



Revisiting augmented decoding techniques for LTE Turbo Codes

Titouan Gendron, Emmanuel Boutillon, Charbel Abdel Nour, David Gnaedig

► To cite this version:

Titouan Gendron, Emmanuel Boutillon, Charbel Abdel Nour, David Gnaedig. Revisiting augmented decoding techniques for LTE Turbo Codes. ISTC 2021: 11th IEEE International Symposium on Topics in Coding, Aug 2021, Montreal, Canada. 10.1109/ISTC49272.2021.9594119 . hal-03280680

HAL Id: hal-03280680

<https://hal.science/hal-03280680>

Submitted on 7 Jul 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Revisiting augmented decoding techniques for LTE Turbo Codes

Titouan Gendron
PHD student
Université Bretagne Sud
F-56100 Lorient, France
titouan.gendron@univ-ubs.fr

Emmanuel Boutillon
Université Bretagne Sud
Lab-STICC
UMR CNRS 6285
F-56100 Lorient, France
emmanuel.boutillon@univ-ubs.fr

Charbel Abdel Nour
IMT Atlantique
Lab-STICC
UMR CNRS 6285
F-29238 Brest, France
charbel.abdelnour-
@imt-atlantique.fr

David Gnaedig
TurboConcept
F-29238 Brest, France
david.gnaedig-
@turboconcept.com

Abstract—This paper revisits several methods used in iterative decoders to improve the performance of the LTE Turbo code for small information blocks. By considering the iterative decoder as a chaotic system, some explanations are given on the unexpected behavior of several post-processing algorithms. A "dither-inspired" algorithm, called blind candidate decoding, is proposed. This algorithm improves error correction by performing several decoding attempts from a set of perturbed input. Combined with the use of the CRC as an outer code, a performance improvement of up to 1 dB in signal to noise ratio can be obtained for a frame error rate of 10^{-3} . Moreover, a stopping criterion based on the Euclidean distance between the output and the input is proposed to mitigate the critical problem of undetected errors.

Index Terms—Turbo Codes, candidate decoding, Flip and Check, Forced Symbol Methods

I. INTRODUCTION

The Internet of Things (IoT) generally requires the use of particular error correcting codes, where error correction performance at low Signal to Noise Ratio (SNR) is crucial due to the transmitter's limited energy supply. The SNR range where the Frame Error Rate (FER) decreases rapidly is called the waterfall region, in opposition with the error floor region limited to high SNRs. Turbo Codes [1] are one of the error correcting codes that can be used for IoT application [2]. This paper focuses on the Long Term Evolution (LTE) Forward Error Correction (FEC) code. It includes two binary Recursive Systematic Convolutional (RSC) codes separated by an interleaver, a rate matching system and a Cyclic Redundancy Check (CRC) for outer error detection [3].

Several methods have been proposed in the literature to achieve better decoding performance with a turbo decoder without modifying the design of the code at the transmitter side. Some of them rely on the CRC to achieve lower FER, by computing several possible hard decisions that can be validated by the CRC. The most intuitive method is the Flip and Check (FC) algorithm [4]. If the hard decision of the decoder doesn't pass the CRC check, the FC algorithm consists in first selecting the q_{FC} less reliable bits, then generating all $2^{q_{FC}}$ candidate CRC codewords by flipping decoded values of these bits, and finally evaluating the CRC check on all candidates, keeping the one that passes the CRC. This method can improve the performance of the LTE

decoder in the waterfall region, but it increases the number of undetected errors (wrong codewords that satisfy the CRC), leading to a higher FER floor.

List decoding [5] is another method that can be used to improve the performance of codes, generating a list of possible codewords and choosing the one that satisfies the CRC constraints. Unlike FC, List decoding computes several hard decisions by modifying the turbo decoding algorithm. This modification generates several sets of extrinsic outputs by each component decoder using a reliability classifier algorithm like the List Viterbi Algorithm [6] or the 2^k method [7]. Processing these multiple candidate extrinsic sets for the same number of decoding iterations would lead to an increased latency unless multiple list Soft Input Soft Output (SISO) decoders are introduced with their associated added complexity.

Candidate decoding, consists in making several hypotheses on the received information and trying to decode them, leading to a plurality of standard decoding processes. Such algorithm was first introduced in [8] and later named Forced Symbol Method (FSM) [9]. By selecting different hypotheses on the set of the q_b less reliable bits, it provides 2^{q_b} candidates that span all corresponding possibilities. Each decoding process assumes these q_b bits as reliable, i.e. their input Log Likelihood Ratio (LLR) is set to plus or minus infinity. A standard decoding process is required to choose these q_b bits, and the number of candidates must be a power of 2. Another type of candidate decoding have also been proposed in [10] for Low Density Parity Check (LDPC) codes: The Randomized Initial State (RIS). Unlike FSM, the RIS algorithm consists in several decoding attempts with randomly perturbed input LLRs. To the best of the authors' knowledge, this idea has not yet been applied for turbo decoders. Inspired by these different variants of the candidate decoding algorithm and the FC algorithm, we propose in this work a way to combine different methods to obtain a performance gain for short blocks LTE decoders.

The rest of the paper is organized as follows : Section II analyzes candidate decoding. Section III presents a new candidate decoding algorithm. Section IV presents a study on the association of the new algorithm with the Flip and Check

algorithm and introduces a criterion based on the Euclidean distance to avoid undetected errors. Section V provides the performance of the considered algorithms through different examples. Section VI concludes the paper.

II. SYSTEM DESCRIPTION AND ANALYSIS OF CANDIDATE DECODING

A. System Description

The LTE Turbo encoder applies two component RSC encoders separated by an interleaver, each generating K redundancy bits from the K information bits. The decoder is then made of two component decoders separated by the interleaver. Each component decoder generates extrinsic output from the received information and its *a-priori* input. The extrinsic output of a component decoder is interleaved to be used as *a-priori* input by the other component decoder at the next iteration.

B. Analysis of Candidate Decoding

The Forced Symbol Method (FSM) [9] is a candidate decoding algorithm consisting in making several decoding attempts from different initial states. After an unsuccessful decoding attempt, a metric is used to select the q_b bits that are more likely erroneous, and 2^{q_b} different input information sets are built, corresponding to all their possible binary combinations. In these input information sets, the q_b least reliable bits are saturated, i.e. their LLR values are set to the maximum or minimum (quantized) value of the decoder.

At a first glance, it seemed likely that the choice of the q_b bits would be crucial for this algorithm and would play a major role in the performance of the candidate decoding. For example, the selection of q_b least reliable bits in [9] is based on a metric that calls upon the LLR difference after decoding (see [9] for details). In order to evaluate the impact of the bit selection method, we have compared their impact using a LTE decoder with floating point precision for different cases of block lengths and code rates ($K = \{40, 256\}$, $R = \{1/3, 9/10\}$), and for different strategies for the choice of the q_b selected bits. Compared methods allow or not the selection of redundancy and punctured bits separately. The list of the methods used to select the q_b bits is the following:

- Select randomly the q_b bits.
- Select the q_b least reliable received bits (i.e. with the smallest absolute LLR value).
- Use the metric described in [9].

After simulating all cases, surprisingly all methods were equivalent in terms of performance. In order to verify if the performance gain of the candidate decoding is due only to the multiple decoding attempts and not to the selection method of the candidates, we performed the following experiment: A rate $R = 1/3$, $K = 40$ all-zero codeword with an LTE turbo code is transmitted over an Additive White Gaussian Noise (AWGN) channel at a signal to noise ratio $\frac{E_b}{N_0}$ of 2.75 dB until an error event occurs after 8 decoding iterations. Using this erroneous frame, we performed several decoding attempts after modifying only the *a priori* LLR L_{12} and L_{18} of the 12th and the 18th systematic bits (the LTE rate

matching system punctures these two bits but it can be any couple of bits). In Fig.1.a), the couple (L_{12}, L_{18}) takes values in $[-2.0, 2.0]$ with a step of 0.002, thus generating 1000×1000 distinct combinations. For each combination, the decoding process is executed and if the decoded codeword gives the two decoded bits $(\hat{b}_{12}, \hat{b}_{18})$ equal to $(0, 0)$, the corresponding pixel in Fig. 1.a) is colored in white. Similarly, if $(\hat{b}_{12}, \hat{b}_{18}) = (0, 1), (1, 0)$ and $(1, 1)$, the corresponding pixel is colored in blue, red and magenta, respectively. Fig. 1.a) shows no clear border between decision areas. It is still the case when a zoom is performed, as shown in Fig. 2.b) where (L_{12}, L_{18}) takes there values in $[-0.25, 0.25]$.

In Fig. 1, we observe a chaotic, and to some extent, a fractal frontier between the different areas. We have two explanations to this observation. The first one is due to the nature of iterative processing where each iteration is a non linear function using the result of the previous iteration, thus amplifying the small variations between the initial LLRs. This effect is known as the "butterfly effect" [11] in the chaos theory. The second explanation can be our incapacity to fully grasp the subtle geometry of a highly dimensional vector space. These hypotheses have yet to be verified, which could be an interesting research topic.

The chaotic behavior of LDPC codes has already been studied in [12] through the analysis of LDPC Belief Propagation decoding algorithm as a dynamic system. The methods used in [12] could be used to study the chaotic nature of iterative decoding using a different approach than the one presented above. In any case, we can see that there is no clear frontier between a decision or another, and that it is possible that a candidate is successfully decoded even if the hypothesis drives away from the original codeword, and vice versa. Then each candidate decoding attempt can be seen as a pick of a random variable with the following 3 possible realizations with unknown probability distribution:

- The candidate leads to the right codeword
- The candidate leads to a wrong codeword
- The candidate doesn't lead to a codeword.

Increasing the number of decoding attempts increases simultaneously the probability to successfully decode on one side, and the undetected error probability on the other side.

With these results in mind, we propose a new candidate decoding algorithm that randomly samples candidates in the vicinity of the received information.

III. PROPOSED BLIND CANDIDATE DECODING

In 2012, [10] made the same assessment on LDPC codes and presented a candidate decoding algorithm called Random Initial State (RIS). This algorithm consists in performing several decoding attempts with different initial states randomly generated. [10] showed that such algorithm increases the convergence probability, and presented a method to reduce the undetected error probability. This method consists in storing every codeword found by the candidate algorithm and in keeping only the closest one to the received LLR.

The proposed candidate decoding algorithm, called Blind Candidate Decoding (BCD), is similar to RIS and consists in

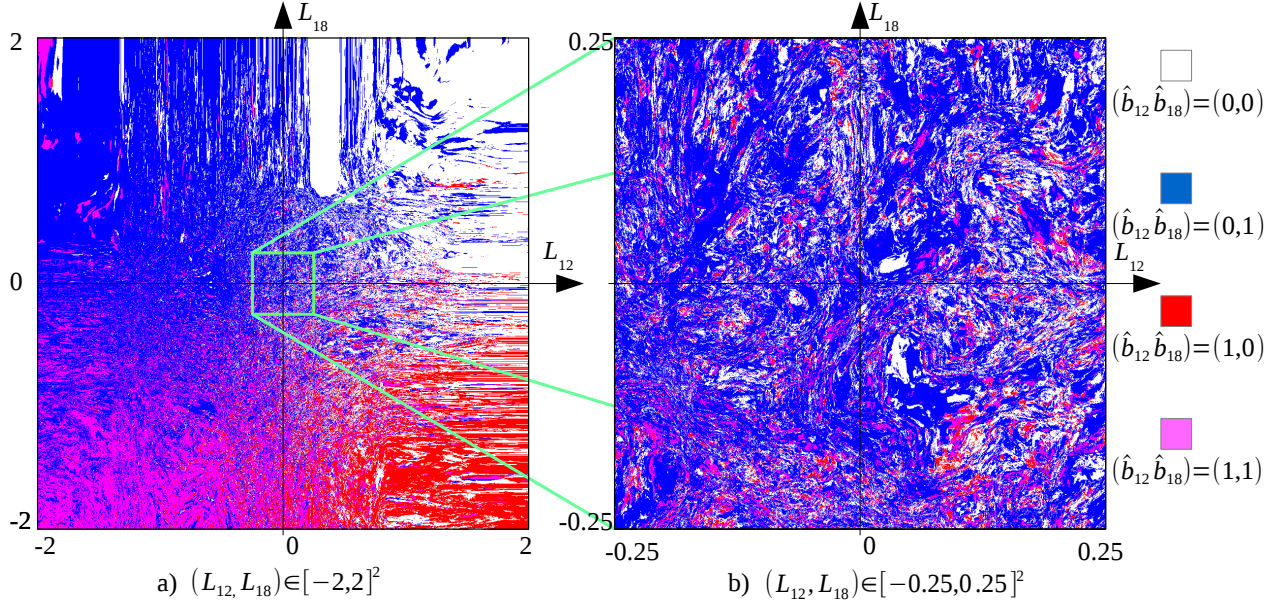


Fig. 1. Representation of the convergence of a LTE code in function of the added perturbation on the 12th and the 18th bit

generating candidates by introducing random perturbations to the received information. This method implies the creation of a set of candidates within an Euclidean sphere around the original received information. For a given number of iterations and number of candidates, we can estimate by Monte Carlo simulation the optimal distance d_c between each candidate and its corresponding received information that maximizes the convergence probability of the whole process.

Let n_c be the number of random candidates generated for the BCD algorithm. Consider a (N, K) LTE turbo code, let Y_0 be the original received LLR input vector. This latter is a N size vector and $Y_0(k), k \in \llbracket 0, K-1 \rrbracket$ are the LLR values of the systematic bits, while $Y_0(k), k \in \llbracket K, N-1 \rrbracket$ represent the LLRs of the redundancy bits. Note that if the k^{th} bit is punctured, $Y_0(k) = 0$. Let G_γ be a vector of size K filled with equally spaced values centered on 0:

$$G_\gamma(k) = \gamma \frac{2k}{K-1} - 1, k \in \llbracket 0, K-1 \rrbracket, \quad (1)$$

where γ is a parameter that controls the value of the norm $d_\gamma = \gamma^2 E_K$ of the vector G_γ . The constant E_K is given by

$$E_K = \frac{K}{3} \left(\frac{2}{K-1} + 1 \right). \quad (2)$$

One possible set of candidates can be obtained by the addition of a random perturbation of magnitude G_γ to the LLRs of the systematic part of Y_0 , such that each systematic bit is randomly affected by an additive noise, keeping each candidate codeword within a constant value of the Euclidean distance d_γ to the original received information vector. Algorithm 1 gives the details of the proposed method.

Note that in this algorithm, the distribution type of the G_γ vector isn't important as long as it is symmetrically centered around 0. The optimal distance d_γ for a given

Algorithm 1: Blind Candidate Decoding

```

input :  $Y_0, \gamma$ 
output: hard decision
 $\hat{C} = \text{Decode}(Y_0)$ ;
if  $\text{CRC\_check}(\hat{C})$  then
  | RETURN  $\hat{C}$ 
end
 $G_\gamma = \gamma \frac{2k}{K-1} - 1, k \in \llbracket 0, K-1 \rrbracket$ ;
for  $i \leftarrow 0$  to  $n_c - 1$  do
   $\Pi = \text{Random\_Permutation}(\llbracket 0, K-1 \rrbracket)$ ;
  for  $k \leftarrow 0$  to  $K-1$  do
    |  $Y'(k) = Y_0(k) + G_\gamma(\Pi(k))$ ;
  end
   $\hat{C} = \text{Decode}(Y')$ ;
  if  $\text{CRC\_check}(\hat{C})$  then
    | RETURN  $\hat{C}$ 
  end
end

```

set of parameters and a block length can be determined by Monte-Carlo simulation at the targeted SNR. The number of candidates denoted by n_c and the number I_C of decoding iterations per candidates can also be adjusted, given the constraints on throughput and latency. The values of the parameter γ and the couple (n_c, I_C) can be optimized separately since the most suitable value for γ remains the best for different couples of (n_c, I_C) .

The BCD algorithm is more versatile than the FSM [9] algorithm because no prior classical decoding is necessary to build the set of candidates, and the number of candidates is not required to be a power of 2. With more degrees of freedom, it is possible to finely optimize the algorithm parameters

for a given architecture. In general, BCD outperforms FSM in most practical cases. More detailed results are presented in the fifth part of this paper.

IV. IMPROVED FLIP AND CHECK ALGORITHM AND THE PROPOSED EUCLIDEAN DISTANCE-BASED CRITERION

For further performance gain, any candidate decoding can be used with a post processing method such as the Flip and Check (FC) algorithm. This latter was first introduced in [4] to lower the error floor of the LTE code, but it can also be used to improve performance in the waterfall region. It consists in trying several hard decision codeword candidates with CRC during each decoding iteration. To do so, the q_{FC} bits with the lowest *a-posteriori* probabilities are chosen, then the $2^{q_{FC}}$ possible hard decisions on these bits are tested. This method was already used in [13] for candidate decoding, with the particularity to choose the q_{FC} candidates from the critical positions of bits. The main drawback of this algorithm is that, combined with a candidate decoding algorithm, it increases significantly the undetected error rate i.e. the probability to have a wrong codeword satisfy the CRC. If no other stopping criterion than the CRC is used on a BCD FC decoder, an error floor will appear around $FER = 10^{-3}$. This floor is due to the high proportion of wrong codewords that satisfy the CRC. When the FC is used on classic decoding algorithms, the first iterations are usually

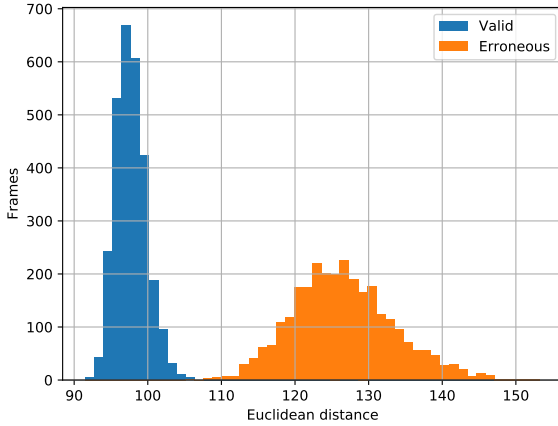


Fig. 2. Distribution of the Euclidean distances between the re-encoded hard decision and the received information for $K = 40$, $R = 9/10$, $E_b/N_0 = 4.0$ of 2850 successfully decoded frames (valid) and 2850 unsuccessfully decoded frames (erroneous) satisfying the CRC.

In order to avoid undetected errors, we propose another criterion based on the Euclidean distances. In case of an undetected error, the CRC is verified on a codeword containing erroneous hard decisions of systematic bits (i.e. decoded bits). However, the encoded codeword obtained by re-encoding these systematic hard decisions should be far in terms of Euclidean distance from the received codeword LLR information. Indeed, the redundancy bits obtained after re-encoding have a high probability to diverge from the original coded message. It is also possible to predict *a-priori* the distribution of the distances between the received information and the original coded message, knowing the SNR.

In Fig. 2, the distribution of the Euclidean distances between the encoded hard decision and the received information is plotted. We can see a clear separation between the Euclidean distance of the successfully decoded codewords and the Euclidean distance of codewords with undetected errors. Hence, undetected errors can be avoided if we only take the hard decision on information bits that gives an encoded codeword that is close enough to the received information, i.e. if the Euclidean distance is below a certain threshold.

V. PERFORMANCE COMPARISON AND RESULTS

In this section, a performance comparison between several candidate decoding algorithms using the standard LTE decoder is presented. It has been performed for 4 cases listed in Table I and Table II. For the results presented in this paper, the code rate values do not take into account the 24 redundancy bits of the CRC. To fairly compare FSM and BCD with or without FC, the total number of iterations I_{max} is kept unchanged: $I_{max} = 64$ or $I_{max} = 256$. Let q_b be the number of tested bits by the FSM, and n_c the number of tested candidates by the BCD. Several couples (q_b, I_C) for FSM and (n_c, I_C) for BCD are possible as long as, respectively, $I_C 2^{q_b} \leq I_{max}$, and $I_C n_c \leq I_{max}$.

A first interesting result is the selection of the best parameters for each type of decoding algorithm. They are chosen by identifying a sample of undecoded codewords after 8 iterations of the reference decoder. Then these codewords are used for another decoding run with all possible combinations of the parameters ($\gamma \in \{0.05; 0.1; \dots; 1\}$). Chosen parameters are those which successfully decode the largest proportion of these codewords. The results are shown in Table I.

TABLE I
TABLE OF OPTIMUM PARAMETERS

Case	I_{max}	FSM		BCD		
		q_b	I_C	n_c	I_C	d_c
$K = 40$	64	3	8	16	4	10.1
$R = 1/3$	256	5	8	32	8	14.0
$K = 40$	64	5	2	64	1	4.24
$R = 9/10$	256	7	2	256	1	5.92
$K = 256$	64	3	8	8	8	17.4
$R = 1/3$	256	3	32	16	16	17.4
$K = 256$	63	5	2	32	2	7.74
$R = 9/10$	256	7	2	128	2	10.5

For the cases with high code rates, a limited number of iterations is enough to have a large proportion of these candidates decoded, as shown in Table I. Indeed, only one iteration is generally enough to see if the candidate method will converge or not for short blocks. In that case, a criterion on the Euclidean distance is required to avoid a too high undetected error rate, even without FC. The other interesting information is that the distance d_c of the BCD candidates can be larger for low coding rates.

When applying the FC algorithm, the same parameters can be re-used since the specifically optimized parameters for FC have values quite close the selected parameters without FC. Moreover, the performance gained by specifically optimizing the parameters for FC is negligible. For the performance comparison of Fig 3, 8 bits are flipped by the FC algorithm for each iteration.

VI. CONCLUSION

In this paper, a new perspective on candidate decoding for Turbo Codes was presented through potential synergies with the chaotic nature of the non-optimal iterative decoding.

A new candidate decoding algorithm denoted by Blind Candidate Decoding is proposed. In addition to an improved performance at low SNR values, it reveals to be less complex than reference state of the art alternatives.

Coupling candidate decoding with the Flip and Check algorithm can improve performance at the price of unacceptable undetected error rate. As a solution, a new stopping criterion based on the values of the Euclidean distance between the received information and the encoded hard decision of the decoder was introduced. Indeed, it was shown that a distinct separation exists between the Euclidean distance values for valid and erroneous codewords that can be simply exploited through a comparison with a well defined threshold. The value of this threshold can be computed for a given block length, code rate and SNR. This comparison reveals to be quite efficient to avoid undetected errors.

Future work includes a deeper analysis of the behaviour of iterative decoding using tools from the complex dynamic systems theory and the chaos theory.

REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *IEEE Int. Conf. on Commun.(ICC)*, volume 2, pages 1064–1070 vol.2, May 1993.
- [2] M. Chen, Y. Miao, Y. Hao, and K. Hwang. Narrow band internet of things. *IEEE Access*, 5:20557–20577, 2017.
- [3] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC). Technical Specification (TS) 36.331, 3rd Generation Partnership Project (3GPP), 04 2017. Version 14.2.2.
- [4] T. Tonnellier, C. Leroux, B. Le Gal, B. Gadat, C. Jégo, and N. Van Wambeke. Lowering the error floor of Turbo Codes with CRC verification. *IEEE Wireless Commun. Lett.*, 5(4):404–407, Aug 2016.
- [5] K. R. Narayanan and G. L. Stuber. List decoding of turbo codes. *IEEE Trans. on Commun.*, 46(6):754–762, 1998.
- [6] N. Seshadri and C. . W. Sundberg. List viterbi decoding algorithms with applications. *IEEE Trans. on Commun.*, 42(234):313–323, 1994.
- [7] C. Nill and C. . W. Sundberg. List and soft symbol output viterbi algorithms: extensions and comparisons. *IEEE Trans. on Commun.*, 43(2/3/4):277–287, 1995.
- [8] Y. Ould-Cheikh-Mouhamedou, S. Crozier, K. Gracie, P. Guinand, and P. Kabal. A method for lowering Turbo Code error flare using Correction Impulses and Repeated Decoding. In *Int. Symp. on Turbo Codes and Iterative Info. Process. (ISTC)*, pages 1–6, April 2006.
- [9] Y. Ould-Cheikh-Mouhamedou and S. Crozier. Improving the Error Rate performance of Turbo Codes using the Forced Symbol Method. *IEEE Commun. Lett.*, 11(7):616–618, July 2007.
- [10] F. Leduc-Primeau, S. Hemati, S. Mannor, and W. J. Gross. Dithered Belief Propagation Decoding. *IEEE Trans. on Commun.*, 60(8):2042–2047, August 2012.
- [11] Étienne Ghys. The butterfly effect. In Sung Je Cho, editor, *Int. Congress on Math. Education*, pages 19–39, Cham, 2015. Springer International Publishing.
- [12] Jean-Christophe Sibel, Sylvain Reynal, and David Declercq. Evidence of chaos in the Belief Propagation for LDPC codes. *Chaotic Modeling and Simulation Journal*, pages pp 3–18, ISSN 2241–0503, January 2013. 16 pages.
- [13] S. Weithoffer and N. Wehn. Enhanced decoding for high-rate LTE Turbo-Codes with short block lengths. In *2017 IEEE Int. Conf. on Commun. Workshops (ICC Workshops)*, pages 967–972, May 2017.

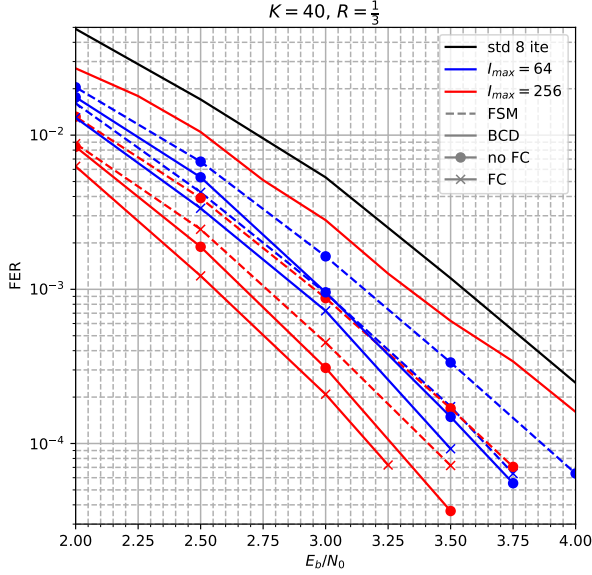


Fig. 3. FER for $K = 40$, $R = 1/3$ of FSM and BCD with and without FC, compared to simple Turbo decoder 8 iterations

TABLE II
PERFORMANCE GAIN IN DB FOR FER= 10^{-3}

Case	l_{max}	FSM		BCD	
		no FC	FC	no FC	FC
$K = 40$ $R = 1/3$	64	0.39	0.57	0.56	0.65
	256	0.61	0.81	0.87	9/108
$K = 40$ $R = 9/10$	64	2.86	4.05	2.81	4.22
	256	3.19	4.49	3.57	4.86
$K = 256$ $R = 1/3$	64	0.08	0.08	0.14	0.16
	256	0.19	0.22	0.25	0.24
$K = 256$ $R = 9/10$	63	0.45	0.97	0.81	1.10
	256	0.59	1.12	1.09	1.33

The performance gain of the different algorithms, for each simulation case, in comparison to 8 iterations of the conventional turbo decoding is presented in Table II. We can see that the BCD algorithm is almost always better than the FSM, especially when combined with FC. Candidate decoding is globally more efficient when used on short block sizes and high coding rates. The performance of the $K = 40$, $R = 1/3$ case is provided in Fig. 3. In this case, the decoding process of a frame ends if the hard decision of the decoder meets the CRC or if a codeword below the chosen threshold for the Euclidean distance is found by the FC. The value of this threshold is chosen to be larger than the one obtained by 99.99% of the valid codewords. No undetected error occurred for the curves in Fig. 3. Moreover, 300 erroneous frames were detected for each point of the curves. Hence, we can reasonably predict that the undetected error rate is expected to be, at least, 2.5 decades below. We can also mention that, for FER below 10^{-3} , the average number of iterations is noticeably lower with BCD than FSM, and FC can reduce it even further.