Turbo Source Coding

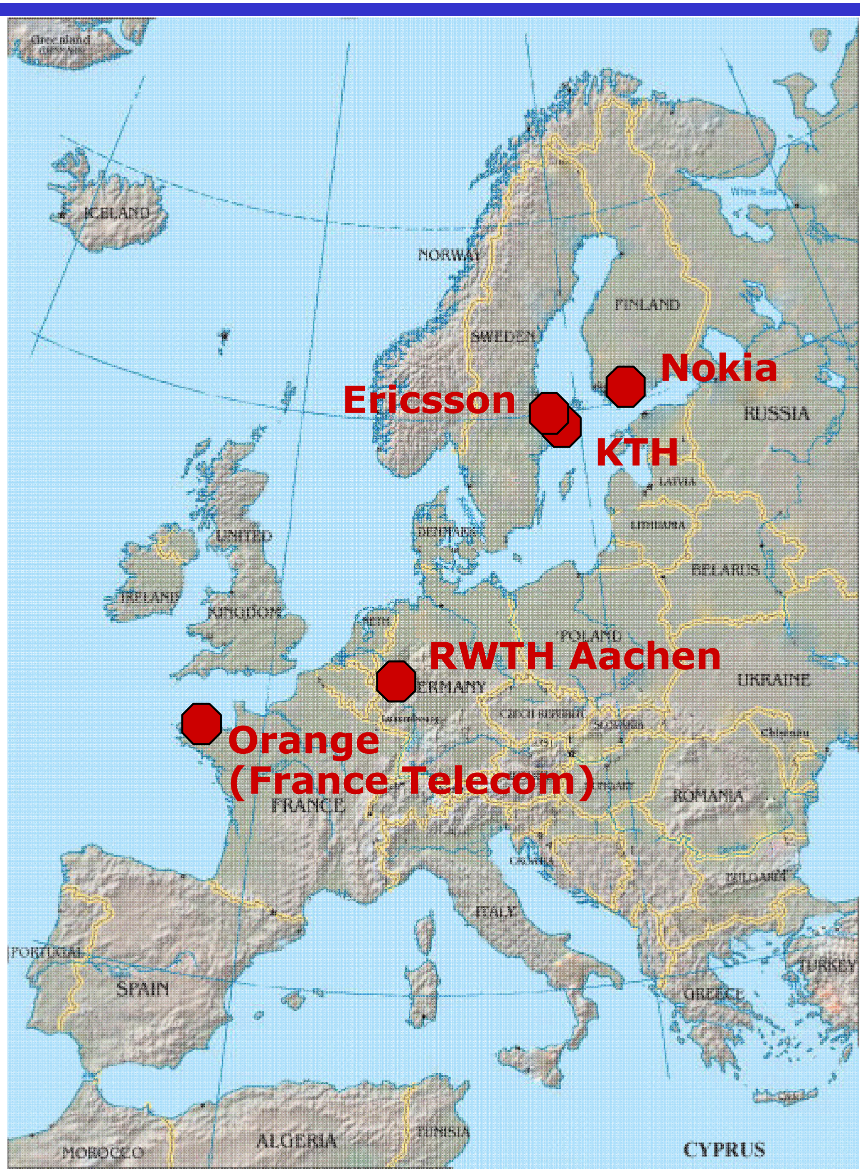
Laurent Schmalen and Peter Vary

FlexCode Public Seminar
June 16, 2008

Outline

- FlexCode in a nutshell
- Entropy coding
- Turbo coding and decoding
- Application of Turbo codes as source codes
- A joint-source channel coding scheme with iterative decoding for compression
- Possible Application to FlexCode

Who?

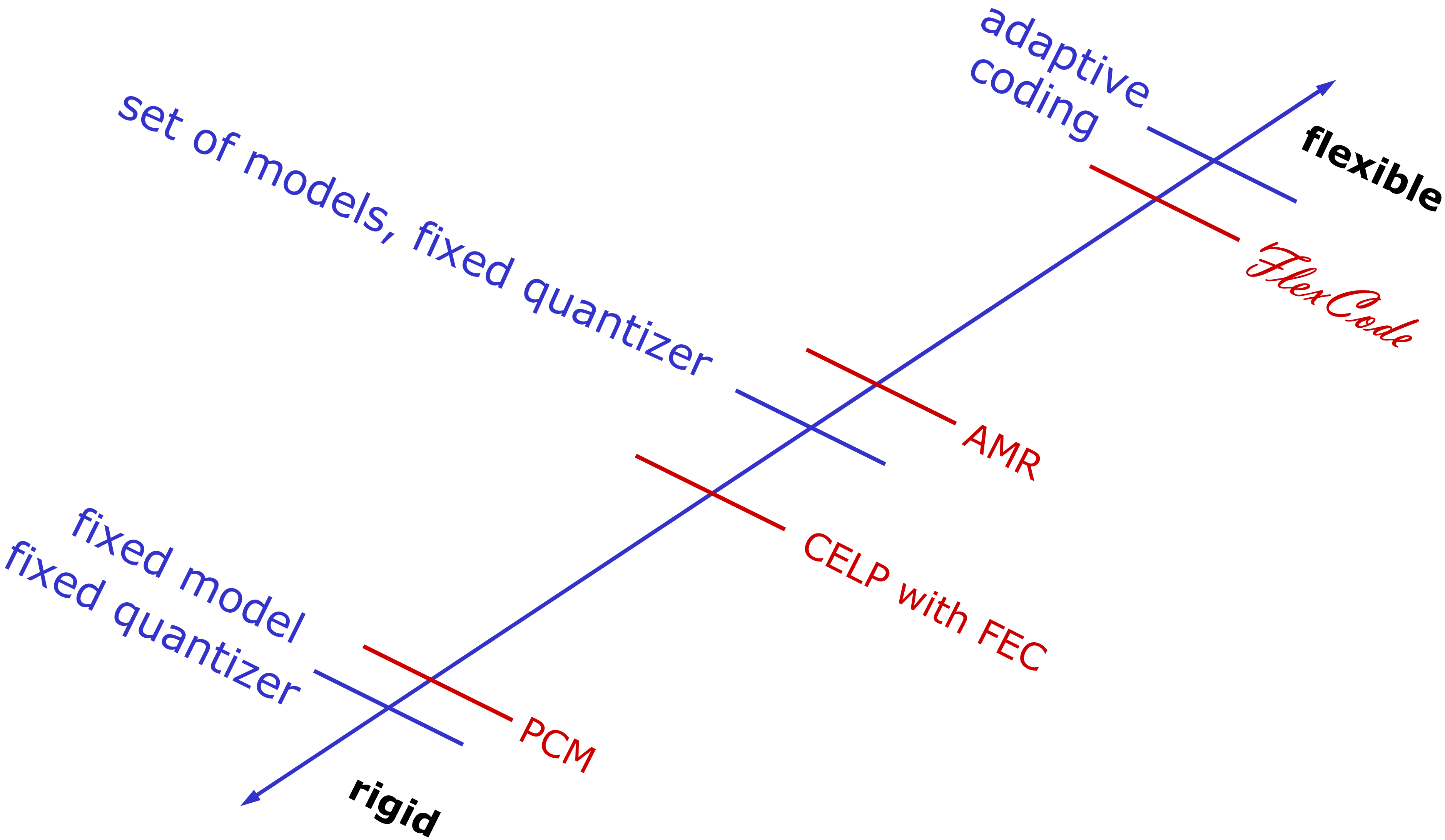

Nokia
Ericsson
KTH
RWTH Aachen
Orange
(France Telecom)
ICELAND
NORWAY
SWEDEN
FINLAND
RUSSIA
LATVIA
LITHUANIA
DENMARK
BELARUS
UNITED
KINGDOM
IRELAND
POLAND
UKRAINE
GERMANY
FRANCE
ROMANIA
ITALY
SPAIN
PORTUGAL
GREECE
TURKEY
ALGERIA
MOROCCO
TUNISIA
CYPRUS

The Problem

- Heterogeneity of networks increasing
- Networks inherently variable (mobile users)

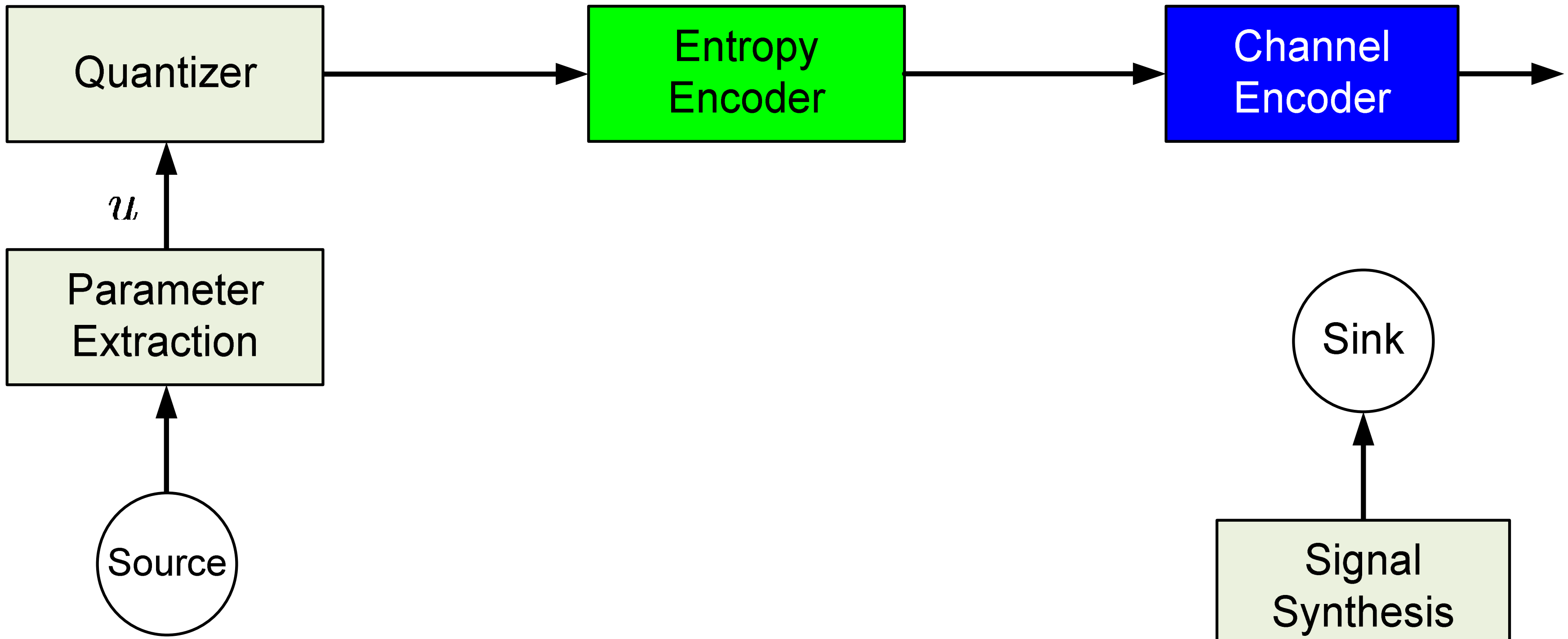
- But:
  – Coders not designed for specific environment
  – Coders inflexible (codebooks and FEC)
  – Feedback channel underutilized

Adaptation and Coding

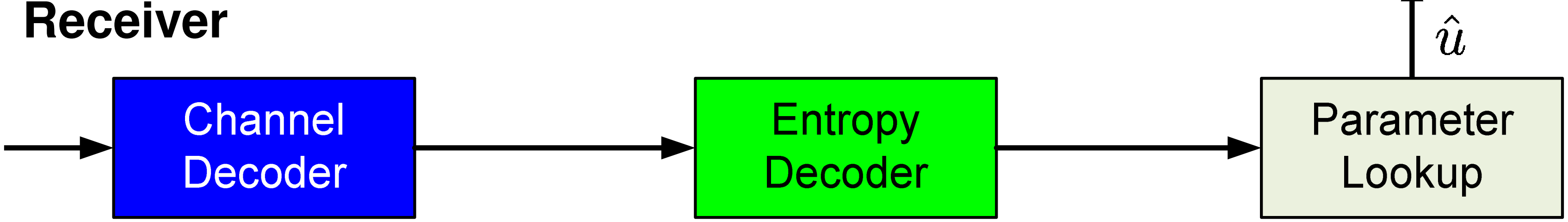


Conventional Transmission

- **Transmitter**



- **Receiver**



Lossless Source Coding

- Prominent examples of lossless entropy coders
  - Huffman coding
  - Lempel-Ziv coding
  - Arithmetic coding

- Example: Huffman code
  - AACABA → 0 0 110 0 10 0
  - No bit pattern is prefix of another
  - Unambiguous decoding

Symbol S	P(S)	Bit pattern
A	0.5	0
B	0.3	10
C	0.15	110
D	0.05	111

FlexCode

Lossless Source Coding

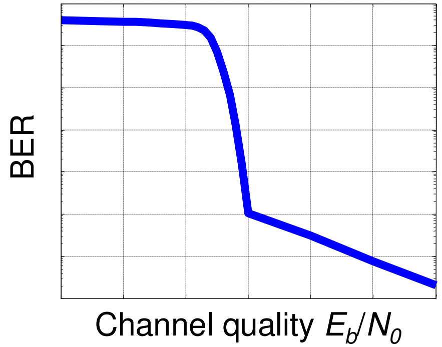
RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN RWTH AACHEN UNIVERSITY

- Usability of lossless source codes
- High sensitivity against transmission errors
  - Huffman code: synchronization loss
  - Arithmetic code: selection of wrong interval, complete decoding failure

FlexCode

Lossless Source Coding

- Usability of lossless source codes
- High sensitivity against transmission errors
  - Huffman code: synchronization loss
  - Arithmetic code: selection of wrong interval, complete decoding failure
- Very strong channel codes required
  - Error floor, i.e., seldom bit errors leading to decoding failures



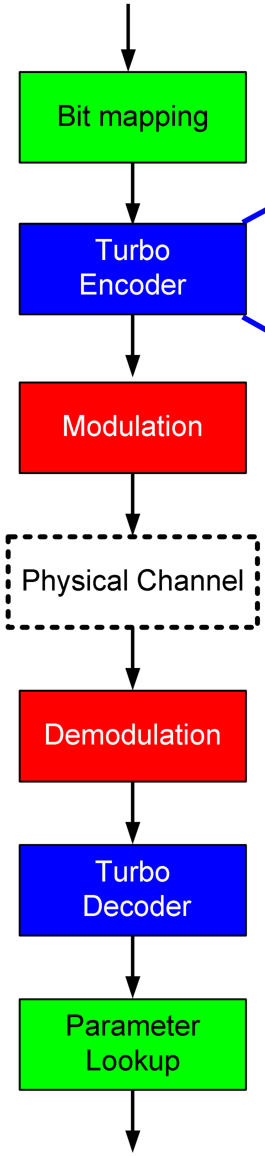
Lossless Source Coding

- Usability of lossless source codes
- High sensitivity against transmission errors
  - Huffman code: synchronization loss
  - Arithmetic code: selection of wrong interval, complete decoding failure
- Very strong channel codes required
  - Error floor, i.e., seldom bit errors leading to decoding failures
- Iterative source-channel decoding schemes for Entropy Codes
  - High complexity
  - Difficult to apply to arithmetic coding **[Guionnet 04]**

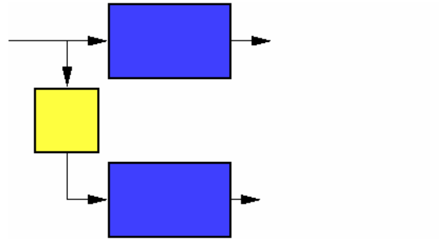
Lossless Source Coding

- Wanted:
  - A flexible compression scheme (entropy coder) which
    - Has similar performance as known compression schemes
    - Is robust against transmission errors
    - Can instantaneously adapt to varying channel conditions by exchanging compression ratio against error robustness
  - Analogy between channel codes and source codes
    - A good channel code is often also a good source code
    - Use of LDPC codes for compression **[Caire 03]**
  - Can Turbo codes be used for compression?

Turbo Codes, Concept



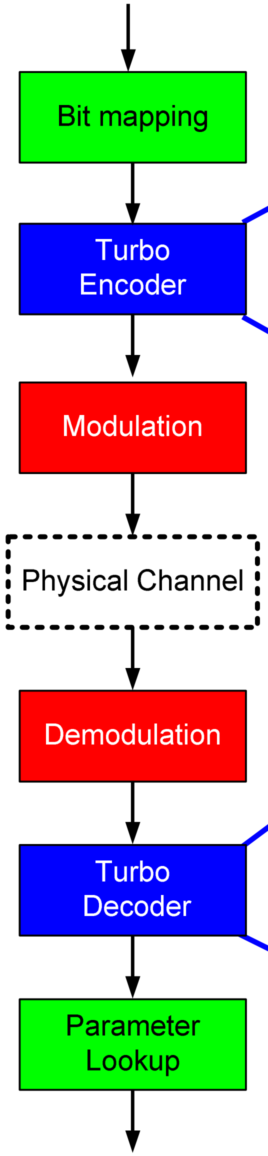
Concatenated Encoding



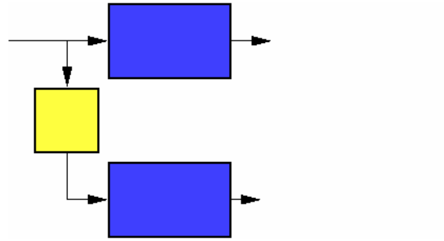
parallel scheme

[Berrou 93]

Turbo Codes, Concept



Concatenated Encoding



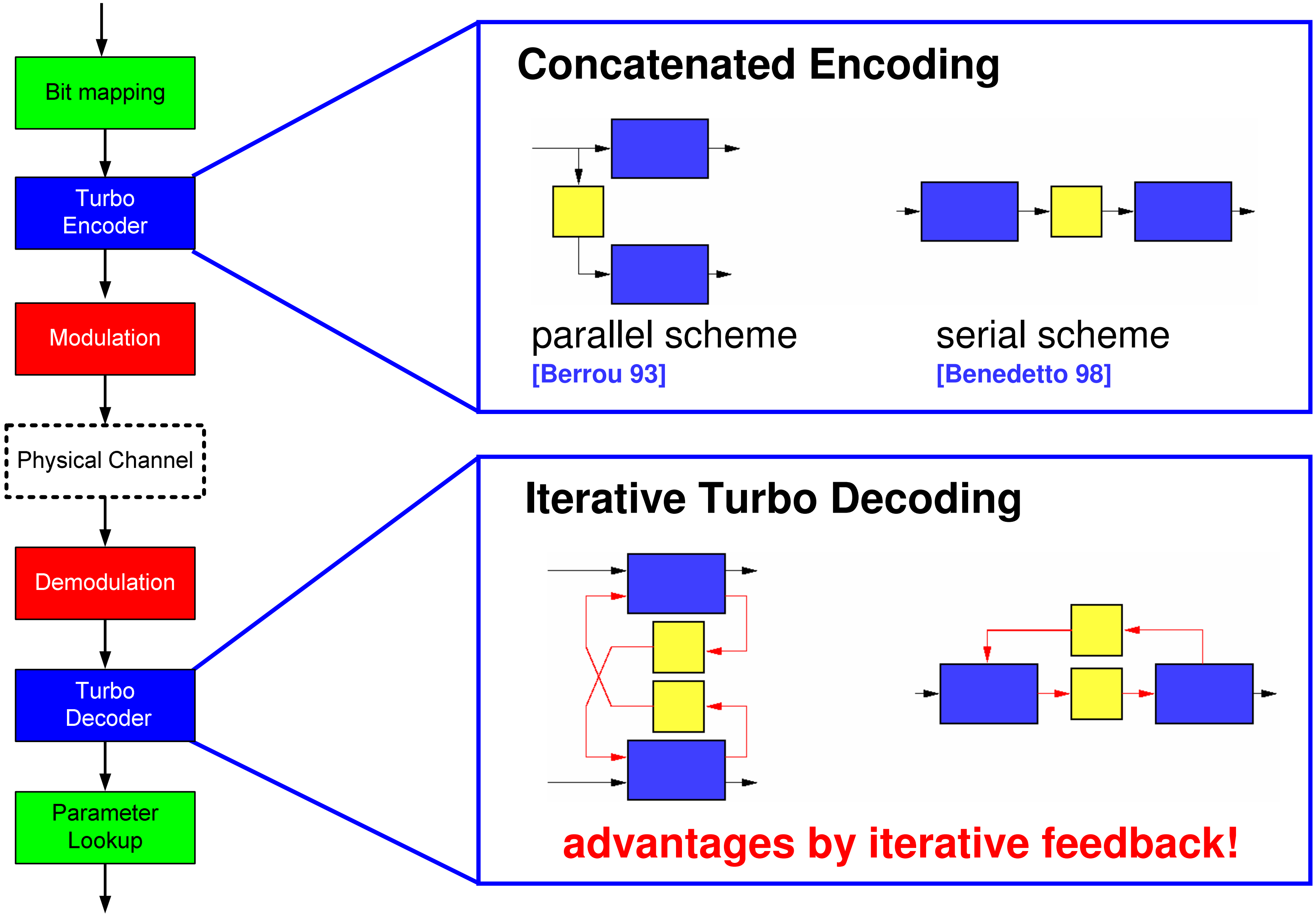
parallel scheme

[Berrou 93]

Iterative Turbo Decoding

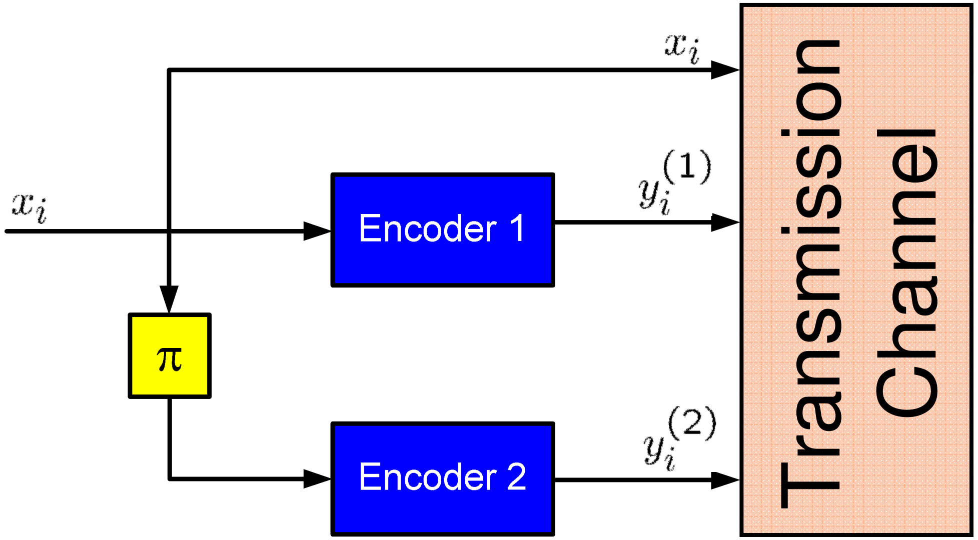
advantages by iterative feedback!

Turbo Codes, Concept



Turbo Coding & Decoding

- Turbo Code Encoder and Decoder **[Berrou 93]**

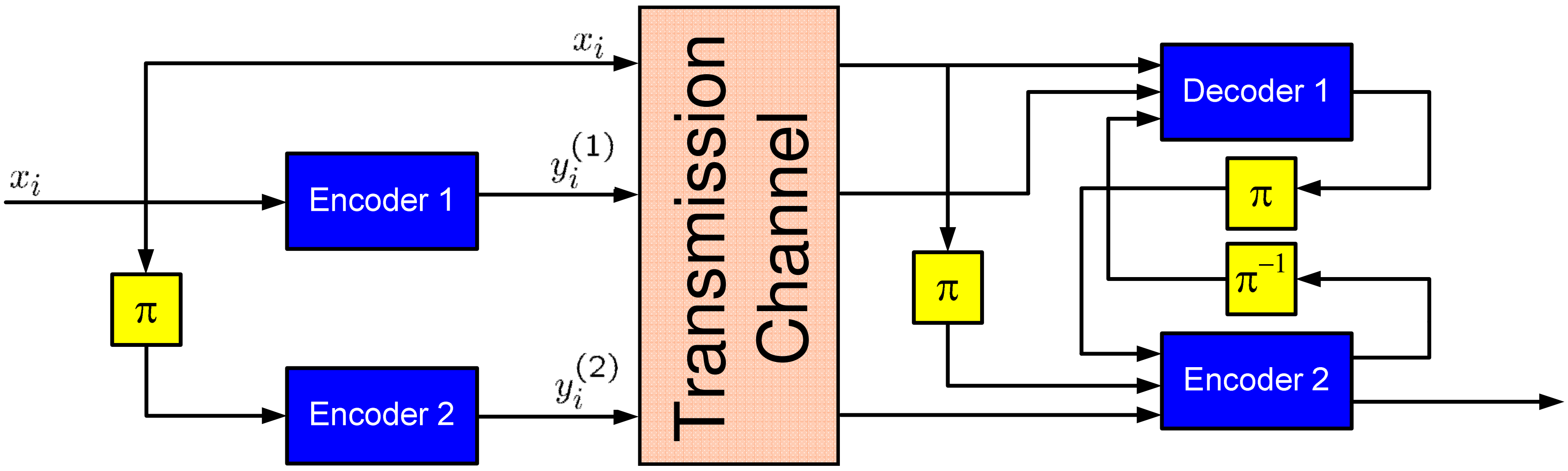


π denotes an interleaver

Encoder x is an (almost) arbitrary channel encoder

Turbo Coding & Decoding

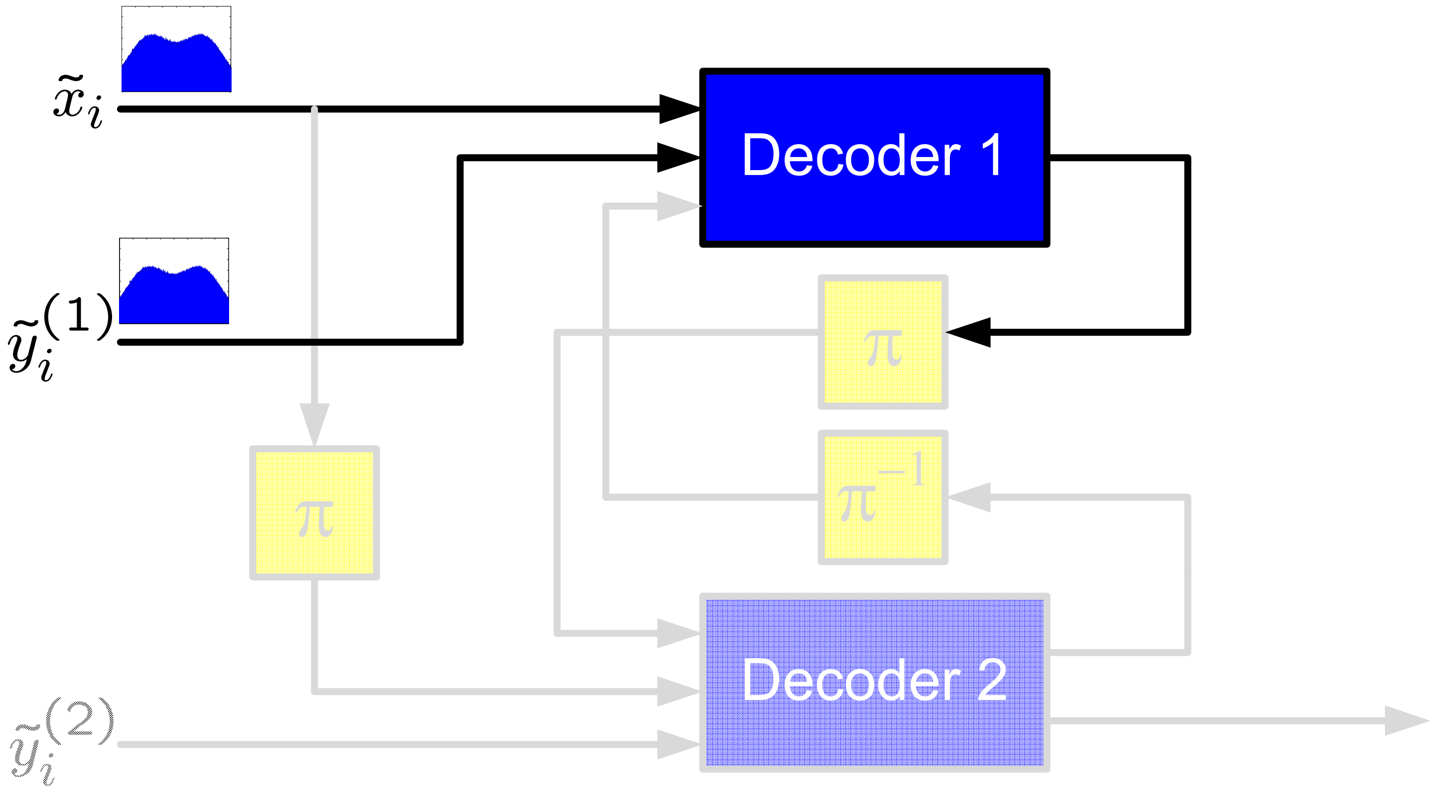
- Turbo Code Encoder and Decoder **[Berrou 93]**



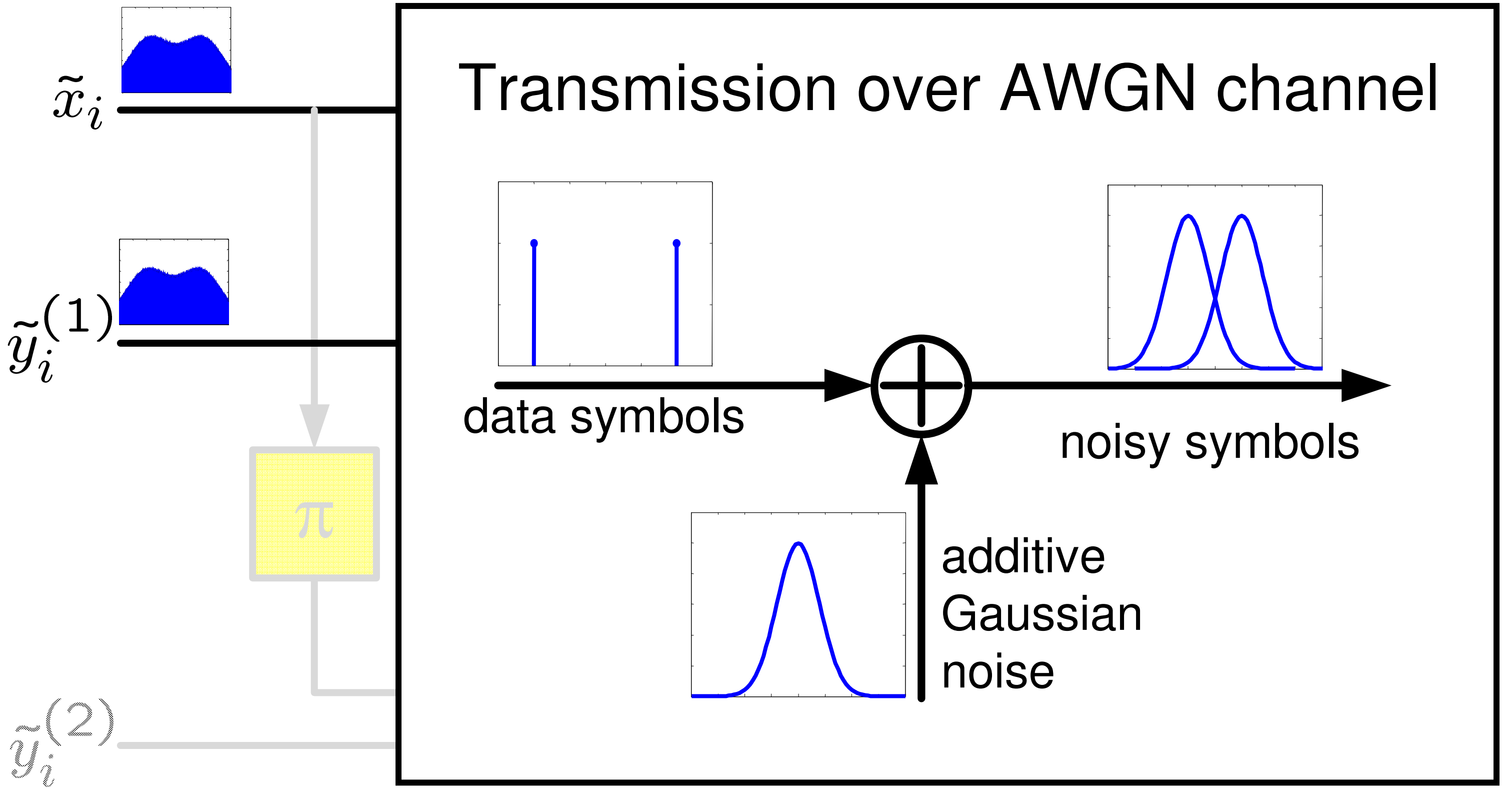
π denotes an interleaver

Encoder x is an (almost) arbitrary channel encoder

- Turbo Decoding (1st iteration, 1st step)

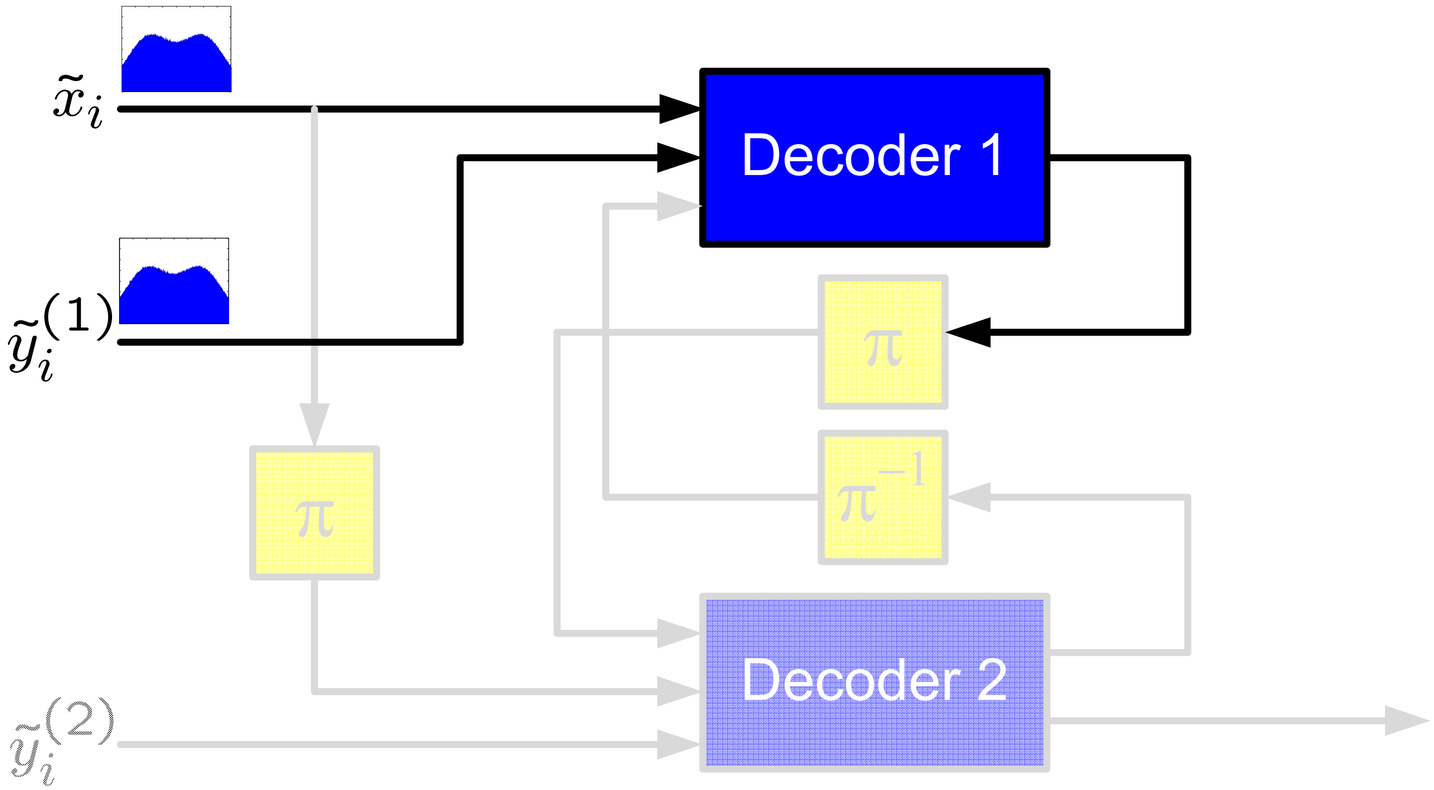


- Turbo Decoding (1st iteration, 1st step)



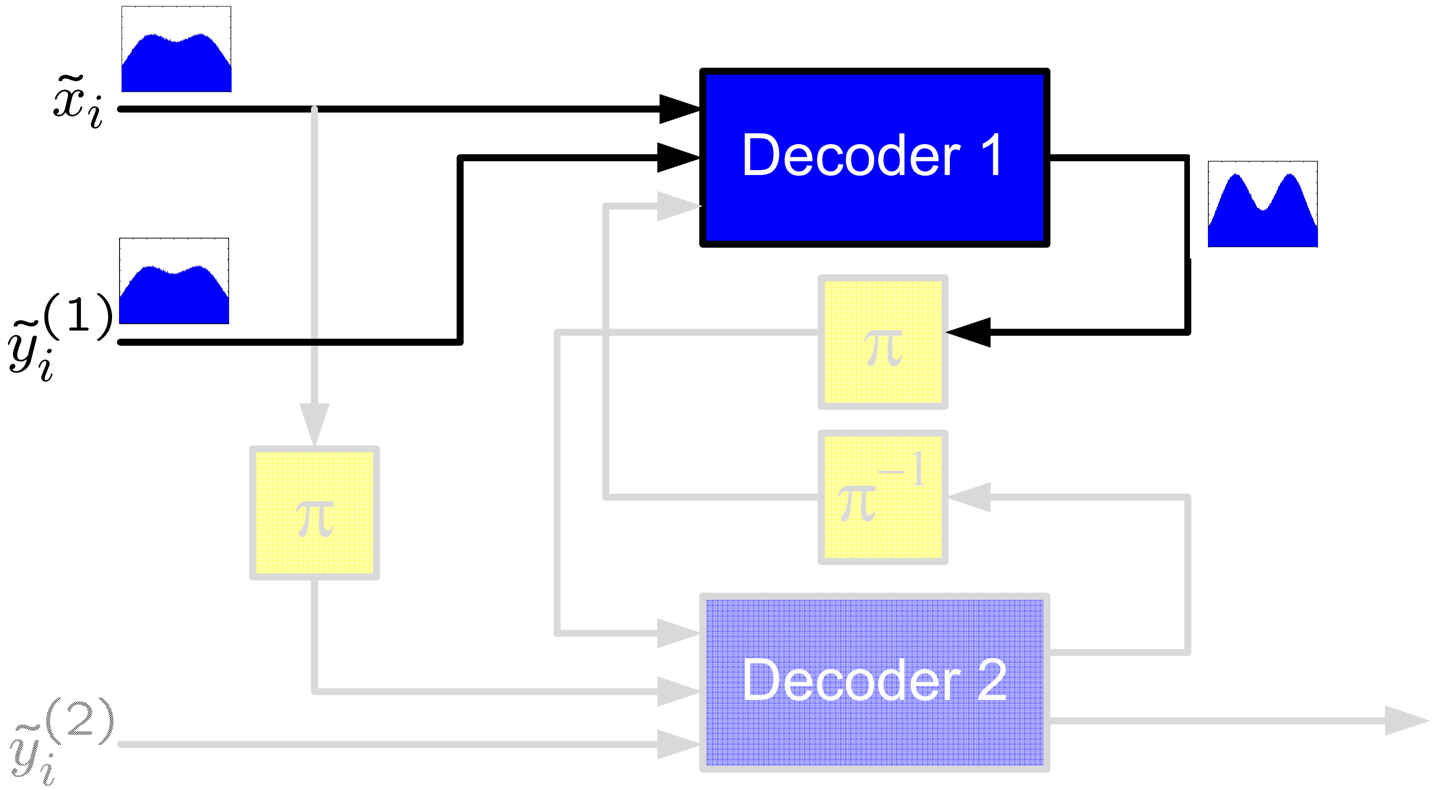
Turbo Coding & Decoding

- Turbo Decoding (1st iteration, 1st step)



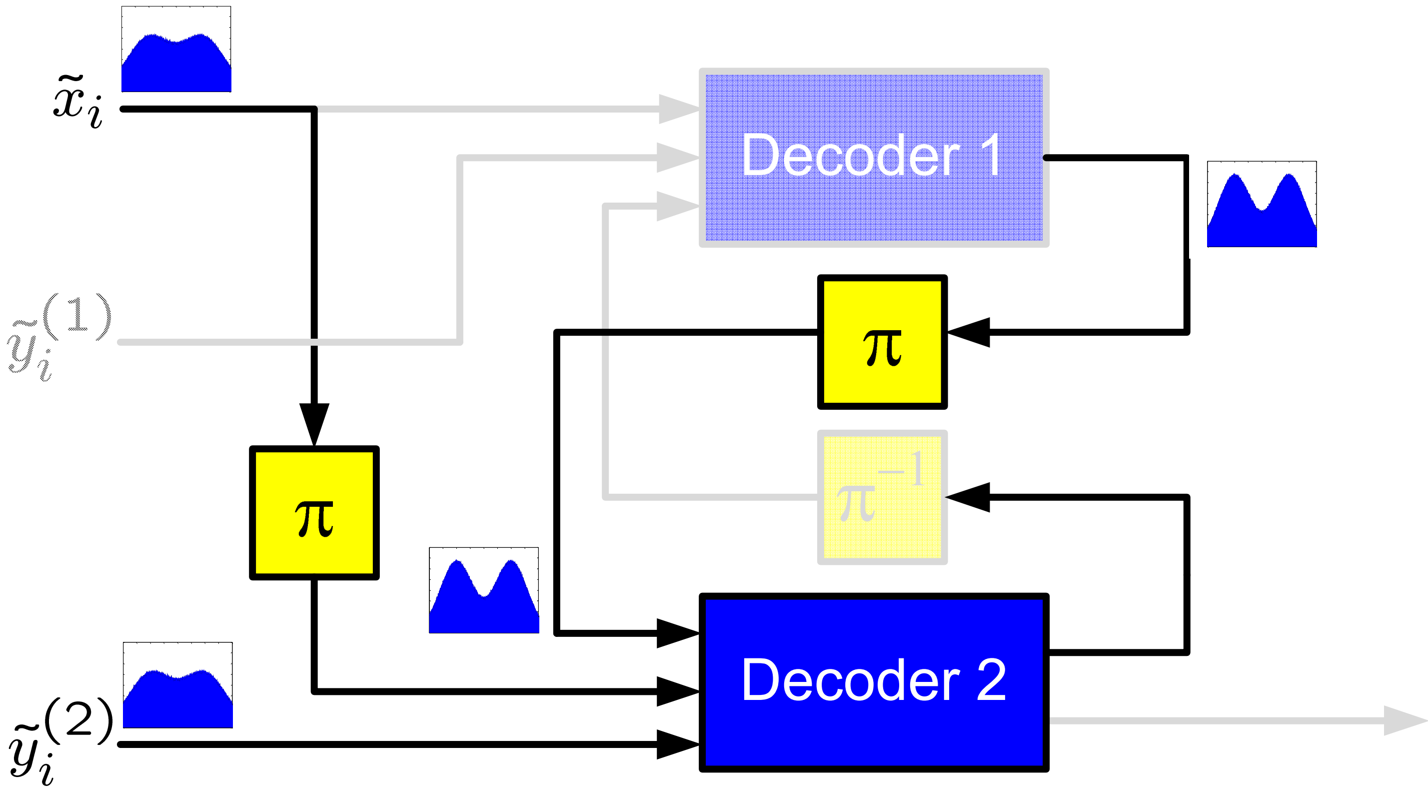
Turbo Coding & Decoding

- Turbo Decoding (1st iteration, 1st step)



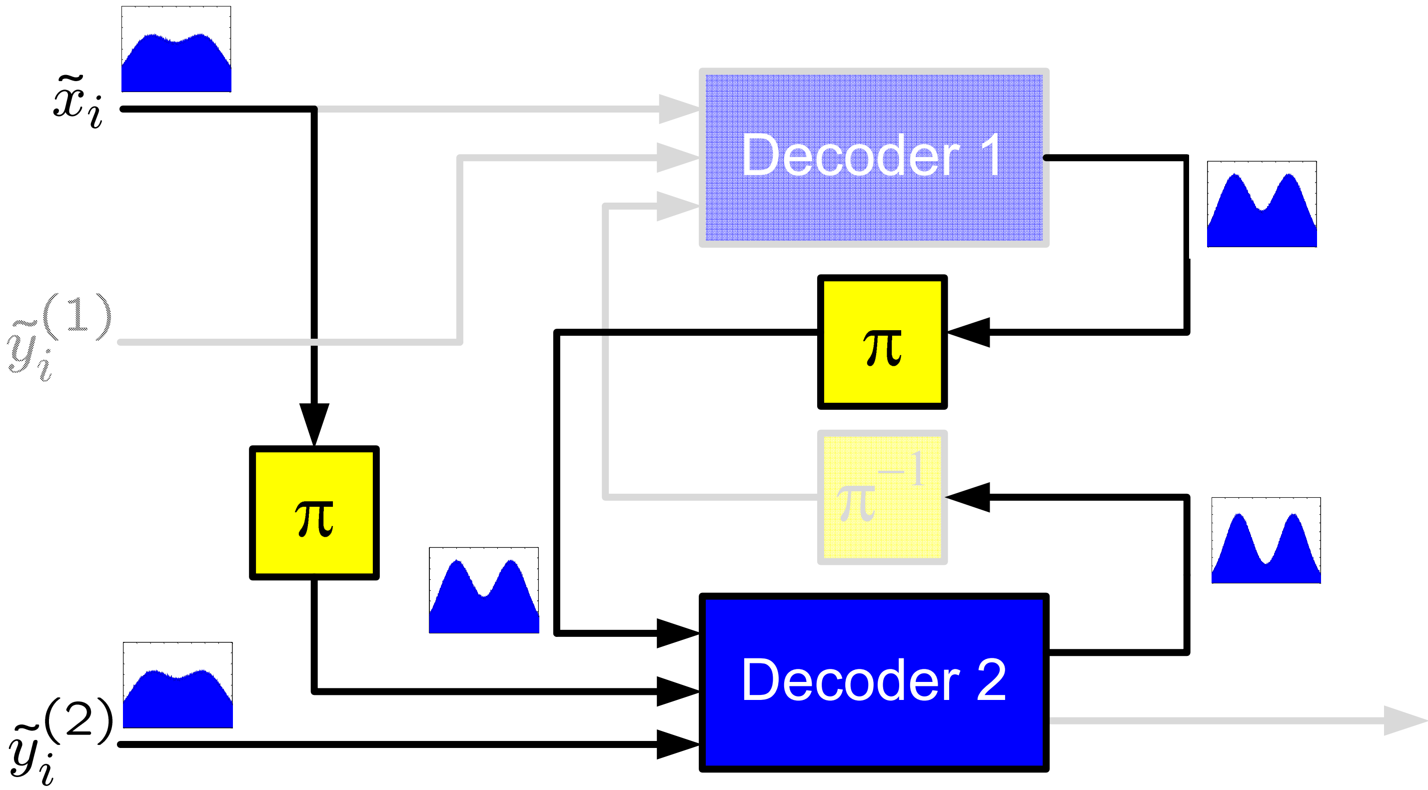
Turbo Coding & Decoding

- Turbo Decoding (1st iteration, 2nd step)



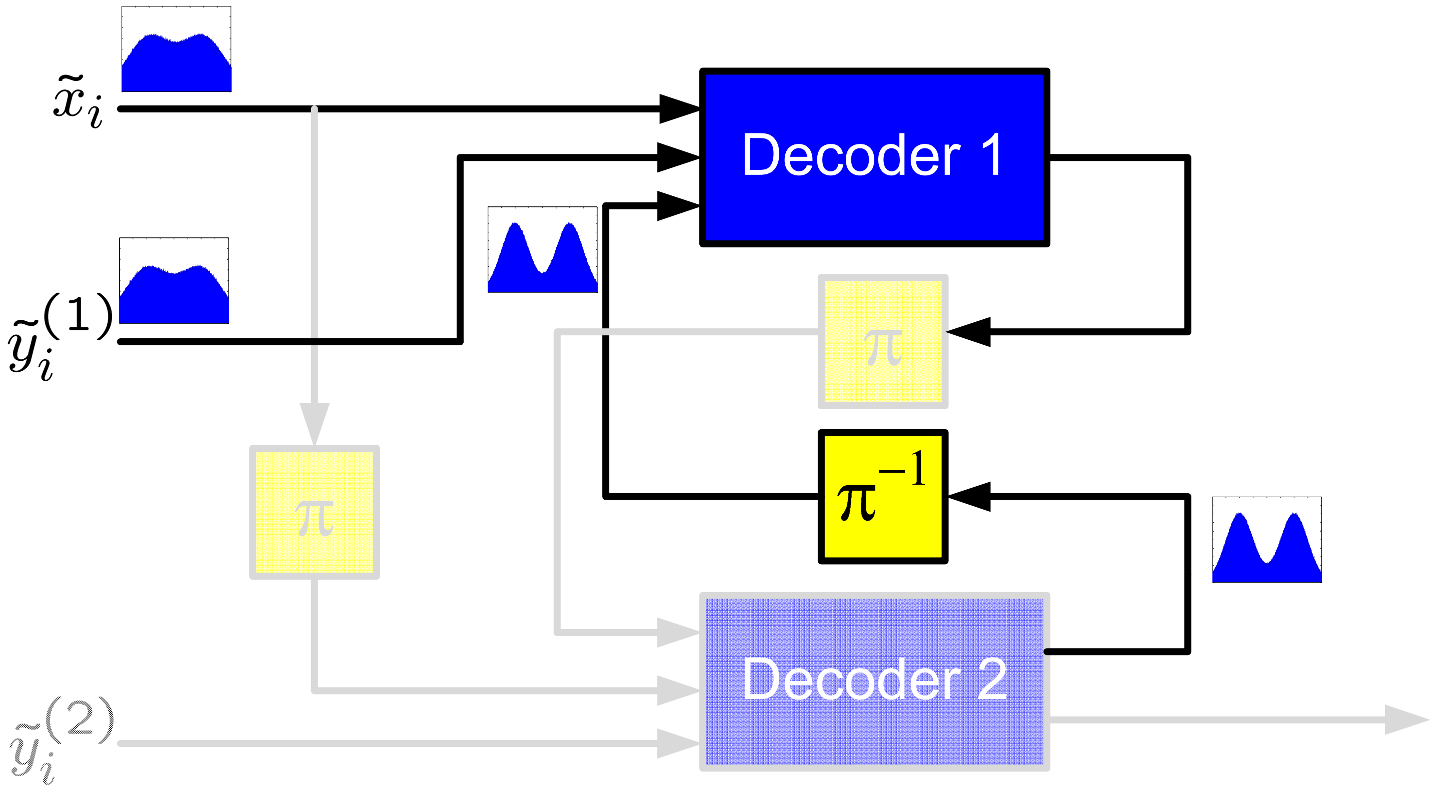
Turbo Coding & Decoding

- Turbo Decoding (1st iteration, 2nd step)



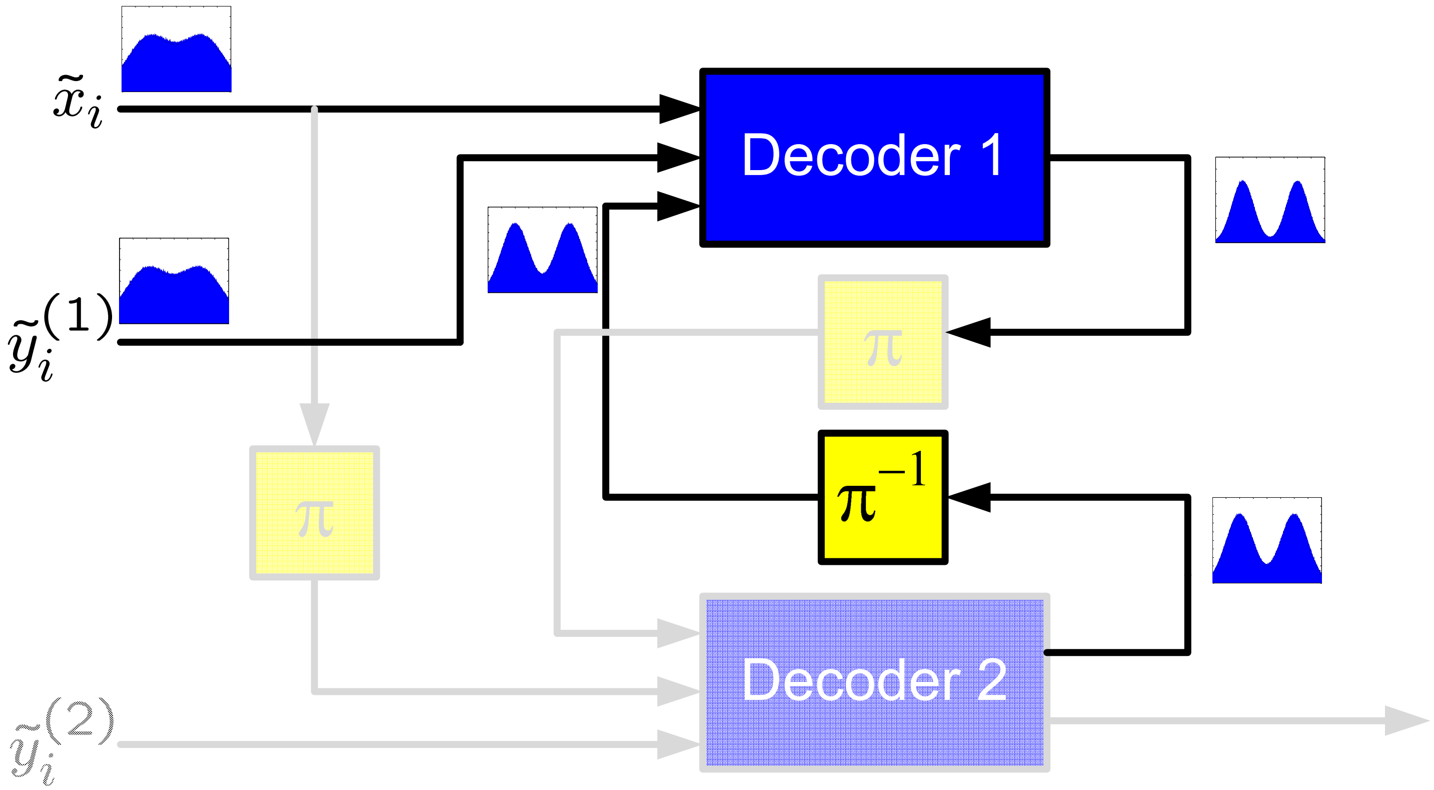
Turbo Coding & Decoding

- Turbo Decoding (2nd iteration, 1st step)



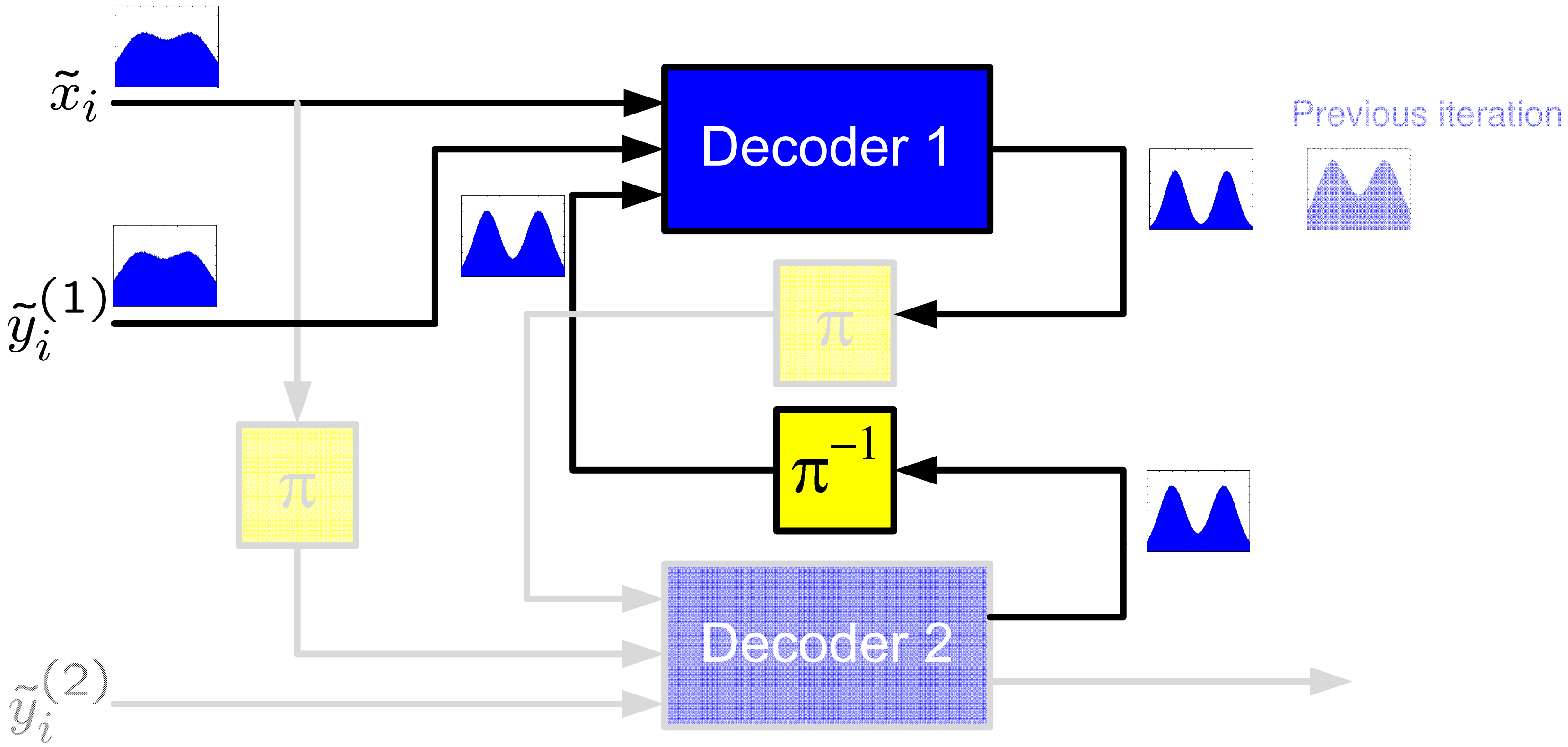
Turbo Coding & Decoding

- Turbo Decoding (2nd iteration, 1st step)



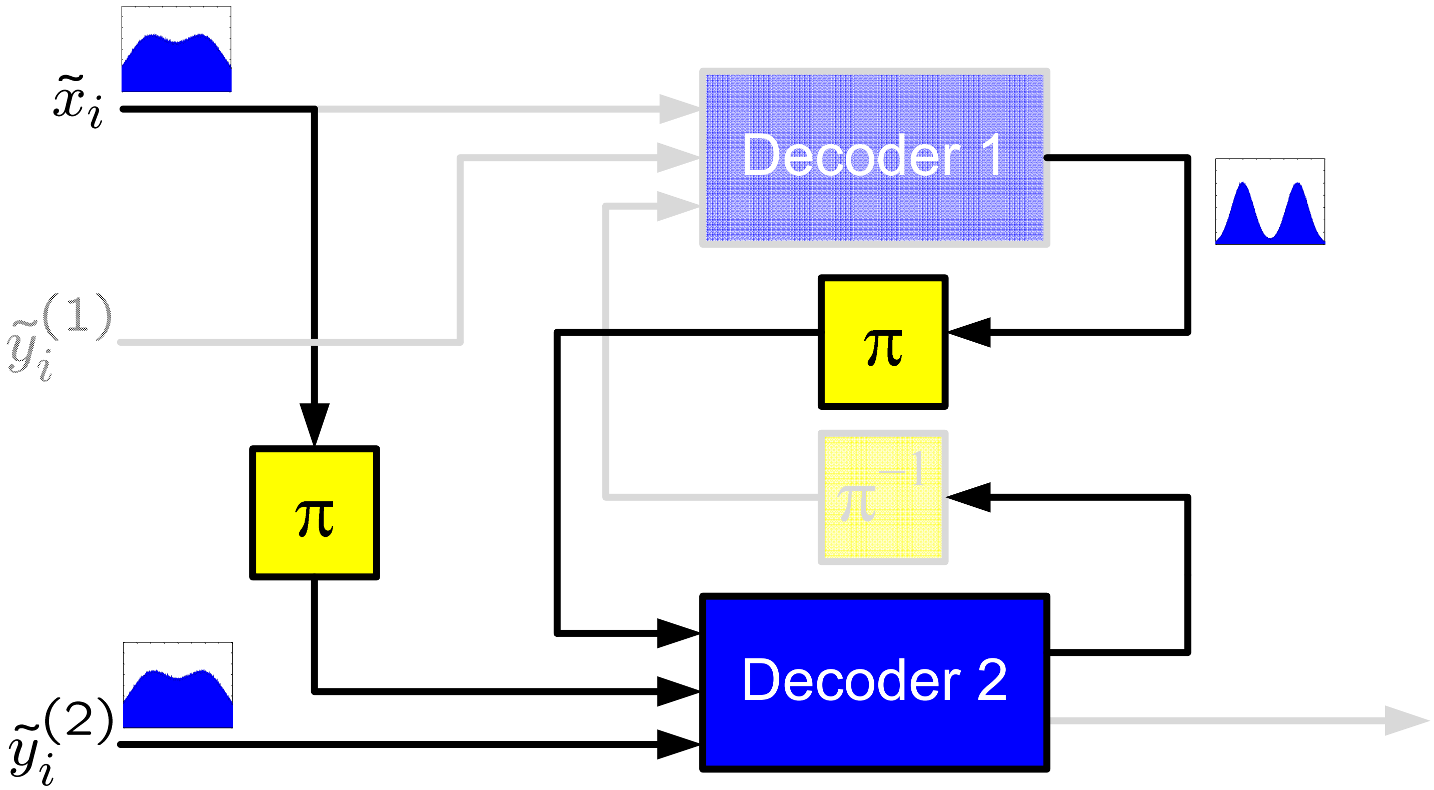
Turbo Coding & Decoding

- Turbo Decoding (2nd iteration, 1st step)



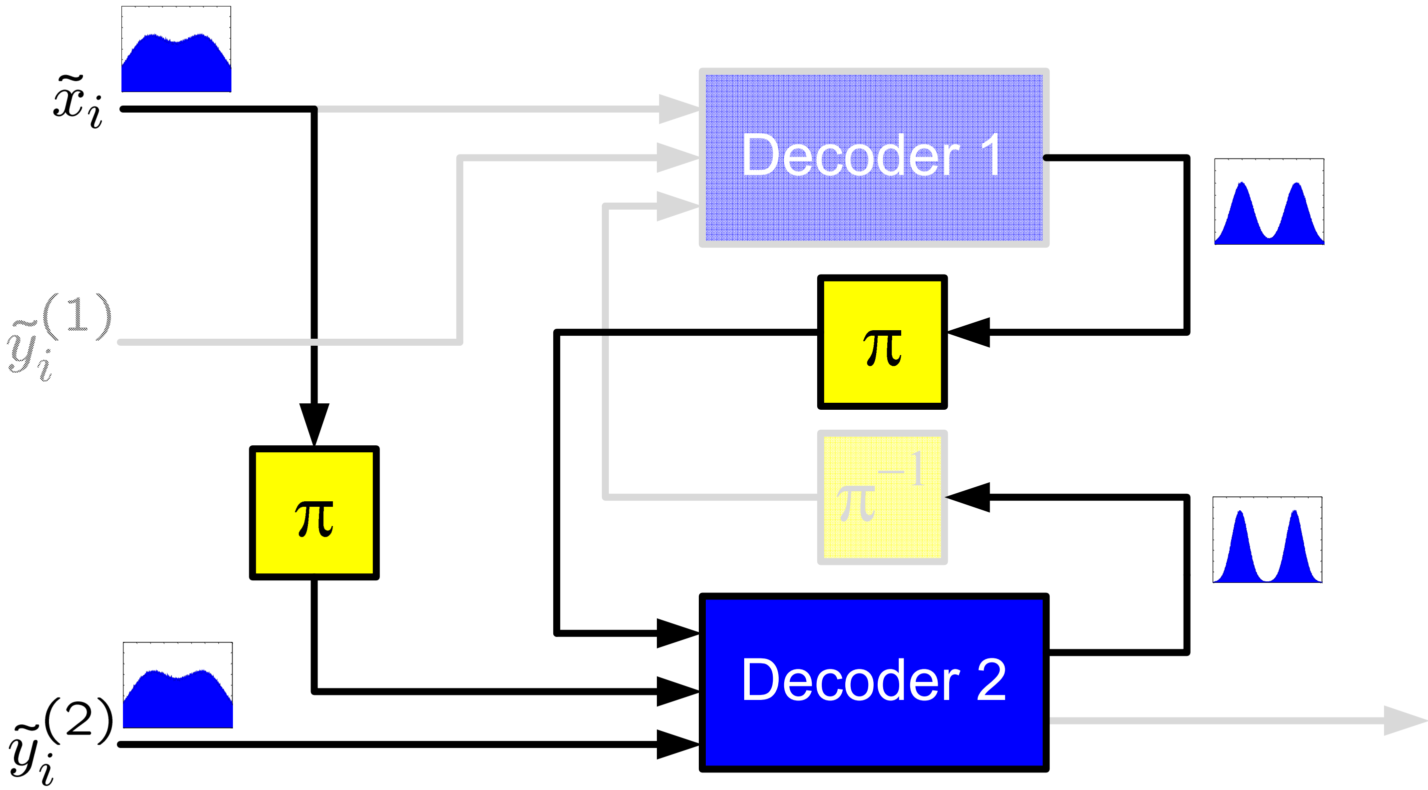
Turbo Coding & Decoding

- Turbo Decoding (2nd iteration, 2nd step)

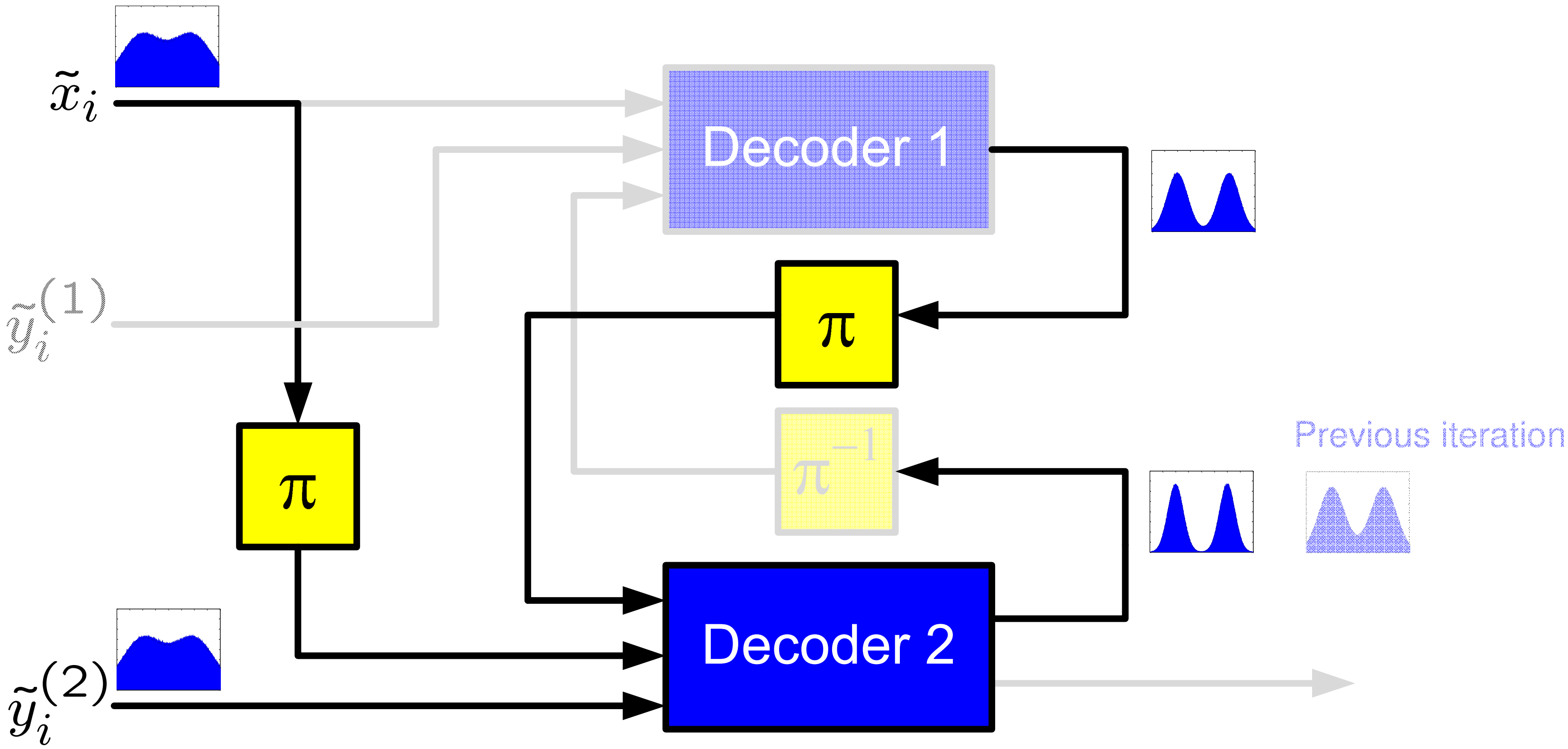


Turbo Coding & Decoding

- Turbo Decoding (2nd iteration, 2nd step)

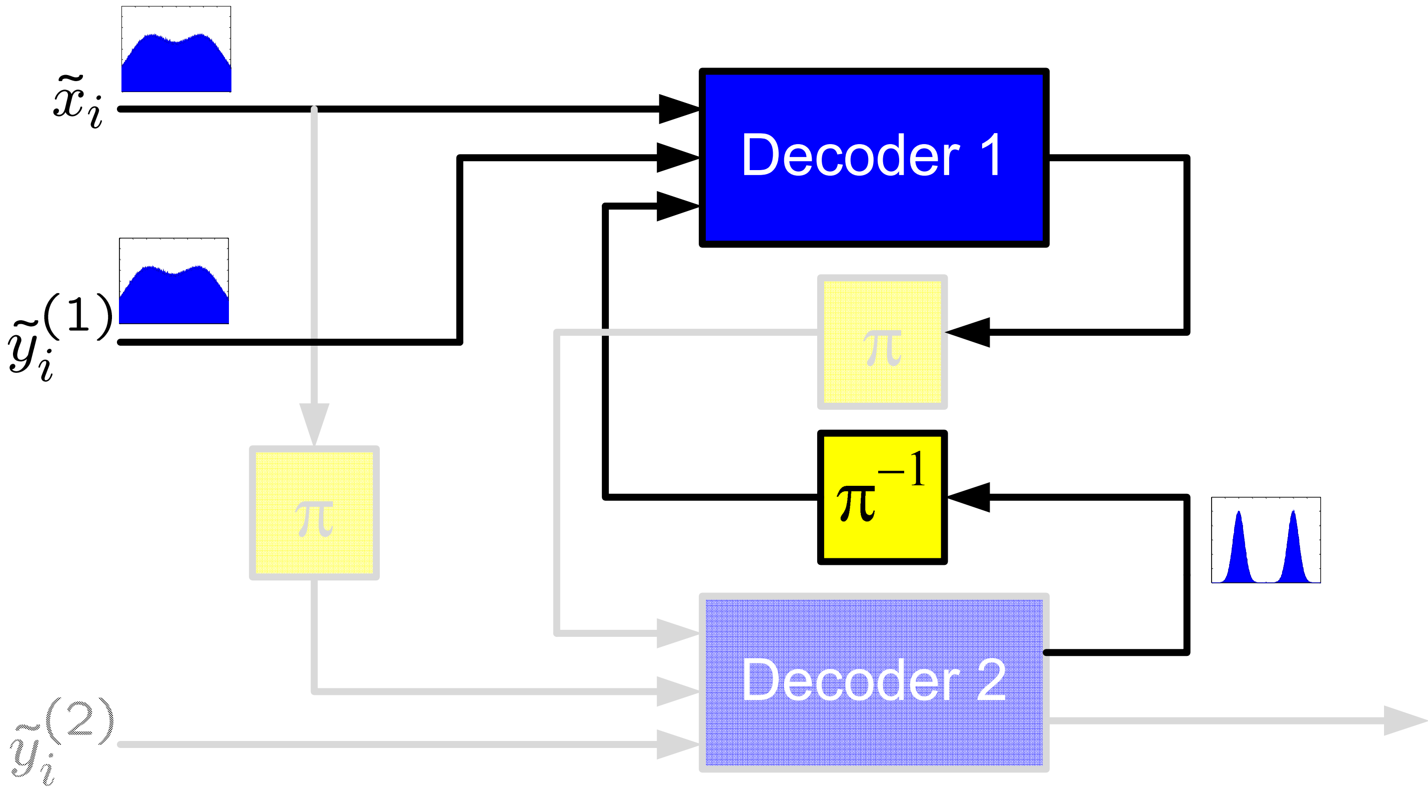


- Turbo Decoding (2nd iteration, 2nd step)



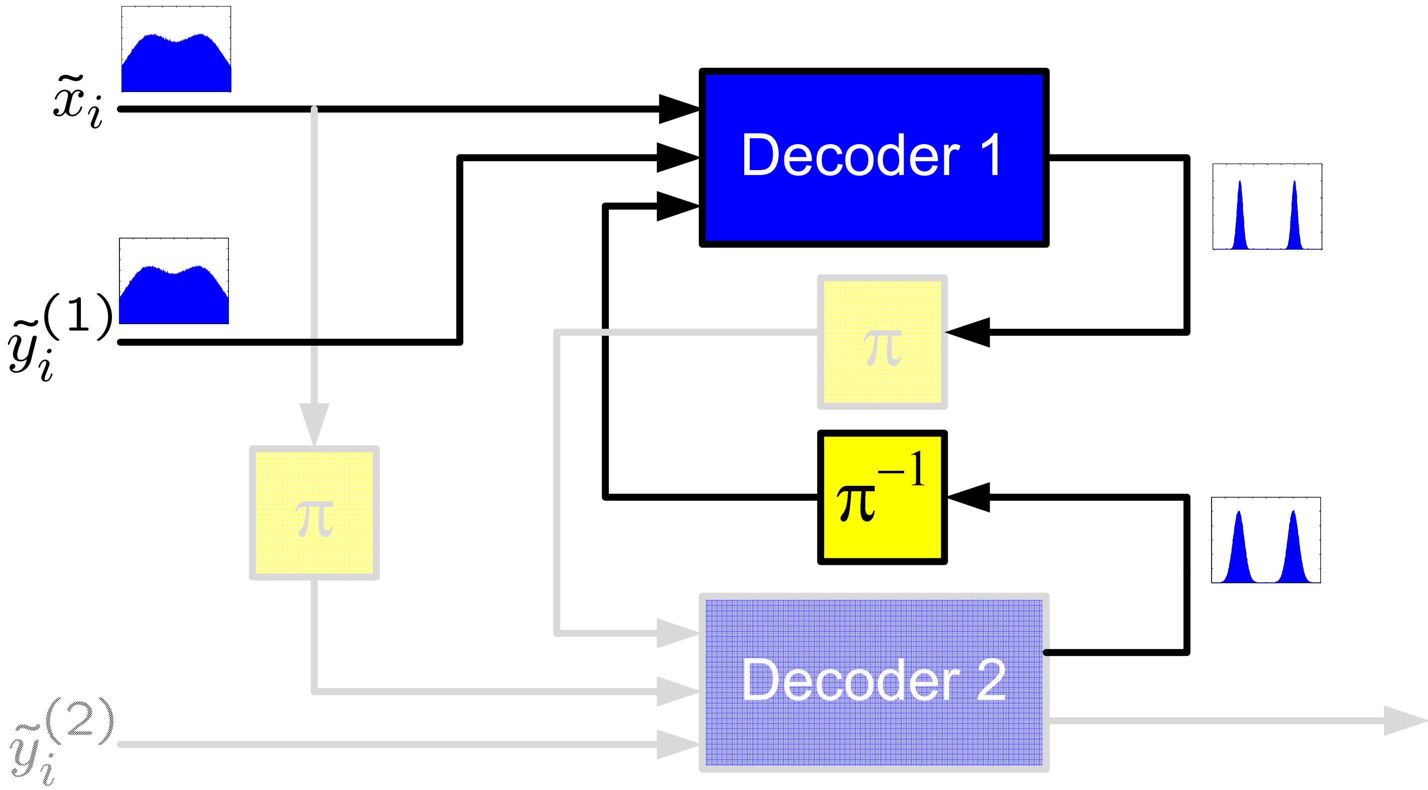
Turbo Coding & Decoding

- Turbo Decoding (n'th iteration, 1st step)



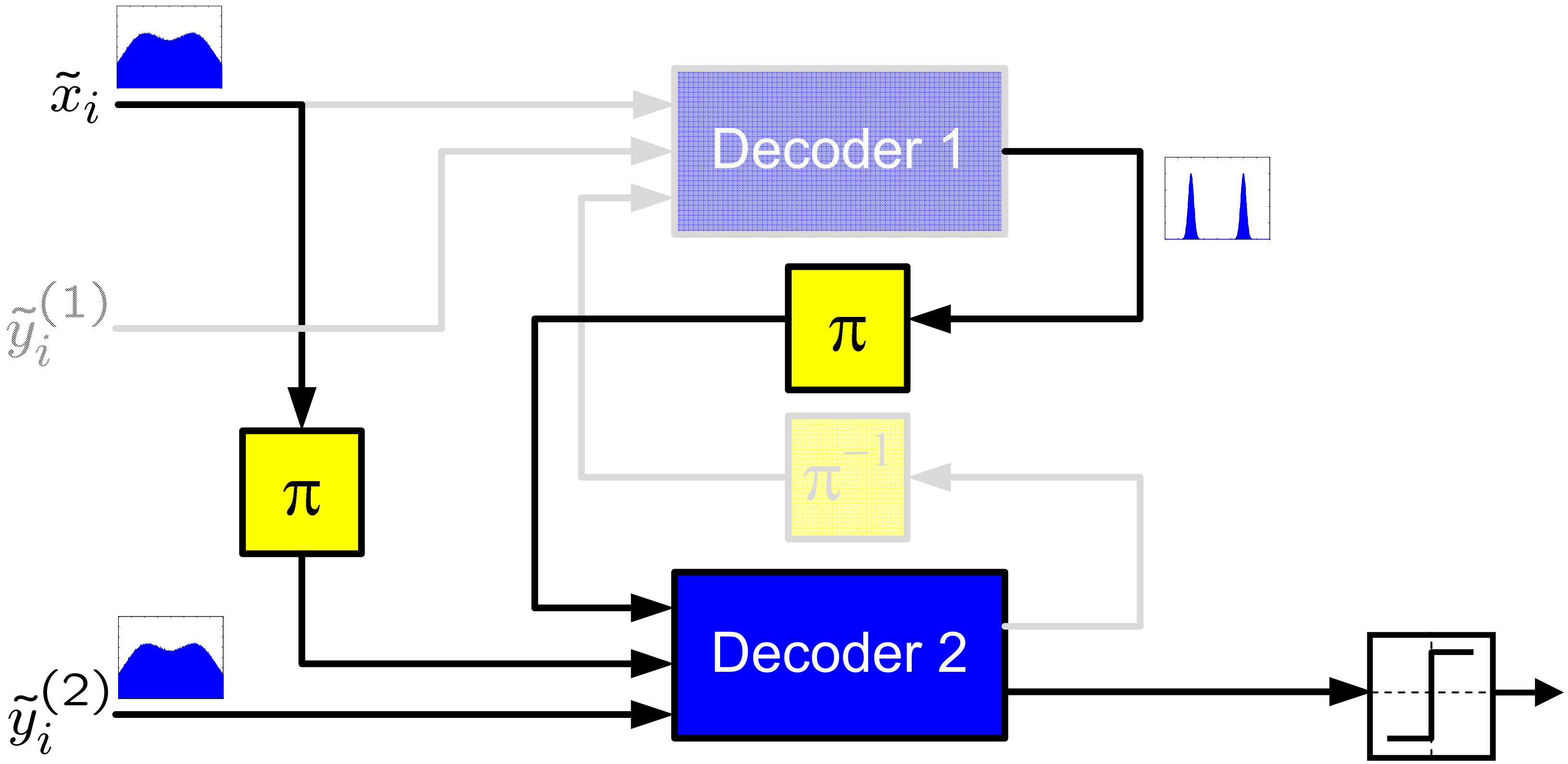
Turbo Coding & Decoding

- Turbo Decoding (n'th iteration, 1st step)



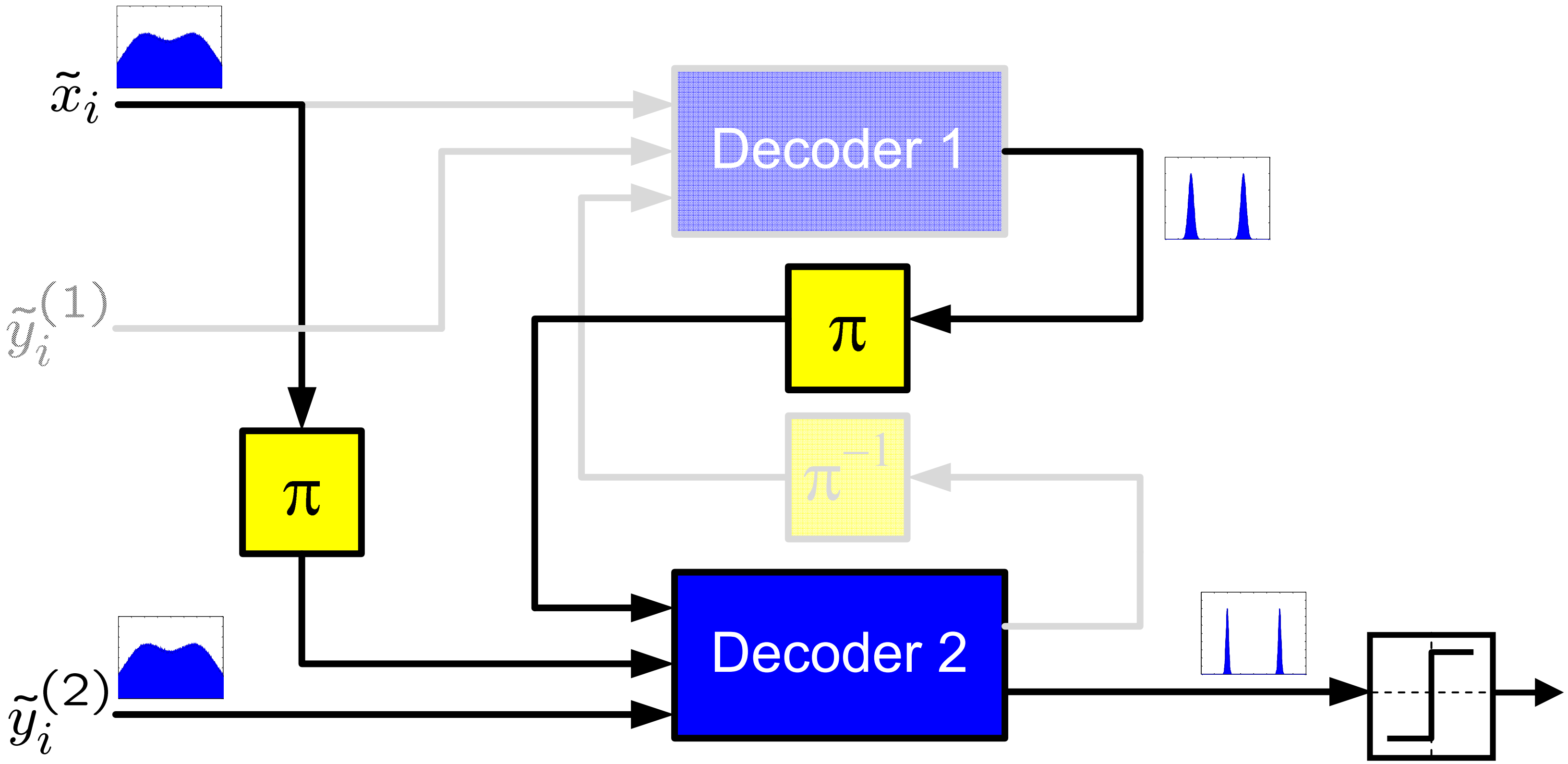
Turbo Coding & Decoding

- Turbo Decoding (n'th iteration, 2nd step)



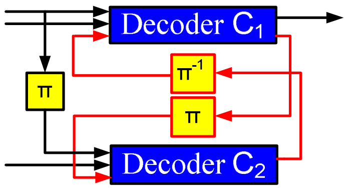
Turbo Coding & Decoding

- Turbo Decoding (n'th iteration, 2nd step)

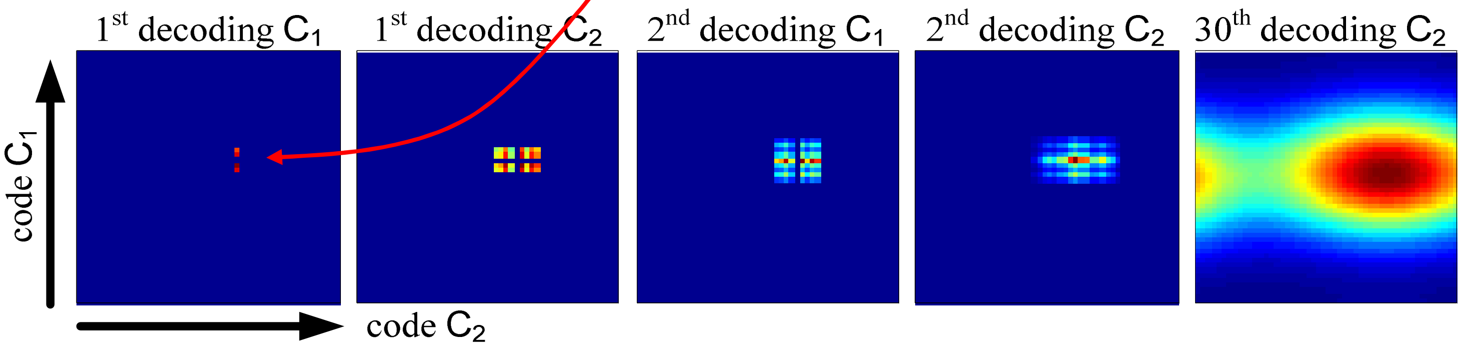


Turbo Principle - Interleaver Design

- Block interleaver vs. random interleaver
  - Example: Propagation of a single information

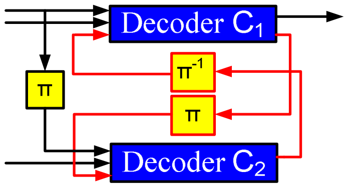


  - Block: (50 x 50)

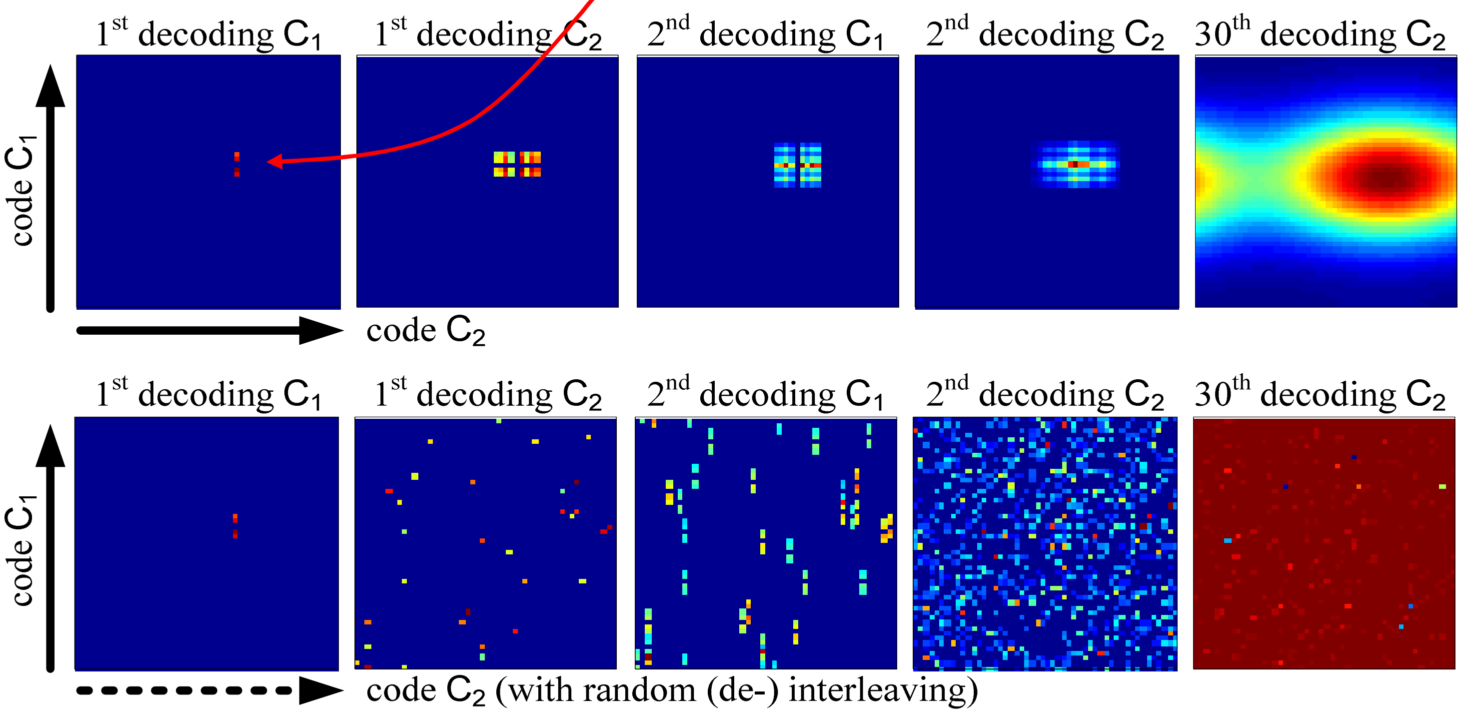


FlexCode Turbo Principle - Interleaver Design

- Block interleaver vs. random interleaver



  - Example: Propagation of a single information
  - Block:
    (50 x 50)
  - Random:



- Better information distribution with random interleavers
- Careful interleaver design required

FlexCode

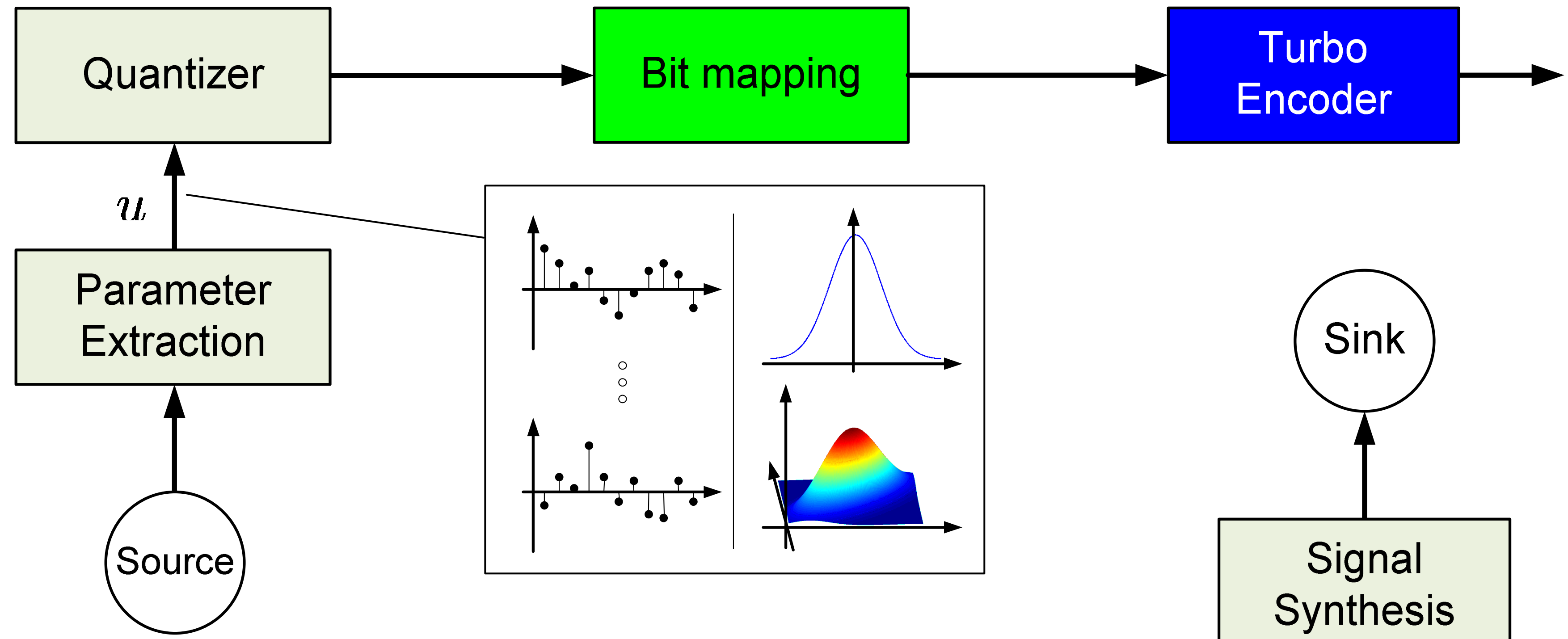
Turbo Source Coding

RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN RWTH AACHEN UNIVERSITY

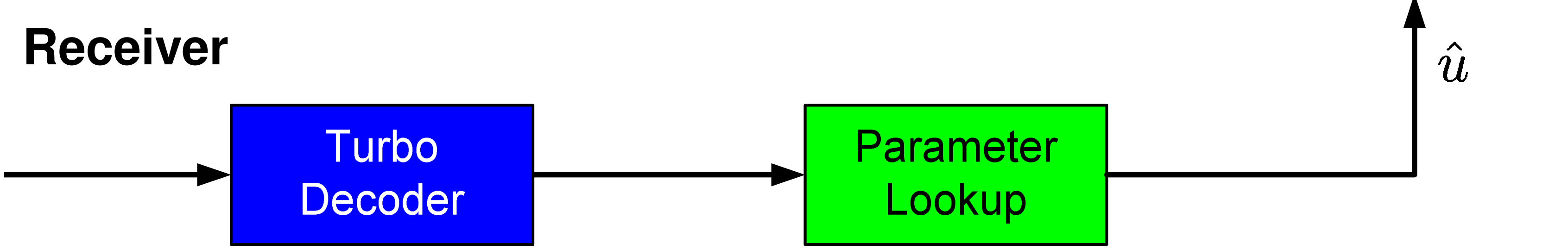
- Can Turbo codes be used for entropy coding?
- Yes!
  - Turbo Codes for compressing **binary memoryless** sources **[Garcia-Frias 02]**
  - Only transmitting a fraction of the output bits such that the overall coding rate > 1
  - Decoder has to take into consideration the statistics of the source (unequal distribution of bits)
  - Source statistics can be estimated at the decoder

Transmission with Turbo Source Coding

- **Transmitter**

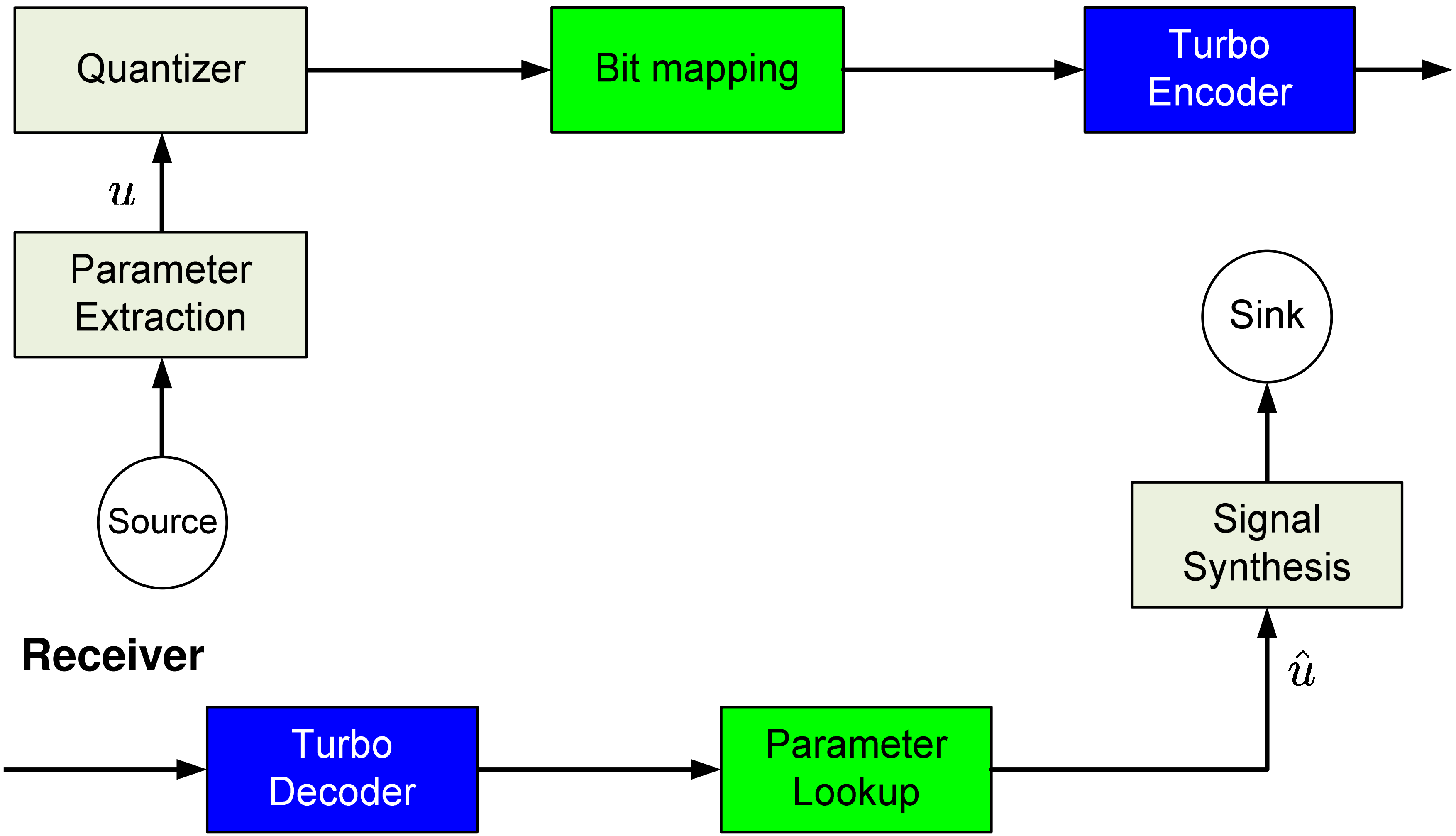


- **Receiver**



Transmission with Turbo Source Coding

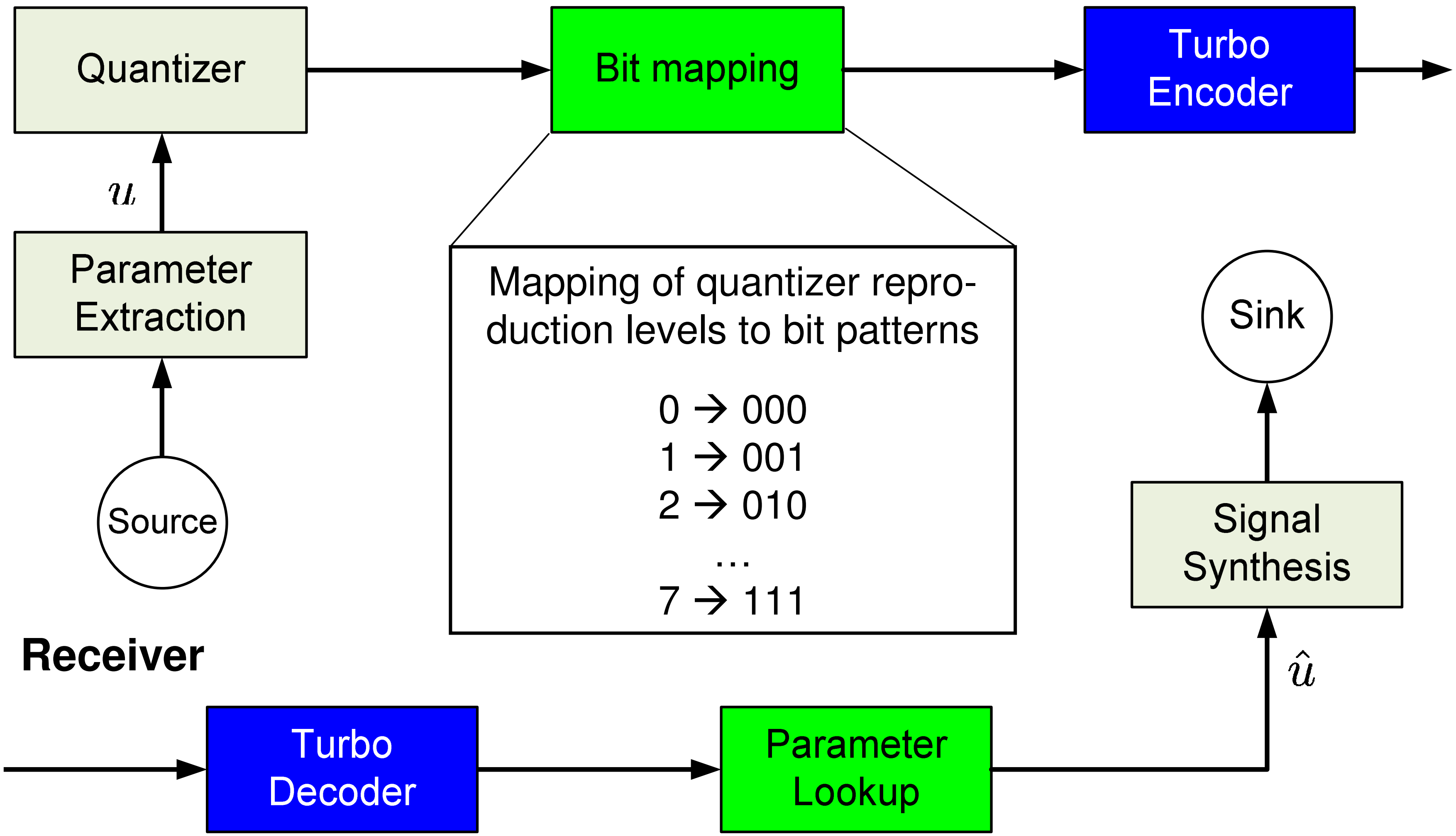
- **Transmitter**



- **Receiver**

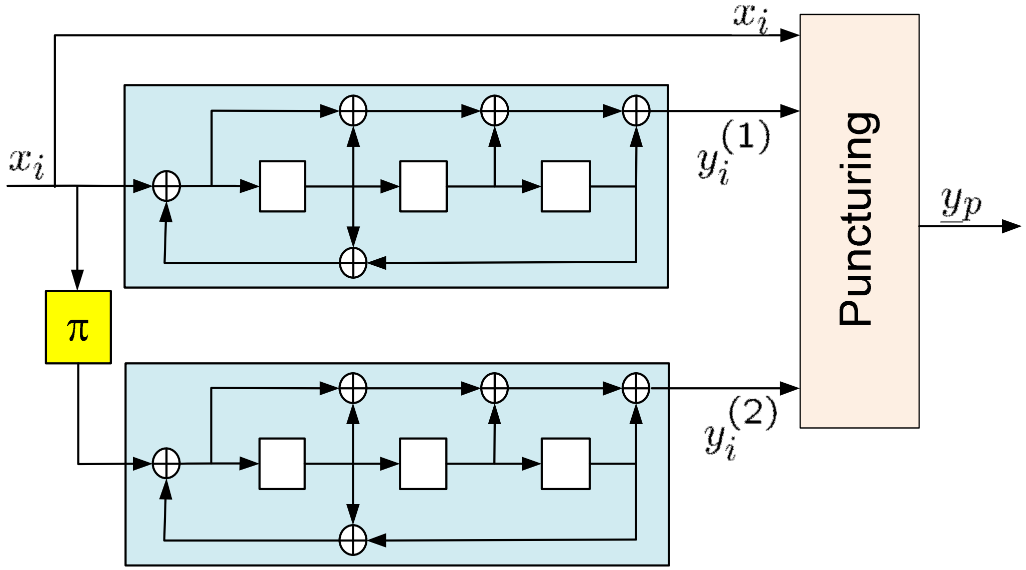
Transmission with Turbo Source Coding

- **Transmitter**



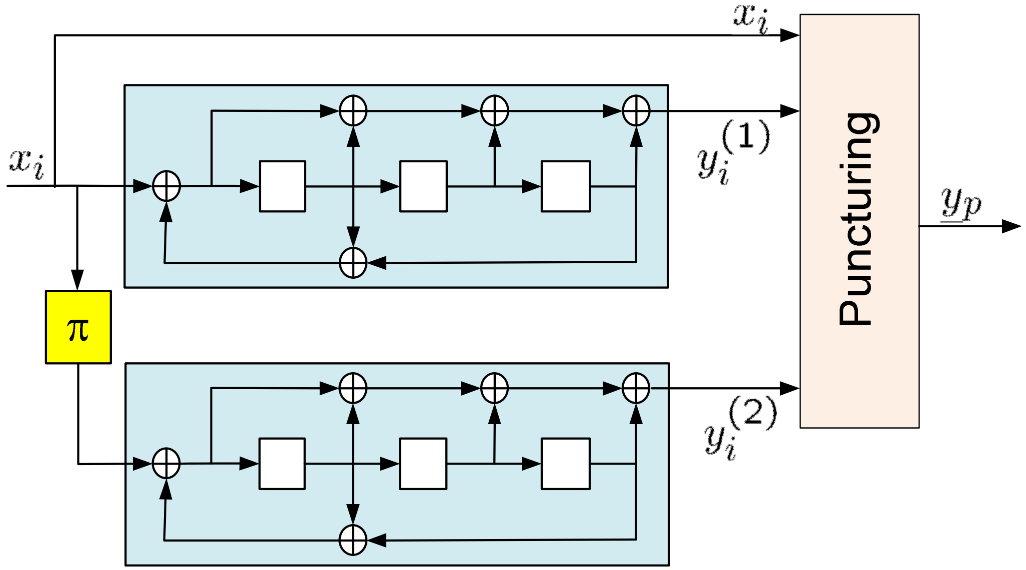
Turbo Source Coding

- Turbo coder, channel coding rate 1/3 (1 bit → 3 bit)



Turbo Source Coding

- Turbo coder, channel coding rate 1/3 (1 bit → 3 bit)

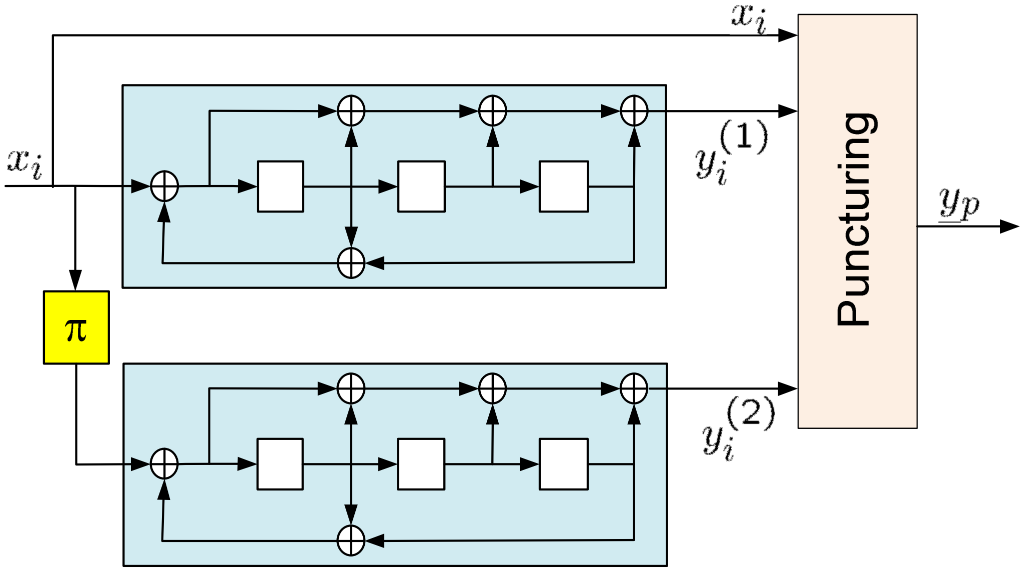


- Before puncturing:

$$\underline{y} = \left(x_1, y_1^{(1)}, y_1^{(2)}, \quad x_2, y_2^{(1)}, y_2^{(2)}, \quad x_3, y_3^{(1)}, y_3^{(2)}, \ldots\right)$$

Turbo Source Coding

- Turbo coder, channel coding rate 1/3 (1 bit → 3 bit)



- Before puncturing:

$$\underline{y} = \left(x_1, y_1^{(1)}, y_1^{(2)}, \quad x_2, y_2^{(1)}, y_2^{(2)}, \quad x_3, y_3^{(1)}, y_3^{(2)}, \ldots\right)$$

- After puncturing (compression ratio 0.5, 1 bit → 0.5 bit)

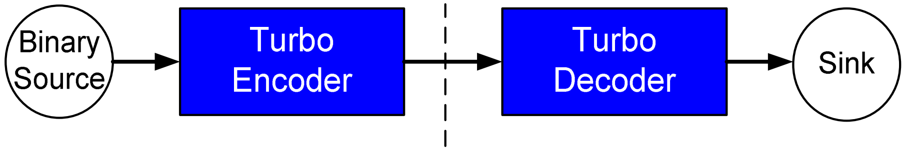
$$\underline{y}_P = \left(x_1, y_3^{(1)}, y_5^{(2)}, x_7, y_9^{(1)}, y_{11}^{(2)} \ldots\right)$$

Turbo Source Coding

- Puncturing unit corresponds to *binary erasure channel*
- Adapt puncturing w.r.t. source statistics
- Theoretical minimum rate: source entropy
- Realization of puncturing
  - Regular puncturing
  - Pseudo-random puncturing
- Puncturing has to be known at the receiver
- Adaptively increase number of transmitted bits with increasing channel noise

Turbo Source Coding

- Simulation results for a binary source **[Garcia-Frias 02]**
- Comparison with standard Unix compression tools *compress*, *gzip*, and *bzip2*



Source entropy $H(X)$	Achieved compression ratios: **Turbo**	*compress*	*gzip*	*bzip2*
0.0808	**0.16**	0.16	0.16	0.15
0.2864	**0.38**	0.41	0.41	0.44
0.4690	**0.58**	0.65	0.60	0.67
0.6098	**0.75**	0.83	0.72	0.83
0.7219	**0.87**	0.98	0.80	0.96

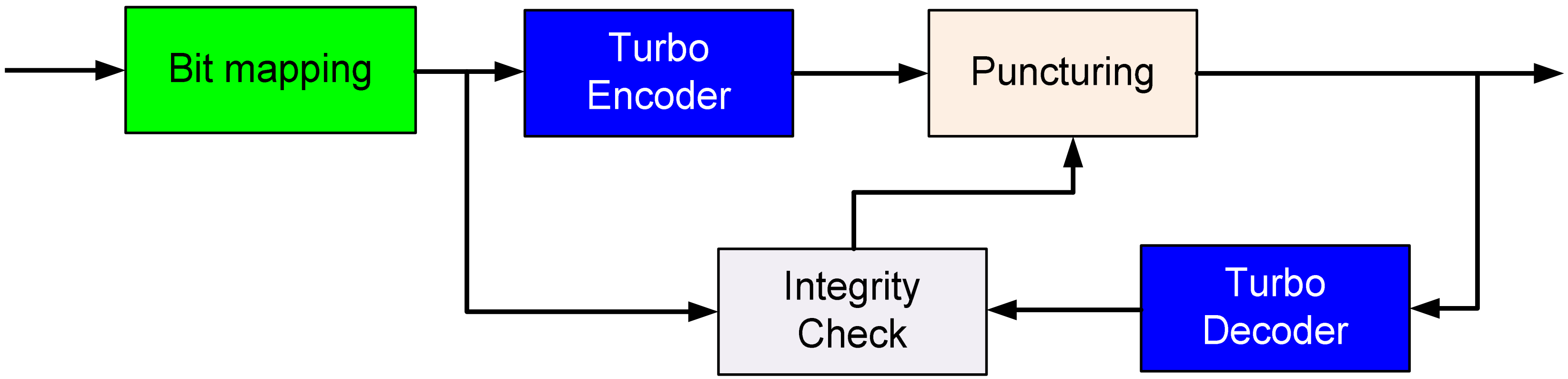
Turbo Source Coding

- Advantages:
  - Robustness against transmission errors
  - If the channel quality drops, puncturing can be changed in order to transmit more parity bits
  - High flexibility by adapting puncturing on the fly

- Disadvantages:
  - Higher computational costs than conventional entropy coding schemes
  - Lossless compression is not guaranteed, Turbo decoder might not decode every bit correctly
  - Difficult to adapt to varying parameter statistics for the use in speech and audio codecs

Lossless Turbo Source Coding

- Lossless Turbo source coding [Hagenauer 04]
- Adapt puncturing such that lossless decoding is possible
- Analysis-by-Synthesis encoder:
  - change puncturing and test if decodable without error
  - Small amount of side information required



Non-Binary Sources

- Turbo Source Coding only for binary sources
- Feasible only if bit pattern after source coding has low entropy $H(X) < 1$
- Extension towards non-binary sources
  - Utilization of non-binary Turbo codes **[Zhao 02]**
  - Utilization of special binary LDPC **[Zhong 05]**
  - Utilization of non-binary LDPC codes **[Potluri 07]**
- Joint source-channel coding approach

> „However, any redundancy in the source will usually help if it is utilized at the receiving point. [...] redundancy will help combat noise.“, **Shannon 1948**

Iterative Source-Channel Decoding

- Approach: Enhancement of the robustness of transmission of variable length codes
- Iterative Source-Channel Decoding (ISCD) of variable length codes (VLC) **[Guyader 01]**
  - Combats adverse effects of channel noise
  - Exploits the structure and redundancy of variable length codes
  - Achieve near-capacity system performance
  - Works considerably well for Huffman codes
  - Difficult to adapt to arithmetic codes
  - Very high computational complexity
  - Limited flexibility!

Iterative Source-Channel Decoding

- Approach: Enhancement of the robustness of transmission of variable length codes
- Iterative Source-Channel Decoding (ISCD) of variable length codes (VLC) **[Guyader 01]**
  - Combats adverse effects of channel noise
  - Exploits the structure and redundancy of variable length codes
  - Achieve near-capacity system performance
  - Works considerably well for Huffman codes
  - Difficult to adapt to arithmetic codes
  - Very high computational complexity
  - Limited flexibility!

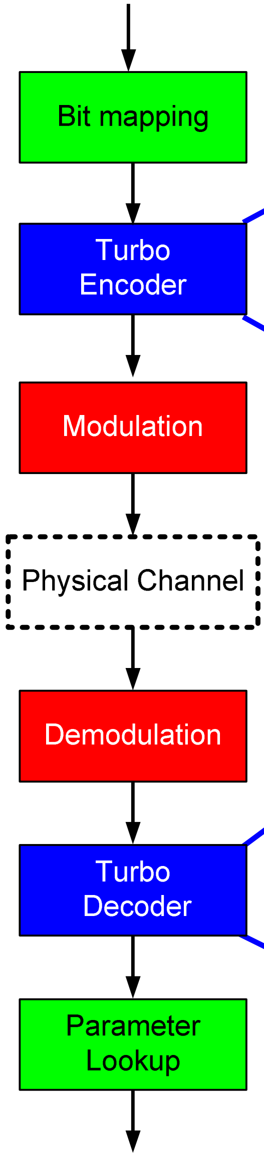
FlexCode

Iterative Source-Channel Decoding

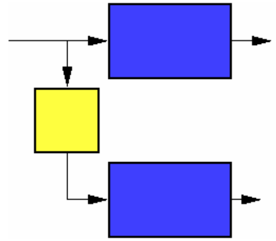
RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN RWTH AACHEN UNIVERSITY

- Iterative Source-Channel Decoding (ISCD) **[Adrat 01]** for fixed-length codes (FLC)
  - Combat adverse effects of channel noise
  - Exploit residual source redundancy
  - Achieve near-capacity overall system performance **[Clevorn 06]**
- Can also be used effectively for compression **[Thobaben 08]**
  - Leave all redundancy in the source symbols
  - Utilization of redundant bit mappings
  - Puncture output of convolutional code in order to obtain compression

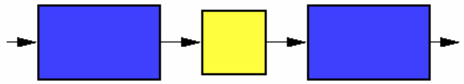
Turbo Codes, Concept



Concatenated Encoding

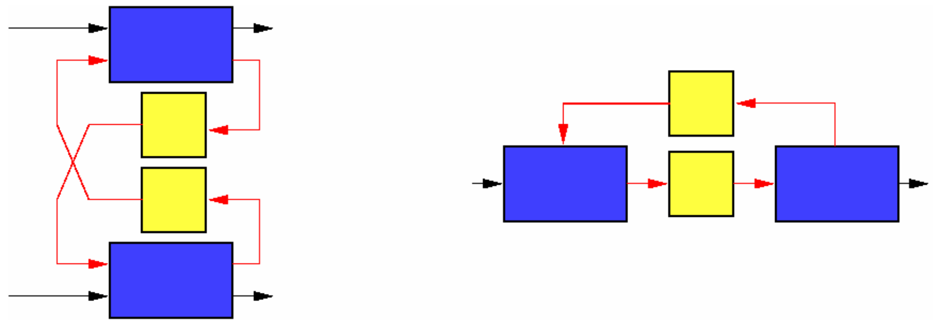


parallel scheme
[Berrou 93]



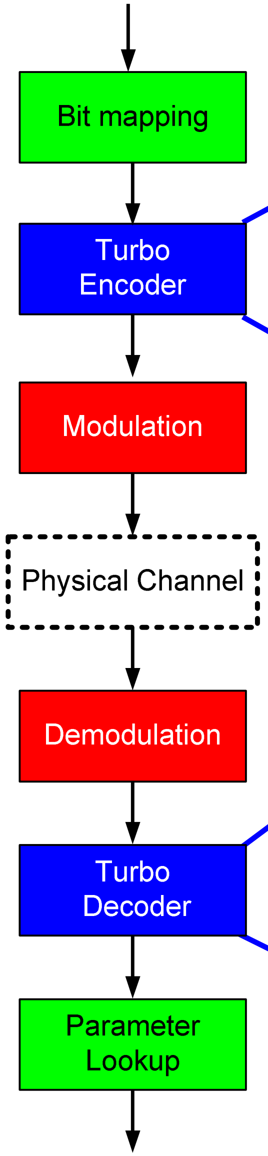
serial scheme
[Benedetto 98]

Iterative Turbo Decoding

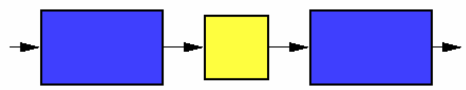


advantages by iterative feedback!

Turbo Codes, Concept

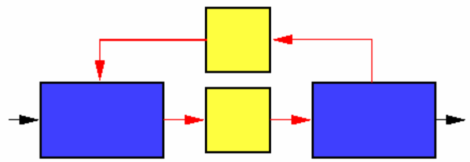


Concatenated Encoding



serial scheme
[Benedetto 98]

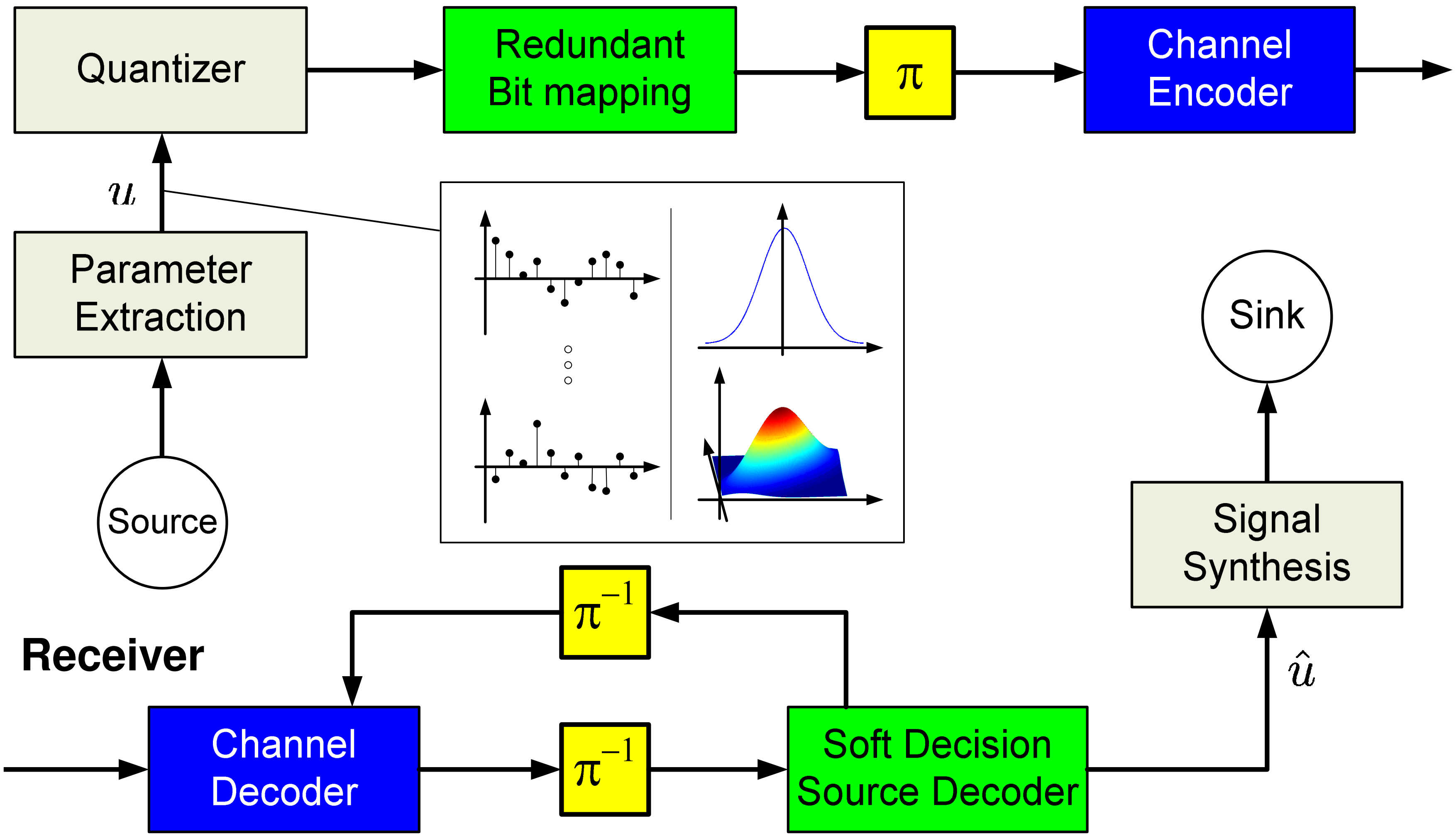
Iterative Turbo Decoding



advantages by iterative feedback!

Transmission with ISCD

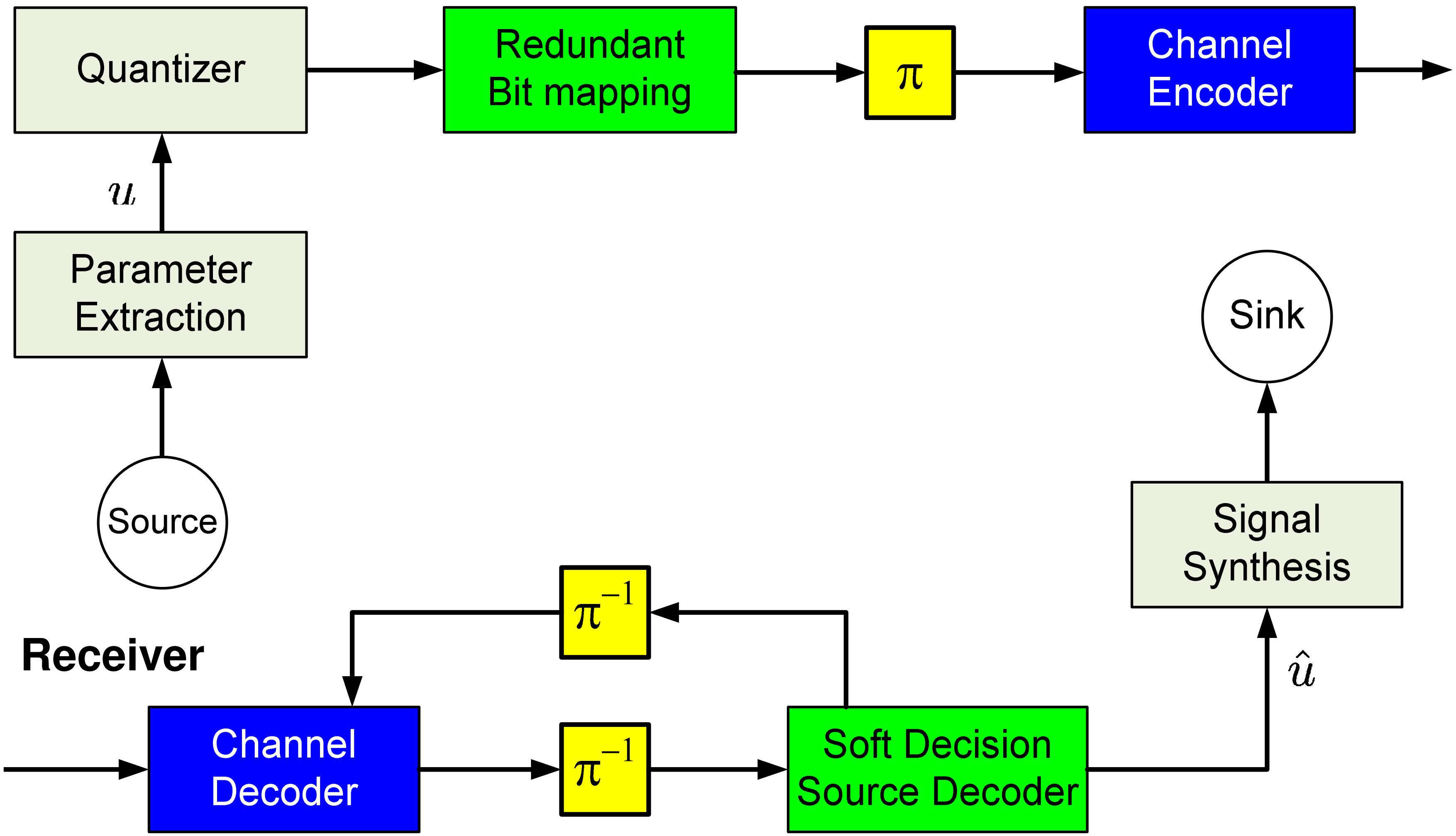
- **Transmitter**



- **Receiver**

Transmission with ISCD

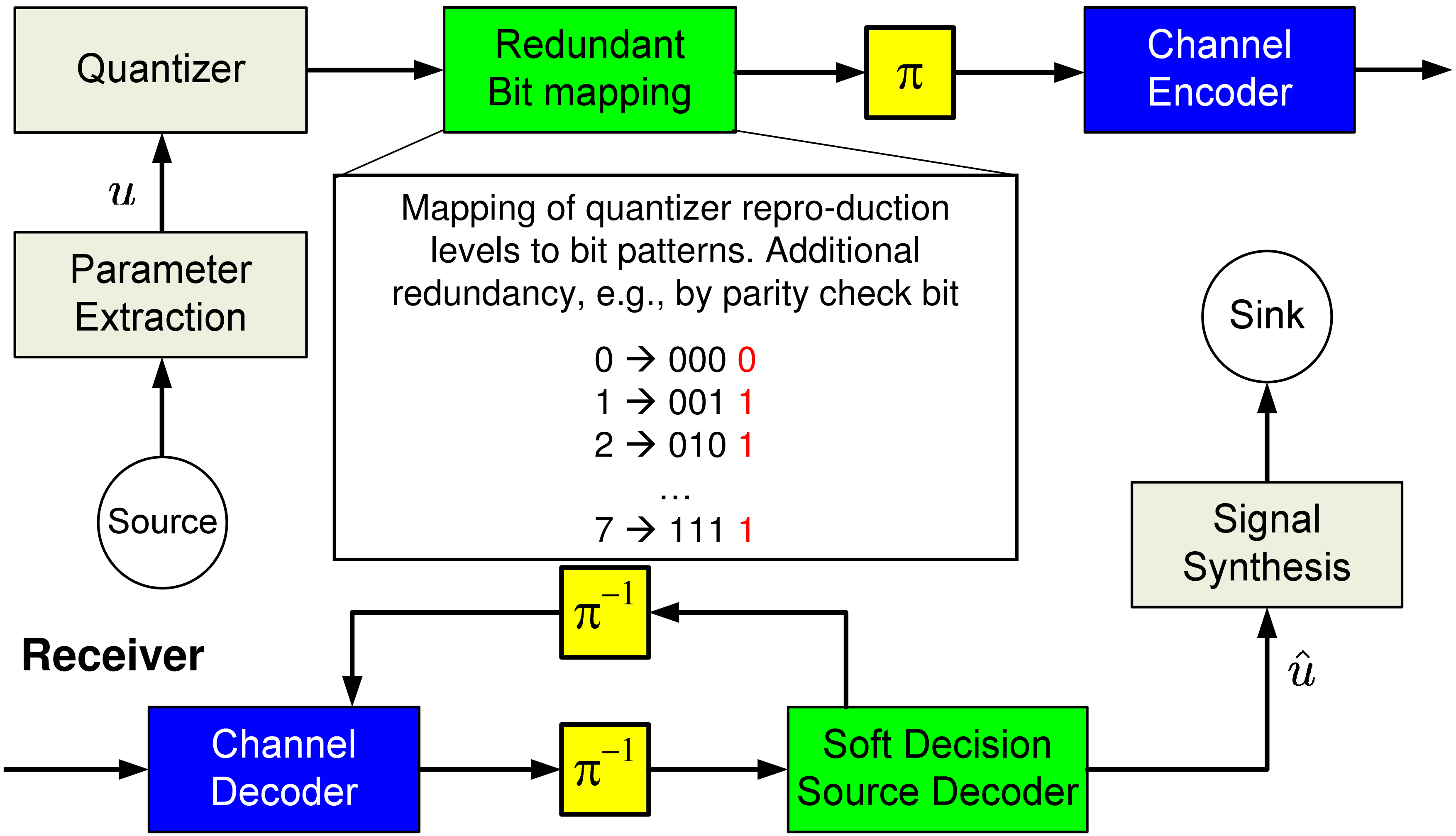
- **Transmitter**



- **Receiver**

Transmission with ISCD

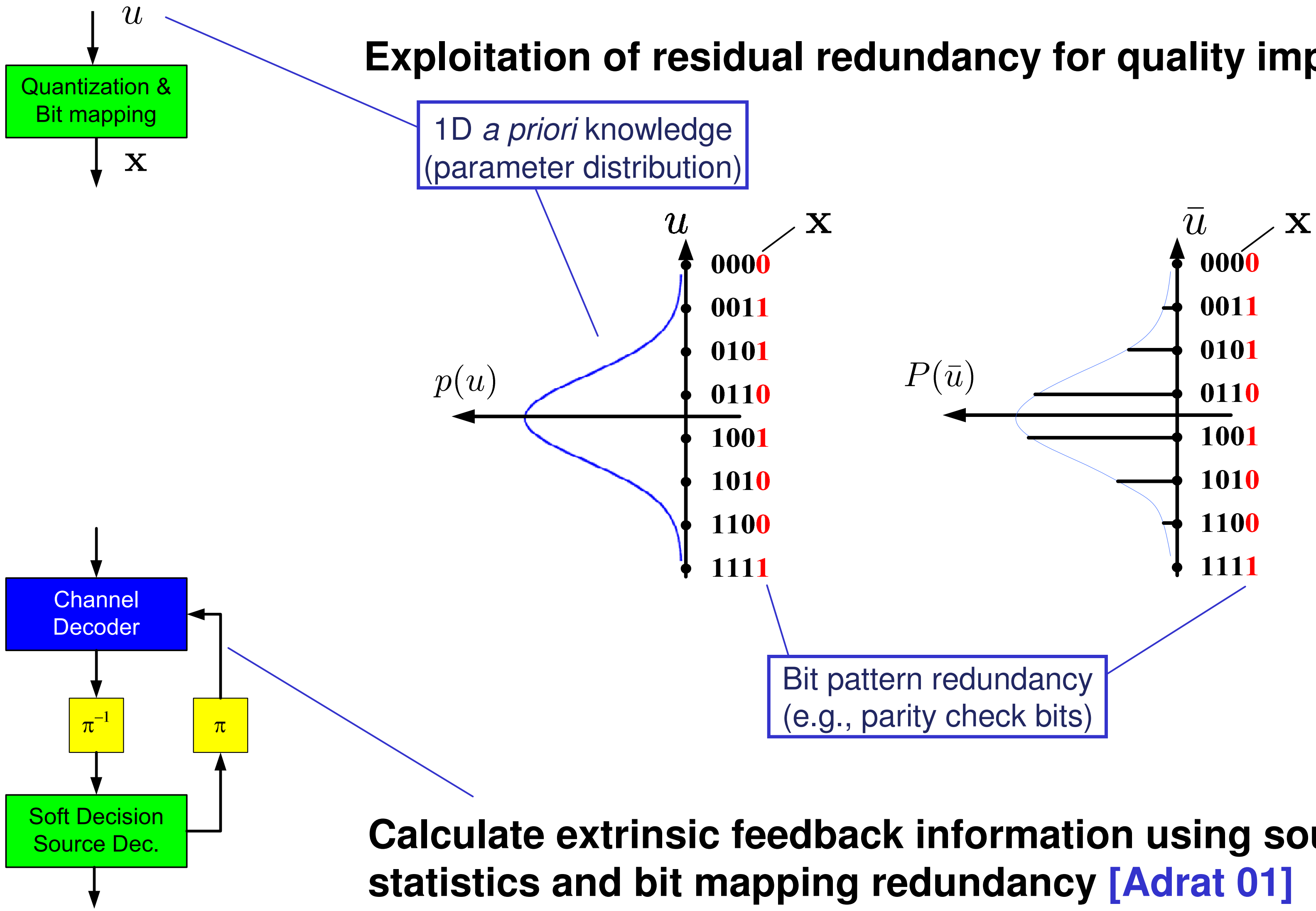
- **Transmitter**



- **Receiver**

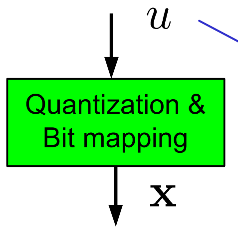
Soft Decision Source Decoding

Exploitation of residual redundancy for quality improvement



Calculate extrinsic feedback information using source statistics and bit mapping redundancy [Adrat 01]

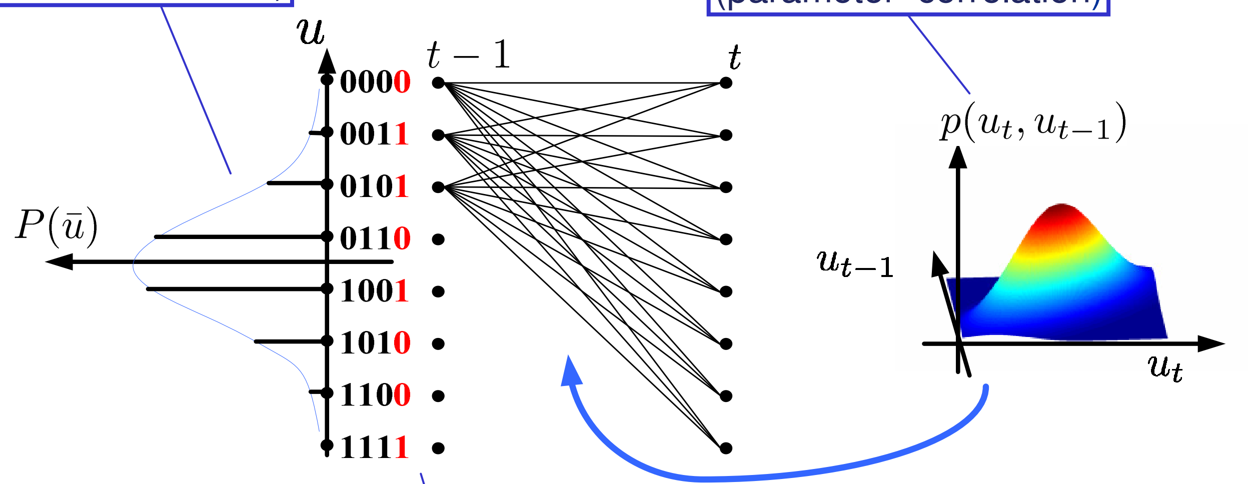
Soft Decision Source Decoding



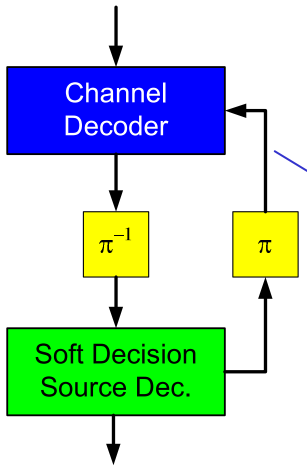
Exploitation of residual redundancy for quality improvement

1D *a priori* knowledge (parameter distribution)

2D *a priori* knowledge (parameter correlation)



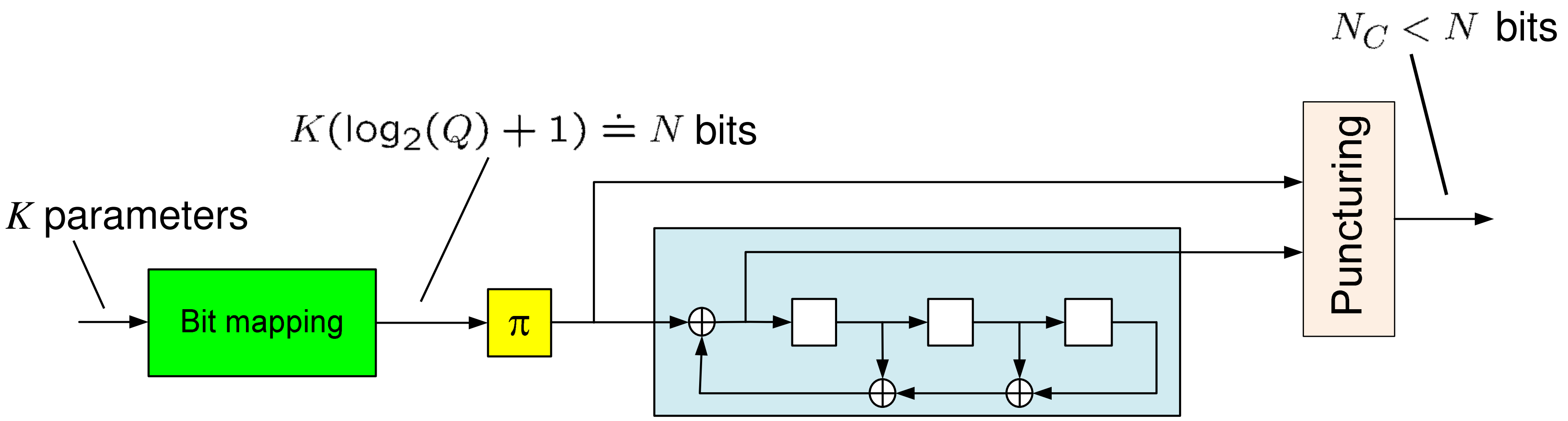
Bit pattern redundancy (e.g., parity check bits)



Calculate extrinsic feedback information using source statistics of order 0 and 1 and bit mapping redundancy [Adrat 01]

ISCD Source Compression

- Simulation example
- System components
  - Scalar quantization with Q levels
  - Single parity check code index assignment
  - R > 1 convolutional code, random puncturing **[Thobaben 07]**



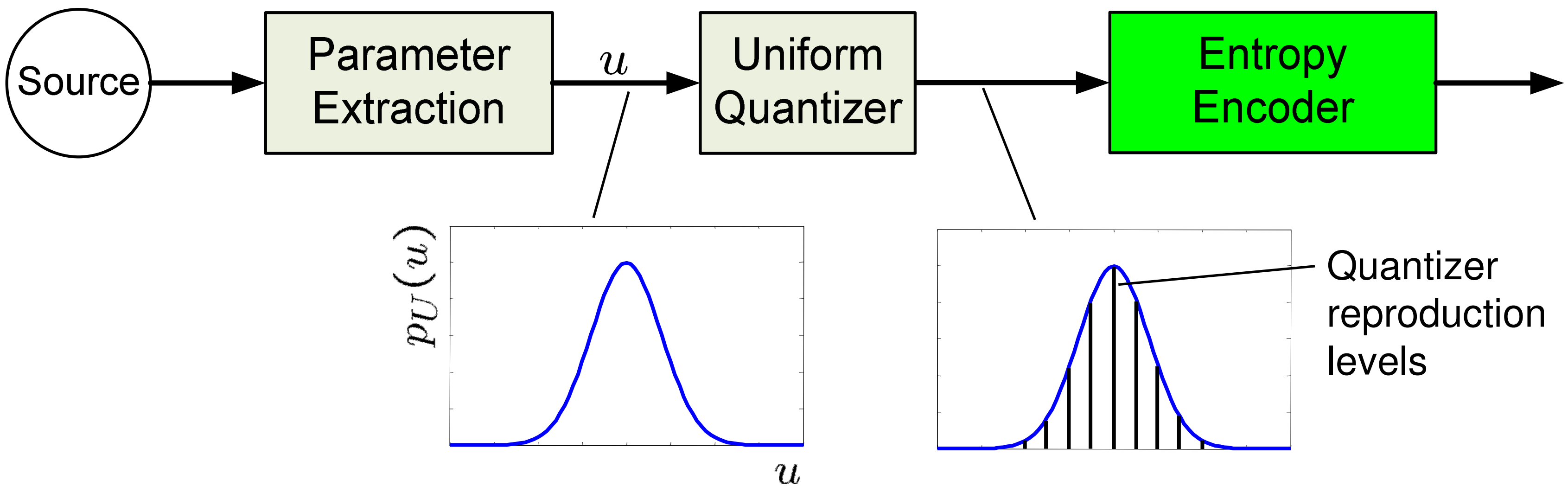
Iterative Source-Channel Decoding

- Simulation results for simple experiment
  - Gauss-Markov source (AR(1) process)
  - Lloyd-Max Quantization with Q = 16 levels
  - 25 decoding iterations
  - Unoptimized standard system components!

Source correlation ρ	Entropy $H(U_t \mid U_{t-1})$ / 4	Achieved compression ratios: **ISCD**	*compress*	*gzip*	*bzip2*
0.8	0.76	**0.83**	0.99	0.86	0.82
0.85	0.72	**0.78**	0.94	0.83	0.78
0.9	0.66	**0.74**	0.87	0.80	0.72
0.95	0.55	**0.62**	0.73	0.71	0.62

Possible Application in FlexCode

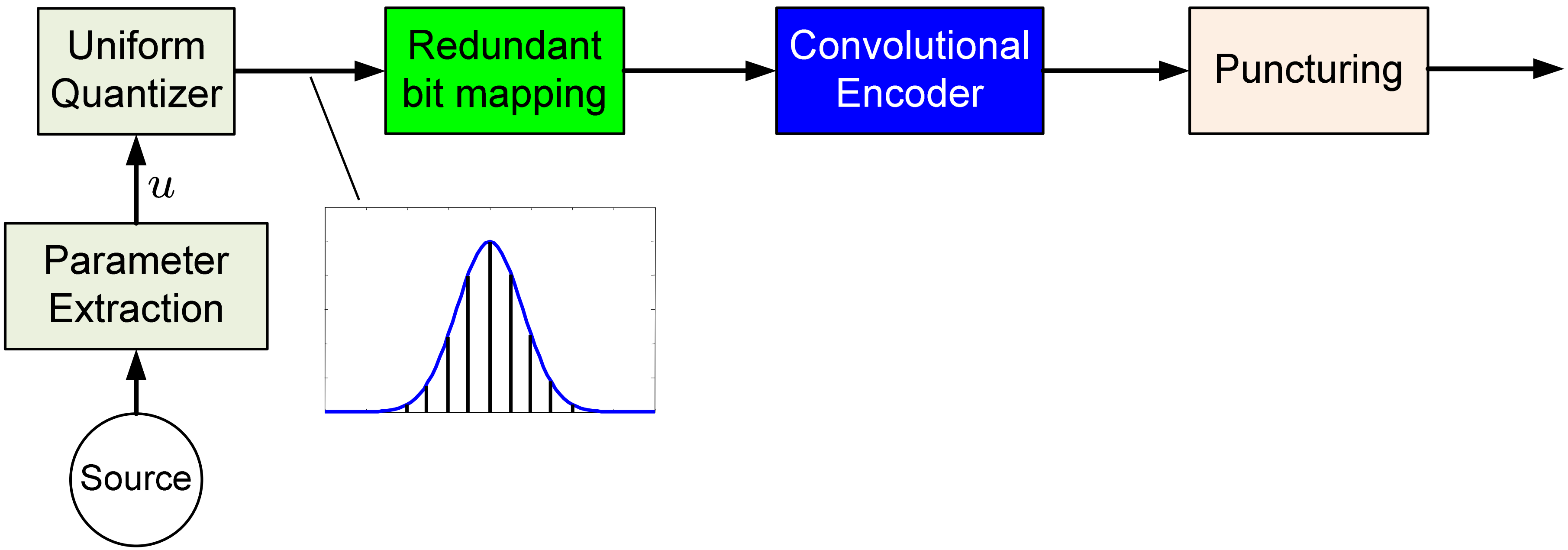
- Entropy coding for constrained entropy quantization



- Utilization of entropy coding, e.g., arithmetic coding
- Can be replaced by the presented compression scheme

Possible Application in FlexCode

- Entropy coding for constrained entropy quantization



- Flexible adaptation to varying channel conditions by puncturing adaptation

Conclusions

- Channel codes can be used for compression
- Turbo codes for compressing binary sources
- Iterative Source-Channel Decoding for fixed length codes
- Joint Source-Channel Coding with Iterative Decoding for Source Compression
- Promising results already with unoptimized system components

References

[Guionnet 04] T. Guionnet and C. Guillemot, “Soft and joint source-channel decoding of quasi-arithmetic codes,” *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 3, pp. 393–411, Mar. 2004.

[Caire 03] G. Caire, S. Shamai and S. Verdú, "Universal Data Compression with LDPC Codes," *Third International Symposium On Turbo Codes and Related Topics,* Brest, France, September 1-5, 2003

[Berrou 93] C. Berrou, A. Glavieux, P. Thitimajshima, „Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes“, *International Conference on Communications*, Geneve, Switzerland, Mai, 1993

[Benedetto 98] S. Benedetto, D. Divsalar, G. Montorsi, F. Pollara, „Analysis, Design, and Iterative Decoding of Double Serially Concatenated Codes with Interleavers“, *IEEE Journal on Sel. Areas in Comm.*, vol. 16, no. 2, February 1998

[Garcia-Frias 02] J. Garcia-Frias, Y. Zhao, „Compression of Binary Memoryless Sources Using Punctured Turbo Codes“, *IEEE Comm. Letters*, vol. 6, no. 9, September 2002

[Hagenauer 04] J. Hagenauer, J. Barros, A. Schaefer, „Lossless Turbo Source Coding with Decremental Redundancy“, *ITG Conference on Source and Channel Coding (SCC)*, Erlangen, 2004

[Zhao 02] Y. Zhao, J. Garcia-Frias, „Data Compression of Correlated Non-Binary Sources Using Punctured Turbo Codes“, *IEEE Data Compression Conference* (DCC), 2002

[Zhong 05] W. Zhong and J. García-Frías: “Compression of Non-Binary Sources Using LDPC Codes”, *CISS*, March 2005, Baltimore, Maryland.

[Potluri 07] M. Potluri, S. Chilumuru, S. Kambhampati, K. R. Namuduri, „Distributed Source Coding using non-binary LDPC codes for sensor network applications“, *Canadian Workshop on Information Theory*, June 2007

[Guyader 01] A. Guyader, E. Fabre, C. Guillemot, and M. Robert, “Joint source-channel turbo decoding of entropy-coded sources,” *IEEE Journal on Sel. Areas in Comm.*, vol. 19, no. 9, pp. 1680–1696, Sept. 2001.

[Adrat 01] M. Adrat, J.-M. Picard, P. Vary, „Soft-Bit Source Decoding Based on the Turbo Principle“, *IEEE Vehicular Technology Conference (VTC-Fall),* Atlantic City, Oct. 2001.

References

[Clevorn 06] T. Clevorn, L. Schmalen, P. Vary „On the Optimum Performance Theoretically Attainable for Scalarly Quantized Correlated Sources, *International Symposium on Information Theory and its Applications (ISITA),* Seoul, Korea, Oct. 2006.

[Thobaben 08] R. Thobaben, L. Schmalen, P. Vary, „Joint Source-Channel Coding with Inner Irregular Codes", *International Symposium on Information Theory (ISIT)*, Toronto, CA, July 2008

[Thobaben 07] R. Thobaben, „A new transmitter concept for iteratively-decoded source-channel coding schemes", *IEEE SPAWC*, Helsinki, June 2007

RWTH AACHEN UNIVERSITY

FlexCode

Turbo Source Coding

Laurent Schmalen and Peter Vary

http://www.flexcode.eu