Reliable LDPC Encoding on Faulty Hardware

Elsa Dupraz[1], Satish Kumar Grandhi[2], Valentin Savin[3],
Emanuel Popovici[2], David Declercq[1]

[1] *ETIS / ENSEA - Université de Cergy-Pontoise - CNRS UMR 8051*
[2] Department of Electrical and Electronic Engineering, University College Cork
[3] CEA-LETI

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

Motivations

Context of error-correcting codes on faulty hardware

Motivations

- Until now, focus on the analysis of noisy decoders [Huang13] [Balatsoukas14] [Ngassa14]
- Construction of robust LDPC decoders [Dupraz15] (noise level up to 10^{-2})
- In this work : what about LDPC encoding ?

Goals

- Analysis of the robustness of standard encoding techniques
- Construction of robust encoding solutions

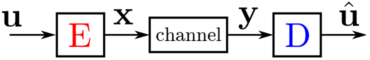
Outline

1. The Encoding Problem
2. Robust Encoding Solution
3. Performance Comparison

Outline

1. The Encoding Problem
2. Robust Encoding Solution
3. Performance Comparison

LDPC codes



Decoding

H ($n \times k$) : parity check matrix
$\mathbf{x}$: codeword (n)

$$H^T \mathbf{x} = 0$$

H sparse, optimized for good perf.

Encoding

$\mathbf{u}$: information sequence (m)

$$\mathbf{x} = \mathcal{E}(\mathbf{u})$$

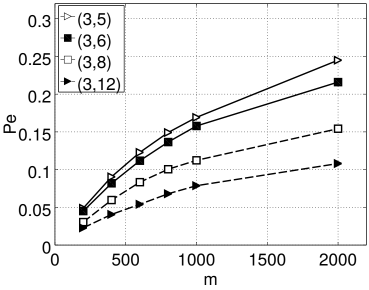
Systematic encoding : $\mathbf{x} = [\mathbf{u}, \mathbf{p}]^T$

- Error Model for the noise in the encoder, faulty XOR gates

$$p_{xor} = P(\tilde{c} \neq a \oplus b)$$

Standard Encoding Solutions

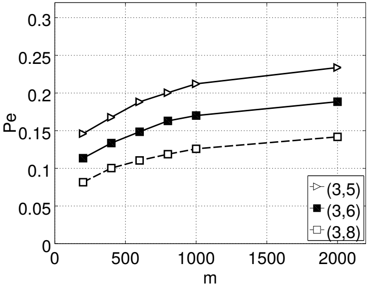
- Encoding from Generator matrix
  - $G\ (n \times m)$ s.t. $H^T G = [0]$
  - Encoding : $\mathbf{x} = G\mathbf{u}$



Encoding error probability, for $pxor = 10^{-3}$

G is not sparse : high error probability

- Lower Triangular Encoding
  - $H_t = [Q, T]^T$
  - $p_j = \sum_{k \in Q_j} u_k + \sum_{i \in T_j} p_i$



Encoding error probability, for $pxor = 10^{-3}$

Error Propagation during the encoding

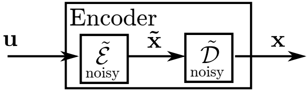
1. The Encoding Problem

2. Robust Encoding Solution

3. Performance Comparison

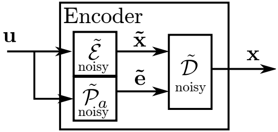
Codeword Prediction Encoder (CPE)

- First solution : decoder at the encoder



- Decoding from $H^T \mathbf{x} = 0$

- To go further : Codeword Prediction Encoder (CPE)

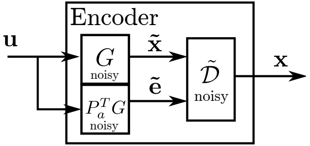


- Augmented codeword $\mathbf{x}_a = [\mathbf{x}, \mathbf{e}]^T$
- Decoding from $H_a^T \mathbf{x}_a = 0$
- Only $\mathbf{x}$ is transmitted on the channel

Code Construction (Matrix Multiplication)

- Objective : design H and H_a for good decoding performance from both
  - $H_a^T \mathbf{x}_a = 0$ (CPE), with $\mathbf{x}_a = [\mathbf{x}, \mathbf{e}]^T$
  - $H^T \mathbf{x} = 0$ (Channel transmission)

- Encoding from Matrix Multiplication , with $H_a^T = [P_a, I]$

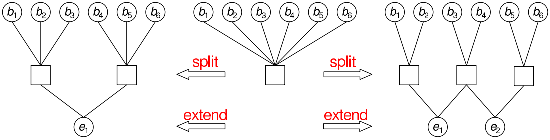


CPE Decoding from $H_a^T \mathbf{x}_a = 0$,
and then from $H^T \mathbf{x} = 0$

Problems
- $P_a^T G$ has a lot of non-zero components
- Two successive decodings
- Independent construction of H and H_a

Code Construction (Split-Extension)

- Objective : design H and H_e for good decoding performance from both
  - $H_e^T \mathbf{x}_a = 0$ (CPE), with $\mathbf{x}_a = [\mathbf{x}, \mathbf{e}]^T$
  - $H^T \mathbf{x} = 0$ (Channel transmission)
- Split-Extended codes [Savin10]



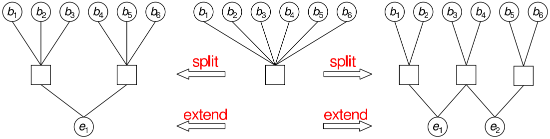
Example

- In H : $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 0$
- Compute new parity bit $e_1 = x_1 + x_2 + x_3$
- In H_e : $e_1 + x_1 + x_2 + x_3 = 0$ and $e_1 + x_4 + x_5 + x_6 = 0$

Code Construction (Split-Extension)

- Objective : design H and H_e for good decoding performance from both
  - $H_e^T \mathbf{x}_a = 0$ (CPE), with $\mathbf{x}_a = [\mathbf{x}, \mathbf{e}]^T$
  - $H^T \mathbf{x} = 0$ (Channel transmission)

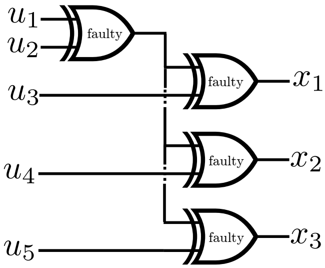
- Split-Extended codes [Savin10]



  - From original code H, construct extended code H_e
  - Use H_e at the encoder
  - Use H after channel transmission

Individual Gate Protection

- With iterative encoding : error propagation



- Critical degree CT(F) of gate F : number of outputs to which it participates
- Criticality Threshold CT : protect any gate such that CT(F) > CT.

Experimental results (1)

- Random (3, 5)-code for H, Random (3, 6)-code for H_a, $m = 400$
- Faulty Min-Sum decoder (*i.i.d.* errors, error probability $p = 10^{-3}$)

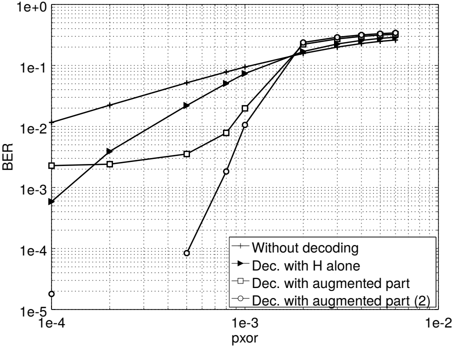


FIGURE: Encoding from Generator Matrix

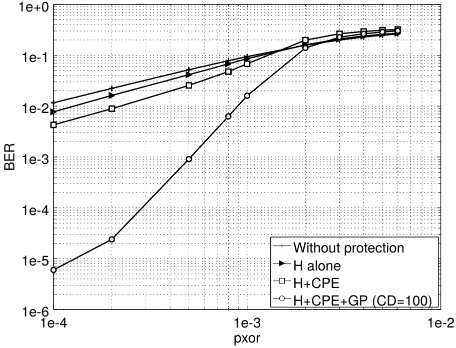


FIGURE: Encoding from Circuit Design

Experimental Setup (2)

- Four QC-LDPC iRisc codes
  - dv3-r12, dv4-r12 ($m = 975$, $n = 1296$, $n_a = 1620$)
  - dv3-r34, dv4-r34 ($m = 650$, $n = 1296$, $n_a = 1944$)
- Three faulty decoders (*i.i.d.* errors, error probability p, perfect APP)
  - Gallager B
  - Min-Sum
  - Self-corrected Min-Sum

TABLE: Number of XOR gates

Code	Generator matrix	Circuit Design
dv3-r12	296734	44399
dv3-r34	170362	28182
dv4-r12	300205	45175
dv4-r34	303163	27167

Complexity Analysis

TABLE: Critical Gate count for different codes

Code	Circuit design node count	CT=10	CT=20	CT=50
dv3-r12	44399	3373	1844	833
dv3-r34	28182	2288	1240	537
dv4-r12	45175	3424	1851	824
dv4-r34	27167	2112	1183	488

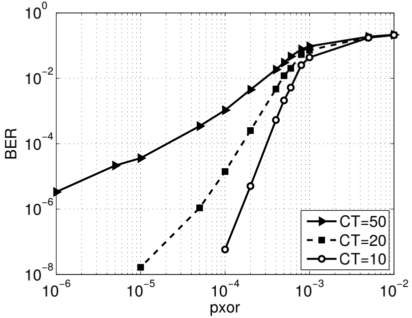


FIGURE: BER with respect to p_{xor} for dv4-r34 code with Min-Sum decoder ($p = 10^{-3}$)

Performance Comparison

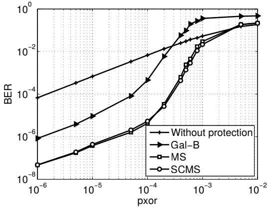


FIGURE: dv3-r34, $p = 0.001$, $CT = 20$

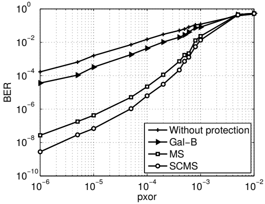


FIGURE: dv3-r12, $p = 0.01$, $CT = 50$

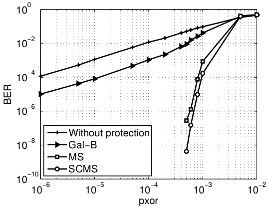


FIGURE: dv4-r12, $p = 0.001$, $CT = 20$

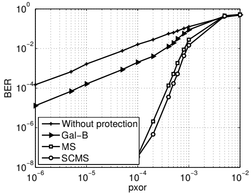


FIGURE: dv4-r12, $p = 0.001$, $CT = 50$

Conclusions

- CPE consists of computing extra parity bits to protect the encoding
- CPE with Split-Extension provides a robust encoding solution

- More accurate error models to be considered
- Address the issue of critical gates
- Consider other encoding techniques in CPE