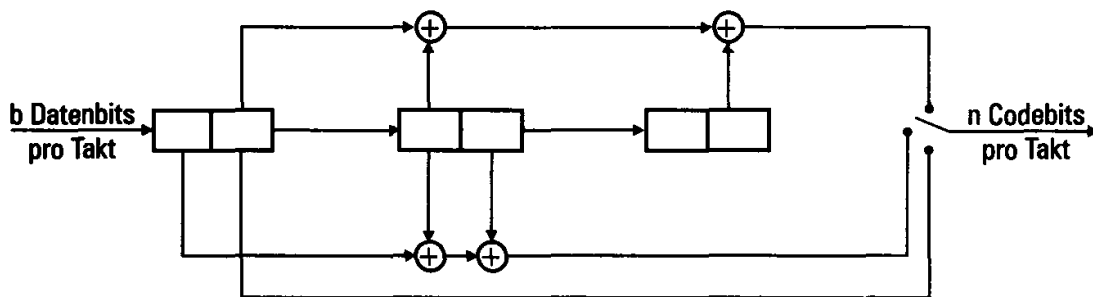


## Kanalcodierung 2

### Faltungscodierung

Ein Faltungs-Encoder kann als Schieberegister mit Verknüpfungslogik angesehen werden. Jeweils  $k > 1$  Bit werden zu einem Symbol zusammengefaßt. Diese Symbole werden anschließend getaktet in ein Schieberegister der Länge  $L = b \times k$  eingegeben (Bild ). Die Größe  $b = L/k$  heißt Eindringtiefe und  $L$  heißt Einflußlänge des Encoders. Die Größe  $r = k/n$  gibt die Coderate an.

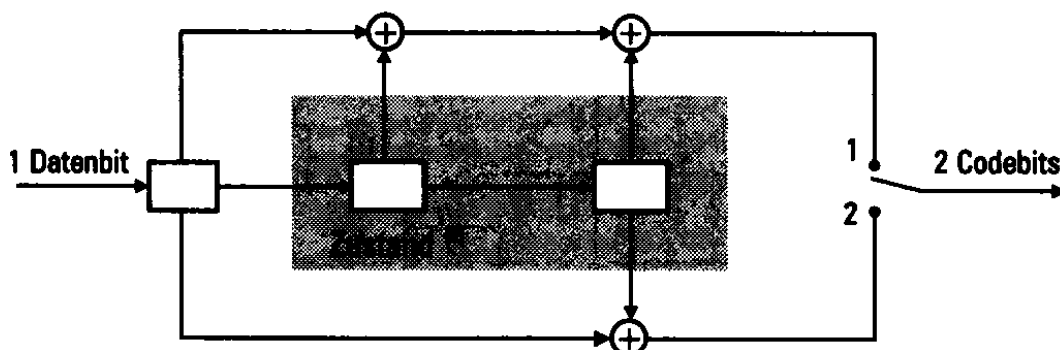


$k = 2$	Datenbits pro Takt	$L = b \cdot k = 6$	Einflußlänge
$n = 3$	Codebits pro Takt	$r = k/n = 2/3$	Coderate
$b = 3$	Eindringtiefe		

#### Beispiel für einen Faltungs-Encoder

Wir setzen im folgenden  $k = 1$  und machen uns die Faltungscodierung an dem im Bild skizzierten Encoder klar. Stehen zu Beginn Nullen in den Schieberegister-Zellen gibt der Encoder auf

die Datenbits ..... 0    1    0    1    1    0  
 die Codebits ..... 01   00   10   10   11   00  $\rightarrow$  t        aus.



$k = 1$	$L = 3$	*) Zustandsnummer: 1. Bit rechts 2. Bit links
$n = 2$	$r = 1/2$	
$b = 3$	Startbelegung: 0 0 0	

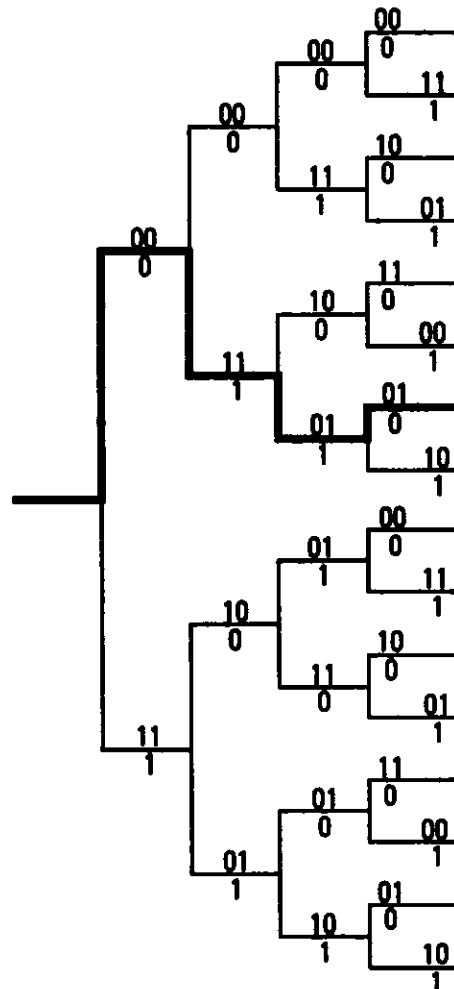
#### Standardbeispiel für einen Faltungs-Encoder

## Darstellungen des Encoders

Zunächst einmal kann der Encoder in Form eines *Codebaumes* (Bild ) dargestellt werden.

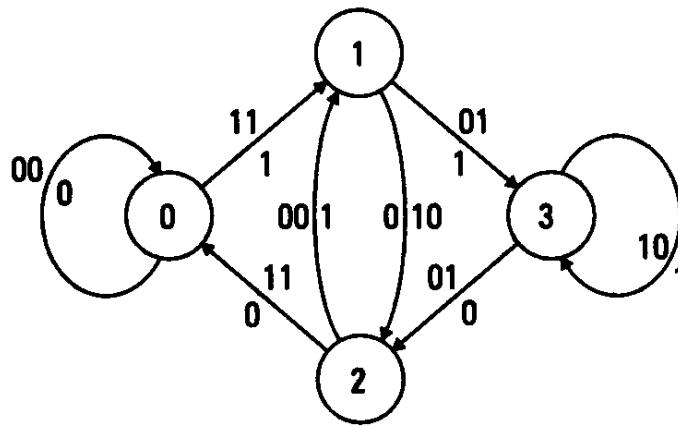
Ist das Datenbit eine 0, wird nach oben gegangen, ist es eine 1 wird nach unten gegangen. Die Datenbits sind jeweils unter, die auszugebenden Codebits (auch *Chips* genannt) sind über dem Ast angegeben. Eine bestimmte Bitfolge führt auf genau einen *Pfad durch den Baum*. Der Baum wächst exponentiell mit dem Takt.

Die zweite Darstellung des Encoders ist sein *Zustandsdiagramm*: Es gibt  $2^{b-1}$  Zustände, die durch den Inhalt der  $b - 1$  rechts stehenden Schieberegisterzellen bestimmt sind. Die Zustandnummer ergibt sich als Binärzahl aus dem von rechts nach links gelesenen Inhalt dieser  $b - 1$  Zellen. Der Inhalt der links stehenden Schieberegisterzelle (das ist das aktuelle Datenbit) bestimmt den Übergang zum nächsten Zustand.



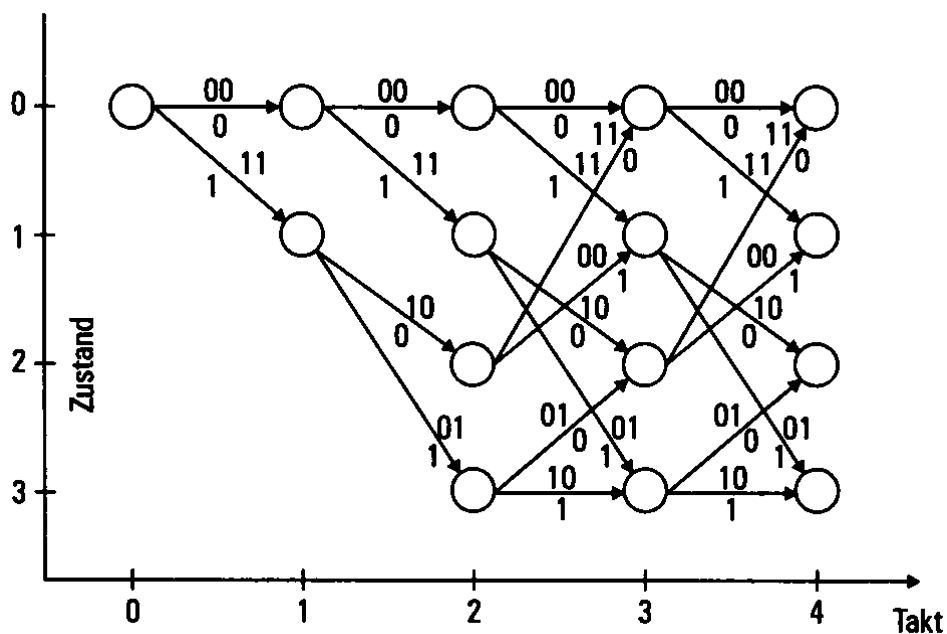
Codebaum für den Encoder des Standardbeispiels

Datenbits und Codebits sind an den Übergangspfeilen des Zustandsdiagramms angegeben, wobei die Codebits hier und in den folgenden Bildern in ihrer "natürlichen" Reihenfolge (1. Bit links, 2. Bit rechts) notiert sind.



Zustandsdiagramm für den Encoder des Standardbeispiels

Eine Kombination von Codebaum und Zustandsdiagramm stellt das Trellisdiagramm dar:



Trellisdiagramm für den Encoder des Standardbeispiels

Darin werden für jeden Takt die  $2^{b-1}$  Zustände und die Übergangspfeile zwischen diesen dargestellt. Ist das nächste Datenbit eine 0 wird der obere, ist es eine 1 der untere Pfeil, der aus dem aktuellen Zustand herausführt, zum Übergang in den nächsten Zustand genutzt. Auch hier können an den Pfeilen sowohl das Datenbit als auch die auszugebenden Codebits abgelesen werden.

## Decoder

Während der Block-Decoder  $n$  Codebits in  $k$  Datenbits zurückverwandelt, ist die Lage beim Faltungs-Decoder etwas komplizierter, da aufgrund des Codiervorgangs zusätzlich zum aktuellen Datenbit vorhergehende Datenbits Einfluß auf die auszugebenden Codebits haben. Empfängerseitig muß also zu einer vom Demodulator an den Decoder abgegebenen Codebitfolge  $y$  (Empfangsbitfolge) die

wahrscheinlichste Sendebitfolge gefunden werden. Die Anzahl der möglichen codierten Sendebitfolgen  $x$  wächst allerdings exponentiell mit der Taktzeit (Codebaum). Wir nehmen für die folgende Betrachtung an, daß Sendebitfolge und Empfangsbitfolge dieselbe endliche Länge  $N$  haben. Die Entscheidung kann entweder aus Sicht des Empfängers oder aus Sicht des Senders getroffen werden:

### Sicht des Empfängers:

Die Entscheidung fällt für die Sendebitfolge  $x^*$ , die nach Empfang der Empfangsbitfolge die höchste Wahrscheinlichkeit hat.

$$P(x^*|y) = \max_x \{P(x|y)\}$$

Die Entscheidung fällt also nach dem *maximum a posteriori (MAP-) Kriterium*. Der Decoder wird als *MAP-Decoder* bezeichnet.

### Sicht des Senders:

Die Entscheidung fällt nach dem *Maximum-Likelihood-Verfahren*. D.h. für die gegebene Empfangsbitfolge  $y$  wird diejenige Sendebitfolge  $x^*$  bestimmt, für die die bedingte Dichte  $f(y|x^*)$  maximal wird:

$$f(y|x^*) = \max_x \{f(y|x)\}$$

Der Decoder heißt *ML-Decoder*.

Im allgemeinen unterscheiden sich MAP- und ML-Decoder. Nimmt man aber an, daß alle möglichen Sendebitfolgen gleichwahrscheinlich sind (was bei vernünftiger Quellencodierung zumindest näherungsweise gilt), folgt über die Formel von Bayes, daß beide Verfahren identisch sind.

Liefert der Demodulator an den Decoder Bitfolgen, arbeitet der Decoder mit Hard-Decision. Es bietet sich in diesem Fall an, die Entscheidung über den Hammingabstand herbeizuführen. Der Abstand zwischen der Sendebitfolge  $x$  und der am Demodulatorausgang vorliegenden Empfangsbitfolge  $y$  (beide haben die Länge  $N$ ) ist bei  $m$  Bitfehlern einfach  $d(y,x) = m$ . Die Entscheidung nach dem minimalen Hammingabstand ist der Entscheidung nach dem ML-Verfahren gleich: Für  $m$  Bitfehler in einer Bitfolge der Länge  $N$  gilt nämlich unter der Annahme eines symmetrischen Binärkanals

$$P(y|x) = p_b^m (1 - p_b)^{N-m}$$

wobei  $p_b$  die Bitfehlerwahrscheinlichkeit ist. Der Logarithmus ist monoton nicht fallend, also kann dafür auch

$$\begin{aligned} \ln\{P(x|y)\} &= m \cdot \ln\{p_b\} + (N - m) \cdot \ln\{1 - p_b\} \\ &= m \cdot \ln\left\{\frac{p_b}{1 - p_b}\right\} + N \cdot \ln\{1 - p_b\} \end{aligned}$$

maximiert werden. Für  $p_b < 0,5$  (was sicher eine sinnvolle Annahme ist), sind beide Summanden auf der rechten Seite  $< 0$  und, da  $N$  eine Konstante ist, wird die Summe

für minimales  $m$  maximal.

Wird die Übertragung durch additives weißes Gaußsches Rauschen der Rauschleistungsdichte  $N_0$  gestört, ist die Verteilungsfunktion der Empfangsbits  $\mathbf{y}$  zunächst

durch 
$$f(y_n | x_n) = \frac{1}{\sqrt{2pN_0}} \exp \left\{ -\frac{|y_n - x_n|^2}{2N_0} \right\}$$

gegeben. D.h. der Demodulator muß an dieser Stelle noch nicht unbedingt "hart" auf Bits entscheiden: Er kann dem Decoder die Möglichkeit zur **Soft-Decision** geben. Werden die Bits nämlich (wie im symmetrischen Binärkanal) stochastisch unabhängig übertragen, gilt

$$f(\mathbf{y} | \mathbf{x}) = \prod_{n=1}^N f(y_n | x_n)$$

und für die Likelihoodfunktion ergibt sich hier

$$\begin{aligned} \ln\{f(\mathbf{y} | \mathbf{x})\} &= \sum_{n=1}^N \ln\{f(y_n | x_n)\} \\ &= \sum_{n=1}^N \left[ -\frac{1}{2} \ln\{2pN_0\} - \frac{1}{2N_0} |y_n - x_n|^2 \right] \\ &= -\frac{1}{2N_0} \sum_{n=1}^N |y_n - x_n|^2 + \text{const.} \end{aligned}$$

Der ML-Decoder entscheidet also für die Sendebitfolge  $\mathbf{x}$ , die der Empfangsbitfolge  $\mathbf{y}$  nach dem Euklid'schen Abstand am nächsten kommt. Gegenüber Hard-Decision kann man bei Soft-Decision mit einem um ca. 2, 5 dB höheren Codegewinn rechnen.

### Der Viterbi-Algorithmus

Faltungs-Decoder, die nach dem sequentiellen ML-Prinzip arbeiten, müssen also nur die Hamming- (Hard-Decision) oder die Euklid'schen Abstände (Soft-Decision) der Empfangsbitfolge zu allen möglichen Sendebitfolgen bestimmen und unter diesen Abständen den minimalen identifizieren. Für große Folgenlängen  $N$  ist das hier beschriebene Vorgehen nicht möglich, da die Anzahl der Vergleichspfade exponentiell mit  $N$  wächst.

So waren Faltungscode praktisch nicht einsetzbar, bis 1967 Andrew J. Viterbi einen Algorithmus veröffentlichte, mit dem in jedem Takt nicht mehr  $2^N$  sondern nur noch  $2^{L-1}$  ( $L$  ist die Einflußlänge des Encoders) Vergleiche ausgeführt werden müssen.

Wir studieren den Viterbi-Algorithmus wieder an dem bereits dargestellten Beispiel des Standardencoders und dem zugehörigen Trellisdiagramm, in dem auffällt, daß ab dem dritten Takt in jeden Zustand jeweils zwei Pfade münden.

Der Einfachheit halber befassen wir uns zunächst mit Hard-Decision- Decodierung. Der Viterbi-Algorithmus läßt sich einfach (s.u.) auf die Soft- Decision-Decodierung erweitern.

Wir kehren zum Trellis zurück und schreiben in die Knoten zunächst bis zum zweiten Takt den Hammingabstand zwischen Empfangsbit- und (möglicher)

Sendebitfolge. Ab dem dritten Takt münden in jeden Zustand zwei Pfade, von denen wir aber nur denjenigen weiterverfolgen, der gegenüber der Empfangsbitfolge den kleineren Hammingabstand hat. Zu diesem Vorgehen müssen zunächst zwei Bemerkungen gemacht werden:

1. Der Pfad, der beim Einmünden in einen Zustand gegenüber der Empfangsbitfolge den größeren Hamming-Abstand hat, wird in der Folge nie eine Chance haben, Teil des optimalen Decodier-Pfads zu werden, da der Hamming-Abstand eine mit dem Takt monoton nichtfallende Funktion ist.
2. Besitzen die in einem Zustand einlaufenden Pfade denselben Hammingabstand zur Empfangsbitfolge, wird die Entscheidung für einen der beiden durch das Los getroffen (oder beide Pfade werden weiterverfolgt).

Bild 6.3-1(a) zeigt das aus der Encodierung der Datenfolge 0 1 1 0 1 bei Start des Encoders (Bild 6-2) im Nullzustand sich am Decoder nach "Ausdünnung" durch den Viterbi-Algorithmus ergebende Trellisdiagramm, wenn kein Übertragungsfehler aufgetreten ist. Die Sendebitfolge 0 0 11 01 01 0 0 ist dann identisch mit der oben im Bild angegebenen Empfangsbitfolge. Bild 6.3-1(b) zeigt das Vorgehen bei der Decodierung, wenn einfach vom fünften Takt aus der Pfad mit dem minimalen Hamming-Abstand bis zum nullten Takt zurückverfolgt wird. Die Bilder 6.3-1(c) und 6.3-1(d) zeigen das Vorgehen, wenn das erste und das fünfte Bit der Sendebitfolge falsch empfangen wurden. Der Decoder korrigiert in diesem Fall die Übertragungsfehler.

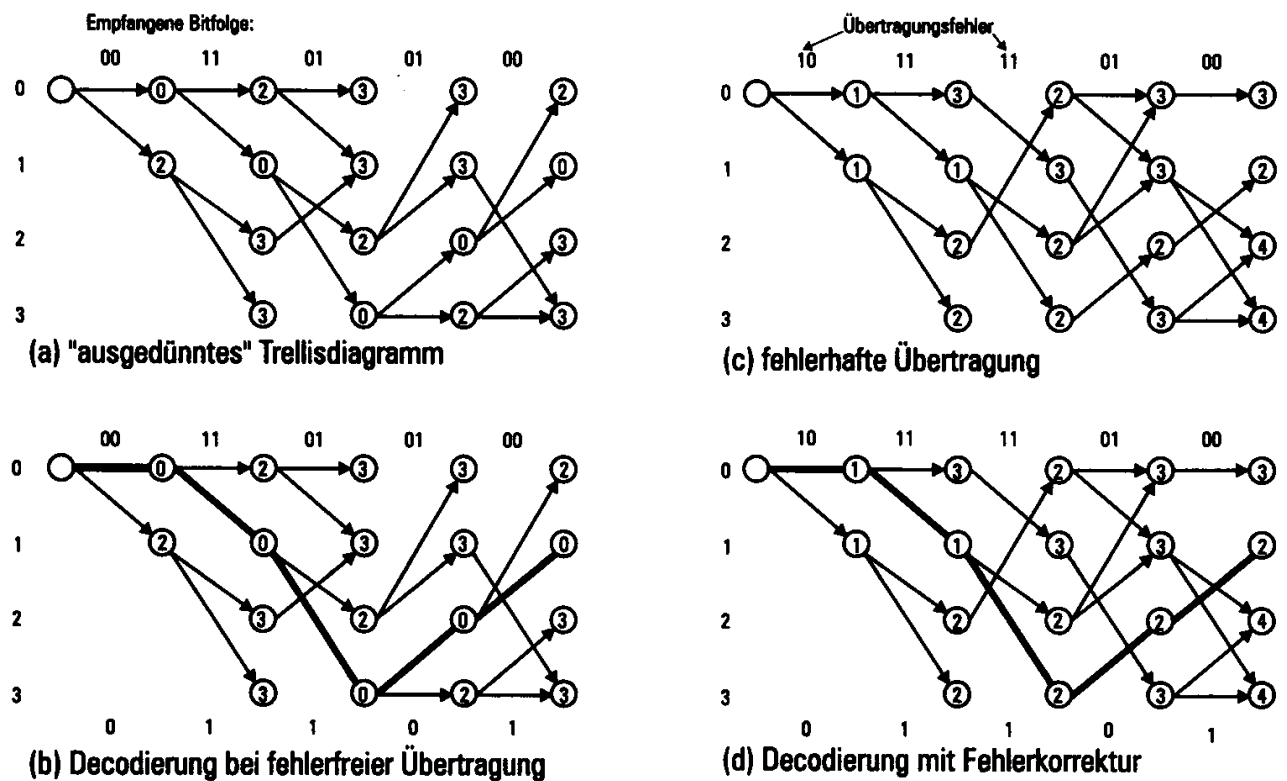


Bild 6.3-1: Viterbi-Algorithmus für ein Encodierungsbeispiel nach Bild 6-2

Der Faltungs-Encoder erzeugt im Prinzip eine beliebig lange Sendebitfolge. Der unmotivierter Abbruch des Decodierverfahrens, so wie wir es in unserem Beispiel durchgeführt haben, zieht eine erhöhte Fehleranfälligkeit am Ende der Empfangsfolge nach sich. In der Praxis wird daher häufig so vorgegangen, daß durch das Einfügen von Nullen in den senderseitigen Datenstrom nach einer festen Anzahl von Informationsbits der Encoder in den Nullzustand gebracht wird. Von dort aus wird dann auch die Encodierung der nächsten Datenbitfolge begonnen. Für das Trellisdiagramm bedeutet dies, daß (wenn am Ende kein Fehler auftritt) alle Pfade im Nullzustand enden. Für unser Beispiel ist das in Bild 6.3-2 dargestellt.

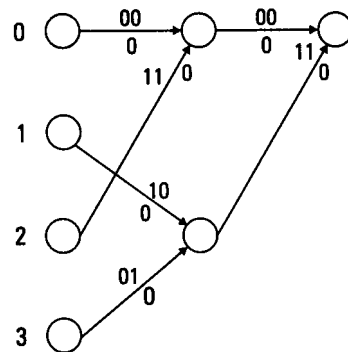


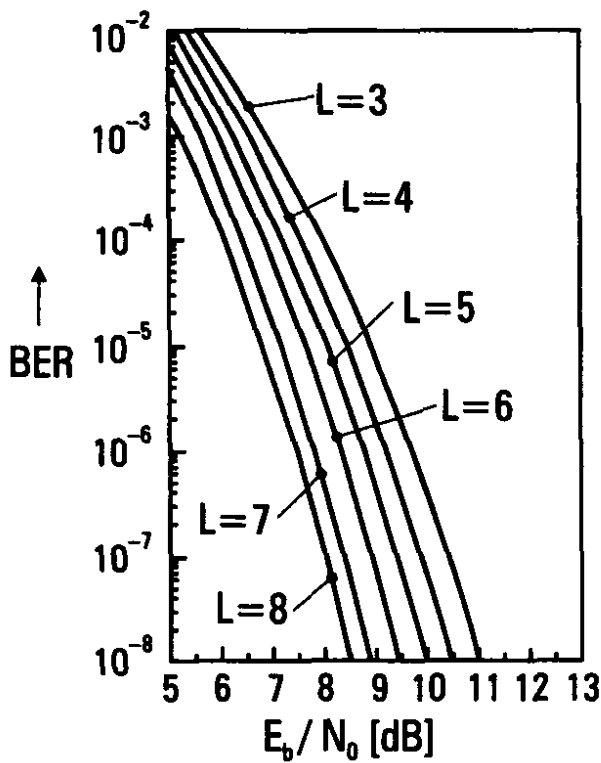
Bild 6.3-2: Auslaufen des Encoders (bzw. des Trellisdiagramms) in den Nullzustand

Für den Fall der Soft-Decision-Decodierung werden statt der Bits die Matched-Filterausgänge  $y$  direkt an den Decoder gegeben, der dann statt des Hamming-Abstands den Euklid'schen Abstand auswertet.

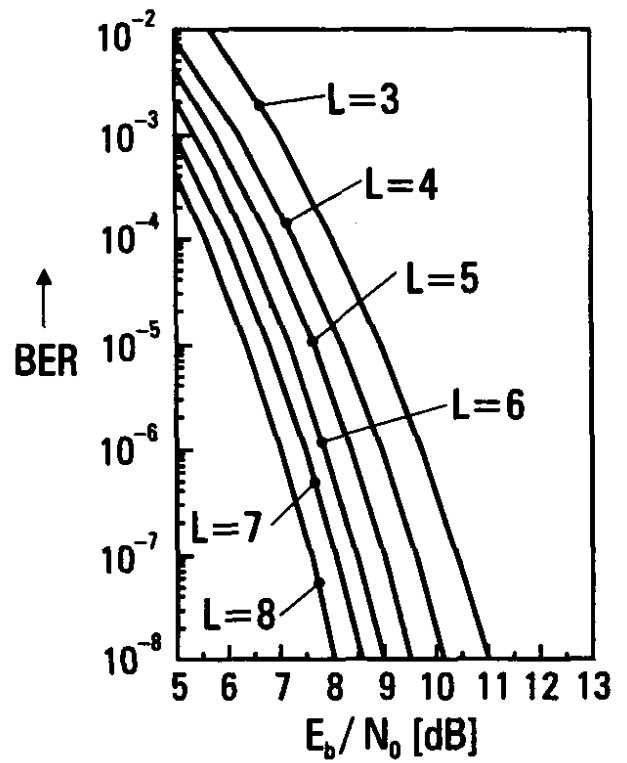
### Korrektoreigenschaften von Faltungscodes

Die Leistungsfähigkeit eines Faltungscodes bei gleichverteilten Bitfehlern hängt - wie bei den Blockcodes - von der Minimaldistanz zwischen Empfangsbitfolge und Sendebitfolge ab. Der Zusammenhang ist aber für Faltungscodes nicht so einfach wie für Blockcodes, insbesondere kann die Restfehlerwahrscheinlichkeit nicht mehr über die Binomialverteilung abgeschätzt werden. Die Restfehlerwahrscheinlichkeit hängt bei Faltungscodes nämlich nicht nur von der Anzahl sondern auch von der Position der Fehler im Bitstrom ab.

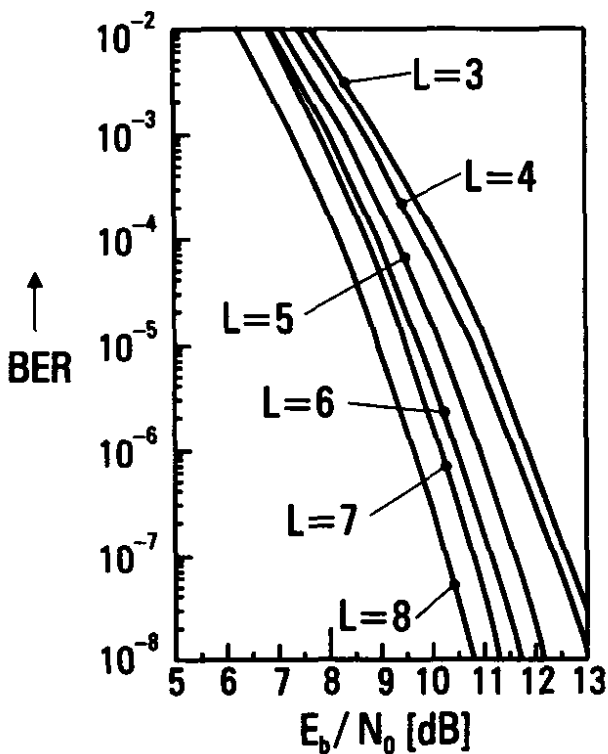
Die Bilder 6.4-1(a) bis 6.4-1(d) geben die Bitfehlerwahrscheinlichkeiten am Decoderausgang für die besten bekannten Faltungscodes mit den Raten  $1/2$  und  $1/3$  sowohl für Hard- als auch für Soft-Decision-Decodierung wieder. Dabei wurden BPSK-Modulation und ein AWGN- (Additive White Gaussian Noise) Kanal zugrunde gelegt. Der zusätzliche Codiergewinn durch Soft-Decision-Decodierung liegt bei ca. 2, 5 dB. Die Leistungsfähigkeit von Faltungscodes steigt mit wachsender Einflußlänge  $L$  des Encoders.



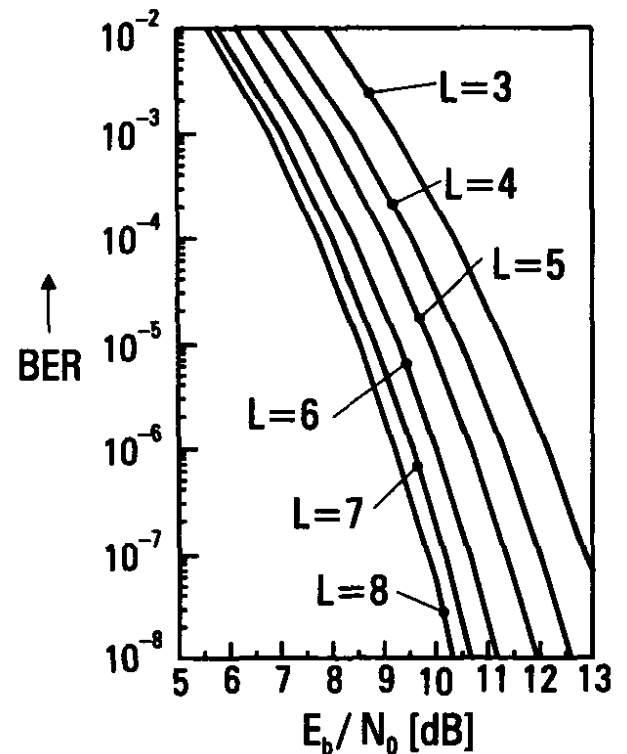
(a) Soft-Decision,  $r=1/2$



(b) Soft-Decision,  $r=1/3$



(c) Hard-Decision,  $r=1/2$



(d) Hard-Decision,  $r=1/3$

Bild 6.4-1: Leistungsfähigkeit von Faltungscodes, BPSK-Modulation, AWGN-Kanal, Bitfehlerraten am Decoderausgang