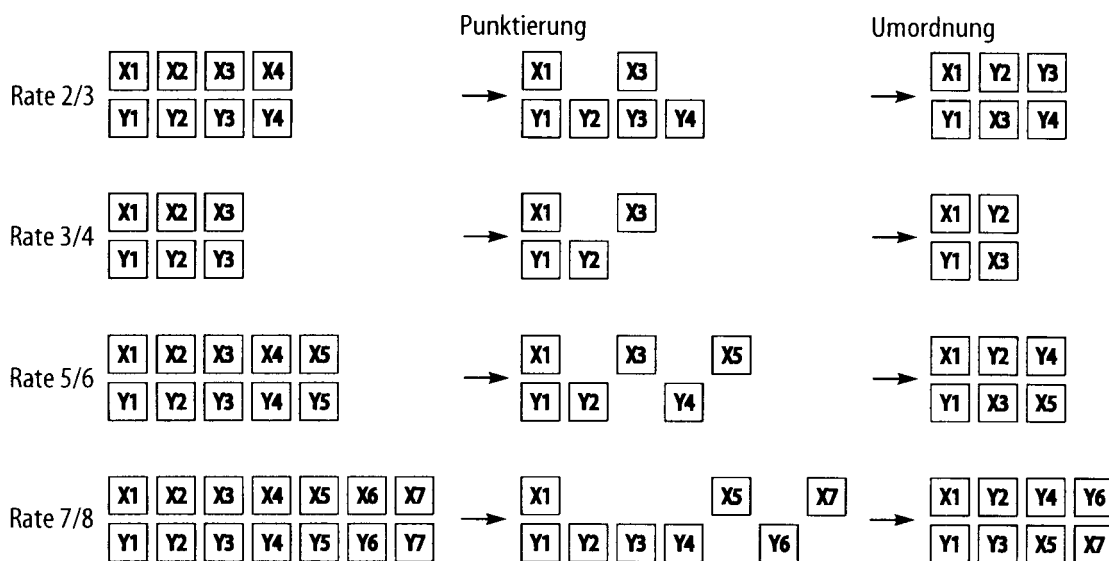


Bild 2.8: Punktierungs-Schema bei DVB-S

Beim Faltungscodierer in Bild 2.7 ist kein ausgangsseitiger Umschalter gezeichnet wie z.B. in Bild 1.1. Dieser ist tatsächlich entbehrlich, wenn die beiden Ausgänge auf den I bzw. Q Eingang eines digitalen Modulators gegeben werden. Damit beide Zweige in jedem Fall die gleiche Datenrate erhalten, wurde in Bild 2.8 nach der Punktierung entsprechend umsortiert.

¹Hierbei sind der obere und der untere Pfad gegenüber Bild 1.1 vertauscht.

3 Verkettete Codes

Funk-Kanäle führen ohne Codierung auf Fehler-Raten $10^{-1} < BER < 10^{-2}$. Faltungen-Codes mit Soft Decision und $R_c = 1/2$ können die Fehler-Raten auf $10^{-3} < BER < 10^{-4}$ erniedrigen. Wenn in einer Anwendung $10^{-6} < BER < 10^{-11}$ gefordert ist, geht dies nur dadurch, daß eine Verkettung (*Concatination*) von Codes angewendet wird. Man verwendet als inneren Code einen Faltungs-Code (für das Grobe) und als äußeren Code einen Block-Code (z.B. Reed-Solomon-Code), der mit einer Code-Rate $R_c > 0.9$ auskommt. Damit erhält man eine Struktur für das Übertragungs-System gemäß Bild 3.1.

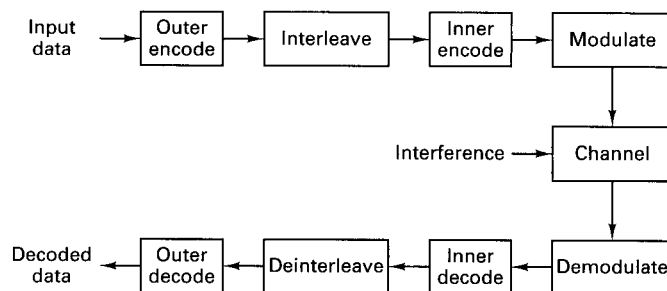


Bild 3.1: Übertragungs-System mit Code-Verkettung

Da die Fehler bei der Faltungs-Codierung / Viterbi-Decodierung burstartig auftreten, werden diese mit Hilfe von Interleaver und Deinterleaver in Einzelfehler aufgelöst, was zum korrekten Arbeiten des Block-Dekodierers erforderlich ist.

3.1 Faltungen-Interleaver

Ein Interleaver kann als Matrix ausgeführt werden, in die zeilenweise eingeschrieben und spaltenweise ausgelesen wird. Die sich durch das Interleaven ergebende Verzögerung kann halbiert werden, wenn ein Faltungen-Interleaver zur Anwendung kommt. Dieser besteht aus Verzögerungs-Registern unterschiedlicher Länge $k \cdot j$; $k = 0, 1, 2, \dots, l$, die über einen Multiplexer im Bit-Takt nach einander mit dem Eingang verbunden werden. Beim De-Interleaver wird eine dazu „punktsymmetrische“ Struktur angewendet, Bild 3.2.

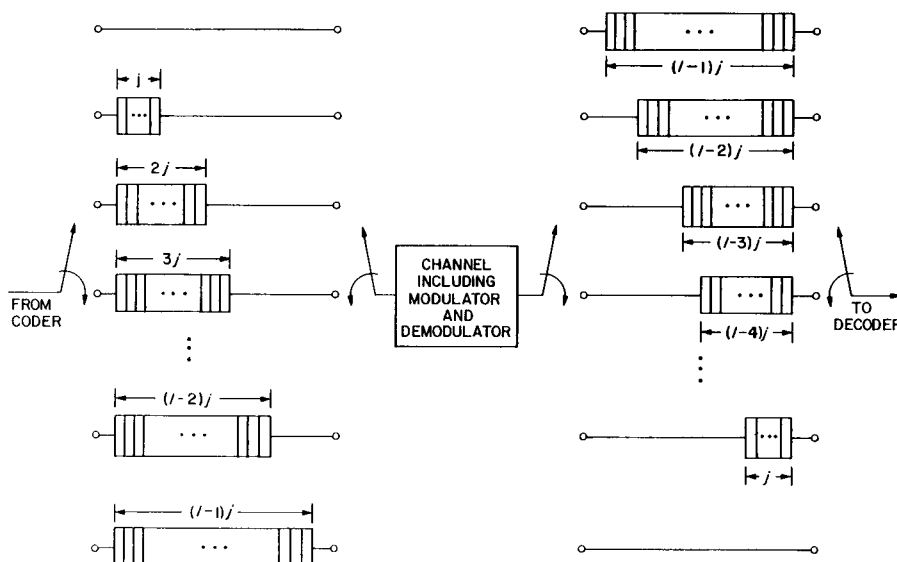


Bild 3.2: Faltungen-Interleaver und -Deinterleaver

Literatur

- [1] Sklar, B.: *Digital Communications, Fundamentals and Applications*, Prentice Hall, 1988
- [2] Sklar, B.: *Digital Communications, Fundamentals and Applications*, Prentice Hall, 2nd ed. 2001
- [3] Blahut, R.E.: *Theory and Practice of Error Control Codes*, Addison Wesley, 1983
- [4] Blahut, R.E.: *Digital Transmission of Information*, Addison Wesley, 1990
- [5] Haykin, S.: *Communication Systems*, Wiley, 2001
- [6] Simon, M.K.; Hinedi, S.M.; Lindsey, W.C.: *Digital Communication Techniques, Signal Design and Detection*, Prentice Hall, 1994
- [7] Simon, M.K.; Omura, J.K.; Scholtz, R.A.; Levitt, B.K.: *Spread Spectrum Communications Vol. 1*, Computer Science Press, 1985
- [8] Reimers, U.: *Digitale Fernsehtechnik*, Springer 1995
- [9] Sweeney, P.: *Codierung zur Fehlererkennung und Fehlerkorrektur*, Hanser & PrenticeHall, 1992
- [10] Proakis, J.G.: *Digital Communications*, 2nd ed. 1989
- [11] Proakis, J.G.; Salehi, M.: *Communication Systems Engineering*, Prentice Hall, 1994
- [12] Ziemer, R.E.; Peterson, R.L.: *Introduction to Digital Communication*, McMillan, 1992
- [13] Taub, H.; Schilling, D.L.: *Principles of Communication Systems*, 2nd ed. 1989
- [14] Lee, E.A.; Messerschmitt, D.G.: *Digital Communication*, Kluver 1988