Noise-Aided Gradient Descent Bit-Flipping Decoders approaching Maximum Likelihood Decoding

D. Declercq, C. Winstead, B. Vasic, F. Ghaffari, P. Ivanis and E. Boutillon

September 7, 2016

This work has been funded in parts by the ANR NAND Project under grant n^o ANR-15-CE25-0006-01.

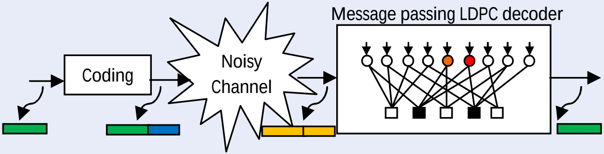
Outline

Contents

Context & Objectives

Low Density Parity Check (LDPC) Decoders

- Applications: Wired, Wireless communication, Storage...



2 types of LDPC decoding algorithms:

- **Soft information algorithms :** Sum-Product, Min-Sum..., powerful error correction capacity but high complexity.
- **Hard-decision algorithms :** Bit Flipping (BF), Gradient Descent Bit Flipping (GDBF), Gallager-B, etc. low complexity, usually weak in error correction.
- **Towards a new type of decoder :** low complexity + noise perturbation ⇔ powerful error correction,

Context and Objectives

A new noisy decoding framework

- In [Varshney 2011], [Vasic 2007], [Yazdi 2013], authors focused on trying to compensate faulty hardware effects, by making the iterative decoders robust and fault-tolerant,
- Inspired by the robustness of iterative decoders, a new direction of research has been identified:

Additionnal noise could help error correction instead of being an enemy

- the Additional noise could come from the circuit design, or through deliberate noise injection.

Injected Noise to break the attraction of Trapping Sets (TS)

- Failures of iterative decoders are mainly due to TS : fixed points or loopy attractors,
- Not only can it help in the error floor, but also in the waterfall,

[Varshney 2011] L. Varshney, "Performance of LDPC codes under faulty iterative decoding", *IEEE Trans. on Info. Theory*, 2011
[Vasic 2007] B. Vasic et al., "An information theoretical framework for analysis and design of nanoscale fault-tolerant memories based on LDPC codes", *IEEE Trans. on Circuits and Systems I*, 2007
[Yazdi 2013] S. Yazdi et al., "Gallager-B decoder on noisy hardware", *IEEE Trans. on Commun.*, 2013

Very simple decoder + Noise ⇔ Powerful Decoder

Injected Noise in the decoder can help to combat the channel errors

- First demonstration of this phenomenon on the Gradient-Descent Bit-Flipping decoder:
  - Probabilistic Gradient Descent Bit-Flipping (PGDBF) for the BSC, [Rasheed 2014]
  - Noisy Gradient Descent Bit-Flipping (NGDBF) for the BI-AWGN, [Sundararajan 2014]
- Could be generalized to other hard-decision decoders (Gal-B for example), [Ivanis 2015]

Issues raised about noise injection

- **Where ?** VNU ? CNU ? Memories ? Only localized computing units ?
- **When ?** From the first iteration ? After a given number of iterations ? When a decoding failure is detected ?
- **How ?** Which Noise model ? Values of the parameters ? Hardware realization ?

[Rasheed 2014] O. Rasheed et al., "Fault-tolerant probabilistic gradient-descent bit flipping decoder", *IEEE Communications Letters*, 2014

[Sundararajan 2014] Sundararajan et al., "Noisy gradient descent bit-flip decoding for LDPC codes", *IEEE Transactions on Communications*, 2014

[Ivanis 2015] P. Ivanis et al., "MUDRI: A fault-tolerant decoding algorithm", *ICC*, 2015

Contents

Gradient Descent Bit-Flipping Decoder

Concept of Gradient Descent Bit Flipping (GDBF) Algorithm [Wadayama 2010]

- Iterative propagation of binary information between 2 groups of processing units:
  1. **Check Nodes Units** (CNU): compute parity check equations (XOR operations),

$$c_m^{(l)} = \bigoplus_{v_n \in \mathcal{N}(c_m)} v_n^{(l)}$$

  2. **Variable Nodes Units** (VNU): the VN value is flipped if the number of violated CN neighbors is too large.
- Requires the computation of an Energy/Inversion Function in order to select the bit-flips.

Energy function for the BSC channel (high energy = low reliability)

$$E_{v_n}^{(l)} = L(y_n)^{(l)} + \sum_{c_m \in \mathcal{N}(v_n)} c_m^{(l)} \qquad L(y_n)^{(l)} = y_n \oplus v_n^{(l)}$$

Energy function for the AWGN channel (low energy = low reliability)

$$E_{v_n}^{(l)} = L(y_n)^{(l)} + w \sum_{c_m \in \mathcal{N}(v_n)} (1 - 2\, c_m^{(l)}) \qquad L(y_n)^{(l)} = (1 - 2\, v_n^{(l)}) \log \left(\frac{prob(y_n | v_n = 0)}{prob(y_n | v_n = 1)} \right)$$

Flipping Sets

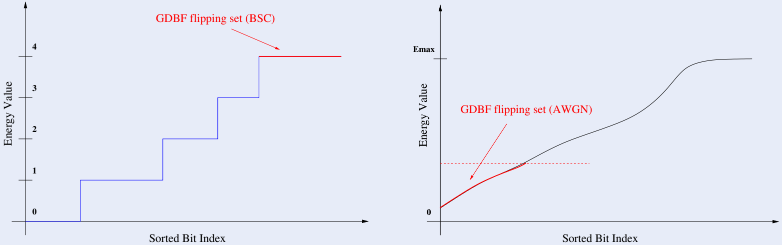
Flipping sets contain indices of bits to be flipped

$$\mathcal{F}_{\text{BSC}}^{(l)} = \left\{ n \in \{1, N\} \,;\, E_{v_n}^{(l)} = b^{(l)} \right\} \qquad \text{[BSC]}$$

$$\mathcal{F}_{\text{AWGN}}^{(l)} = \left\{ n \in \{1, N\} \,;\, E_{v_n}^{(l)} \leq \theta \right\} \qquad \text{[AWGN]}$$

- $b^{(l)} = \max_{1 \leq n \leq N} (E_{v_n}^{(l)})$ is the maximum energy (BSC), and θ is a pre-determined threshold.

Graphical representation of flipping sets



Noisy Flipping Sets

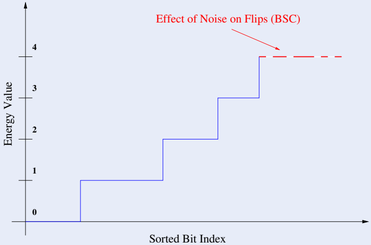
PGDBF : random variable $\epsilon_n^{(l)} \sim \mathcal{U}_{[0,1]}$ main parameter p_0

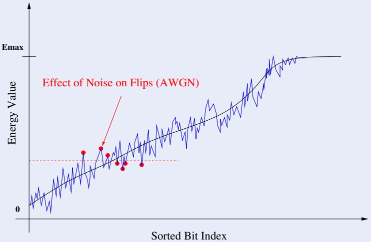
$$\widetilde{\mathcal{F}}_{\text{PGDBF}}^{(l)} = \left\{ n \in \mathcal{F}_{\text{BSC}}^{(l)} \; ; \; \epsilon_n^{(l)} < p_0 \right\}$$ [PGDBF]

NGDBF : random variable $\epsilon_n^{(l)} \sim \mathcal{N}(0, \sigma_p)$ main parameter σ_p

$$\widetilde{\mathcal{F}}_{\text{NGDBF}}^{(l)} \left\{ n \in [1, N] \; ; \; E_{v_n}^{(l)} \leq \theta + \epsilon_n^{(l)} \right\}$$ [NGDBF]

Graphical representation of noisy flipping sets





Unified presentation of GDBF and Noisy GDBF decoders

GDBF and NA-GDBF Algorithms : [iteration l]

[Step 1] Compute CNs values

$c_m^{(l)}, \forall m = 1, \dots, M,$

[Step 2] Compute Energy functions at VNs

$E_{v_n}^{(l)}, \forall n = 1, \dots, N$

[Step 3] Compute the flipping sets

$\mathcal{F}_{\text{BSC}}^{(l)}$ or $\mathcal{F}_{\text{AWGN}}^{(l)}$ for the deterministic GDBF

$\mathcal{F}_{\text{PGDBF}}^{(l)}$ or $\mathcal{F}_{\text{NGDBF}}^{(l)}$ for the noise-aided GDBF

[Step 4] Bit flipping

$$\forall n \in \mathcal{F}^{(l)} \quad v_n^{(l+1)} = \overline{v_n^{(l)}}$$

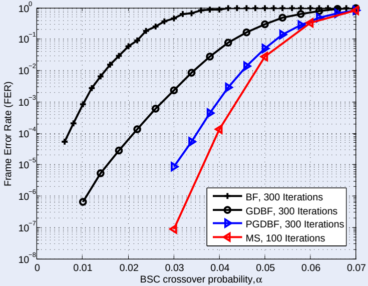
$$\forall n \notin \mathcal{F}^{(l)} \quad v_n^{(l+1)} = v_n^{(l)}$$

Complexity and Performance of Noise Injection

Deterministic GDBF and MS versus Probabilistic GDBF

- Regular quasi-cyclic LDPC code $(d_v, d_c) = (3, 6)$, $N = 1296$ bits,

Noise-aided simple decoders can approach performance of powerful decoders with a very small hardware overhead ($\simeq$ 5%)



dv3R050N1296	Area and Throughput θ Comparison							
	Code length	AREA (μm^2)	f_{max} (MHz)	N_c	$FER = 1e-5$		$\alpha = 0.02$	
					It_{ave}	θ (Gbit/s)	It_{ave}	θ (Gbit/s)
GDBF	1296	87810	222	1	2.00 (@$\alpha = 0.005$)	144.00	2.95 ($FER = 3e^{-4}$)	97.63
PGDBF	1296	92645	232	1	3.50 (@$\alpha = 0.012$)	86.11	2.88 ($FER = 5e^{-6}$)	104.65
Min-Sum	1296	950000	111	6	1.94 (@$\alpha = 0.025$)	12.36	1.15 ($FER = 1e^{-7}$)	20.85

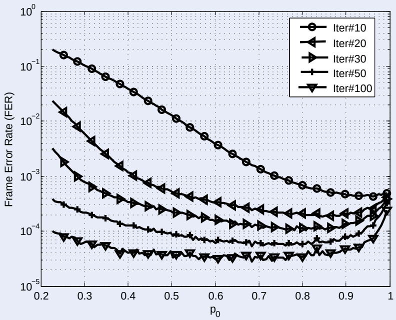
Contents

Statistical Analysis in the Waterfall : PGDBF

Analyze and quantify the amount of noise that should be introduced

- **Objective :** optimize the amount of noise that maximizes the coding gains
- Through Monte-Carlo simulations in the waterfall and the error-floor.

Binary Symmetric Channel - Waterfall Region - Tanner Code ($M = 93$, $N = 155$)



- **Conclusion 1:** for the first decoding iterations random noise does not help,
- **Conclusion 2:** a wide range of $p_0 \in [0.5; 0.9]$ values achieve the same coding gain.

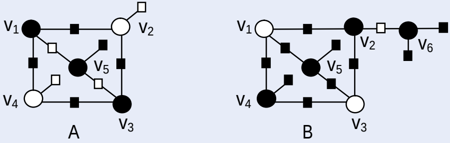
Statistical Analysis in the Error Floor : PGDBF

Errors located on Trapping Sets

- In the error floor region, the dominant uncorrectable error configurations are concentrated on Trapping Sets
- Trapping Sets $TS(a, b)$ are defined as a small set of a VNs for which the neighboring CNs contains exactly b odd degree CNs
- $TS(5, 3)$ is the smallest trapping set for regular $d_v = 3$ LDPC codes with girth $g = 8$.

Smallest Error Events not correctable by the GDBF

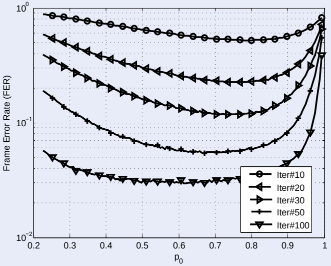
- weight-3 error patterns which does not satisfy 5 parity-checks,
- weight-4 error patterns which does not satisfy 10 parity-checks,



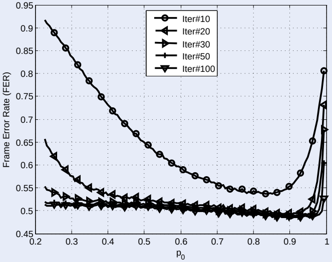
Statistical Analysis in the Error Floor : PGDBF

Frame error rate with fixed input errors

- for each Monte-Carlo round, only the random noise $\epsilon_n^{(l)}$ differ, the channel errors are kept the same.



3 bits error pattern



4 bits error pattern

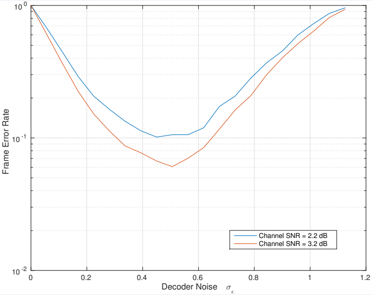
- **Conclusion 1:** the random noise is useful in the first iterations,
- **Conclusion 2:** the same wide range of $p_0 \in [0.5; 0.9]$ values achieve the maximum coding gain.

Statistical Analysis in the Error Floor : NGDBF

Errors located on Trapping Sets

- **Step 1 :** apply a bias on the channel samples associated with a TS,
- **Step 2 :** select only frames for which the GDBF fails on the selected TS,
- **Step 3 :** restart decoding of the same frame with the NGDBF,
- **Step 4 :** compute the residual FER on the selected frames.

TS(5, 3) on the Tanner (155, 64) code



- **Conclusion :** the choice of optimum σ_p is narrower than for the PGDBF.

Contents

Trapping sets of noiseless GDBF decoders

Assumptions/Definitions

- We analyze errors located on a trapping set that is isolated from the rest of the graph,
- Shortest example for $d_v = 3$, $g = 8$ LDPC codes: the TS(5, 3) trapping set,
- Definition of a TS state : $S = (v_1, v_2, v_3, v_4, v_5)_2$
- *Examples :* correct state $S_0 = (0, 0, 0, 0, 0)_2$ - error state $S_{21} = (1, 0, 1, 0, 1)_2$.

A trapping set for the GDBF decoder : oscillating behavior between S_{21} and S_{26}.

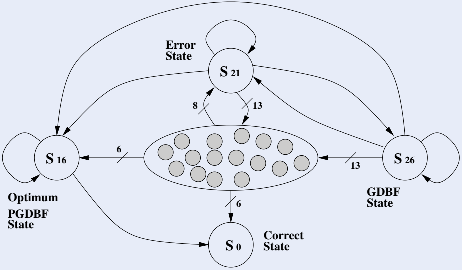
- black circles are in error,
- circled-dashed circles have maximum energy,

State space of the PGDBF decoder

Some observations starting with error state $S_{21} = (1,0,1,0,1)_2$

- Deterministic GDBF oscillates indefinitely between S_{21} and S_{26},
- Only 20 out of the 32 possible states are achievable
- S_0 is one of the achievable states,

State space of the PGDBF on the TS(5,3)

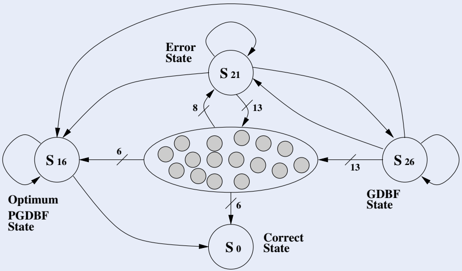


PGDBF probability of escaping a TS

Optimization of the parameter p_0

- The shortest path between the error and the correct state is : $S_{21} \rightarrow S_{16} \rightarrow S_0$,
- This is the only path with 2 transitions, its probability of occurence is $p_0^3(1-p_0)^2$,
- Optimization strategy : maximize the probability of the shortest path $\rightarrow$ $p_0 = 0.6$

State space of the PGDBF on the TS(5,3)



NGDBF probability of escaping a TS

Assumptions/Definitions

- All 32 states are achievable,
- let S and S' be two consecutive states, and $T(S,S')$ the indices of bits in which S and S' differ,
- Under the isolation assumption, only the state nodes $(v_1, v_2, v_3, v_4, v_5)$ have negative LLRs,

Transition Probabilities

- The transition probabilities between states S and S'

$$\Lambda(S,S') = \prod_{n \in T(S,S')} F(E_{v_n}, \theta, \sigma_p) \prod_{n \notin T(S,S')} (1 - F(E_{v_n}, \theta, \sigma_p)),$$

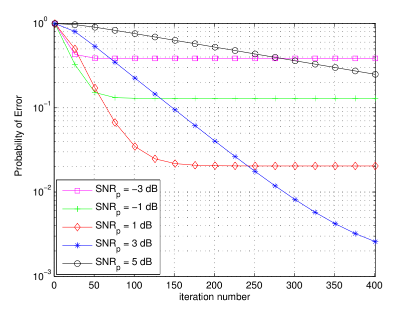
- with

$$F(E_{v_n}, \theta, \sigma_p) = \frac{1}{\sqrt{2\pi}\sigma_p} \int_{-\infty}^{\theta - E_{v_n}} e^{-\frac{x^2}{2\sigma_p^2}} dx,$$

- using $\Lambda(S,S')$ and the Perron-Frobenius theorem, we can compute the probability of escaping a TS, at each iteration.

NGDBF probability of escaping a TS

Probability of TS error for the NGDBF



Optimization of σ_p

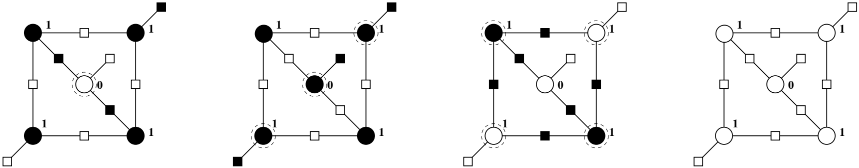
- Trade-off between convergence speed and TS correction probability,
- Optimization of σ_p depends on how the TS are effectively isolated, and on the required latency.

Contents

Advantage of modifying the PGDBF threshold

PGDBF with maximum threshold $E_{v_n}^{(l)} = b^{(l)}$: oscillation

PGDBF with decreased threshold $E_{v_n}^{(l)} \geq b^{(l)} - 1$: convergence



Decoder-Dynamic Shift PGDBF

- Instead of decreasing the threshold randomly, we let the threshold variations follow the dynamics of the decoder,
- Our proposed modification is to use the value of the maximum energy at the previous iteration $b^{(l-1)}$.

DDS-PGDBF : [iteration l]

[Step 1] Compute CNs values

$c_m^{(l)}, \forall m = 1, \ldots, M,$

[Step 2] Compute energy values at VNs

$E_{v_n}^{(l)}, \forall n = 1, \ldots, N,$

[Step 3] Compute the modified flipping set with threshold value from the previous iteration

$$\widetilde{\mathcal{F}}^{(l)}_{\text{DDS-PGDBF}} = \left\{ n \in \{1, N\} \,;\, E_{v_n}^{(l)} \geq b^{(l-1)} \text{ and } \epsilon_n^{(l)} < p_0 \right\}$$

[Step 4] Bit flipping

$$\forall n \in \widetilde{\mathcal{F}}^{(l)}_{\text{DDS-PGDBF}} \quad v_n^{(l+1)} = \overline{v_n^{(l)}}$$

$$\forall n \notin \widetilde{\mathcal{F}}^{(l)}_{\text{DDS-PGDBF}} \quad v_n^{(l+1)} = v_n^{(l)}$$

[Step 5] Update energy values at VNs

$$E_{v_n}^{(l)} = E_{v_n}^{(l)} - y_n \oplus v_n^{(l)} + y_n \oplus v_n^{(l+1)}.$$

[Step 6] Compute new maximum energy

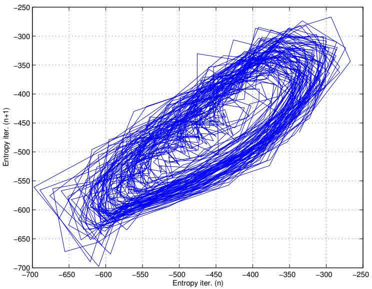
$$b^{(l)} = \max_{1 \leq n \leq N} \left(E_{v_n}^{(l)}\right)$$

Decoding Dynamics

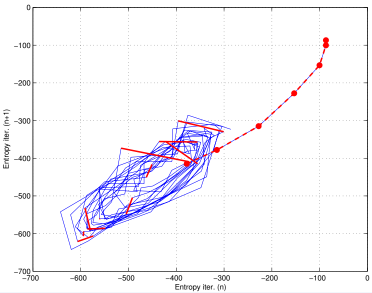
Behavior of the DDS-PGDBF decoder

- We plot the state space of the average binary entropy for the VN values,
- Errors are mainly located on a collection of uncorrectable TS,
- Using the previous threshold allows big jumps in the state space of the decoder,

PGDBF decoder ($p_0 = 0.6$)

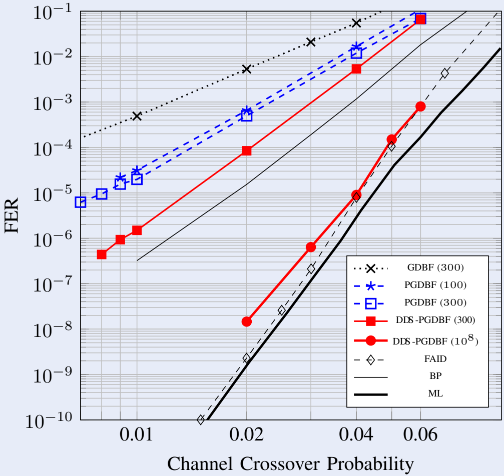


DDS-PGDBF decoder ($p_0 = 0.6$)



Performance of DDS-PGDBF

With a very large number of iterations, DDS-PGDBF approaches MLD



Contents

Conclusion

Noise-Aided Iterative Decoders

- Injected Noise turn **weak decoders into powerful decoders**,
- Extra hardware complexity is negligible,
- The coding gains come at the cost of a larger number of iterations.

Optimization of Noise Statistics

- for the PGDBF on the BSC and for the NGDBF on the AWGN channel,
- using Monte-Carlo simulations and a semi-analytical analysis on isolated TS,
- the NGDBF gains require a precise optimization of σ_p.

PGDBF Decoders with Dynamic Shifts

- we proposed a modified PGDBF algorithm, with low complexity, and the ability to escape strong TS attraction,
- the DDS-PGDBF greatly improves the PGDBF performance, and approaches MLD using a very large number of iterations.