Analysis and design of Min-Sum-based decoders running on noisy hardware

Christiane Kameni-Ngassa[*], Valentin Savin[*], David Declercq[#]

[*] CEA-LETI, MINATEC campus, Grenoble, France

[#] ETIS ENSEA, Université de Cergy-Pontoise, France

i-RISC Workshop, Bucharest, September 20, 2013.

Innovative Reliable Chip Designs from Low-Powered Unreliable Components

Research reported in this presentation was supported by the Seventh Framework Programme of the European Union, under Grant Agreement number 309129 (**i-RISC** project)

Context & Objective

- Objective
  - Design fault tolerant solutions for LDPC decoders operating on circuits built out from unreliable (faulty) components
  - Can **MP decoders** provide reliable error protection when they operate on faulty devices?

Linear codes

$$H = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \end{matrix} \\ \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array} \quad \begin{array}{ll} c_1: & x_1 + x_2 + x_4 + x_7 = 0 \\ c_2: & x_1 + x_3 + x_5 + x_8 = 0 \\ c_3: & x_2 + x_3 + x_6 + x_9 = 0 \\ c_4: & x_4 + x_5 + x_6 + x_{10} = 0 \\ c_5: & x_7 + x_8 + x_9 + x_{10} = 0 \end{array}$$

- Linear code is defined by a parity-check matrix H
- Codeword: vector $(x_1, \ldots, x_N)$ such that $H \times (x_1, \ldots, x_N)^{\mathrm{T}} = 0$

LDPC code

- A linear code defined by a **sparse** parity-check matrix
  - The number of non-zeo entries per row/column is upper-bounded by a positive constant (even if the size of the matrix goes to infinity)

LDPC codes

$$\begin{array}{cccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} \end{array}$$

$$\begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$c_1: \quad x_1 + x_2 + x_4 + x_7 = 0$$

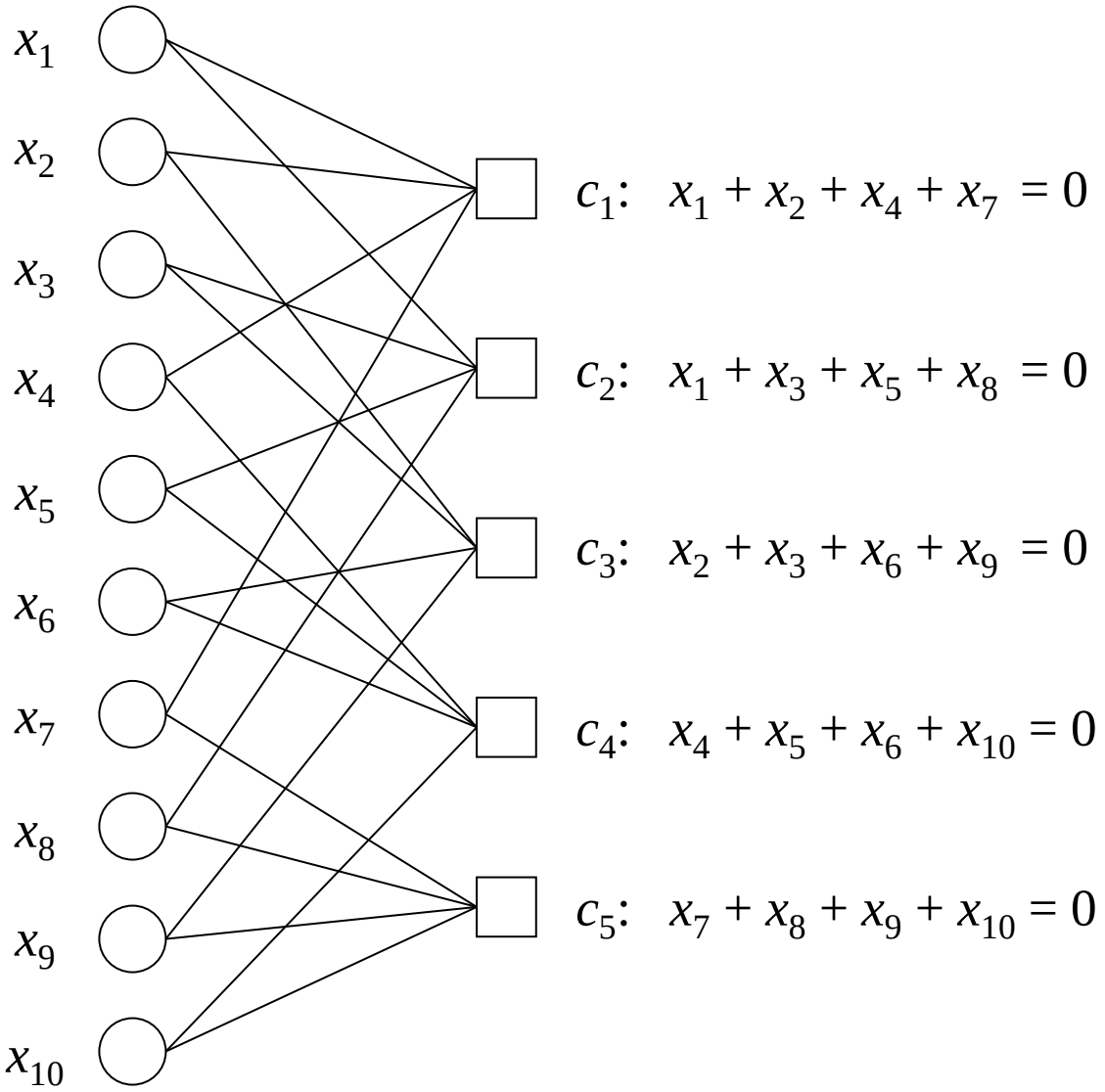
$$c_2: \quad x_1 + x_3 + x_5 + x_8 = 0$$

$$c_3: \quad x_2 + x_3 + x_6 + x_9 = 0$$

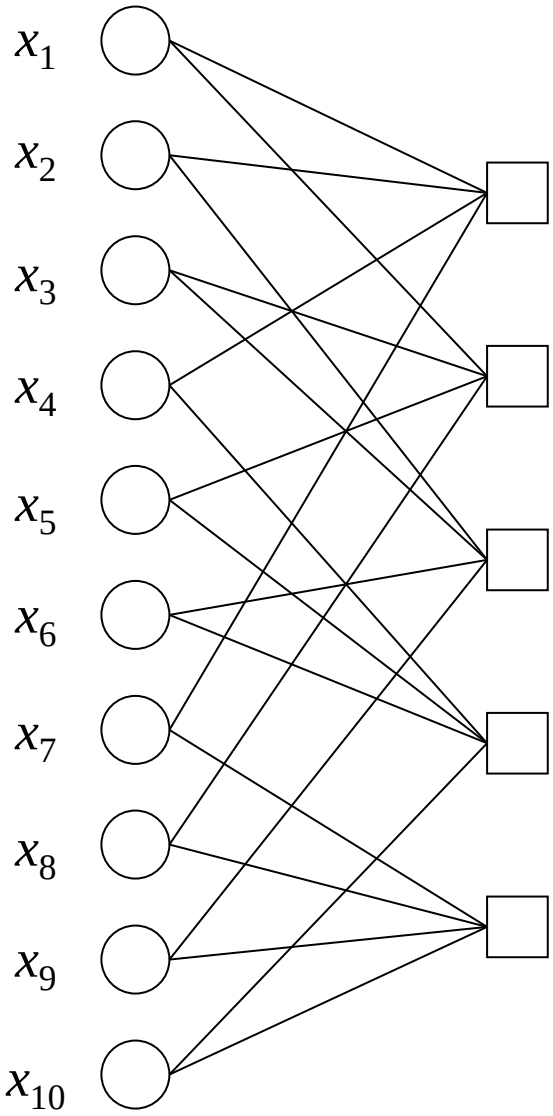
$$c_4: \quad x_4 + x_5 + x_6 + x_{10} = 0$$

$$c_5: \quad x_7 + x_8 + x_9 + x_{10} = 0$$

LDPC codes



LDPC codes



c_1: $x_1 + x_2 + x_4 + x_7 = 0$

c_2: $x_1 + x_3 + x_5 + x_8 = 0$

c_3: $x_2 + x_3 + x_6 + x_9 = 0$

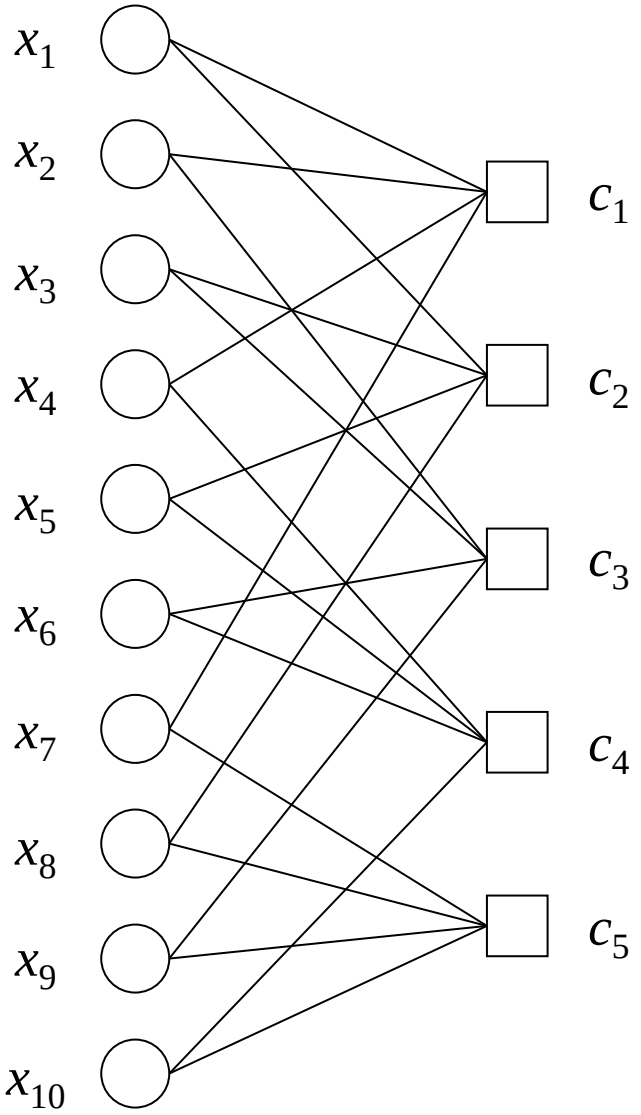
c_4: $x_4 + x_5 + x_6 + x_{10} = 0$

c_5: $x_7 + x_8 + x_9 + x_{10} = 0$

LDPC codes

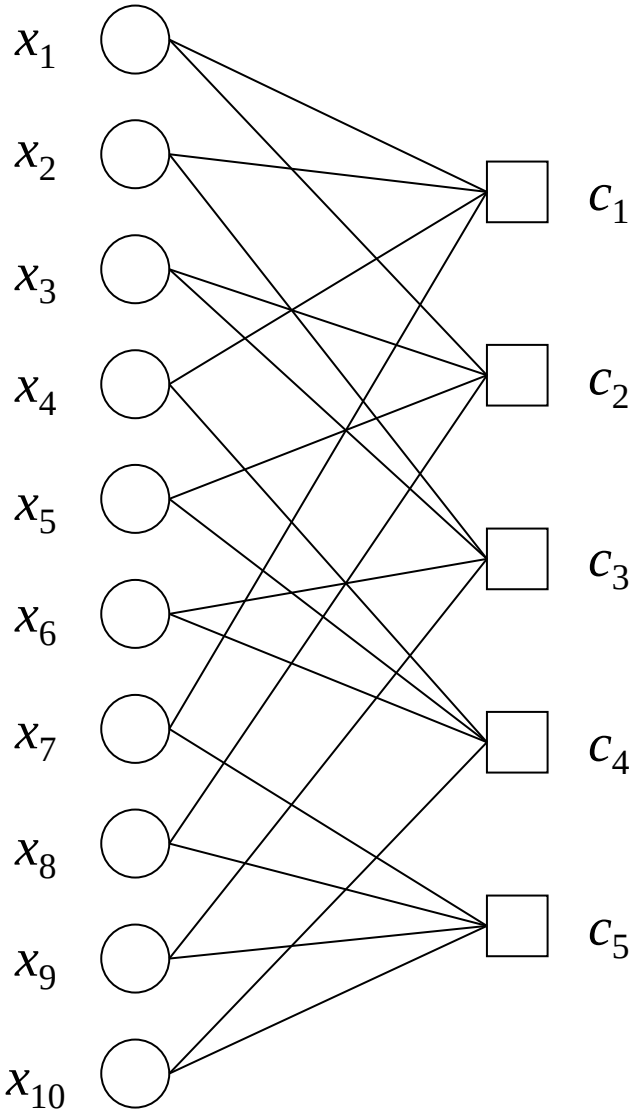
- *H* is advantageously represented by a bipartite (Tanner) graph
  - circles: variable-nodes (or bit-nodes) representing coded bits (columns of *H*)
  - squares: check-nodes representing or parity-check equations (rows of *H*)
  - edges correspond to non-zero entries of *H*

Rather than a family of codes, ***Gallager invented a new method of decoding linear codes***, by using iterative message-passing (MP) algorithms



LDPC decoding

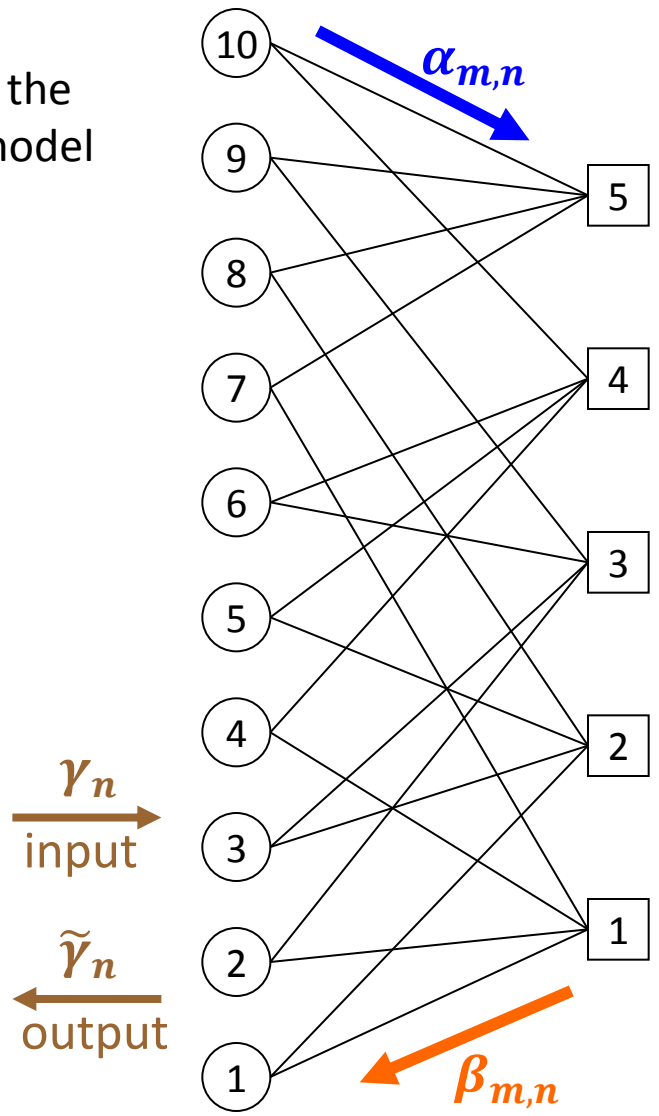
- Iterative MP decoders
  - exchange of ***extrinsic information*** (messages) between bit-nodes and check-nodes
  - each message provides an estimation (in the form of a probability or LLR value) of either the sender or the recipient coded bit
  - takes place in several rounds, or iterations



Message-Passing (MP) decoders

- Input information (usually LLR value):
  - γ_n: computed acc. to the *"channel"* output and the information we have about the channel noise model
- Extrinsic-information exchange
  - $\beta_{m,n} = \text{funct}\left(\alpha_{m,n'};\ n' \in H(m)\backslash n\right)$
  - $\alpha_{m,n} = \text{funct}\left(\gamma_n, \beta_{m',n};\ m' \in H(n)\backslash m\right)$
- A posteriori information (updated LLR value)
  - $\tilde{\gamma}_n = \text{funct}\left(\gamma_n, \beta_{m,n}; m \in H(n)\right)$
  - hard-decision is taken based on $\tilde{\gamma}_n$ values

Remark: γ_n = input LLR; $\tilde{\gamma}_n$ = output (AP-)LLR



Message-Passing (MP) decoders

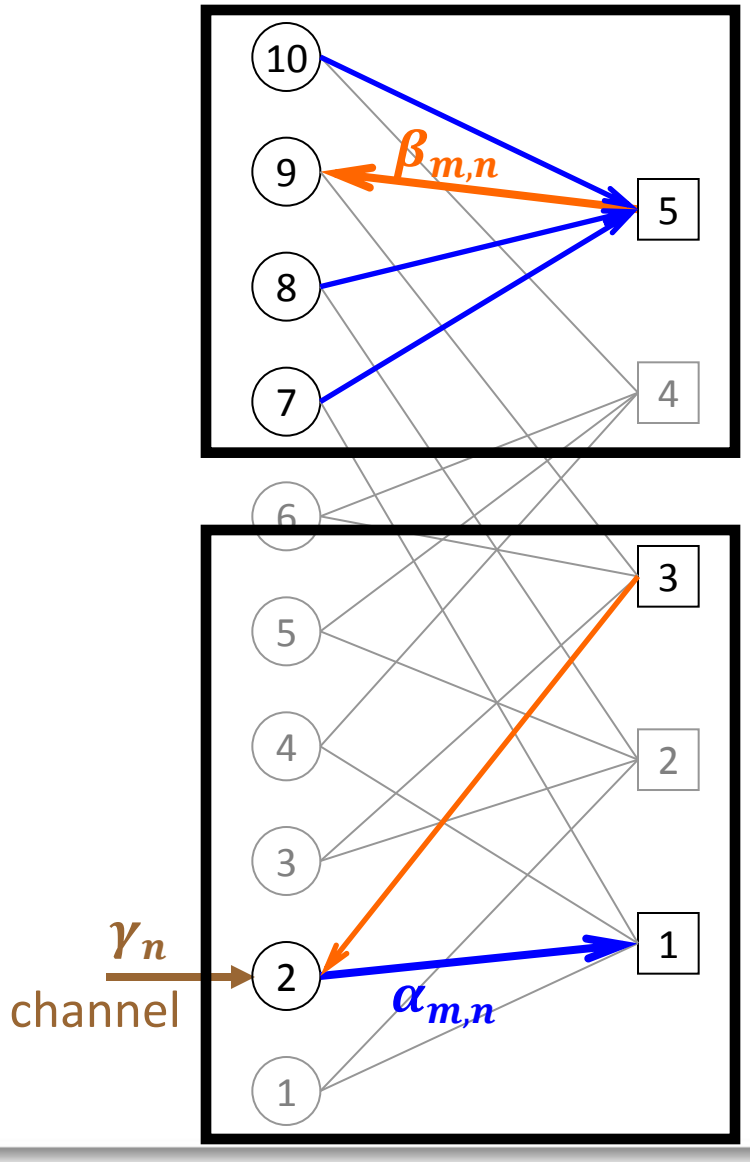
- Extrinsic-information exchange

Check-Node Messages

- $\beta_{m,n} = \text{funct}\left(\alpha_{m,n'};\ n' \in H(m)\backslash n\right)$

Bit-Node Messages

- $\alpha_{m,n} = \text{funct}\left(\gamma_n, \beta_{m',n};\ m' \in H(n)\backslash m\right)$



Message-Passing (MP) decoders

- Several MP decoders, distinguished by the formulae used to compute check and variable-node messages
  - Gallager A, Gallager-B (Majority-Voting)
  - Belief-Propagation (or Sum-Product)
  - Min-Sum

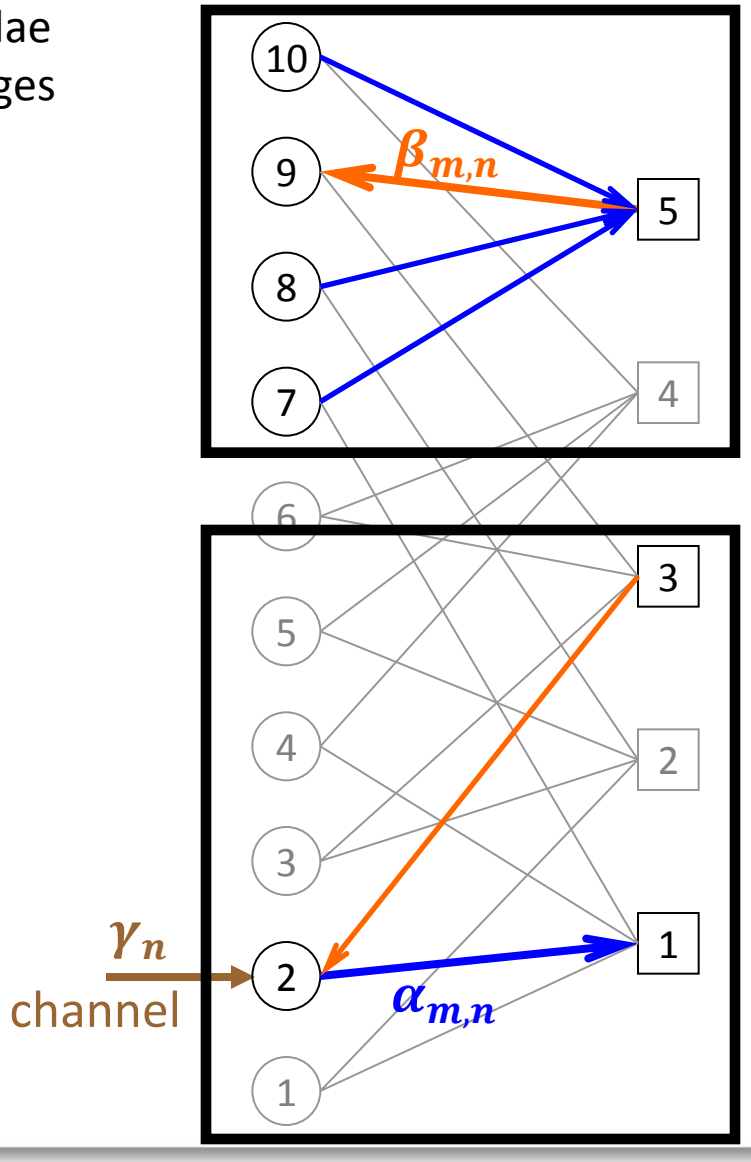
- **Belief-Propagation decoding**

$$\beta_{m,n} = \left(\prod_{n' \in H(m) \setminus n} \operatorname{sgn}(\alpha_{m,n'}) \right) \phi \left(\sum_{n' \in H(m) \setminus n} \phi(|\alpha_{m,n'}|) \right)$$

where $\phi(x) = \log\left(\frac{e^x+1}{e^x-1}\right)$

$$\alpha_{m,n} = \gamma_n + \sum_{m' \in H(n) \setminus m} \beta_{m',n}$$

The outgoing $\alpha_{m,n}$ message is the sum on channel LLR and incoming $\beta_{m,n}$ messages



Message-Passing (MP) decoders

- Several MP decoders, distinguished by the formulae used to compute check and variable-node messages
  - Gallager A, Gallager-B (Majority-Voting)
  - Belief-Propagation (or Sum-Product)
  - Min-Sum

- **Min-Sum decoding**

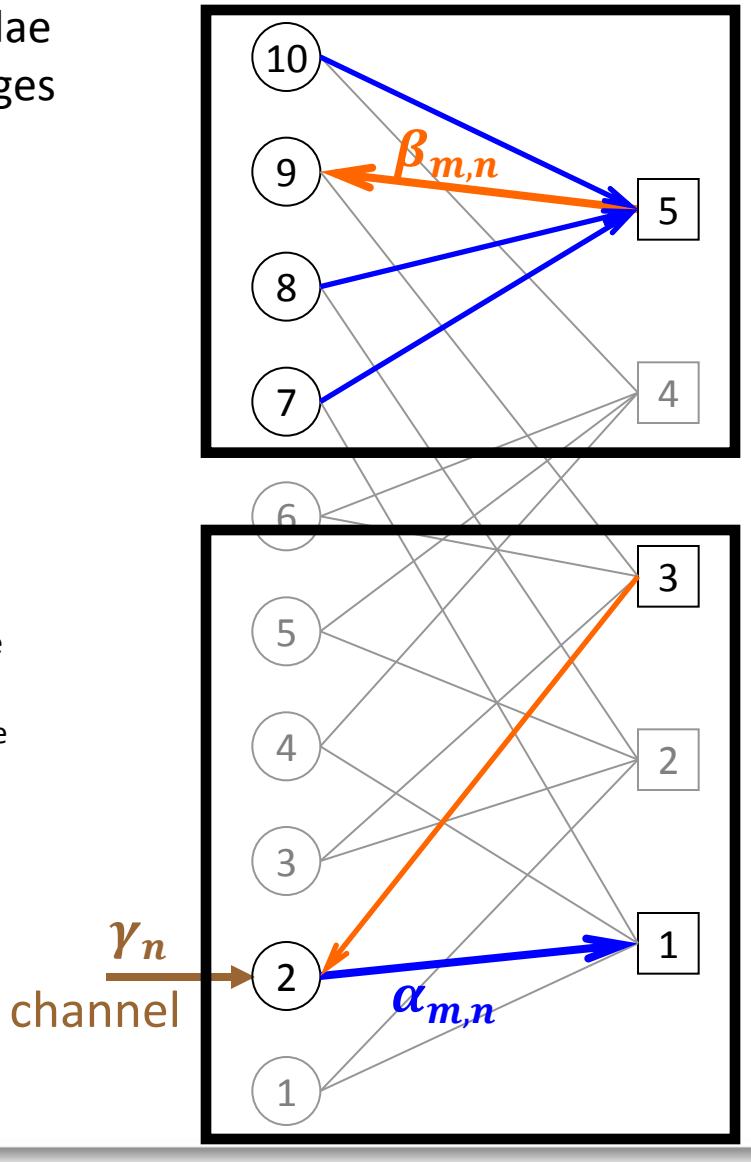
$$\beta_{m,n} = \left(\prod_{n' \in H(m) \setminus n} \operatorname{sgn}(\alpha_{m,n'}) \right) \min_{n' \in H(m) \setminus n} (|\alpha_{m,n'}|)$$

The sign of the outgoing $\beta_{m,n}$ message is the product of the signs of the incoming $\alpha_{m,n'}$ messages

The absolute value of the outgoing $\beta_{m,n}$ message is the minimum of the absolute values of the incoming $\alpha_{m,n'}$ messages

$$\alpha_{m,n} = \gamma_n + \sum_{m' \in H(n) \setminus m} \beta_{m',n}$$

The outgoing $\alpha_{m,n}$ message is the sum on channel LLR and incoming $\beta_{m,n}$ messages



Min-Sum decoder

Initialization: $\forall n = 1, \dots, N;\ \forall m \in H(n)$

$$\gamma_n = \log(\Pr(x_n = 0 \mid y_n)/\Pr(x_n = 1 \mid y_n))$$
$$\alpha_{m,n} = \gamma_n$$

Iterations

- **CNU**: $\forall m = 1, \dots, M;\ \forall n \in H(m)$

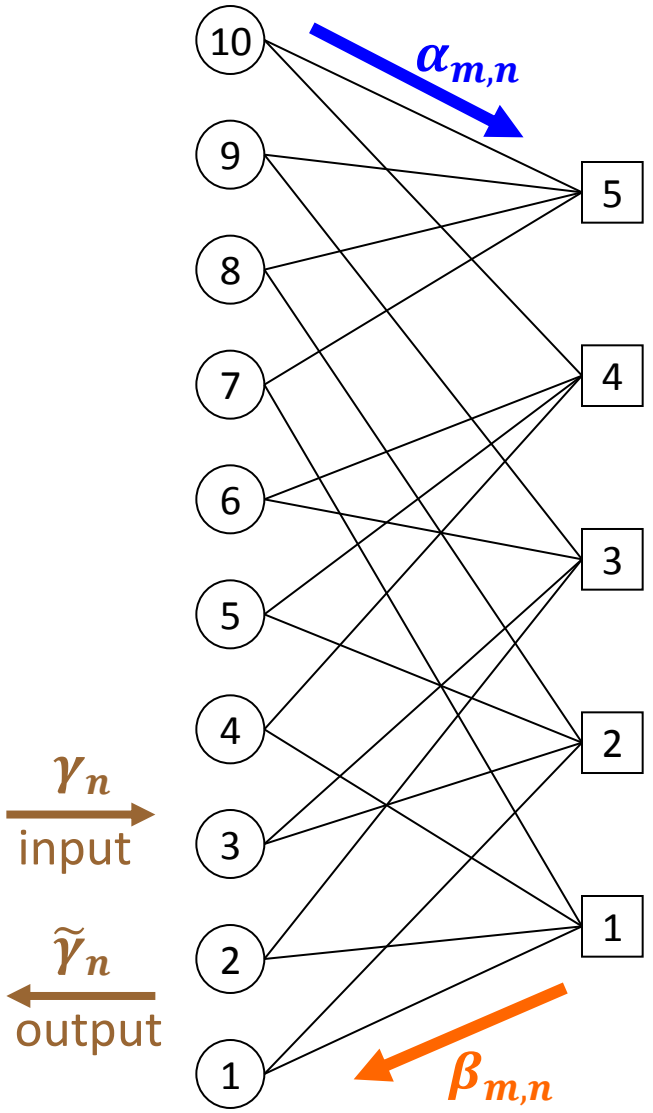
$$\beta_{m,n} = \left(\prod_{n' \in H(m) \setminus n} \operatorname{sgn}(\alpha_{m,n'}) \right) \min_{n' \in H(m) \setminus \mathrm{n}} (|\alpha_{m,n'}|)$$

- **VNU**: $\forall n = 1, \dots, N;\ \forall m \in H(n)$

$$\alpha_{m,n} = \gamma_n + \sum_{m' \in H(n) \setminus m} \beta_{m',n}$$

- **AP-LLR**: $\forall n = 1, \dots, N$

$$\tilde{\gamma}_n = \gamma_n + \sum_{m \in H(n)} \beta_{m,n}$$



Min-Sum decoder on faulty devices

Initialization: $\forall n = 1, \dots, N;\ \forall m \in H(n)$

$$\gamma_n = \log(\Pr(x_n = 0 \mid y_n)/\Pr(x_n = 1 \mid y_n))$$
$$\alpha_{m,n} = \gamma_n$$

Iterations

- **CNU**: $\forall m = 1, \dots, M;\ \forall n \in H(m)$

$$\beta_{m,n} = \left(\prod_{n' \in H(m) \setminus n} \operatorname{sgn}(\alpha_{m,n'}) \right) \min_{n' \in H(m) \setminus \mathrm{n}} (|\alpha_{m,n'}|)$$

- **VNU**: $\forall n = 1, \dots, N;\ \forall m \in H(n)$

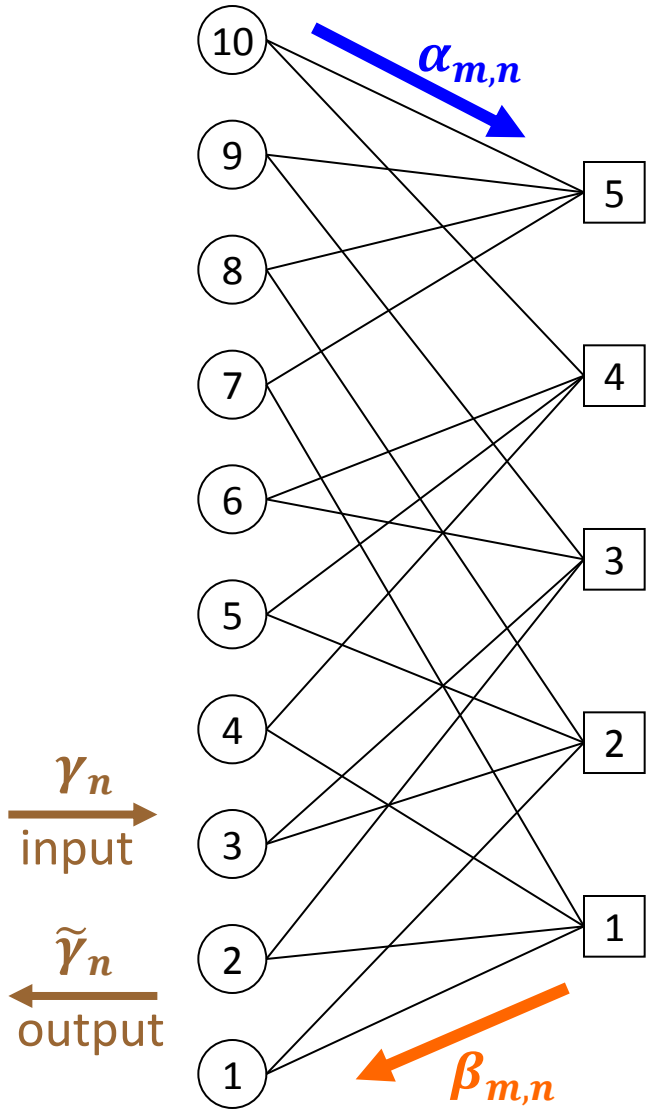
$$\alpha_{m,n} = \gamma_n + \sum_{m' \in H(n) \setminus m} \beta_{m',n}$$

- **AP-LLR**: $\forall n = 1, \dots, N$

$$\tilde{\gamma}_n = \gamma_n + \sum_{m \in H(n)} \beta_{m,n}$$

comparators

adders



Min-Sum decoder on faulty devices

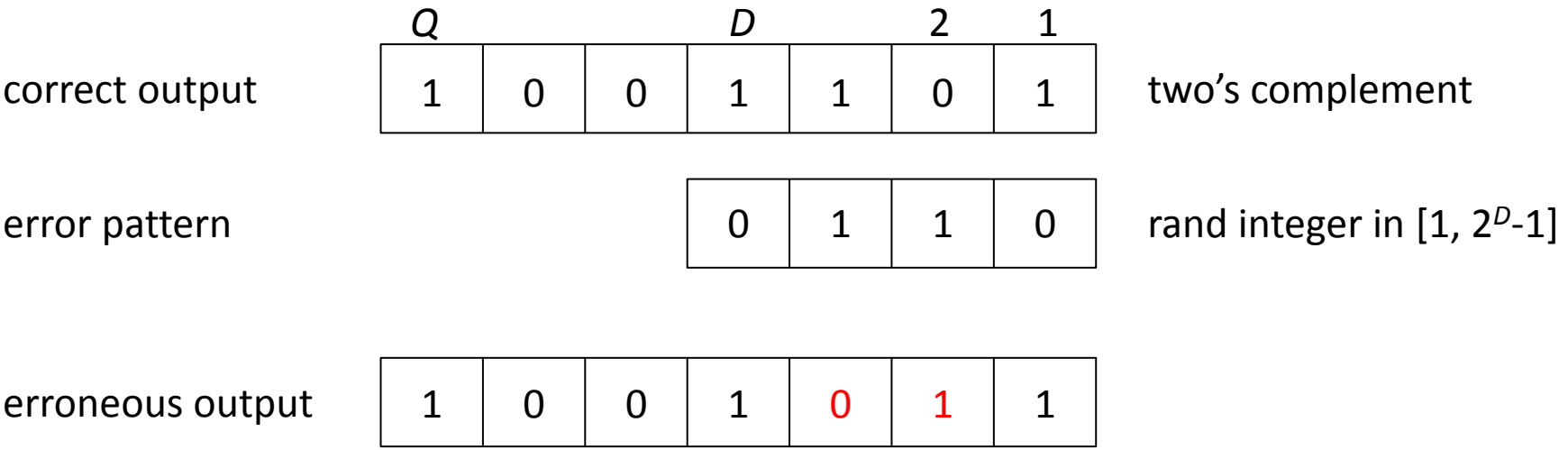
- Noisy components: new source of errors
  - Such errors may propagate through decoding iterations…
  - How does this impact on the error-correction capability of the decoder?
    - how to make sure that such an error propagation is not catastrophic?

- Theoretical analysis of “noisy” Min-Sum
  - Develop “noisy versions” of density-evolution
    - evaluate the theoretical performance loss due to noisy components
    - serve as guidelines for practical fault-tolerant implementations

- Practical fault-tolerant Min-Sum-based decoders
  - Evaluate the impact of faulty components on the performance of practical “finite-length” Min-Sum-based decoders

Error models for faulty arithmetic units

- Probabilistic adder (Q bits)
  - Two parameters: the **depth *D*** and the **error probability P_a**
  - P_a is the probability that an error occurs on at least one of the D LSBs



- Probabilistic comparator
  - P_c is the probability that the output is in error

Noisy density evolution

- We derived DE for fixed-point Min-Sum decoder
  - integrates above error models for arithmetic units (adder/comparator)

- Exchanged messages are random variables
  - Fixed-point implementation $\Rightarrow$ finite alphabet
  - $\boldsymbol{C}$ the PMF of input LLR values $\boldsymbol{\gamma_n}$ (depends only on the channel model)
  - $\boldsymbol{A}^{(\ell)}$, $\boldsymbol{B}^{(\ell)}$, and $\widetilde{\boldsymbol{C}}^{(\ell)}$ the PMFs of $\boldsymbol{\alpha_{m,n}}$, $\boldsymbol{\beta_{m,n}}$, and $\widetilde{\boldsymbol{\gamma}}_{\boldsymbol{n}}$ at iteration ℓ

- **DE equations**
  - Recursive formula (by tracking the update rules of exchanged messages):

$$\left(\boldsymbol{A}^{(\ell+1)}, \boldsymbol{B}^{(\ell+1)}, \widetilde{\boldsymbol{C}}^{(\ell+1)}\right) = f\left(\boldsymbol{A}^{(\ell)}, \boldsymbol{B}^{(\ell)}, \widetilde{\boldsymbol{C}}^{(\ell)}, \boldsymbol{C}\right)$$

  - Under the assumption that incoming messages to any bit and check node are independent
  - In particular, the graph must be cycle-free $\Rightarrow$ asymptotic performance

Noisy density evolution

- $P_\ell = \Pr(\widetilde{\gamma}_n < 0)$ is the **error probability** at iteration ℓ
- $P_\infty = \lim_{\ell \to \infty} P_\ell$ – **output error probability** (does not always exist!)

- **Useful decoder:** P_∞ exits and $P_\infty < P_0$
- **η-threshold:** $P_{\text{th}}(\eta) = \sup\{P_0 \mid P_\infty \text{ exists and } P_\infty < \eta\}$

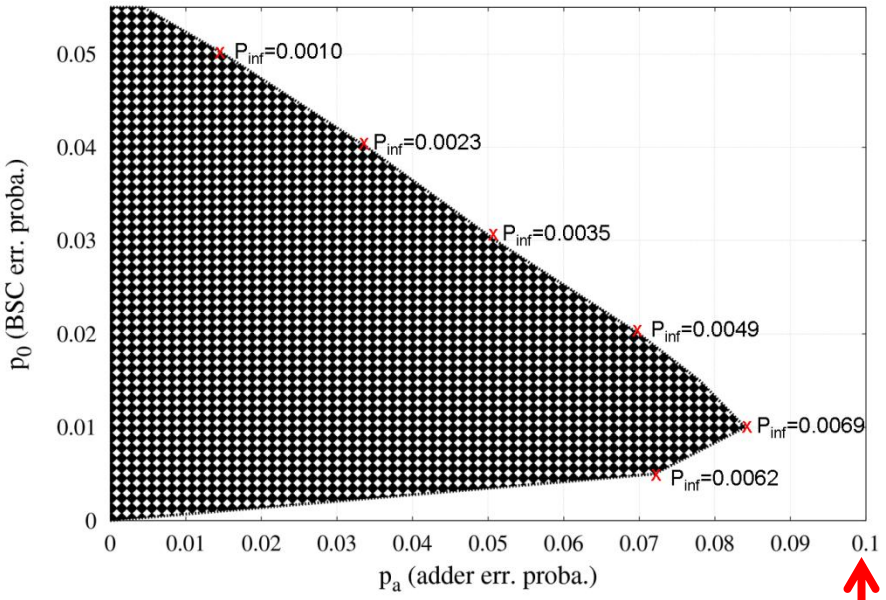
- **DE equations**
  - Recursive formula (by tracking the update rules of exchanged messages):

$$\left(\boldsymbol{A}^{(\ell+1)}, \boldsymbol{B}^{(\ell+1)}, \widetilde{\boldsymbol{C}}^{(\ell+1)}\right) = f\left(\boldsymbol{A}^{(\ell)}, \boldsymbol{B}^{(\ell)}, \widetilde{\boldsymbol{C}}^{(\ell)}, \boldsymbol{C}\right)$$

  - Under the assumption that incoming messages to any bit and check node are independent
  - In particular, the graph must be cycle-free $\Rightarrow$ asymptotic performance

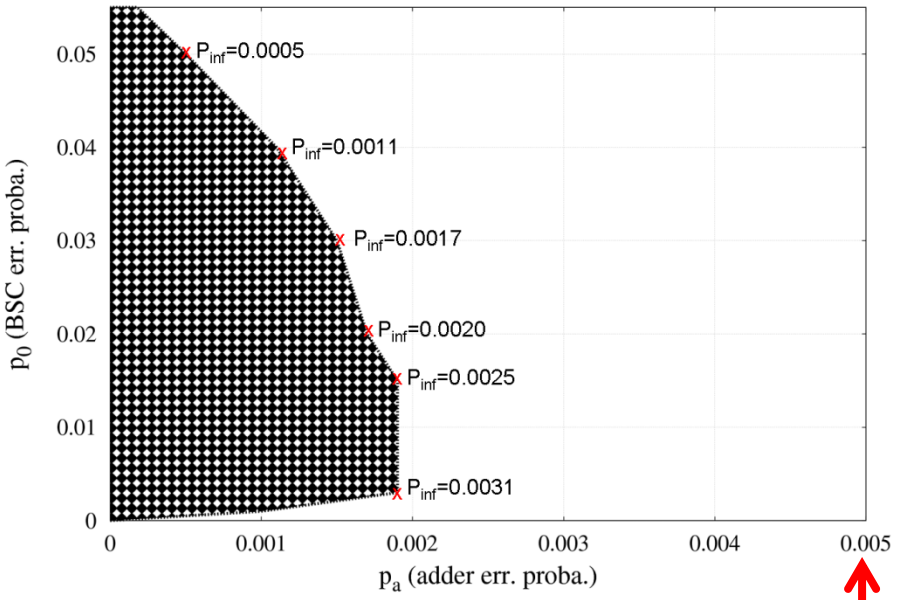
Useful regions for Min-Sum decoder / BSC

- (3, 6)-regular LDPC codes, fixed-point MS
  - Q = 5 bits (number of bits of the adder)
  - P_c = 0.001 (error probability of the comparator)



Depth D = 4

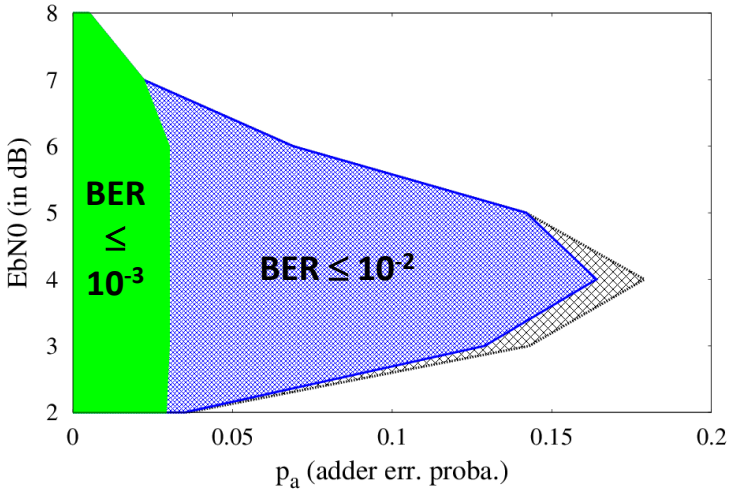
10 %



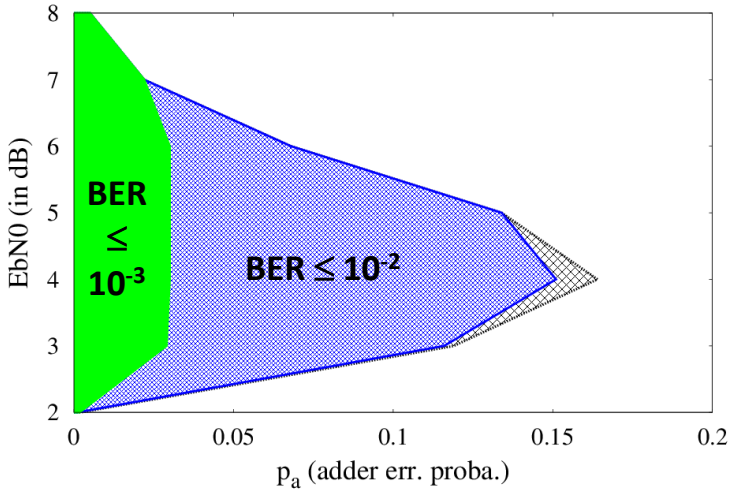
Depth D = 5

0.5 %

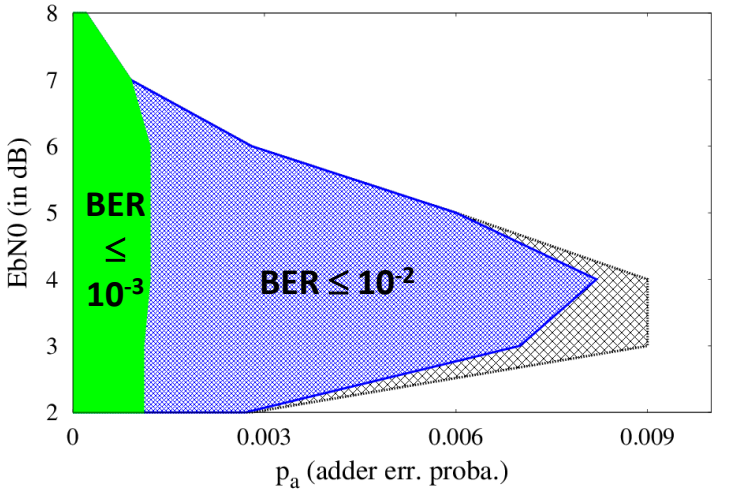
Useful regions for Min-Sum decoder / BI-AWGN



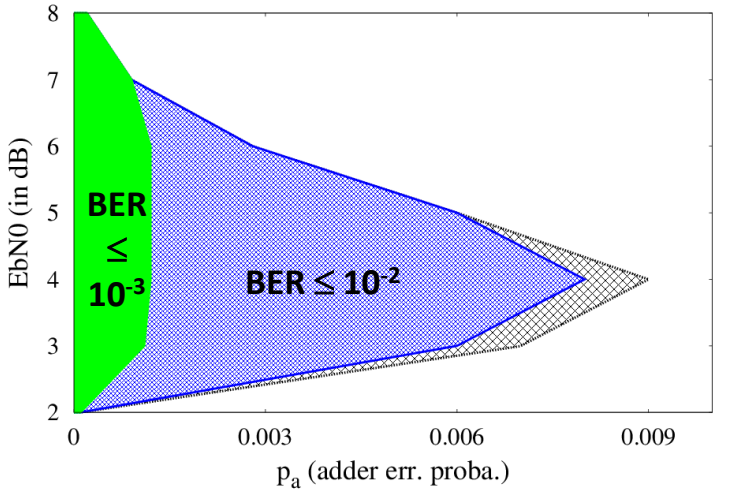
$D = 4$, $P_c = 0.001$



$D = 4$, $P_c = 0.01$



$D = 5$, $P_c = 0.001$



$D = 5$, $P_c = 0.001$

Useful regions for Min-Sum decoder

- Errors caused by noisy components do not necessarily propagate catastrophically through decoding iterations
  - Min-Sum decoder can still provide error protection with a given level of reliability, assuming that decoder's components are reasonably noisy…

- Some characteristics of the Min-Sum decoder
  - Less sensitive to **errors in comparators**
  - Less sensitive to **errors in the LSBs** of the adder
  - Highly sensitive to **errors in the sign bit** of the adder

Performance of Min-Sum-based decoder

- Min-Sum-based decoders
  - improved versions of the MS algorithm, with only a very limited (usually negligible) increase in complexity
  - Offset-Min-Sum (OMS)
  - Self-Corrected Min-Sum (SCMS)

Performance of Min-Sum-based decoder

- Min-Sum

Initialization: $\forall n = 1, \dots, N;\ \forall m \in H(n)$

$$\gamma_n = \log(\Pr(x_n = 0 \mid y_n)/\Pr(x_n = 1 \mid y_n))$$
$$\alpha_{m,n} = \gamma_n$$

Iterations

- **CNU**: $\forall m = 1, \dots, M;\ \forall n \in H(m)$

$$\beta_{m,n} = \left(\prod_{n' \in H(m) \setminus n} \operatorname{sgn}(\alpha_{m,n'}) \right) \min_{n' \in H(m) \setminus \mathrm{n}} (|\alpha_{m,n'}|)$$

- **AP-LLR**: $\forall n = 1, \dots, N$

$$\tilde{\gamma}_n = \gamma_n + \sum_{m \in H(n)} \beta_{m,n}$$

- **VNU**: $\forall n = 1, \dots, N;\ \forall m \in H(n)$

$$\alpha_{m,n} = \tilde{\gamma}_n - \beta_{m,n}$$

Performance of Min-Sum-based decoder

- **Self-Corrected** Min-Sum
  - intrinsic ability to detect and discard unreliable messages during the iterative decoding process
  - **Self-correction** = *noiseless patch* applied to the noisy MS decoder
  - a variable-to-check message $\alpha_{m,n}$ is erased (set to zero) if its sign changed with respect to the previous iteration

Initialization: $\forall n = 1, \dots, N;\ \forall m \in H(n)$

$$\gamma_n = \log(\Pr(x_n = 0 \mid y_n)/\Pr(x_n = 1 \mid y_n))$$
$$\alpha_{m,n} = \gamma_n$$

Iterations

- **CNU**: $\forall m = 1, \dots, M;\ \forall n \in H(m)$

$$\beta_{m,n} = \left(\prod_{n' \in H(m) \setminus n} \operatorname{sgn}(\alpha_{m,n'}) \right) \min_{n' \in H(m) \setminus n} (|\alpha_{m,n'}|)$$

- **AP-LLR**: $\forall n = 1, \dots, N$

$$\tilde{\gamma}_n = \gamma_n + \sum_{m \in H(n)} \beta_{m,n}$$

- **VNU**: $\forall n = 1, \dots, N;\ \forall m \in H(n)$

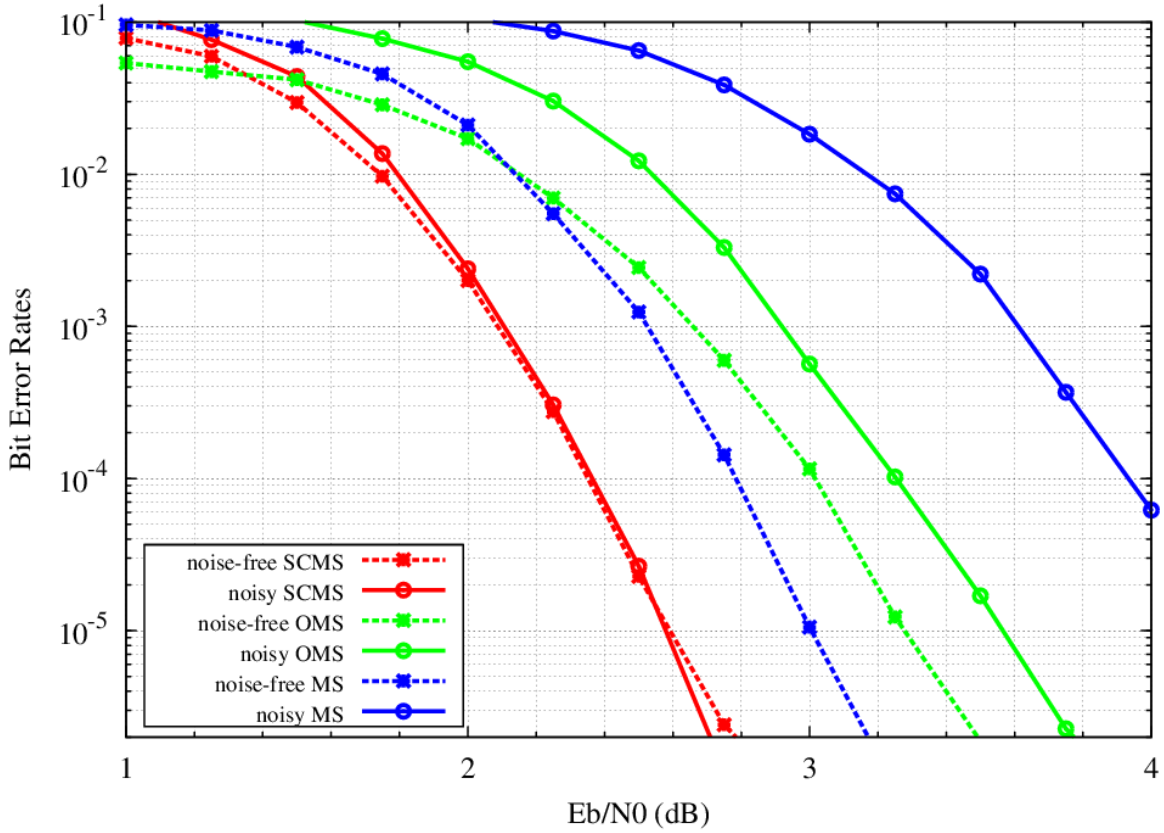
$$\alpha_{m,n}^{\text{tmp}} = \tilde{\gamma}_n - \beta_{m,n}$$

if $\operatorname{sgn}(\alpha_{m,n}^{\text{tmp}}) = \operatorname{sgn}(\alpha_{m,n})$ then $\alpha_{m,n} = \alpha_{m,n}^{\text{tmp}}$

else $\alpha_{m,n} = 0$

Performance of Min-Sum-based decoder

- Mackay's regular (3,6)-LDPC code, [K = 504, N = 1008]
- Fixed-point decoders: 4 / 5 bits (exchanged messages / AP-LLR)



Comp.err. prob: $P_c = 0.01$
Adder err. prob: $P_a = 0.01$
depth = 4

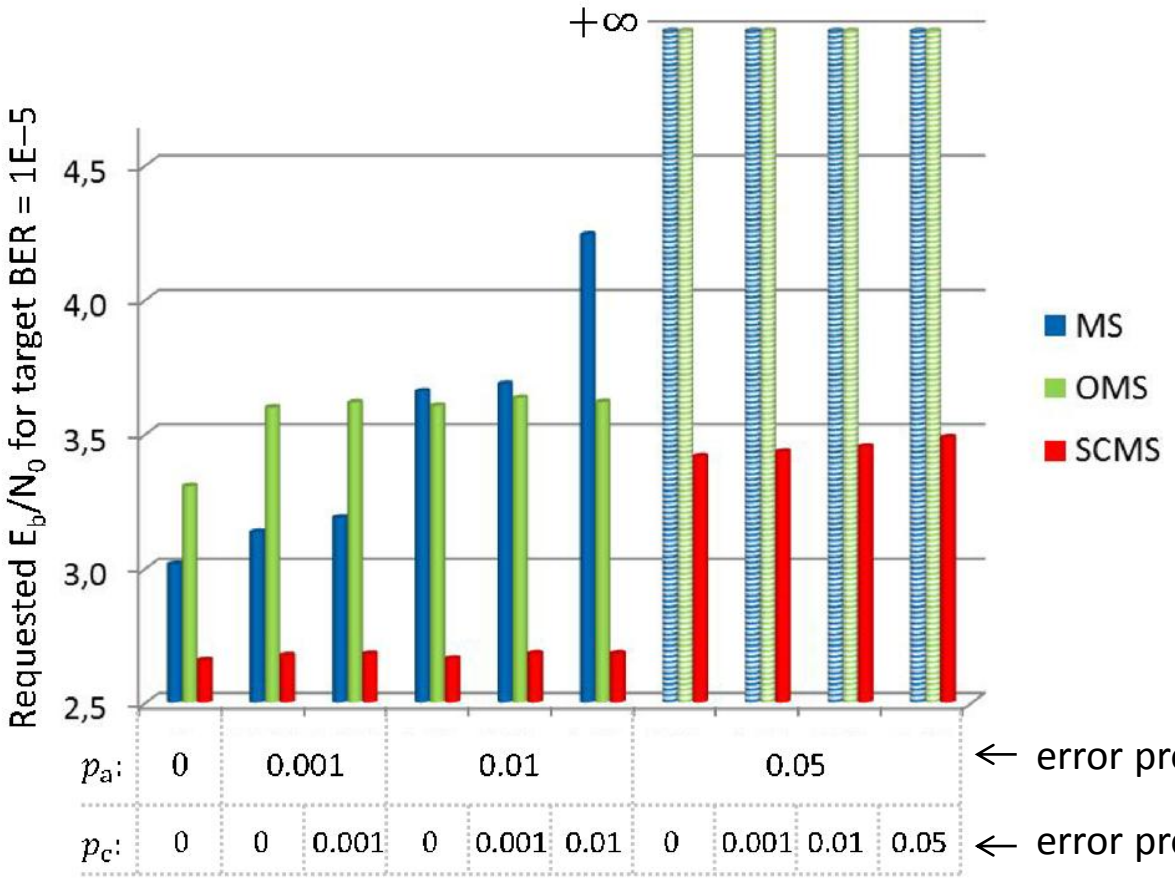
Color code:
SCMS
MS
OMS

Dashed curves: noiseless
Solid curves: noisy

Remark: noiseless SCMS achieves ≈ Belief Propagation performance

Performance of Min-Sum-based decoder

- Mackay's regular (3,6)-LDPC code, [K = 504, N = 1008]
- Fixed-point decoders: 4 / 5 bits (exchanged messages / AP-LLR)



Requested channel E_b/N_0 for target decoded BER = 10^{-5}

($E_b/N_0 = \infty$ means target BER cannot be achieved)

← error prob. of the adder (**depth = 4**)

← error prob. of the comparator

Conclusion

- "Adjustable" error-models for noisy Min-Sum-based decoders
- Density evolution analysis of the noisy Min-Sum decoder
  - proved that error protection (with a certain level of reliability) is still possible
  - characterized the sensitivity of the decoder to variations of the parameters of the error model, in terms of useful regions
- Finite-length performance of Min-Sum-based decoders
  - highlighted the limitations of the **theoretical analysis** with respect to **practical implementations**
  - evaluate finite-length performance for various parameters of the hardware noise model
  - **SCMS**: intrinsic ability to detect and discard unreliable messages, which proves to be particularly useful for noisy implementations

LABORATOIRE D'ÉLECTRONIQUE
ET DE TECHNOLOGIES
DE L'INFORMATION

CEA-Leti
MINATEC Campus, 17 rue des Martyrs
38054 GRENOBLE Cedex 9
Tel. +33 4 38 78 36 25

www.leti.fr

Merci de votre attention