The Turbo Principle: Tutorial Introduction and State of the Art

Joachim Hagenauer
Technical University of Munich
Department of Communications Engineering (LNT)
Arcisstr. 21, 80290 Munich, Germany

Abstract— **The turbo principle can be used in a more general way than just for the decoding of parallel concatenated codes. Using log-likelihood algebra for binary codes two simple examples are given to show the essentials of the turbo iteration for decoding and equalization. For reference the basic symbol-by-symbol MAP algorithm is stated and simplified in the log-domain. The results of turbo applications in parallel and serial decoding, in source-controlled channel decoding, in equalization, in multiuser detection and in coded modulation are described.**

I. Introduction

In 1993 decoding of two and more dimensional product-like codes has been proposed with iterative ('turbo') decoding [1] using similar ideas as in [3] and [4]. The basic concept of this new (de)coding scheme is to use a parallel concatenation of at least two codes with an interleaver between the encoders. Decoding is based on alternately decoding the component codes and passing the so-called *extrinsic information* which is a part of the soft output of the soft-in/soft-out decoder to the next decoding stage. Even though very simple component codes are applied, the 'turbo' coding scheme is able to achieve a performance rather close to Shannon's bound, at least for large interleavers and at bit error rates of approximately 10^{-5}.
However, it turned out that the method applied for these parallel concatenated codes is much more general. Strictly speaking there is nothing 'turbo' in the codes. Only the decoder uses a 'turbo' feedback and the method should be named the 'Turbo-Principle', because it can be sucessfully applied to many detection/decoding problems such as serial concatenation, equalization, coded modulation, multiuser detection, joint source and channel decoding and others. We will explain the basic principle and hereby we will try to follow Einstein's principle: 'Everything should be as simple as possible but not simpler'. Thererfore we restrict ourselves to binary data and codes, use consequently the log-likelihood notation and start with two rather simple examples which show the basic ingredients of the turbo iteration.

Hag@LNT.E-Technik.TU-Muenchen.DE, phone: ++49 89 28923467, fax.: ++49 89 28923490

II. MAP Symbol-by-Symbol Estimation and the Turbo Principle

In detection and decoding we discriminate between sequence and symbol estimation where both can use the maximum-likelihood (ML) or the maximum-a-posteriori (MAP) rule. Here we consider MAP symbol-by-symbol estimation of the symbols u_k of a vector $\mathbf{u}$ which is received as a vector $\mathbf{y}$ after encoding and distortion by a Gauss-Markov process according to the distribution $p(\mathbf{y}|\mathbf{u})$. In addition we might have available the a priori probability $P(\mathbf{u})$ as an input to the estimator. The output of the estimator provides us with the a posteriori probability

$$P(\hat{u}_k|\mathbf{y})$$

to be used in subsequent processing.
The '**Turbo Principle**' can be formulated as follows:
Perform iterative MAP-estimations of the symbols with sucessively refined a priori distributions $P_i(\mathbf{u})$. For the calculation of $P_i(\mathbf{u})$ use all the preferably statistically independent information which is available at iteration i (set of sufficient statistics).
Examples how to obtain $P_i(\mathbf{u})$ are:
- a priori if available or a posteriori obtained from source bit statistics
- a posteriori probabilities from parallel transmissions, such as diversity, parallel

concatenations, correlated multiuser channels

- a posteriori probabilities from the (i-1)th decoding of an outer code (serial concatenation)
- combinations of a posteriori probabilities from previous decoding of parallel and serial concatenations.

The name 'turbo' is justified because the decoder uses its processed output values as a priori input for the next iteration, similar to a turbo engine.

A. Log-Likelihood Algebra

Let U be in GF(2) with the elements $\{+1,-1\}$, where +1 is the 'null' element under the $\oplus$ addition. The log-likelihood ratio of a binary random variable U, $L_U(u)$, is defined as

$$L_U(u) = \log \frac{P_U(u=+1)}{P_U(u=-1)}. \quad (1)$$

Here $P_U(u)$ denotes the probability that the random variable U takes on the value u. The log-likelihood ratio $L_U(u)$ will be denoted as the L-value of the random variable U. The sign of $L_U(u)$ is the hard decision and the magnitude $|L_U(u)|$ is the reliability of this decision. Unless stated otherwise, the logarithm is the natural logarithm. We will henceforth skip the indices for the probabilities and the log-likelihood ratios.

If the binary random variable u is conditioned on a different random variable or vector y, then we have a conditioned log-likelihood ratio $L(u|y)$ with

$$L(u|y) = L(u) + L(y|u), \quad (2)$$

employing Bayes rule. Using

$$P(u=\pm 1) = \frac{e^{\pm L(u)}}{1+e^{\pm L(u)}}, \quad (3)$$

it is easy to prove for statistically independent random variables u_1 and u_2 that

$$\begin{aligned} L(u_1 \oplus u_2) &= \log \frac{1+e^{L(u_1)}e^{L(u_2)}}{e^{L(u_1)}+e^{L(u_2)}} \quad (4) \\ &\approx \operatorname{sign}(L(u_1)) \cdot \operatorname{sign}(L(u_2)) \cdot \\ &\cdot \min(|L(u_1)|, |L(u_2)|). \end{aligned}$$

We use the symbol $\boxplus$ as the notation for the addition defined by

$$L(u_1) \boxplus L(u_2) \triangleq L(u_1 \oplus u_2), \quad (5)$$

with the rules

$$L(u) \boxplus \pm\infty = \pm L(u), \qquad L(u) \boxplus 0 = 0. \quad (6)$$

The reliability of the sum $\boxplus$ is therefore determined by the smallest reliability of the terms. Equation (5) can be reformulated using the 'soft' bit

$$\lambda_i = \tanh(L(u_i)/2) \quad (7)$$

which, using (3), can be shown to be the expectation of u_i:

$$\begin{aligned} \mathrm{E}\{u_i\} &= (+1)\frac{e^{+L(u_i)}}{1+e^{+L(u_i)}} + (-1)\frac{e^{-L(u_i)}}{1+e^{-L(u_i)}} \\ &= \tanh(L(u_i)/2). \quad (8) \end{aligned}$$

Then

$$L(u_1 \oplus u_2) = 2 \operatorname{artanh}(\lambda_1 \lambda_2). \quad (9)$$

Soft Channel Outputs

After transmission over a binary symmetric channel (BSC) or a Gaussian/Fading channel we can calculate the log-likelihood ratio of the transmitted bit x conditioned on the matched filter output y

$$L(x|y) = \log \frac{P(x=+1|y)}{P(x=-1|y)} \quad (10)$$

With our notation we obtain

$$\begin{aligned} L(x|y) &= \log \frac{\exp(-\frac{E_s}{N_0}(y-a)^2)}{\exp(-\frac{E_s}{N_0}(y+a)^2)} + \log \frac{P(x=+1)}{P(x=-1)} \\ &= L_c \cdot y + L(x), \quad (11) \end{aligned}$$

with $L_c = 4a \cdot E_s/N_0$. For a fading channel a denotes the fading amplitude whereas for a Gaussian channel we set $a=1$. We further note that for statistically independent transmission, as in dual diversity or with a repetition code

$$L(x|y_1, y_2) = L_{c_1} y_1 + L_{c_2} y_2 + L(x). \quad (12)$$

B. The 'Turbo'-Principle using L-values

With the L-values we can reformulate the 'Turbo' principle using Fig.1. A turbo decoder accepts a priori and channel L-values and delivers soft-output L-values $L(\hat{u})$. In addition the so-called *extrinsic* L-values for the information bits $L_e(\hat{u})$ and/or the coded bits $L_e(\hat{x})$ are produced. Extrinsic information refers to the incremental information about the current bit obtained through the decoding process from all the other bits. Only this extrinsic values should be used to gain the new a priori value for the next iteration, because this is statistically independent information,- at least during the first iteration. The decoders described in section 3 deliver the soft output in the form

$$L(\hat{u}) = L_c y + L(u) + L_e(\hat{u})$$

showing that the MAP estimate contains 3 parts: from the channel, from the a priori knowledge and from the other bits through constraints of the code or the Markov property.

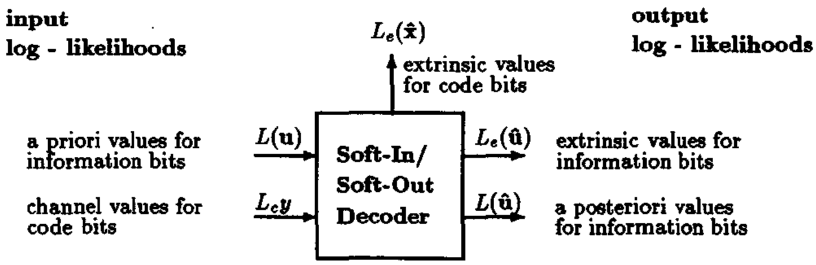

Fig. 1. Soft-in/soft-out decoder for turbo iterations

A) coded values

+	+	+
+	-	-
+	-	

B) received values $L_c \cdot y$

+0.5	+1.5	+1.0
+4.0	+1.0	-1.5
+2.0	-2.5	

C) after first decoding in horizontal direction

+0.5	+1.5	+1.0
+4.0	+1.0	-1.5
+2.0	-2.5	

+1.0	+0.5
-1.0	-1.5

$L_e^-(\hat{u}_k)$

D) after first decoding in vertical direction

+0.5	+1.5	+1.0
+4.0	+1.0	-1.5
+2.0	-2.5	

+1.0	+0.5
-1.0	-1.5

$L_e^-(\hat{u}_k)$

+2.0	+0.5
+1.5	-2.0

$L_e^|(\hat{u}_k)$

E) soft output after the first decoding in horizontal and vertical direction

+3.5	+2.5
+4.5	-2.5

Fig. 2. Tutorial example of a parallel concatenated code with 4 (3,2,2) single parity check codes

C. *Tutorial Example with a Simple Parallel Concatenated 'Turbo'-Scheme Using (3,2,2) Single Parity Check Codes*

Let us encode four information bits by two $(3,2,2)$ single parity check codes with elements $\{+1,-1\}$ in GF(2) as shown in Figure 2 A) and let us assume we have received the values $L_c \cdot y$ shown in Figure 2 B). No *a priori* information is yet available. Let us start with horizontal decoding: The information for bit u_{11} is received twice: Directly via u_{11} and indirectly via $u_{12} \oplus p_1^-$. Since u_{12} and p_1^- are statistically independent we have for their L-value: $L(u_{12} \oplus p_1^-) = L(u_{12}) \boxplus L(p_1^-) = 1.5 \boxplus 1.0 \approx 1.0$. This indirect information about u_{11} is called the extrinsic value and is stored in the right matrix of table 2 C). For u_{12} we obtain by the same argument a horizontal extrinsic value of $0.5 \boxplus 1.0 \approx 0.5$ and so on for the second row. When the horizontal extrinsic table is filled we start vertical decoding using these L_e^- as *a priori* values for vertical decoding. This means that after vertical decoding of u_{11} we have the following three L-values available for u_{11}

- the received direct value +0.5,
- the *a priori* value L_e^- from horizontal decoding +1.0 and
- the vertical extrinsic value $L_e^|$ using all the available information on $u_{21} \oplus p_1^|$, namely $(4.0 + (-1.0)) \boxplus 2.0 \approx 2.0$.

The vertical extrinsic value is stored in Table 2 D). For u_{21} it amounts to $(0.5{+}1.0) \boxplus 2.0 \approx 1.5$, for u_{12} to $(1.0+(-1.5)) \boxplus (-2.5) \approx 0.5$ and for u_{22} to $(1.5+0.5) \boxplus (-2.5) \approx -2.0$. If we were to stop the iterations here we would obtain as soft output after the vertical iteration

$$L(\hat{u}) = L_c \cdot y + L_e^- + L_e^| \qquad (13)$$

shown in Figure 2 E). The addition in (13) is justified from (12) because up to now the three terms in (13) are statistically independent. We could now continue with another round of horizontal decoding using the respective $L_e^|$ as *a priori* information.

D. Tutorial Example with a Simple Serial Concatenated 'Turbo'-Scheme

The basic idea of iterative decoding of serial concatenated codes with feedback between the inner and outer decoders is to go back to the inner decoding after successfully finishing an outer decoding trial. If the decisions made by the outer decoder are assumed to be correct, the inner decoder is provided with reliability information about the bits to be decoded. Using this information as a *priori* information the inner decoder restarts decoding and will deliver less erroneous decisions which are passed again to the outer decoder.
We consider binary coded multipath transmission similar as in the GSM system, but for tutorial purposes in a much simpler setup: Three times two information bits generate three codewords of a (3,2,2) SPC-Code. They are block-interleaved and transmitted over a 2-tap multipath channel as shown in Fig. 3 The transition

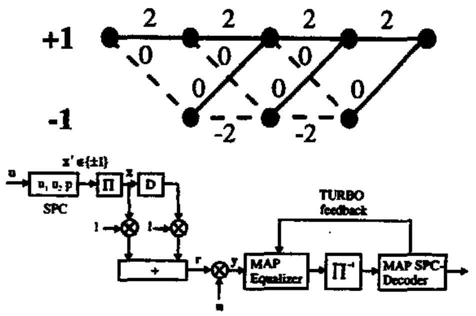

Fig. 3. Tutorial example of a serial concatenated scheme using a 2-tap multipath channel as the inner code and interleaved (3,2,2) single parity check (SPC) codes as outer codes

metric of the MAP or Viterbi (VA) algorithm is from (29) $-(y_k - x_k - x_{k-1})^2$, where the variance of the noise is set to 1/2 in our example which corresponds to an $\overline{E}_s/N_0$ of 3dB. Assume that all 6 information bits are +1. After transmission of the 3 u_1 bits of the codewort which is followed by a +1 tail bit, we received the following y_k values: $\{+0.5\ -1.0\ +1.0+2.0\}$.
It will be shown in section 3-B that for the trellis in Fig. 3 the MAP soft output algorithm can be closely approximated by two VA running back and forth leading to state metrics $M_k^{\rightarrow}$ and $M_k^{\leftarrow}$. The soft output $L(\hat{u}_k)$ is then the difference of $(M_k^{\rightarrow} + M_k^{\leftarrow})$ of the upper states minus the respective values of the lower states, see (25). By hand we obtain the non-bracket values

k	1	2	3
$M_k^{\rightarrow}$	(-0.75)-2.25	-1.25	-2.25
	(-1.75)-0.25	-1.25	-2.25
$M_k^{\leftarrow}$	-2.00	-1.00	0.00
	-2.00	-1.00	-4.00
$L(\hat{u}_1)$	(+1.00)-2.00	0.00	+4.00

Now assume that for the other bits u_2 and p_1 we also have obtained their output L-values and by vertical $\boxplus$ evaluation of the 2. and 3. row elements the extrinsic information for u_1 as

k	1	2	3
$L(\hat{u}_1)$	(+1,0) -2.00	0.00	+4.00
$L(\hat{u}_2)$	+3,00	+4.00	0.00
$L(\hat{p})$	+6.50	0.00	+3.00
$L_e(\hat{u}_1)$	+3.00	0.00	0.00

This extrinsic information is fed back via the turbo link to the inner decoder (the equalizer) and we perform a second round of equalization Here we evaluate only the second MAP equalization of the u_1 bits. Now we have to add, or -for -1 transmissions- subtract half of the extrinsic values namely { +1.5 0.0 0.0 } to the transition metrics. We execute it here only for u_1 and we get subsequently the values in brackets. For the first u_1 this means that we have now a correct decision after the first turbo iteration.

III. Soft-In/Soft-Out Detectors and Decoders

A. The BCJR-Algorithm for a Binary Trellis

For reference, we cite here the well known Bahl-Cocke-Jelinek-Raviv Algorithm [2] in the fashion as described in [5]: For a binary trellis let S_k be the encoder state at time k. The bit u_k is associated with the transition from time $k-1$ to time k and causes 2 pathes to leave each state. The trellis states at level $k-1$ and at level k are indexed by the integer s' and s, respectively. The goal of the MAP algorithm is to provide us with

$$L(\hat{u}_k) = \log \frac{P(u_k = +1|\mathbf{y})}{P(u_k = -1|\mathbf{y})} = \log \frac{\sum\limits_{\substack{(s',s)\\ u_k=+1}} p(s', s, \mathbf{y})}{\sum\limits_{\substack{(s',s)\\ u_k=-1}} p(s', s, \mathbf{y})}. \quad (14)$$

The index pair s' and s determines the information bit u_k and the coded bits. The sum of the joint probabilities $p(s', s, \mathbf{y})$ in (14) is taken over all existing transitions from state s' to state s labeled with the information bit $u_k = +1$ or with $u_k = -1$, respectively. Assuming a memoryless transmission channel, the joint probability $p(s', s, \mathbf{y})$ can be written as the product of three independent probabilities [2],

$$\begin{aligned} & p(s', \mathbf{y}_{j<k}) \cdot p(s, \mathbf{y}_k|s') \cdot p(\mathbf{y}_{j>k}|s) \\ = \; & \underbrace{p(s', \mathbf{y}_{j<k})}_{} \cdot \underbrace{P(s \mid s') \cdot p(\mathbf{y}_k \mid s', s)}_{} \cdot \underbrace{p(\mathbf{y}_{j>k}|s)}_{} \\ = \; & \alpha_{k-1}(s') \quad \cdot \quad \gamma_k(s', s) \quad \cdot \quad \beta_k(s). \end{aligned}$$

Here $\mathbf{y}_{j<k}$ denotes the sequence of received symbols $\mathbf{y}_j$ from the beginning of the trellis up to time $k-1$ and $\mathbf{y}_{j>k}$ is the corresponding sequence from time $k+1$ up to the end of the trellis. The forward recursion of the MAP algorithm yields

$$\alpha_k(s) = \sum_{s'} \gamma_k(s', s) \cdot \alpha_{k-1}(s'). \quad (15)$$

The backward recursion yields

$$\beta_{k-1}(s') = \sum_{s} \gamma_k(s', s) \cdot \beta_k(s). \quad (16)$$

The branch transition probabilities are given by

$$\gamma_k(s', s) = p(\mathbf{y}_k \mid u_k) \cdot P(u_k). \quad (17)$$

Using the log-likelihoods the *a priori* probability $P(u_k)$ can be expressed as

$$P(u_k) = \left(\frac{e^{-L(u_k)/2}}{1 + e^{-L(u_k)}} \right) \cdot e^{u_k L(u_k)/2} = A_k \cdot e^{u_k L(u_k)/2}. \quad (18)$$

and, in a similar way, the conditioned probability

$$p(\mathbf{y}_k \mid u_k) = B_k \cdot e^{\frac{1}{2} \sum\limits_{\nu=1}^{n} L_c y_{k,\nu} x_{k,\nu}}. \quad (19)$$

for a convolutional code with rate $1/n$ and

$$p(\mathbf{y}_k \mid u_k) = B_{M_k} \cdot e^{-\frac{1}{2\sigma^2}|y_k - \sum\limits_{l=0}^{L} u_{k-l} h_l|^2} \quad (20)$$

for a binary input multipath channel with $L+1$ taps. The terms A_k and B_k in (18) and (19) are equal for all transitions from level $k-1$ to level k and hence will cancel out in the ratio of (14).

B. *A Simplification of the BCJR-Algorithm*

Several approximations of the BCJR algorithm have been studied, i.e. [10]. We will give another one using certain structures of a binary trellis and L-values. If one uses the approximation

$$\log \sum_i e^{L_i} \approx \max_i L_i \quad (21)$$

in (15) and (16) the forward and backward recursions of the BCJR algorithm mutate into two Viterbi algorithms running forth and back the terminated trellis. They produce the state metrics for the forward algorithm

$$M_{\alpha_{k-1}}(s') = \log \alpha_{k-1}(s') \quad (22)$$

and for the backward algorithm

$$M_{\beta_k}(s) = \log \beta_k(s). \quad (23)$$

Using again the approximation (21) the soft-output results in

$$\begin{aligned} L(\hat{u}_k) = \; & \max_{\substack{(s',s)\\ u_k=+1}} (M_{\alpha_{k-1}}(s') + \\ & \log p(\mathbf{y}_k|+1) + L(u_k)/2 + M_{\beta_k}(s)) \\ - \; & \max_{\substack{(s',s)\\ u_k=-1}} (M_{\alpha_{k-1}}(s') + \quad (24) \\ & \log p(\mathbf{y}_k|-1) - L(u_k)/2 + M_{\beta_k}(s)). \end{aligned}$$

For a binary trellis three different butterfly structures exist. For the structure where the two pathes with same u_k merge in one state s– this is the case for feedforward convolutional codes and tapped delay line channels– the first three terms in (25) form $M_{\alpha_k(s)}$ and the maximization is only over the states s:

$$\begin{aligned} L(\hat{u}_k) = \; & \max_{\substack{s\\ u_k=+1}} (M_{\alpha_k}(s) + M_{\beta_k}(s)) \\ - \; & \max_{\substack{s)\\ u_k=-1}} (M_{\alpha_k}(s) + M_{\beta_k}(s)) \quad (25) \end{aligned}$$

For the structure where the two pathes with same u_k leave one state s' –this is the case for feedback convolutional codes– the reverse is true and $k-1$ replaces k in the right side of Eqn.(25). A similar approach has been taken in [11].

In summary:

The BCJR algorithm for the mostly used binary terminated trellises can be closely approximated by

- Two VA algorithms running backwards and forwards
- using the update metric

$$\log p(\mathbf{y}_k|u_k) + c_k + u_k L(u_k)/2$$

where c_k is a suitable simplifying normalization constant independent of u_k

- A memory storing the metrics
- Add the forward-(α)- to the backward-(β)-metrics either to the right (k) or to the left (k-1) of the current bit u_k
- Find the maxima over the plus and minus states and subtract them to obtain the soft output.

Note, that the channel part of the update metric has the SNR as a factor, e.g. $4E_s/N_0$. Therefore, if the the SNR is very small the soft-output equals $L(u_k)$, only the a priori value as it should be.

C. Soft-in/soft-out decoder using the direct code

Although often very convenient, it is not necessary to use a trellis for decoding. A closed yet complex formula exists using all codewords of the code. Define

$$L(x_k; y_k) \triangleq \begin{cases} L_c y_k + L(u_k), & 1 \leq k \leq K, \\ L_c y_k, & K+1 \leq k \leq N. \end{cases} \tag{26}$$

then the soft-output is [5] $L(\hat{u}_k) =$

$$L(u_k) + L_c y_k + \underbrace{\log \frac{\sum\limits_{\substack{\mathbf{x} \in \mathcal{C} \\ x_k=+1}} \prod\limits_{\substack{j=1 \\ j \neq k}}^{N} e^{L(x_j;y_j)\, x_j/2}}{\sum\limits_{\substack{\mathbf{x} \in \mathcal{C} \\ x_k=-1}} \prod\limits_{\substack{j=1 \\ j \neq k}}^{N} e^{L(x_j;y_j)\, x_j/2}}}_{L_e(\hat{u}_k)}. \tag{27}$$

The second part is the extrinsic information which can be shown to vanish when $L_c \to 0$, leaving us with the a priori value alone. Again we can use the approximation (21) to obtain a logmax approximation.

D. Soft-in/soft-out decoder using the dual code

For a high rate code the number of the code words $\mathbf{x}^\perp$ of the dual code $\mathcal{C}^\perp$ is smaller than the number of the codewords of the direct code. Therefore we better decode the original transmitted codeword as shown in [5] with the dual code where we use the soft bit

$$\lambda_j = \tanh(L(x_j; y_j)/2)$$

to obtain the soft output

$$L(\hat{u}_k) = L_c y_k + L(u_k) + \underbrace{\log \frac{1 + \sum\limits_{i=2}^{2^{N-K}} \prod\limits_{\substack{j=1, j \neq k \\ x_{ij}^\perp = -1}}^{N} \lambda_j}{1 - \sum\limits_{i=2}^{2^{N-K}} (-x_{ik}^\perp) \prod\limits_{\substack{j=1, j \neq k \\ x_{ij}^\perp = -1}}^{N} \lambda_j}}_{L_e(\hat{u}_k)}. \tag{28}$$

Several approximations of this equation have been discussed, such as using only the minimum distance codewords of the dual code or working in the $\log \lambda$ domain to transfer multiplications into additions. The latter modification is used in [13] on a 16 bit fixed point computer for decoding the (1023,1013) Hamming code in a turbo decoder.

Especially interesting is the modification of (28) for the decoding of high rate convolutional codes via the reciprocal dual code as shown in [12]. This efficient implementation allows the use of high rate non-punctured convolutional codes with rates such as 23/24 as constituent codes for parallel concatenation, leading to high-rate 'turbo' codes.

IV. Applications of the Turbo-Principle

The application of soft-in/soft-out decoders to serial and parallel concatenated coding/detection schemes offers the possibility of iterative decoding within the inner and between the inner and outer decoders.

A. Parallel Concatenation of Codes

This is the classical and well explored field where the turbo principle has been used first by [1] and [4]. The literature is too extensive to be referenced here. We only refer to two special issues [31] and [32]. The so-called turbo code encodes the information twice by systematic codes, the second after interleaving, similar as in a product code. Extrinsic information is exchanged between the two soft-in/soft-out decoders as shown in the tutorial example in section 2-C. The following results and observations have been derived with this setup:

- Block and feedback convolutional codes can be used
- Although the minimum distance and the asymptotic gain of these codes is not too good, they perform surprisingly well at low to medium channel SNR. Consequently a leveling out of the waterfall curve is observed.
- At a BER of 10^{-5} the appropriate Shannon limit is approched by 0.5 dB for rates

around 0.5 and by 0.27 dB at rates around 0.98. Very simple component codes such as Hamming and low density parity check codes have been applied, although with interleavers of size 10^4 to 10^6.

- Block, pseudorandom and trained interleavers have been used to combat the leveling out behavior.
- Distance spectrum and performance analysis of these codes has been performed with random [6] interleavers leading to average upper bounds for codes averaged and with unrealizable perfect interleavers [8] leading to the best possible performance.
- No analysis is yet available to give upper and lower bounds of the BER performance after a certain number of iterations for a fixed code and interleaver. Therefore our understanding of these codes is limited.

B. Serial Concatenation of Codes

Serial concatenation of codes employing the turbo principle has been extensively investigated by several authors using different kind of inner and outer codes. The theory can be found in [7]. The standard concatenated schemes use convolutional codes as inner and Reed- Solomon codes as outer codes. With turbo decoding it turned out that binary outer codes yield much better results [9], [24]. A system which is comparable to the deep space and ETSI/MPEG digital TV standard performs better by a margin of 1.6 dB with inner parallel concatenated codes and outer BCH codes and turbo iterations between the serially concatenated inner and outer codes [24].

C. Source-Controlled Channel Decoding

In an asymptotic sense source and channel coding as well as decoding can be treated separately according to Shannon's famous separation theorem. Practical source coding schemes still contain residual redundancy which can be utilized by the channel decoder in a turbo feedback as source a posteriori information. The channel FEC code is the inner code and the statistically correlated source bits constitute the outer code. The a priori information needed by the inner FEC decoder is gained from inter- and intra-frame bit dependencies in the outer source decoder, see [18] and [19]. Gains of 1 to 2 dB in channel SNR are achieved for existing source coding schemes such as GSM speech coding and still image compression by such a turbo feedback. Furthermore the soft bit produced by (7) is useful in generating a soft and graceful degradation while reconstructing the source signal [20].

D. Equalization of Coded Data

A multipath channel is nothing but an inner encoder of rate 1. The optimal equalizer is a MAP estimator which uses the metric increment

$$-\frac{1}{2\sigma^2} \mid y_k - \sum_{l=0}^{L} x_{k-l} h_l \mid^2. \qquad (29)$$

The MAP or suboptimal MAP or SOVA equalizer therefore has to know the tap values $\{h_l\}$ from a channel estimator using training bits or decided bits. The turbo feedback interaction of this equalizer as inner decoder with the outer decoder is shown in Fig.4. Care has to be taken to pass only extrinsic- respectively channel-information of the coded bits between the two MAP decoders. This turbo equalizer princi-

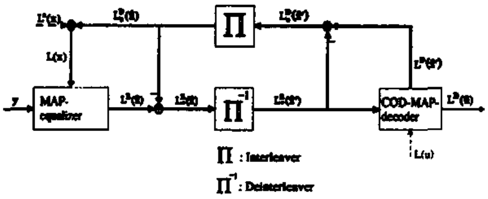

Fig. 4. Turbo equalizer scheme

ple was first proposed in [21]. We investigated several multipath channels and found that the channels which have the worst performance in classical equalization (without turbo feedback) gained most by the iterations. A dramatic example taken from [22] with an outer convolutional code with rate 1/2 and memory 4 is shown in Fig.5. The total interleaver size was 4096 bits and no ajustment factors as in [21] were used. We applied the same method to the GSM equalizer with time-varying taps of a mobile system. The average gain by the turbo equalization is less pronounced, because the really bad (strongly encoding) tap sets occur not too often. However, it helps a lot in bad static situations.

E. Turbo Iteration in CDMA Systems (IS-95)

Even CDMA systems can be viewed as serial concatenation: The inner code is for instance as in the IS-95 system a Hadamard code. For this code a very efficient soft-in/soft-out decoder has been developed and turbo iterations with the outer MAP or SOVA- decoder of convolutional codes were performed [26]. The figure 6 shows the setup for IS-95 downlink where the inner

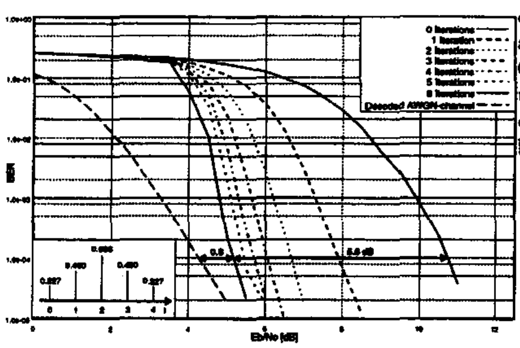

Fig. 5. Performance of a coded system: Transmission over the shown multipath channel with turbo iterations between MAP eqalizer and CODMAP decoder

encoder uses a Hadamard code. The iterative decoder setup is similar to the one with the equalizer as inner decoder and we achieved a gain of 1.3 dB through a few turbo iterations. Similar results are obtained for the noncoherent uplink case.

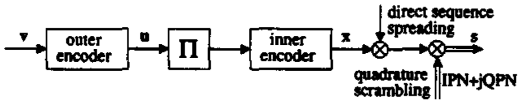

Fig. 6. CDMA system with inner and outer serial concatenation, cf. IS-95

F. Multiuser Detection with Turbo-Feedback

A classical DS spread spectrum system with K users employs spreading sequences $c_k(t)$ with a correlation matrix $\mathbf{R} = (R_{k,k'})$, where the diagonal elements are $R_{k,k} = 1$. After synchronous transmission we receive after the matched filter

$$y_k = \int_0^T r(t) a_k c_k(t) \mathrm{d}t,$$

where

$$r(t) = \sum_{k=1}^{K} a_k b_k c_k(t)$$

and a_k is the channel gain factor. Of course the optimal detector would be the joint MLSE detector and its known suboptimal approximations as decribed with further references in [23]. We will describe a suboptimal iterative turbo scheme where in the first iteration the slightly modified matched filter output $L(b_k|y_k)$ of the inner decoder (despreader) $\mathrm{DEC_i}$ is supplied to the outer decoder $\mathrm{DEC_o}$. The outer decoder $\mathrm{DEC_o}$ is also a soft-in/soft-out decoder as shown in Fig.1. Fig.7 describes the complete CDMA system as described first in [25]: After the first round of decoding with all the K outer decoders we can use two types of turbo feedback for iterating:

- The extrinsic L-value $L_e(\hat{b}_k)$ which is the soft-output $L(\hat{b}_k)$ minus the soft input $L(b_k|y_k)$. At the first iteration this extrinsic estimate is uncorrelated with the input. Therefore using this value as a priori value for all code bits of the outer code improves decoding. For subsequent iterations the correlation causes a diminishing return.

- The soft output of all the other users $L(\hat{b}_{k'})$ is used to calculate the expected value of $\hat{b}_{k'}$ via (8), where only a simple sigmoid nonlinearity (7) is required. After interleaving this soft bit in the range $(-1, +1)$ is weighted by its channel gain $a_{k'}$ and spreading code correlation value $R_{k,k'}$. The sum of all user channels with $k' \neq k$ is then subtracted from the matched filter output y_k. After weighting with L_{c_k} and adding the new a priori value we have the new soft-output of the inner decoder and are ready for the next round of iteration. Note, that wrong decisions of the outer decoder usually have small L-values and small $E\{b_{k'}\}$ and do not contribute to the feedback. Therefore error propagation is avoided. This iterative scheme is a low complexity approximation to the full MLSE joint detection scheme.

Very similar ideas for multiuser detection have been independently and carefully treated in [15]. Moher uses a tractable multiuser channel model with one correlation parameter and his work gives a lot of insight in the theme of this chapter. In [16] he gives impressive simulation results showing turbo feedback asymptotically eliminates the multiuser degradation for FEC coded systems under a variety of channel conditions.

G. Coded Modulation with Turbo detection

It is not too surprising that the turbo principle should be also useful in coded modulation, be it Imai's multilevel coding or Ungerboeck's trellis coded modulation. It is straightforward just to replace codes used before in multilevel modulation by turbo codes. More appropriate and a sophisticated use of the turbo principle is the turbo-coded modulation scheme by Robertson and Woerz [27] where channel and

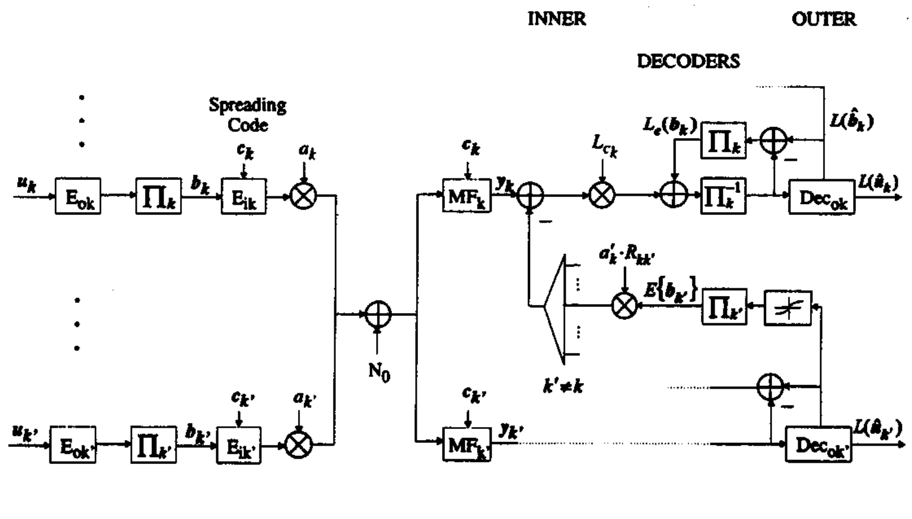

Fig. 7. Coded CDMA with double feedback in the decoder

extrinsic information is exchanged between two soft-in/ soft-out decoders in a clever way. They achieve a notable gain over Ungerboeck's scheme for 8-PSK and for higher level QAM formats. For further discussion of turbo coded modulation we refer to [28].

H. Miscellaneous Items

Cross-Entropy:
Already Battail had mentioned that the cross- or Kullback entropy is a useful means to look at decoding schemes. The turbo decoding process for product codes was recognized by Moher as early as 1993 as a variation of the principle of minimum cross entropy. In his later thesis [15] he gives very illuminating graphical explanations of the iterative turbo process using cross-entropy. This leads to an information theory based better understanding of the turbo principle an area also treated by Caire, Taricco and Biglieri as well as Shamai and Verdu [31]. Cross-entropy has been further used as a stop criterion for the iterations [5] which reduces the number of necessary iterations considerably.

Tanner Graphs:
Tanner graphs [29] which are connected by the interleaver connections have been used to explain the parallel and serial concatenated turbo decoding process by Wiberg, Loeliger and Forney [30]. The networks can also be evaluated by applying the belief propagation algorithm known from artificial intelligence. They can be further evaluated by the so-called min-sum algorithm which is based on a similar approximation as used above in (5) or (21). Forney mentioned that this variation of a decoding algorithm goes back to the sixties where it was used by Gallager and Massey.

REFERENCES

[1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes (1)," *Proc. IEEE International Conference on Communication (ICC)*, Geneva, Switzerland, May 1993, pp. 1064-1070.

[2] L.R. Bahl, J. Cocke,F. Jelinek and J. Raviv, "Optimal decoding of linear Codes for minimzing symbol error rate," *IEEE Transactions on Information Theory*, vol. IT-20, 1974, pp. 284-287.

[3] G. Battail, M. C. Decouvelaere and P. Godlewski, "Replication decoding," *IEEE Transactions on Information Theory*, vol. IT-25, May 1979, pp. 332-345.

[4] J. Lodge, R. Young, P. Hoeher, and J. Hagenauer, "Separable MAP 'filters' for the decoding of product and concatenated codes," *Proc. IEEE International Conference on Communication (ICC)*, Geneva, Switzerland, May 1993, pp. 1740-1745.

[5] J. Hagenauer, E. Offer and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. on Information Theory*, vol. 42, no.2, pp. 429–445, March 1996.

[6] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Trans. on Information Theory*, vol. 42, no.2, pp. 409-429, March 1996.

[7] S. Benedetto, D. Divsalar, G. Montorsi and F. Pollara, "Analysis, design and iterative decoding of serially concatenated codes with interleavers," *Special isue : IEEE Journal on Special Areas in Communications* , S. Benedetto, D. Divsalar and J. Hagenauer, Eds., vol. JSAC-15, to appear January 1998.

[8] Y. V. Svirid, "Weight distributions and bounds for turbo codes," *European Transactions on Telecommunications*, Vol. 6, No. 5, pp. 507 – 511, Sept - Oct. 1995.

[9] J. Andersen, "'Turbo' coding for deep space applications," *Proc. IEEE 1995 Int. Symposium on Information Theory*, Whistler, Canada, September 1995, p. 36.

[10] P. Robertson, P. Hoeher, E. Villebrun "Optimal and suboptimal maximum a posteriori algorithms suitable for turbo decoding," *European Transactions on Telecommunications*, ETT vol. 8, pp.119-125, March/Apr. 1997.

[11] B. Tang, G. Pottie "Soft output bidirectional Viterbi decoding ," *in submission* Jan. 1997.

[12] S. Riedel, "Symbol-by-symbol MAP decoding of convolutional codes using reciprocal dual convolutional codes," *Proc. of the IEEE Int. Symp. on Inf. Theory*, Ulm, Germany, June 1997, p. 226 , also accepted for *IEEE Trans. on Information Theory.*

[13] H. Nickl, J. Hagenauer, F. Burkert, "Approaching Shannon's capacity limit by 0.27 dB using Hamming codes in a 'turbo'- decoding scheme." *Proc. of the IEEE Int. Symp. on Inf. Theory*, Ulm, Germany, June 1997, p. 226 , also accepted for *IEEE Communications Letters, Sept. 1997.*

[14] J. Hagenauer, E. Offer and L. Papke, "Matching Viterbi decoders and Reed–Solomon decoders in a concatenated system," *Reed–Solomon codes and their applications*, S. Wicker and V. K. Bhargava, Eds., New York: IEEE Press, 1994, ch. 11, pp. 242-271.

[15] M. L. Moher, "Cross-entropy and iterative decoding" *PhD Thesis Dep. of Systems and Computer Engineering, Carleton University*, Ottawa, Canada May 1997, 141 pages

[16] M. L. Moher, "Turbo-based multiuser detection," *Proc. of the IEEE Int. Symp. on Inf. Theory*, Ulm, Germany, June 1997, p. 195.

[17] J. Hagenauer, "Soft–In/Soft–Out: The benefits of using soft decisions in all stages of digital receivers", in *Proc. 3rd Int. Workshop on DSP Techniques applied to Space Communications*, ESTEC Noordwijk, The Netherlands, Sept. 1992.

[18] J. Hagenauer, "Source-controlled channel decoding," *IEEE Transactions on Communications*, vol. COM-43, no. 9, pp. 2449–2457, September 1995.

[19] A. Ruscitto and Th. Hindelang, "Joint source and channel decoding in the full-rate GSM system," submitted to *IEEE Communications Letter*, May 1997.

[20] T. Fingscheidt and P. Vary, "Robust speech decoding: A universal approach to bit error concealment ", *Proc. IEEE ICASSP'97*, Munich, Germany, Vol. 3, pp. 1667-1701, June 1997.

[21] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier, A. Glavieux, "Iterative Correction of Intersymbol Interference: Turbo–Equalization," *European Transactions on Telecommunications*, Vol. 6, No. 5, pp. 507 – 511, Sept - Oct. 1995.

[22] G. Bauch, H.Khorram, J. Hagenauer, "Iterative equalization and decoding in mobile communications systems," in *Proceedings*

of EPMCC'97, Bonn, Germany, September 1997.

[23] P. Hoeher, "On channel coding and multiuser detection for DS-CDMA", *Proc. IEEE ICUPC'93*, Ottawa, Canada, pp. 641-646, Oct. 1993.

[24] F. Burkert and J. Hagenauer, "A serial concatenated coding scheme with iterative 'turbo'- and feedback decoding.," in *Proceedings of the Int. Symp. on Turbo Codes & Related Topics*, Brest, France, 3.-5. September 1997.

[25] J. Hagenauer, "Forward error correcting for CDMA systems," in *Proceedings of the IEEE Fourth International Symposium on Spread Spectrum Techniques and Applications ISSSTA'96*, Mainz, Germany, pp. 566–569, September 1996.

[26] R. Herzog, A. Schmidbauer, J. Hagenauer, "Iterative decoding and despreading improves CDMA-systems using M-ary modulation and FEC," in *Proceedings of the IEEE Int. Conf. on Commun. ICC'97*, Montreal, Canada, pp. , June 1997.

[27] P. Robertson and Th. Woerz, "A novel bandwidth efficient coding scheme employing turbo codes," *Proc. IEEE International Conference on Communication (ICC)*, Dallas, USA, June 1996.

[28] P. Robertson, "An overview of bandwidth efficient turbo coding schemes.," in *Proceedings of the Int. Symp. on Turbo Codes & Related Topics*, Brest, France, 3.-5. September 1997.

[29] R.M. Tanner,"A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. IT-27, pp 533-547, Sep. 1981.

[30] G. D. Forney, "The forward-backward algorithm," *Proc. of the 1996 Allerton Conference*, Allerton, Illinois, Sep. 1996.

[31] "Focus on iterative and turbo decoding ," *Special issue : European Trans. on Telecommun. ETT*, E. Biglieri and J. Hagenauer, Eds., vol. 6 no. 5, Sept./ Oct. 1995.

[32] "Sailing toward channel capacity ," *Special isue : IEEE Journal on Special Areas in Communications* , S. Benedetto, D. Divsalar and J. Hagenauer, Eds., vol. JSAC-15, to appear January 1998.