The Turbo-Compression

Andrej Rikovsky

JASS 2005

Institute for Communications Engineering (LNT)
Munich University of Technology (TUM)
D-80290 Munich, Germany
E-mail: rikovsky@mytum.de

Overview

- Problem statement
- Source coding & decoding
- Decremental redundancy and algorithm
- Algorithm analysis using EXIT charts
- Examples

Introduction

What is Data Compression?

The goal of data compression is to represent an information source as accurately as possible using the fewest number of bits.

- Data compression (source coding) removes all the redundancy to form the most compressed version possible.
- dual problem to Data Transmission (channel coding)

Turbo principle approach

- Source coding
  - The redundancy of data is removed step-by-step as long as the decoder can guarantee perfect reconstruction (Decremental Redundancy)
- Channel coding
  - Additional parity bits are gradually transmitted through a noisy channel until the decoder can correct all errors (Incremental Redundancy)

Turbo-Coding principle

- Developed for **channel coding**
  - performs close to Shannon channel capacity
  - encoded bits are used for error protection

- Applied for **source coding** of BMS (Binary Memoryless Source)
  - performs close to entropy of the source
  - encoded bits are heavily punctured to achieve the desired compression rate

Problem statement

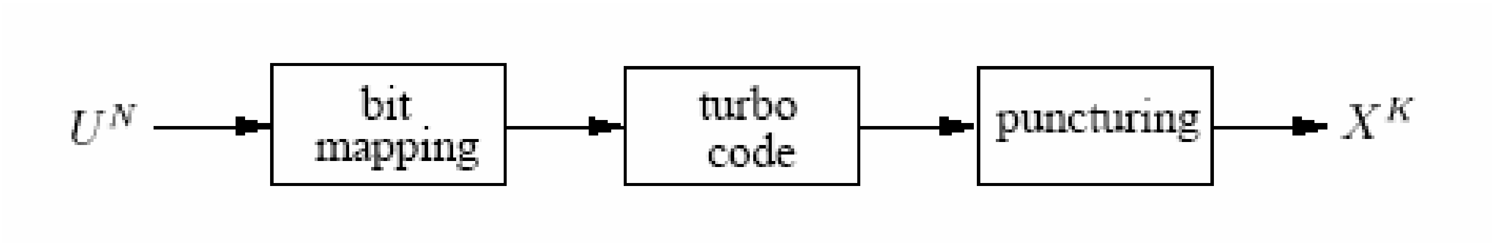
Let U be a DMS (Discrete Memoryless Source) emitting i.i.d symbols from alphabet $\mathcal{U} = \{1,2,\ldots,L\}$ characterized by pmf $p(u)$

Entropy of the source

$$H(U) = E\left\{\log\left(\frac{1}{p(u)}\right)\right\} = -\sum_{i=1}^{L} p(i)\log p(i)$$

Problem statement

Turbo source encoder



- Block of N symbols $U^N = U_1, U_2, ..., U_N$ encoded to binary codeword $X^K = X_1, X_2, ..., X_K$ with $X_i \in \{0,1\}$ $X_i \in \{+1,-1\}$ respectively

- The source code rate

$$R = K / N$$

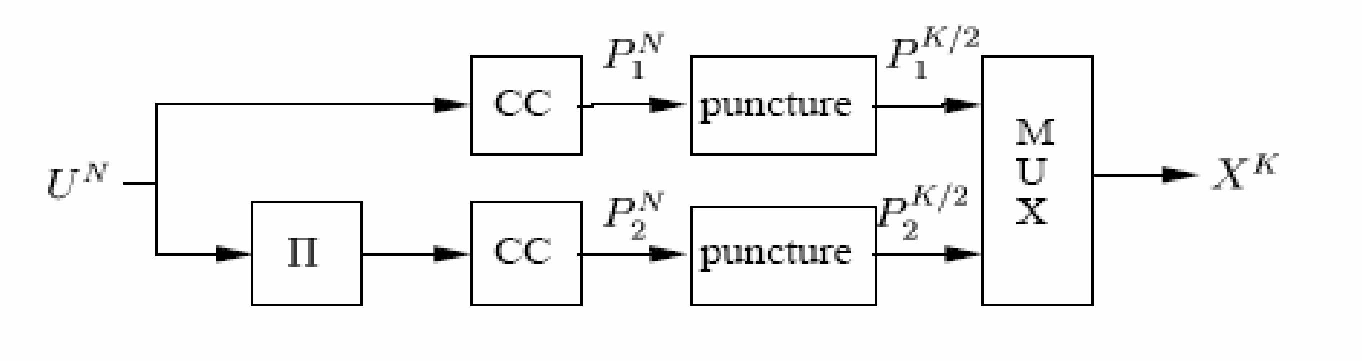
Source coding theorem - U^N can be perfectly (lossless) reconstructed from X^K of length $K \cong NH(U)$ for N sufficiently large.

Problem statement

What are the tasks stated?

- To guarantee lossless source coding.
- To design the turbo-codes and puncturing matrices to achieve maximal compression rate (close to entropy)

Source encoding



- Puncturing parity bits $P_1^{\,N}$, $P_2^{\,N}$
- Puncturing scheme as BEC with adjusted erasure probability
- Ideal proportion of erased bits $\varepsilon \approx 1 - \frac{H(U)}{2}$

gives the length of X sequence $\boldsymbol{K=NH(U)}$

Source encoding

Consider a block of source symbols U^N with the entropy of the source $H(U)$.

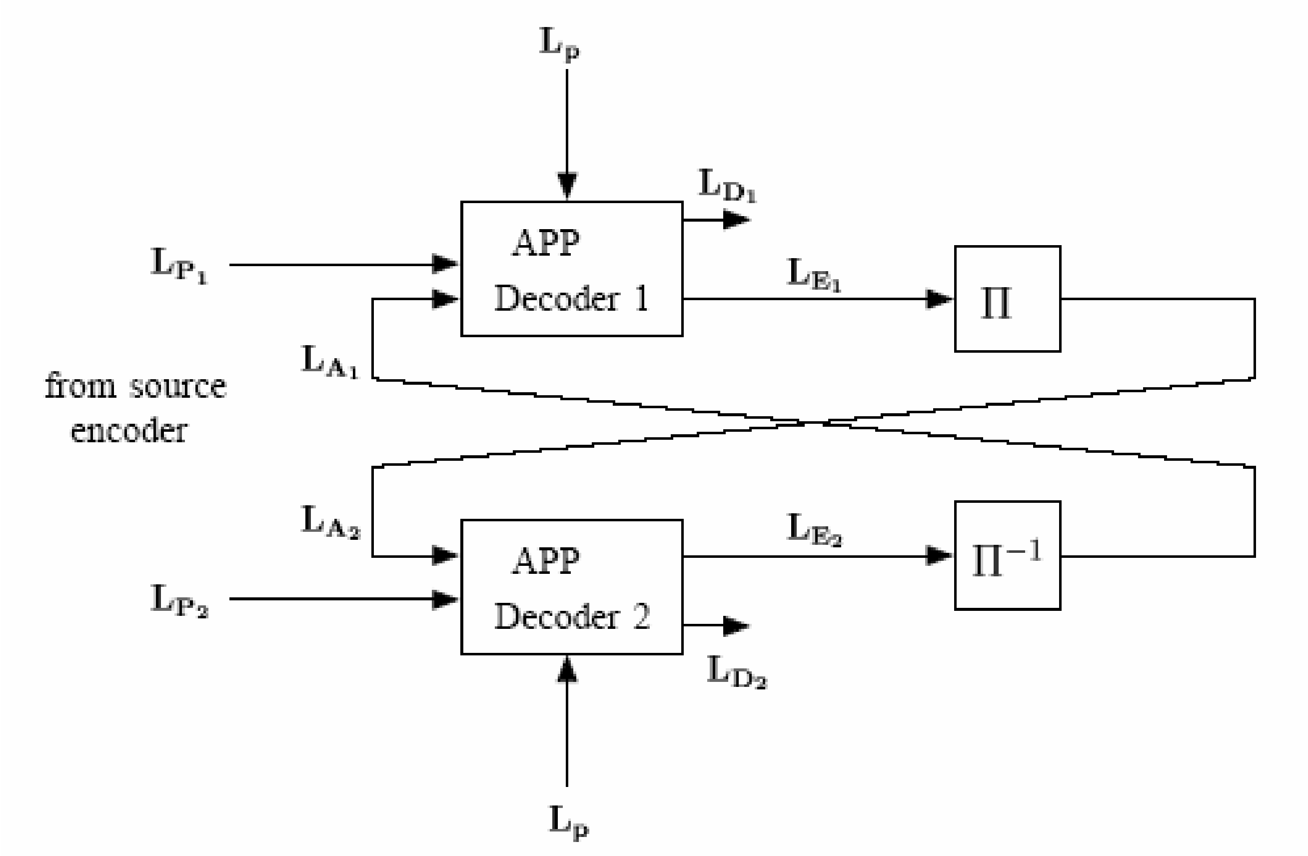
- To encode binary source we pass each of the sequences U^N and $\prod(U^N)$ through rate 1 convolution codes (CC) (or scrambles) with feedback to generate the parity sequences P_1^N, P_2^N

$\prod$ - defines the interleaver function

As opposed to PCC in channel coding we **discard the source bits**.

Source decoding

Parallel turbo source decoder



Source decoding

Decoder uses L-values input. Given random variable *U* where $u = \{+1, -1\}$ the L-values is defined as

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

L_{A1}, L_{A2} – a priori L-value sequences

L_{E1}, L_{E2} – extrinsic L-value sequences

Since parity bit sequences $P_1^{K/2}, P_2^{K/2}$ can be thought of as being transmitted over BEC (i.e. punctured), the corresponding input sequences L_{P1}, L_{P2} take on the L-values $\pm\infty$ (perfectly known bits) or 0 (erased)

Source decoding

In case of non-uniform binary source with $P(U{=}{+}1) = p$ and $P(U{=}{-}1){=}1{-}p$ for which $H(U) = H_b(p)$, each decoder has additional input L_P where each element of the vector is equal to

$$L_p = \ln\left(\frac{p}{1-p}\right)$$

“source a priori knowledge”, different from “learnt a priori knowledge” learnt during iteration.

L_P, L_{A1} – initialized to zero in the first iteration

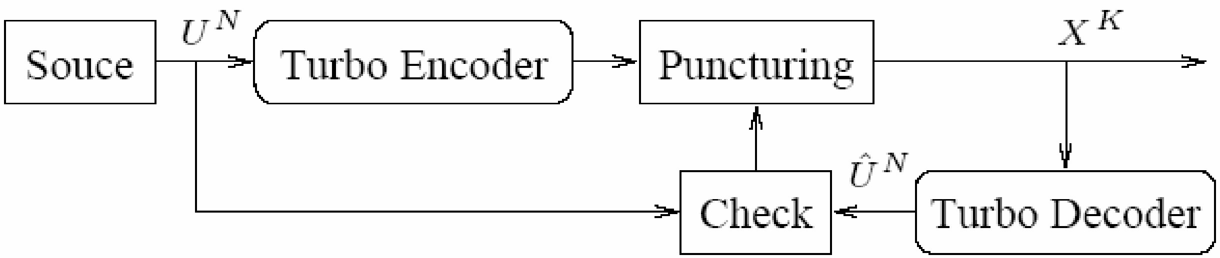
The extrinsic values are given as $L_{E1} = L_{D1} - L_{A1} - L_P$

Decremental redundancy

For lossless source coding the convergence of iterative algorithm has to be guaranteed (puncturing only so many bits that can be corrected)

- The encoder is testing the decodablity of its output.
- The parity bits are punctured on a step by step basis (decremental redundancy) as long as the compressed block can be still decoded error free.

Lossless compression is achieved by puncturing and verifying the integrity of the reconstructed source sequence

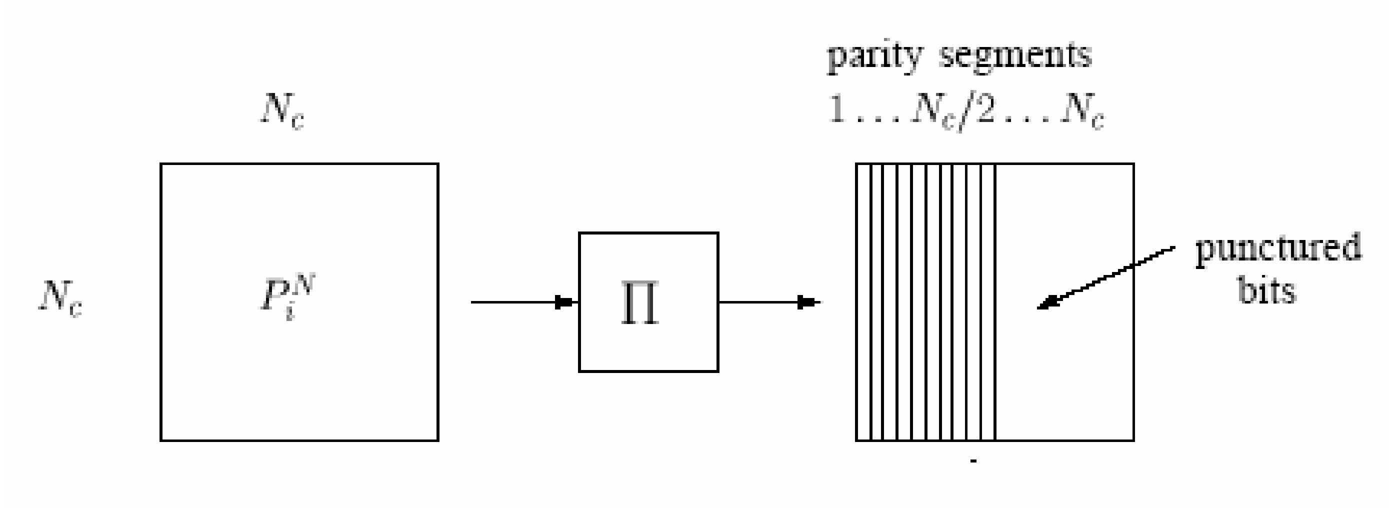


Decremental redundancy

Puncturing requirements

- The puncturing steps must be small enough to allow the compression rates to come close to entropy
- The punctured parity bits should be spread out in the block to guarantee the successful decoding (avoid long erasure bursts)
- The side information, i.e. the extra bits needed for the decoder to identify the positions of the punctured bits should be negligible in comparison to block length

Algorithm for decremental redundancy



- The parity bits P_i^N with $i=1,2$ of each block are written line by line in a matrix such that $N = N_C^2$.
- Parity segments are indexed $i \in \{1,\ldots,N_C\}$ and punctured by step $2N_C/N$
- To spread out the erased bits we interleave the parity bits before puncturing them

Note: We can use the same interleaver as for the turbo code.

Algorithm for decremental redundancy

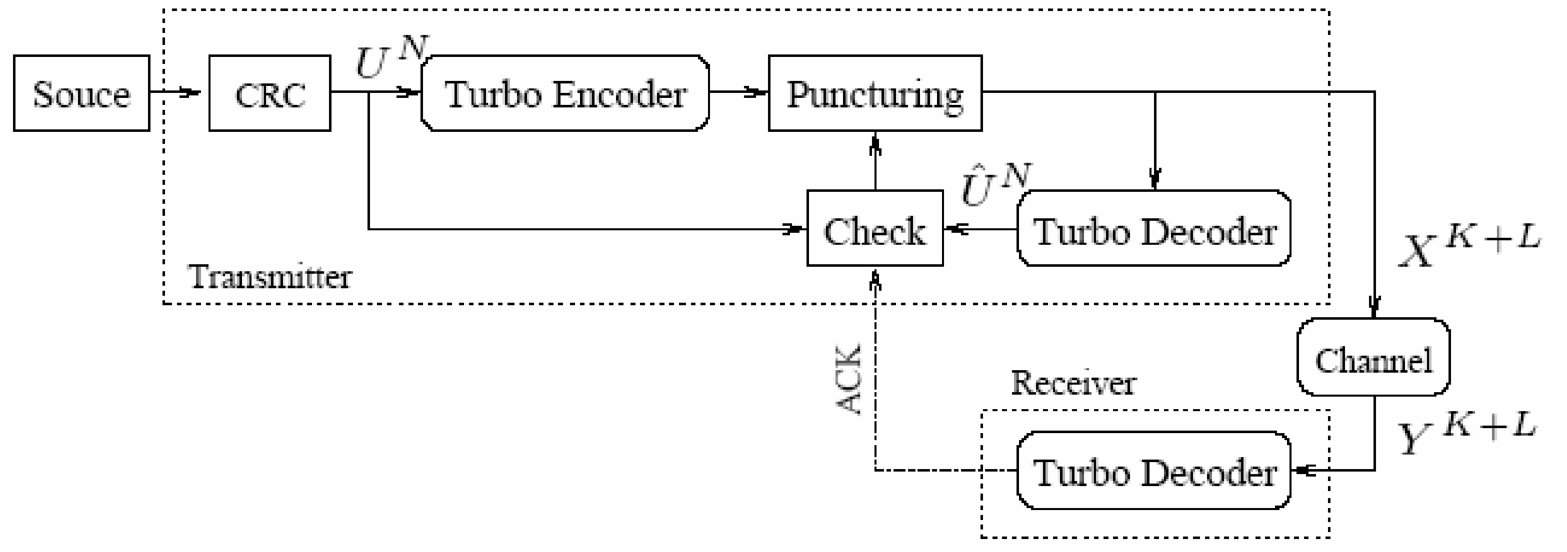
Algorithm

1. Let $i = N_c/2$
2. Encode the source block with a turbo encoder and store the output block
3. Puncture the encoded block using i parity segments
4. Decode the compressed block
5. Check for errors. If the decoded block is error free, let $i = i-1$ and go back to 3.
6. Let $i = i+1$. Repeat 3, include a binary codeword corresponding to the index i, and stop.

Note: Since the decoder on the receiving side knows the block length, the random puncturing matrix, and the interleaver all it requires is the index of the last parity segment that has been erased. At least half of the parity bits are punctured (otherwise we would have no compression at all), so it suffices to use $\lceil log_2(N_C/2) \rceil$ bits as side information to indicate the size of the codeword.

Combining Decremental (DR) and Incremental (IR) redundancy for Joint Source and Channel coding

- The same algorithm can be used for DR and IR
- If channel state information is available at the transmitter, we can add a test channel to the encoding loop that tests the decodability of the encoded block.



Combining Decremental and Incremental redundancy for Joint Source and Channel coding

- As an integrity test a cyclic redundancy check (CRC) word of length N_C is added to the source block U^{N-Nc}

Compressed data X^{K+L} include

  - additional parity block to compensate channel errors
  - punctured source sequence with CRC block
  - index of puncturing matrix

- After decoding, the received data integrity is checked by CRC
- To pass the integrity test we use ARQ Type II protocol with FEC. The index of puncturing matrix is increased until decoding is error free.

Algorithm analysis using EXIT charts

EXIT chart

- predicts the convergence of an iterative decoder without having to simulate the iterative decoder itself
- uses mutual information to parameterize the sequences of L-values being exchanged between decoders and this characterizes the APP decoders

Assume U be a binary source such that $P(U=+1)=p$. Let $f(y|U=u)$ be pdf of the channel with output y.

The mutual information between U and Y stated as

$$I(U;Y)=\sum_{u=\pm 1} p(U=u)\int_{-\infty}^{\infty} f(y\,|\,U=u).\log_2 \frac{f(y\,|\,U=u)}{p.f(y\,|\,U=+1)+(p-1)f(y\,|\,U=-1)}$$

By source ergodicity it can be simplified to

$$I(U;Y)=H_b(p)-E\left\{\log_2\left(1+e^{-u.L(U|Y)}\right)\right\}\cong H_b(p)-\frac{1}{N}\sum_{n=1}^{N}\left(1+e^{-u_n.L(u_n|y_n)}\right)$$

Note: If no information is transmitted $L(U\,|\,Y)=\ln\frac{p}{1-p}$, which result in $I(U|Y)=0$, thus we are using our knowledge of the source statistics.

Algorithm analysis using EXIT charts

- To construct the EXIT charts we consider

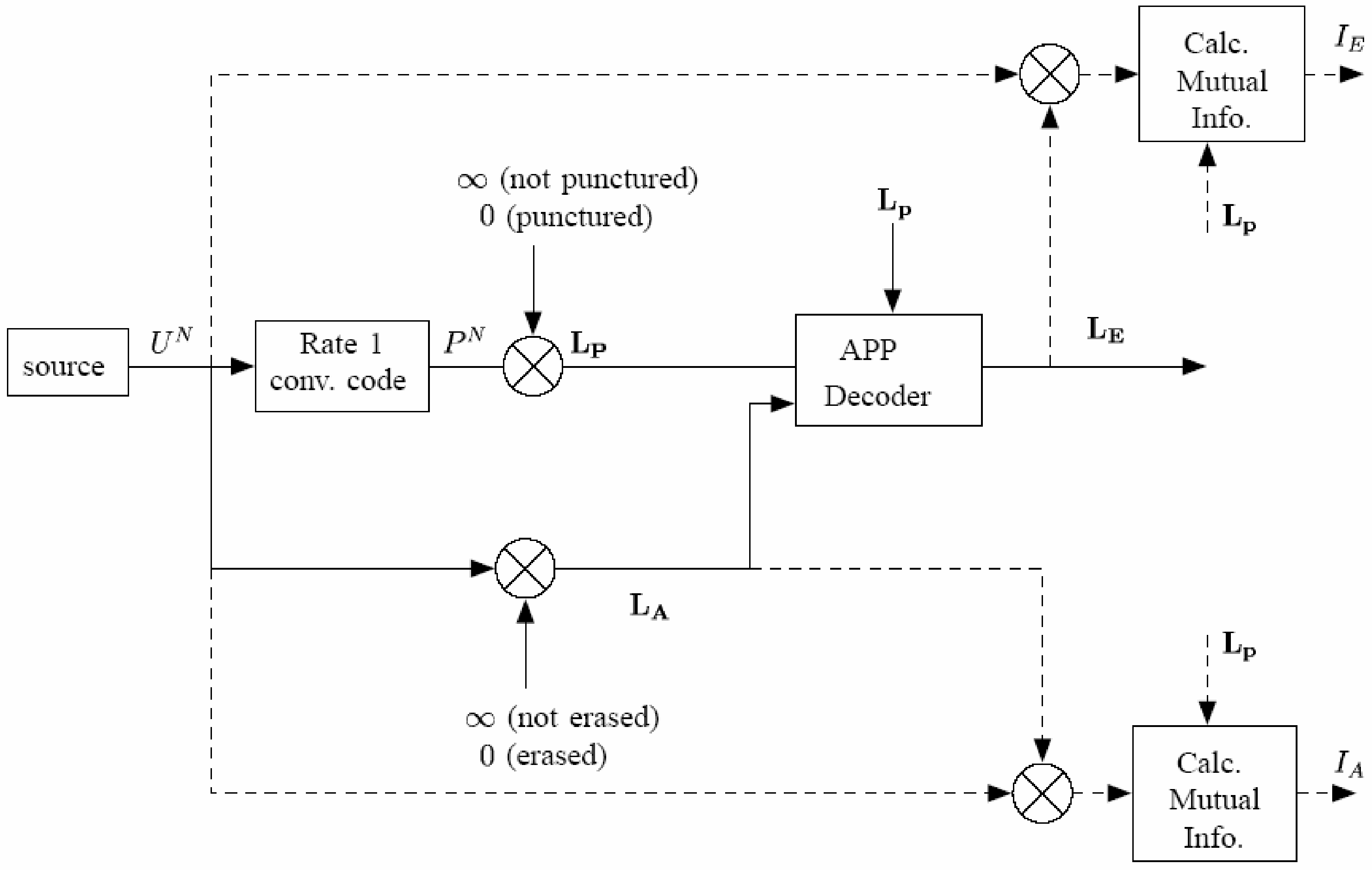
$$I(U;L_{E_1})=I_{E1},\ I(U;L_{A_2})=I_{A2},\ I(U;L_{E_2})=I_{E2},\ I(U;L_{A_1})=I_{A1}$$

as functions $I_{E1}=f_1(I_{A1}), I_{E2}=f_2(I_{A2})$

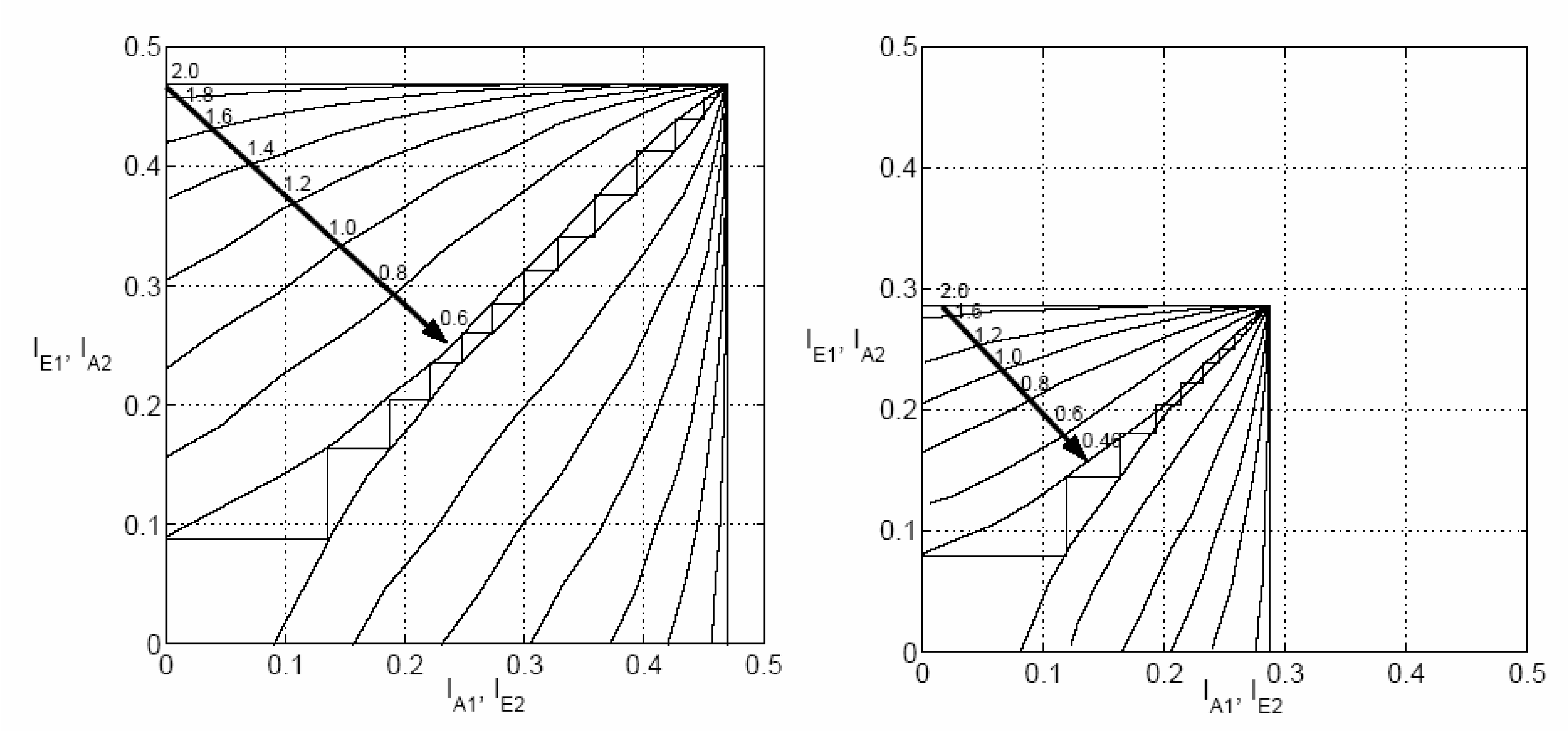
- Convergence can be interpreted as the ability of the turbo decoder to rebuild the source sequence from the compressed sequence.

- EXIT charts construction
  - Starting by noting $I_{E1}=I_{A2}$ and $I_{E2}=I_{A1}$
  - We plot $I_{E1}=f_1(I_{A1})$ against a mirror version of $I_{E2}=f_2(I_{A2})$
  - If a tunnel exits between the two curve the sequence can be successfully decompressed
  - The width of the tunnel can be changed by changing the compression rate

Algorithm analysis using EXIT charts



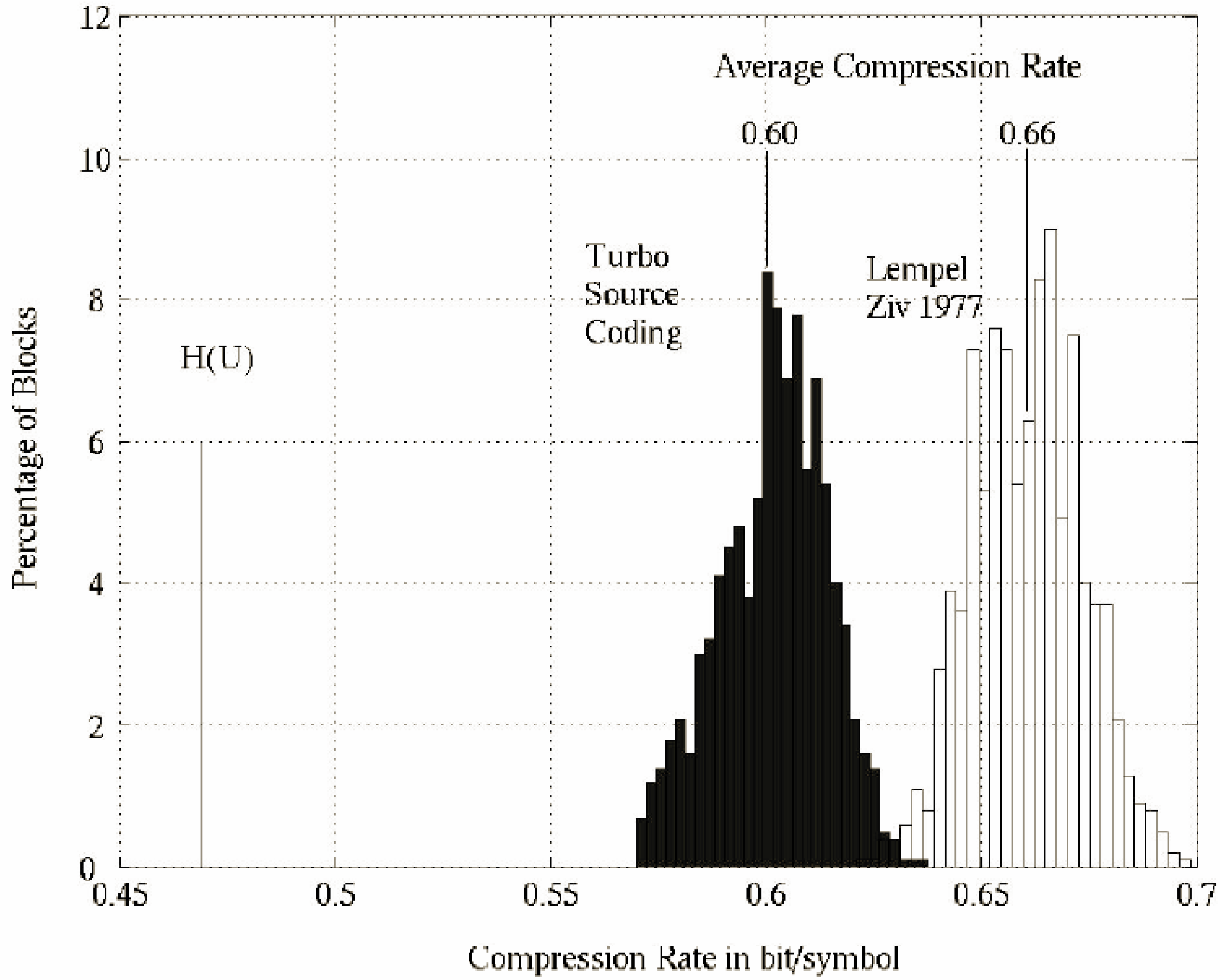
Examples - EXIT charts



- Binary source with block length $N = 9.10^4$; parallel concatenation of two convolution codes with polynomials [7,5]

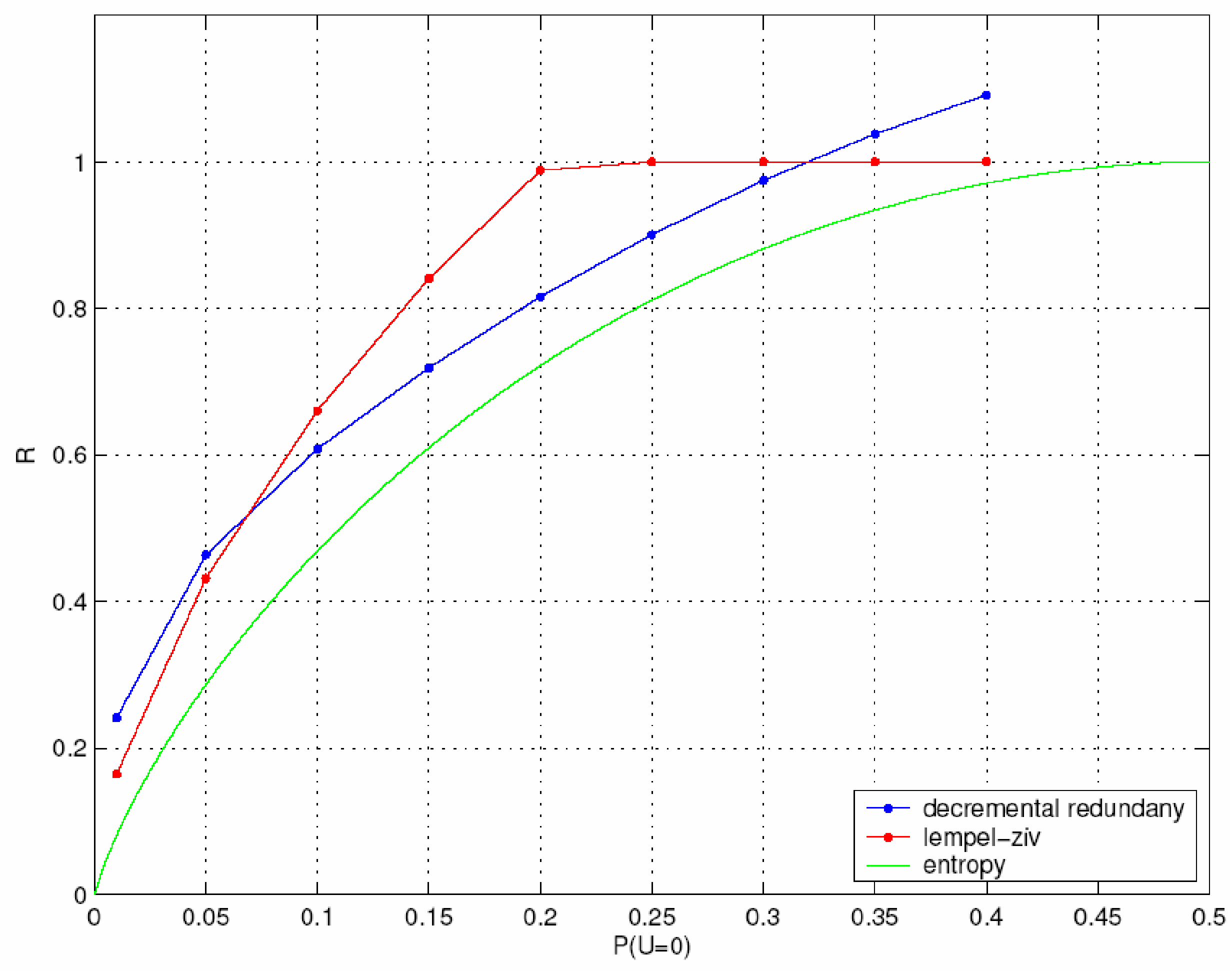
H(U)=0.469 *H(U)=0.286*

Examples – performance of turbo source coding



- Normalized histograms
- Compressing 1000 blocks of $N = 10^4$ from binary source with $H(U)=0.469$ using less than 10 iterations.

Examples – source coding

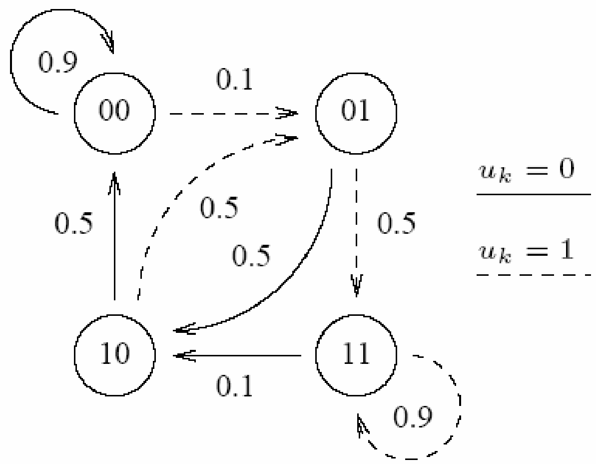


Examples – channel source coding

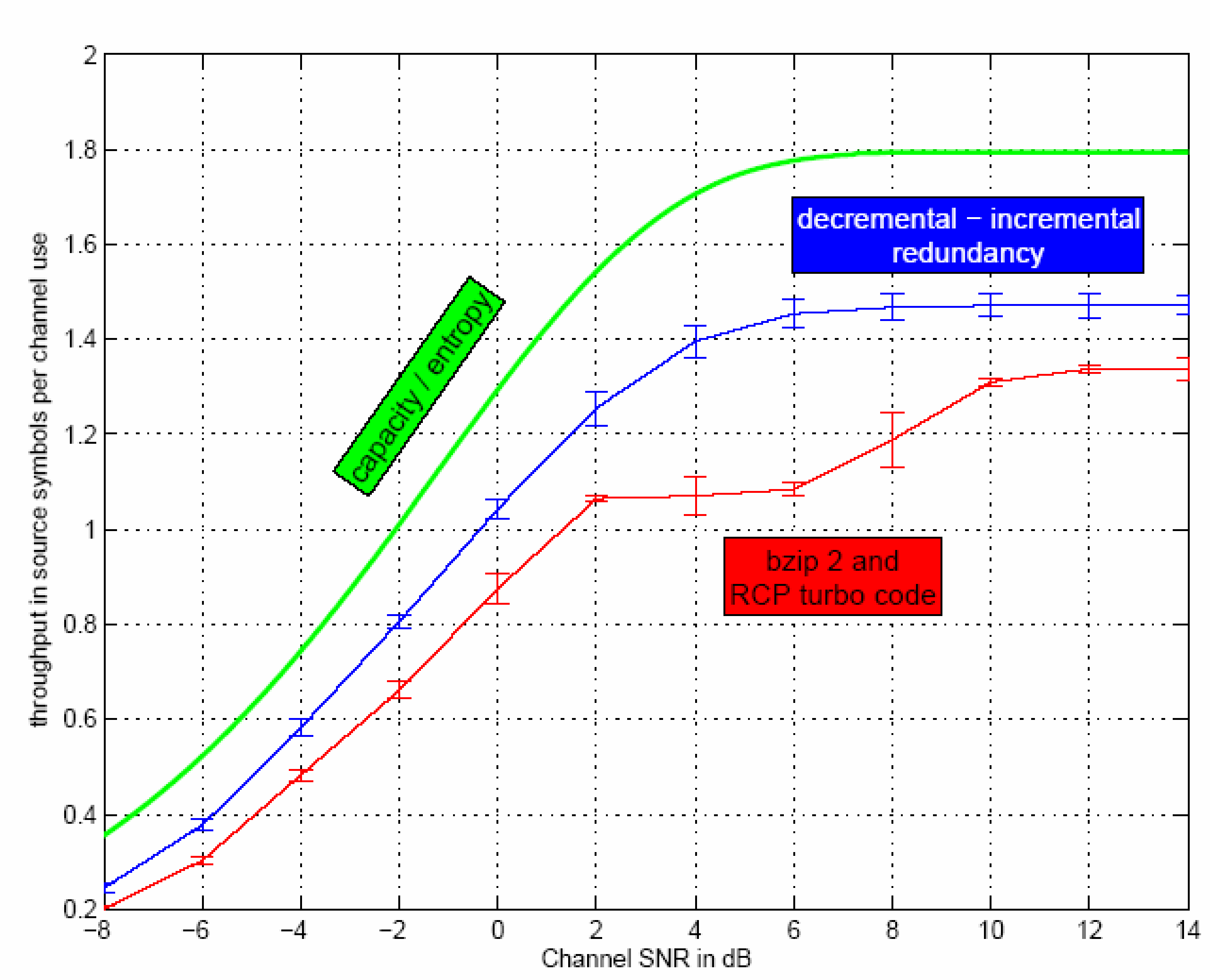
Comparing

- Bzip2 compression + turbo channel coding
  - Source coding (bzip2) - consisting of Burrows Wheeler transform and Huffman coding
  - Channel coding - using PCC turbo code using recursive systematic convolutional code (memory of 2)
  - Rate adjusted by rate-compatible puncturing
- Turbo channel source coding with DR and IR

Encoded 250 blocks, each 2^{16} bits from 4-state bin. Markov source



Examples – channel source coding



Conclusion

- Presented turbo source coding scheme which guarantees lossless recovery of the source information through DR loop in the encoder.
- The convergence of turbo source encoder can be analyzed by modified EXIT chart (tool to optimize codes and puncturing rates)
- By using turbo-coding algorithm for source compression we can achieve compression rates close to entropy
- IR and DR source-channel coding increases the effectiveness of transmission

References

- Hagenauer, J.; Barros, J.; Schaefer, A.: Lossless Turbo Source Coding with Decremental Redundancy. - In: Proc. of the 5th International ITG Conference on Source and Channel Coding (SCC'04), Erlangen, Germany, January 2004, p. 333-340

- Dütsch, N. ; Hagenauer, J.: Combined Incremental and Decremental Redundancy in Joint Source-Channel Coding. - In: Proc. International Symposium on Information Theory and its Applications (ISITA 2004) Parma, Italy, October 2004

- Hagenauer, J.; Dütsch, N.; Barros, J.; Schaefer, A.: Incremental and Decremental Redundancy in Turbo Source-Channel Coding. - In: *1st International Symposium on Control, Communications and Signal Processing, Hammamet, Tunisia*, March 2004, p. 595-598

- Dütsch, N.: Decremental Redundancy in Turbo Source Coding. - In: *13th Joint Conference on Communications and Coding (JCCC 2004), Galtür, Austria*, March 2004