# State-of-the-Art Channel Coding

Prof. Dr.-Ing. Volker Kühn
Institute of Communications Engineering
Phone: 0381/498-7330, Room: W 8233
Email: volker.kuehn@uni-rostock.de

http://www.int.uni-rostock.de/

# Outline of Lectures

- Lesson 1: Introduction to Error Correcting Codes
  - Principle structure of communication systems
  - Coding principles
  - Basics of error correcting codes
  - Examples: simple block codes and - if time permits - convolutional codes

- Lesson 2: One Lesson of Information Theory
  - Definitions of entropy, mutual information, …
  - Channel coding theorem of Shannon

- Lesson 3: State-of-the-art channel coding
  - Coding strategies to approach the capacity limits
  - Definition of soft-information and turbo decoding principle
  - EXIT Chart Analysis

# Shorty Review of Milestones

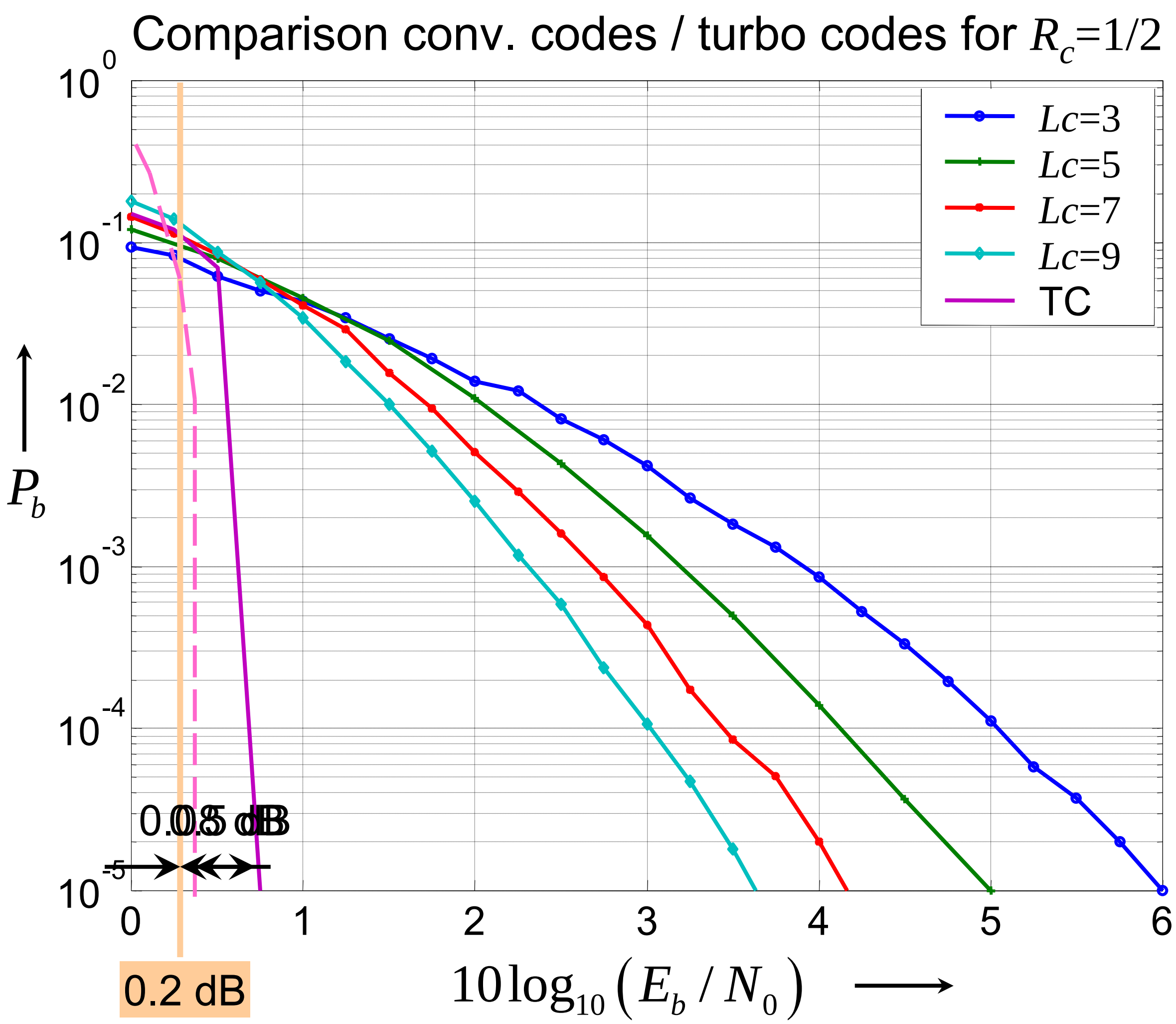
- 1948: Shannon defines his information theory
  - Definition of entropy and mutual information
  - Channel coding theorem

- 1963: Robert G. Gallager: “Low Density Parity Check Codes”
- 1966: G. David Forney: “Concatenated Codes”
  - Computers to that time not strong enough to demonstrate potential of investigated coding schemes
  - Turbo decoding was implicitly already invented

- 1993: First presentation of Turbo-Codes by Berrou, Glavieux, et al.
  - Approaching Shannon’s capacity for half-rate code by 0.5 dB

- 2001: Stephan ten Brink: EXIT Chart Analysis
  - Leads to further understanding of iterative decoding principles
  - Allows design / optimization of powerful concatenated codes
  - Repeat Accumulate Code approaches capacity up to 0.08 dB

# Potential of Turbo Codes

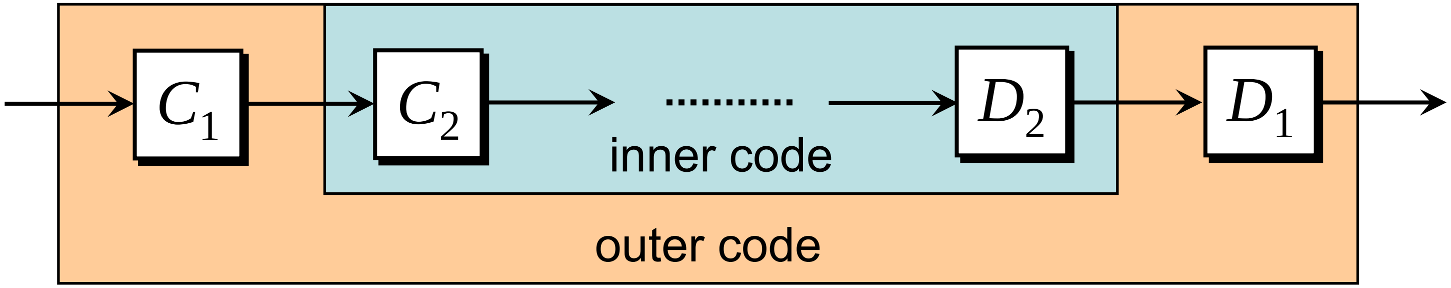


- Optimized interleaver of length 256 x 256 = 65384 bit
- For this interleaver gain of nearly 3 dB over conv. code with $L_c = 9$
- Gap to Shannon's channel capacity only 0.5 dB
- Tremendous performance loss for smaller interleavers

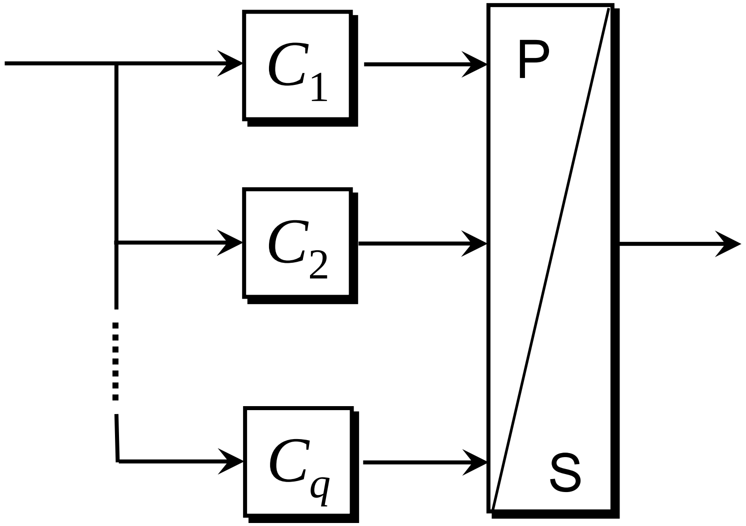
- World record: 0.1 dB gap to Shannon capacity by Stephan ten Brink

# Serial and Parallel Code Concatenation

- Serial Code Concatenation
  - Example: Repeat Accumulate Codes



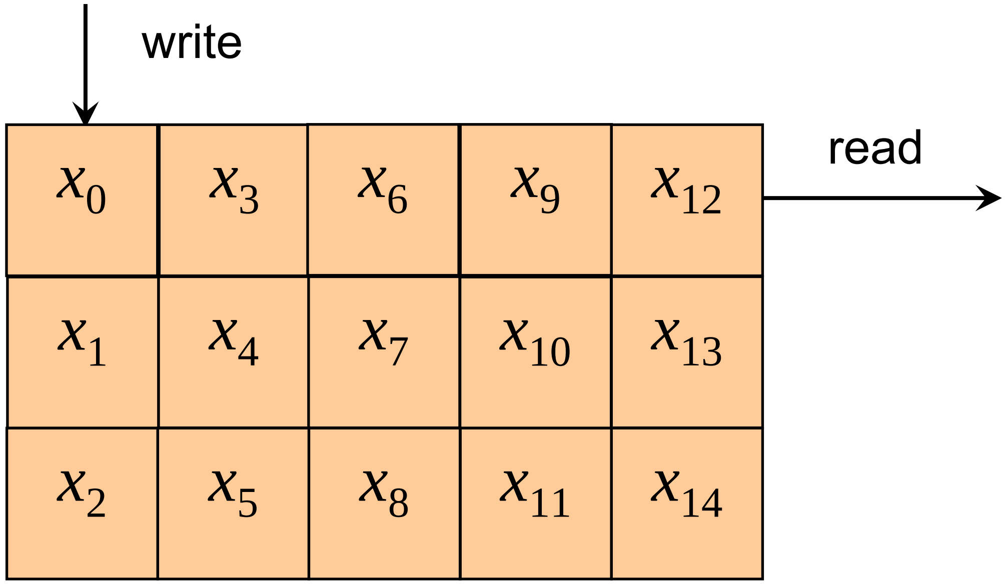
- Parallel Code Concatenation
  - Example: Turbo Codes



# Interleaving

- Simple block interleaver

interleaving depth: 5

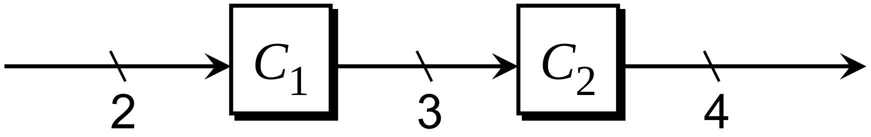


| | | | | |
|---|---|---|---|---|
| $x_0$ | $x_3$ | $x_6$ | $x_9$ | $x_{12}$ |
| $x_1$ | $x_4$ | $x_7$ | $x_{10}$ | $x_{13}$ |
| $x_2$ | $x_5$ | $x_8$ | $x_{11}$ | $x_{14}$ |

  - Input sequence: $x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}$
  - Output sequence: $x_0, x_3, x_6, x_9, x_{12}, x_1, x_4, x_7, x_{10}, x_{13}, x_2, x_5, x_8, x_{11}, x_{14}$
- Convolutional interleaver
- Random interleaver

# Simple Example of Serial Concatenation

- Concatenation of (3,2,2)-SPC and (4,3,2)-SPC code



- Total code rate: $R_c = 2/4 = 0.5$

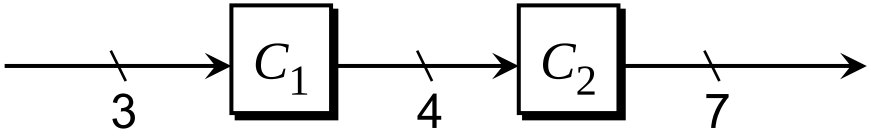
| $\mathbf{u}$ | $\mathbf{c}_1$ | $\mathbf{c}_2$ | $w_H(\mathbf{c}_2)$ |
|---|---|---|---|
| 00 | 000 | 0000 | 0 |
| 01 | 011 | 0110 | 2 |
| 10 | 101 | 1010 | 2 |
| 11 | 110 | 1100 | 2 |

$d_{\min} = 2$

- Minimum Hamming distance is not improved by code concatenation

# Another Example of Serial Concatenation

- Concatenation of (4,3,2)-SPC and (7,4,3)-Hamming code



- Total code rate: $R_c = 3/7$

| $\mathbf{c}_1$ | $\mathbf{c}_2$ | $w_H(\mathbf{c}_2)$ | $\mathbf{c}_2$ | $w_H(\mathbf{c}_2)$ |
|---|---|---|---|---|
| 0000 | 0000 000 | 0 | 0000 000 | 0 |
| 0011 | 0011 001 | 3 | 0001 111 | 4 |
| 0101 | 0101 010 | 3 | 0110 011 | 4 |
| 0110 | 0110 011 | 4 | 0111 100 | 4 |
| 1001 | 1001 100 | 3 | 1010 101 | 4 |
| 1010 | 1010 101 | 4 | 1011 010 | 4 |
| 1100 | 1100 110 | 4 | 1100 110 | 4 |
| 1110 | 1111 111 | 7 | 1101 001 | 4 |

original concatenation:

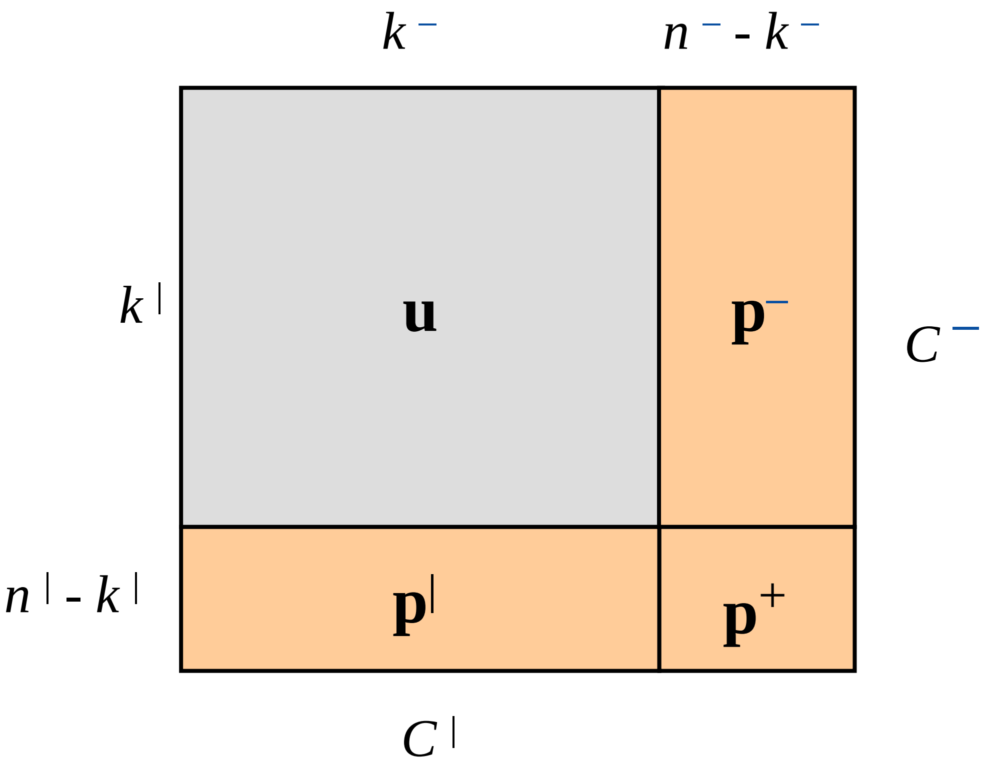
$d_{\min} = 3$

optimized concatenation:

$d_{\min} = 4$

- Minimum Hamming distance can only be improved by careful selection of subset of inner code

# Serial Code Concatenation: Product Codes



- Information bits arranged in $(k^{|},k^{-})$-matrix
- Row-wise encoding with code $C^{-}$ of rate $k^{-} / n^{-}$
- Column-wise encoding with code $C^{|}$ of rate $k^{|} / n^{|}$
- Entire code rate:

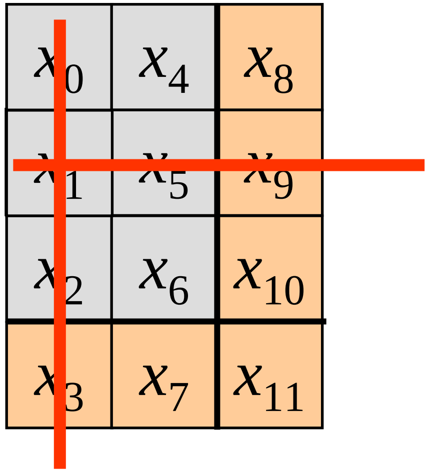
$$R_{\mathrm{c}} = \frac{k^{-} \cdot k^{|}}{n^{-} \cdot n^{|}} = R_{\mathrm{c}}^{-} \cdot R_{\mathrm{c}}^{|}$$

- Minimum Hamming distance:

$$d_{\min} = d_{\min}^{-} \cdot d_{\min}^{|}$$

# Examples of Product Codes (1)

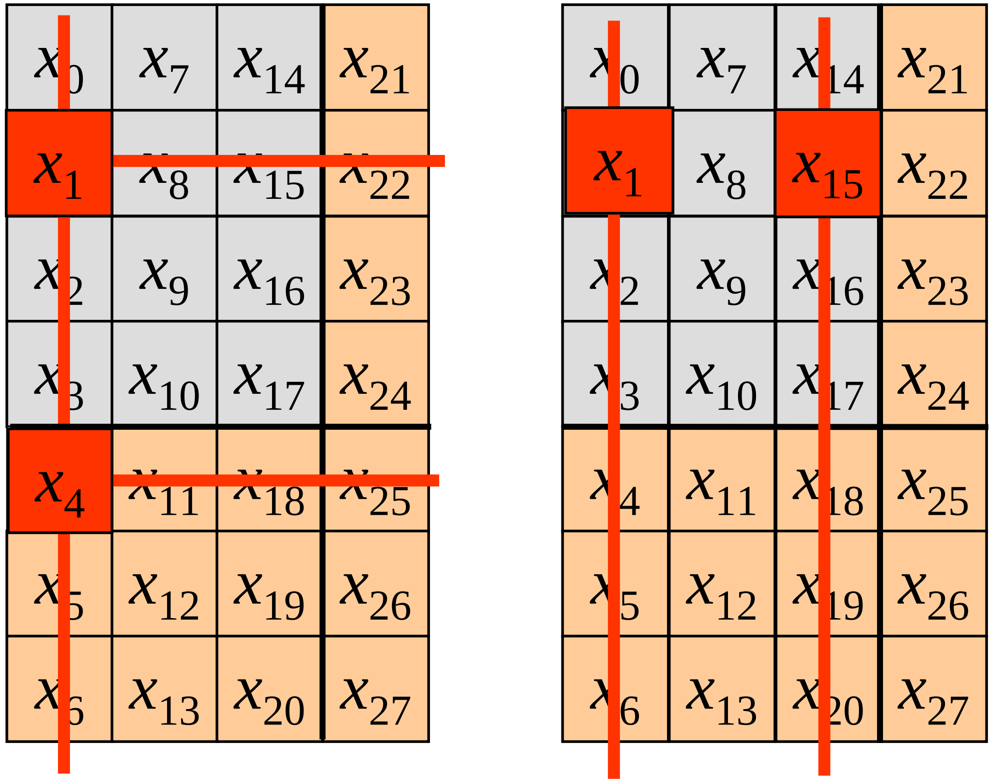
- (12,6,4) product code



  - Horizontal: (3,2,2)-SPC code → no error correction possible
  - Vertical: (4,3,2)-SPC code → no error correction possible
  - Code rate: 1/2
  - $d_{\min} = 2 \cdot 2 = 4$
  - Correction of 1 error possible

# Examples of Product Codes (2)

- (28,12,6) product code

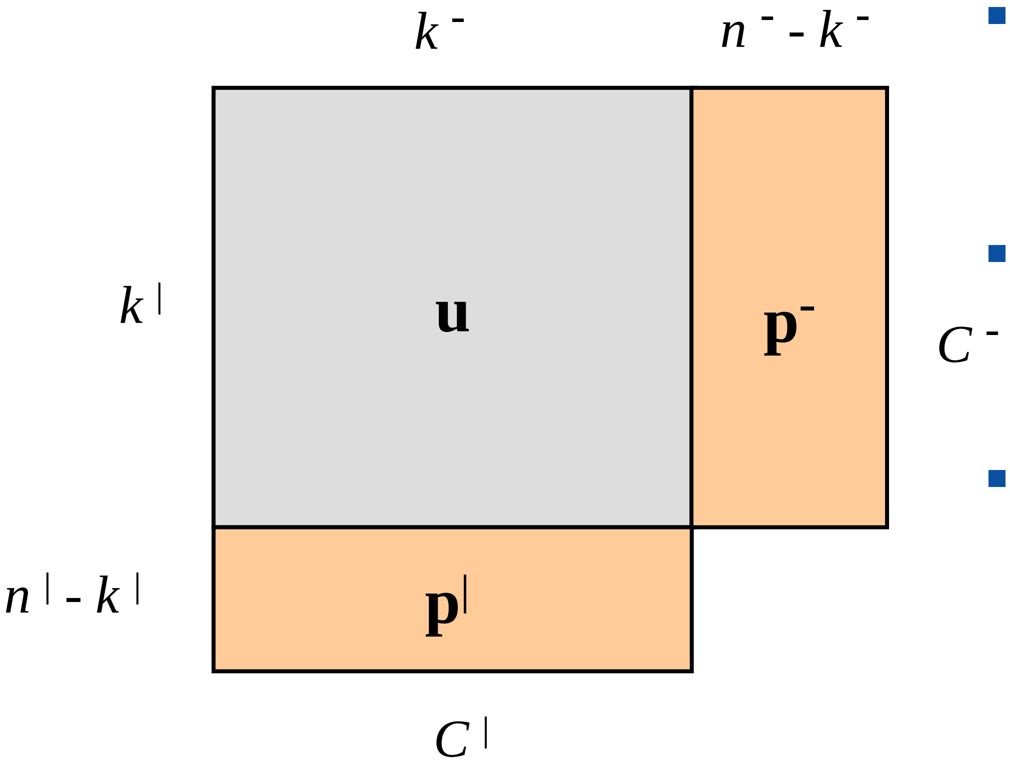


| | | | |
|---|---|---|---|
| $x_0$ | $x_7$ | $x_{14}$ | $x_{21}$ |
| $x_1$ | $x_8$ | $x_{15}$ | $x_{22}$ |
| $x_2$ | $x_9$ | $x_{16}$ | $x_{23}$ |
| $x_3$ | $x_{10}$ | $x_{17}$ | $x_{24}$ |
| $x_4$ | $x_{11}$ | $x_{18}$ | $x_{25}$ |
| $x_5$ | $x_{12}$ | $x_{19}$ | $x_{26}$ |
| $x_6$ | $x_{13}$ | $x_{20}$ | $x_{27}$ |

| | | | |
|---|---|---|---|
| $x_0$ | $x_7$ | $x_{14}$ | $x_{21}$ |
| $x_1$ | $x_8$ | $x_{15}$ | $x_{22}$ |
| $x_2$ | $x_9$ | $x_{16}$ | $x_{23}$ |
| $x_3$ | $x_{10}$ | $x_{17}$ | $x_{24}$ |
| $x_4$ | $x_{11}$ | $x_{18}$ | $x_{25}$ |
| $x_5$ | $x_{12}$ | $x_{19}$ | $x_{26}$ |
| $x_6$ | $x_{13}$ | $x_{20}$ | $x_{27}$ |

  - Horizontal: (4,3,2)-SPC code → no error correction possible
  - Vertical: (7,4,3)-Hamming code → single error correction possible
  - $d_{\min} = 2 \cdot 3 = 6$ → 2 errors correctable

# Parallel Code Concatenation



- Information bits $\mathbf{u}$
  - row-wise encoded with $C^-$ and
  - column-wise and $C^|$
- Parity check bits of component codes not encoded (no checks on checks)
- Entire code rate

$$R_c = \frac{k^- \cdot k^|}{n^- \cdot n^| - (n^- - k^-) \cdot (n^| - k^|)}$$

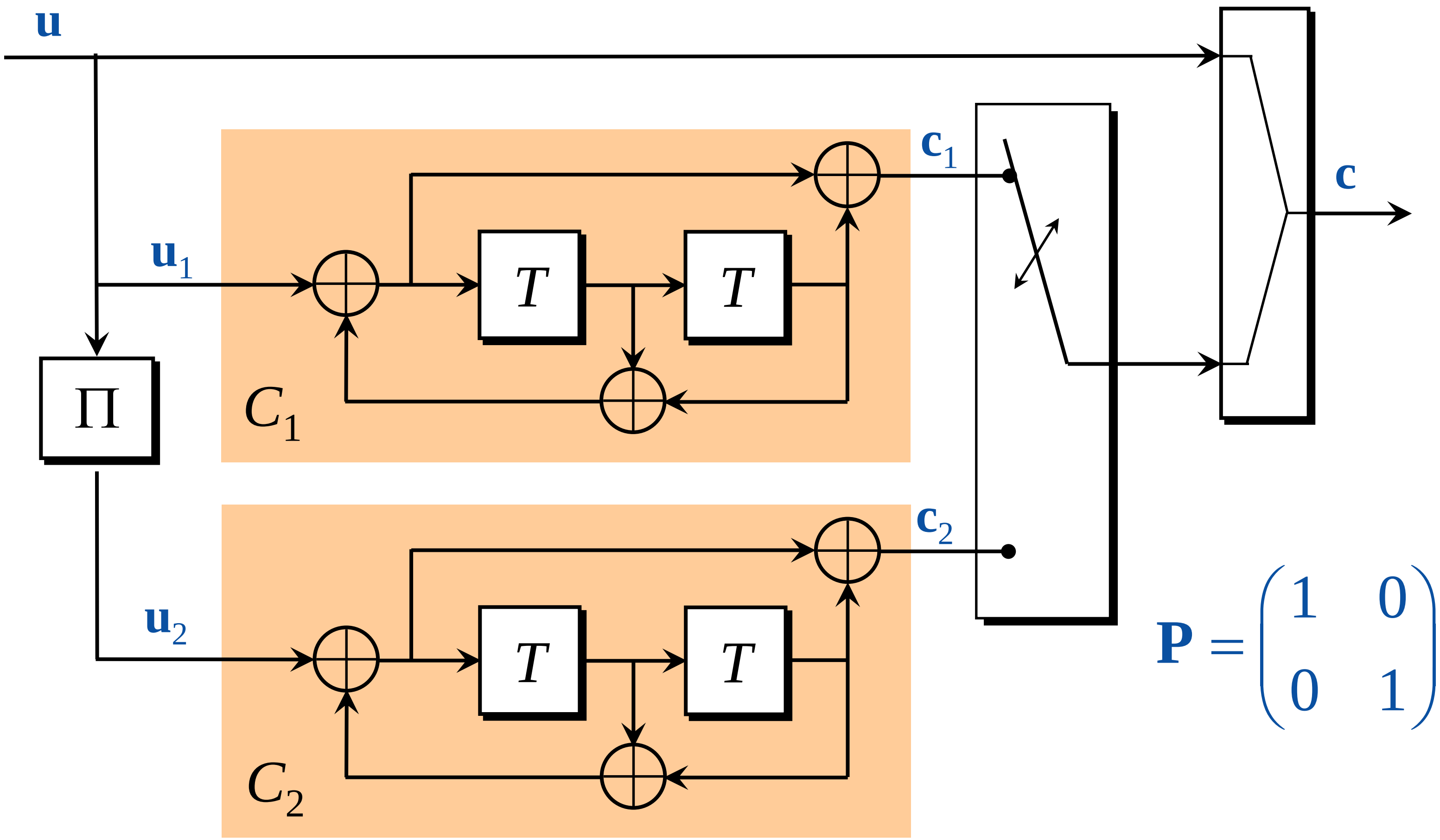
$$= \frac{1}{1/R_c^- + 1/R_c^| - 1}$$

- Minimum Hamming distance:

$$d_{\min} = d_{\min}^- + d_{\min}^| - 1$$

# Example of Turbo Code

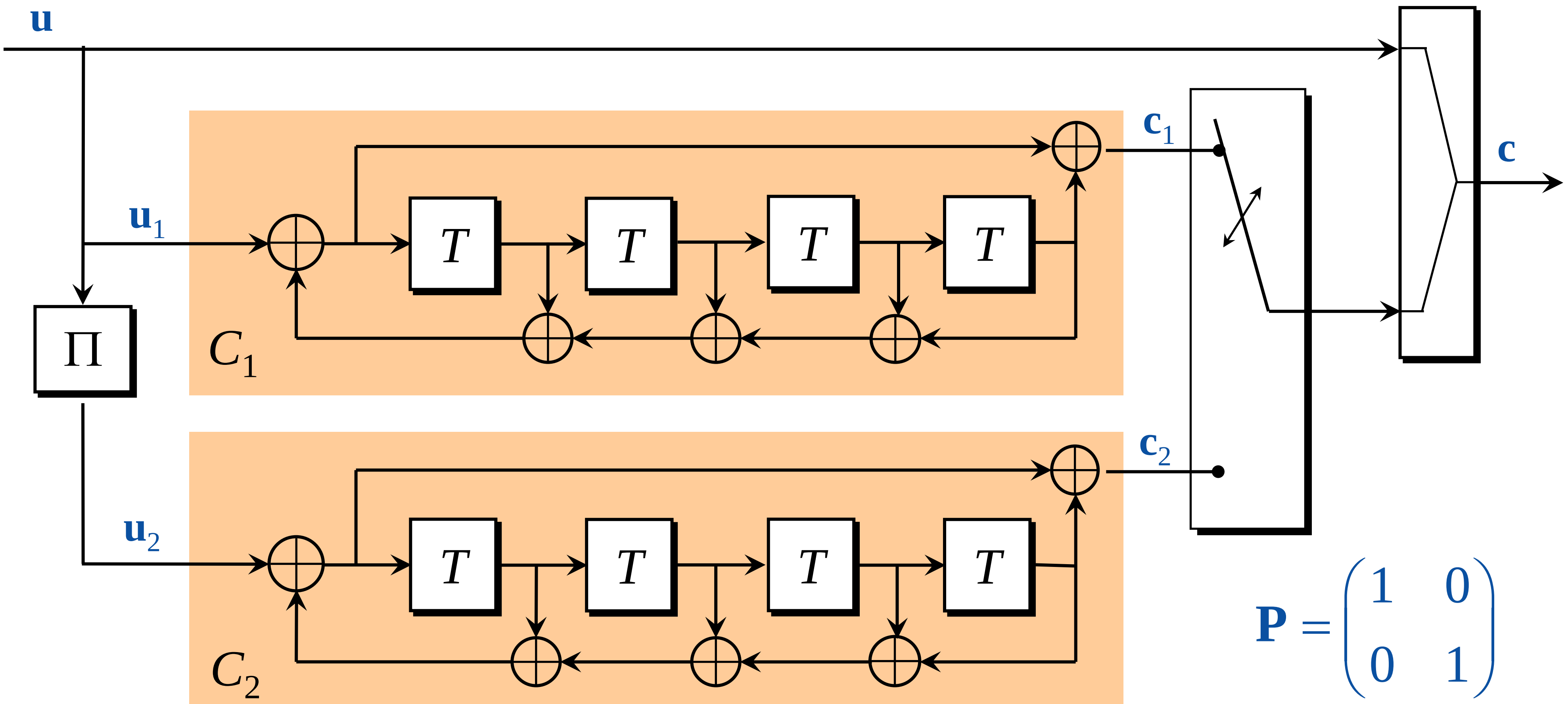
- 2 systematic, recursive convolutional encoders ($L_c = 3$)
- Constituent code rates $R_\mathrm{c} = 2/3$ → total code rate $R_\mathrm{c} = 1/2$



$$\mathbf{P} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

# Turbo Code from Berrou and Glavieux

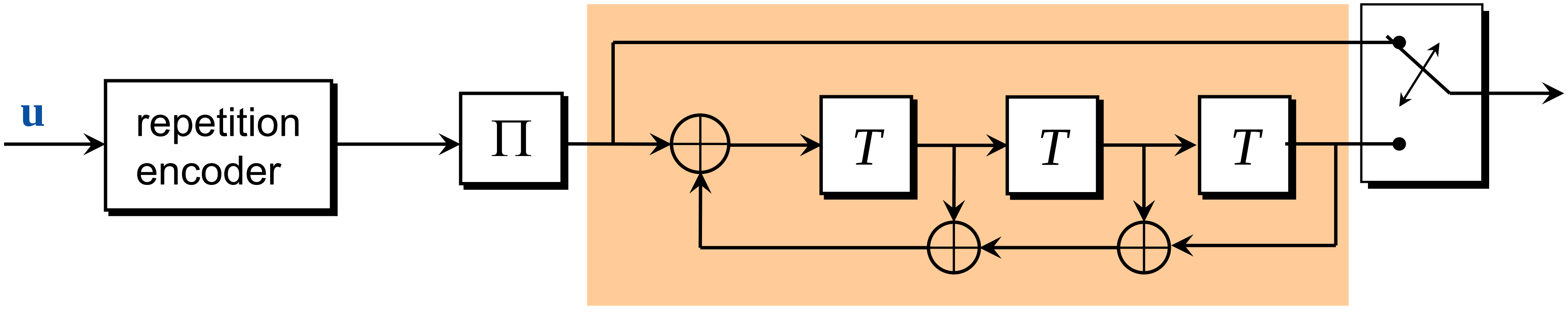
- 2 systematic, recursive convolutional encoders ($L_c = 5$)
- Constituent code rates $R_c = 2/3$ → total code rate $R_c = 1/2$
- Pseudo random interleaver of length 65536 bits



$$\mathbf{P} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

# Repeat Accumulate Code from ten Brink

- Outer half-rate repetition code
- Inner convolutional code (scrambler) of rate 1
- → total code rate $R_c = 1/2$
- Random interleaver of different lengths
- Code doping: replace a few code bits (1%) by information bits for improving the convergence of iterative decoding process



inner convolutional code

# Symbol-by-Symbol Soft-Output Decoding
# Turbo-Decoding

# Log-Likelihood Ratios (LLRs)

- Definition log-likelihood ratio:
  - Sign determines hard decision
  - Magnitude represents reliability of hard decision

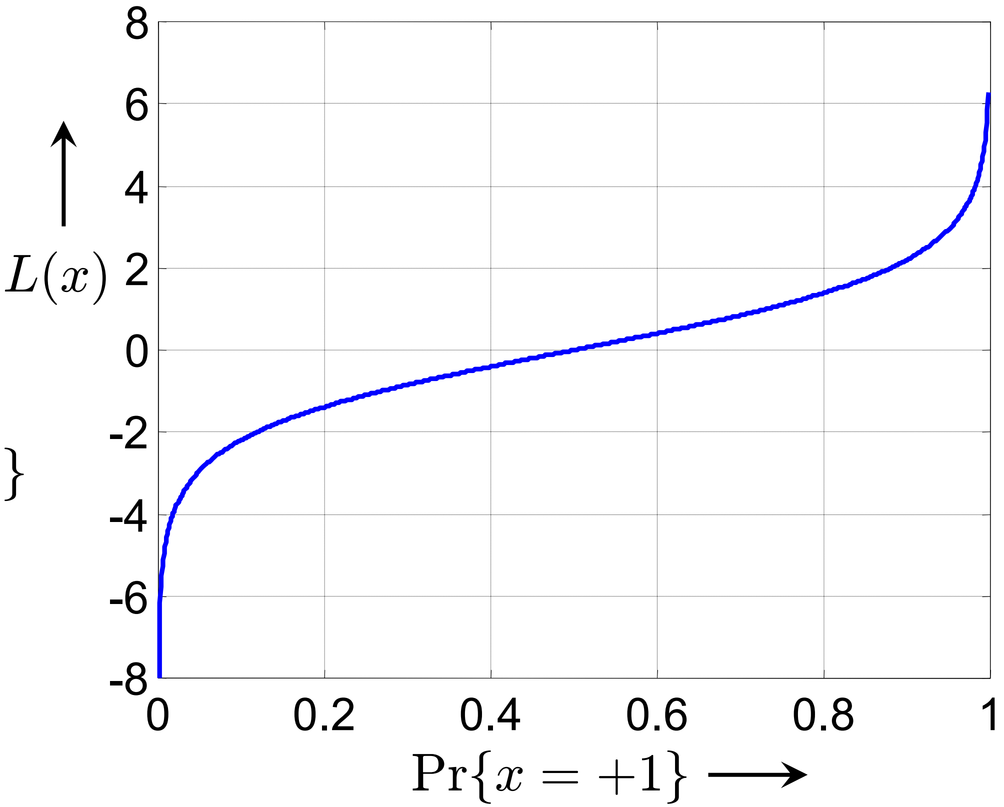
$$L(x) = \log \frac{\Pr\{x=+1\}}{\Pr\{x=-1\}}$$

- Probability of correct decision:

$$P_{\text{correct}} = \frac{e^{|L(x)|}}{1+e^{|L(x)|}}$$

- Expectation of LLR (soft  bit)

$$\begin{aligned} E\{x\} &= \Pr\{x=+1\} - \Pr\{x=-1\} \\ &= \frac{e^{L(x)}}{1+e^{L(x)}} - \frac{1}{1+e^{L(x)}} \\ &= \tanh\left(\frac{L(x)}{2}\right) \end{aligned}$$

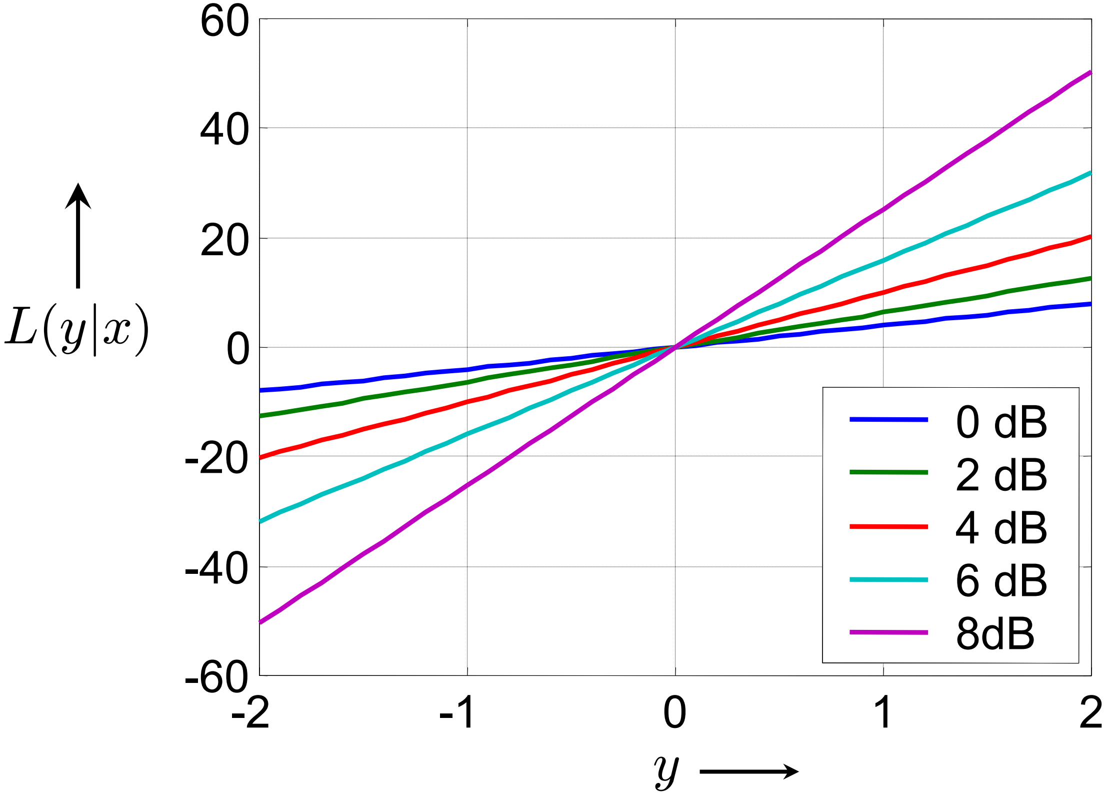


# LLRs for AWGN Output

$$L(y \mid x) = \log \frac{p(y \mid x=+1)}{p(y \mid x=-1)} = \log \frac{\exp[-(y-1)^2/2/\sigma_N^2]}{\exp[-(y+1)^2/2/\sigma_N^2]}$$

$$= -\frac{(y-1)^2}{2\sigma_N^2} + \frac{(y+1)^2}{2\sigma_N^2} = \frac{2}{\sigma_N^2} \cdot y$$

Scaled matched filter output equals LLR



# Example for Soft-Output Decoding

- Single parity check code $\boxed{u_1}\boxed{u_2}\boxed{p}$
- Parity check equation: $p = u_1 \oplus u_2$
- Question: What is the LLR of $u_1$ given the LLRs of $u_2$ and $p$?
  - Resolving parity check equation w.r.t. $u_1$: $u_1 = u_2 \oplus p$
  - Extrinsic LLR does not depend on $u_1$ itself:

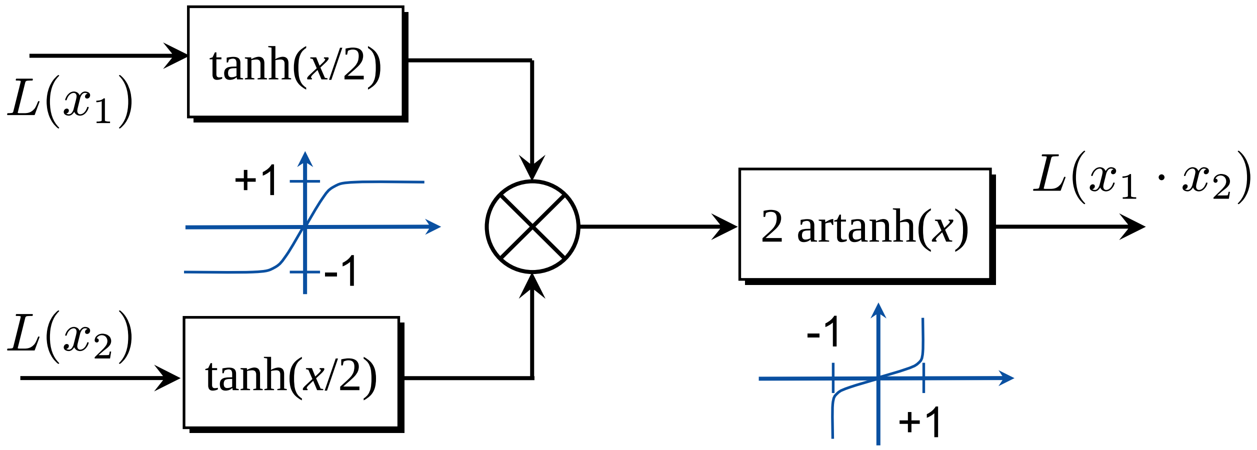
$$
\begin{aligned}
L_e(u_1) &= \log_2 \frac{\Pr\{u_2 \oplus p = 0\}}{\Pr\{u_2 \oplus p = 1\}} = \log_2 \frac{\Pr\{u_2 = 0, p = 0\} + \Pr\{u_2 = 1, p = 1\}}{\Pr\{u_2 = 0, p = 1\} + \Pr\{u_2 = 1, p = 0\}} \\
&= \log_2 \frac{\Pr\{u_2 = 0\}\Pr\{p = 0\} + \Pr\{u_2 = 1\}\Pr\{p = 1\}}{\Pr\{u_2 = 0\}\Pr\{p = 1\} + \Pr\{u_2 = 1\}\Pr\{p = 0\}} \\
&\ \ \vdots \\
&= 2 \cdot \operatorname{atanh}\left[\tanh\left(\frac{L(u_2)}{2}\right) \cdot \tanh\left(\frac{L(p)}{2}\right)\right] = 2 \cdot \operatorname{atanh}\left[E\{u_2\} \cdot E\{p\}\right]
\end{aligned}
$$

# $L$-Algebra

- mod-2-sum of 2 statistical independent random variables:

$$\underset{u_1 \oplus u_2}{L(x_1 \cdot x_2)} = 2 \cdot \operatorname{atanh}\left[\tanh\left(L(x_1)/2 \cdot \tanh\left(L(x_2)/2\right)\right]$$

$$\approx \operatorname{sgn}\left[L(x_1)\right] \cdot \operatorname{sgn}\left[L(x_2)\right] \cdot \min\left\{|L(x_1)|, |L(x_2)|\right\}$$



- mod-2-sum of $n$ variables: $L(x_1 \cdots x_n) = 2\operatorname{artanh}\left[\prod_{i=1}^{n} \tanh\left(\frac{L(x_i)}{2}\right)\right]$

$$\approx \prod_{i=1}^{n} \operatorname{sgn}\left[L(x_i)\right] \cdot \min_i \left\{|L(x_i)|\right\}$$

# General Approach for Soft-Output Decoding

- For systematic encoders, soft-output of decoder can be split into 3 statistically independent parts:

$$L(\hat{u}_i) = \ln \frac{p(u_i = 0, \mathbf{y})}{p(u_i = 1, \mathbf{y})} = \ln \frac{\sum_{\mathbf{c} \in \Gamma_i^{(0)}} p(\mathbf{y} \mid \mathbf{x}) \cdot \Pr\{\mathbf{c}\}}{\sum_{\mathbf{c} \in \Gamma_i^{(1)}} p(\mathbf{y} \mid \mathbf{x}) \cdot \Pr\{\mathbf{c}\}}$$

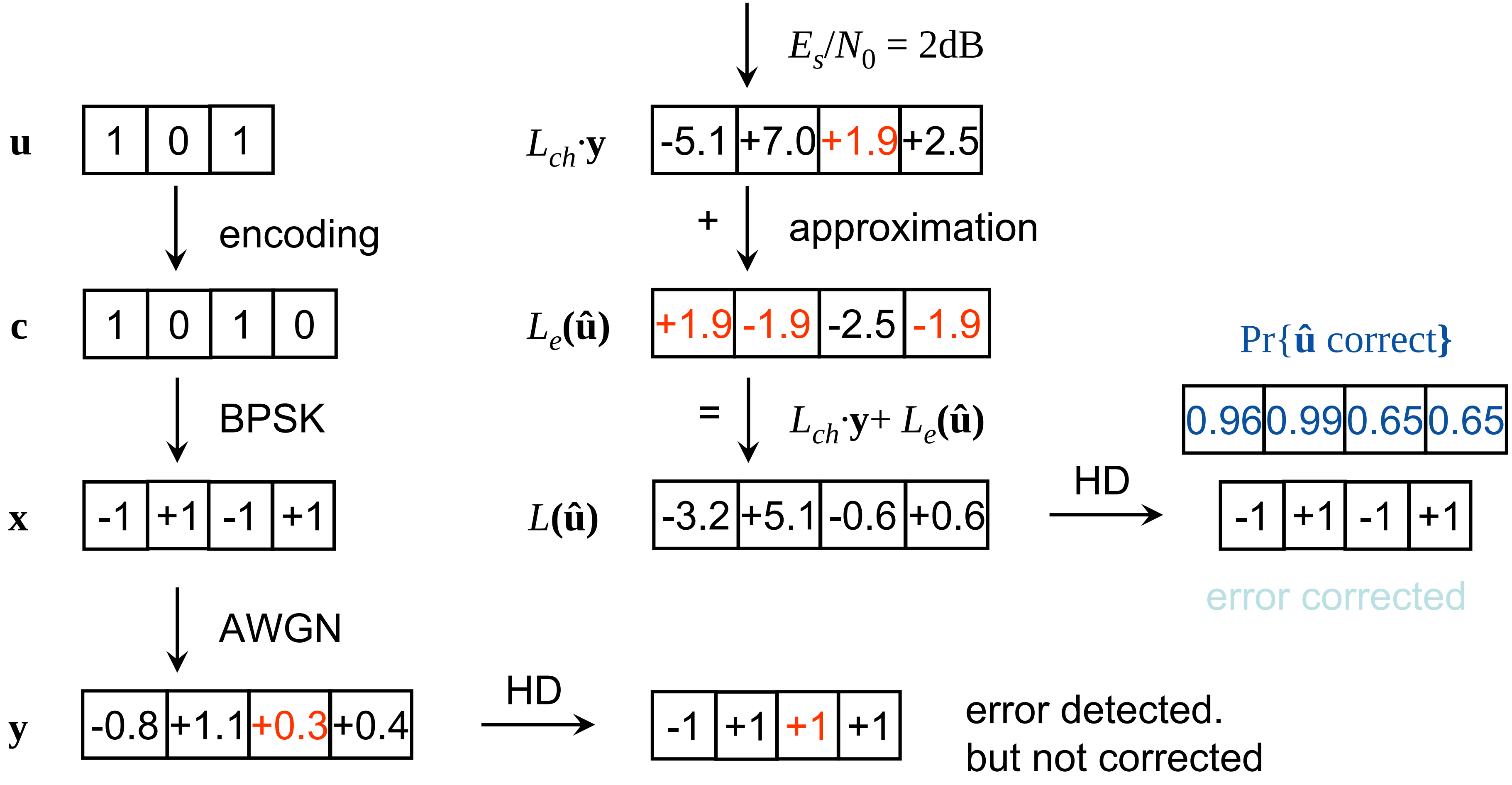
$$= \underbrace{\ln \frac{p(y_i \mid x_i = +1)}{p(y_i \mid x_i = -1)}}_{L_{\text{ch}} y_i} + \underbrace{\ln \frac{\Pr\{u_i = 0\}}{\Pr\{u_i = 1\}}}_{L_a(u_i)} + \underbrace{\ln \frac{\sum\limits_{\mathbf{c} \in \Gamma_i^{(0)}} \prod\limits_{\substack{j=1 \\ j \neq i}}^{n} p(y_j \mid x_j) \cdot \prod\limits_{\substack{j=1 \\ j \neq i}}^{k} \Pr\{c_j\}}{\sum\limits_{\mathbf{c} \in \Gamma_i^{(1)}} \prod\limits_{\substack{j=1 \\ j \neq i}}^{n} p(y_j \mid x_j) \cdot \prod\limits_{\substack{j=1 \\ j \neq i}}^{k} \Pr\{c_j\}}}_{L_e(\hat{u}_i)}$$

Intrinsic LLR
(systematic part)

A-priori
LLR

Extrinsic
LLR

# Soft-Output Decoding for (4,3,2)-SPC-Code



|  |  |  |  |  |
|---|---|---|---|---|
| $\mathbf{u}$ | 1 | 0 | 1 |  |

encoding

|  |  |  |  |  |
|---|---|---|---|---|
| $\mathbf{c}$ | 1 | 0 | 1 | 0 |

BPSK

|  |  |  |  |  |
|---|---|---|---|---|
| $\mathbf{x}$ | -1 | +1 | -1 | +1 |

AWGN

|  |  |  |  |  |
|---|---|---|---|---|
| $\mathbf{y}$ | -0.8 | +1.1 | +0.3 | +0.4 |

HD →

| | | | |
|---|---|---|---|
| -1 | +1 | +1 | +1 |

error detected.
but not corrected

$E_s/N_0 = 2\text{dB}$

|  |  |  |  |  |
|---|---|---|---|---|
| $L_{ch}\cdot\mathbf{y}$ | -5.1 | +7.0 | +1.9 | +2.5 |

\+ approximation

|  |  |  |  |  |
|---|---|---|---|---|
| $L_e(\hat{\mathbf{u}})$ | +1.9 | -1.9 | -2.5 | -1.9 |

= $L_{ch}\cdot\mathbf{y}+L_e(\hat{\mathbf{u}})$

|  |  |  |  |  |
|---|---|---|---|---|
| $L(\hat{\mathbf{u}})$ | -3.2 | +5.1 | -0.6 | +0.6 |

HD →

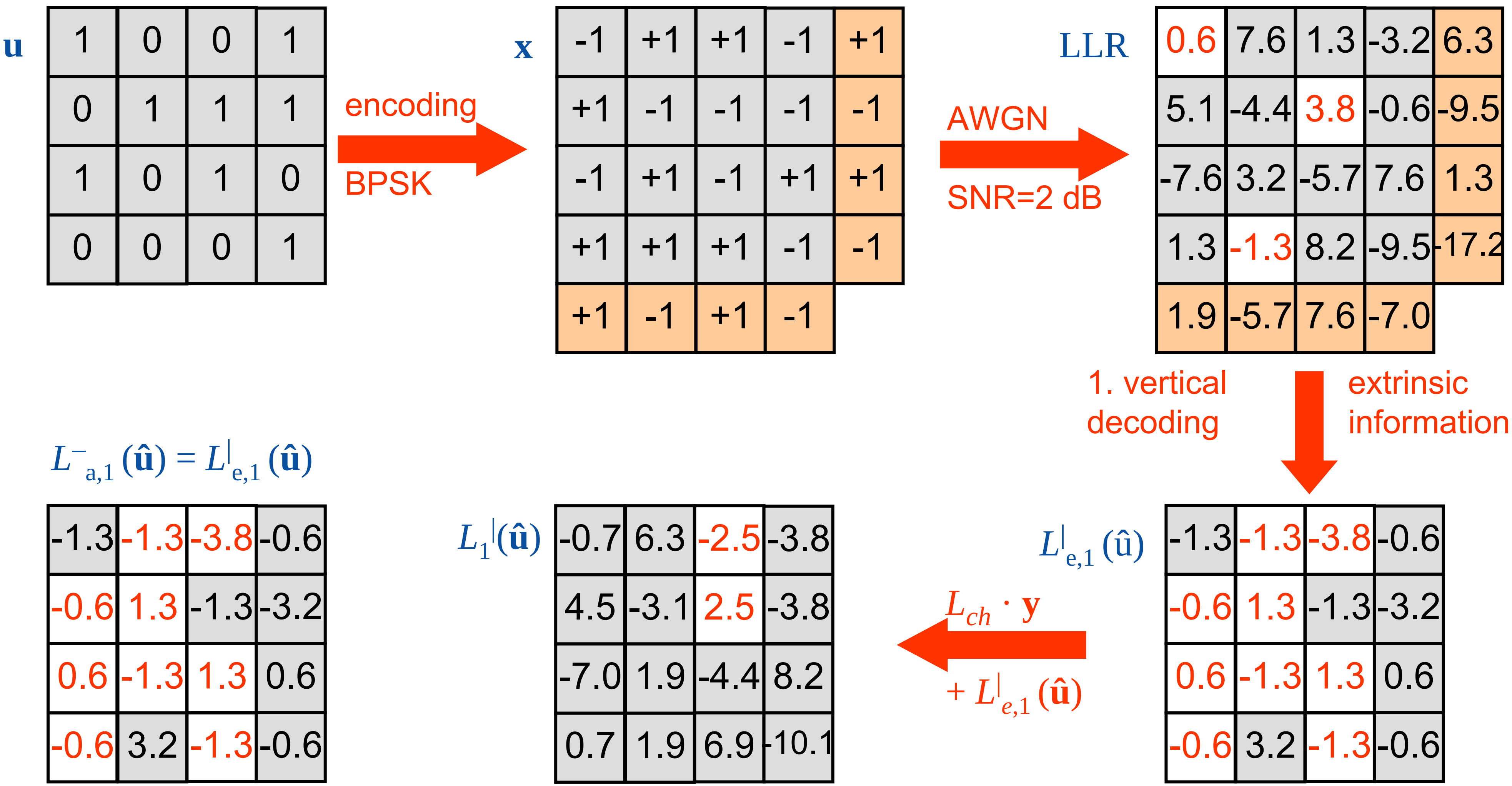
| | | | |
|---|---|---|---|
| -1 | +1 | -1 | +1 |

error corrected

$\Pr\{\hat{\mathbf{u}}\text{ correct}\}$

| | | | |
|---|---|---|---|
| 0.96 | 0.99 | 0.65 | 0.65 |

# Turbo Decoding of (24,16,3)-Produktcode (1)



**u**

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |

encoding

BPSK

**x**

| | | | | |
|---|---|---|---|---|
| -1 | +1 | +1 | -1 | +1 |
| +1 | -1 | -1 | -1 | -1 |
| -1 | +1 | -1 | +1 | +1 |
| +1 | +1 | +1 | -1 | -1 |
| +1 | -1 | +1 | -1 | |

AWGN

SNR=2 dB

LLR

| | | | | |
|---|---|---|---|---|
| 0.6 | 7.6 | 1.3 | -3.2 | 6.3 |
| 5.1 | -4.4 | 3.8 | -0.6 | -9.5 |
| -7.6 | 3.2 | -5.7 | 7.6 | 1.3 |
| 1.3 | -1.3 | 8.2 | -9.5 | -17.2 |
| 1.9 | -5.7 | 7.6 | -7.0 | |

1. vertical decoding

extrinsic information

$L^{|}_{e,1}(\hat{u})$

| | | | |
|---|---|---|---|
| -1.3 | -1.3 | -3.8 | -0.6 |
| -0.6 | 1.3 | -1.3 | -3.2 |
| 0.6 | -1.3 | 1.3 | 0.6 |
| -0.6 | 3.2 | -1.3 | -0.6 |

$L_{ch} \cdot \mathbf{y}$

$+ L^{|}_{e,1}(\hat{\mathbf{u}})$

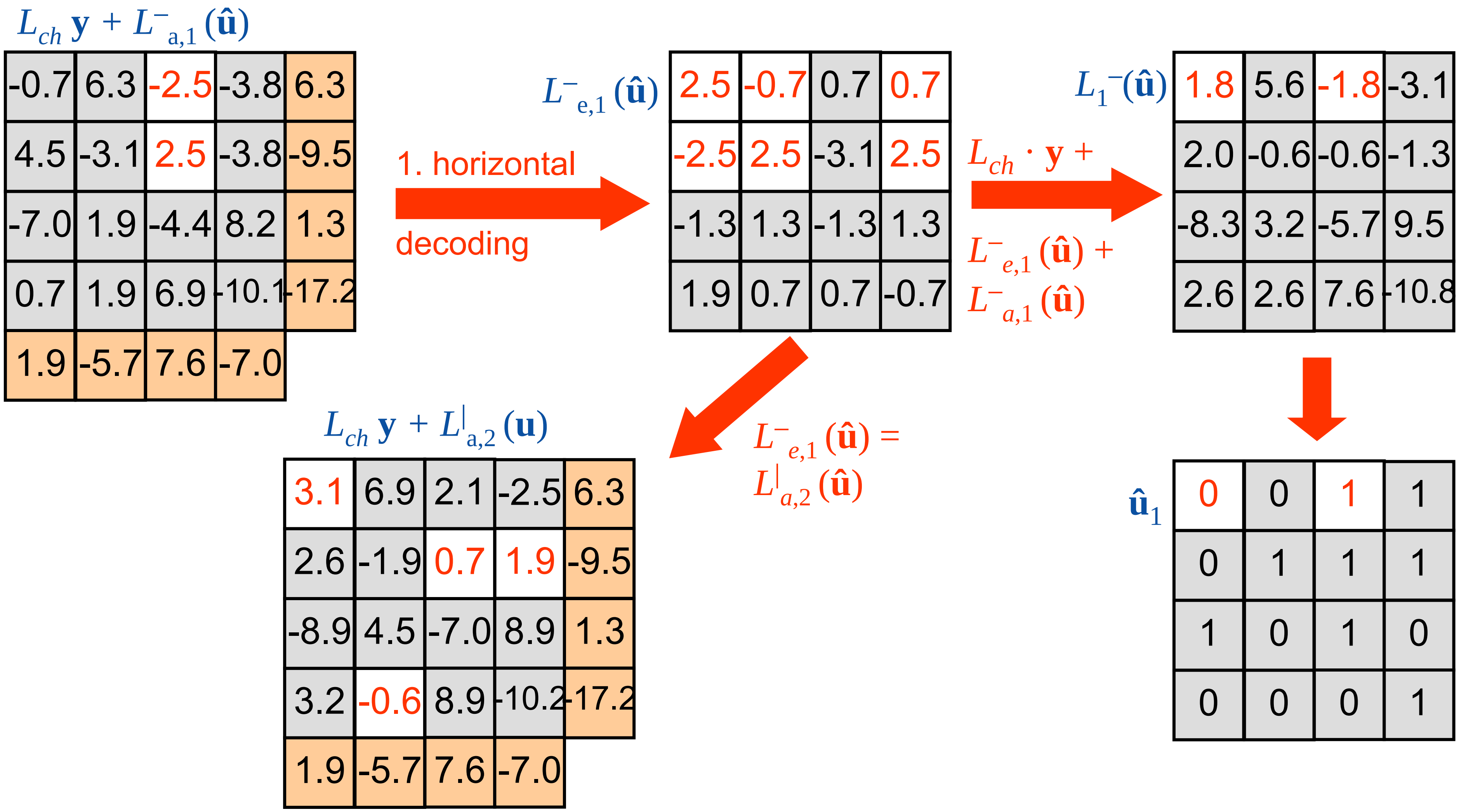
$L_1^{|}(\hat{\mathbf{u}})$

| | | | |
|---|---|---|---|
| -0.7 | 6.3 | -2.5 | -3.8 |
| 4.5 | -3.1 | 2.5 | -3.8 |
| -7.0 | 1.9 | -4.4 | 8.2 |
| 0.7 | 1.9 | 6.9 | -10.1 |

$L^{-}_{a,1}(\hat{\mathbf{u}}) = L^{|}_{e,1}(\hat{\mathbf{u}})$

| | | | |
|---|---|---|---|
| -1.3 | -1.3 | -3.8 | -0.6 |
| -0.6 | 1.3 | -1.3 | -3.2 |
| 0.6 | -1.3 | 1.3 | 0.6 |
| -0.6 | 3.2 | -1.3 | -0.6 |

# Turbo Decoding of (24,16,3)-Produktcode (2)



$L_{ch}\,\mathbf{y} + L^{-}_{a,1}(\hat{\mathbf{u}})$

| | | | | |
|---|---|---|---|---|
| -0.7 | 6.3 | -2.5 | -3.8 | 6.3 |
| 4.5 | -3.1 | 2.5 | -3.8 | -9.5 |
| -7.0 | 1.9 | -4.4 | 8.2 | 1.3 |
| 0.7 | 1.9 | 6.9 | -10.1 | -17.2 |
| 1.9 | -5.7 | 7.6 | -7.0 | |

1. horizontal decoding

$L^{-}_{e,1}(\hat{\mathbf{u}})$

| | | | |
|---|---|---|---|
| 2.5 | -0.7 | 0.7 | 0.7 |
| -2.5 | 2.5 | -3.1 | 2.5 |
| -1.3 | 1.3 | -1.3 | 1.3 |
| 1.9 | 0.7 | 0.7 | -0.7 |

$L_{ch}\cdot\mathbf{y} + L^{-}_{e,1}(\hat{\mathbf{u}}) + L^{-}_{a,1}(\hat{\mathbf{u}})$

$L_1^{-}(\hat{\mathbf{u}})$

| | | | |
|---|---|---|---|
| 1.8 | 5.6 | -1.8 | -3.1 |
| 2.0 | -0.6 | -0.6 | -1.3 |
| -8.3 | 3.2 | -5.7 | 9.5 |
| 2.6 | 2.6 | 7.6 | -10.8 |

$\hat{\mathbf{u}}_1$

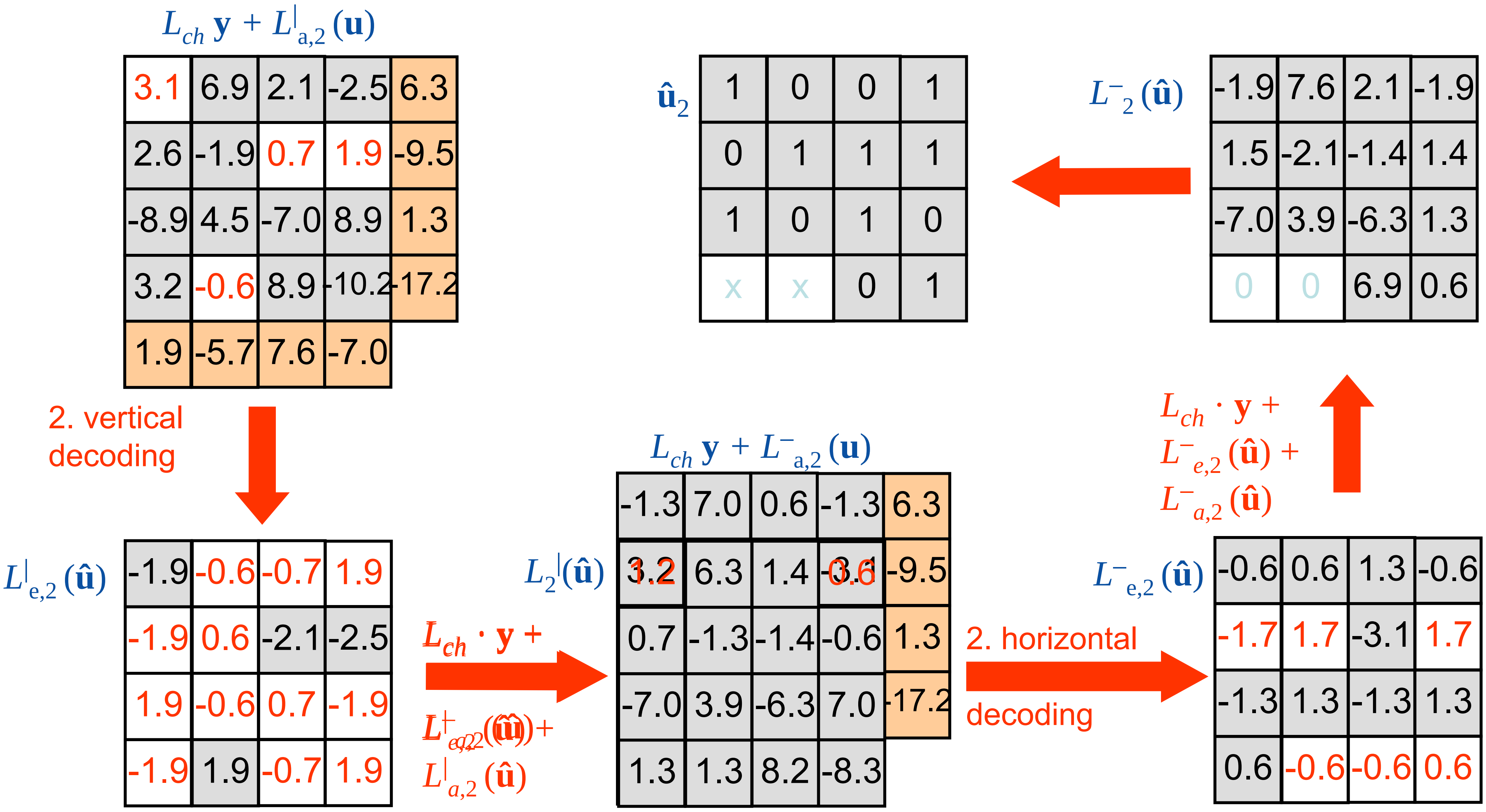
| | | | |
|---|---|---|---|
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |

$L^{-}_{e,1}(\hat{\mathbf{u}}) = L^{|}_{a,2}(\hat{\mathbf{u}})$

$L_{ch}\,\mathbf{y} + L^{|}_{a,2}(\mathbf{u})$

| | | | | |
|---|---|---|---|---|
| 3.1 | 6.9 | 2.1 | -2.5 | 6.3 |
| 2.6 | -1.9 | 0.7 | 1.9 | -9.5 |
| -8.9 | 4.5 | -7.0 | 8.9 | 1.3 |
| 3.2 | -0.6 | 8.9 | -10.2 | -17.2 |
| 1.9 | -5.7 | 7.6 | -7.0 | |

# Turbo Decoding of (24,16,3)-Produktcode (3)



$L_{ch}\,\mathbf{y} + L^{|}_{a,2}\,(\mathbf{u})$

| | | | | |
|---|---|---|---|---|
| 3.1 | 6.9 | 2.1 | -2.5 | 6.3 |
| 2.6 | -1.9 | 0.7 | 1.9 | -9.5 |
| -8.9 | 4.5 | -7.0 | 8.9 | 1.3 |
| 3.2 | -0.6 | 8.9 | -10.2 | -17.2 |
| 1.9 | -5.7 | 7.6 | -7.0 | |

2. vertical decoding

$\hat{\mathbf{u}}_2$

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| x | x | 0 | 1 |

$L^{-}_{2}\,(\hat{\mathbf{u}})$

| | | | |
|---|---|---|---|
| -1.9 | 7.6 | 2.1 | -1.9 |
| 1.5 | -2.1 | -1.4 | 1.4 |
| -7.0 | 3.9 | -6.3 | 1.3 |
| 0 | 0 | 6.9 | 0.6 |

$L_{ch}\cdot\mathbf{y} +$

$L^{-}_{e,2}\,(\hat{\mathbf{u}}) +$

$L^{-}_{a,2}\,(\hat{\mathbf{u}})$

$L^{|}_{e,2}\,(\hat{\mathbf{u}})$

| | | | |
|---|---|---|---|
| -1.9 | -0.6 | -0.7 | 1.9 |
| -1.9 | 0.6 | -2.1 | -2.5 |
| 1.9 | -0.6 | 0.7 | -1.9 |
| -1.9 | 1.9 | -0.7 | 1.9 |

$L_{ch}\cdot\mathbf{y} +$

$L^{+}_{e,2}(\hat{\mathbf{u}})+$

$L^{|}_{a,2}\,(\hat{\mathbf{u}})$

$L_{ch}\,\mathbf{y} + L^{-}_{a,2}\,(\mathbf{u})$

$L_2^{|}(\hat{\mathbf{u}})$

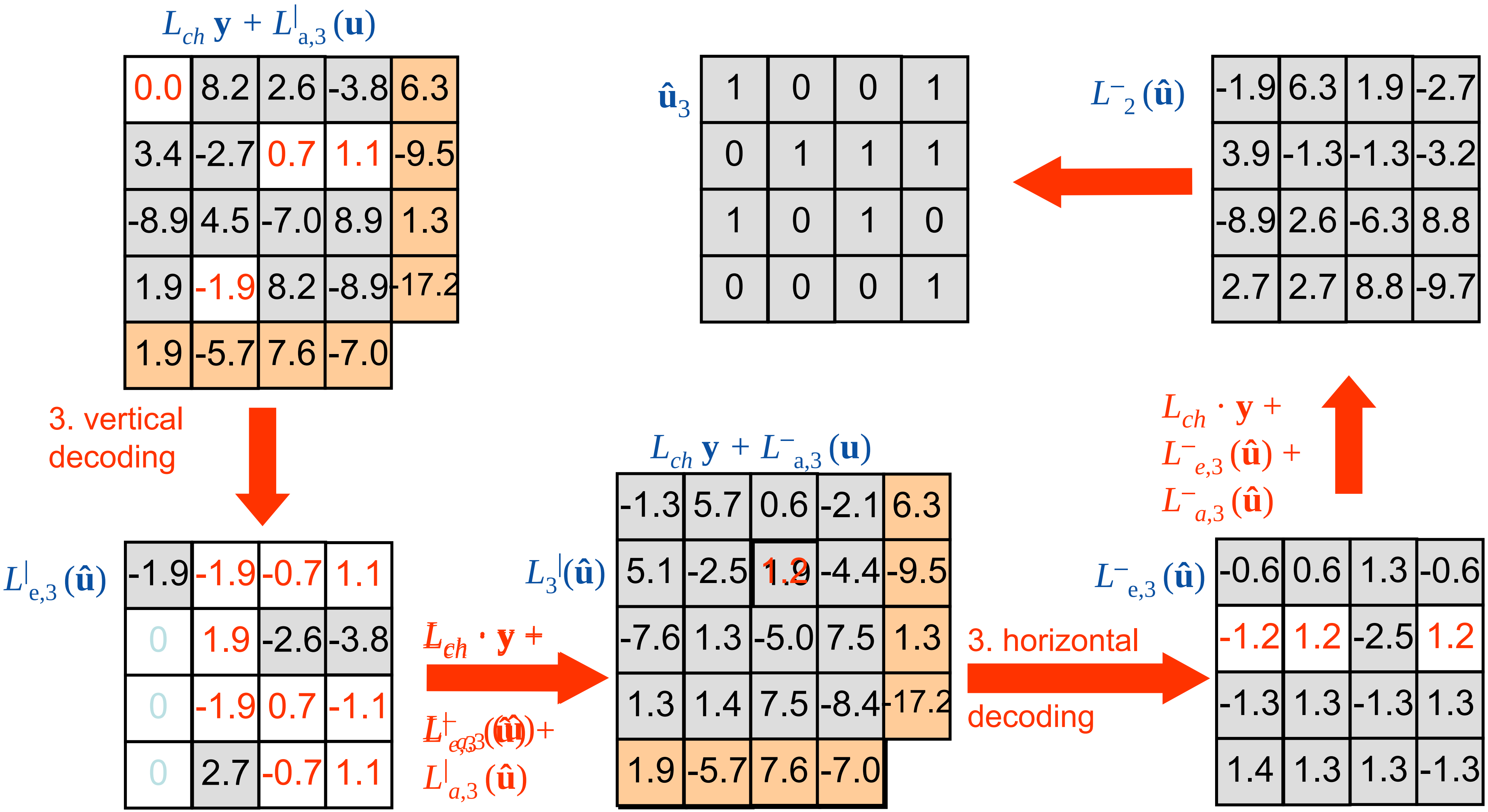
| | | | | |
|---|---|---|---|---|
| -1.3 | 7.0 | 0.6 | -1.3 | 6.3 |
| 3.2<br>1.2 | 6.3 | 1.4 | -3.1<br>0.6 | -9.5 |
| 0.7 | -1.3 | -1.4 | -0.6 | 1.3 |
| -7.0 | 3.9 | -6.3 | 7.0 | -17.2 |
| 1.3 | 1.3 | 8.2 | -8.3 | |

2. horizontal decoding

$L^{-}_{e,2}\,(\hat{\mathbf{u}})$

| | | | |
|---|---|---|---|
| -0.6 | 0.6 | 1.3 | -0.6 |
| -1.7 | 1.7 | -3.1 | 1.7 |
| -1.3 | 1.3 | -1.3 | 1.3 |
| 0.6 | -0.6 | -0.6 | 0.6 |

# Turbo Decoding of (24,16,3)-Produktcode (4)



$L_{ch}\,\mathbf{y} + L^{|}_{a,3}\,(\mathbf{u})$

| | | | | |
|---|---|---|---|---|
| 0.0 | 8.2 | 2.6 | -3.8 | 6.3 |
| 3.4 | -2.7 | 0.7 | 1.1 | -9.5 |
| -8.9 | 4.5 | -7.0 | 8.9 | 1.3 |
| 1.9 | -1.9 | 8.2 | -8.9 | -17.2 |
| 1.9 | -5.7 | 7.6 | -7.0 | |

$\hat{\mathbf{u}}_3$

| | | | |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 |

$L^{-}_{2}\,(\hat{\mathbf{u}})$

| | | | |
|---|---|---|---|
| -1.9 | 6.3 | 1.9 | -2.7 |
| 3.9 | -1.3 | -1.3 | -3.2 |
| -8.9 | 2.6 | -6.3 | 8.8 |
| 2.7 | 2.7 | 8.8 | -9.7 |

3. vertical decoding

$L^{|}_{e,3}\,(\hat{\mathbf{u}})$

| | | | |
|---|---|---|---|
| -1.9 | -1.9 | -0.7 | 1.1 |
| 0 | 1.9 | -2.6 | -3.8 |
| 0 | -1.9 | 0.7 | -1.1 |
| 0 | 2.7 | -0.7 | 1.1 |

$L_{ch}\cdot\mathbf{y} +$
$L^{|}_{e,3}\,(\hat{\mathbf{u}}) +$
$L^{|}_{a,3}\,(\hat{\mathbf{u}})$

$L_{ch}\,\mathbf{y} + L^{-}_{a,3}\,(\mathbf{u})$

$L_3^{|}(\hat{\mathbf{u}})$

| | | | | |
|---|---|---|---|---|
| -1.3 | 5.7 | 0.6 | -2.1 | 6.3 |
| 5.1 | -2.5 | 1.2 | -4.4 | -9.5 |
| -7.6 | 1.3 | -5.0 | 7.5 | 1.3 |
| 1.3 | 1.4 | 7.5 | -8.4 | -17.2 |
| 1.9 | -5.7 | 7.6 | -7.0 | |

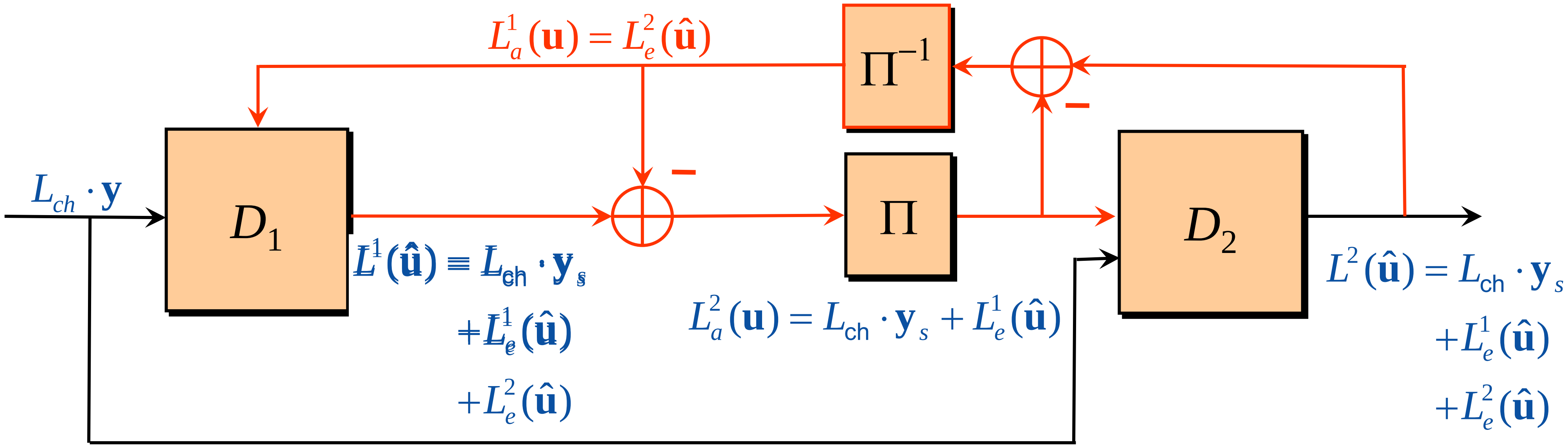
3. horizontal decoding

$L^{-}_{e,3}\,(\hat{\mathbf{u}})$

| | | | |
|---|---|---|---|
| -0.6 | 0.6 | 1.3 | -0.6 |
| -1.2 | 1.2 | -2.5 | 1.2 |
| -1.3 | 1.3 | -1.3 | 1.3 |
| 1.4 | 1.3 | 1.3 | -1.3 |

$L_{ch}\cdot\mathbf{y} +$
$L^{-}_{e,3}\,(\hat{\mathbf{u}}) +$
$L^{-}_{a,3}\,(\hat{\mathbf{u}})$

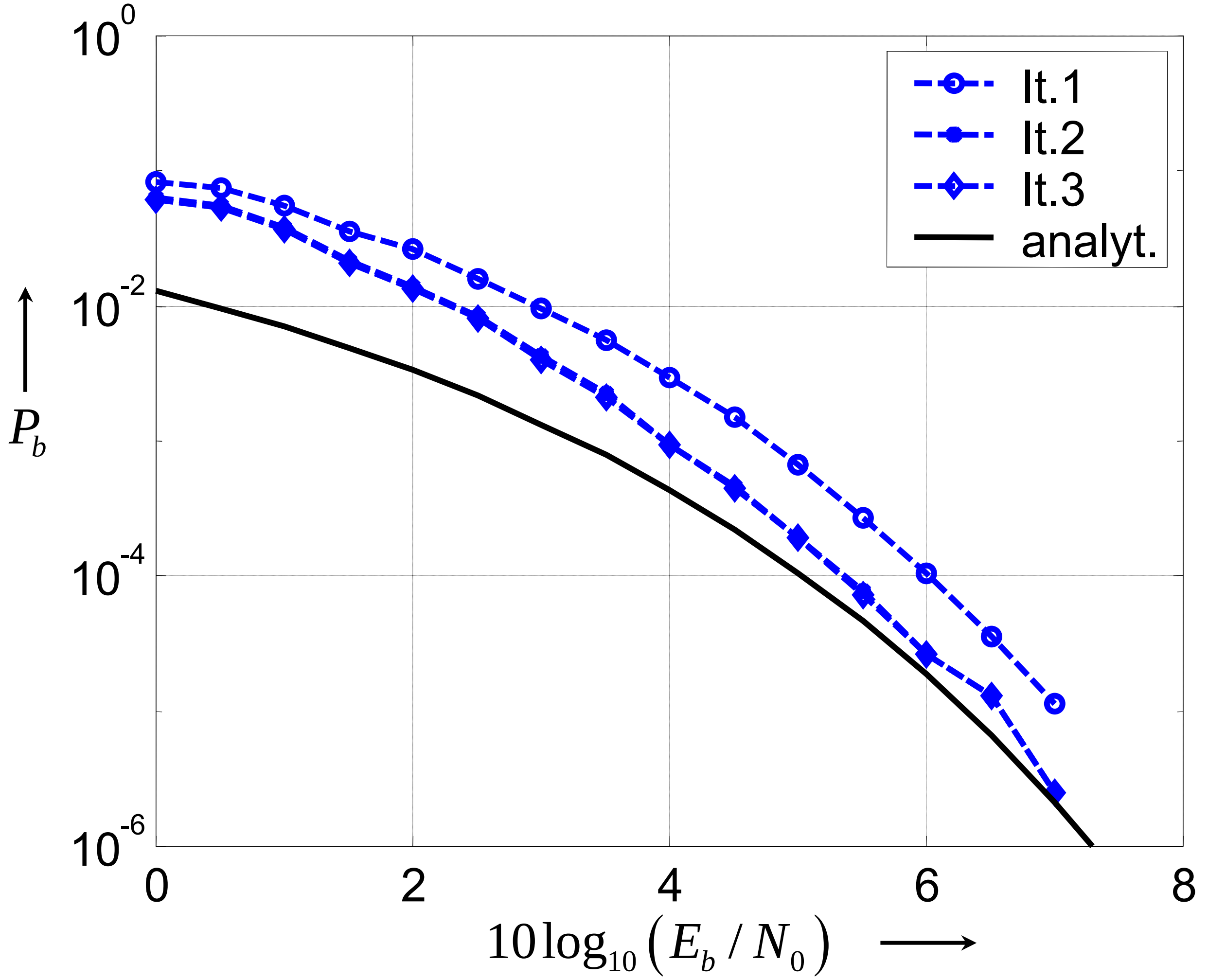
# General Concept of „Turbo“ Decoding



- Each decoder supplies extrinsic information as a priori information to other decoder
- $L_\text{e}(\hat{\mathbf{u}})$ is incorporated in $L(\hat{\mathbf{u}})$ for systematic encoders
- Improvement by additional decoding iteration with a-priori knowledge if $L_\text{e}(\hat{\mathbf{u}})$, $L_\text{a}(\hat{\mathbf{u}})$ and $L_\text{ch}\mathbf{y}_\text{s}$ are statistically independent

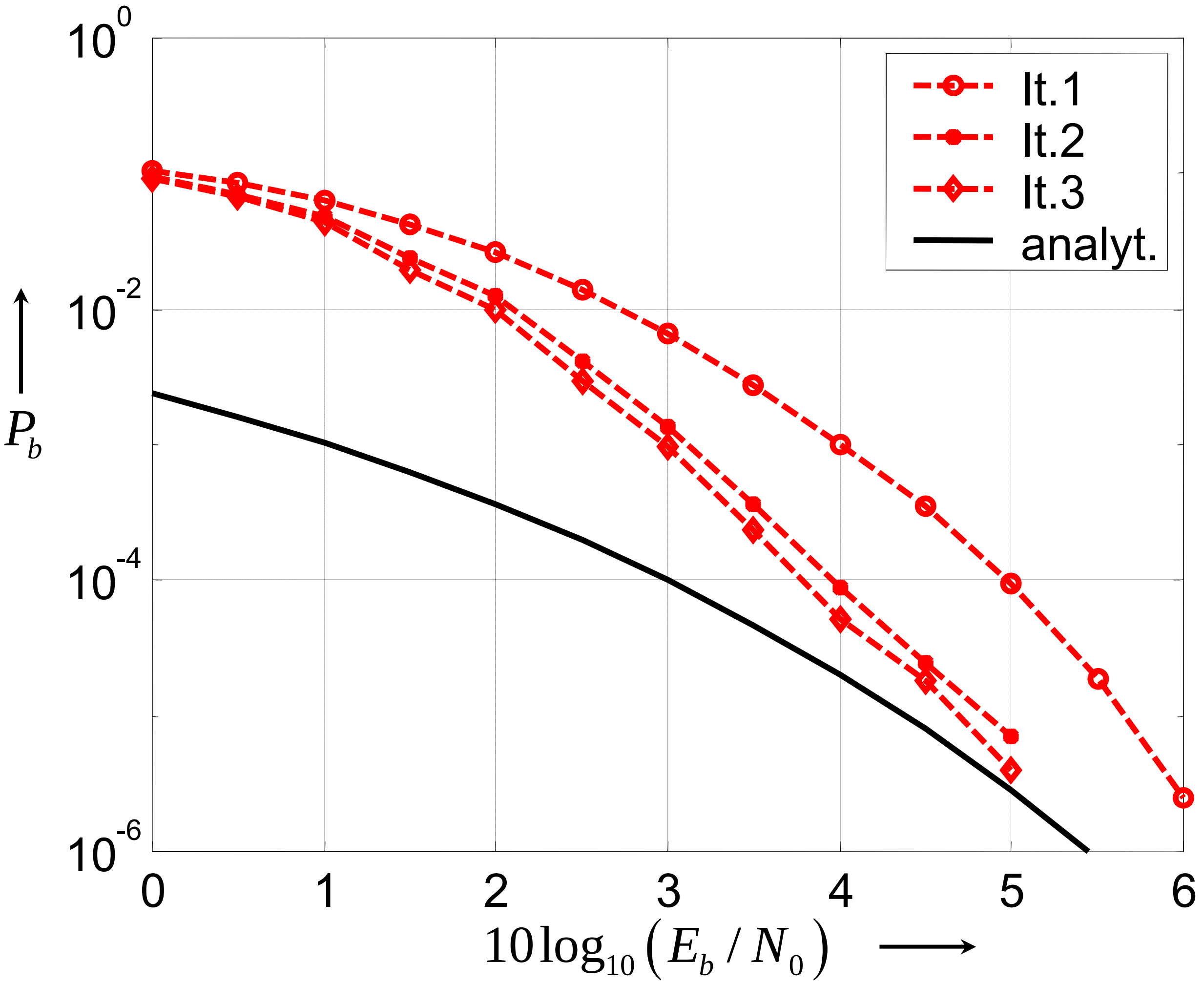
# Simulation Results for Product Code

- (7,4,3)-Hamming Codes, parallel concatenation



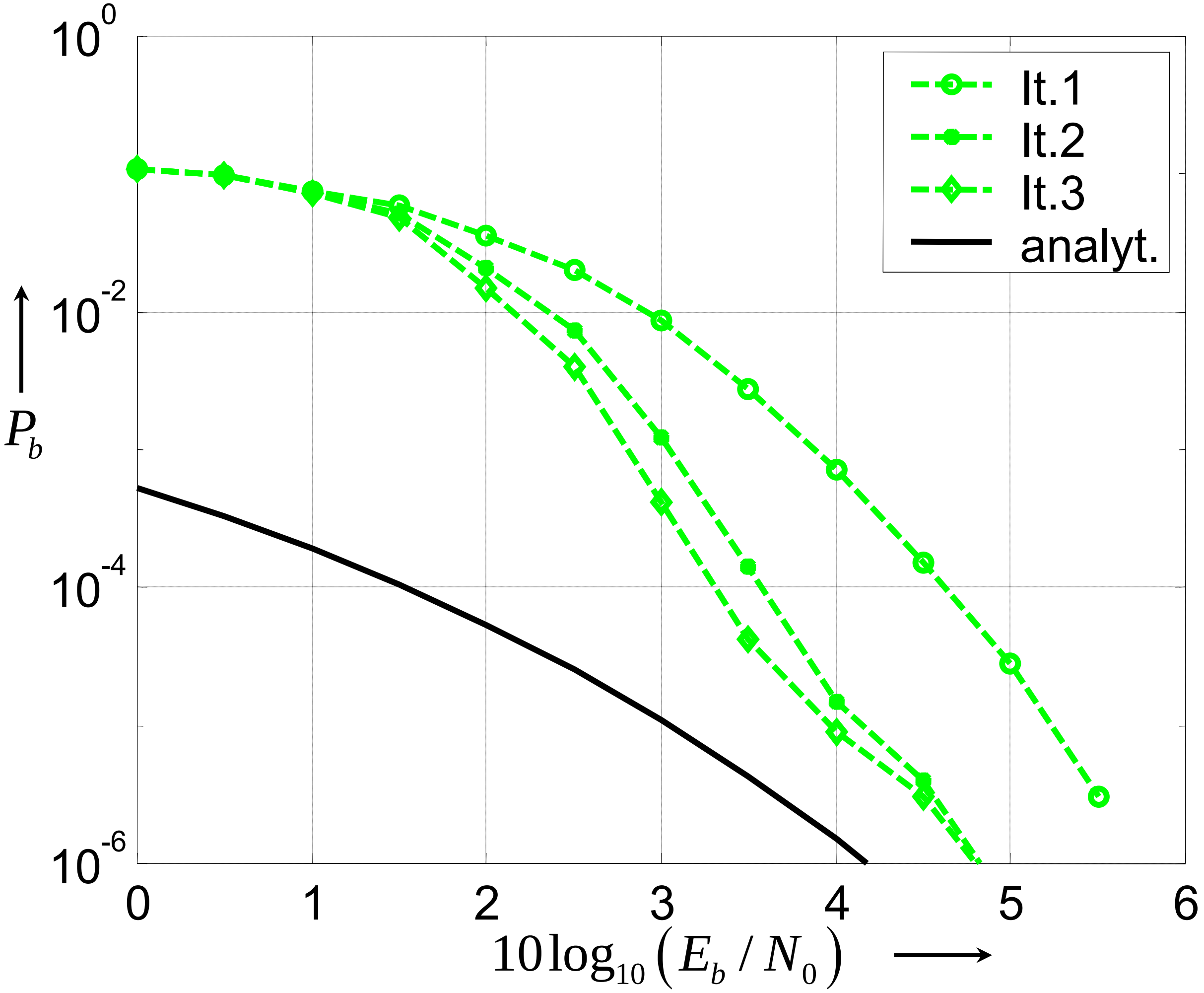
# Simulation Results for Product Codes

- (15,11,3)-Hamming-Codes, parallel concatenation



# Simulation Results for Product Codes

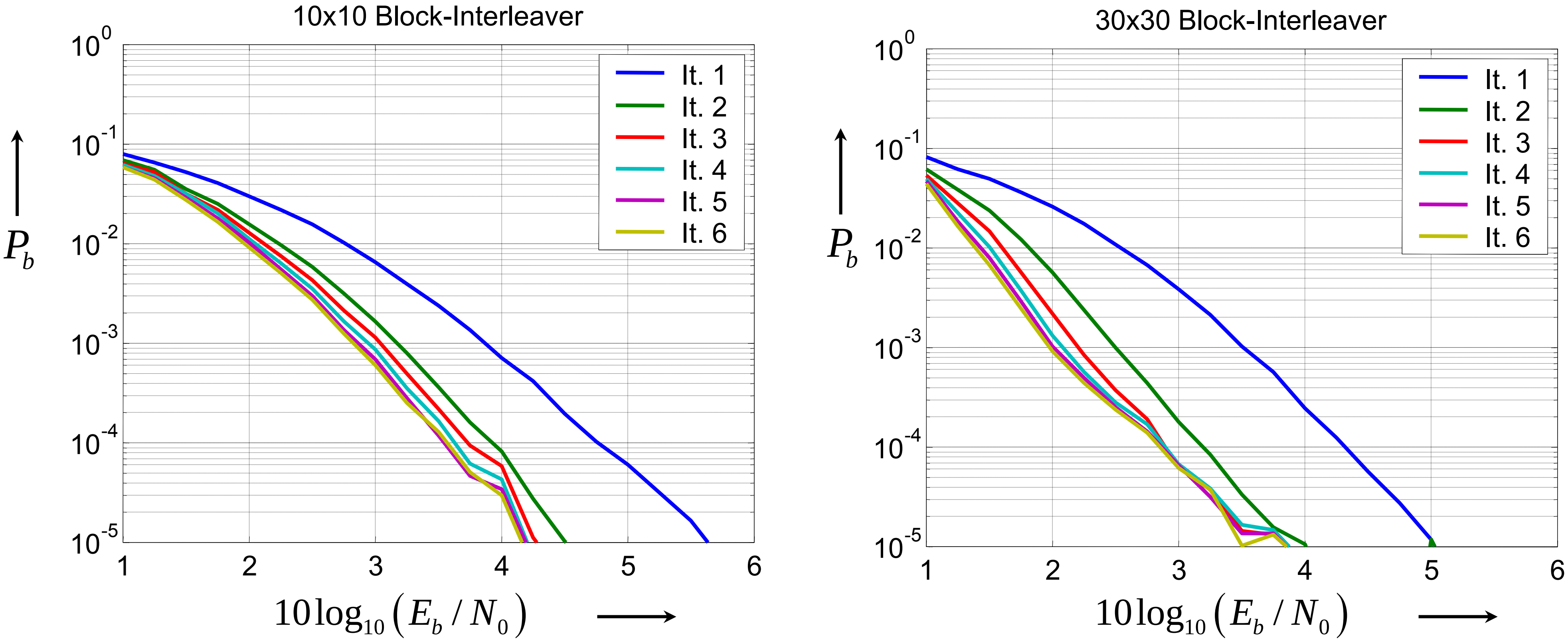
- (31,26,3)-Hamming-Codes, parallel concatenation



# Simulation Results for Turbo Codes ($L_c = 3$)

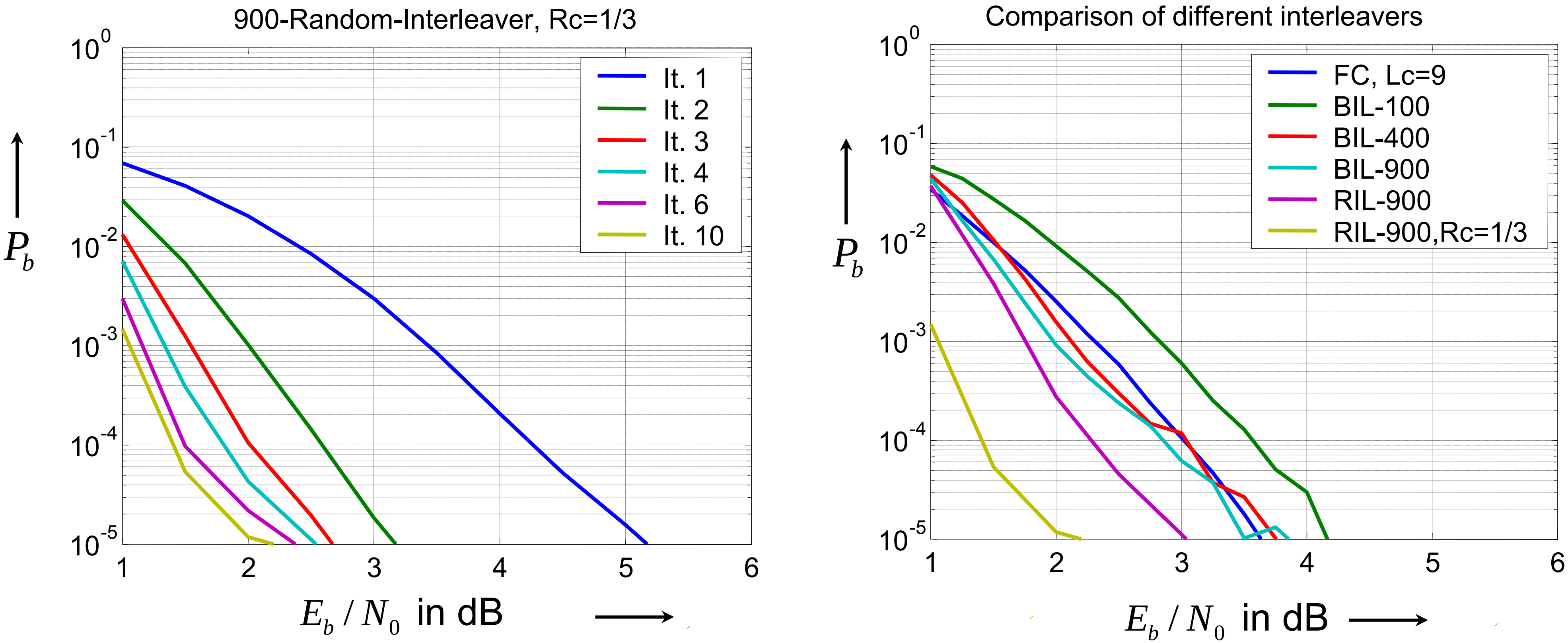
Institut für Nachrichtentechnik

- Simple Block Interleaver
  - No significant improvements after third decoding iteration



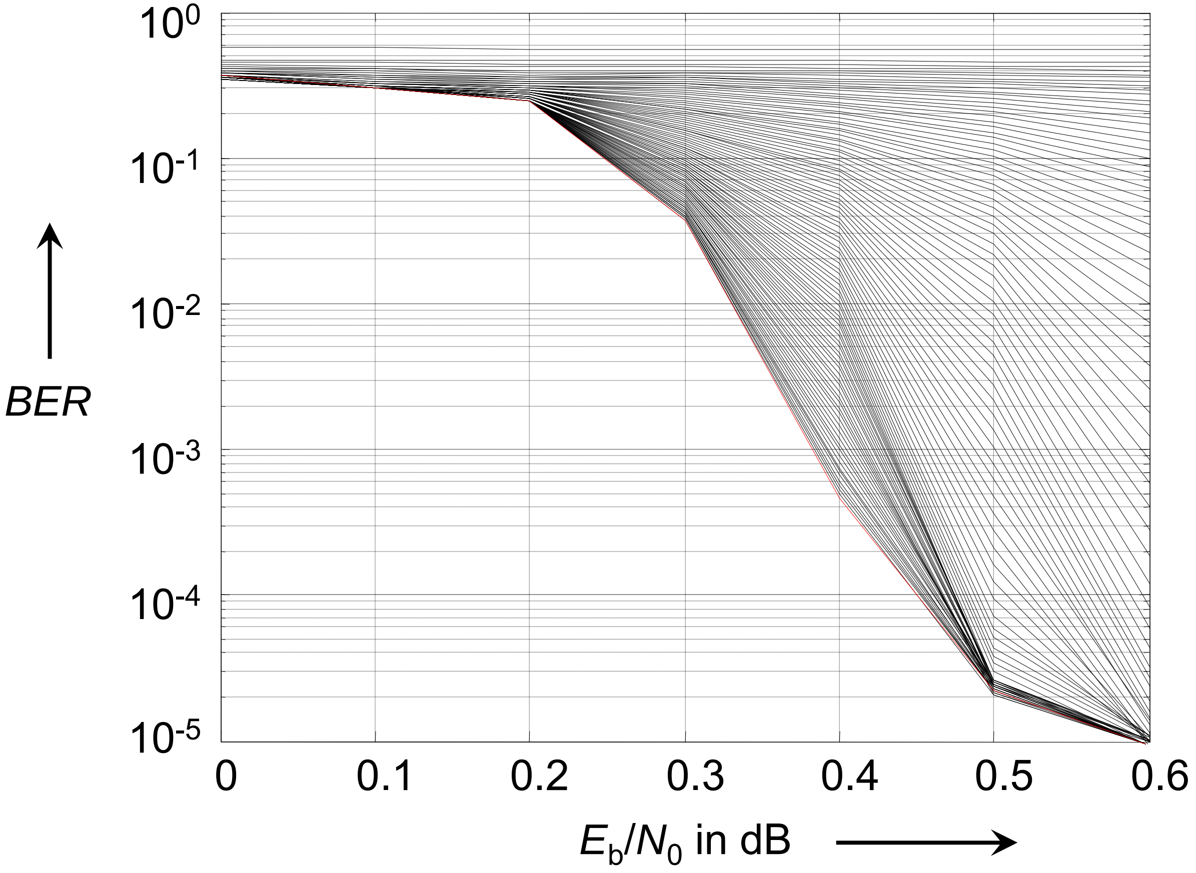
# Simulation Results for Turbo Codes (Lc = 3)

- Block and Random Interleavers
  - Iterative process gains significantly even after sixth iteration
  - Increasing interleaver size improves performance remarkably

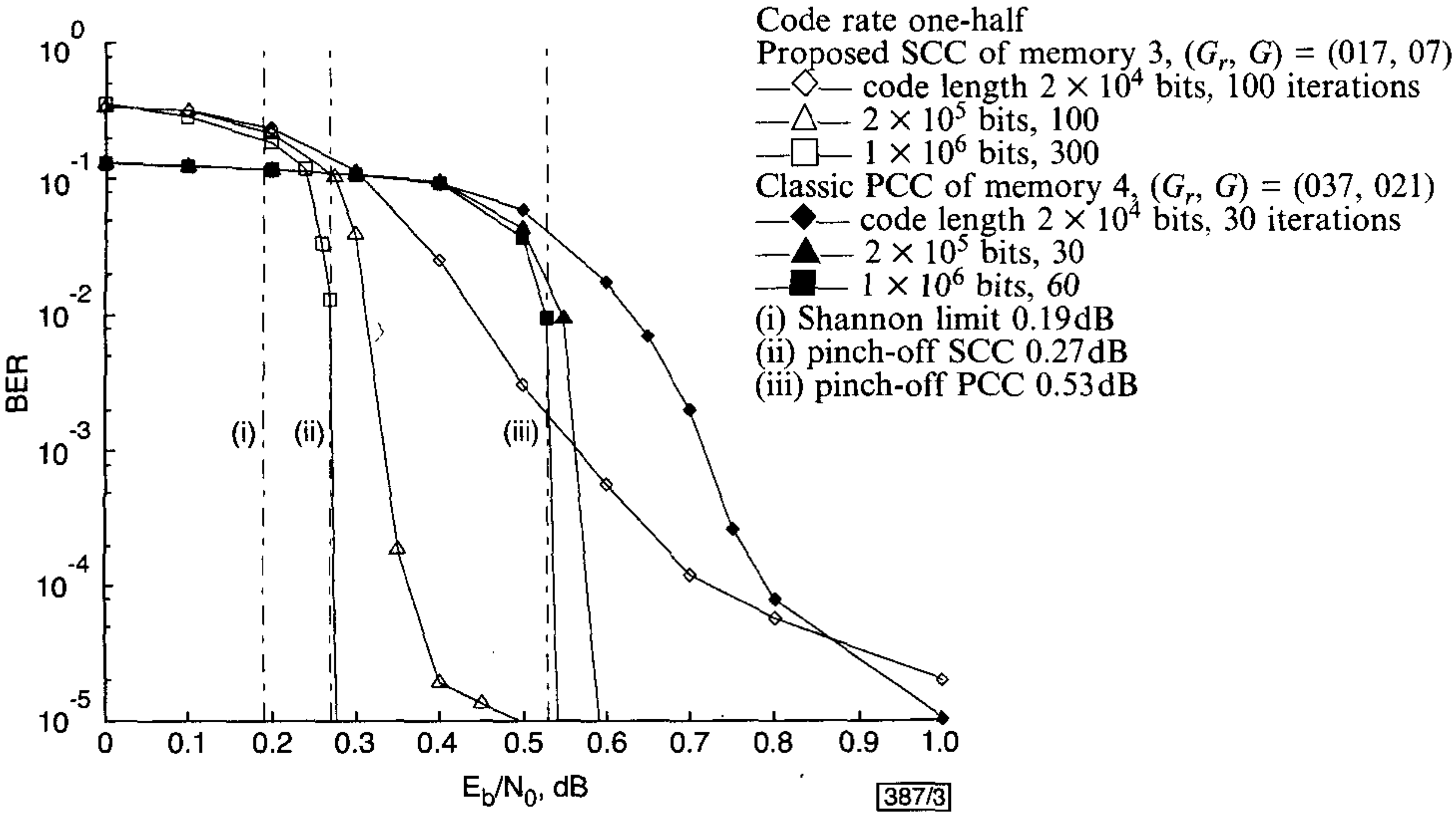


# Repeat Accumulate Code by ten Brink

- Half-rate outer repetition encoder
- Rate-one inner recursive convolutional encoder
- Approximately 100 decoding iterations are needed



# Repeat Accumulate Code by ten Brink

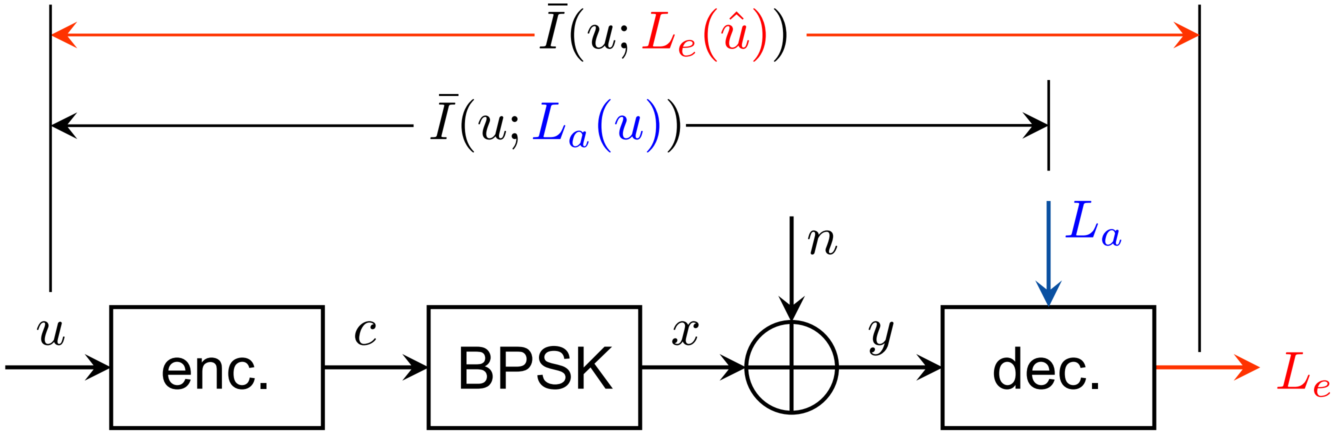


# Analysis of Turbo Decoding Process

- For a long time, turbo decoding process has not really been understood
- Several approaches for the analysis exist:
  - Density evolution
  - EXIT chart analysis

- Basically, decoders exchange extrinsic mutual information
- Semi-analytic analysis
  - Simulation results showed that extrinsic LLRs are approximately Gaussian distributed (at least after many iterations)
  - Generate Gaussian distributed artificial a priori LLRs with specified mutual information $\bar{I}\left(u;L_a(\hat{u})\right)$
  - Determine by simulations extrinsic mutual information $\bar{I}\left(u;L_e(\hat{u})\right)$ at decoder output
  - Behavior of decoder is characterized by this relationship

# Analysis of Decoder



- Artifical a-priori LLRs: $L_a = \frac{2}{\sigma^2_{\mathcal{N}_a}} \cdot y_a = \frac{2}{\sigma^2_{\mathcal{N}_a}} \cdot (x + n_a)$

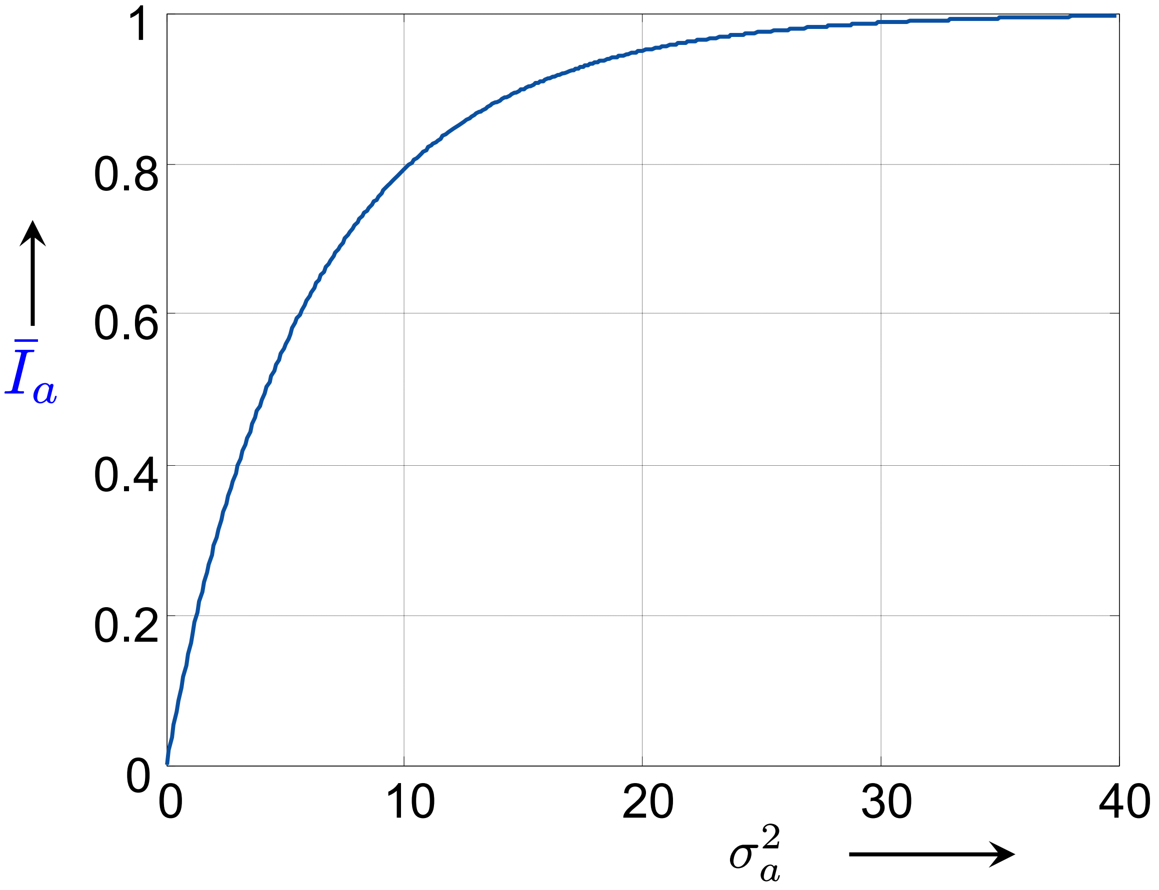
  with Gaussian noise $n_a \in \mathcal{N}(0, \sigma^2_{\mathcal{N}_a})$ and $\bar{I}_a = \bar{I}(u; L_a(u))$

- Measure extrinsic mutual information at decoder output $\bar{I}_e = \bar{I}(u; L_e(\hat{u}))$

# Modeling a priori Information

- Mutual a-priori information only depends on noise variance

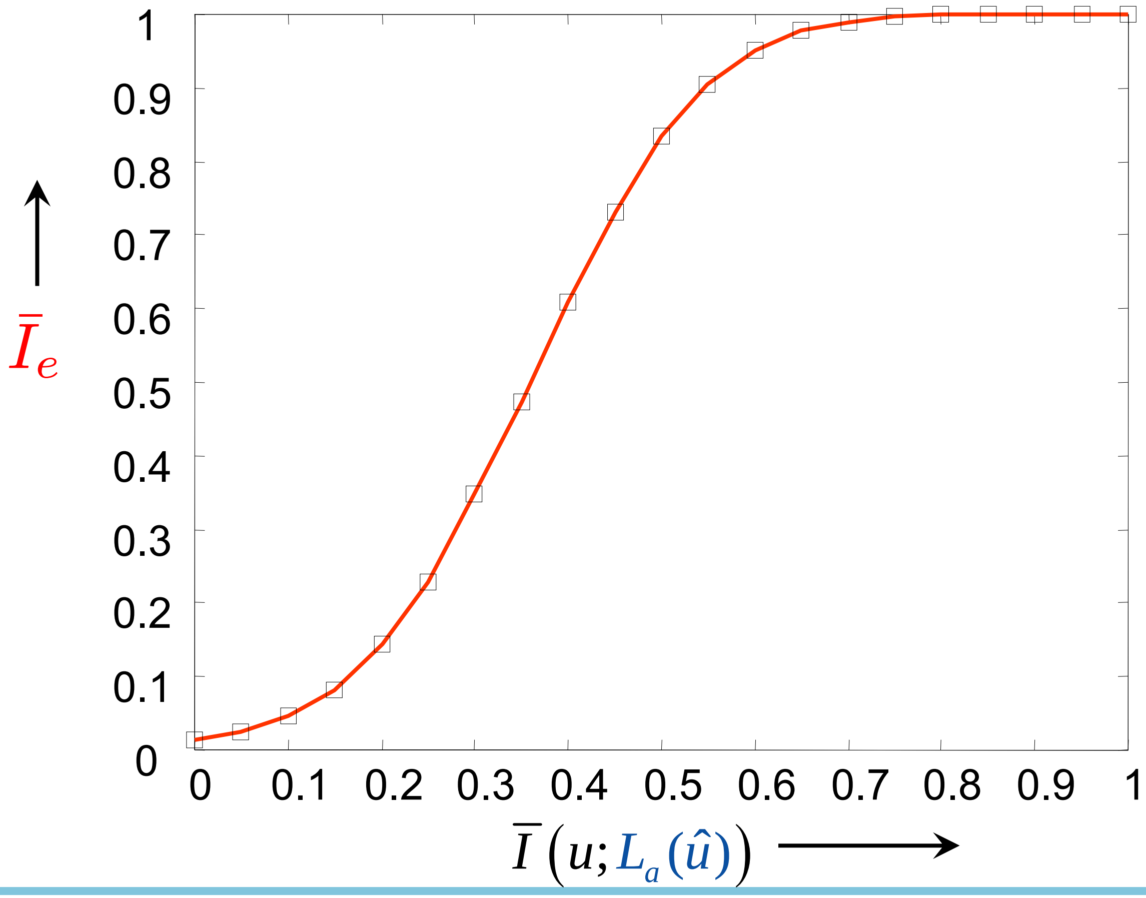
$$\bar{I}_a = 1 + \frac{1}{2} \cdot \sum_{\mu=0}^{1} \int_{-\infty}^{\infty} p(L_a \mid x = X_\mu) \cdot \log_2 \left[ \frac{p(L_a \mid x = X_\mu)}{p(L_a \mid x = X_0) + p(L_a \mid x = X_1)} \right] dL_a$$

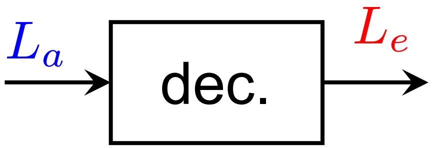


# Transfer Characteristic of Decoder

- Dependency of mutual information at decoder input and output

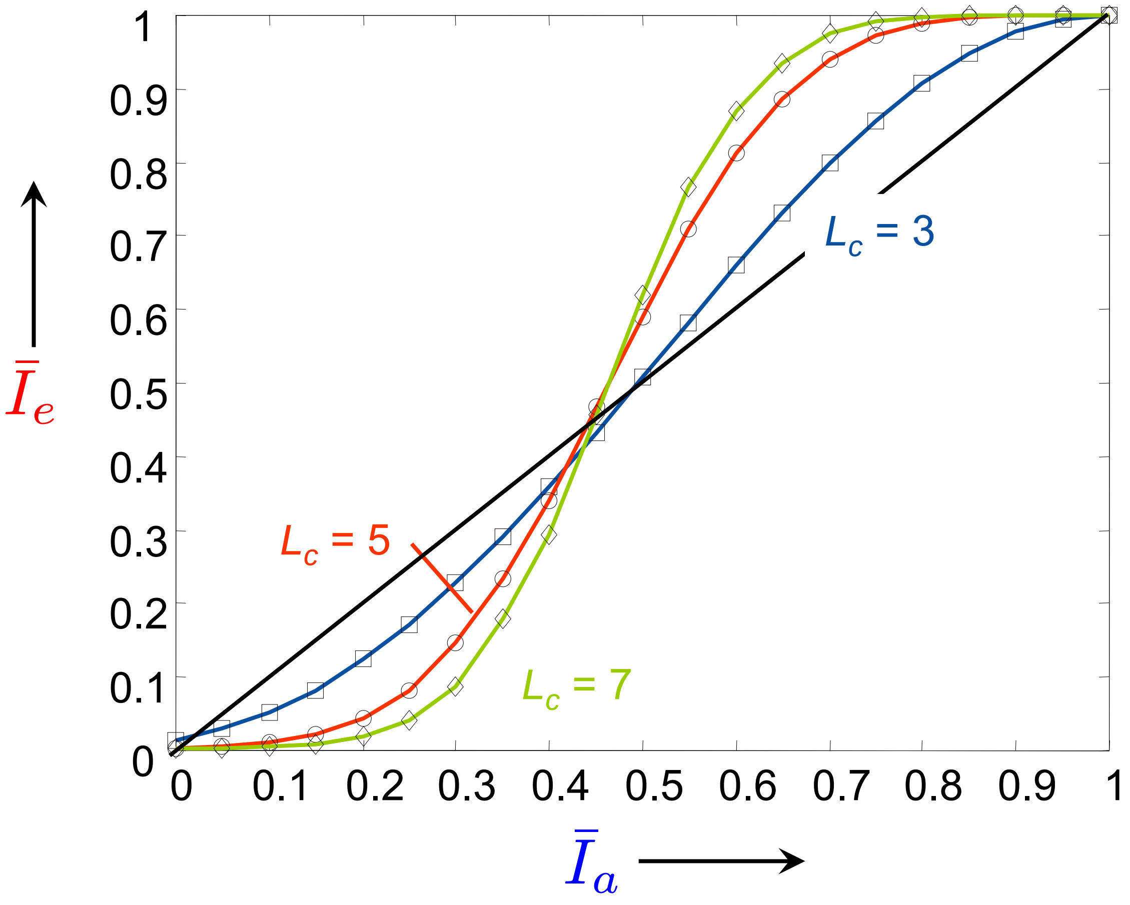
$$\bar{I}_e = 1 + \frac{1}{2} \cdot \sum_{\mu=0}^{1} \int_{-\infty}^{\infty} \hat{p}(L_e \mid x = X_\mu) \cdot \log_2 \left[ \frac{\hat{p}(L_e \mid x = X_\mu)}{\hat{p}(L_e \mid X_0) + \hat{p}(L_e \mid X_1)} \right] dL_e$$





# Behavior for different NSC Codes

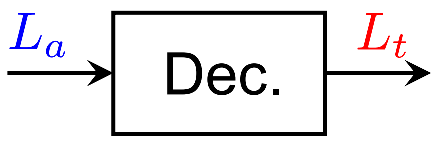
- Only a-priori information at decoder input (no channel output)

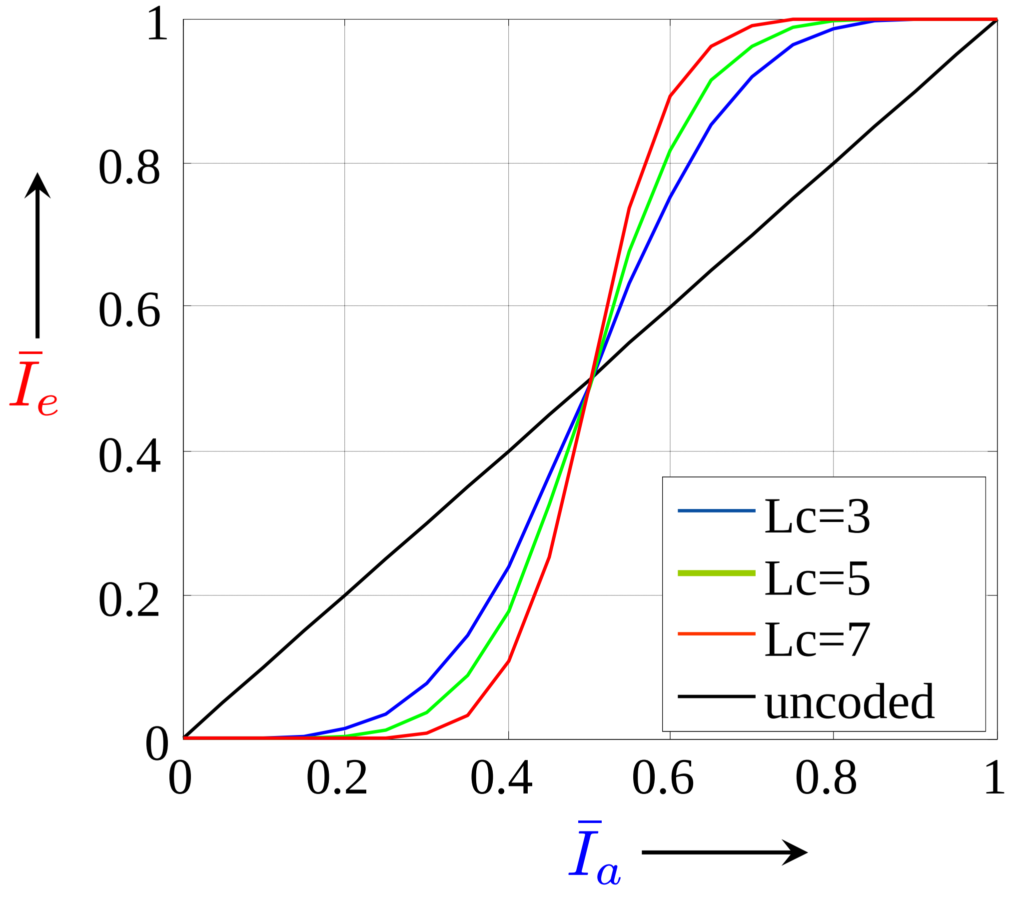


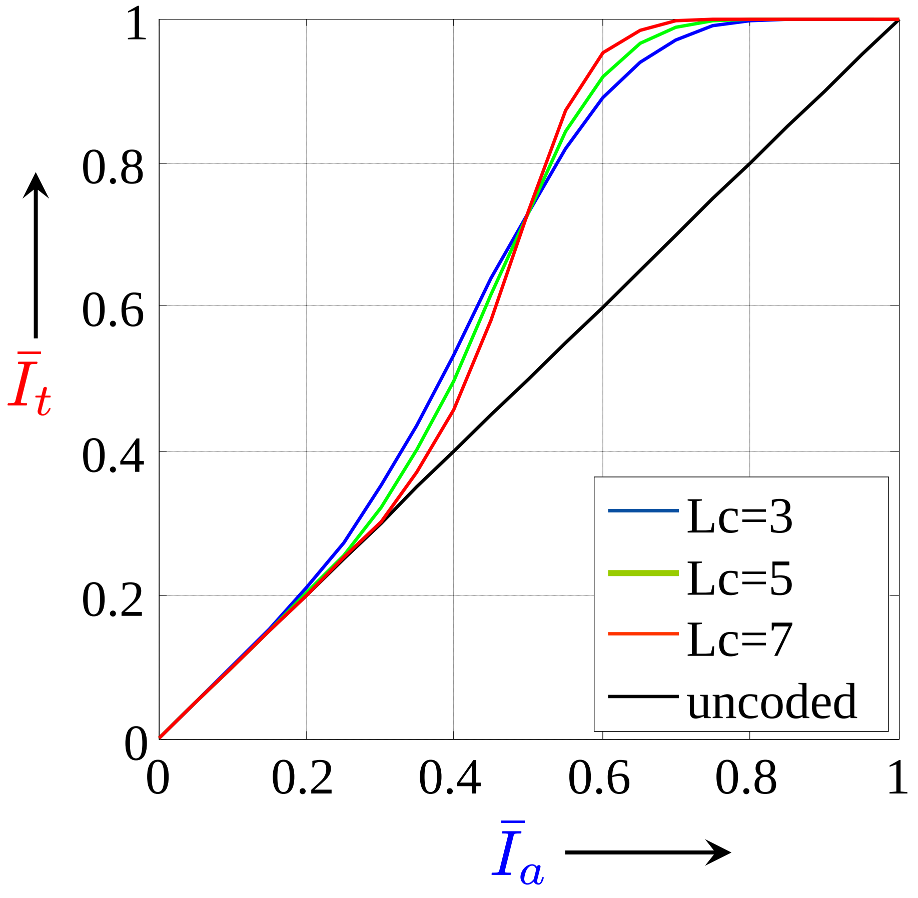
  - Weak codes better for low a priori information
  - Strong codes better for high a priori information
  - Point of intersection for all convolutional codes at (0.5,0.5) (explanation for this behavior unknown!)

# Extrinsic and Total Information

- Only a-priori information at decoder input
- Extrinsic and a priori information do **not** sum up

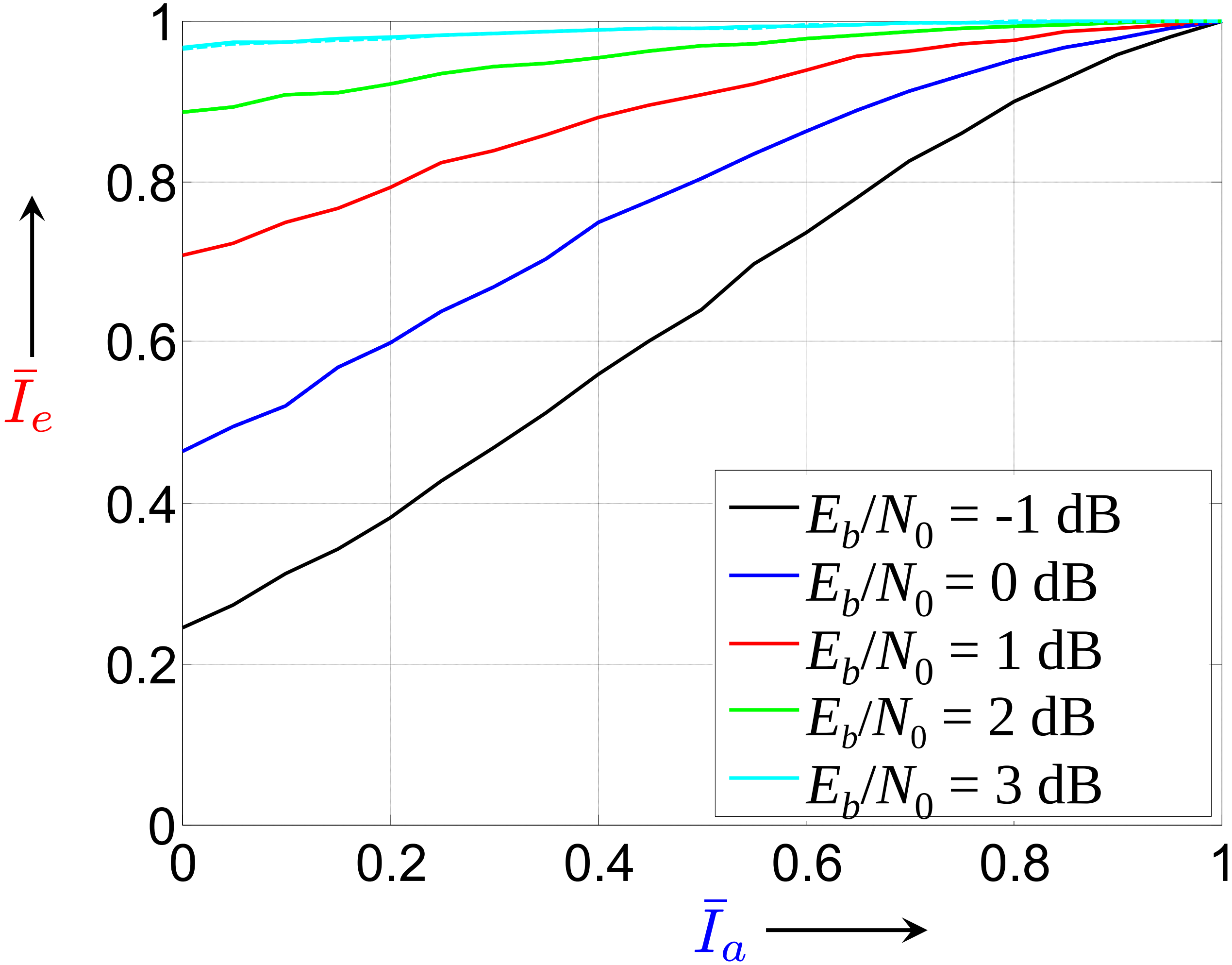




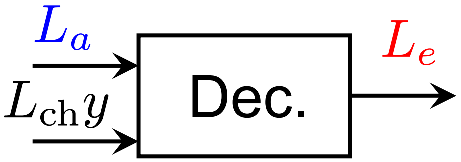


# Intrinsic and a priori Input

- A-priori and intrinsic information at decoder input

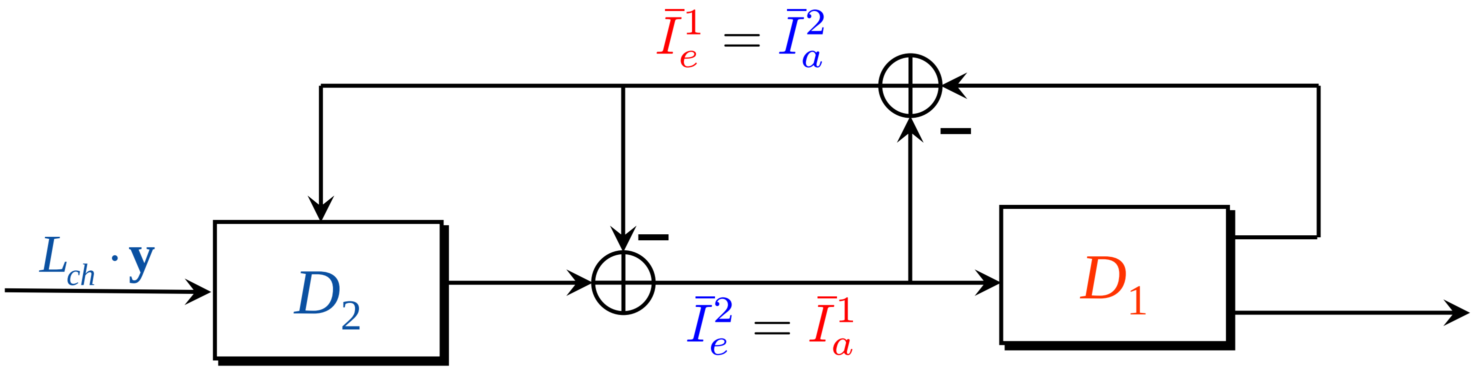


  - High channel SNR leads to high extrinsic information
  - Large a-priori information can compensate bad channel conditions



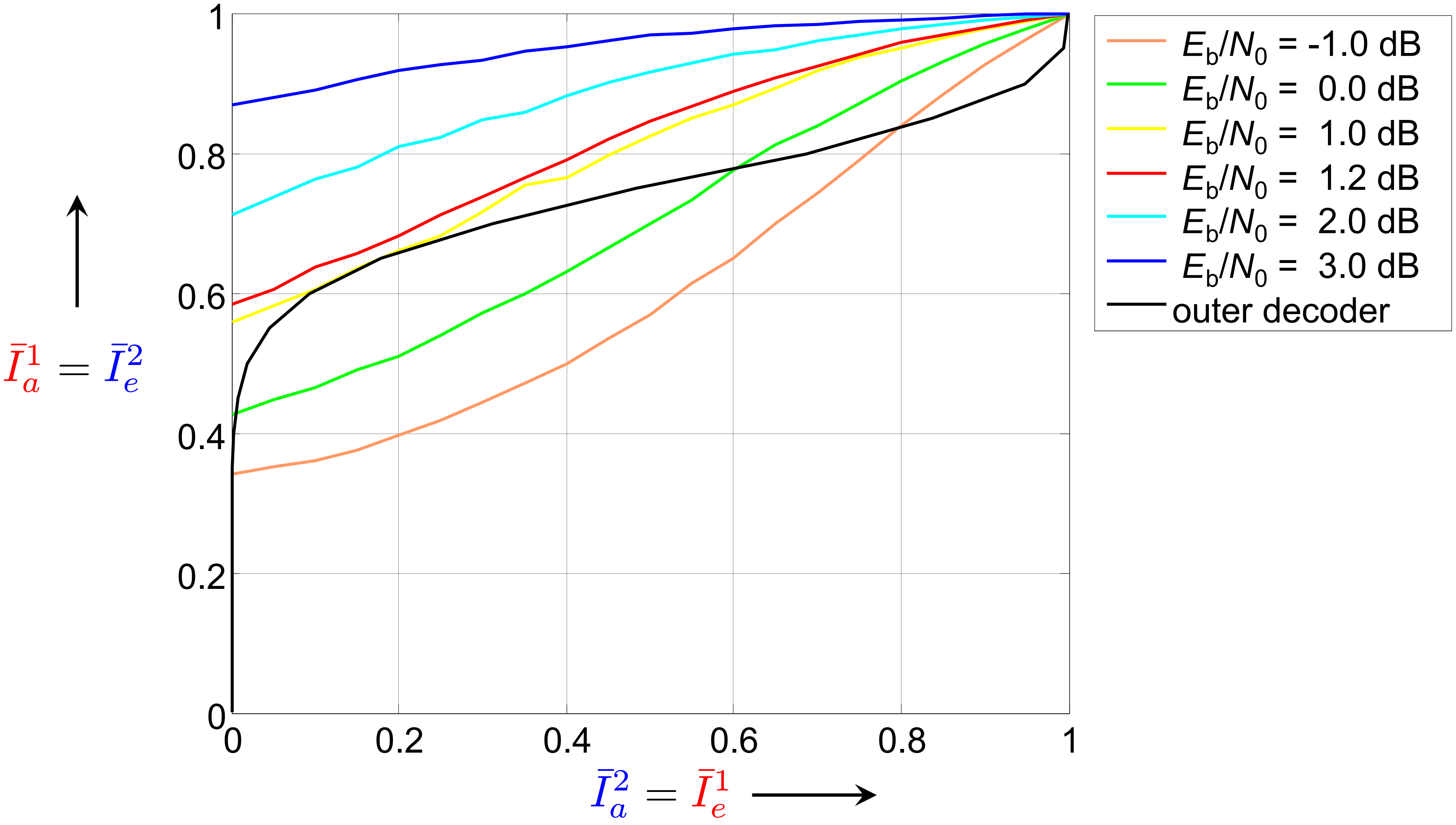
# EXIT Charts

- EXtrinsic Information Transfer (EXIT)
- Serial concatenation of inner Walsh Hadamard Code and outer convolutional code
- Extrinsic information of one inner Hadamard decoder is a priori information for outer convolutional decoder and vice versa
- Drawing trajectories of both decoders into one diagram and flipping abscissa and ordinate for outer convolutional code leads to EXIT charts



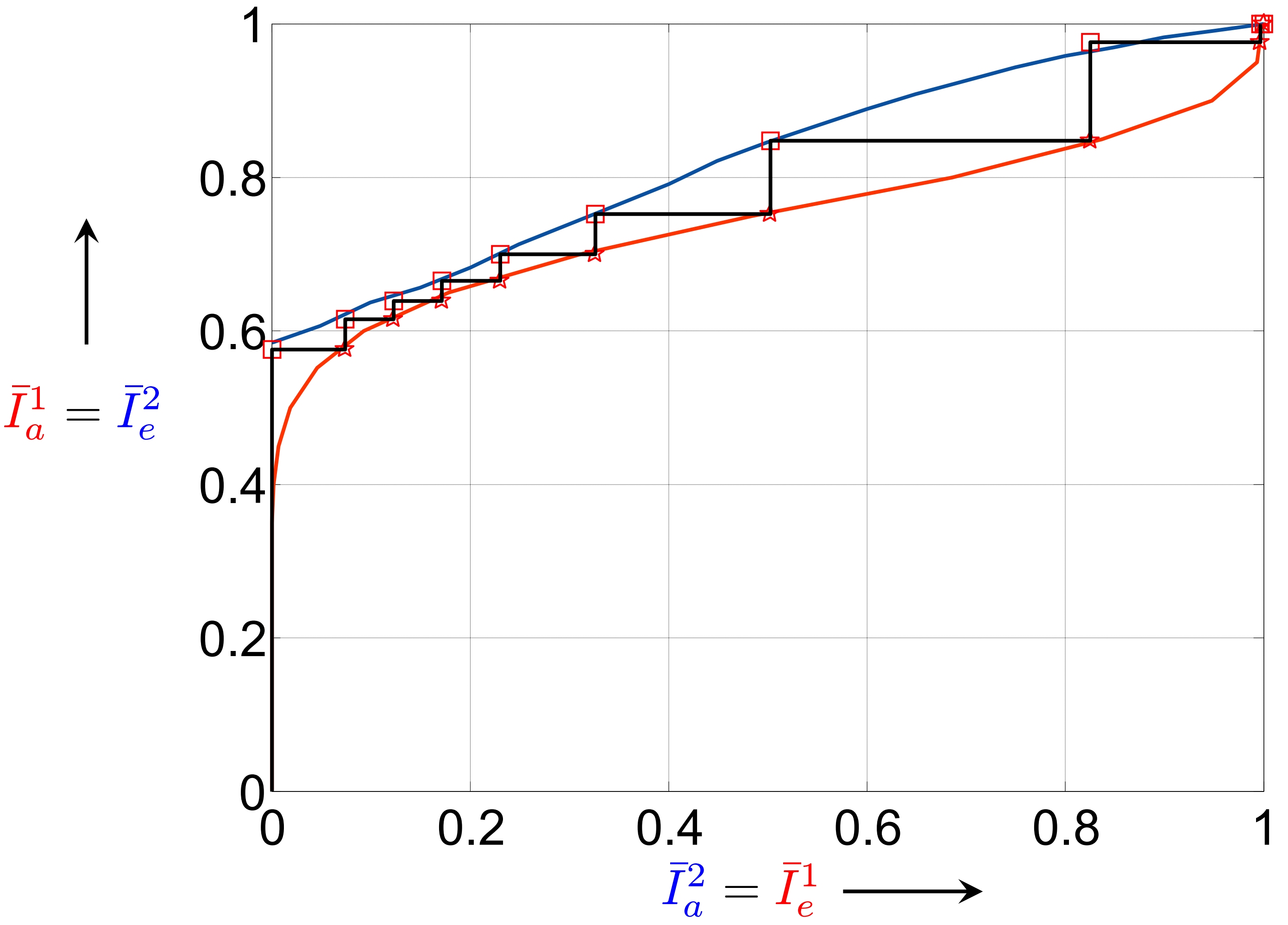
# EXIT Charts

- Determining pinch-off SNR: minimum SNR for convergence



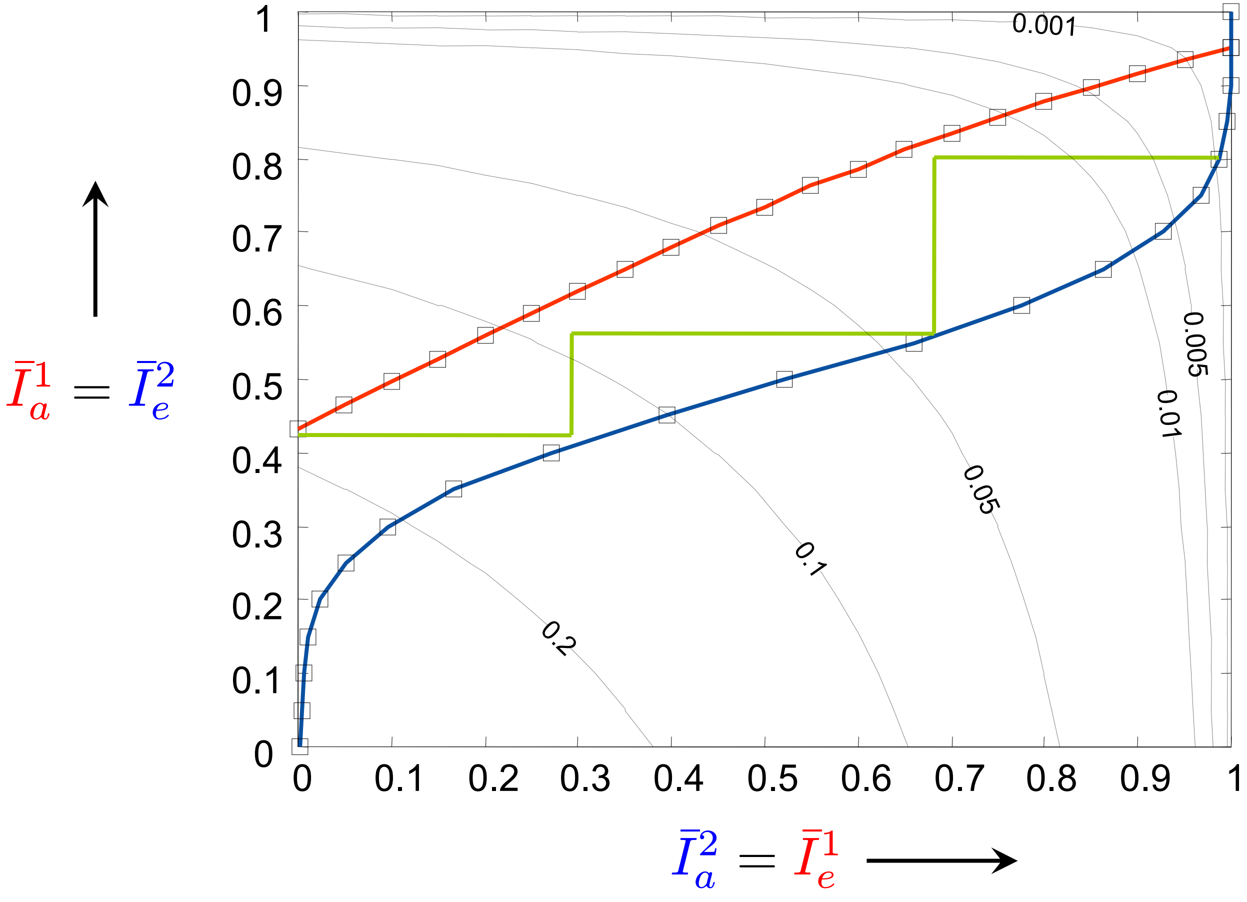
# EXIT Charts

- Serial concatenation of inner RSC and outer NSC code



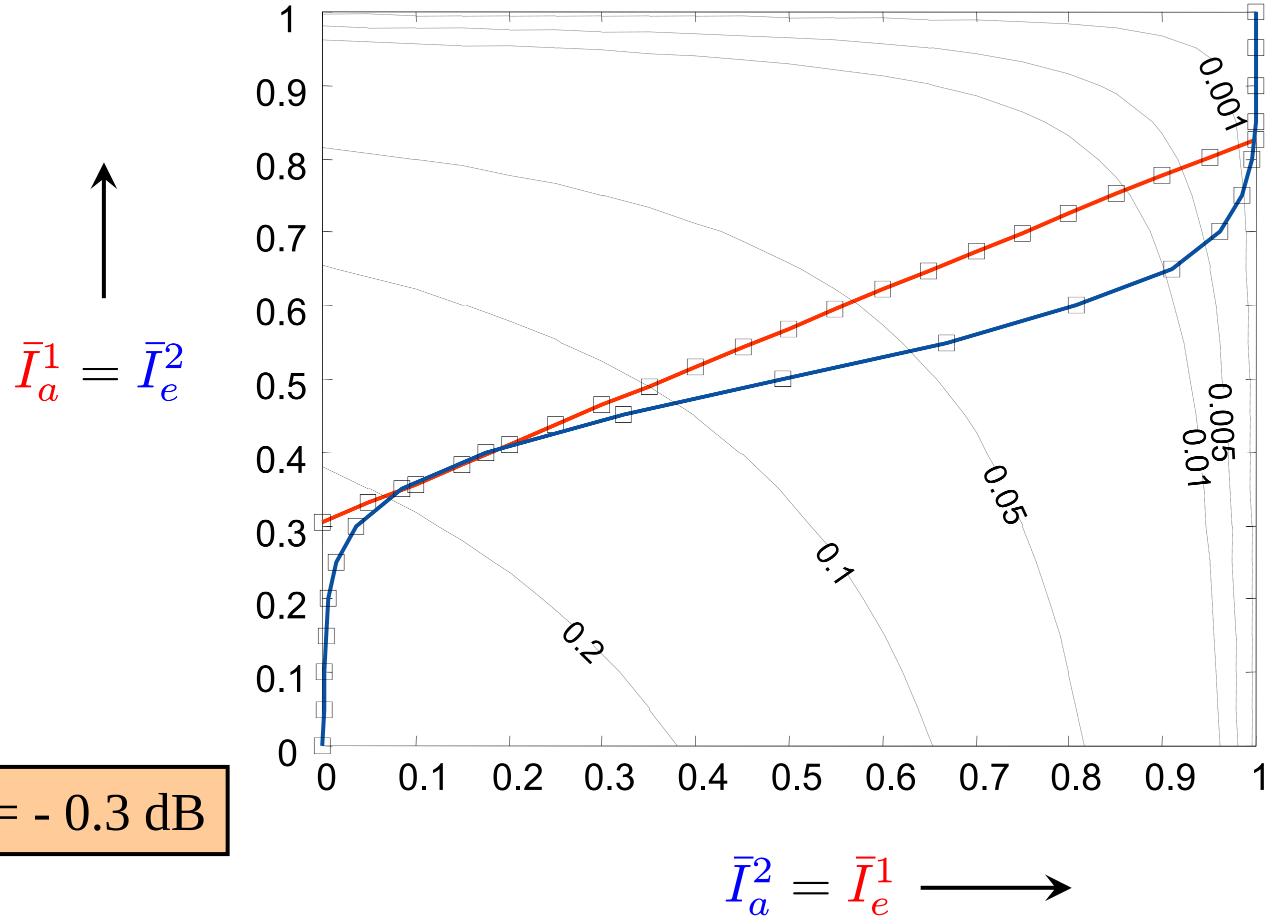
# EXIT Charts

- Outer convolutional code
- Inner Walsh-Hadamard code



# EXIT Charts

- Determining pinch-off SNR: minimum SNR for which convergence is maintained



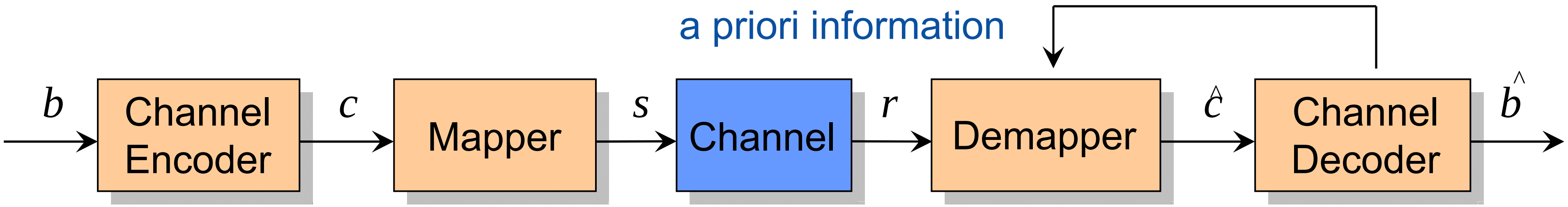
$10\log_{10}(E_b/N_0) = -0.3\text{ dB}$

# Application Areas of Turbo Detection

- Application of turbo processing not restricted to concatenated codes
- Applicable for any concatenated system
  - Concatenation of source and channel coding
    (exploitation of residual redundancy from source coding)
  - Concatenation of coding and modulation
    (bit-interleaved coded modulation)
  - Channel equalization and decoding can be performed iteratively
  - Multi-user detection and decoding can be performed iteratively

# Thanks for your attention!

# Bit-Interleaved Coded Modulation (BICM)



a priori information

$b$ Channel Encoder $c$ Mapper $s$ Channel $r$ Demapper $\hat{c}$ Channel Decoder $\hat{b}$

- Coded transmission with higher order modulation:
  - Usually Gray mapping employed
  - Minimizes bit error probability without channel coding

- Iterative detection and decoding
  - Channel decoder provides a priori information to the demapper

- Are there better mapping strategies than Gray mapping?

# Bitwise Mutual Information

- Bijective mapping of $m = \log_2 M$ interleaved coded bits $c_1,\ldots,c_m$ onto transmit symbol $s$
- Chain rule of mutual information

$$I(s;r) = I(c_1,\ldots,c_m;r) = \sum_{k=1}^{m} I(c_k;r\,|\,c_1,\ldots,c_{k-1}) = \sum_{L=0}^{m-1} I_L$$

- Example: 8-PSK (three coded bits per symbol)

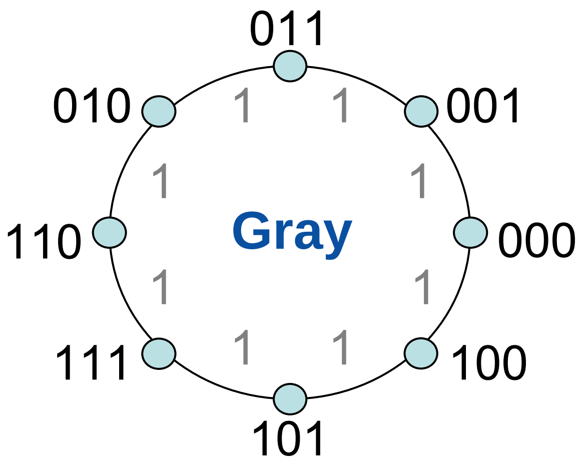
Averaging

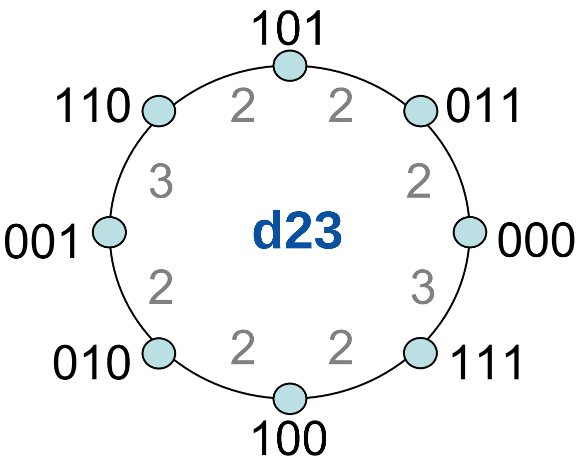
$$\begin{array}{rccccc}
I(s;r) = & I(c_1;r) & + & I(c_2;r\,|\,c_1) & + & I(c_3;r\,|\,c_1,c_2) \\
= & I(c_1;r) & + & I(c_3;r\,|\,c_1) & + & I(c_2;r\,|\,c_1,c_3) \\
= & I(c_2;r) & + & I(c_1;r\,|\,c_2) & + & I(c_3;r\,|\,c_1,c_2) \\
= & I(c_2;r) & + & I(c_3;r\,|\,c_2) & + & I(c_1;r\,|\,c_2,c_3) \\
= & I(c_3;r) & + & I(c_1;r\,|\,c_3) & + & I(c_2;r\,|\,c_1,c_3) \\
= & I(c_3;r) & + & I(c_2;r\,|\,c_3) & + & I(c_1;r\,|\,c_2,c_3) \\
 & I_0 & & I_1 & & I_2
\end{array}$$

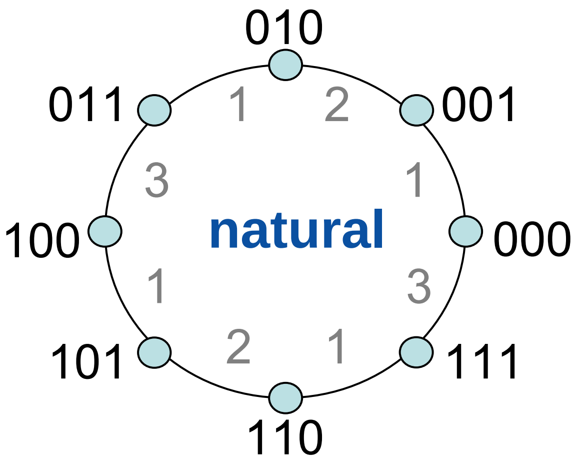
$I_L$: average bitwise mutual information if $L$ bits are already known to the receiver

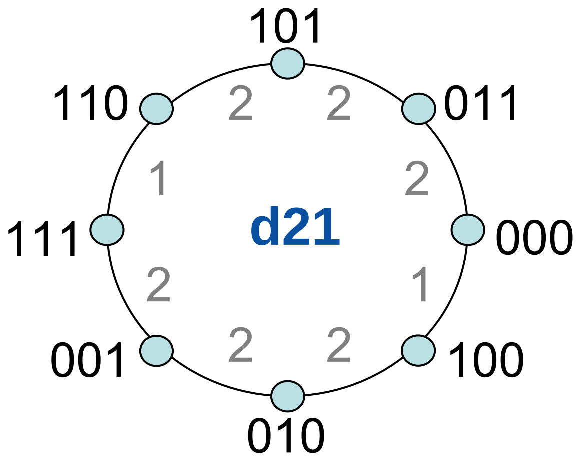
→ parallel sub-channels

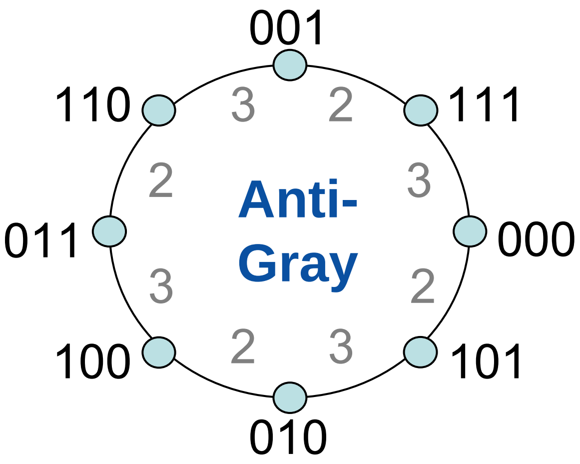
# Selected Bit-Mappings for 8-PSK











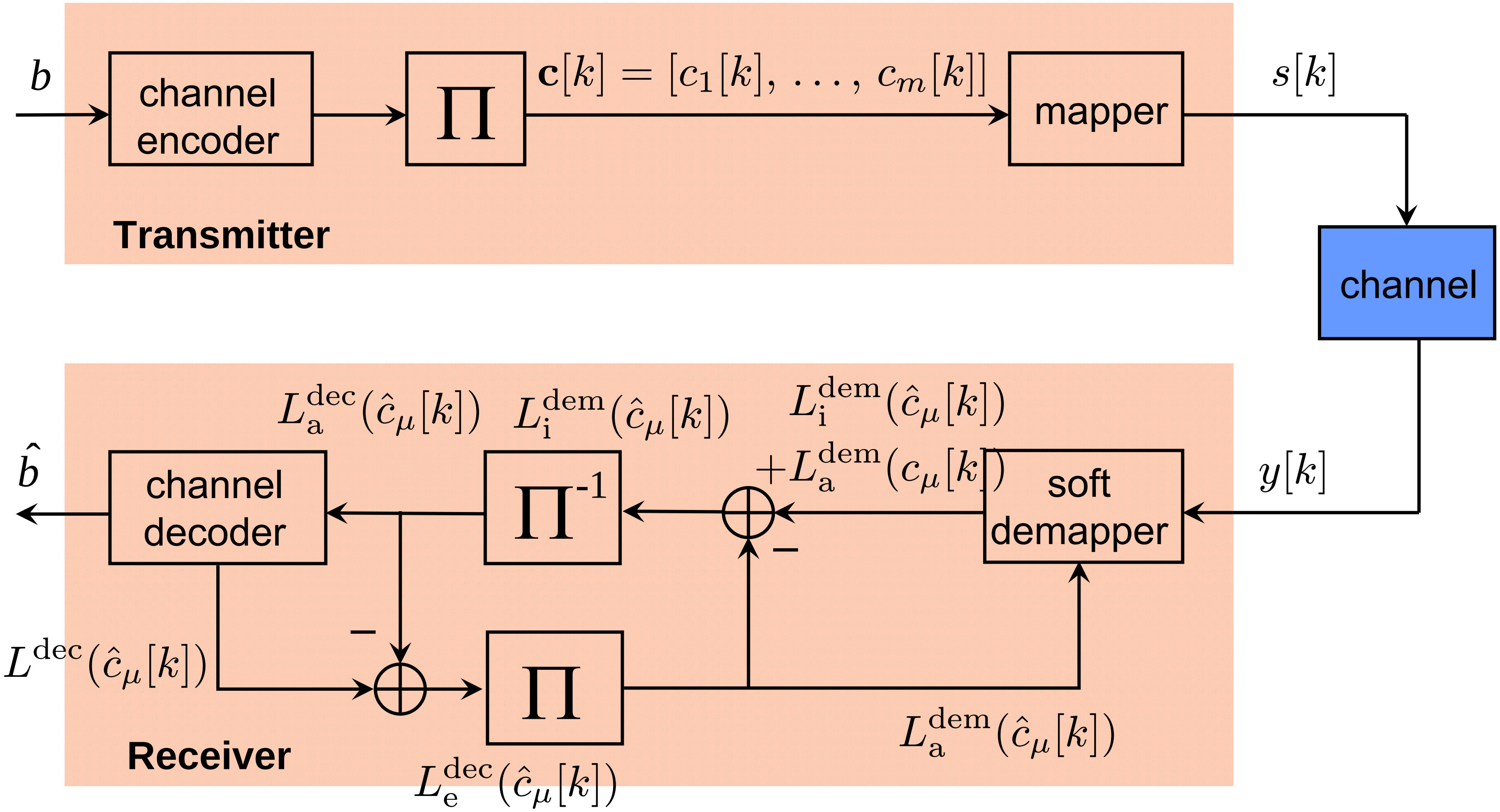
# Bitwise Mutual Information for Different Mappings

- Results for 8-PSK, AWGN channel, $E_b/N_0 = 6$ dB

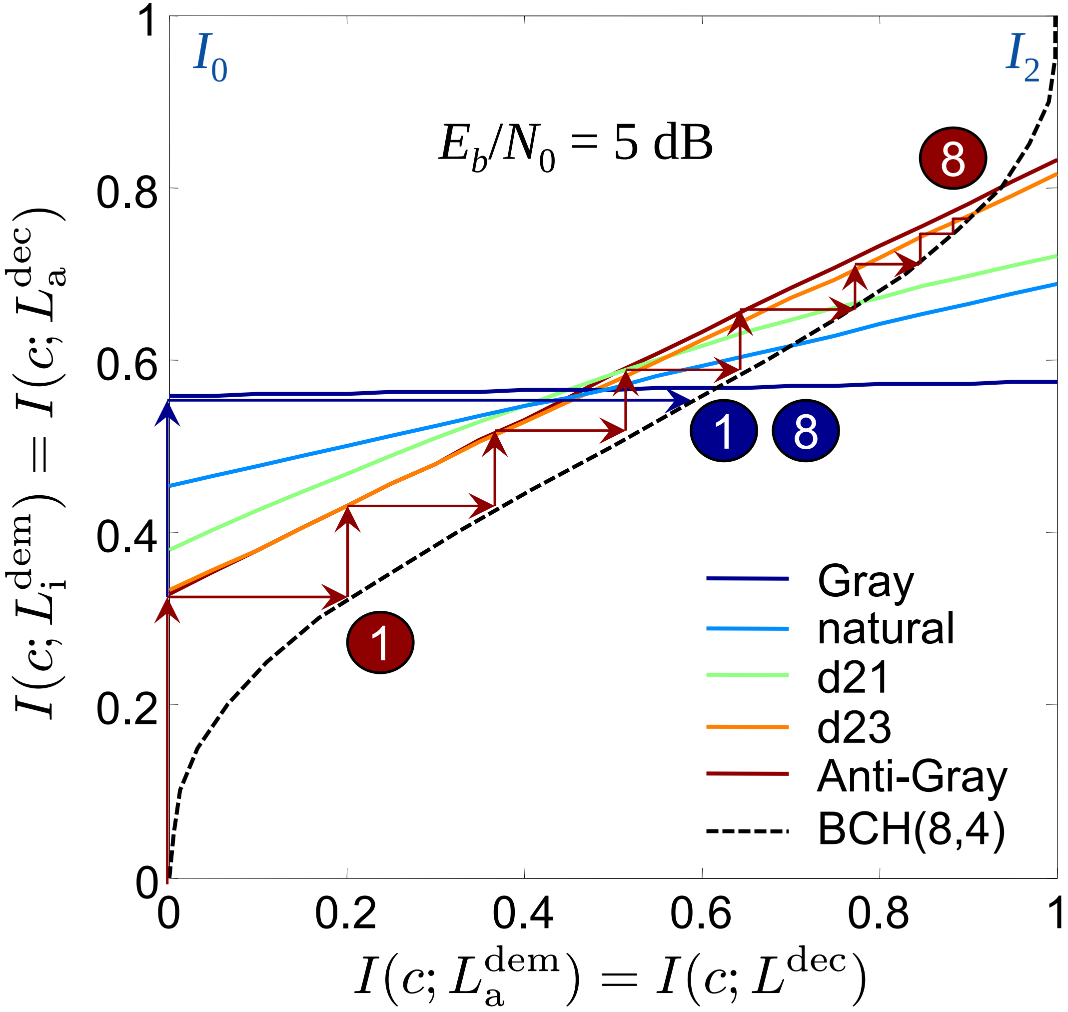
| **Mapping** | $I_0$ | $I_1$ | $I_2$ | $I(s;r) = \sum I_L$ |
|---|---|---|---|---|
| Gray | 0.7810 | 0.7821 | 0.7831 | 2.3461 |
| natural | 0.6373 | 0.8268 | 0.8820 | 2.3461 |
| d21 | 0.6327 | 0.7739 | 0.9395 | 2.3461 |
| d23 | 0.5385 | 0.8186 | 0.9890 | 2.3461 |
| Anti-Gray | 0.4938 | 0.8726 | 0.9797 | 2.3461 |

- Total mutual information is independent of mapping scheme
- Bit-specific mutual information highly depends on bit-mapping
  - Gray: nearly constant
  - d23, Anti-Gray: strong increase

# System Model for BICM

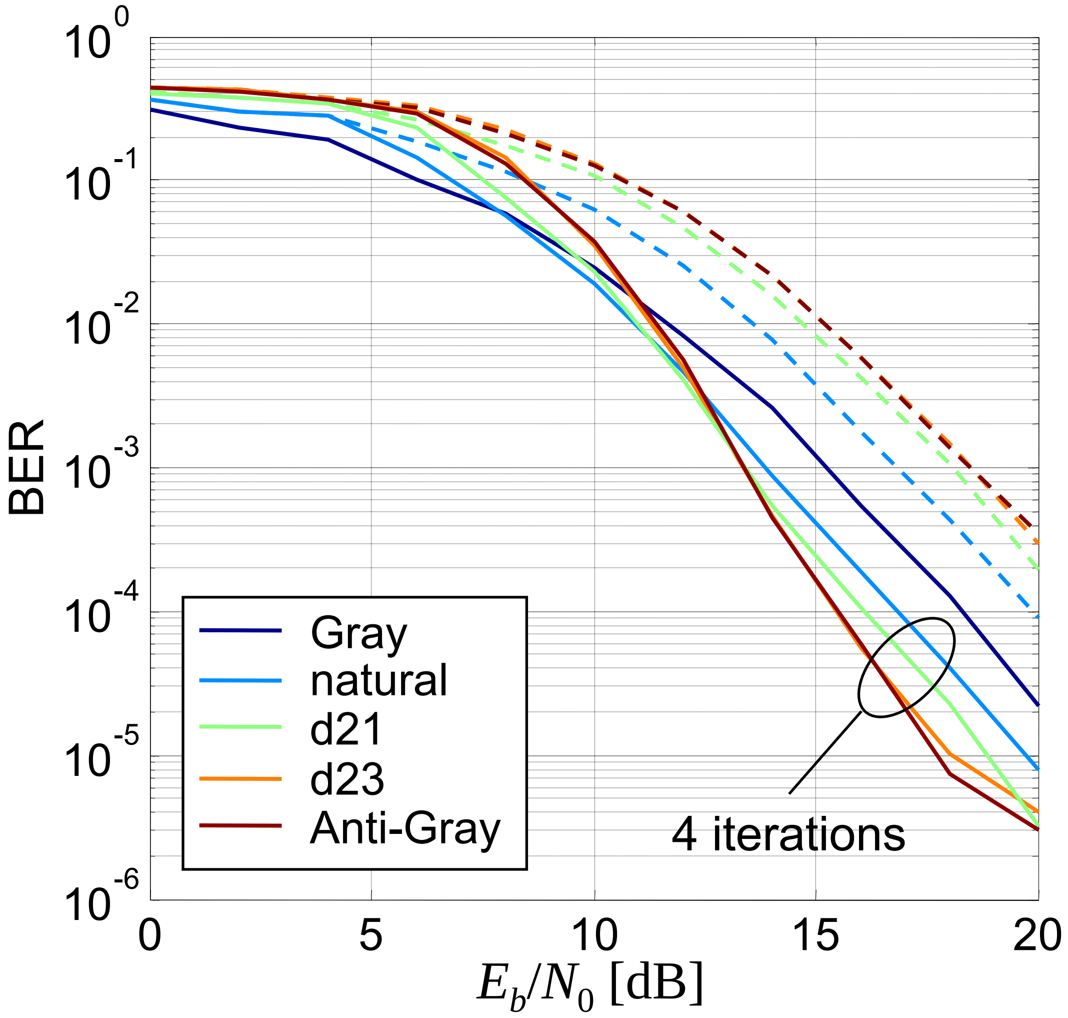


# EXtrinsic Information Transfer Charts



- Demapper: *a priori* information
  - so far discrete
    → 0, 1, 2 bits perfectly known
  - now continuous
    → mutual information $I(c; L_a^{\mathrm{dem}})$
- Result in principle as before
  - Boundary values $I_0$ and $I_2$
- Detection and decoding only once
  - Gray is best
- Iterative detection and decoding
  - Anti-Gray is best

# Bit Error Rates



- Simulation parameters
  - BCH(8,4)
  - 8-PSK
  - Alamouti scheme
  - 360 coded bits per frame
  - Independent Rayleigh fading
  - Channel const. for 24 symbols
- First detection and decoding
  - Gray good, Anti-Gray bad
- After four iterations
  - Anti-Gray is best
- Same results as predicted by EXIT charts