

Fehlerschutz-Codierung 3: Turbo-Code

Inhaltsverzeichnis

1 Turbo-Codierung	1
1.1 Zum Prinzip der Turbo-Codierung	1
1.2 Zur Entscheidungs-Theorie	2
1.2.1 À priori Wahrscheinlichkeit	2
1.2.2 À posteriori Wahrscheinlichkeit	3
1.2.3 Das Theorem von Bayes	4
1.2.4 Das Maximum à posteriori Kriterium	5
1.2.5 Maximum Likelihood Kriterium	6
1.2.6 Binär-Entscheidung nach Maximum Likelihood	6
1.2.7 Schwelle bei ungleichen à priori Wahrscheinlichkeiten	7
2 Turbo-Encoder	8
2.1 Struktur des Turbo-Encoders	8
2.2 Beispiel eines Systematischen Rekursiven Faltungs-Encoders	9
3 Turbo-Decoder	9
3.1 Struktur des Turbo-Decoders	10
3.2 Zuverlässigkeits-Information und L -Algebra	11
3.2.1 L -Algebra	11
3.3 Beispiel für Turbo-Decodierung	12

Abbildungsverzeichnis

1.1 Prinzip des Turbo-Encoders	1
1.2 Prinzip des Turbo-Decoders	1
1.3 Prinzip einer digitalen Binär-Übertragung im Basisband	2
1.4 Die bedingten Wahrscheinlichkeits-Dichten $p(z s_1)$ und $p(z s_2)$	3
1.5 Beispiel für das Bayes'sche Theorem	4
1.6 Harte und 3 Bit Soft Entscheidung bei einer BPSK Übertragung	5
1.7 Wahrscheinlichkeitsdichten bei unterschiedlicher à priori Wahrscheinlichkeit	7
2.1 Blockschaltbild eines Turbo-Encoders	8
2.2 Blockschaltbild eines RSC Encoders zur Erzeugung des Codes C_1 bzw. C_2	9
3.1 Blockschaltbild eines SISO Decoders und die Gewinnung der „extrinsic“ Information	10
3.2 Blockschaltbild eines Turbo Decoders. Die Decodierer sind mit $BCJR$ bezeichnet.	10
3.3 relative Häufigkeitsverteilungen der LLR's der Informations-Bits	12
3.4 relative Häufigkeitsverteilungen der „extrinsic“ LLR's	12
3.5 Mittelwerte, Varianz und Korrelation zwischen „extrinsic“ Information und dem systematischen Anteil der Empfangsdaten	12

Fehlerschutz-Codierung 3: Turbo-Code

Turbo-Codes wurden erst in der 1. Hälfte der '90er Jahre erfunden. Shannon hat bereits im Jahre '48 die theoretisch mögliche Grenze für eine fehlerfreie Übertragung angegeben. Als Bedingung nannte er, daß die Code-Worte lang sein müssen und daß sie statistisch von einander unabhängig sein sollen.

Block-Codes betrachten die Codierung als algebraisches Problem. Dies führt (bei langen Code-Worten) auf astronomisch hohe Zahlen für mögliche Code-Worte, aus denen die geeigneten ausgesucht werden müssen. Aus diesem Grunde mußten zusätzliche Randbedingungen angenommen werden, damit sich überhaupt geeignete Lösungen finden ließen.

Faltungs-Codes betrachten die Codierung als stochastisches Problem. Eine Unterteilung des Datenstromes in Blöcke ist daher nicht mehr notwendig, wird jedoch in der Praxis trotzdem angewendet (Tail-Bits). Bei Faltungs-Codes werden beim (empfangsseitigen) Viterbi-Decoder die Kanal-Zustände ebenfalls berücksichtigt, was zu einer „Soft Decision“ führt. Dadurch verbessert sich die erreichbare Fehlerrate bei gegebenem E_b/N_0 Verhältnis.

Als weitere Verbesserung stellte sich eine „Verkettung“ von Codes heraus. Hierbei erreicht man mit einer Kettenschaltung von 2 (relativ) kurzen Codes bessere Ergebnisse als mit einem einzelnen entsprechend langen Code.

Turbo-Codes wenden das Prinzip der Rückkopplung, das in der Nachrichtentechnik allgemein üblich ist, auf die De-Codierung an. Hierzu werden 2 parallele Encoder verwendet, d.h. es wird eine Parallelschaltung verwendet. Auf der Empfangsseite gibt es daher auch 2 parallele Decoder, die die Informationen über die Kanal-Zustände gegenseitig austauschen und so iterativ zu dem richtigen Ergebnis kommen. Die Bits, die der 2. Encoder erhält werden zuvor „pseudo-random“ interleaved. Damit ist sicher gestellt, daß zusammengehörige Informations-Bits vom Kanal statistisch unabhängig beeinflusst werden, womit die von Shannon erhobene Forderung (näherungsweise) erfüllt wird.

1 Turbo-Codierung

1.1 Zum Prinzip der Turbo-Codierung

Der Encoder besteht aus 3 parallelen Teilen, Bild 1.1. Die einlaufenden Daten werden zunächst direkt in das Code-Wort übernommen. Es handelt sich somit bei den Turbo-Codes um systematische Codes. Das Code-Wort enthält dann ein erstes Parity Wort, das mit Hilfe eines 1. Faltungs-Coders berechnet wird. Das 2. Parity Wort entsteht mit Hilfe eines 2. (identischen) Faltungs-Coders nachdem die Daten zuvor pseudo-random interleaved wurden. Dieser Interleaver ist ein sehr wichtiger Teil des Turbo-Encoders und beeinflusst ganz wesentlich die erreichbaren Fehlerraten einer Turbo-Codierung.

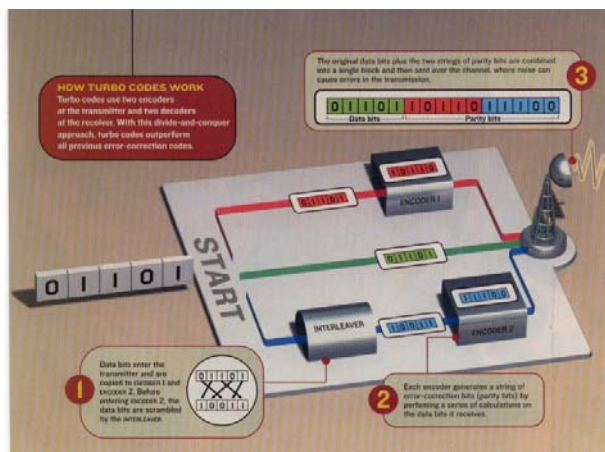


Bild 1.1: Prinzip des Turbo-Encoders

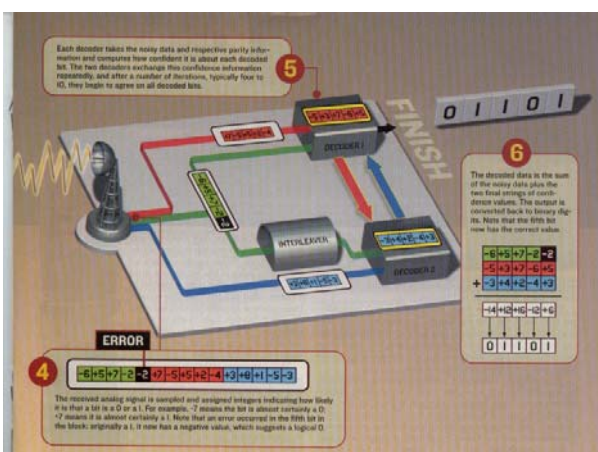


Bild 1.2: Prinzip des Turbo-Decoders

Die empfangenen Symbole werden zunächst „soft“ decodiert. Dadurch wird nicht sofort „hart“ auf +1

oder -1 entschieden, sondern es wird ein Intervall $[7, -7]$ (ggf. auch $[127, -127]$) benutzt, womit dann auch die Kanal-Eigenschaften erfaßt werden, Bild 1.2.

Der Decoder 1 vergleicht die Daten-Bits mit den Parity-Bits 1 und bestimmt daraus eine Zusatzinformation 1 (Zahlenbereich mit gleichem Intervall wie die „soft“ Bits), die dem Decoder 2 mitgeteilt wird. Der Decoder 2 vergleicht die Datenbits (nachdem diese interleaved wurden) mit den Parity-Bits 2. Daraus entsteht eine Zusatzinformation 2, die dem Decoder 1 mitgeteilt wird. Die Zusatzinformationen werden mit „extrinsic“ (äußerlich) „informations“ bezeichnet.

Unter Kenntnis der jeweiligen Zusatz-Informationen berechnen die Decoder im 1. Iterations-Schritt je eine verbesserte Zusatz-Information. Diese werden erneut ausgetauscht und damit ein 2. Iterations-Schritt durchgeführt. Nach ca. 4 – 10 Iterations-Schritten sind sich die beiden Decoder schließlich „einig“ über das Decodier-Ergebnis, das hier noch als „soft“ vorliegt. Anschließend wird dann eine „harte“ Bit-Entscheidung getroffen.

Die beiden Decodierer im Turbo-Decoder sind „soft in – soft out“ Decodierer. Sie unterscheiden sich daher von einem gewöhnlichen Viterbi-Decoder, der vom Typ „soft in – hard out“ ist.

1.2 Zur Entscheidungs-Theorie

Am Beispiel einer Binärübertragung (Binärsymbole ± 1) wird die Gewinnung des logarithmischen Wahrscheinlichkeits-Verhältnis (*Log-Likelihood Ratio*) dargestellt. Dieses wird verwendet um die „extrinsic“ Informationen bei der Turbo-Decodierung zu berücksichtigen.

Hierzu wird als Beispiel eine Übertragung von Binärsymbolen $s_1(t), s_2(t)$ im Basisband betrachtet. Die empfangenen (und durch Gauß verteiltes weißes Rauschen gestörten) Symbole $r(t) = s_i(t) + n(t); i = 1, 2$ werden mit Hilfe eines *matched Filters* optimal vom Rauschen befreit und liegen nach der Abtastung an den Symbolgrenzen T als $z(T) = a_i(T) + n_0(T)$ vor, Bild 1.3.

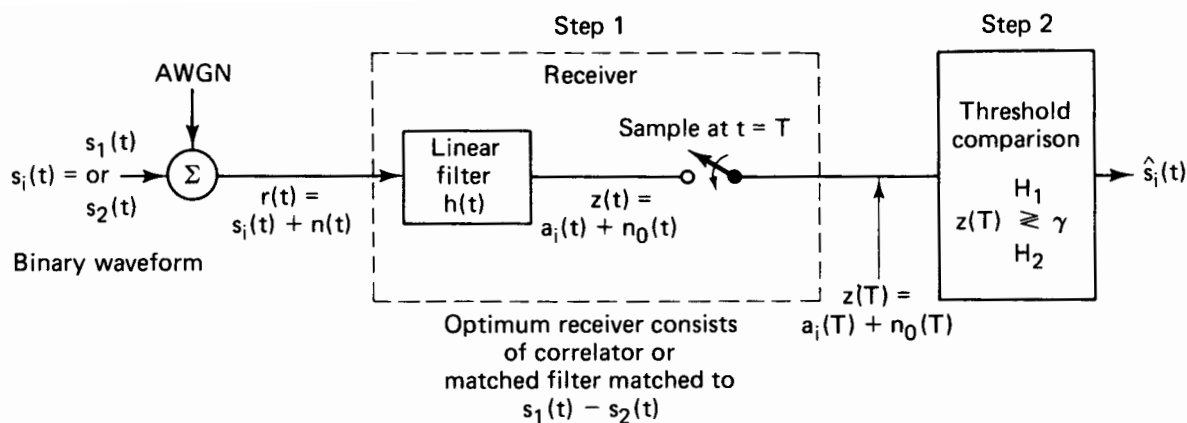


Bild 1.3: Prinzip einer digitalen Binär-Übertragung im Basisband

Es ist nunmehr ein Entscheider gesucht, der mit der größten Wahrscheinlichkeit (*ML Maximum Likelihood*) sich für das richtige Symbol entscheidet. Hierfür müssen die statistischen Eigenschaften der Symbole berücksichtigt werden.

1.2.1 À priori Wahrscheinlichkeit

Der Sender sendet die Symbole $s_1(t)$ und $s_2(t)$ mit den Wahrscheinlichkeiten $P(s_1), P(s_2)$. Dabei gilt

$$P(s_1) + P(s_2) = 1 \quad , \quad (1.1)$$

denn mit Sicherheit wird immer eines der Symbole gesendet.

Für eine Binärübertragung ist i.a. (*Regelfall*, bzw. es ist nichts genaueres bekannt)

$$P(s_1) = P(s_2) = \frac{1}{2} \quad , \quad (1.2)$$

d.h. logisch „0“ und logisch „1“ tritt bei der Übertragung gleich wahrscheinlich auf.¹ Diesen Fall nimmt man als Regelfall immer an, wenn nichts genaues über die tatsächliche Wahrscheinlichkeitsverteilung der Symbole bekannt ist.²

Es sei nun bekannt, daß der Sender

- $s_1(t)$ mit der Wahrscheinlichkeit $P(s_1)$
- $s_2(t)$ mit der Wahrscheinlichkeit $P(s_2)$

sendet. Hingegen sei nicht bekannt, was der Empfänger im Moment tatsächlich empfangen habe. Damit kann man nur eine *à priori* Aussage über das vermutlich empfangene Symbol machen.

Also sind $P(s_1), P(s_2)$ die **à priori Wahrscheinlichkeiten (APriP)** für die Symbole $s_1(t), s_2(t)$.

1.2.2 À posteriori Wahrscheinlichkeit

Nunmehr sei (zusätzlich) bekannt, welche Zahl

$$z(T) = a_i(T) + n_0(T) \quad \text{vereinfacht geschrieben: } z = a_i + n_0 \quad (1.3)$$

dem Entscheider des Empfängers vorliege. z ist daher eine „beobachtbare“ Random Variable. Die Frage ist nun, wie die *Entscheider-Schwelle* γ einzustellen ist, damit dieser mit der größten Wahrscheinlichkeit (*Maximum Likelihood*) die richtige Entscheidung trifft.

Aufgrund der Gauß'schen Amplitudenverteilung des Rauschens streuen die empfangenen Zahlenwerte um die Sollwerte a_1, a_2 jeweils gemäß einer Gauß'schen Wahrscheinlichkeits-Dichte-Funktion, Bild 1.4

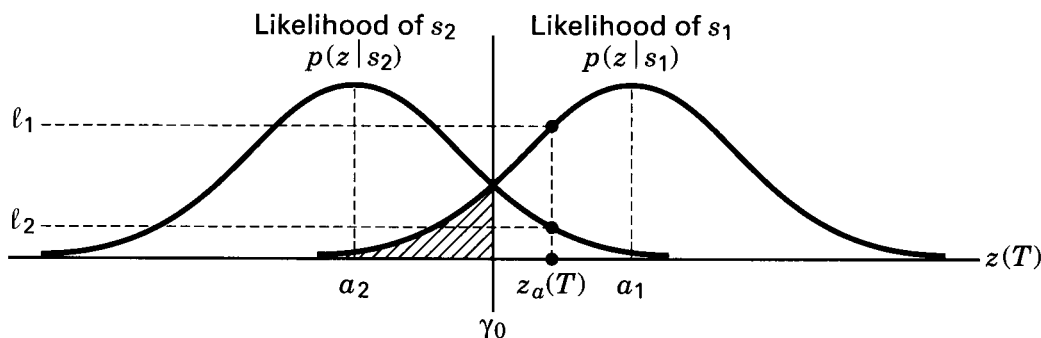


Bild 1.4: Die bedingten Wahrscheinlichkeits-Dichten $p(z|s_1)$ und $p(z|s_2)$

Die dadurch erhaltenen Wahrscheinlichkeits-Dichte-Funktionen werden **bedingte Wahrscheinlichkeits-Dichte-Funktionen (conditional pdf)** genannt:

- $p(z|s_1)$: pdf für den Fall daß s_1 gesendet (*Bedingung*) und z empfangen wurde.
- $p(z|s_2)$: pdf für den Fall daß s_2 gesendet (*Bedingung*) und z empfangen wurde.

Für die pdf des Gauß'schen Rauschens (mit der Standard-Abweichung = Effektivwert σ_0) und damit auch für die pdf's der um einen Gleichanteil a_1 bzw. a_2 (nämlich der Datenbits) angehobenen Rauschanteile gilt:

$$p(n_0) = \frac{1}{\sigma_0 \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{n_0}{\sigma_0} \right)^2} \quad (1.4)$$

$$p(z|s_1) = \frac{1}{\sigma_0 \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{z-a_1}{\sigma_0} \right)^2} \quad (1.5)$$

$$p(z|s_2) = \frac{1}{\sigma_0 \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{z-a_2}{\sigma_0} \right)^2} \quad (1.6)$$

¹Diesen Fall wird man z.B. für eine Basisband-Übertragung durch eine geeignete Leitungs-Codierung immer sicherstellen, damit das Empfangssignal gleichanteilsfrei wird.

²Als Anfangswert für die „extrinsic“ Information wird das auch für die Turbo-Decodierung angenommen.

Hat man nun tatsächlich die Zahl $z_a(T)$ empfangen, muß man entscheiden, ob mit der größeren Wahrscheinlichkeit $s_1(t)$ oder $s_2(t)$ gesendet wurde. Gesucht sind also die **bedingten Wahrscheinlichkeiten**

- $P(s_1|z)$:
Wahrscheinlichkeit daß $s_1(t)$ gesendet wurde, für den Fall, daß $z(T)$ empfangen wurde (*Bedingung*).
- $P(s_2|z)$:
Wahrscheinlichkeit daß $s_2(t)$ gesendet wurde, für den Fall, daß $z(T)$ empfangen wurde (*Bedingung*).

Diese Wahrscheinlichkeiten heißen **à posteriori Wahrscheinlichkeit (APoP)**, da jetzt ja das Empfangsergebnis vorliegt. Der Entscheider trifft dann die Entscheidung:

$$z(T) \begin{matrix} H_1 \\ \geq \\ \gamma \\ H_2 \end{matrix} \rightsquigarrow \begin{cases} z(T) > \gamma & \rightsquigarrow H_1 \\ z(T) < \gamma & \rightsquigarrow H_2 \end{cases} \quad (1.7)$$

Die Entscheidung heißt damit:

- Ist $z(T) > \gamma$, entscheide für die Hypothese H_1 , d.h. $s_1(t)$ ist mit größter Wahrscheinlichkeit gesendet worden.
- Ist $z(T) < \gamma$, entscheide für die Hypothese H_2 , d.h. $s_2(t)$ ist mit größter Wahrscheinlichkeit gesendet worden.

1.2.3 Das Theorem von Bayes

Nach dem Theorem von Bayes gilt für die APoP $P(s_i|z)$:

$$P(s_i|z) = \frac{p(z|s_i)P(s_i)}{p(z)}; \quad i = 1, 2, \dots, M \quad \text{mit} \quad p(z) = \sum_{i=1}^M p(z|s_i)P(s_i) \quad (1.8)$$

Hierbei sind

- $P(s_i|z)$: À posteriori Wahrscheinlichkeit (APoP) dafür, daß s_i gesendet wurde, wenn z empfangen wurde (*als Bedingung*).
- $p(z|s_i)$: pdf von z , wenn s_i gesendet wurde (*als Bedingung*).
- $P(s_i)$: Wahrscheinlichkeit dafür, daß s_i gesendet wurde (à priori). Dies ergibt einen *Gewichtsfaktor* für die bedingte pdf $p(z|s_i)$.
- $p(z)$: \sum aller gewichteten (bedingten) pdf.

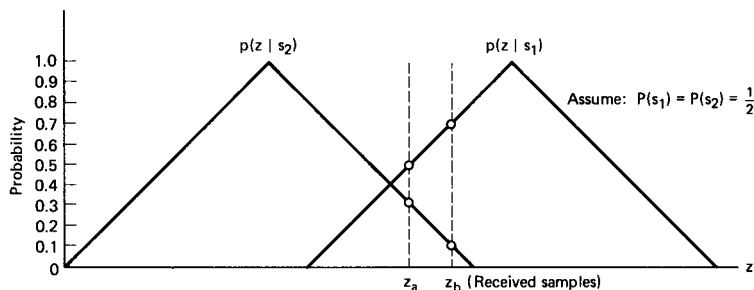


Bild 1.5: Beispiel für das Bayes'sche Theorem

Anhand eines vereinfachten Beispiels, Bild 1.5, soll die Berechnung gezeigt werden. Aus Bild 1.5 liest man ab:

$$p(z_a|s_1) = 0.5, p(z_a|s_2) = 0.3$$

Ferner sei für die à priori Wahrscheinlichkeiten angenommen $P(s_1) = P(s_2) = 1/2$.

Diese Werte werden in die Gleichung (1.8) eingesetzt. Damit ergibt sich für die à posteriori Wahrscheinlichkeiten:

$$\begin{aligned} P(s_1|z_a) &= \frac{p(z_a|s_1)P(s_1)}{p(z_a|s_1)P(s_1) + p(z_a|s_2)P(s_2)} & (1.9) \\ &= \frac{0.5 \cdot 0.5}{0.5 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{5}{8} \\ P(s_2|z_a) &= \frac{0.3 \cdot 0.5}{0.5 \cdot 0.5 + 0.3 \cdot 0.5} = \frac{3}{8} \end{aligned}$$

Die Entscheidung fällt für die größte à posteriori Wahrscheinlichkeit (APoP). Danach ist s_1 gesendet worden. Für den Empfangswert z_b ergeben sich folgende à posteriori Wahrscheinlichkeiten:

$$\begin{aligned} P(s_1|z_b) &= \frac{0.7 \cdot 0.5}{0.7 \cdot 0.5 + 0.1 \cdot 0.5} = \frac{7}{8} \\ P(s_2|z_b) &= \frac{0.1 \cdot 0.5}{0.1 \cdot 0.5 + 0.7 \cdot 0.5} = \frac{1}{8} \end{aligned}$$

Die Entscheidung fällt wieder für s_1 , diesmal aber mit einem viel größeren Vertrauen in ihre Richtigkeit. Die getroffene Entscheidung ist in beiden Fällen eine „harte“ Entscheidung (*hard decision*). Gibt man zusätzlich die Information in die *Vertrauenswürdigkeit der Entscheidung* an die nachfolgende Verarbeitungseinheit (z.B. Fehlerkorrektur) weiter (z.B. 111 für $P(s_i|z) = 1$; 000 für $P(s_i|z) = 0$), so hat man eine „weiche“ Entscheidung (*soft decision*) getroffen, Bild 1.6.

Bei dieser „soft“ Entscheidung können die Vorzeichen-Bits als „harte“ Entscheidungen aufgefaßt werden, während die restlichen Bits ein Maß für die Sicherheit einer solchen Entscheidung darstellen. Damit werden Zusatzinformationen „extrinsic“ gewonnen, die die Eigenschaften des Übertragungskanals berücksichtigen.

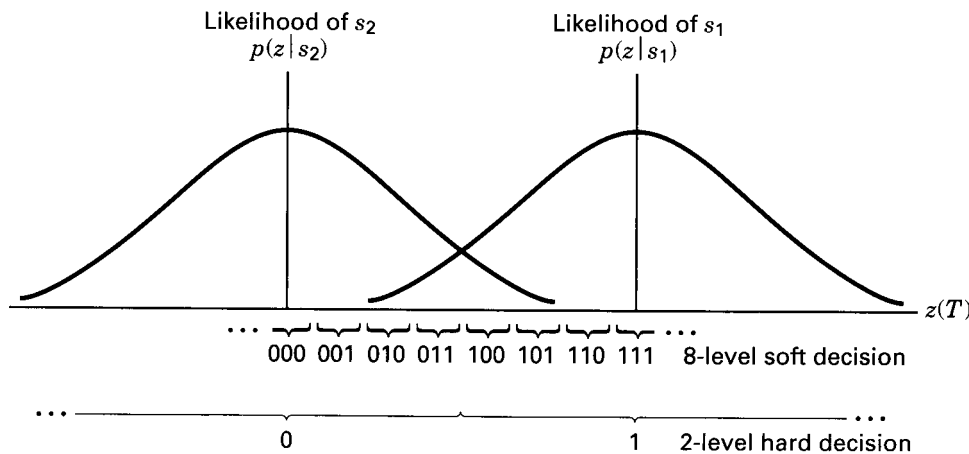


Bild 1.6: Harte und 3 Bit Soft Entscheidung bei einer BPSK Übertragung

1.2.4 Das Maximum à posteriori Kriterium

Das *Maximum à posteriori Kriterium* (MAP) besagt, daß die Wahl zu treffen ist, die der größten à posteriori Wahrscheinlichkeit entspricht. Dies wurde bereits in Gleichung (1.7) verwendet.

$$P(s_1|z) \underset{H_2}{\overset{H_1}{>}} P(s_2|z) \rightsquigarrow \begin{cases} P(s_1|z) > P(s_2|z) & \rightsquigarrow H_1 \\ P(s_1|z) < P(s_2|z) & \rightsquigarrow H_2 \end{cases} \quad (1.10)$$

Setzt man die Bayes-Beziehung, Gleichung (1.8) (Seite 4), hier ein, so folgt

$$p(z|s_1) \cdot P(s_1) \underset{H_2}{\overset{H_1}{>}} p(z|s_2) \cdot P(s_2) \quad (1.11)$$

Diese Gleichung läßt sich umformen zu:

$$\Lambda(z) = \frac{p(z|s_1)}{p(z|s_2)} \underset{H_2}{\overset{H_1}{>}} \frac{P(s_2)}{P(s_1)} \quad (1.12)$$

$\Lambda(z)$ ist das **Wahrscheinlichkeits-Verhältnis** (*Likelihood Ratio*).

1.2.5 Maximum Likelihood Kriterium

Oft kennt man die à priori Wahrscheinlichkeiten der Symbole nicht. Dann setzt man sie gleich wahrscheinlich an. Damit ergibt sich das **Maximum Likelihood Kriterium**:

$$\Lambda(z) = \frac{p(z|s_1)}{p(z|s_2)} \underset{H_2}{\overset{H_1}{>}} 1 \quad \rightsquigarrow \quad p(z|s_1) \underset{H_2}{\overset{H_1}{>}} p(z|s_2) \quad (1.13)$$

Damit kann die Entscheidung allein aus der Kenntnis der pdf getroffen werden.

1.2.6 Binär-Entscheidung nach Maximum Likelihood

Die dem Entscheider angebotenen Zahlenwerte sind

$$z(T) = \begin{cases} a_1 + n_0 \\ a_2 + n_0 \end{cases} \quad (1.14)$$

Aufgrund des Gauß'schen Rauschens erhält man die bedingten Wahrscheinlichkeits-Dichte-Funktionen für die Werte von $z(T)$ gemäß Bild 1.4 (Seite 3).

Für die pdf des (Gauß-verteilten) Rauschens gilt gemäß Gleichung (1.4):

$$p(n_0) = \frac{1}{\sigma_0 \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{n_0}{\sigma_0} \right)^2} \quad (1.15)$$

Aus den Gleichungen (1.5) und (1.6) folgt damit das Wahrscheinlichkeitsverhältnis $\Lambda(z)$:

$$\begin{aligned} \Lambda(z) &= \frac{p(z|s_1)}{p(z|s_2)} = \frac{e^{-\frac{1}{2} \left(\frac{z-a_1}{\sigma_0} \right)^2}}{e^{-\frac{1}{2} \left(\frac{z-a_2}{\sigma_0} \right)^2}} \underset{H_2}{\overset{H_1}{>}} \frac{P(s_2)}{P(s_1)} \\ &= \frac{e^{\left\{ -\frac{z^2}{2\sigma_0^2} \right\}} e^{\left\{ -\frac{a_1^2}{2\sigma_0^2} \right\}} e^{\left\{ -\frac{2za_1}{2\sigma_0^2} \right\}}}{e^{\left\{ -\frac{z^2}{2\sigma_0^2} \right\}} e^{\left\{ -\frac{a_2^2}{2\sigma_0^2} \right\}} e^{\left\{ -\frac{2za_2}{2\sigma_0^2} \right\}}} = e^{\left\{ \frac{z(a_1-a_2)}{\sigma_0^2} - \frac{a_1^2-a_2^2}{2\sigma_0^2} \right\}} \underset{H_2}{\overset{H_1}{>}} \frac{P(s_2)}{P(s_1)} \end{aligned} \quad (1.16)$$

Auf beiden Seiten wird der (natürliche) Logarithmus gebildet:

$$L(z) = \ln(\Lambda(z)) = \frac{z(a_1 - a_2)}{\sigma_0^2} - \frac{a_1^2 - a_2^2}{2\sigma_0^2} \underset{H_2}{\overset{H_1}{>}} \ln \frac{P(s_2)}{P(s_1)} \quad (1.17)$$

Wenn beide à priori Wahrscheinlichkeiten (APriP) gleich sind (*maximum likelihood*, ML), wird die rechte Seite von Gleichung (1.17):

$$\ln \frac{P(s_2)}{P(s_1)} = 0 \quad (1.18)$$

Damit kann nach der Größe der optimalen Schwelle γ_0 aufgelöst werden.

$$z \underset{H_2}{\overset{H_1}{>}} \frac{a_1^2 - a_2^2}{2(a_1 - a_2)} = \frac{a_1 + a_2}{2} = \gamma_0 \quad \text{Schwelle: arithmetischer Mittelwert} \quad (1.19)$$

Für den Fall, daß die Signale gegenphasig (*antipodal*) sind,

$$s_1(t) = -s_2(t) \quad \rightsquigarrow \quad a_1 = -a_2 \quad , \quad (1.20)$$

wird die Schwelle $\gamma_0 = 0$.

$$z \underset{H_2}{\overset{H_1}{>}} 0 = \gamma_0 \quad (1.21)$$

Das Ergebnis der Rechnung bestätigt den nach der Anschauung erwarteten Fall der Schwelle genau in der Mitte zwischen beiden Symbolwerten a_1, a_2 . Die genau symmetrische Schwelle ergibt sich aufgrund der angenommenen gleichen à priori Wahrscheinlichkeiten.

1.2.7 Schwelle bei ungleichen à priori Wahrscheinlichkeiten

Sind die à priori Wahrscheinlichkeiten (APriP) $P(s_1), P(s_2)$ dem Empfänger bekannt, kann dieser seine Entscheidungs-Schwelle geeignet festlegen. Hierdurch resultiert eine geringere Bit-Fehlerrate (BER).

Als Beispiel werde ein Code (bzw. ein *einzelnes* Code-Wort) betrachtet, bei dem die Symbole s_1 3 mal so häufig sein sollen wie die Symbole s_2 . Mit Gleichung (1.1) ist $P(s_1) + P(s_2) = 1$, so daß für das Beispiel gilt:

$$P(s_1) = \frac{3}{4}; \quad P(s_2) = \frac{1}{4} \quad (1.22)$$

Der Effektivwert des Rauschens σ_0 (des Kanals) ist für beide Symbole identisch. Daher ist auch die Form der Gauß-Kurven für die jeweilige pdf identisch. Als Unterschied bleibt aber die jeweilige Höhe der Gauß-Glocke, da ja die Fläche unter dieser den Wert der jeweiligen APriP annehmen muß, Bild 1.7.

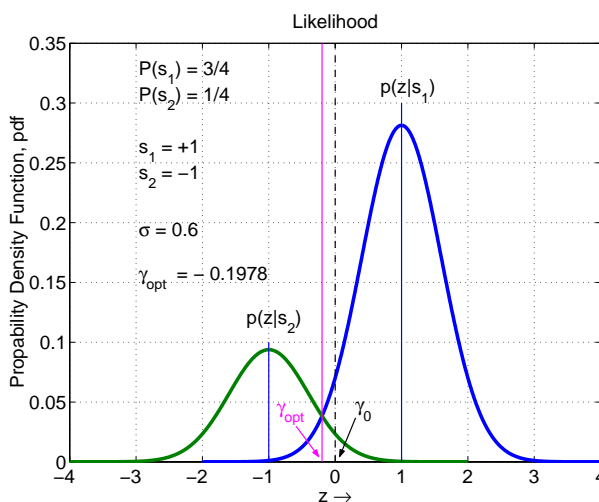


Bild 1.7: Wahrscheinlichkeitsdichten bei unterschiedlicher à priori Wahrscheinlichkeit

Wie aus Bild 1.7 ersichtlich, ist für den Fall, daß die beiden APriP nicht mehr gleich sind, die Entscheidungs-Schwelle bei $\gamma_0 = 0$ nicht mehr optimal. Die hierfür günstige Schwelle bestimmt sich mit Gleichung (1.17), wobei zur Vereinfachung

$$\ln \frac{P(s_2)}{P(s_1)} = \xi \quad (1.23)$$

gesetzt wird. Damit folgt:

$$\frac{z(a_1 - a_2)}{\sigma_0^2} - \frac{a_1^2 - a_2^2}{2\sigma_0^2} \underset{H_2}{\overset{H_1}{>}} \xi \quad (1.24)$$

Nach z aufgelöst ergibt sich:

$$z \underset{H_2}{\overset{H_1}{>}} \underbrace{\frac{a_1 + a_2}{2}}_{\gamma_0} + \frac{\xi \cdot \sigma_0^2}{a_1 - a_2} = \gamma_{\text{opt}} \quad (1.25)$$

Es zeigt sich hier, daß die optimale Entscheidungs-Schwelle γ_{opt} mit dem Schnittpunkt der beiden Gauß-Glocken zusammenfällt.

2 Turbo-Encoder

Besonders günstig wirkt sich die Optimierung der Entscheidungs-Schwelle aus, wenn der Empfänger die APriP für jedes einzelne Codewort kennt und er daher individuell seine Entscheidungs-Schwelle wählen kann. Die Stärke der Turbo-Codierung beruht genau darauf, daß diese in der Lage ist, die APriP näherungsweise zu bestimmen.

2.1 Struktur des Turbo-Encoders

Die Turbo-Codierung verwendet eine parallele Code-Verkettung. Es ist ein systematischer Code, bei dem die Daten-Bits als Teil des Code-Wortes auftreten. Die Parity-Bits p_1, p_2 werden durch 2 (i.a. gleiche) Faltungs-Codierer C_1 und C_2 berechnet. Der 2. Faltungs-Codierer erhält die Daten nachdem diese einen Interleaver (Int) durchlaufen haben, Bild 2.1.

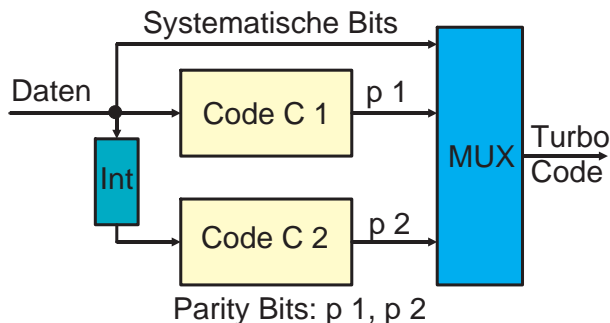


Bild 2.1: Blockschaltbild eines Turbo-Encoders

Die Daten werden in (möglichst große) Blöcke unterteilt. Die Blocklänge wird durch die Länge des verwendeten Interleavers (Int) bestimmt. Im Multiplexer (MUX) werden systematische Bits und Parity-Bits zum turbo-codierten Datenstrom zusammengefaßt.

Damit der Turbo-Decoder die gewünschte APriP (Zusatz-Information, *Extrinsic Information*) möglichst gut gewinnen kann, müssen folgende Bedingungen erfüllt sein:

- Die Länge (bzw. Tiefe) des Interleavers muß (sehr) groß sein. (typisch $> 20,000$ Bits)
Der Turbo-Code wird besser mit wachsender Länge des Interleavers.
- Durch das Interleaven muß der (interleave)te Datenblock statistisch unabhängig von dem eingangsseitigen Datenblock werden.
Dadurch werden die Parity-Worte p_1, p_2 statistisch von einander unabhängig.

- Bei großen Block-Längen werden pseudo-random Interleaver verwendet.
Hiermit werden (rechteckförmige) Fehlermuster vermieden, die bei Blockinterleavern in beiden Paritäts-Wörtern auftreten könnten.
- Die zugehörigen Blöcke der Parity-Worte p_1, p_2 dürfen nicht beide gleichzeitig ein geringes Gewicht haben.
Der Interleaver bestimmt daher wesentlich die Eigenschaften des Turbo-Codes.
- Die Faltungs-Codierer müssen eine rückgekoppelte Struktur aufweisen. (entsprechend zu IIR-Filtern)
Dadurch lassen sich Parity-Bits mit geringem Gewicht vermeiden.
Wegen der IIR-Struktur genügen wenige Verzögerungs-Stufen. Einfluß-Länge K (Constraint Length) $3 \leq K \leq 5$.
- Das Rückkopplungs-Polynom der Faltungs-Codierer muß teilerfremd sein.
Hiermit wird die effektive Distanz d_{eff} des Codes maximiert.
- Es dürfen keine katastrophalen Faltungs-Codes verwendet werden.
Dies betrifft die Teil-Codes infolge von Punktierung.

2.2 Beispiel eines Systematischen Rekursiven Faltungs-Encoders

Ein Beispiel für einen Rekursiven Systematischen Faltungs-Coder (RSC) zeigt Bild 2.2 (Coder C_1 & Coder C_2 in Bild 2.1). Das Generator-Polynom lautet:

$$g_{\text{rsc}}(D) = \frac{1 + D^2}{1 + D + D^2} \quad (2.1)$$

Diese Gleichung liest man aus der Struktur direkt ab: ÜTF = Σ Vorwärts-Wege / ($1 + \Sigma$ Schleifen-Wege).

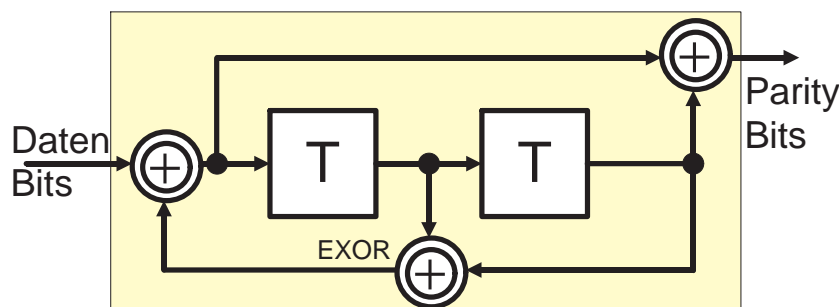


Bild 2.2: Blockschaltbild eines RSC Encoders zur Erzeugung des Codes C_1 bzw. C_2

Die beiden Coder in Bild 2.1 bestehen dann jeweils aus einem derartigen RSC Faltungs-Encoder. Diese liefern die Parity Bits p_1 bzw. p_2 .

Es ist üblich, für die Parity-Bits eine Punktierung anzuwenden um so von der Code-Rate $1/3$ zu besseren Code-Raten (maximal $1/2$) zu gelangen (wodurch dann allerdings die BER verschlechtert wird).

3 Turbo-Decoder

Der bei Faltungs-Codierung verwendete Viterbi Decodier-Algorithmus ist in der Lage, die Zuverlässigkeits-Informationen des Kanals auszuwerten. Dies geschieht mit Hilfe einer Soft-Entscheidung, Bild 1.4 (Seite 3). Allerdings liefert der Viterbi-Algorithmus im Anschluß eine „harte“ Entscheidung über die Bits.

Der für die Turbo-Decodierung verwendete Algorithmus soll im Gegensatz dazu als Ausgangs-Größe eines jeden Teil-Decoders eine „soft“ Entscheidung liefern, die dann dem anderen Teil-Decoder als „extrinsic“ Zusatzinformation zur Verfügung gestellt wird. Dieser liefert im Anschluß daran seine dadurch gewonnenen Zusatzinformationen an den anderen ab usw..

Es handelt sich daher bei der Turbo-Decodierung um einen iterativen Prozess. Der erste Teil-Decodiervorgang kann dabei auf keine „extrinsic“ Informationen zugreifen und setzt daher beide APriP zu $P(s_1) = P(s_2) = 1/2$ an.

3.1 Struktur des Turbo-Decoders

Der Turbo-Decoder verwendet 2 Decodierer vom Typ „Soft-Input / Soft-Output“ (SISO). Die für eine optimale Entscheider-Schwelle γ_{opt} notwendige APriP Information gewinnt man dabei in einem iterativen Prozess aus den Decodier-Ergebnissen der beiden Decodierer. Dies wird (näherungsweise) deshalb möglich, weil die Daten-Bits vor der Bildung der Parity Bits des 2. Encoders interleaved wurden und sie damit statistisch unabhängig von den Parity Bits des 1. Encoders sind.

Die Struktur eines einzelnen SISO Decodierers und die Gewinnung der „extrinsic“ Information ist in Bild 3.1 dargestellt.

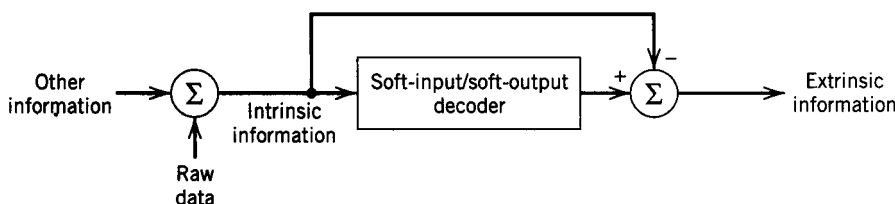


Bild 3.1: Blockschaltbild eines SISO Decoders und die Gewinnung der „extrinsic“ Information

Die „extrinsic“ Information gewinnt man aus der Differenz zwischen den decodierten und den nicht decodierten Daten. Diese wird dann als „andere“ Information an den 2. Decodierer weitergereicht.

Der SISO Algorithmus ist vom Aufwand mehr als doppelt so groß wie der Viterbi Algorithmus. Dies kommt dadurch zustande, daß hier nicht nur Vorwärts-Wege durch den Trellis berücksichtigt werden, sondern auch Rückwärts-Wege und als Ergebnis ein „soft“ Wert entstehen muß. Ein entsprechender Algorithmus wurde von *Bahl, Cocke, Jelinek und Raviv* angegeben. Er heißt deshalb *BCJR* Algorithmus.

Die Struktur des Turbo-Decoders zeigt Bild 3.2. Die Decodierer sind mit *BCJR* bezeichnet.

Close switch at time step $n = 0$ and set $\tilde{l}_2(\mathbf{x}) = 0$

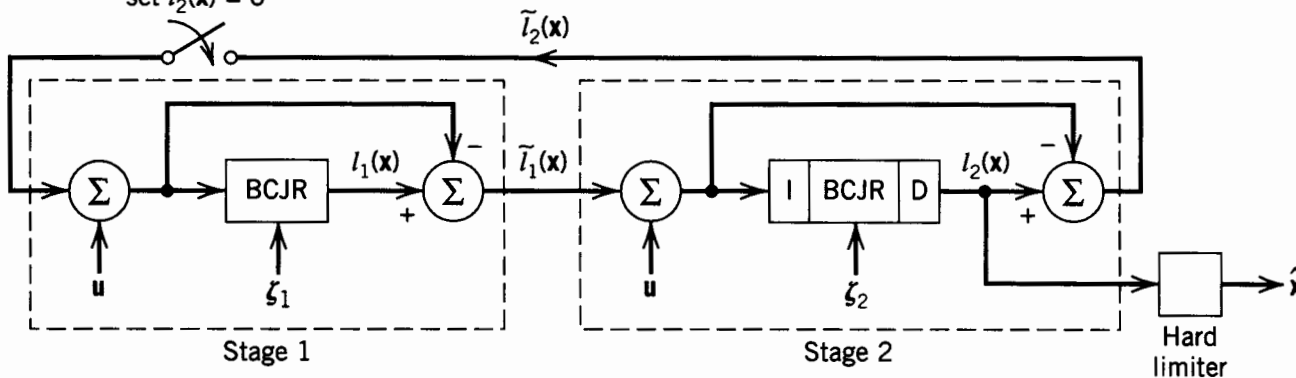


Bild 3.2: Blockschaltbild eines Turbo Decoders. Die Decodierer sind mit *BCJR* bezeichnet.

In Bild 3.2 sind u die gestörten systematischen Bits, ζ_1, ζ_2 die gestörten Parity Bits. Da die Parity Bits ζ_2 interleaved sind, müssen daher die Eingangsdaten $u, L_1(x)$ für den Decodierer 2 ebenfalls interleaved werden (Block I), damit sie zusammen passen. Am Ausgang des Decodierers 2 muß nun ein De-Interleaver (Block D) folgen.

Die nach einigen Iterationen sich ergebenden (weitestgehend) fehlerfreien Decodier-Ergebnisse werden dem Decodierer 2 entnommen. Das Vorzeichen dieser „soft“ Werte gibt die Binärdaten und der Betrag die

Zuverlässigkeit der Entscheidung.

Ein Ausgang hinter Decodierer 2 ist ausreichend, da sich die beiden Decodierer im Verlauf des Iterations-Prozesses über das Decodier-Ergebnis „einigen“ und somit am Ausgang des 1. Decodierers das gleiche Ergebnis entsteht.

Beim 1. Decodiervorgang (in Stufe 1) gibt es noch keine „extrinsic“ Information. Die Rückkopplung ist zu diesem Zeitpunkt unterbrochen. Dafür wird dann „0“ eingespeist.

3.2 Zuverlässigkeits-Information und L -Algebra

Hierfür wird Gleichung (1.23) für die empfangenen Signale z aufgestellt (siehe Bild 1.3, Seite 2) und umgeformt. Man erhält dadurch eine (logarithmische) Zuverlässigkeitsinformation (*log-Likelihood-Ratio* LLR) $L(z)$ für die Random Variable z .

$$L(z) = \ln \frac{P(z_1)}{P(z_2)} = \ln \frac{P(z = z_1)}{P(z = z_2)} \quad (3.1)$$

Das Vorzeichen von $L(z)$ stellt dabei eine „harte“ Entscheidung dar, während der Betrag $|L(z)|$ die Zuverlässigkeit dieser Entscheidung angibt und damit die „soft“ Informationen liefert. Je größer der Unterschied zwischen $P(z = z_1)$ und $P(z = z_2)$ ist, desto größer ist der Betrag des Logarithmus und um so sicherer ist die Entscheidung.

Mit dem Theorem von Bayes, Gleichung (1.8), kann die Zuverlässigkeits-Information aus dem Ausdruck

$$L(\hat{s}) = L(s|z) = \ln \left[\frac{P(s_1|z)}{P(s_2|z)} \right] = \underbrace{\ln \left[\frac{p(z|s_1)}{p(z|s_2)} \right]}_{L(z|s)=L_C(z)} + \underbrace{\ln \left[\frac{P(s_1)}{P(s_2)} \right]}_{L_A(s)} \quad (3.2)$$

gewonnen werden. Der Term $L(z|s) = L_C(z)$ enthält dabei die Kanalstatistik $p(z|s)$ und ist somit das Resultat einer Kanal-Messung durch den Empfänger, während der Term $L_A(s)$ das à priori Wissen APriP enthält.

Aufgrund der statistischen Unabhängigkeit der à priori Informationen von den empfangenen Symbolen können die LLR's einfach addiert werden.

$$L(\hat{s}) = L(s|z) = L(z|s) + L_A(s) = L_C(z) + L_A(s) \quad (3.3)$$

Gleichung (3.3) beschreibt das LLR, das mit Hilfe eines Decoders gewonnen werden kann, wenn die APriP Informationen bekannt sind. Bei der Turbo-Decodierung wirken nun 2 Decodierer zusammen, so daß die fehlenden APriP Informationen durch die „extrinsic“ Information $L_e(\hat{s})$ ersetzt wird, die der jeweils andere Decoder liefert. Somit erhält man:

$$L(\hat{s}) = L_C(z) + L_e(\hat{s}) \quad (3.4)$$

Hat $L_e(\hat{s})$ gleiches Vorzeichen wie $L_C(z)$, wird die Zuverlässigkeit von $L(\hat{s})$ verstärkt.

3.2.1 L -Algebra

Der iterative Rückkopplungs-Prozess der Turbo-Decodierung läßt sich relativ einfach mit der L -Algebra beschreiben.

Die Summe zweier LLR's ist dabei wie folgt definiert.[2]

$$L(s_1) \boxplus L(s_2) \equiv L(s_1 \oplus s_2) = \ln \left[\frac{1 + e^{L(s_1)} \cdot e^{L(s_2)}}{e^{L(s_1)} + e^{L(s_2)}} \right] \quad (3.5)$$

Dies läßt sich approximieren mit (Herleitung in [2]):

$$L(s_1) \boxplus L(s_2) \approx \text{sgn}[L(s_1)] \cdot \text{sgn}[L(s_2)] \cdot \min\{|L(s_1)|, |L(s_2)|\} \quad (3.6)$$

Hierbei bedeutet die Funktion $\text{sgn}[\cdot]$ „das Vorzeichen von“ (Signum Funktion) und $\min\{|L(s_1)|, |L(s_2)|\}$ „das Minimum von“. \oplus ist eine modulo-2 Addition und \boxplus bedeutet die Log-Likelihood Addition.

Für die Log-Likelihood Addition gelten folgende Regeln, die die Berechnung sehr stark vereinfachen:

$$L(s) \boxplus \infty = L(s); \quad L(s) \boxplus -\infty = -L(s); \quad L(s) \boxplus 0 = 0 \quad (3.7)$$

In der Praxis wird (aufgrund der obigen Approximation) zur Vereinfachung das betragsmäßig größere LLR stets als ∞ angesetzt.

3.3 Beispiel für Turbo-Decodierung

Gewählt wird ein Turbo-Code mit 2 RSC-Codes gemäß Bild 2.1. Der Interleaver macht eine zufällig ausgewürfelte (pseudo random) Permutation der Länge 20000. Die Code-Rate wird durch Punktierung zu $R_c = 1/2$ gewählt. Bild 3.3 zeigt die relativen Häufigkeitsverteilungen der LLR's der Informations-Bits und Bild 3.4 die relative Häufigkeitsverteilungen der „extrinsic“ LLR's, jeweils abhängig von der Anzahl der Iterationen.

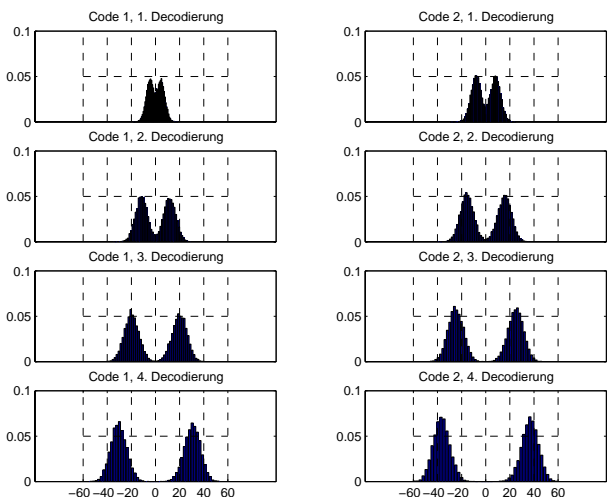


Bild 3.3: relative Häufigkeitsverteilungen der LLR's der Informations-Bits

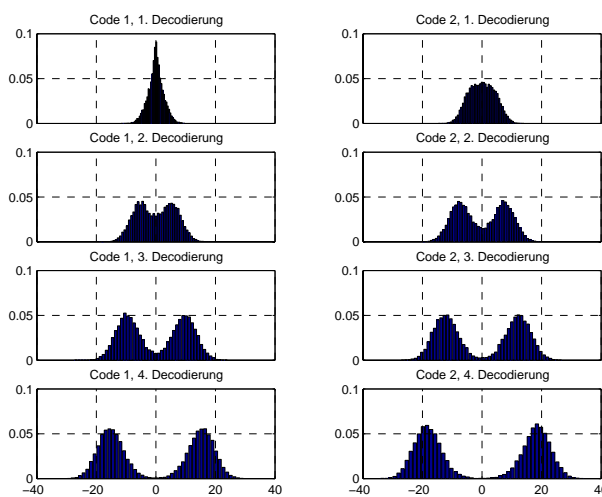


Bild 3.4: relative Häufigkeitsverteilungen der „extrinsic“ LLR's

Mit zunehmender Zahl der Iterationen nimmt die Korrelation zwischen der „extrinsic“ Information und dem systematischen Teil der Empfangsdaten zu, so daß der Gewinn pro Iteration abnimmt, Bild 3.5.

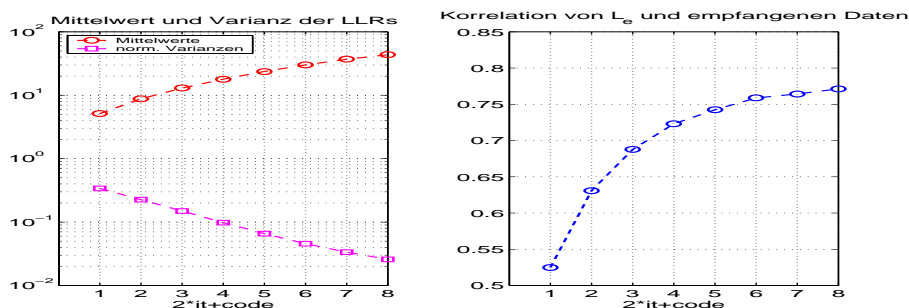


Bild 3.5: Mittelwerte, Varianz und Korrelation zwischen „extrinsic“ Information und dem systematischen Anteil der Empfangsdaten

Literatur

- [1] Guizzo, E.: *Closing in on the Perfect Code*, IEEE Spectrum, 04. 2004, pp. 28 — 34
- [2] Hagenauer, J.; Offer, E.; Papke, L.: *Iterative Decoding of Binary Block and Convolutional Codes*, IEEE Trans. on Information Theory, 03. 1996, pp. 429 — 445
- [3] Hagenauer, J.: *Faltungscodes und Turbocodierung*, CCG Kurs, 1997
- [4] Kammeyer, K.D.; Kühn, V.: *Matlab in der Nachrichtentechnik*, J. Schlembach, 2001
- [5] Sklar, B.: *Digital Communications, Fundamentals and Applications*, Prentice Hall, 2nd ed. 2001
- [6] Haykin, S.: *Communication Systems*, Wiley, 4th ed. 2001