HAL open science

Characteristics of a sixteen-state turbo-encoder/decoder (Turbo4)

Michel Jezequel, Claude Berrou, Catherine Douillard, Pierre Penard

To cite this version:

Michel Jezequel, Claude Berrou, Catherine Douillard, Pierre Penard. Characteristics of a sixteen-state turbo-encoder/decoder (Turbo4). International Symposium on Turbo Codes & Related Topics, Télécom Bretagne, Sep 1997, Brest, France. pp.280-283. hal-01767925

HAL Id: hal-01767925

https://hal.science/hal-01767925

Submitted on 16 Apr 2018

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.

CHARACTERISTICS OF A SIXTEEN-STATE TURBO-ENCODER/DECODER *(TURBO4)*

Michel Jézéquel*, Claude Berrou*, Catherine Douillard* and Pierre Pénard**.

*ENST de Bretagne, Technopôle Brest Iroise, BP 832, 29285 BREST Cedex FRANCE.
Phone: +33 (0)2 98 00 13 06, Fax: 33 (0)2 98 00 13 43

**CCETT, 4 rue du Clos Courtel, BP 59, 35512 CESSON SEVIGNE Cedex FRANCE.
Phone: +33 (0)2 99 12 49 05, Fax: +33 (0)2 99 12 40 98

Email: Michel.Jezequel@enst-bretagne.fr/ Claude.Berrou@enst-bretagne.fr/
Catherine.Douillard@enst-bretagne.fr/ pierre.penard@cnet.francetelecom.fr

ABSTRACT

This paper presents the characteristics of an integrated circuit called "turbo4" which can be used as a turbo-encoder or as a turbo-decoder. The turbo-encoder is built using a parallel concatenation of two recursive systematic convolutional codes with constraint length K=5. The turbo-decoder is cascadable, each circuit processing one iteration of the turbo-decoding algorithm. It is designed around 2 sixteen-modified Viterbi decoders and 2 matrices of 64 x 32 bits for interleaving and deinterleaving. Some measures for Gaussian and Rayleigh channels and for different coding rates are presented.

1 INTRODUCTION

Turbo codes are a new family of error correcting codes introduced by C. Berrou and *al*. [1,2]. They implement a parallel concatenation of recursive and systematic convolutional codes, possibly punctured. The decoding process is iterative. Therefore, the turbo-decoder can be implemented in a modular pipelined structure, in which each module is associated with one iteration. Then, performance in Bit Error Rate (BER) terms is a function of the number of chained modules. Turbo codes show results which are very close to the theoretical channel limit.

Turbo codes have been implemented in two different integrated circuits. The first one called "CAS5093" and distributed by COMATLAS is built around 5 eight-state modified Viterbi decoders [3] and 4 matrices of 32 x 32 bits for interleaving and deinterleaving. This circuit contains 2.5 modules. The second one called *turbo4* includes one module and is cascadable, so the user can create a decoder consisting of several modules.

This paper presents the characteristics of *turbo4*. It is organised as follows : the next section gives the main technical characteristics of *turbo4*. Section 3 is dedicated to the architecture of the circuit. Finally, we conclude by presenting some results of simulations and tests.

2 MAIN CHARACTERISTICS OF *TURBO4*

Turbo4 can be used as an encoder or as a decoder. It is designed with a 2-metal, 0.8µm CMOS technology. The chip with a size of 78 mm^2 contains 0.6 M transistors. Features of the circuit are shown below:

- Turbo code with constraint length K=5 (polynomials 23,35)
- Non uniform interleaving (size 64 x 32 bits)
- Cascadable decoder
- Soft output
- Coding gain for a Gaussian channel with 4 iterations for the decoding process (4 circuits for the decoder):
    - 9 dB @ BER 10^{-7}, R=1/2
    - 8 dB @ BER 10^{-7}, R=2/3
- Encoding latency: 4
- Decoding latency: 2178 per iteration
- Intrinsic self-synchronisation (R=1/2), help for synchronisation (R≠1/2) or external synchronisation
- Output giving an estimation of the channel quality

3 ARCHITECTURE

Turbo4 is built around 4 blocks: the encoder, the decoder, the interleaver/deinteleaver and the synchronisation/supervision block.

3.1 TURBO-ENCODER

The turbo-encoder is built using a parallel concatenation of two recursive systematic convolutional codes (figure 1). The incoming data (XI) is fed into a first encoder that produces

redundancy Y1 while the second encoder receives interleaved data and produces redundancy Y2.

The required coding rate is obtained by puncturing Y1, Y2 and, possibly XI. For a 1/2 coding rate, the puncturing function is included in the circuit. In this case, composite redundancy is output following the sequence: Y2 Y1 Y1 Y1. For other coding rates, the puncturing function has to be designed on the board.

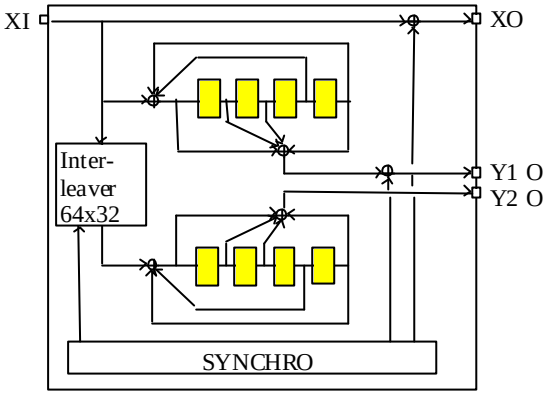

figure 1 : Turbo-encoder

A synchronisation block is added in order to make synchronisation between the encoder and the decoder possible.

3.2 TURBO-DECODER

The decoder processes one turbo-decoding iteration. It is designed (figure 2) around 2 sixteen-state SOVAs (Soft Output Viterbi Algorithm, an acronym proposed by J. Hagenauer [4]) and 2 matrices of 64x32 bits for interleaving and deinterleaving. Some delay lines are added in order to compensate for latency of other blocks like SOVAs.

The decoder is cascadable, so the user can choose the number of iterations, each circuit corresponding to one iteration. Its programmability makes it possible to adapt some coefficients to the number of the iteration. Programmability is ensured by some control inputs like: MY2[1:0], E7, MXZ, ...

The decoder receives noisy symbols X (data), Y1 (redundancy produced by the first encoder) and Y2 (redundancy produced by the second encoder) from the channel. It receives Z (extrinsic information) computed by the previous circuit. These incoming data are coded on 4 bits in 2's complement.

SOVA1 works on redundancy Y1 with noisy data X + Z. SOVA2 processes Y2 and the interleaved output of SOVA1 from which incoming data Z has been subtracted.

The X input of SOVA2 is subtracted from its output and after deinterleaving, ZO may be used by the subsequent module as input Z.

3.3 INTERLEAVING / DEINTERLEAVING

Interleaving and deinterleaving are convolutional, each function using one memory (a matrix of 64 rows by 32 columns). Data are written row by row, then read following specific rules which ensure non-uniformity of the interleaving process. Global latency due to one interleaver and deinterleaver is 2048.

3.3 SYNCHRONISATION / SUPERVISION

Synchronisation of the circuit can be divided into two phases: the synchronisation search phase and the tracking phase, also called supervising phase.

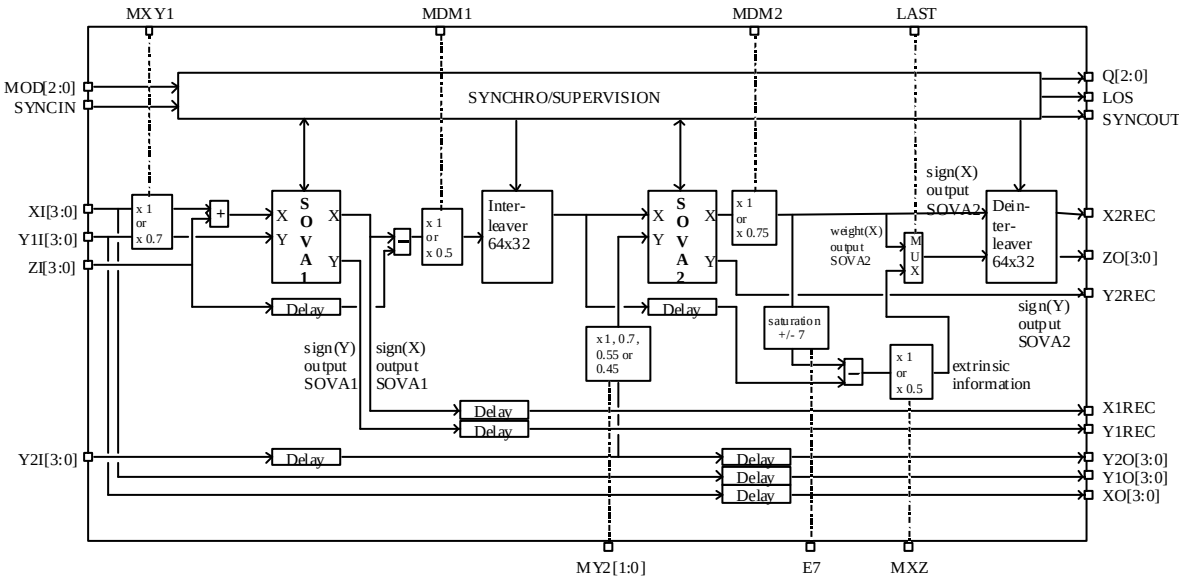

figure 2 : Turbo-decoder

No synchronisation word is required for the synchronisation search phase, thus there is no loss in the coding rate. This phase uses a controlled inversion of symbols coming from the turbo-encoder.

The supervising phase uses the pseudo-syndrome procedure [5]. This method is decorrelated from the one implemented for the synchronisation search phase. It deals with out-of-synchronisation detection and forces the circuit to return automatically to the synchronisation search phase. It also gives an estimation of the channel quality.

Tests on *turbo4* showed an error in the design of the synchronisation block. A new version of the circuit is in process in order to overcome this problem.

4 PERFORMANCE

This section presents results of simulations and tests [6].

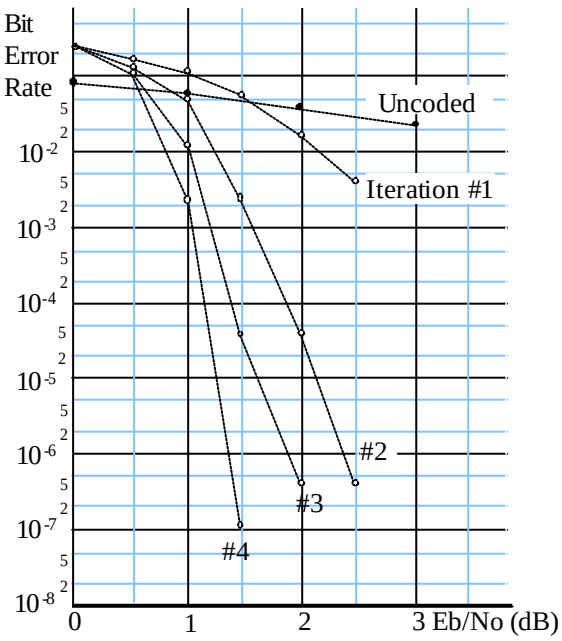

Figure 3 : Gaussian channel, R=1/3

Figure 3 and 4 present simulation results for a Gaussian channel. Figure 3 shows results for 1 to 4 modules with a global coding rate R=1/3, each code working with a coding rate equal to a half. In figure 4 simulations are made with 3 modules (3 iterations) for BPSK or QPSK modulations. These curves are given for different coding rates. If R is the global coding rate, R_1 the coding rate associated with the first code and R_2 the coding rate associated with the second code we have :

- R= 1/2 : R_1 = 4/7, R_2 = 4/5;
- R= 2/3 : R_1 = 4/5, R_2 = 4/5;
- R= 3/4 : R_1 = 6/7, R_2 = 6/7;
- R= 4/5 : R_1 = 8/9, R_2 = 8/9;

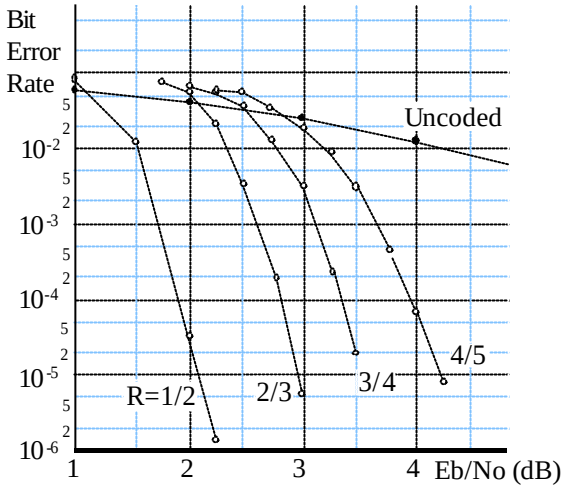

Figure 4 : Gaussian channel (3 iterations)

Figure 5 shows measured results for a Gaussian channel. Measures are made with 1 to 5 modules for a global coding rate R= 1/2 (R_1= 4/7, R_2= 4/5)

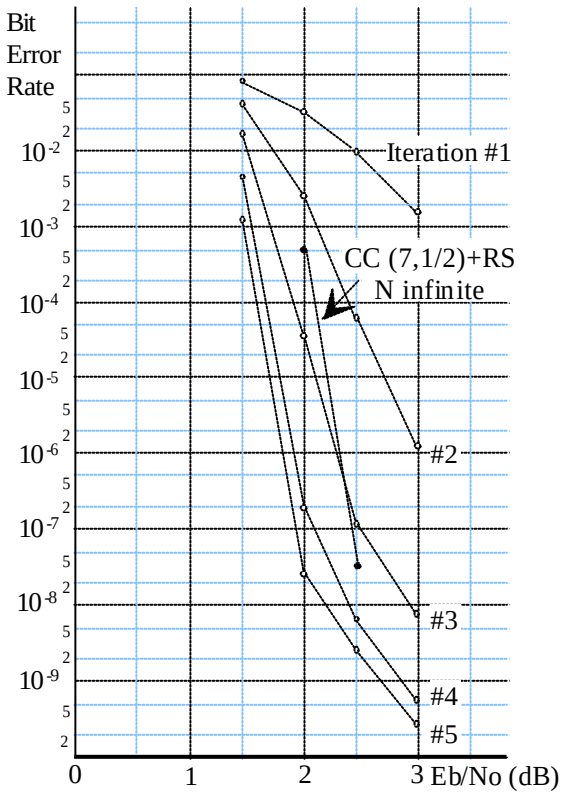

Figure 5 : Gaussian channel, R=1/2

In addition, figure 5 presents a comparison between *turbo4* performance and a classical concatenation of a convolutional code K=7 and a 8-bit (255,223) Reed-Solomon code with an infinite interleaver. This comparison shows that the coding gain given by 4 or 5 modules is better than that given by the classical concatenation for a bit error rate higher than 10^{-8}. This comparison is

made without taking into account the interleaver size which would decrease classical concatenation performance. Moreover the coding rate is more efficient for *turbo4* (R=1/2) than for the classical concatenation (R=0.437).

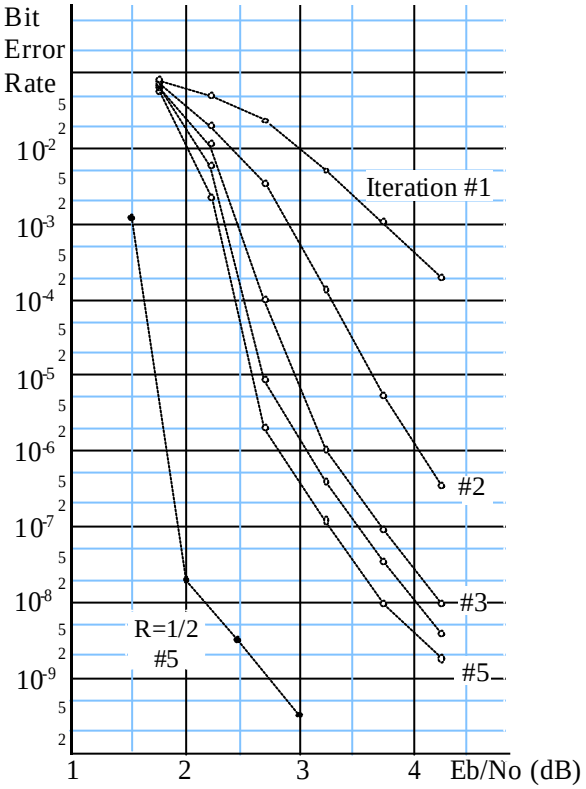

Figure 6 : Gaussian channel, R=2/3

Figure 6 presents measured results for a Gaussian channel. Measures are made with 1 to 5 modules for a global coding rate R=2/3 (R_1= 4/5, R_2= 4/5).

Figure 7 presents measured results for a Rayleigh channel with optimum channel interleaving and weighting. Measures are made with 1 to 5 modules for a global coding rate R = 1/2 (R_1 = 4/7, R_2 = 4/5). Results are most satisfactory, as the slope is the same as with the Gaussian channel with a gap of 2.5 dB.

Figure 5, 6 and 7 show a flattening degradation for low bit error rates. This degradation is not due to turbo codes but to the internal accuracy of *turbo4*, which works with 4 bits [6].

5 CONCLUSION

The coding gain of turbo codes was verified by tests on *turbo4*. Except for the Big Viterbi Decoder (using the constraint length 15), *turbo4* is at the moment the circuit with the best coding gain, on both Gaussian and Rayleigh channels.

The error detected in the design of the synchronisation block will be overcome in a new version of the circuit.

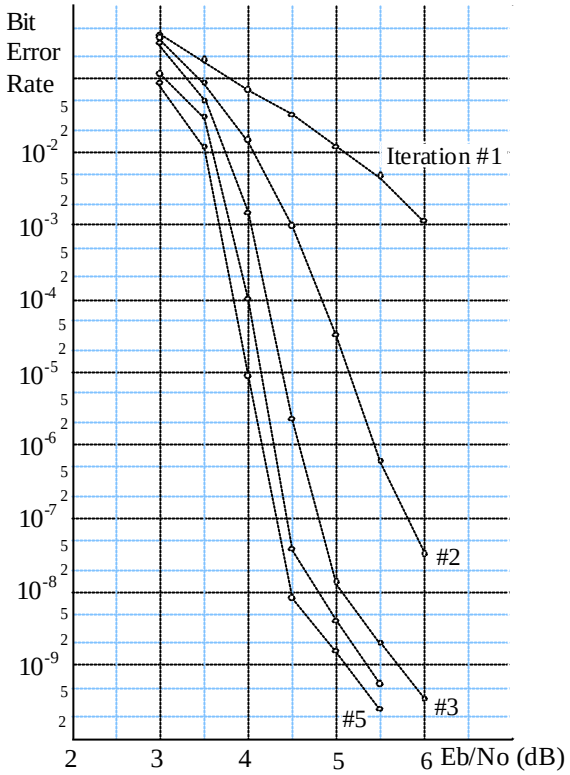

Figure 7 : Rayleigh channel, R=1/2

Acknowledgements.

The authors would like to thank P. Ferry and J.R. Inisan for their help.

REFERENCES

[1] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes", Proc. of IEEE ICC '93, Geneva, pp. 1064-1070, May 1993.

[2] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding : turbo-codes", IEEE Transactions on communications, Vol. 44, N°10, pp. 1261-1271.

[3] C. Berrou, P. Adde, E. Angui and S. Faudeil, "A low complexity soft-output Viterbi decoder architecture", Proc. of IEEE ICC '93, Geneva, pp. 737-740, May 1993.

[4] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications", Proc. IEEE Globecom'89,Dallas, Texas, Nov. 1989, p.47.1.1-47.1.7.

[5] C. Berrou and C. Douillard, "Pseudo-syndrome method for supervising Viterbi decoders at any coding rate", Electron. Lett., Vol 30, n°13, pp 1036-1037, June 1994.

[6] M. Jézéquel, C. Berrou, J. R. Inisan and Y. Sichez, "Test of a Turbo-Encoder/Decoder", TURBO CODING Seminar, Lund, Sweden, pp. 35-41, 28-29 August 1996.

[7] Onyszchuk, I. M., "Coding Gains and Error Rates From the Big Viterbi Decoder," TDA PR 42-106: April-June 1991, pp.170-174, August 15, 1991.