Influence of the Variation of Parameters on the Turbo Codes' Performances

Adil El Makrani
Laboratoire d'Electronique, Signaux, Systèmes et Informatique (LESSI)
Faculté des Sciences Dhar El Mahraz Fès (FSDM) USMBA

Mostafa Mrabti
Laboratoire Sciences de l'Information et Systèmes (LSIS), ENSAF USMBA

ABSTRACT
This paper aims at illustrating the influence of the change of parameters on the performances of the Turbo Codes in a transmission chain with AWGN. To this end, we used the MAP algorithm as a decoding algorithm. This is how we found that the variation in some parameters necessary in the construction of the Turbo Code such as the width of the information frame, the use of High Stress Length Decoding, the use of decoding in low yields, the choice of the generating code, and finally the increase of the number of iterations improved significantly the performances.

General Terms
Signal and computer security.

Keywords
Turbo code, Algorithm MAP, Interleavers.

1. INTRODUCTION
The Turbo Codes were invented in 1991, and presented to the scientific community in 1993 by a team from the Superior National School of Telecommunications of Brest directed by Claude Berrou and Alain Glavieux.

Specialists of error correcting codes have first welcomed this invention with great scepticism, because of extraordinary performances it announced. However, in short time period, other teams around the world reached the same results, which contributed to the development of Turbo Codes. Yet, all the results related to these codes have been established for the moment only experimentally.

The paper is divided into three main parts. In section 2, we present the main characteristics of turbo codes. We definitions turbo encoding structure based on the parallel concatenation of recursive systematic convolutional codes (RSC) constraint length k and a random interleaver. Section 3 treats the iterative decoding of turbo codes, and finally Section 4 gives some simulation results when sending a bit sequences through AWGN channels, then compare the different results when varying certain parameters.

2. PRINCIPLE OF TURBO CODES
This new family of codes is the work of Berrou et al[1]. Initially, their construction was composed of two recursive systematic convolutional codes concatenated in parallel and connected via an interleaver (Fig. 1). These codes permit to approach significantly the limit capacity of Shannone equal to 10^{-5} for binary rate error, especially through the use of iterative decoding algorithms. Using a variant of the BCJR algorithm we extract information on each of the codes and we exchange it between them[4]. The information extracted is called "extrinsic" and is fed back to the next iteration in another component code. When this process is repeated several times, we speak of iterative decoding. These decoding algorithms apply to both types of turbo construction: parallel turbo codes and serial turbo codes.

2.1 Turbo encoder (parallel Turbo codes)
To illustrate the parallel construction, we consider here the use of two recursive systematic convolutional codes (RSC1 and RSC2 in Fig. 2.) which are represented using a cyclic lattice. In practice, we often use two identical codes RCS1 and RSC2 connected by an interleaver.

The information bits Si are first encoded by the RSC1 encoder to provide a first redundancy x_i^{p1} then these bits are interlaced before being encoded by the encoder RSC2 to deliver a second redundancy: x_i^{p2} . The message to be transmitted is therefore the concatenation of information Si with x_i^{p1} and x_i^{p2} redundancies.

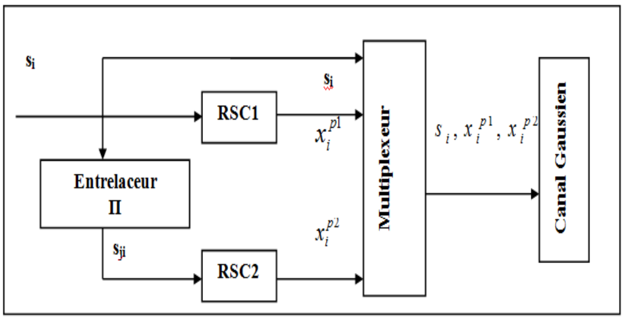

Fig 1: Principle of a parallel turbo code.

The overall rate Rc of the constructed code is directly related to rates R1(of RSC1 code) and R2 (of RSC2 code) by the following relation:

$$R_c = \frac{1}{R_1} + \frac{1}{R_2} - 1 = \frac{R_1 R_2}{R_1 + R_2 - R_1 R_2}$$

If R1 and R2 are identical, the relationship becomes:

$$R_c = \frac{R_1}{2 - R_1}$$

2.2 Recursive systematic convolutional codes
A convolutional code is said recursive when its generating polynomials are replaced by the quotient of two polynomials. Part of the output is then fed back into the shift register as defined by the polynomials connections located at the denominators. A convolutional code is said systematic when a portion of its output is exactly equal to its inputs.

The structure of the RSC code is as follows:

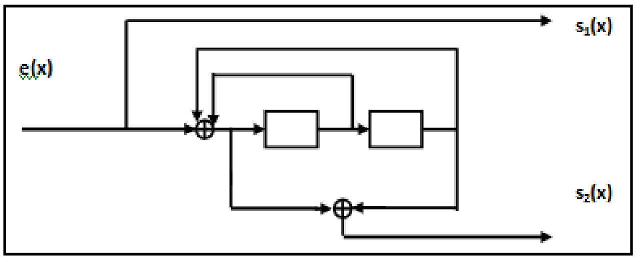

Fig 2: RSC convolutional code (7.5)

2.3 Interleavers

The interleaver performs a permutation of the order of the input bits. In practice, this permutation is chosen randomly to disrupt the good order of the symbols before being fed to the second recursive encoder. The objective is to produce code words that are separated in space signals and resembling something which could give a random coding [2].

Several permutation methods are possible; however, the choice of the structure of the interleaver is a key factor which determines the performance of a turbo code [3].

3. TURBO DECODERS

A maximum likelihood detection can be considered for decoding turbo codes. However, this optimal decoding process requires considerable complexity that makes it unusable. Fortunately, an iterative decoding process of lower complexity can be applied while providing quasi-optimal performances. In general, this process involves a mutual exchange of information between the two so-called extrinsic decoders. The decoding module associated to each code is constituted of a flexible input-output MAP algorithm. There are several variants of the MAP algorithm seeking to reduce its complexity. The best known are the Log MAP algorithms and Max Log MAP [5].

3.1 Decoding of turbo codes

Turbo codes are obviously linear codes. Therefore, to decode them we can use the technique of lattice codes and the Viterbi algorithm. However, a negative impact of the interleaving operation is that it actually leads to the lattice code whose depth can be very large, and thus invalidates the decoding by the Viterbi algorithm. Conversely, the partial decoding which uses only the systematic part and the parity control suites alone (For Exp: s_i, x_i^{p1}) can be done efficiently (assuming that the decoder adjust the permutation rule) provided that the memory of the recursive encoder is not too large. The trick of iterative decoding of turbo codes consists of decoding the received message through a succession of simple decodingo, which exchange information at each iteration.

Fig. 3 shows the scheme principle of the iterative decoder. We can see that the system is composed of two decoders that work iteratively exchanging information L_{ij}^{ext} . Each of these decoders calculates a posterior the probability distribution for each source bit from the three following information:

The noisy version y^s corresponding to the systematic part of the encoder.

The noisy version y^{pi} corresponding to one of the two parties controlling the encoder's parity.

The a priori information about the probabilities of the source bits (symbolised by L_{ij}^{ext} in Fig. 3). At the first iteration, this information will correspond to the a priori source bit information for decoder number 1, and then this information is refreshed according to the result of decoding of the second decoder.

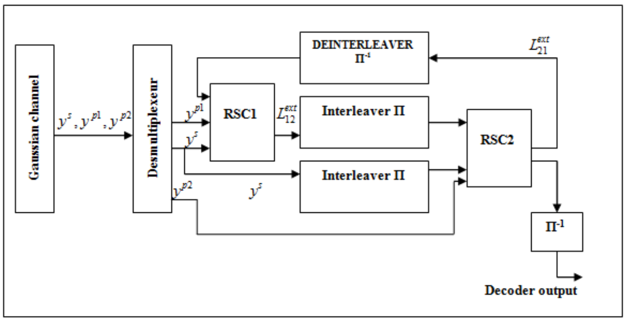

Fig 3: Principle of an iterative decoder for turbo code.

4. PERFORMANCE OF TURBO CODES

4.1 Introduction

To study the performance of Turbo codes, we insisted on doing the convolutional decoding using the MAP algorithm. We will try to show/ or highlight the performance of turbo codes taking into account the variation of different parameters involved in the construction of a Turbo decoder, namely: the number of iterations, the performance, the channel, the generating matrix and the interleaver at play in the Turbo Decoder.

4.2 Results of simulations and analysis

4.2.1 Influence of the variation of the generator code

Result and Comment

In Fig 4 and Fig 5. We observe that for two different generating codes, with different variation in the size of the message to transmit, that the BER which represents the binary error rate as a function of signal to noise ratio (SNR) gradually diverge for different lengths of stresses when the SNR increases, especially for medium-sized message.

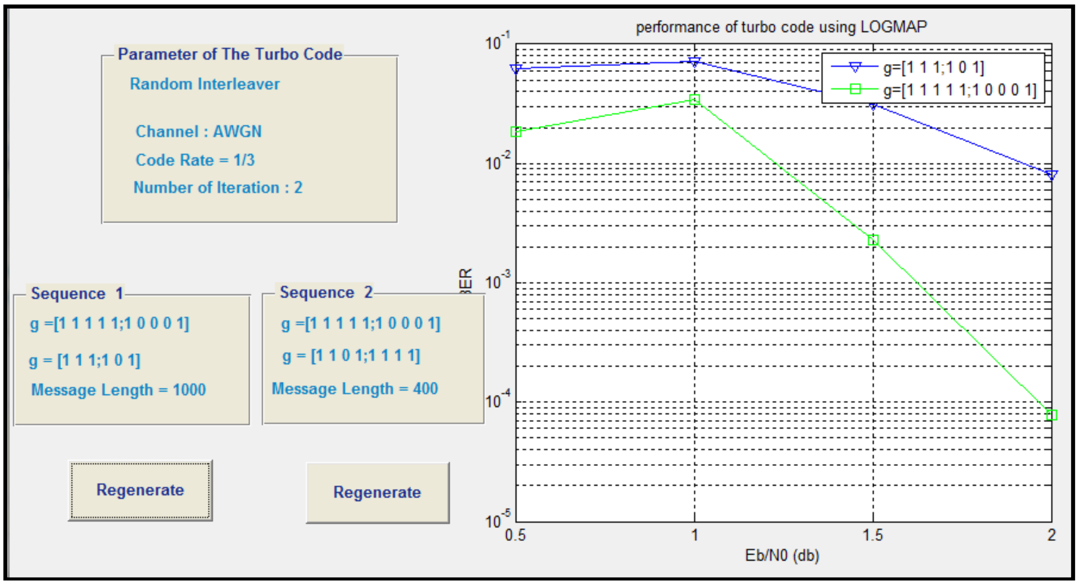

Fig 4: Effect of variation of generator with a long message

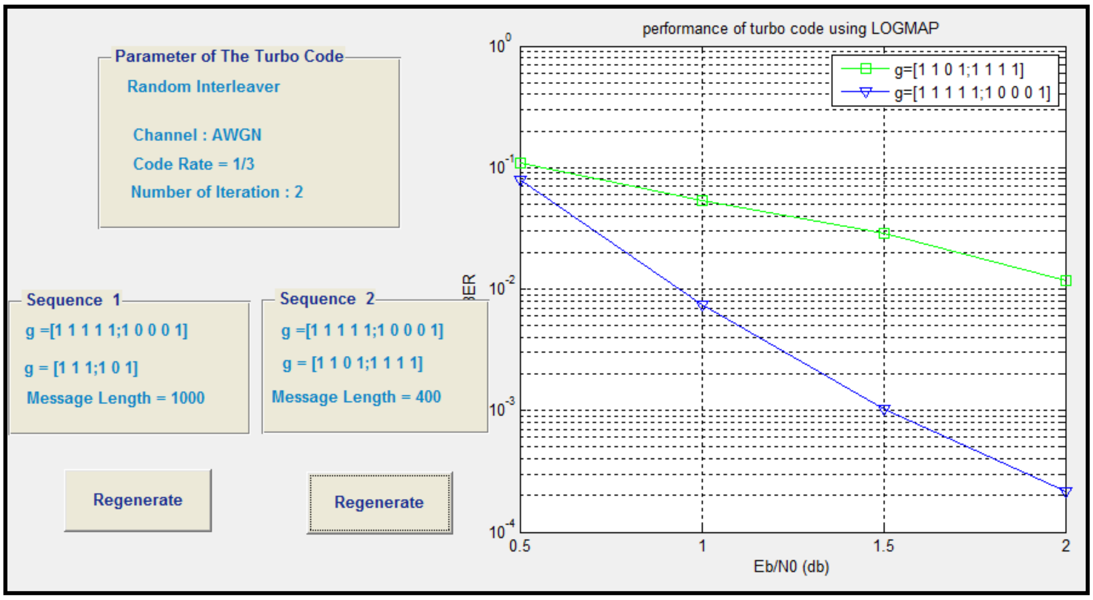

Fig 5: Effect of variation of a message generator with less long.

4.2.2 Influence of Code Rate

Results and comment

The code is more performant when the Rate is lower and the stress length is higher. However; the Rate is lower when the bandwidth required for transmission is important.

According to Fig. 6 We see that for the same sequence of bits the BER of a turbo code of lower Rate (in our case 1/3) is weaker than a turbo code whose Rate is 1/2

4.2.3 Performance by change of interleaver

For this simulation, a permutation of the symbols is used at the emission by using two types of interleavers.

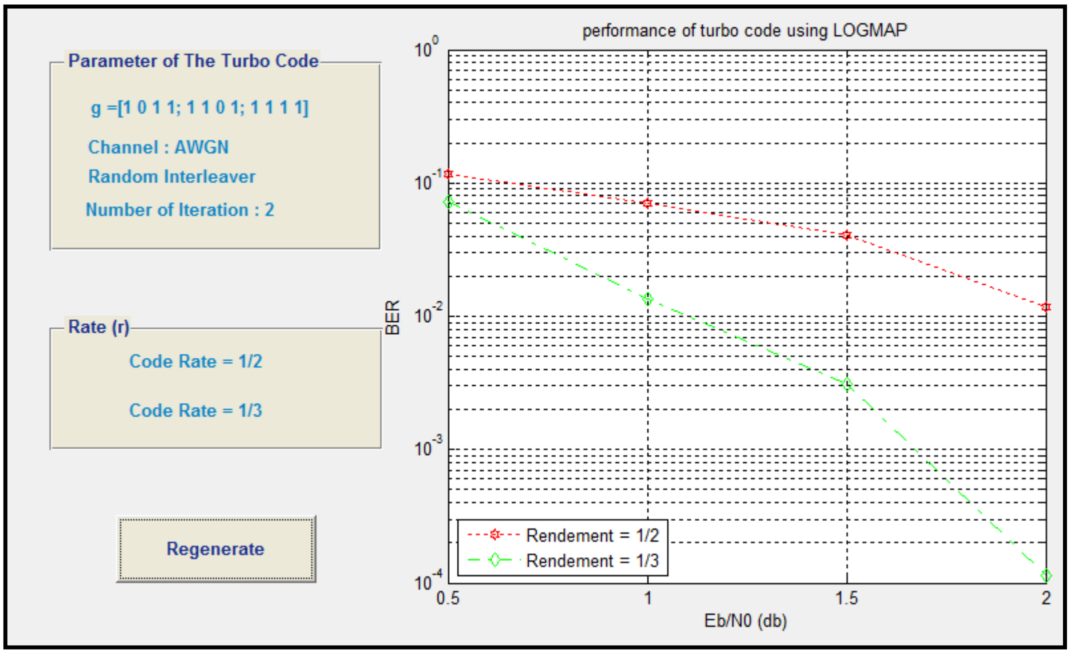

Fig. 6.Effect of variation in Rate code (R)

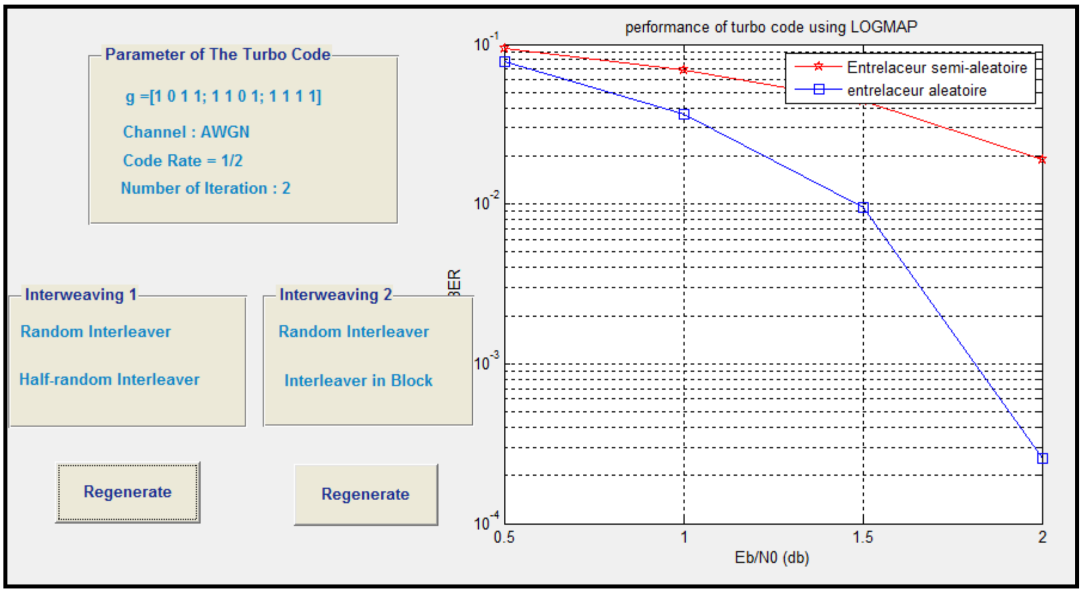

Fig. 7.Performance of turbo code with interleaver (random & semi-random)

Results and comment

According to the above figures (Fig. 6 and Fig. 7), we observe that the random interleaver gives a BER smaller than the two other interleavers (in block and semi-random). Indeed, the random interleaver allows switching a sequence of bits in the way that two originally close symbols are as far as possible distant from each other, permitting particularly to transform an error related to grouped bits into an error distributed over the entire sequence.

4.2.4 Performance by number of iteration

The principle of the iterative decoder is the following (Fig 3):

The two decoders operate iteratively.

At each step, a set-top boxes (say i) receives as input a priori information on the source symbols in the form of partial likelihood ratios computed by the decoder L_{ij}^{k} above (or from the source model, if no further information available from this type). Then it proceeds as follows:

It decorates the edges of the lattice using the γ_k using the three terms related to (i) the a priori information, (ii) the channel outputs on systematic bits, (iii) the output bits on parity check. (As the latter two do not change, it suffices in fact to take into account the change a priori information).

It calculates the α_k and β_k from γ_k and then the terms $L_{ij}^{e}(u_k)$ source for each bit. It provides this term input of the decoder below.

Results and comment

The figure (Fig 8) show that if we increase the number of iterations the bit error rate (BER) has become smaller it is to say that the performance of Turbo code is proportional with the number of iteration.

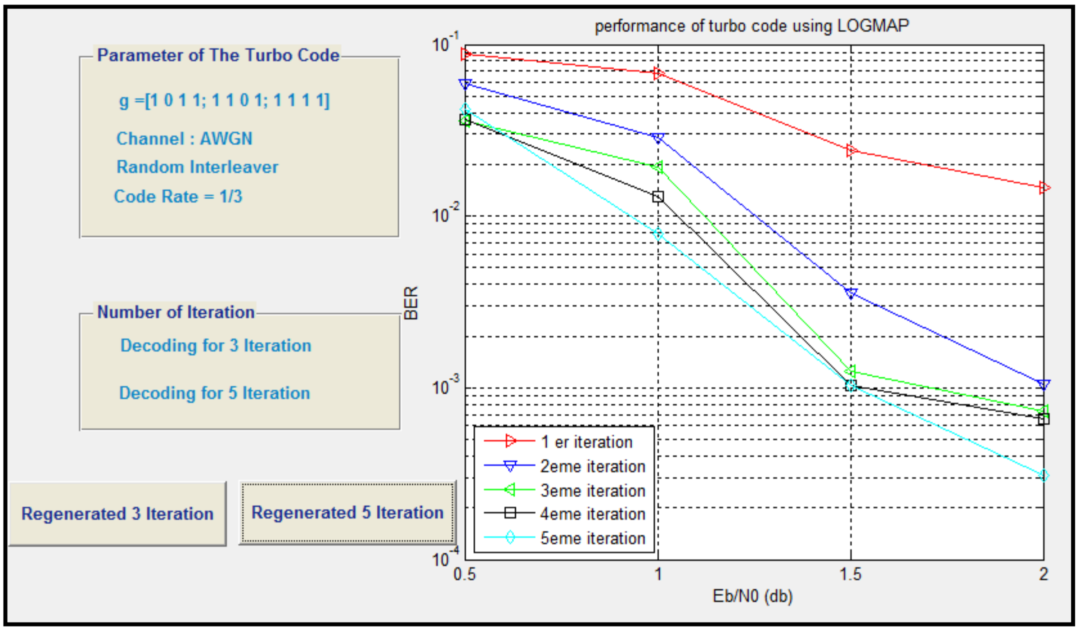

Fig 8: Performance of turbo code with 5 iterations.

5. CONCLUSIONS

The performance of error correcting codes and more precisely the family of Turbo Codes depends on some parameters on which we can say:

The performance of the code has a significant influence on the speed of convergence: a low rate code with enough iterations leads to the convergence of the binary error rate. The stress length of the generating code and the size of the sequence are very important parameters in the performance of Turbo Codes. The number of iterations is the most important parameter for a lower BER. If we increase this number we can have a BER in the order of 10^{-5}

And finally to improve the performance of Turbo codes we have to choose an appropriate interleaver in order to have very good results.

6. REFERENCES

[1] Berrou, C, Glavieux, A. and Thitimajshima, P. (1993). "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes, " Proceeding of IEEE ICC , pp. 1064-1070.

[2] Muhammad Arif, Noor Muhammad Sheikh, Asrar U.H. Sheikh. (September 2008). "Design of two step deterministic interleaver for turbo codes" Computers & Electrical Engineering, Volume 34, Issue 5, Pages 368-377

[3] Ji LUO, Xi-lin ZHANG, Dong-feng YUAN. (May 2005). "Turbo Codes with Modified Code-matched Interleaver over AWGN and Rayleigh Fading Channel "Chinese Journal of Aeronautics, Volume 18, Issue 2, Pages 147-152.

[4] Johann Barbier. (2005) "Reconstruction of turbo-code encoders ", Proc. of SPIE Vol. 5819 463-473

[5] Mustafa Taskaldiran, Richard C.S. Morling, and Izzet Kale. (2009). "The Modified Max-Log-MAP Turbo Decoding Algorithm by Extrinsic Information Scaling for Wireless Applications" Springer. ISBN 9780387717869

[6] Lina Fagoonee , Bahram Honary.(2004). "Application of turbo codes to tactical communications", Computer Networks 46 741–749