FEC LDPC codecs for deep space and QKD Reconciliation based on FPGA

R. Prosperococco[a], B. De Flaviis[b], D.Giancristofaro[c], L.Simone[c], V.Stornelli[b], G.Ferri[b], F.Santucci[a], L.Trotta[c]

[a]Università degli studi dell'Aquila (DISIM),
[b]Università degli studi dell'Aquila (DIEEE),
[c]Thales Alenia Space – Italia.

Abstract

- **Thales Alenia Space Italia** often implements FEC codec on FPGA,
- Long dated collaboration with ESA and academic institutions [1],[2], and CCSDS [3].
- TAS-I has implemented PCCC [3],SCCC [1] and **LDPC** codecs.
- As FPGA technology advances, **efficient high-speed designs require optimized use of hardware architectures**,
- Extrinsic information transfer requires a state-of-the-art trade off among random memory access and switch fabrics.
- Belief propagation for among processors and bit-decision nodes for SCCC, LDPC, and PCCC [4].
- LDPC is currently used in AWGN channels (deep space, ESPRIT [7],[8]), earth observation (PLATINO).
- The present study **implements LDPC on Binary Symmetric Channels (BSC)** for BB84 QKD Reconciliation.
- In-depth analysis of regular and irregular LDPC, using density evolution [5] and progressive edge growth [6] has allowed reducing simulation time via Tanner graph tree analysis [9].

Progressive improvement of LDPC code design

LDPC codes are classically designed following a few subsequent algorithms based on sound theoretical background:

1. Probability density evolution theory that leads to variable and check nodes optimal degree distribution (generating non uniform LDPCs);
2. Progressive edge growth (PEG) that connects edges among variable and check nodes avoiding cycles shorter than 4 (code girth);
3. Subsequent theories (Divsalar) leading to codes with guaranteed minimum distance increasing with block size: these codes belong to the RA, ARA, ARJA categories.

Precoding gain decreases for high rate codes for which the regular codes have already satisfactory performances. This is why the 7/8 or 223/255 code of CCSDS is the only to be a regular one.

Definition of CCSDS LDPC (n=8176, k=7154)

Parity Check Matrix (H)

H is structured as a 2 × 16 array of 511 × 511 circulants, where each row of each circulant contains exactly two '1's (positions defined in a table), resulting in a 1022 × 8176 matrix with rank 1020.

$$H = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} & A_{1,5} & A_{1,6} & A_{1,7} & A_{1,8} & A_{1,9} & A_{1,10} & A_{1,11} & A_{1,12} & A_{1,13} & A_{1,14} & A_{1,15} & A_{1,16} \\ A_{2,1} & A_{2,2} & A_{2,3} & A_{2,4} & A_{2,5} & A_{2,6} & A_{2,7} & A_{2,8} & A_{2,9} & A_{2,10} & A_{2,11} & A_{2,12} & A_{2,13} & A_{2,14} & A_{2,15} & A_{2,16} \end{bmatrix}$$

Derivation Process

The derivation of G from H requires computing the inverse of a matrix in GF(2) to obtain the zi vectors, which are then adjusted to form the first rows of the Bij circulants.

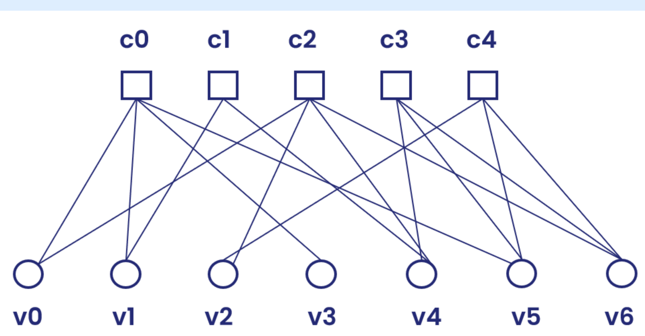
Generator Matrix (G)

G has a quasi-cyclic structure, where the left portion consists of a 7154 × 7154 identity matrix and the right portion includes two columns of 511 × 511 circulants.

$$G = \begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{1,1} & B_{1,2} \\ 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{2,1} & B_{2,2} \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{3,1} & B_{3,2} \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{4,1} & B_{4,2} \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{5,1} & B_{5,2} \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{6,1} & B_{6,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & 0 & B_{7,1} & B_{7,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & B_{8,1} & B_{8,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & B_{9,1} & B_{9,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & 0 & B_{10,1} & B_{10,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & 0 & B_{11,1} & B_{11,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & 0 & B_{12,1} & B_{12,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & 0 & B_{13,1} & B_{13,2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & I & B_{14,1} & B_{14,2} \end{bmatrix}$$

Tanner Graphs and node neighbourhoods

Tanner graphs provide a graphical representation of LDPC codes, modelling the **connections** between **variable nodes** and **check nodes** as defined by the parity-check matrix. Understanding the structure of these graphs is essential for analysing decoding performance and optimizing hardware implementations.



Tanner Graph ⟶ **enables the analysis of node neighbourhoods:** which examines how nodes are interconnected within a given depth (focusing on depth-2 and depth-4 neighbourhoods)

Error Patterns Selection (Trapping Sets)

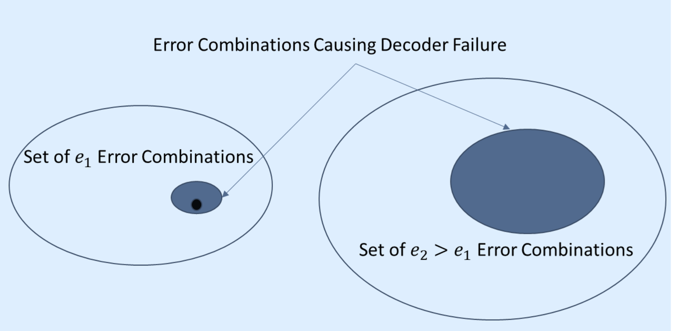
To accurately compute the Frame Error Rate of a codec, an **exhaustive simulation** should ideally be performed, testing all possible error patterns. ⟶ $FER = \frac{\text{Number of error patterns causing decoding failure}}{\text{Total number of error patterns}}$

Unfeasible: 50 errors on an 8176-bit codeword ⟶

$$\binom{50}{8176} \approx 1.2 \times 10^{116}$$

First Approach: *Structured Method for FER Semi-Analytical Estimation*

- Identification of the error patterns most likely to cause decoding failures: In general, since LDPC is designed to avoid **girth** smaller than 4, we expect that the most critical patterns occur when the errors are filled in paths within a depth-4 neighbourhood.
- Verification via Simulation of the **Threshold** : the minimum number of errors leading to failure.
- Analytical upper bound FER Computation: by comparing the number of pattern groups that lead to failure and the total.
- This analysis provides deeper insights into the properties of the LDPC code.



Second Approach: *Monte Carlo Analysis*

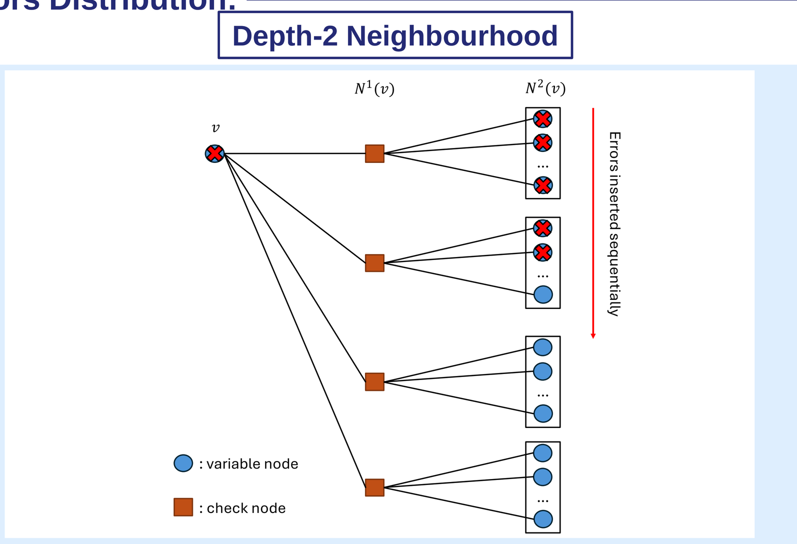
We performed numerous simulations by randomly injecting errors into codewords. By repeatedly simulating different error patterns, we gathered statistical data that allowed us to approximate the FER.

Error insertion strategy

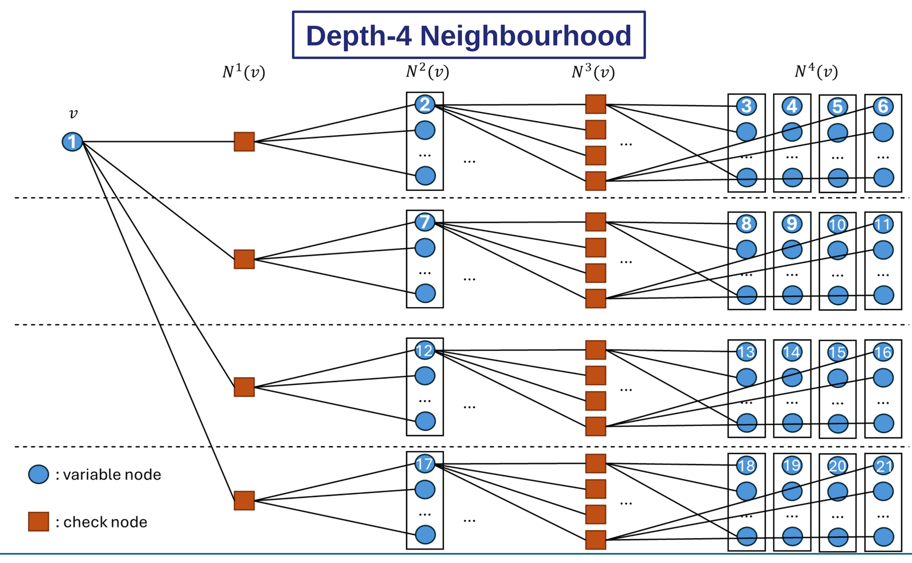
Errors were inserted into different depths of the Tanner graph, Then, errors were spread progressively across 1, 2, 3, and 4 neighborhoods

Errors Distribution:

Depth-2 Neighbourhood

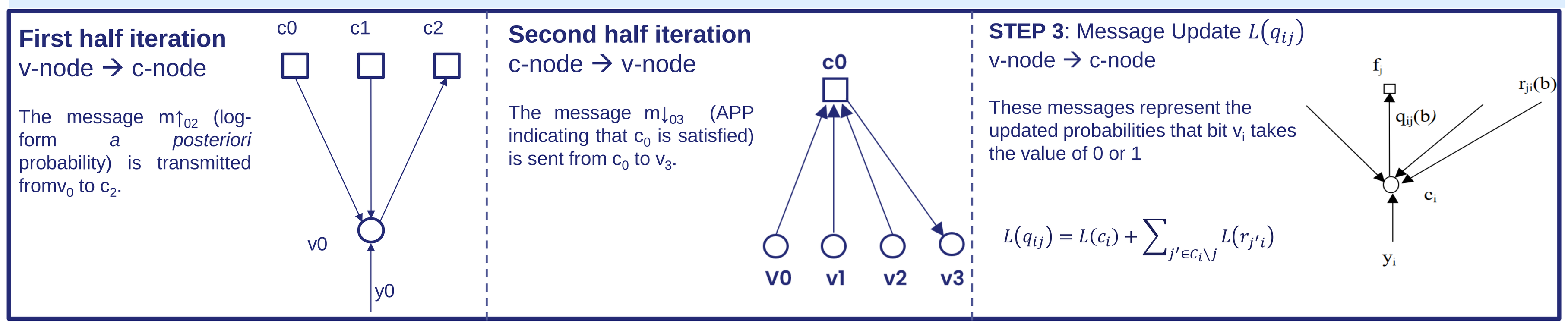


Depth-4 Neighbourhood

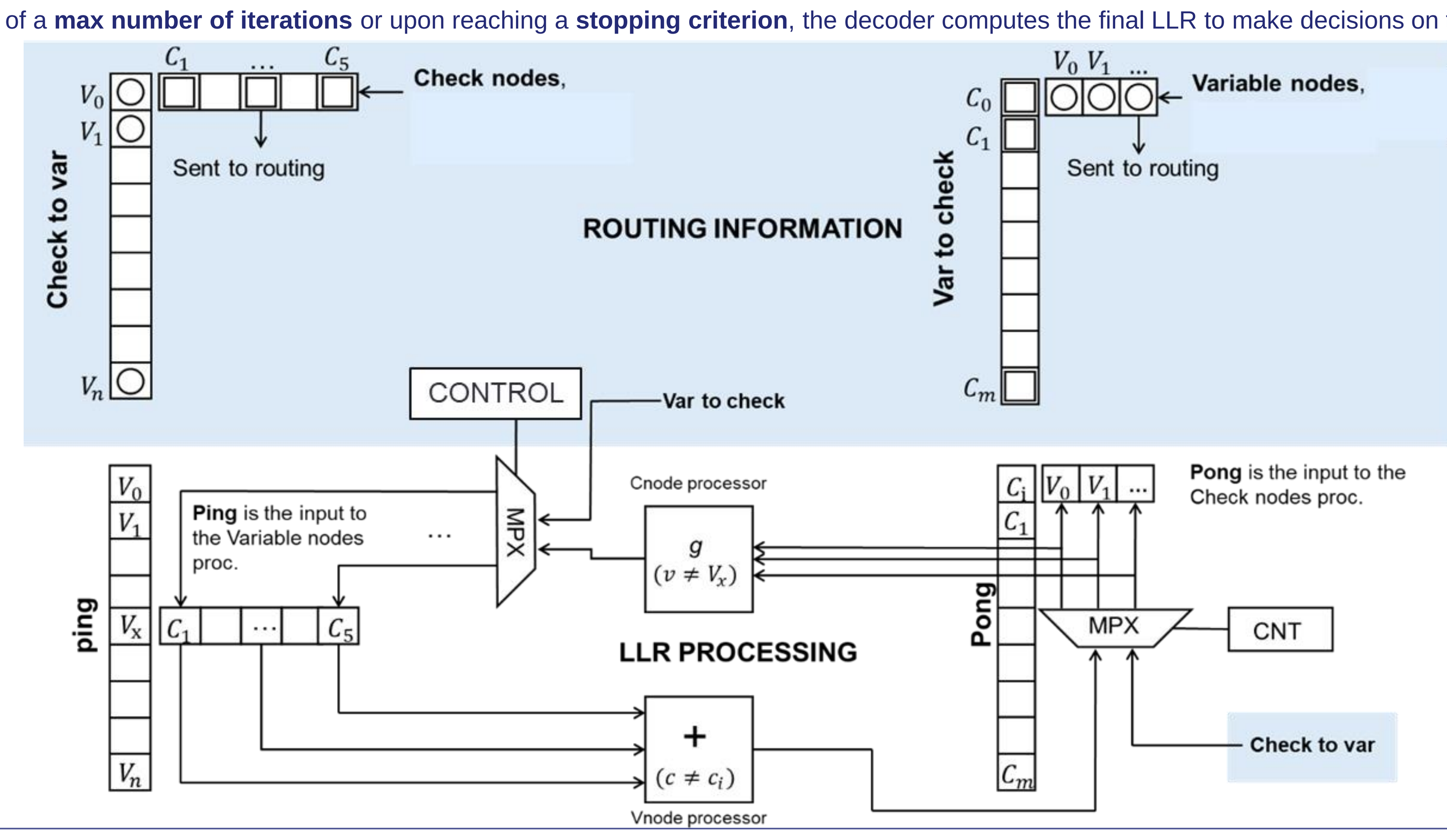


Belief Propagation for LDPC

- An **iterative approach** based on the Tanner graph of the code is used.
- The **variable nodes** represent the transmitted code bits that need to be estimated, while the **check nodes** represent the equations or parity constraints defined by the code.
- The **edges** represent the connections between variables and constraints, along which messages are exchanged.



At the end of a **max number of iterations** or upon reaching a **stopping criterion**, the decoder computes the final LLR to make decisions on the bits v_i.



Code Threshold and Trapping Sets Impact

To evaluate the performance of the LDPC decoder, we analyse the **Frame Error Rate** ⟶ $FER = \frac{N_{err}}{N_{tot}}$

where:

- N_{err} represents the number of error patterns that cause decoding failure
- N_{tot} is the total number of possible error patterns: $N_{tot} = \binom{e}{n} = \binom{e}{8176}$

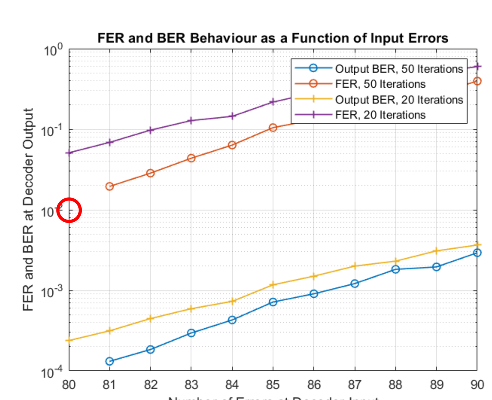
ERROR CASE	NUMBER OF ERRORS 62	63	64	65	66	67	68	69	70
1	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓	✓	✓	✓	✓
5	✓	✓	✓	✓	✓	✓	✓	✓	✓
6	✓	✓	✓	✓	✓	✓	✓	✓	✓
7	✓	✓	✓	✓	✓	✓	✓	×	×
8	✓	✓	✓	✓	✓	×	×	×	×
9	✓	✓	✓	✓	×	×	×	×	×
10	✓	✓	✓	✓	✓	✓	×	×	×

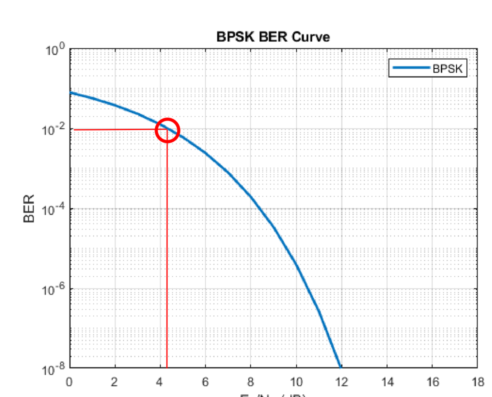
With this simulation, we observe that the decoder starts failing at **66 errors** and we can see which patterns cause decoding failures.

For each scenario, we define a variable representing the **total number of error patterns** with a specific distribution.

However, we consider only the **first occurrence** where a specific error pattern determines the **threshold** beyond which the decoder fails. This is because, once a given pattern causes failure at a certain number of errors, adding more errors will only require placing the new error in any other position among the variable nodes.

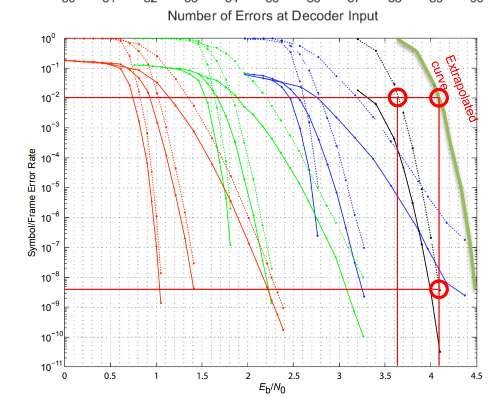
FER as a function of BER in (agreement with reference)

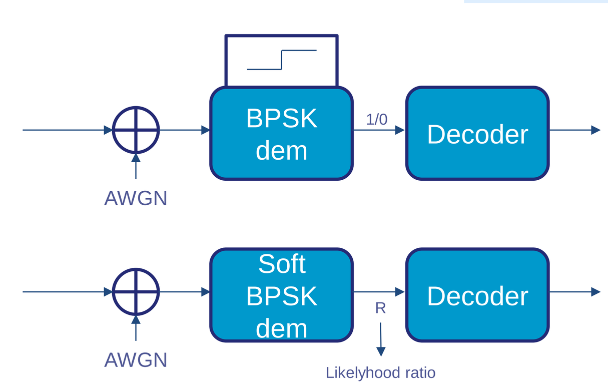




1. **A FER of 10^{-2}** can be extrapolated to correspond to 80 error bits, which is a BER of $9.7x10^{-3}$
2. A BER di $9.7x10^{-3}$ corresponds to an Eb/N0 of about **4.1dB**
3. In here, a FER of 10^{-2}corresponds to an Eb/N0 of **3.6dB**, that is about 0.5dB better than our calculated curve, This could be caused by the loss of information due to our use of a hard decoder.

The black curve is for code rate 7/8, solid is BER, dotted it FER, The other one is an extrapolation of our equivalent FER curve.

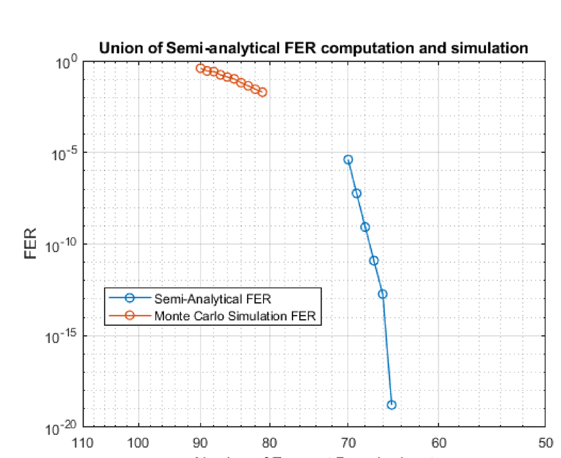




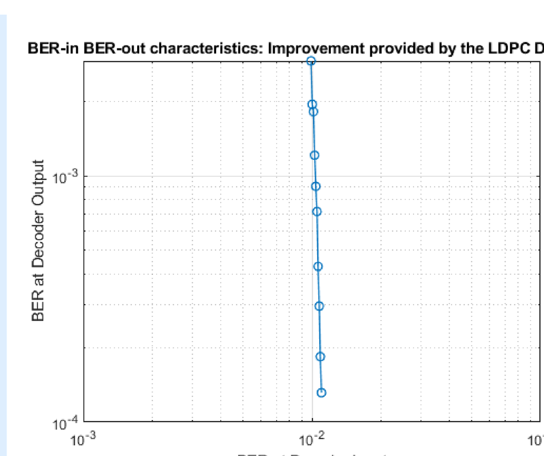
Some of the information content is lost during the hard quantization process, resulting in worse performances.

Most of the information content is maintained thanks to the soft demodulation into likelihood ratios.

Results



- Combination of semi-analytical and Montecarlo simulation FER
- Semi-Analytical method achieves very low FER levels that cannot be achieved by simulation
- Note: The semi-analytical is an upper bound



- BER-in BER-out performances of our decoder

Bibliography

[1] D.Giancristofaro, V.Piloni, R.Novello, R. Giubilei, J. Tousch: "Performances of Novel DVB-RCS Standard Turbo Code and its Applications in On-Board Processing Satellites"; IEEE EMPS-PIMRC 2000, London, 2000 .
[2] S. Benedetto, C. Berrou, C. Douillard, R. Garello, D. Giancristofaro, A. Ginesi, L. Giugno, M. Luise, G. Montorsi, "MHOMS: High Speed ACM Modem for Satellite Applications", IEEE Wireless Communications Journal, April 2005.
[3] S.Benedetto, G.Montorsi, A.Ginesi, D.Giancristofaro, M.Fonte: "A Flexible Near-Shannon SCCC Turbo Code for Telemetry Applications", ESA STR-250, 2005, ESTEC.
[4] E. Rossini, G. Chiassarini, A. Masci, R. Romanato, D. Silvi, P. Burzigotti, D. Giancristofaro: "AMPIST: An Advanced Modem Prototype For Interactive Satellite Terminals with Contention Resolution Diversity Slotted ALOHA (CRDSA) Capabilities", in proceedings of the 17th Edition of the Ka and Broadband Communications, Navigation and Earth Observation Conference, October 3-5, 2011 Palermo, Italy.
[5] EP3622642 B1 - Minimum-Size Belief Propagation Network for FEC Iterative Encoders and Decoders and Related Routing Method
[6] T. J. Richardson and R. L. Urbanke: "The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding", IEEE on IT, 2001.
[7] X.Y. Hu, E. Elefteriou, D.M. Arnold, "Regular and Irregular Progressive Edge Growth Tanner Graphs", IEEE on IT, Jan 2005.
[8] TM Synchronization and Channel Coding CCSDS 131.0-B-3, Blue Book, September 2017;
[9] W. H. Zhao, J. P. Long: "Implementing the NASA Deep Space LDPC Codes for Defense Applications", MILCOM 2013.

Email: riccardo.prosperococco@graduate.univaq.it, benedetta.deflaviis@student.univaq.it, domenico.giancristofaro@thalesaleniaspace.com