IMPROVED SYNDROME-BASED NEURAL DECODER FOR LINEAR BLOCK CODES

Gastón De Boni Rovella[1,2], Meryem Benammar[2], Hugo Meric[3], Tarik Benaddi[4]

[1]Laboratoire TéSA, Toulouse, France, [2]ISAE-SUPAERO, [3]CNES, [4]TAS
{gaston.de-boni-rovella, meryem.benammar}@isae-supaero.fr

1. System model

1.1. The decoding problem

Consider the following system:

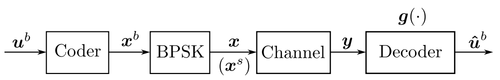

Figure 1: System model.

- $\boldsymbol{u}^b \in \{0,1\}^k$ the input message;
- $\boldsymbol{x}^b \in \{0,1\}^n$, $\boldsymbol{x}^b = \mathrm{G}\boldsymbol{u}^b$ the codeword mapped through a linear code $\mathcal{C}$;
- $\boldsymbol{x} \in \{-1,+1\}^n$ the BPSK modulated codeword;
- $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{z}$ the received signal, where $\boldsymbol{z} \sim \mathcal{N}(0, \frac{\sigma^2}{2}\mathbb{I}_n)$.

The *decoding problem* consists in producing a function $\boldsymbol{g}(\cdot)$ such that the message estimate $\hat{\boldsymbol{u}}^b \triangleq \boldsymbol{g}(\boldsymbol{y})$ minimizes the Bit Error Probability (BEP):

$$P_e^b \triangleq \frac{1}{k}\sum_{j=1}^{k} \mathrm{P}(\hat{U}_j^b \neq U_j^b). \quad (1)$$

1.2. Optimal decoder: Bit-MAP

This probability is minimized by the so-called Maximum A Posteriori (MAP) rule, given by:

$$g_j^{\star}(\boldsymbol{y}) = \mathbb{I}\Big\{\sum_{\substack{\boldsymbol{u}\\ u_j=1}} \mathrm{P}_{\boldsymbol{Y}|\boldsymbol{U}}(\boldsymbol{y}|\boldsymbol{u}) > \sum_{\substack{\boldsymbol{u}\\ u_j=0}} \mathrm{P}_{\boldsymbol{Y}|\boldsymbol{U}}(\boldsymbol{y}|\boldsymbol{u})\Big\}. \quad (2)$$

Complexity problem:
This decoder has an exponential complexity $\approx \mathcal{O}(2^k)$, and is thus too complex to be implemented in realistic applications.

2. Previous works

2.1. Equivalent noise model

The following equivalent noise model can be established:

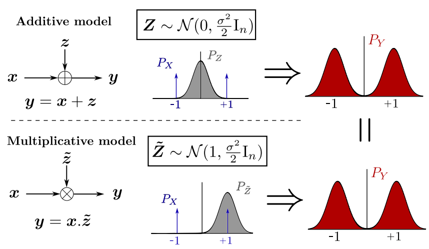

Figure 2: Equivalent noise model.

Thus, for a noise $\tilde{\boldsymbol{Z}} \sim \mathcal{N}(1, \frac{\sigma^2}{2}\mathbb{I}_n)$, the channel output can be expressed as follows:

$$\boldsymbol{Y} = \boldsymbol{X}.\tilde{\boldsymbol{Z}}, \quad (3)$$

and the *bit-flip* probability:

$$\mathrm{P}(Y^s \neq X) = \mathrm{P}(\tilde{Z} < 0). \quad (4)$$

2.2. Syndrome-based neural decoder

Bennatan et al. [1] proved the following result:

$$\mathrm{P}(X^b = x^b | \boldsymbol{Y} = \boldsymbol{y}) = \mathrm{P}(Z^s = xy^s | \, |\boldsymbol{Z}| = |\boldsymbol{y}|, \mathrm{H}\boldsymbol{Z}^b = \mathrm{H}\boldsymbol{y}^b), \quad (5)$$

establishing the following equivalence,

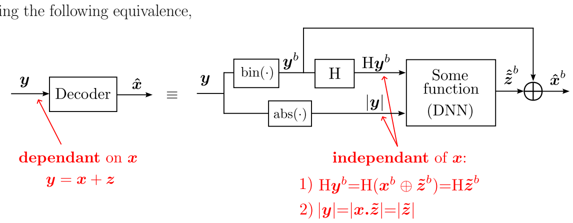

Figure 3: New equivalent method for estimating the *bit-flips*.

3. Our solution

3.1. Proposed system: improved syndrome-based neural decoder

To focus only on **information bits**, we proved the following results:

$$\mathrm{P}(\boldsymbol{U}^b = \boldsymbol{u}^b | \boldsymbol{Y} = \boldsymbol{y}) = \mathrm{P}(\boldsymbol{W}^s = \boldsymbol{u}^s \tilde{\boldsymbol{u}}^s | \, |\boldsymbol{Z}| = |\boldsymbol{y}|, \mathrm{H}\boldsymbol{Z}^b = \mathrm{H}\boldsymbol{y}^b), \quad (6)$$

where $\tilde{\boldsymbol{u}} = \mathrm{pinv}(\boldsymbol{y}^b)$, which yields the following proposed system [3]:

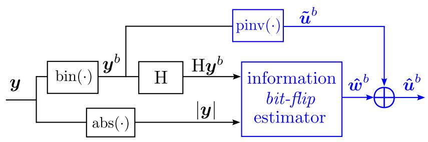

Figure 4: System that estimates **information** *bit-flips*.

3.2. Implementation of the bit-flip estimator: RNN

The *bit-flip* estimator is implemented using Recurrent Neural Networks (RNN):

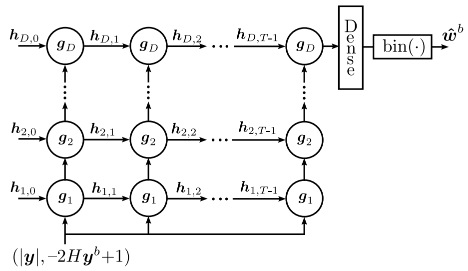

Figure 5: Implementation of the **information** *bit-flip* estimator.

3.3. Numerical results

The proposed solution [3] was implemented for two polar codes of size $(64,32)$ and $(128,64)$, and for a BCH code of size $(63,51)$. It was compared with the solutions in [1] and [2], which use the previous framework of Figure 3.

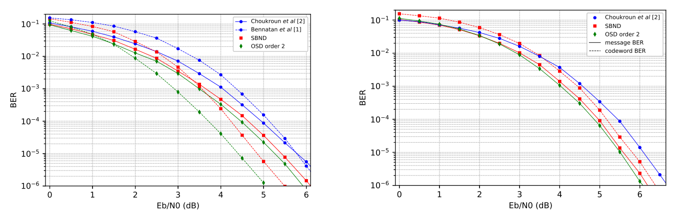

Figure 6: Bit Error Rate for two polar codes of sizes (64,32) and (128,64) (**left**) and a BCH of size (63,51) (**right**).

4. Conclusions

Our system generalized the previous work of [1], with three main aspects to be considered:

1. it improves the decoding accuracy by focusing on minimizing the error over the **information bits**;
2. it can be directly applied to any linear code, either **systematic** or **non systematic** and;
3. the **single-codeword** training property is preserved.

5. References

[1] A. Bennatan, Y. Choukroun and P. Kisilev, "Deep Learning for Decoding of Linear Codes - A Syndrome-Based Approach," 2018 IEEE International Symposium on Information Theory (ISIT), Vail, CO, USA, 2018, pp. 1595-1599, doi: 10.1109/ISIT.2018.8437530.

[2] Y. Choukroun and L. Wolf, "Error Correction Code Transformer," Adv. Neural Inf. Process. Syst., vol. 35, pp. 38 695–38 705, 2022.

[3] G. De Boni Rovella and M. Benammar, "Improved Syndrome-based Neural Decoder for Linear Block Codes," 2023 IEEE Global Communications Conference. [Online]. Available: https://www.tesa.prd.fr/documents/26/improved_syndrome-based_neural_decoder_for_linear_block_codes.pdf