The Turbo Principle in Communication Systems
Introduction

Joachim Hagenauer

Institute for Communications Engineering (LNT)
Munich University of Technology (TUM)
D-80290 München, Germany

Email: Hagenauer@ei.tum.de

Outline Basic Turbo Techniques

- The Turbo Principle and its Applications
- Log-Likelihood Ratios (LLR) and the APP Decoders
- The BCJR Algorithm
- Introduction to Parallel Decoding (Turbo Codes)
- Introduction to Serial Decoding

Outline Applications

- Turbo applications: Coded Equalization of Multipath Channels
- Turbo Applications: Pre-coded QAM with Irregular Channel Codes
- Turbo Applications: Coded MIMO Systems
- Turbo Applications: Source Channel Coding with Variable Length Codes (VLC)
- Turbo Applications: Source Channel Coding for continuous sources
- Turbo Applications: Analog Turbo Decoders
- Turbo Applications: Turbo Source Compression
- Conclusions

Introduction

History:

- 1948: Shannon's absolute limits in communications, e.g. 0.2 dB in E_b/N_0 for binary codes with rate 1/2 on AWGN channel
- 1962: Gallager's low density parity check codes with iterative decoding
- 1966: Forney: Concatenated codes
- before 1993: Concatenated codes (Viterbi plus RS codes) approach Shannon's limit by by 2.5 dB and with iterations by 1.5 dB.
- 1993: Berrou, Glavieux and Thitimajshima: Turbo decoding approaches Shannon's limit by 0.5 dB.
- 1995: Douillard, Glavieux, Berrou et al: Turbo equalization
- 1997: Turbo principle recognized as general method in communications systems
- 2001: Chung, Forney, Richardson, Urbanke :Iterative decoding of Irregular LDPC Codes within 0.0045 dB of Shannon limit

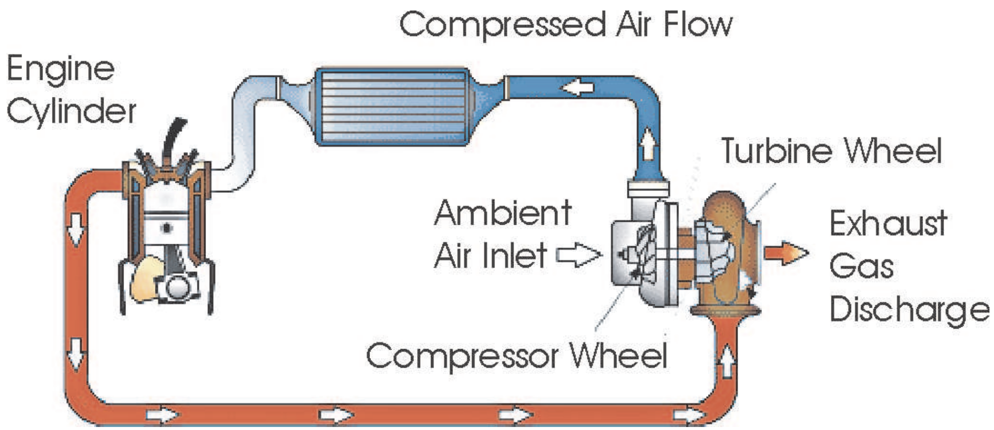
Introduction

The Turbo Principle comprises...

- ... a communication system with serial and/or parallel concatenations of components
- ... a posteriori probability (APP) symbol-by-symbol decoders/detectors
- ... soft-in/soft-out decoders/detectors
- ... interleavers between the components
- ... exchange of extrinsic information between components in the form of probabilities or log-likelihood ratios

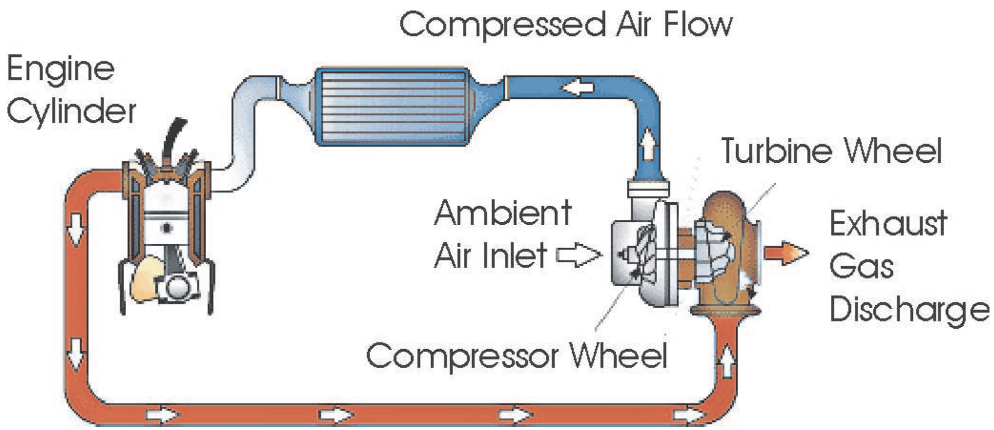
The Turbo Principle ...

... in mechanics

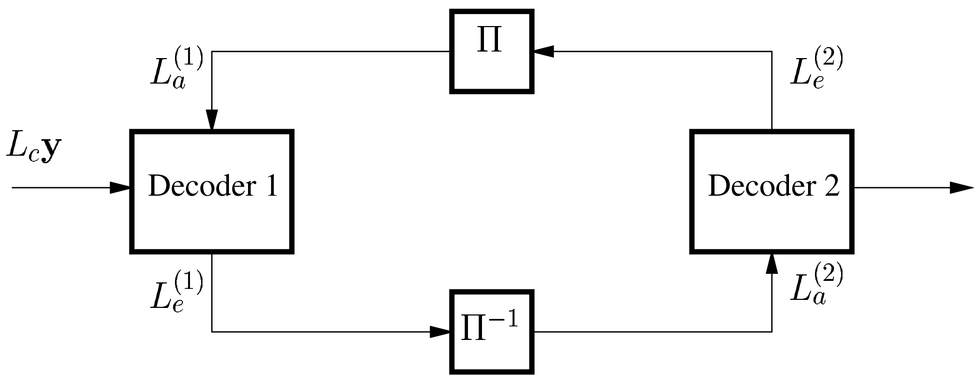


The Turbo Principle ...

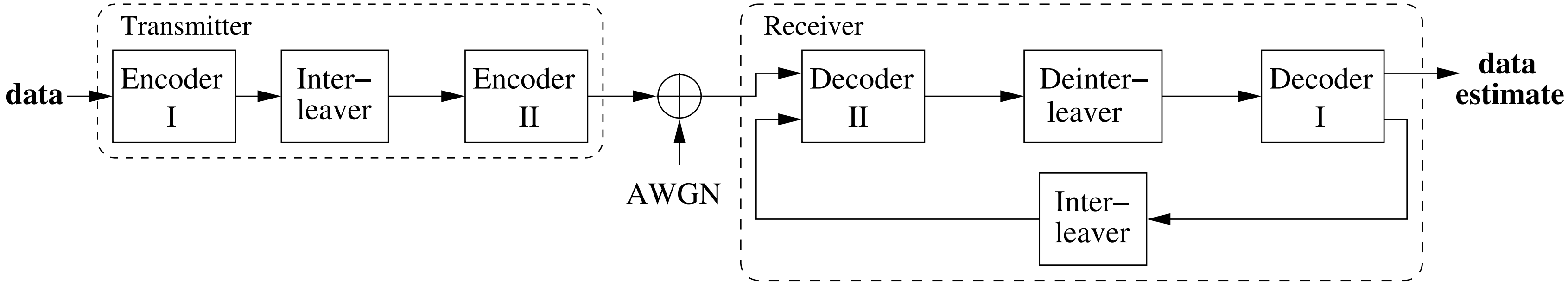
... in mechanics



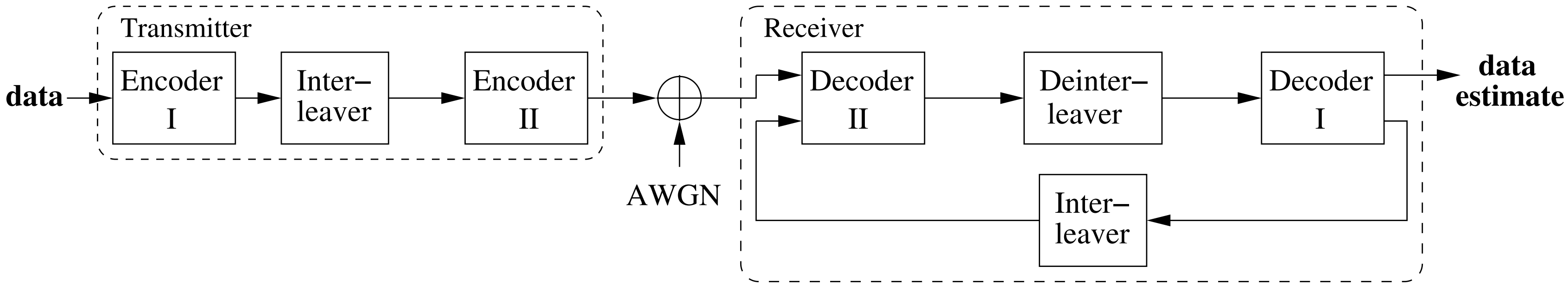
... in communications



Serial Concatenation



Examples for serial concatenation in communications systems



configuration	en-/decoder I (outer code)	en-/decoder II (inner code)
serial code concat.	FEC en-/decoder	FEC en-/decoder
turbo equalization	FEC en-/decoder	Multipath channel/detector
turbo BiCM	FEC en-/decoder	Mapper/demapper
turbo DPSK	convolutional code	DPSK accumulator
turbo MIMO	FEC en-/decoder	Mapper & MIMO detector
turbo source-channel	source encoder	FEC en-/decoder
LDPC code/decoder	check nodes	variable nodes
RA code/decoder	repetition code	scrambler

Examples for standardized applications for the turbo principle systems

- Rate 1/3 PCC Code in UMTS
- IEEE 802 Wireless LAN
- NASA Deep space standard
- LDPC for ESA Standard Digital Video Broadcasting (DVB)

The Turbo Principle in Communication Systems
Log-Likelihood Values and APP Decoders

Joachim Hagenauer

Institute for Communications Engineering (LNT)
Munich University of Technology (TUM)

D-80290 München, Germany

(Short Course 2004)

Basics of Turbo Decoding

Log-Likelihood Ratios and the APP Decoders:

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 (LLR) or L-value of the binary variable is

$$L(u) = \ln \frac{P(u=+1)}{P(u=-1)} \tag{1}$$

with the inverse

$$P(u=\pm 1) = \frac{e^{\pm L(u)/2}}{e^{+L(u)/2} + e^{-L(u)/2}}. \tag{2}$$

Note: The sign of $L(u)$ is the hard decision and the magnitude $|L(u)|$ is the reliability of this decision.

The soft bit and the binary sum

The soft bit $\lambda(u)$ is

$$\begin{aligned}\lambda(u) = \mathrm{E}\{u\} &= (+1)\cdot P(u=+1) + (-1)\cdot P(u=-1)\\ &= \tanh(L(u)/2).\end{aligned}$$

GF(2) addition $u_1 \oplus u_2$ of two independent binary random variables:

$$\mathrm{E}\{\mathrm{u}_1 \cdot \mathrm{u}_2\} = \mathrm{E}\{\mathrm{u}_1\}\mathrm{E}\{\mathrm{u}_2\} = \lambda(u_1)\cdot\lambda(u_2).$$

L value of the sum :

$$L(u_1 \oplus u_2) = 2\tanh^{-1}(\tanh(L(u_1)/2)\cdot\tanh(L(u_2)/2)) = L(u_1) \boxplus L(u_2).$$

with the **boxplus** $\boxplus$ abbreviation.

LNT

The boxplus element and its approximation (1)

The boxplus element

$$L(u_1 \oplus u_2) = 2\tanh^{-1}(\tanh(L(u_1)/2) \cdot \tanh(L(u_2)/2)) =$$

can be approximated by

$$L(u_1) \boxplus L(u_2) \approx \mathrm{sign}(L(u_1)) \cdot \mathrm{sign}(L(u_2)) \cdot \min\{|L(u_1)|, |L(u_2)|\}$$

Examples:

$$-3.0 \boxplus +0.5 \approx -0.5$$

$$-4.0 \boxplus -1.2 \approx +1.2$$

$$L(u) \boxplus \; 0 = 0$$

$$L(u) \boxplus +\infty = L(u)$$

The boxplus element and its approximation (2)

The boxplus element

$$L(u_1 \oplus u_2) = 2\tanh^{-1}(\tanh(L(u_1)/2) \cdot \tanh(L(u_2)/2)) =$$

can be exactly expressed by its approximation and a correction term

$$L(u_1) \boxplus L(u_2) = \mathrm{sign}(L(u_1)) \cdot \mathrm{sign}(L(u_2)) \cdot \min\{|L(u_1)|, |L(u_2)|\} - \ln \frac{1 + e^{-|\,|L(u_1)|-|L(u_2)|\,|}}{1 + e^{-|\,|L(u_1)|+|L(u_2)|\,|}}$$

If one of the the two magnitudes is dominant the correction term disappears. It is necessary when both magnitudes are the same and has a maximum value of $\ln 2$.

A similar approximation is known as the max* operation (Jacobian logarithm):

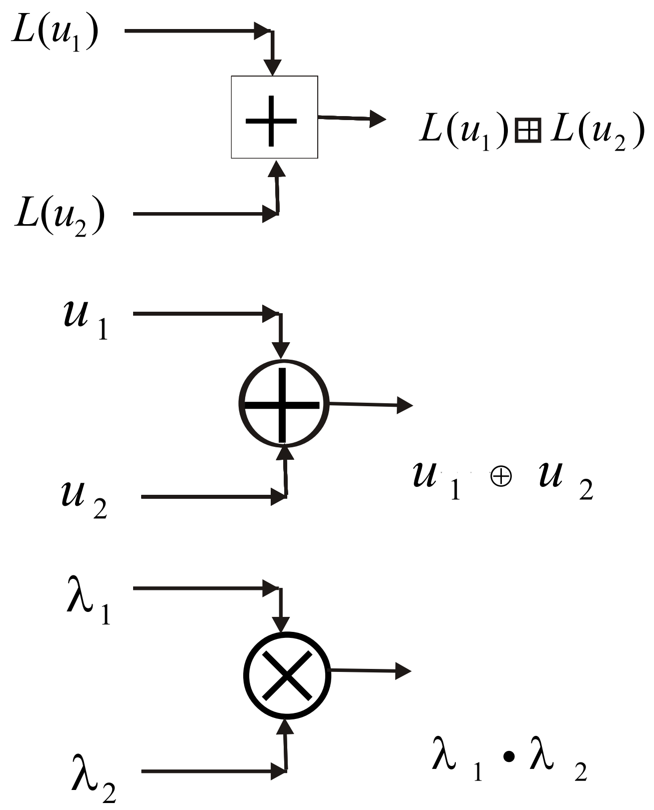
$$\ln(e^{L_1} + e^{L_2}) = \max\{L_1, L_2\} + \ln(1 + e^{-|L_1 - L_2|})$$

The binary XOR, the boxplus and the softbit operation

The boxplus element

$$L(u_1 \oplus u_2) = L(u_1) \boxplus L(u_2) = 2\tanh^{-1}(\tanh(L(u_1)/2) \cdot \tanh(L(u_2)/2)) =$$

corresponds to the binary XOR operation and the softbit multiplication:



Transmission and combining after fading/AWGN channels

The a posteriori probability (APP) in $y = ax + n$ is

$$P(x|y) = \frac{p(y|x)P(x)}{p(y)} \tag{3}$$

with the pdf

$$p(y|x) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{\frac{-(y-ax)^2}{2\sigma_c^2}} \tag{4}$$

The complementary APP LLR equals

$$L_{CH} = L(x|y) = \ln \frac{P(x=+1|y)}{P(x=-1|y)} = L_c \cdot y + L(x). \tag{5}$$

$L(x)$ is the a priori LLR of x and L_c is the channel state information (CSI):

$$L_c = \frac{2a}{\sigma_c^2} = 4aE_s/N_0 \tag{6}$$

For statistically independent transmission

$$L(x|y_1, y_2) = L_{c_1} y_1 + L_{c_2} y_2 + L(x). \tag{7}$$

Practical Usefulness of Log-Likelihood Calculation

Did it rain in NY at 1:00 pm today?
A Yes (rain!) is binary coded as $+1$,transmitted over an unreliable link.
Two rain detection devices measured :

$$x_1 = +1$$

$$x_2 = +1$$

Additional a priori value is available: From Farmer's Almanac:
Probability of rain in NY today is 75%

$$L(x) = \ln(0.75/0.25) = +1.1$$

	transmitted value x	received value y	channel state L_c	$L_c y$
link 1	+1.0	-1.5	2.0	-3.0
link 2	+1.0	+0.9	3.0	+2.7
a priori				+1.1

For statistically independent information

$L(x|y_1, y_2) = L_{c_1} y_1 + L_{c_2} y_2 + L(x) = +0.8$ with 31% error : rain in NY !!.

The extrinsic information as a LLR

Assume a parity check equation of statistically independently transmitted bits x_j

$$\sum_{j=1}^{j=N}\oplus\, x_j = 0.$$

Then the extrinsic bit x_i equals

$$x_i = \sum_{j=1, j\neq i}^{j=N}\oplus\, x_j$$

and consequently the extrinsic LLR for this bit given the APP LLR's $L(x_j|y_j)$ of all the other bits equals

$$L_E(x_i) = \sum_{j=1, j\neq i}^{j=N}\boxplus\, L(x_j|y_j)$$

<u>Example</u>:
SPC code, $N = 3$, with $L(x_2|y_2) = -0.3$, $L(x_3|y_3) = -5.5$.
Then the extrinsic LLR for the first bit is

$$L_E(x_1) = -0.3 \boxplus -5.5 \approx +0.3.$$

The general formula for a soft output as a LLR

Assume a transmission of a vector $\mathbf{x}$ of length N
over a channel and received as a vector $\mathbf{y}$.

We are interested in the LLR of the n-th bit x_n The a posteriori LLR of the N bits is

$$L(\hat{x}_n) = L(x_n|\mathbf{y}) = \ln \frac{\Sigma_{x_n=+1} P(\mathbf{x}|\mathbf{y})}{\Sigma_{x_n=-1} P(\mathbf{x}|\mathbf{y})} = \ln \frac{\Sigma_{x_n=+1} e^{\ln P(\mathbf{x}|\mathbf{y})}}{\Sigma_{x_n=-1} e^{\ln P(\mathbf{x}|\mathbf{y})}} \quad (8)$$

The metric can be expanded in channel and a priori parts

$$\ln P(\mathbf{x}|\mathbf{y}) = \ln p(\mathbf{y}|\mathbf{x}) + \ln P(\mathbf{x}) - \ln p(\mathbf{y}) \quad (9)$$

Note:
If all pathes have the same length, we can ignore the last term. However, if we search paths in a tree with different length of the paths, we cannot ignore the term $-\ln p(\mathbf{y})$ during the search.

The Channel Part of the Metric

Transmission over a real AWGN oder multiplicative fading channel with

$$y_n = a_n \cdot x_n + w_n$$

The receiving process is corrupted by real valued AWGN with i.i.d. noise samples and pdf

$$p(w) = \frac{1}{\sqrt{(\pi 2\sigma_w^2)}} \mathrm{e}^{-\frac{w^2}{2\sigma_w^2}}$$

with

$$\sigma_w^2 = \frac{N_0}{2E_s}$$

After cancellation of all parts common in the denominator and denominator of the metric we can use for the channel part in the soft output formula

$$\ln p(\mathbf{y}|\mathbf{x})$$

is the correlation metric

$$\sum_{n=1}^{N} \ln p(y_n|x_n) = \sum_{n=1}^{N} \frac{2E_s}{N_0} \cdot a_n \cdot y_n \cdot x_n$$

The Channel Part of the Metric

Transmission over an intersymbol interference (ISI, multipath) channel with L-taps

$$y_n = \sum_{k=0}^{L-1} h_k s_{n-k} + w_n. \tag{10}$$

The receiving process is corrupted by complex-valued AWGN with i.i.d. noise samples and pdf

$$p(w) = \frac{1}{\pi 2\sigma_w^2} \mathrm{e}^{-\frac{|w|^2}{2\sigma_w^2}}$$

This leads to the channel part of the metric

$$\ln p(\mathbf{y}|\mathbf{x}) = -N \cdot \ln(\pi 2\sigma_w^2) - \sum_{n=1}^{N} \frac{|y_n - \sum_{k=0}^{L-1} h_k s_{n-k}|^2}{2\sigma_w^2}$$

The A Priori Part of the Metric

With the statistical independence from the interleaver we have

$$\ln P(\mathbf{x}) = \sum_{n=1}^{N} \ln P(x_n),$$

With the L-values we obtain

$$\ln P(x_n) = x_n L(x_n)/2 \; - \ln(\mathrm{e}^{+L(x_n)/2} + \mathrm{e}^{-L(x_n)/2})$$

where the last two terms can be deleted if all pathes have equal length. This leads for the **soft output** to

$$L(\hat{x}_n) = L(x_n) + \ln \frac{\sum_{x_n=+1} e^{\;\ln p(y_n|x_n) + \sum_{j=1,j\neq n}^{N} \ln p(y_j|x_j) + x_j L(x_j)/2}}{\sum_{x_n=-1} e^{\;\ln P(y_n|x_n) + \sum_{j=1,j\neq n}^{N} \ln P(y_j|x_j) + x_j L(x_j)/2}}$$

The A Priori Part, the Channel Part and the Extrinsic Part

With the fading or AWGN channel we obtain finally from

$$L(\hat{x}_n) = L(x_n) + \ln \frac{\Sigma_{x_n=+1}\, e^{\ \ln P(y_n|x_n) + \Sigma_{j=1,j\neq n}^{N} \ln P(y_j|x_j) + x_j L(x_j)/2}}{\Sigma_{x_n=-1}\, e^{\ \ln P(y_n|x_n) + \Sigma_{j=1,j\neq n}^{N} \ln P(y_j|x_j) + x_j L(x_j)/2}}$$

the three parts

$$L(\hat{x}_n) = L(x_n) + \frac{4E_s}{N_0} a_n y_n + \ln \frac{\Sigma_{x_n=+1}\, e^{\Sigma_{j=1,j\neq n}^{N} \ln P(y_j|x_j) + x_j L(x_j)/2}}{\Sigma_{x_n=-1}\, e^{\Sigma_{j=1,j\neq n}^{N} \ln P(y_j|x_j) + x_j L(x_j)/2}}$$

The last part is called the extrinsic part of the soft-output.It represents the influence of all the other bits on the current bit with index n.

With the fairly tight approximation $\ln \ \Sigma_i e^{\lambda_i} = \max_i \lambda_i$ we obtain the so-called **max-log approximation** for the extrinsic part

$$\max_{x_n=+1} \sum_{j=1,j\neq n}^{N} \ln P(y_j|x_j) + x_j L(x_j)/2 - \max_{x_n=-1} \sum_{j=1,j\neq n}^{N} \ln P(y_j|x_j) + x_j L(x_j)/2$$

The Turbo Principle in Communication Systems

The APP Decoder on a Trellis with the Bahl-Cocke-Jelinek-Raviv (BCJR) Algorithm

Joachim Hagenauer

Institute for Communications Engineering (LNT)
Munich University of Technology (TUM)

D-80290 München, Germany

(Short Course 2004)

The APP Decoder on a Trellis: The BCJR Algorithm

The algorithm is due to Bahl-Cocke-Jelinek-Raviv based on earlier work by Welsh and Baum.

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 paths 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

$$L(\hat{u}_k) = \log \frac{P\left(u_k = +1|\mathbf{y}\right)}{P\left(u_k = -1|\mathbf{y}\right)} = \log \frac{\sum\limits_{\substack{(s',s)\\ u_k=+1}} p(s', s, \mathbf{y})}{\sum\limits_{\substack{(s',s)\\ u_k=-1}} p(s', s, \mathbf{y})}. \tag{11}$$

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 (11) is taken over all existing transitions from state s' to state s labelled with the information bit $u_k = +1$ or with $u_k = -1$, respectively.

The Transition Metrics

Assuming a memoryless transmission channel, the joint probability $p(s', s, \mathbf{y})$ can be written as the product of three independent probabilities following BCJR 1974,

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

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.

Forward and Backward Recursion

The forward recursion of the MAP algorithm yields

$$\alpha_k(s) = \sum_{s'} \gamma_k(s', s) \cdot \alpha_{k-1}(s'). \tag{12}$$

The backward recursion yields

$$\beta_{k-1}(s') = \sum_{s} \gamma_k(s', s) \cdot \beta_k(s). \tag{13}$$

The branch transition probabilities are given by

$$\gamma_k(s', s) = p(\mathbf{y}_k \mid u_k) \cdot P(u_k). \tag{14}$$

LNT

The Transition Metrics as LLR

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

$$P(u_k) = \left(\frac{1}{e^{+L(u_k)/2} + e^{-L(u_k)/2}}\right) \cdot e^{u_k L(u_k)/2} = A_k \cdot e^{u_k L(u_k)/2}. \qquad (15)$$

and, in a similar way, the conditioned probability

$$p(\mathbf{y}_k \mid u_k) \;=\; B_k \cdot e^{\frac{1}{2}\sum\limits_{\nu=1}^{n} L_c y_{k,\nu} x_{k,\nu}}. \qquad (16)$$

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

$$p(\mathbf{y}_k \mid u_k) \;=\; B_{M_k} \cdot e^{-\frac{1}{2\sigma^2}(y_k - \sum\limits_{l=0}^{L} x_{k-l} h_l)^2}. \qquad (17)$$

for a binary input multipath channel with $L+1$ taps. The terms A_k and B_k in (15) and (16) are equal for all transitions from level $k-1$ to level k and hence will cancel out in the ratio of (11).

A Simplification of the BCJR-Algorithm

An approximations of the BCJR algorithm is given by using the approximation

$$\log \sum_i e^{L_i} \approx \max_i L_i \tag{18}$$

Then in (12) and (13) 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

$$M_{\alpha_{k-1}}(s') = \log \alpha_{k-1}(s') \tag{19}$$

and for the backward algorithm

$$M_{\beta_k}(s) = \log \beta_k(s). \tag{20}$$

For the forward algorithm we get

$$M_{\alpha_k}(s) = \max_{s'}\{M_{\alpha_{k-1}}(s') + \log \gamma_k(s', s)\} \tag{21}$$

and for the backward algorithm

$$M_{\beta_{k-1}}(s') = \max_{s}\{M_{\beta_k}(s) + \log \gamma_k(s', s)\} \tag{22}$$

LNT

A Simplification of the BCJR-Algorithm contd

Using again the approximation (18) the soft-output results in

$$
\begin{aligned}
L(\hat{u}_k) = & \max_{\substack{(s',s)\\ u_k=+1}} \left(M_{\alpha_{k-1}}(s') + \log p(\mathbf{y}_k| + 1) + L(u_k)/2 + M_{\beta_k}(s)\right) \\
- & \max_{\substack{(s',s)\\ u_k=-1}} \left(M_{\alpha_{k-1}}(s') + \log p(\mathbf{y}_k| - 1) - L(u_k)/2 + M_{\beta_k}(s)\right) .
\end{aligned}
\tag{23}
$$

The soft-output of the simplification with the feedforward trellis

For a binary trellis three different butterfly structures exist.

For the structure where the two paths 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 (24) form $M_{\alpha_k(s)}$ and the maximization is only over the states s:

$$
\begin{aligned}
L(\hat{u}_k) \; = \; & \max_{\substack{s \\ u_k=+1}} \left(M_{\alpha_k}(s) + M_{\beta_k}(s)\right) \\
- \; & \max_{\substack{s) \\ u_k=-1}} \left(M_{\alpha_k}(s) + M_{\beta_k}(s)\right)
\end{aligned}
\tag{24}
$$

The simplified BCJR Algorithm

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 to the right (k) 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 SNR is very small the soft-output equals $L(u_k)$, only the a priori value as it should be.

Further suboptimal and simplified soft-in/soft-out Algorithm

- Battail algorithm
- The soft-output Viterbi algorithm (SOVA) by Hagenauer/Hoeher 1998
- It is an add-on feature to the VA and can be turned on and off for individual bits
- It has the lowest complexity of all soft-in/soft-out algorithm
- It delivers too optimistic (too large) L-values.
- Several variations of the SOVA exist

The Turbo Principle in Communication Systems
Parallel Concatenation

Joachim Hagenauer

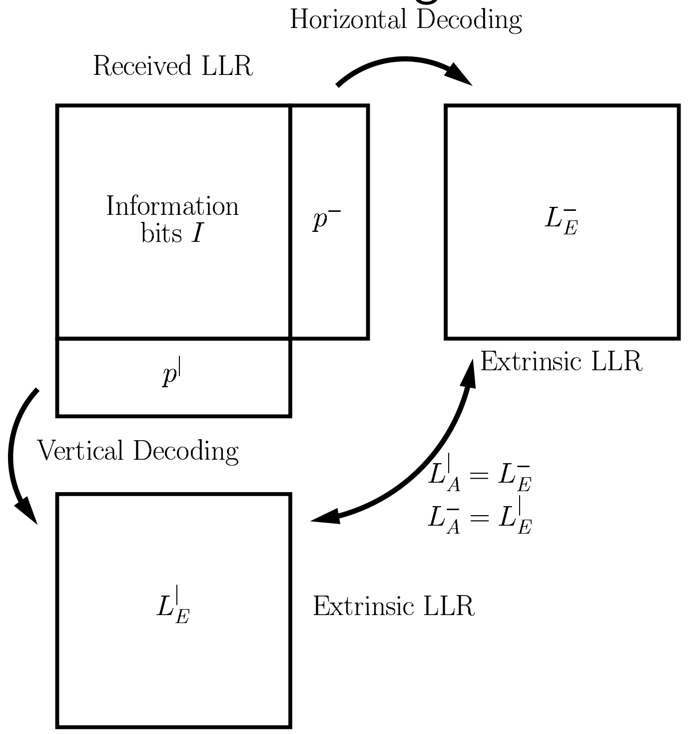
Institute for Communications Engineering (LNT)
Munich University of Technology (TUM)

D-80290 München, Germany

(Short Course 2004)

Principle of Turbo Decoding for a parallel concatenated scheme.

Exchange of extrinsic information between horizontal and vertical decoding



A First Example of PCC Turbo Decoding

u_{11}	u_{12}	p_{13}^{-}
u_{21}	u_{22}	p_{23}^{-}
$p_{31}^{\vert}$	$p_{32}^{\vert}$	

(a) Codeword

0	0	0
0	1	1
0	1	

(b) Coded values

$+0,5$	$+1,5$	$+1,0$
$+4,0$	$+1,0$	$-1,5$
$+2,0$	$-2,5$	

(c) Received values

$+1,0$	$+0,5$
$-1,0$	$-1,5$

(d) Extrinsic information after horizontal decoding

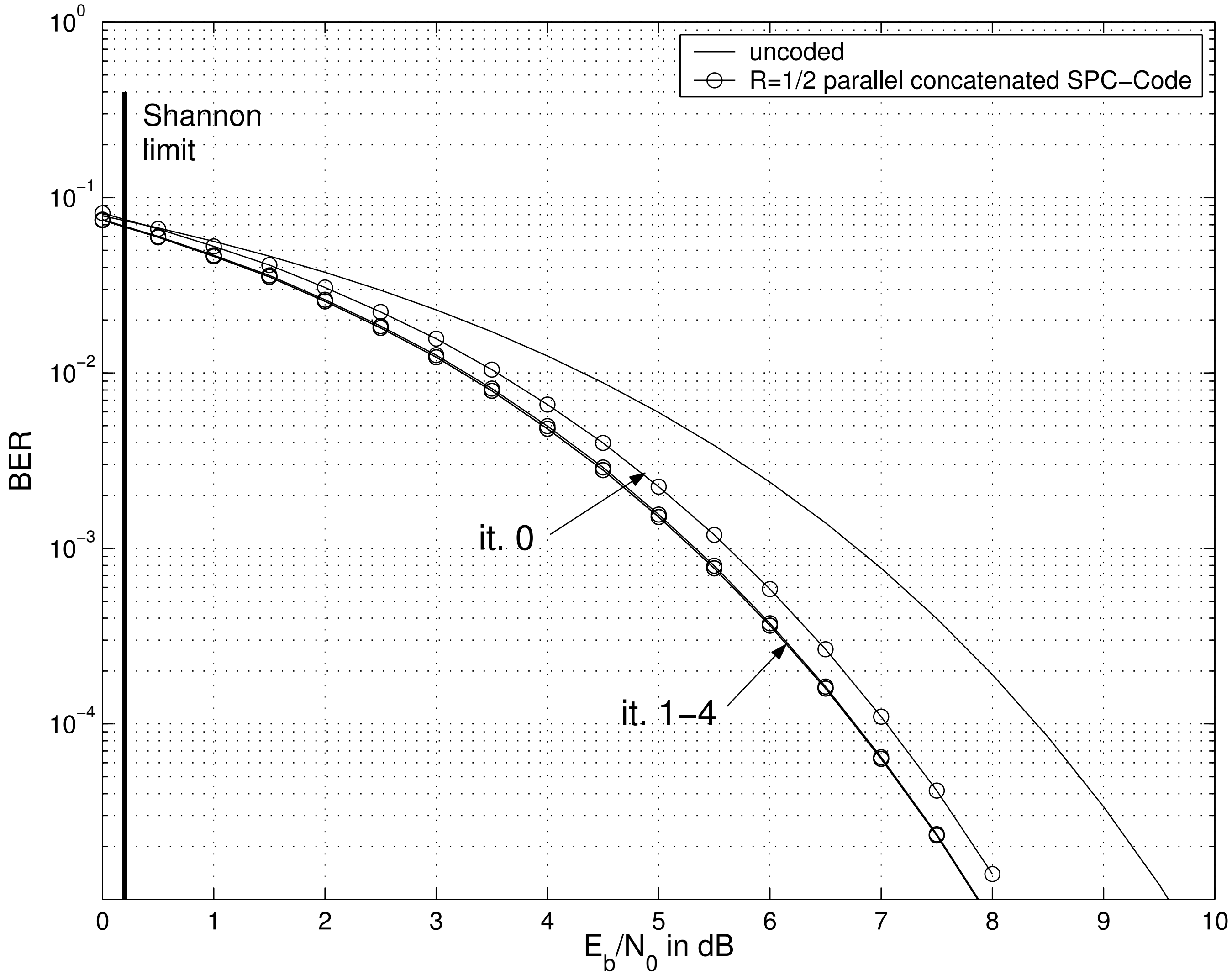
$+2,0$	$+0,5$
$+1,5$	$-2,0$

(e) Extrinsic information after vertical decoding

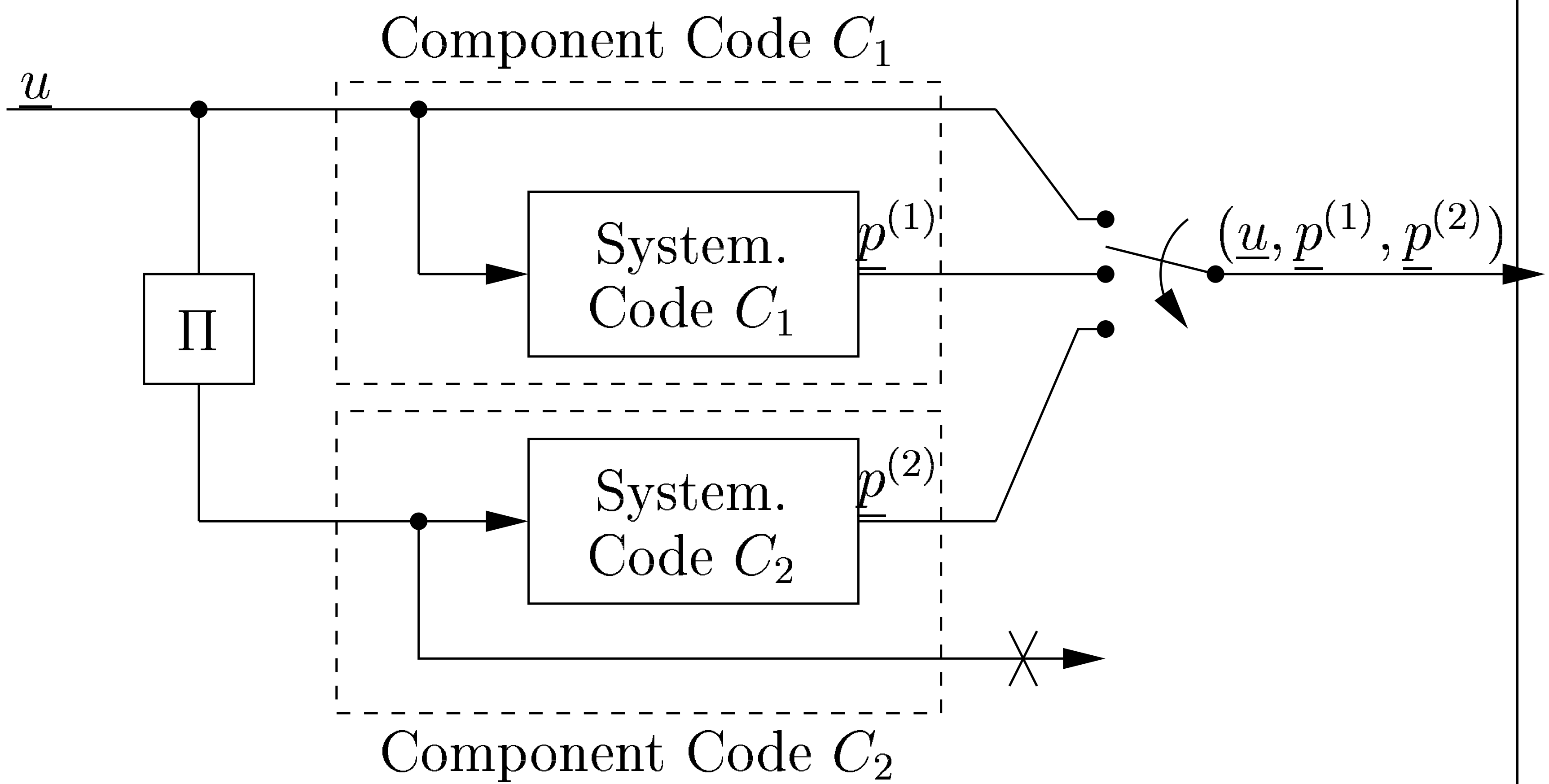
$+3,5$	$+2,5$
$+4,5$	$-2,5$

(f) Soft output after horizontal and vertical decoding

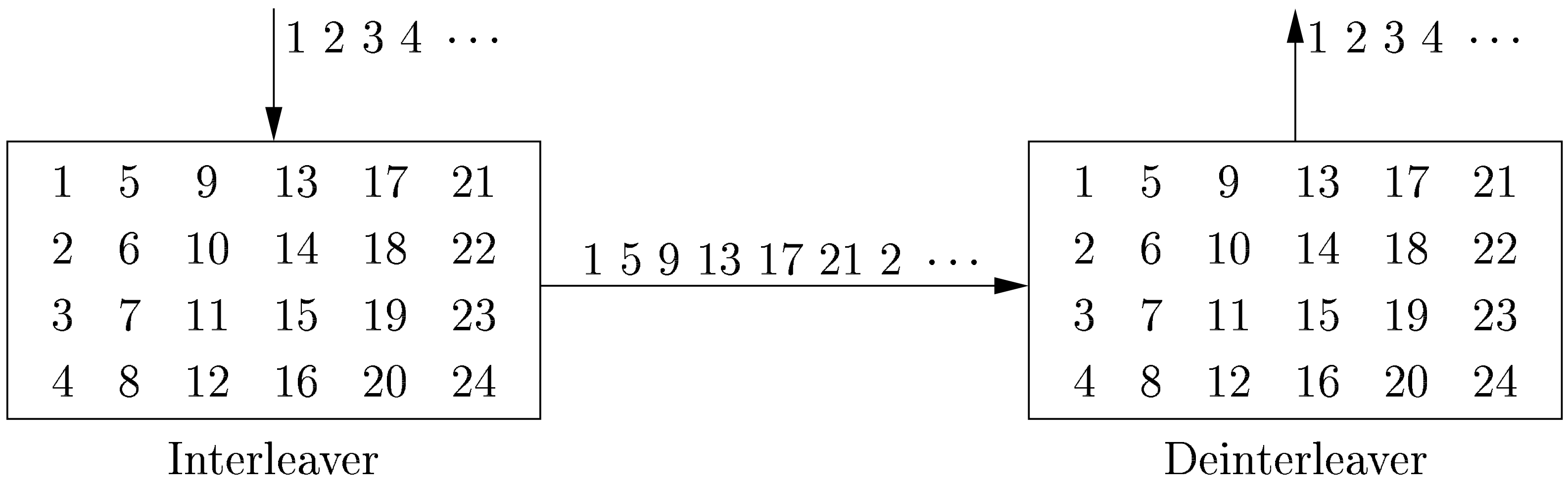
The Performance of the First Example of PCC Turbo Decoding



The general parallel concatenated Code PCC Turbo Code

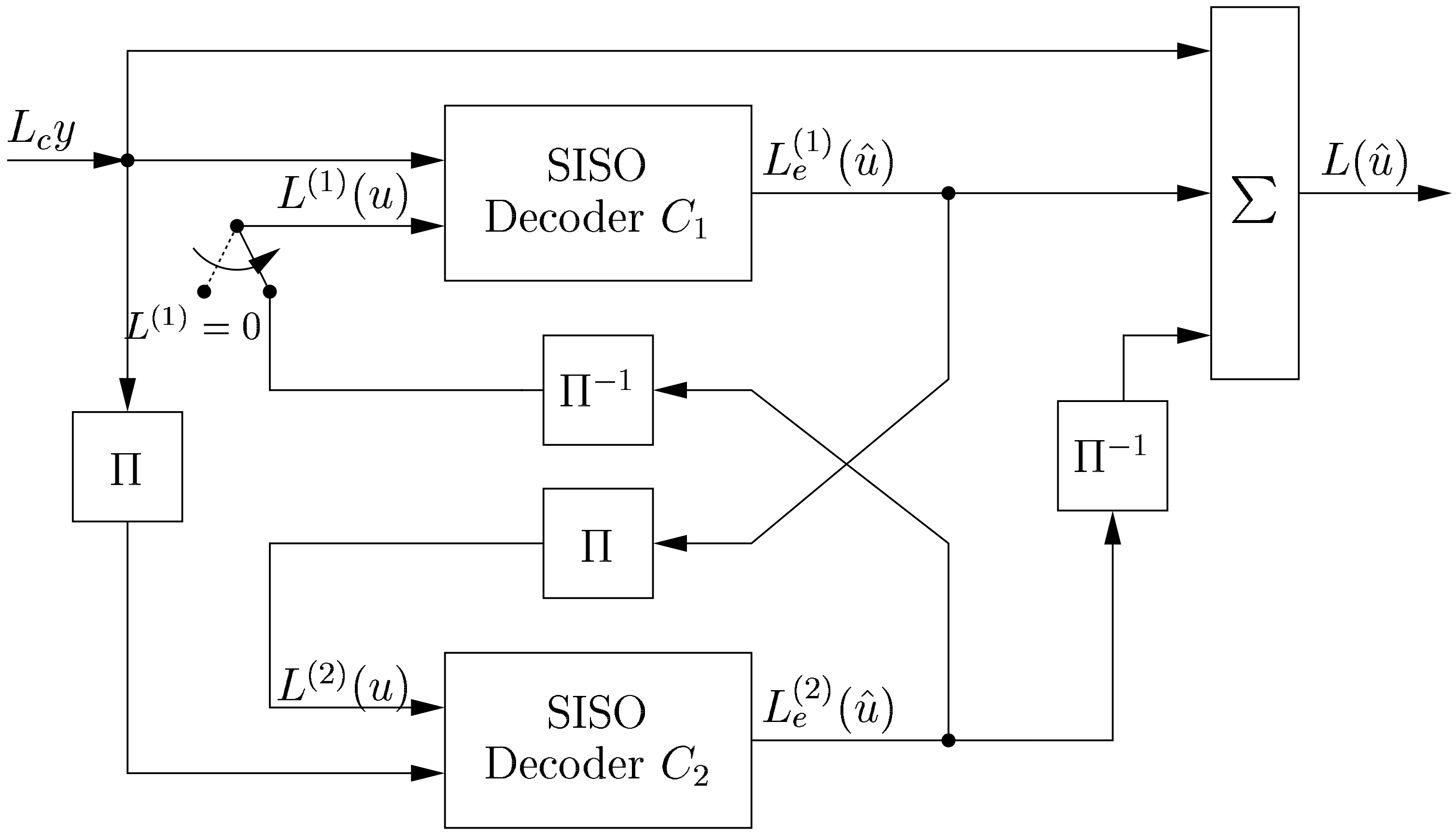


The principle of the Interleaver and Deinterleaver



The Decoder of the Parallel Concatenated Code PCC Turbo Code

Exchange of extrinsic information between horizontal (direct) and vertical (interleaved) decoding



Showing the chaotic behavior of Turbo decoding in a demonstration

- The turbo decoder decodes a parallel concatenated rate $1/2$ code with memory 2, rate $2/3$ convolutional code as constituent codes
- Block interleaver: size 20 x 20 $=$ 400 information bits, 800 transmitted bits
- Decoder: SOVA algorithm with L-values
- Shown are the soft-output L-values of the information bits after each half iteration
- Display shows 20 x 20 interleaver matrix with

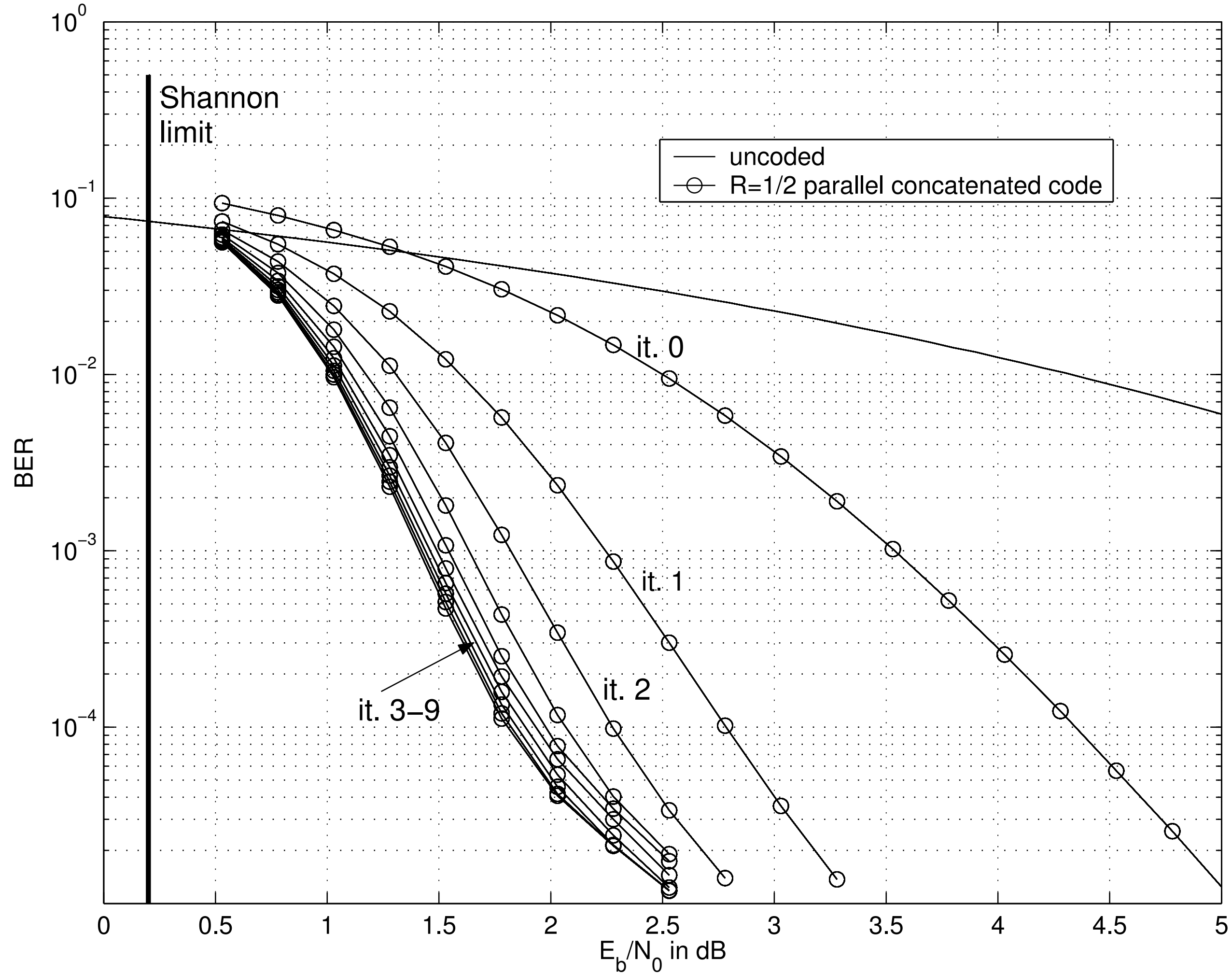
Red Circles for wrong bits
Green circles for correct bits

- Diameter of circles is the reliability (magnitude of L-values)
- Goal of decoding:

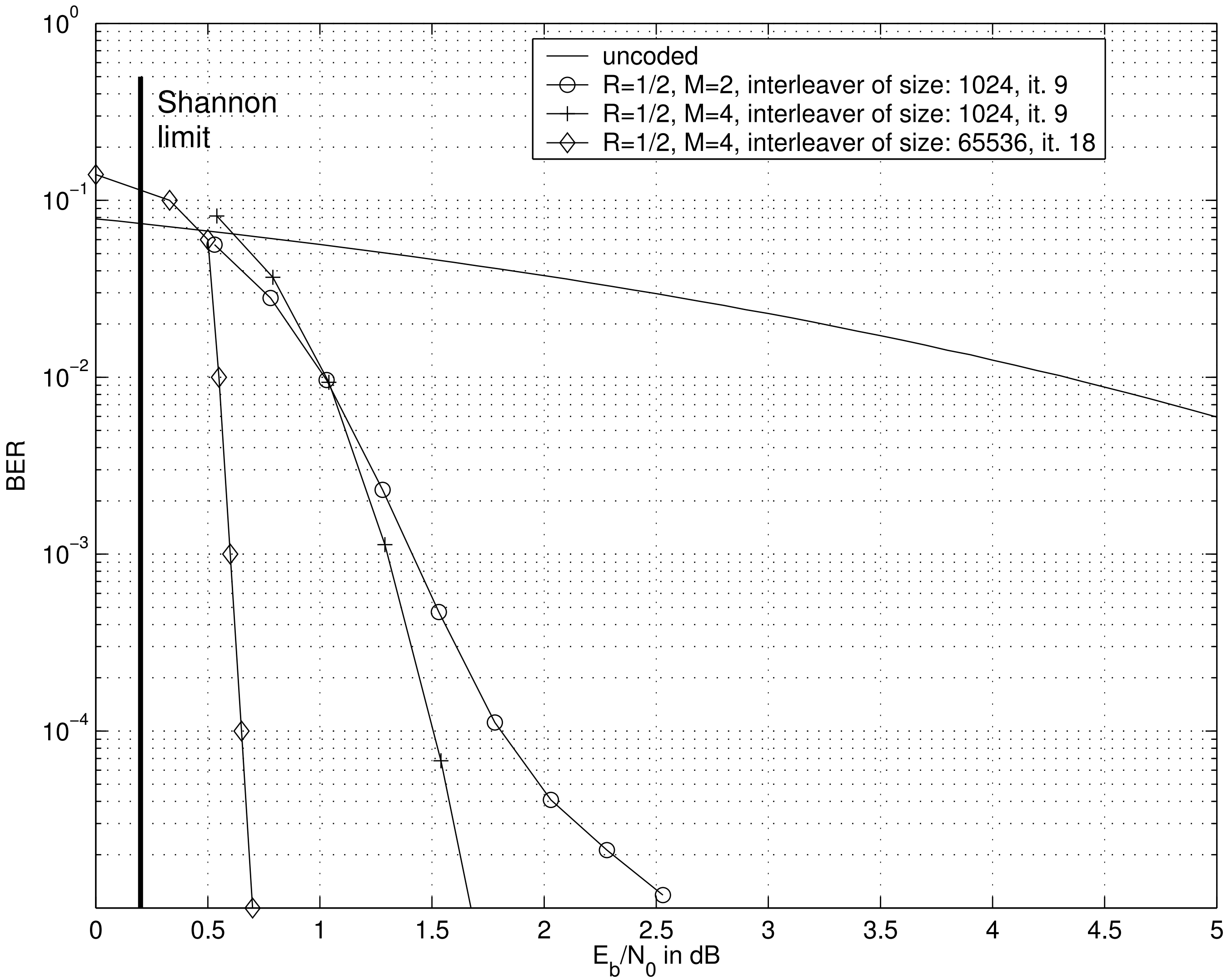
Big green circles !!!

Performance of Decoder of a PCC Turbo Code

Rate 1/2, constituent code: rate 2/3, memory 2, interleaver size 1024



Influence of the Interleaver Size for a PCC Turbo Code



The Turbo Principle in Communication Systems
Serial Concatenation

Joachim Hagenauer

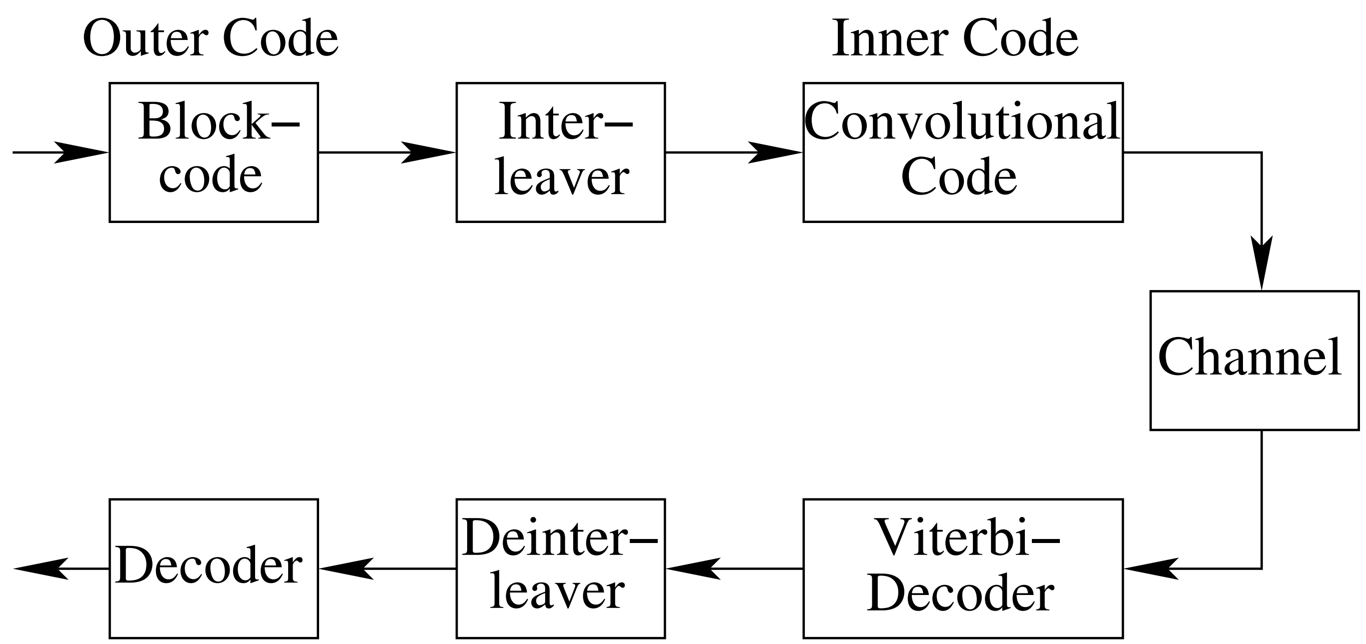
Institute for Communications Engineering (LNT)
Munich University of Technology (TUM)

D-80290 München, Germany

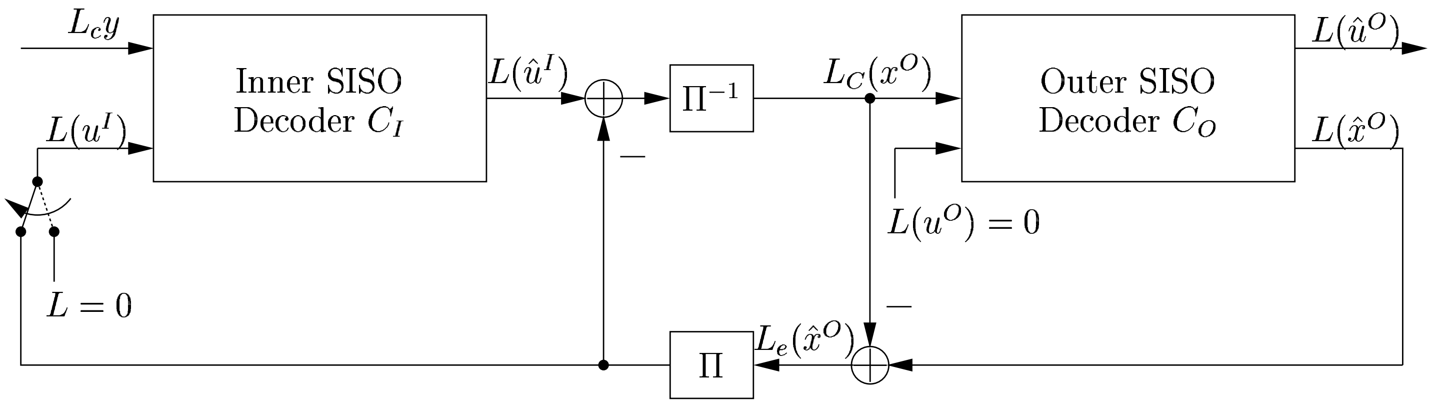
(Short Course 2004)

LNT

Principle of Turbo Coding for a serially concatenated scheme.

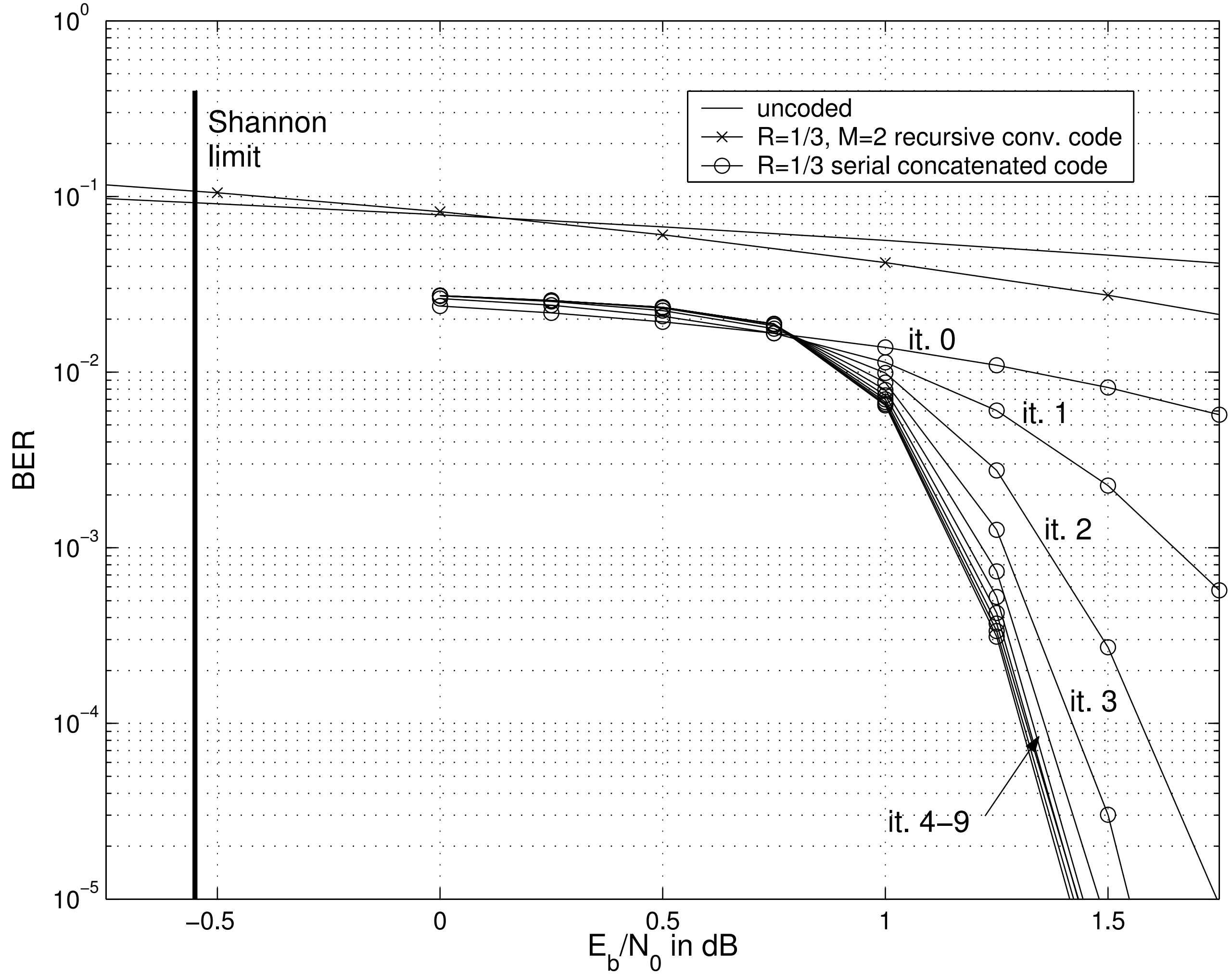


The Decoder for a Serially Concatenated scheme.



The Performance of a Serially Concatenated Scheme

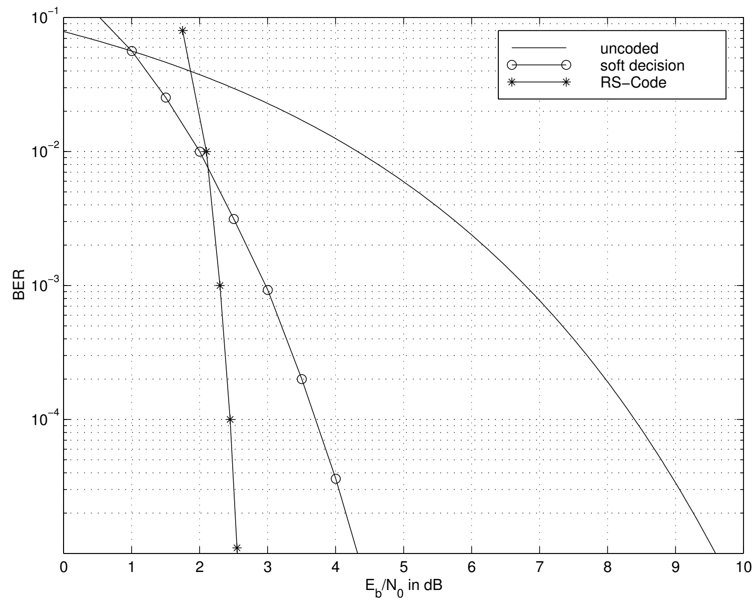
Serial Concatenation of two small convolutional codes



LNT

The Performance of a Serially Concatenated scheme

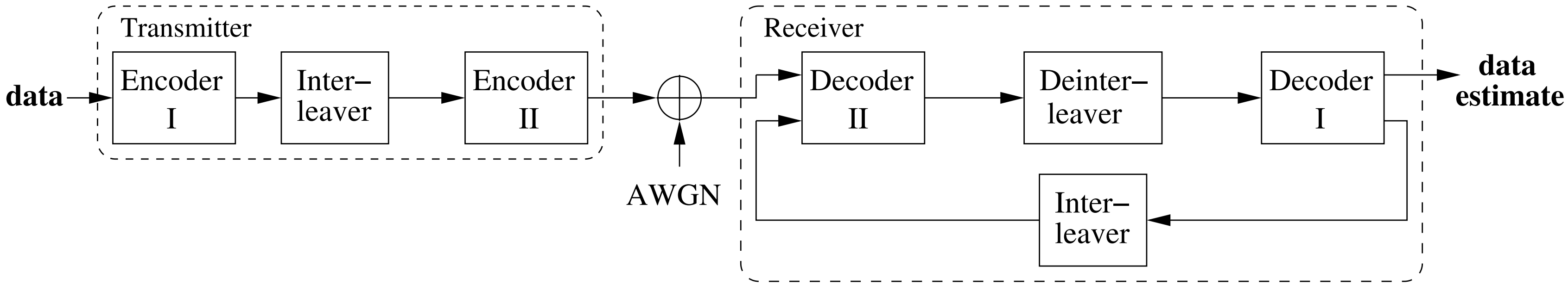
Serial Concatenation without iteration of an
Outer Reed-Solomon Code (255,223) over GF(2^8)
and an Inner Convolutional Code, Rate 1/2 and Memory 6:



With iterations between the RS decoder and the convolutional decoder (state pinning of correctly fe back decoded symbols) we achieved 1 dB further gain!!!

Offer, Hagenauer ICC 1993.

More Serially Concatenated Schemes



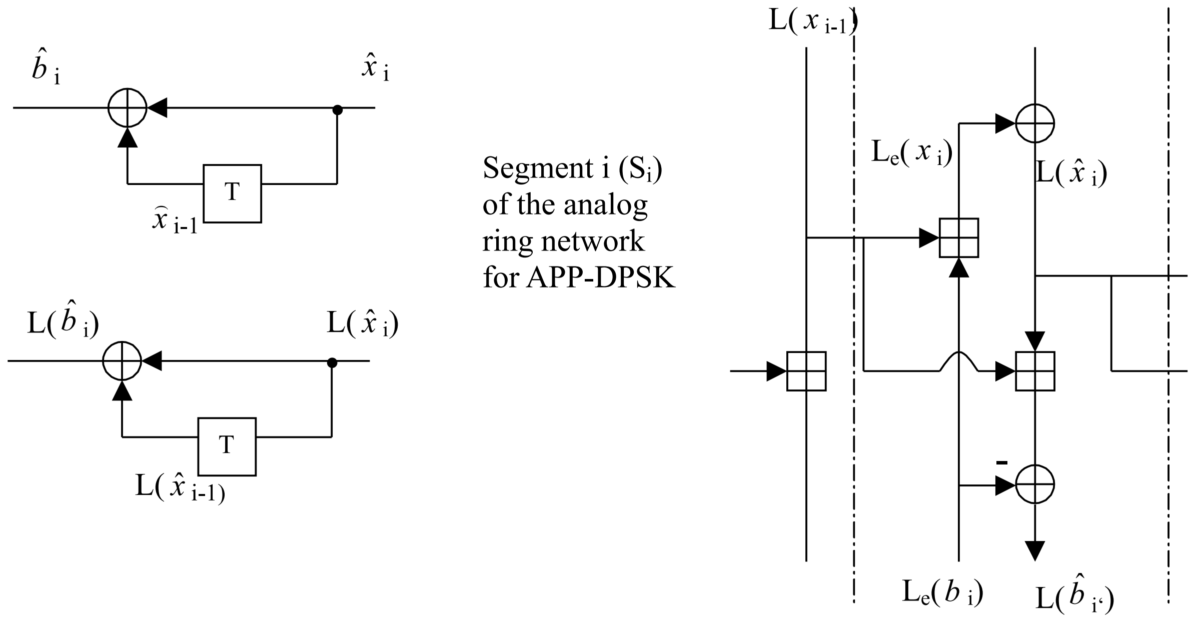
configuration	en-/decoder I (outer code)	en-/decoder II (inner code)
serial code concat.	FEC en-/decoder	FEC en-/decoder
turbo equalization	FEC en-/decoder	Multipath channel/detector
turbo BiCM	FEC en-/decoder	Mapper/demapper
turbo MIMO	FEC en-/decoder	Mapper & MIMO detector
turbo DPSK	FEC en-/decoder	DPSK 2 state trellis decoder
turbo source-channel	source encoder	FEC en-/decoder
LDPC code/decoder	check nodes	variable nodes

Serial concatenation of tailbiting convolutional codes and DPSK

- A block of information symbols is encoded without overhead by a tail-biting convolutional code (TBCC)
- Several codewords are then bit-wise interleaved
- DPSK Modulation is applied
- View the system as a ring for the TBCC and another ring for the DPSK modulation connected by an interleaver ring
- DPSK Demodulator and convolutional decoder operate in sequence when realized with digital processors
  or simultaneously as analog circuits (see section Analog Decoding)

Serial concatenation of tailbiting convolutional codes and DPSK

APP-Decoder for DPSK with forward and backward loop.

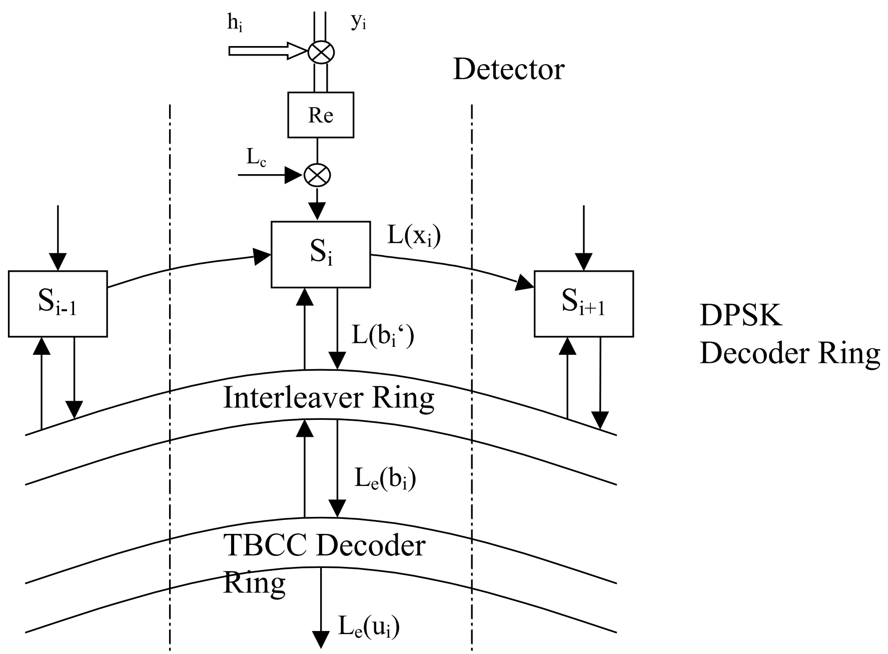


Since $x_i = b_i \oplus x_{i-1}$ and consequently at the receiver $\hat{b}_i = \hat{x}_i \oplus \hat{x}_{i-1}$ we can realize the DPSK decoder by

$$L(\hat{b}_i) = L(\hat{x}_i) \boxplus L(\hat{x}_{i-1}).$$

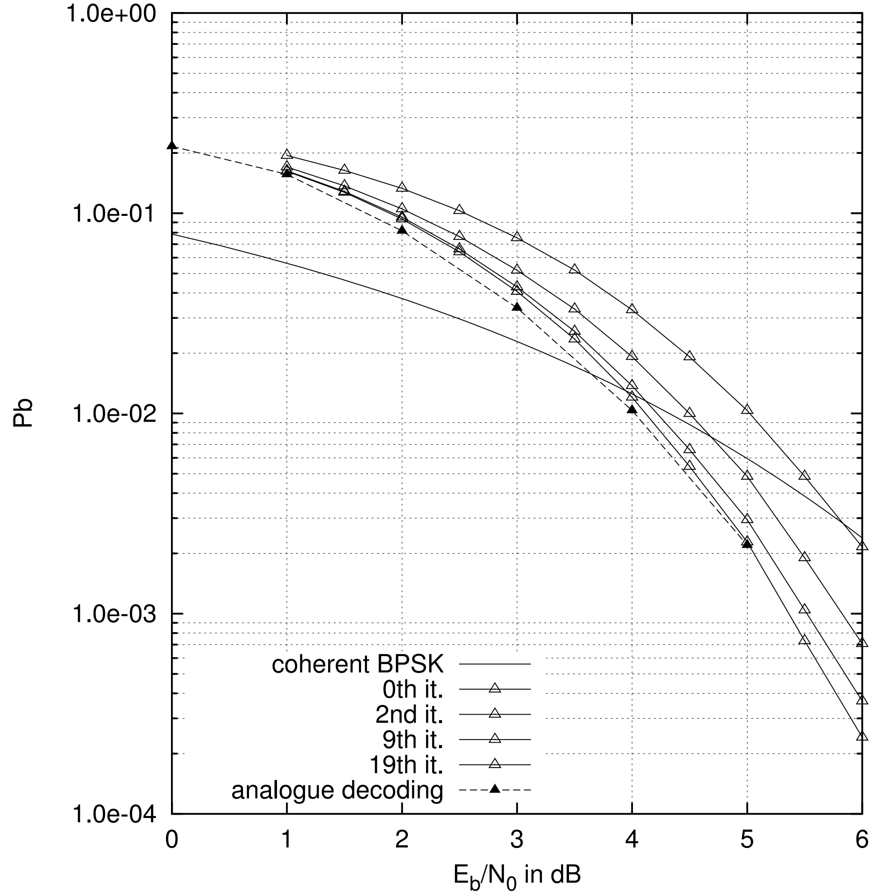
Serial concatenation of tailbiting convolutional codes and DPSK

- APP-DPSK decoder segments S_i
- which are connected via the interleaver ring to the
- TBCC decoder ring circuits.



Serial concatenation of tailbiting convolutional codes and DPSK

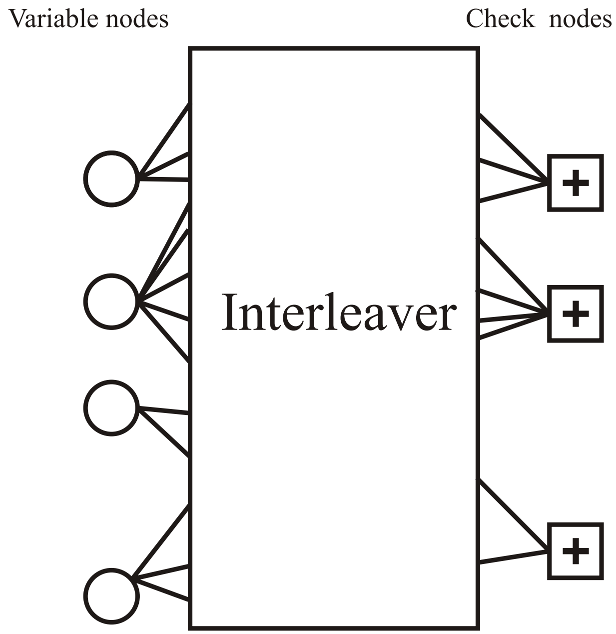
Performance of the analog circuits: BER before decoder



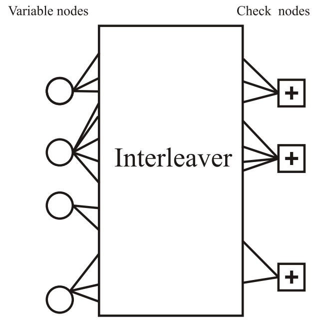
Time-continuous analog circuits better than
19 iterations with digital processors !!

Low Density Parity Check (LDPC) codes and their Turbo decoder

A low density parity check code of rate k/n can be described as a serial concatenation of n variable nodes as inner repetition codes with $n-k$ check nodes as outer single parity check nodes.



Irregular LDPC codes and their Turbo decoder (cnt')



i-th variable node (i-th code bit) with $d_{v,i}$ connections. $n-k$ check nodes where the i-th checks $d_{c,i}$ bits.

More than one extrinsic message

$$L_{i,j}^{(\mathrm{out})} = L_{c,i} \cdot y_i + \sum_{j=1, j\neq i}^{d_{v,i}} L_{i,j}^{(\mathrm{in})} \quad (25)$$

per code bit $x_i, i = 1...n$ is sent to the outer single parity check (SPC) decoders. The SCPC decoders return for $i = 1...n-k$

$$L_{i,j}^{(\mathrm{c,out})} = \sum_{j=1, j\neq i}^{d_{c,i}} \boxplus \; L_{i,j}^{(\mathrm{c,in})} \quad (26)$$

The decoding result is the overall L value of the inner bits.

The Turbo Principle in Communication Systems
Exercises

Joachim Hagenauer

Institute for Communications Engineering (LNT)
Munich University of Technology (TUM)

D-80290 München, Germany

(Short Course 2004)

Problem: Shannons Limit

Given the the capacity formula

$$C = \frac{1}{2}\log_2(1 + \frac{2E_s}{N_0})$$

Calculate the minimal E_b/N_0 in dB for rates 1/2 and $\leftarrow zero$
How do those limits change when you have a complex channel with in-phase ans quadrature-phase?

Problem: Boxplus Calculations

Calculate the following expressions using the box plus approximation:

$$-0.2 \boxplus +5.5 =$$

$$-0.8 \boxplus -5.5 =$$

$$-1.5 \boxplus -1.5 =$$

In which case is the approximation not good?

Problem: APP Decoding of a parity check code

A (3,4,2) parity check code (u_1, u_2, u_3, p) is transmitted over an AWGN channel at an SNR of 3 dB.

The received matched filter values y_i are

$$+2.1, -1.3, -2.6, -4.5$$

.

It is known a priori that the first bit is with probability 0.75 a $+1$.
Give the extrinsic and the APP LLR of the three information bits with APP decoding.
What is the probability the the bit u_1 is in error?

Problem: My first Turbo decoder

A parallel concatenated turbo code as uses a total of 4 rate 2/3 SPC codes as component codes.
A block interleaver is used.
The following $L_c y$ values are received:

-0.5	+1.5	+2.0
+1.0	-0.1	+5.0
+1.5	+3.0	

Perform one vertical and one horizontal iteration.

What is at this stage of iterations the soft-output LLR for the bit at position (1,1) ?

Problem: Irregular LDPC code and decoder

A irregular LDPC code has 4 symbol nodes each of degree 2 and 3 check equations with degrees varying between 2 and 4.

Draw a possible tanner graph.

Using the 4 received values $+1.0, +2.0, -1.0, +3.0$ at an AWGN channel with 0 dB and perform the first steps in message passing with you tanner graph.

Decoding with the simplified BCJR algorithm

- Use a systematic feedforward memory one convolutional encoder
- Use the metric with $L_c = 2$:

$$\log(\mathbf{y}_k|u_k) = \frac{1}{2}L_c \sum_{n=1}^{2} y_{k,n}x_{k,n} = \sum_{n=1}^{2} y_{k,n}x_{k,n}$$

- Assume trellis starts at zero $(+1)$ state followed by 3 sections and is terminated in the zero state $(+1)$.
- The following 8 y values are received during the 4 sections:
  +1.0 +2.0, +0.5 +1.5, -1.0 1.5, -0.5 +2.0
- Perform the α and β recursion and determine the soft-output

Problem: Repeat accumulate code

Describe the RA code as a parallel decodable system and explain its decoder functions.

Problem: BSC as test channel in EXIT Charts

Prove that the ergodic mutual information chart formula gives the capacity of the BSC.

Balakirsky Trellis for Turbo Source Channel Decoding

- Given a source with alphabet A,B,C,D,E and the associated probabilities

letter	A	B	C	D	E
probability	1/2	1/4	1/8	1/16	1/16

- Calculate entropy and average word length of the Huffman code.
- Draw the Balakirsky trellis
- Is this trellis a good component for the iterative turbo scheme?