Digital Communications
SSY125, Lecture 12

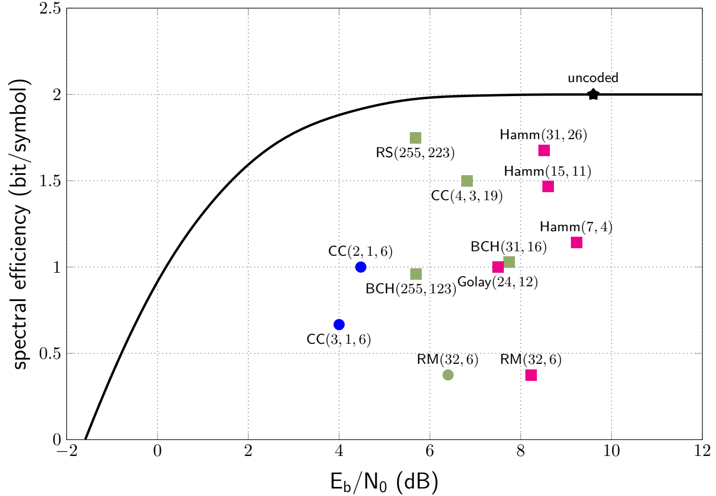
Turbo-Like Codes
(Chapter 10)

Alexandre Graell i Amat
alexandre.graell@chalmers.se
https://sites.google.com/site/agraellamat

November 29, 2023

CHALMERS

1948195819681978



AWGN channel, BPSK/QPSK transmission

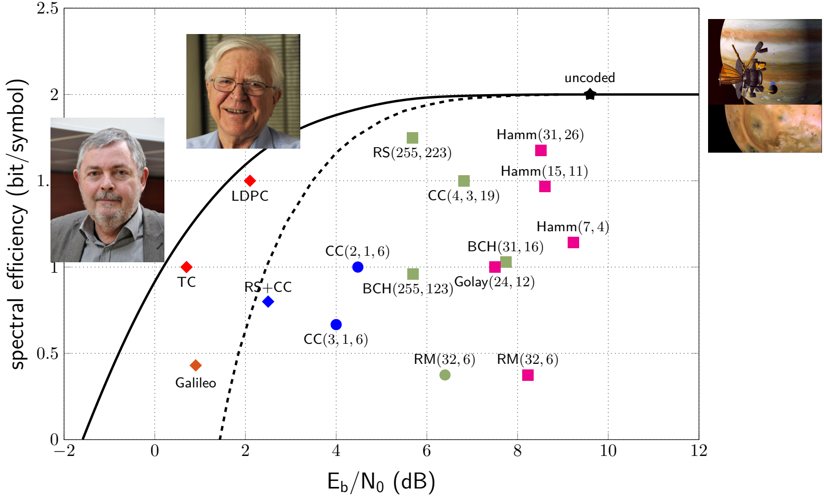
Concatenated Codes

- To approach capacity, large block lengths are required.
- The complexity of block codes and convolutional codes grows exponentially with the block length and the memory of the encoder, respectively.
- Idea: Concatenated codes (David Forney)

$$\boldsymbol{u} \longrightarrow \boxed{\mathcal{C}_{\mathsf{O}}} \xrightarrow{\boldsymbol{c}_{\mathsf{O}}} \boxed{\mathcal{C}_{\mathsf{I}}} \longrightarrow \boldsymbol{c}_{\mathsf{I}}$$

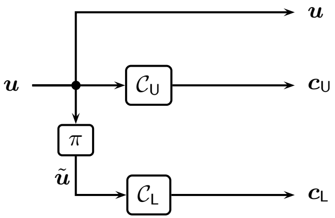
- Typically $\mathcal{C}_{\mathsf{O}}$ is a (nonbinary) block code and $\mathcal{C}_{\mathsf{I}}$ a convolutional code.
- Better performance than standalone codes, but still far from capacity.
- Widely used in deep-space communications (NASA and ESA missions).

197819881998



AWGN channel, BPSK/QPSK transmission

Turbo Codes: Parallel Concatenated Convolutional Codes

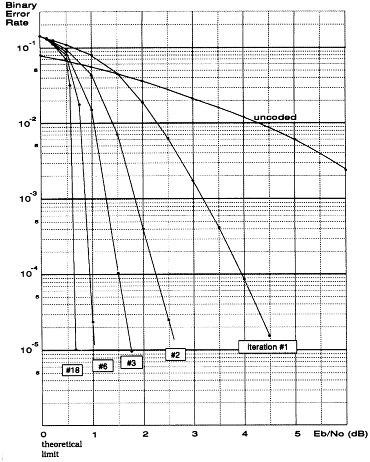


- Parallel concatenation of two recursive, systematic convolutional encoders through a pseudorandom interleaver (Claude Berrou, 1993).
- The encoders are recursive encoders,

$$\boldsymbol{G}_{\mathsf{U}}(\mathsf{D}) = \boldsymbol{G}_{\mathsf{L}}(\mathsf{D}) = \left(\frac{\boldsymbol{g}_1(\mathsf{D})}{\boldsymbol{g}_2(\mathsf{D})}\right).$$

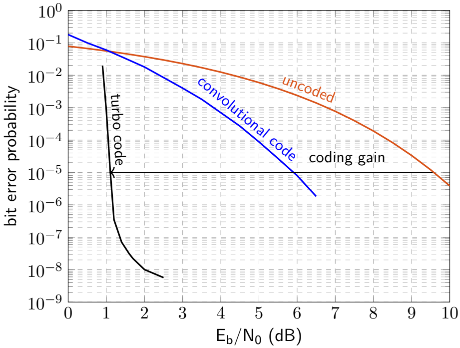
- The codeword is $\boldsymbol{c} = (\boldsymbol{u}, \boldsymbol{c}_{\mathsf{U}}, \boldsymbol{c}_{\mathsf{L}})$, thus the code rate is $R_{\mathsf{c}} = \frac{K}{3K} = \frac{1}{3}$.

The Original Turbo Codes



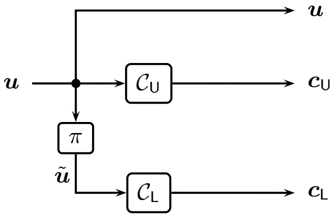
- Original turbo code with component encoders with generator matrix $\boldsymbol{G}(\mathsf{D}) = \left(\frac{1+\mathsf{D}^4}{1+\mathsf{D}+\mathsf{D}^2+\mathsf{D}^3+\mathsf{D}^4}\right)$. $K = 65536$, $R_{\mathsf{c}} = 1/2$.

Typical BER Curve of Turbo Codes



- 8-state component encoders with $\boldsymbol{G}(\mathrm{D}) = \left(\frac{1+\mathrm{D}+\mathrm{D}^3}{1+\mathrm{D}^2+\mathrm{D}^3}\right)$, $R_c = 1/3$, and $K = 1024$ bits. (Turbo code of the 3GPP-LTE standard).
- The performance of aTC is characterized by two well-defined regions:
  - Waterfall region: the BER decreases sharply with E_b/N_0.
  - Error floor region: flattening of the BER curve for medium-to-high E_b/N_0. (Dominated by $d_{\min}$!)

The Need of Recursive Encoders



For feedforward encoders:

- A weight-1 information sequence $\boldsymbol{u}$ will generate a codeword $\boldsymbol{c}_{\mathsf{U}}$ of $\mathcal{C}_{\mathsf{U}}$ of Hamming weight $w_{\mathsf{H}}(\boldsymbol{c}_{\mathsf{U}}) \leq \nu + 1$ (ν is the memory of the encoder).
- The weight of the permuted codeword $\tilde{\boldsymbol{u}}$ is also one.
- $\tilde{\boldsymbol{u}}$ will generate a codeword $\boldsymbol{c}_{\mathsf{L}}$ of $\mathcal{C}_{\mathsf{L}}$ of weight $w_{\mathsf{H}}(\boldsymbol{c}_{\mathsf{L}}) \leq \nu + 1$.
- The minimum distance of the turbo code is thus upperbounded by

$$d_{\mathsf{min}} \leq 1 + 2(\nu + 1).$$

Idea: Recursive Convolutional Encoders

Weight-1 information sequences are not a problem anymore: They generates an infinite weight codeword at the output of each component encoder.

The Need of Recursive Encoders

Rate-$1/3$ TC with 4-state feedforward encoders, $\boldsymbol{G}(\mathsf{D}) = \left(1 + \mathsf{D} + \mathsf{D}^2\right)$

- The codeword at the output of $\mathcal{C}_{\mathsf{U}}$ and $\mathcal{C}_{\mathsf{L}}$ generated by $\boldsymbol{u}(\mathsf{D}) = \mathsf{D}^j$ is

$$\boldsymbol{c}(\mathsf{D}) = \boldsymbol{u}(\mathsf{D})\boldsymbol{G}(\mathsf{D}) = \mathsf{D}^j(1 + \mathsf{D} + \mathsf{D}^2) = \mathsf{D}^j + \mathsf{D}^{j+1} + \mathsf{D}^{j+2},$$

  i.e., $w_{\mathsf{H}}(\boldsymbol{c}) = 3$.

- The codeword of the turbo code has weight $1 + 3 + 3 = 7$, independently of the interleaver size! (the best rate-$1/3$, 4-state convolutional code has minimum distance $d_{\min} = 8$!)

Rate-$1/3$ TC with 4-state RSC encoders, $\boldsymbol{G}(\mathsf{D}) = \left(\frac{1+\mathsf{D}^2}{1+\mathsf{D}+\mathsf{D}^2}\right)$

- The codeword at the output of $\mathcal{C}_{\mathsf{U}}$ and $\mathcal{C}_{\mathsf{L}}$ generated by $\boldsymbol{u}(\mathsf{D}) = \mathsf{D}^j$ is

$$\boldsymbol{c}(\mathsf{D}) = \mathsf{D}^j \frac{1 + \mathsf{D}^2}{1 + \mathsf{D} + \mathsf{D}^2} = \mathsf{D}^j \left(1 + \mathsf{D} + \mathsf{D}^2 + \mathsf{D}^4 + \mathsf{D}^5 + \mathsf{D}^7 + \mathsf{D}^8 + \ldots,\right.$$

  i.e., of infinite weight!

The Role of the Interleaver

- Main role: Ensure a large minimum distance.
- Main idea: If $\boldsymbol{u}$ is such that it produces a codeword $\boldsymbol{c}_{\mathsf{U}}$ of low weight, it should be permuted to $\tilde{\boldsymbol{u}}$ such that it generates a codeword $\boldsymbol{c}_{\mathsf{L}}$ of large weight.
- Special attention must be payed to weight-2 information words $\rightarrow$ tend to yield low-weight codewords at the output of $\mathcal{C}_{\mathsf{U}}$ if the two ones are close to each other.

Good Design Rule

Guarantee that if two input bits of $\boldsymbol{u}$ in positions i and j are within S positions to each other, i.e., $|i-j| \leq S$, then they should be spread further apart in $\tilde{\boldsymbol{u}}$, i.e., $|\pi(i)-\pi(j)| > S$ $\rightarrow$ will likely generate a high-weight codeword $\boldsymbol{c}_{\mathsf{L}}$.

Decoding Turbo Codes: Iterative (Turbo) Decoding

- Optimum decoding rule (MAP rule),

$$\hat{u}_i = \arg\max_{u_i} p(u_i|\boldsymbol{y}).$$

- Goal: Compute the a posteriori probabilities $P_{\mathsf{APP}}(u_i|\boldsymbol{y}) \triangleq p(u_i|\boldsymbol{y})$ based on the received (noisy) sequence $\boldsymbol{y} = (\boldsymbol{y}^{\boldsymbol{u}}, \boldsymbol{y}^{\boldsymbol{c}_{\mathsf{U}}}, \boldsymbol{y}^{\boldsymbol{c}_{\mathsf{L}}})$. Decision rule:

$$\hat{u}_i = \begin{cases} 1 & \text{if } P_{\mathsf{APP}}(u_i = 1|\boldsymbol{y}) > P_{\mathsf{APP}}(u_i = 0|\boldsymbol{y}) \\ 0 & \text{if } P_{\mathsf{APP}}(u_i = 0|\boldsymbol{y}) < P_{\mathsf{APP}}(u_i = 1|\boldsymbol{y}) \end{cases}.$$

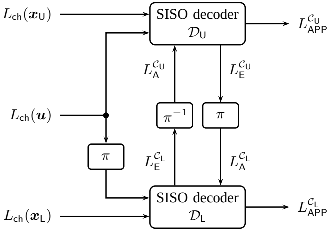
- Typically, decoders work with so-called log-likelihood ratios (LLRs),

$$L_{\mathsf{APP}}(u_i|\boldsymbol{y}) \triangleq \ln \frac{P_{\mathsf{APP}}(u_i = 0|\boldsymbol{y})}{P_{\mathsf{APP}}(u_i = 1|\boldsymbol{y})}.$$

Then, the decision rule is

$$\hat{u}_i = \begin{cases} 1 & \text{if } L_{\mathsf{APP}}(u_i|\boldsymbol{y}) < 0 \\ 0 & \text{if } L_{\mathsf{APP}}(u_i|\boldsymbol{y}) > 0 \end{cases}.$$

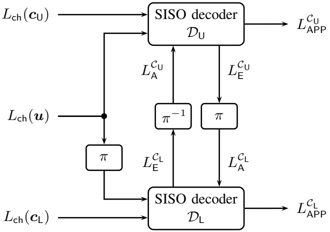
Iterative (Turbo) Decoding



MAP decoding of turbo codes is unfeasible!

- Turbo decoding: A low-complexity, suboptimal iterative decoding algorithm to compute $L_{\mathsf{APP}}(u_i|\boldsymbol{y})$ approximately.
- Two soft-input soft-output (SISO) decoders (matched to the two encoders) exchange information about the reliability of their estimates (soft information) iteratively.

Iterative (Turbo) Decoding



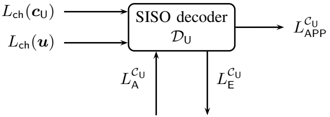
- Each SISO decoder performs MAP decoding of the corresponding component encoder. They compute the log-APPs

$$L_{\mathsf{APP}}^{\mathcal{C}_{\mathsf{U}}}(u_i|\boldsymbol{y}^{\boldsymbol{u}},\boldsymbol{y}^{\boldsymbol{c}_{\mathsf{U}}}) = \ln\frac{P_{\mathsf{APP}}(u_i=0|\boldsymbol{y}^{\boldsymbol{u}},\boldsymbol{y}^{\boldsymbol{c}_{\mathsf{U}}})}{P_{\mathsf{APP}}(u_i=1|\boldsymbol{y}^{\boldsymbol{u}},\boldsymbol{y}^{\boldsymbol{c}_{\mathsf{U}}})}$$

$$L_{\mathsf{APP}}^{\mathcal{C}_{\mathsf{L}}}(u_i|\boldsymbol{y}^{\boldsymbol{u}},\boldsymbol{y}^{\boldsymbol{c}_{\mathsf{L}}}) = \ln\frac{P_{\mathsf{APP}}(u_i=0|\boldsymbol{y}^{\boldsymbol{u}},\boldsymbol{y}^{\boldsymbol{c}_{\mathsf{L}}})}{P_{\mathsf{APP}}(u_i=1|\boldsymbol{y}^{\boldsymbol{u}},\boldsymbol{y}^{\boldsymbol{c}_{\mathsf{L}}})}.$$

- The two decoders work with different channel observations.

The Soft-Input Soft-Output Decoder



The SISO decoder (decoder $\mathcal{D}_{\mathsf{U}}$) has three inputs:

- The channel LLRs of the information bits $\boldsymbol{u}$,

$$L_{\mathsf{ch}}(y_i^u|u_i) = \ln \frac{P(y_i^u|u_i = 0)}{P(y_i^u|u_i = 1)}.$$

- The channel LLRs of the bits of codeword $\boldsymbol{c}_{\mathsf{U}}$,

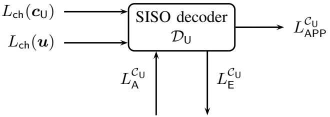
$$L_{\mathsf{ch}}(y_i^{c_{\mathsf{u}}}|c_{\mathsf{u},i}) = \ln \frac{P(y_i^{c_{\mathsf{u}}}|c_{\mathsf{u},i} = 0)}{P(y_i^{c_{\mathsf{u}}}|c_{\mathsf{u},i} = 1)}.$$

- The a priori information on the information bits,

$$L_{\mathsf{A}}^{\mathcal{C}_{\mathsf{U}}}(u_i) = \ln \frac{P_{\mathsf{A}}^{\mathcal{C}_{\mathsf{U}}}(u_i = 0)}{P_{\mathsf{A}}^{\mathcal{C}_{\mathsf{U}}}(u_i = 1)},$$

  provided by the companion decoder $\mathcal{D}_{\mathsf{L}}$.

The Soft-Input Soft-Output Decoder



The SISO decoder (decoder $\mathcal{D}_{\mathsf{U}}$) has two outputs:

- The log-APPs of the information bits,

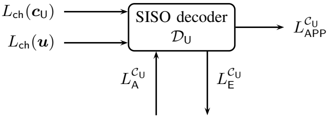
$$L_{\mathsf{APP}}^{\mathcal{C}_{\mathsf{U}}}(u_i|\boldsymbol{y}^{\boldsymbol{u}},\boldsymbol{y}^{\boldsymbol{c}_{\mathsf{U}}}) = \ln \frac{P_{\mathsf{APP}}^{\mathcal{C}_{\mathsf{U}}}(u_i=0|\boldsymbol{y}^{\boldsymbol{u}},\boldsymbol{y}^{\boldsymbol{c}_{\mathsf{U}}})}{P_{\mathsf{APP}}^{\mathcal{C}_{\mathsf{U}}}(u_i=1|\boldsymbol{y}^{\boldsymbol{u}},\boldsymbol{y}^{\boldsymbol{c}_{\mathsf{U}}})}.$$

Applying Bayes', it can be rewritten as

$$\begin{aligned} L_{\mathsf{APP}}^{\mathcal{C}_{\mathsf{U}}}(u_i|\boldsymbol{y}^{\boldsymbol{u}},\boldsymbol{y}^{\boldsymbol{c}_{\mathsf{U}}}) &= \ln \frac{P(\boldsymbol{y}^{\boldsymbol{u}},\boldsymbol{y}^{\boldsymbol{c}_{\mathsf{U}}}|u_i=0)}{P(\boldsymbol{y}^{\boldsymbol{u}},\boldsymbol{y}^{\boldsymbol{c}_{\mathsf{U}}}|u_i=1)} + \ln \frac{P(u_i=0)}{P(u_i=1)} \\ &= \ln \frac{P^{\mathcal{C}_{\mathsf{U}}}(\boldsymbol{y}^{\boldsymbol{u}},\boldsymbol{y}^{\boldsymbol{c}_{\mathsf{U}}}|u_i=0)}{P^{\mathcal{C}_{\mathsf{U}}}(\boldsymbol{y}^{\boldsymbol{u}},\boldsymbol{y}^{\boldsymbol{c}_{\mathsf{U}}}|u_i=1)} + \ln \frac{P_{\mathsf{A}}^{\mathcal{C}_{\mathsf{U}}}(u_i=0)}{P_{\mathsf{A}}^{\mathcal{C}_{\mathsf{U}}}(u_i=1)}. \end{aligned}$$

The second term is the a priori information on $\boldsymbol{u}$ provided by $\mathcal{D}_{\mathsf{L}}$.

The Soft-Input Soft-Output Decoder



The SISO decoder (decoder $\mathcal{D}_{\mathsf{U}}$) has two outputs (cont'd):

- The extrinsic information,

$$L_{\mathsf{E}}^{\mathcal{C}_{\mathsf{U}}}(u_i) = L_{\mathsf{APP}}^{\mathcal{C}_{\mathsf{U}}}(u_i) - L_{\mathsf{A}}^{\mathcal{C}_{\mathsf{U}}}(u_i) - L_{\mathsf{ch}}(u_i),$$

  obtained by removing the a priori knowledge that $\mathcal{D}_{\mathsf{U}}$ has about the bit, $L_{\mathsf{A}}^{\mathcal{C}_{\mathsf{U}}}(u_i)$, and the channel observation $L_{\mathsf{ch}}(u_i)$ from $L_{\mathsf{APP}}^{\mathcal{C}_{\mathsf{U}}}(u_i)$.
- This extrinsic information (after interleaving) will be used by decoder $\mathcal{D}_{\mathsf{L}}$ as a priori information,

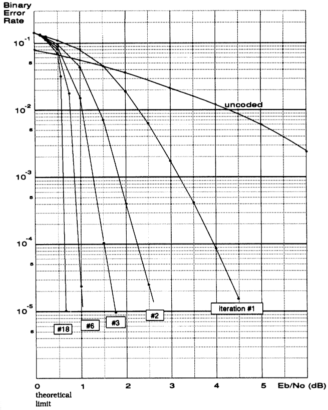
$$L_{\mathsf{A}}^{\mathcal{C}_{\mathsf{L}}}(\tilde{u}_i) = L_{\mathsf{E}}^{\mathcal{C}_{\mathsf{U}}}(\pi^{-1}(\tilde{u}_i)).$$

Iterative (Turbo) Decoding

1. Iteration 1.
   1.1 Decode $\mathcal{C}_{\mathsf{U}}$ running $\mathcal{D}_{\mathsf{U}}$, with inputs $L_{\mathsf{ch}}(\boldsymbol{u})$ and $L_{\mathsf{ch}}(\boldsymbol{c}_{\mathsf{U}})$. $L_{\mathsf{A}}^{\mathcal{C}_{\mathsf{U}},(1)}(\boldsymbol{u})$ is set to zero. The decoder outputs are $L_{\mathsf{APP}}^{\mathcal{C}_{\mathsf{U}},(1)}(\boldsymbol{u})$ and $L_{\mathsf{E}}^{\mathcal{C}_{\mathsf{U}},(1)}(\boldsymbol{u})$.
   1.2 Decode $\mathcal{C}_{\mathsf{L}}$ running $\mathcal{D}_{\mathsf{L}}$, with inputs $L_{\mathsf{ch}}(\tilde{\boldsymbol{u}})$, $L_{\mathsf{ch}}(\boldsymbol{c}_{\mathsf{L}})$, and $L_{\mathsf{A}}^{\mathcal{C}_{\mathsf{L}},(1)}(\tilde{\boldsymbol{u}}) = L_{\mathsf{E}}^{\mathcal{C}_{\mathsf{U}},(1)}(\pi^{-1}(\tilde{\boldsymbol{u}}))$.
2. Iteration $\ell > 1$.
   2.1 Decode $\mathcal{C}_{\mathsf{U}}$ running $\mathcal{D}_{\mathsf{U}}$, with inputs $L_{\mathsf{ch}}(\boldsymbol{u})$, $L_{\mathsf{ch}}(\boldsymbol{c}_{\mathsf{U}})$, and $L_{\mathsf{A}}^{\mathcal{C}_{\mathsf{U}},(\ell)}(\boldsymbol{u}) = L_{\mathsf{E}}^{\mathcal{C}_{\mathsf{L}},(\ell-1)}(\pi(\boldsymbol{u}))$. The decoder outputs are $L_{\mathsf{APP}}^{\mathcal{C}_{\mathsf{U}},(\ell)}(\boldsymbol{u})$ and $L_{\mathsf{E}}^{\mathcal{C}_{\mathsf{U}},(\ell)}(\boldsymbol{u})$.
   2.2 Decode $\mathcal{C}_{\mathsf{L}}$ running $\mathcal{D}_{\mathsf{L}}$, with inputs $L_{\mathsf{ch}}(\tilde{\boldsymbol{u}})$, $L_{\mathsf{ch}}(\boldsymbol{c}_{\mathsf{L}})$, and $L_{\mathsf{A}}^{\mathcal{C}_{\mathsf{L}},(\ell)}(\tilde{\boldsymbol{u}}) = L_{\mathsf{E}}^{\mathcal{C}_{\mathsf{U}},(\ell)}(\pi^{-1}(\tilde{\boldsymbol{u}}))$.
3. Repeat Step 2 until a maximum number of iterations ℓ_{max} is reached. Then make decisions on the bits u_i according to

$$\hat{u}_i = \begin{cases} 1 & \text{if } L_{\mathsf{APP}}^{\mathcal{C}_{\mathsf{U}},(\ell_{\mathsf{max}})}(u_i) < 0 \\ 0 & \text{if } L_{\mathsf{APP}}^{\mathcal{C}_{\mathsf{U}},(\ell_{\mathsf{max}})}(u_i) > 0 \end{cases}.$$

Performance of Turbo Codes



- Beyond a number of iterations there is a marginal gain: The decisions of the two decoders become too correlated so there is not much more to gain by running further iterations!
- Typically, around $8-10$ iterations are enough to fully exploit the potential of a turbo code.

The Use of Extrinsic Information

Why the component decoders exchange extrinsic information and not APPs?

- The APP generated by $\mathcal{D}_{\mathsf{L}}$ on bit u_i is computed based on the channel observations $L_{\mathsf{ch}}(\boldsymbol{u})$ and $L_{\mathsf{ch}}(\boldsymbol{c}_{\mathsf{L}})$, as well as on soft information generated by $\mathcal{D}_{\mathsf{U}}$.
- But...the APPs generated by $\mathcal{D}_{\mathsf{U}}$ also use $L_{\mathsf{ch}}(\boldsymbol{u})$. Since decoder $\mathcal{D}_{\mathsf{L}}$ is already fed with $L_{\mathsf{ch}}(\boldsymbol{u})$ directly, the soft information that $\mathcal{D}_{\mathsf{U}}$ passes to $\mathcal{D}_{\mathsf{L}}$ should not contain $L_{\mathsf{ch}}(\boldsymbol{u})$. Thus, should consider passing

$$L_{\mathsf{APP}}^{\mathcal{C}_{\mathsf{U}}}(u_i) - L_{\mathsf{ch}}(u_i).$$

- Want to pass truly a priori (independent) information to $\mathcal{D}_{\mathsf{L}}$. But $L_{\mathsf{APP}}^{\mathcal{C}_{\mathsf{U}}}(u_i) - L_{\mathsf{ch}}(u_i)$ includes data from $\mathcal{D}_{\mathsf{L}}$ itself: The a priori information that $\mathcal{D}_{\mathsf{L}}$ passes to $\mathcal{D}_{\mathsf{U}}$! Therefore, $\mathcal{D}_{\mathsf{U}}$ should pass to $\mathcal{D}_{\mathsf{L}}$

$$L_{\mathsf{APP}}^{\mathcal{C}_{\mathsf{U}}}(u_i) - L_{\mathsf{A}}^{\mathcal{C}_{\mathsf{U}}}(u_i) - L_{\mathsf{ch}}(u_i),$$

  i.e., extrinsic information!

The Rationale Behind Turbo Codes

Shannon:

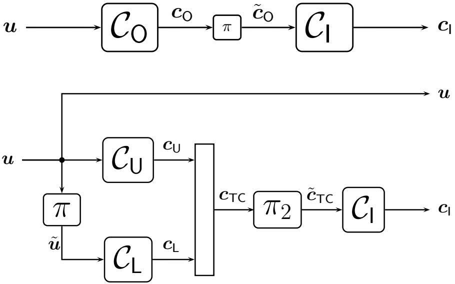
- To achieve capacity very large (infinite, indeed) block lengths are required.
- Shannon's proof of the channel coding theorem uses a random coding argument.

Unfortunately...decoding complexity increases exponentially with block length and random codes are not decodable in practice.

Turbo codes' response:

- Randomness. Make the code appear random while maintaining enough structure to permit decoding: pseudo-random interleaver!
- Decoding complexity. Powerful code that can be decoded in practice by breaking the decoding into simpler steps: turbo decoding.

Other Code Constructions: Turbo-Like Codes



- Other concatenations of convolutional codes through random interleavers are possible: turbo-like codes.
  - Serially concatenated codes [Benedetto, Montorsi, '96]
  - Hybrid concatenated codes [Divsalar, Pollara '97], [Berrou, GiA, Mouhamedou, Saouter '07]