A comparison of turbo codes using different trellis terminations

Hokfelt, Johan; Edfors, Ove; Leanderson, Carl Fredrik

Published in:
Annales des Télécommunications

2001

A comparison of turbo codes using different trellis terminations [Comparaison de turbo codes pour différentes terminaisons du treillis]

Johan Hokfelt[1], Carl Fredrik Leanderson[2], Ove Edfors[2]

[1] Fiberless Society, PO Box 714, SE-220 07 LUND, Sweden
Phone: +46 (0)46 38 65 94, Fax: +46 (0)46 38 65 99
E-mail: Johan.Hokfelt@fiberless-society.com

[2] Dept. of Electroscience, Lund University, PO Box 118,
SE-221 00 LUND, Sweden
Phone: +46 (0)46 222 30 21, Fax: +46 (0)46 222 47 18
E-mail: {Ove.Edfors, Carl_Fredrik.Leanderson}@es.lth.se

Abstract

This paper investigates the incidence of trellis termination on the performance of turbo codes and accounts for the performance degradation often experienced in the absence of trellis termination. Analytical upper bounds on the performance for the ensemble of turbo codes using different trellis termination strategies as well as performance results obtained by computer simulation are presented. In the case of uniform interleaving, the performance differences between various termination methods are relatively small, except when using no trellis termination at all.

Keywords - Turbo codes, trellis termination, distance spectra.

[On étudie l'influence de la terminaison du treillis sur les performances de turbo codes et on rend compte de la dégradation souvent observée en l'absence de terminaison. On présente des bornes supérieures analytiques de la probabilité d'erreur sur l'ensemble des turbo codes pour différentes manières de terminer le treillis, ainsi que l'évaluation de ces performances par simulation sur ordinateur. Dans le cas où l'entrelacement est uniforme, les performances varient assez peu d'un type de terminaison à un autre, excepté quand aucun des treillis n'est terminé.

Mots-clés - Turbo codes, terminaison de treillis, distribution de distances.]

I. INTRODUCTION

Turbo codes are in general implemented as two recursive convolutional encoders in parallel, where the input to the second encoder is an interleaved version of the original information sequence fed to the first encoder [1]. At the beginning of each information block, the encoders are initialized to their zero-states. Similarly, at the end of each information block it is desirable to force the encoders back to the zero-state, an operation known as trellis termination. For feedforward convolutional encoders, this is readily achieved by appending tail bits at the end of the encoder input sequence. However, the recursive property of the component encoders used in turbo codes implies a state-dependency on these tail bits and, hence, individual tail sequences are required for each component encoder.

The performance of a specific trellis termination method is dependent on the particular interleaver used in the turbo encoder. This dependency is a result of *interleaver edge effects* [2]. These edge effects are described in this paper for the case of uniform interleaving [4]. It is demonstrated how the choice of different termination methods influences the performance for turbo codes with different interleaver lengths and different number of memory elements in the component encoders. The distance spectra are calculated using the concept of uniform interleaving and the tangential sphere bound is used to upper bound the achievable maximum likelihood (ML) decoding performance.

The investigated trellis termination strategies are: no termination at all, termination of the first encoder only, termination of both encoders within the length of the interleaver and termination of both encoders with post-interleaver flushing [3].

II. DISTANCE SPECTRA

The calculation of the distance spectrum of a specific turbo code involves taking the particular interleaver into account, a task that becomes prohibitively complex even for short-length interleavers. A less computationally demanding method was introduced by Benedetto *et al.* in [4], where a method to derive the average distance spectrum for the ensemble of all interleavers of a certain length was presented. In this section we summarize their method, and present an extension by which we include the influences of different trellis terminations. The methodology as such is general, but for reasons of simplicity we restrict this presentation to two-component turbo codes with binary systematic recursive convolutional encoders, as introduced in [1].

Benedetto *et al.* introduced the *input-redundancy weight enumerating function* (IRWEF) [4]

$$A(W,Z) \triangleq \sum_{w=0}^{N} \sum_{j=0}^{J} A_{w,j} W^w Z^j \tag{1}$$

for a systematic $(N+J,N)$-code, where $A_{w,j}$ is the number of codewords with input weight w and parity weight j. N is the number of information bits (corresponding to the interleaver length), J is the number of generated parity bits, and W and Z are dummy variables.

Since both component encoders in a turbo code share the same input bits, though in different order, every codeword that belongs to a turbo code is composed of two component-code codewords that both result from sequences of the same weight w. For this reason, Benedetto *et al.* defined the *conditional weight enumerating function* (CWEF)

$$A_w(Z) \triangleq \sum_{j=0}^{J} A_{w,j} Z^j, \qquad w = 0, 1, \ldots, N, \tag{2}$$

which enumerates the number of codewords of various parity weights j, conditioned on the input weight w. The CWEF of the first and second component encoders are denoted $A_w^{C_1}(Z)$ and $A_w^{C_2}(Z)$ respectively, and the CWEF of the overall turbo code $A_w^{TC}(Z)$. By introducing a probabilistic interleaver construction called a *uniform interleaver*, for which all distinct mappings are equally probable, Benedetto *et al.* obtained the CWEF of the ensemble of all turbo codes using interleavers of length N as

$$A_w^{TC}(Z) = \frac{A_w^{C_1}(Z)\, A_w^{C_2}(Z)}{\binom{N}{w}}, \tag{3}$$

where $1 \big/ \binom{N}{w}$ is the probability that a specific weight-w sequence is mapped to another, specific, weight-w sequence. Finally, the number a_d of words of Hamming weight d is equal to

$$a_d = \sum_{w=1}^{N} A_{w,d-w}^{TC}, \tag{4}$$

where $A_{w,d-w}^{TC}$ are the coefficients in the turbo code CWEF, *i.e.* $A_w^{TC}(Z) \triangleq \sum_{j=0}^{J} A_{w,j}^{TC} Z^j$. Since we are addressing systematic codes the codeword weight is the sum of input and parity weight, *i.e.* $d = w + j$.

When deriving the CWEF of the component codes of turbo codes, it is common practice to take only the error events that end up in the zero-state into account, *i.e.* to consider only *zero-terminating* input sequences. Depending on the method of trellis termination, codewords might also exist that result from trellis paths that do not end up in the zero-state after N trellis transitions. In the sequel, a method to derive the CWEF for various trellis termination methods is presented.

A. Interleaver edge effects

Interleaver edge effects refer to the implications on the distance spectrum resulting from the block partitioning of the input sequence, as the result of a limited-length interleaver [2]. Due to this truncation, low-weight parity words can be generated even though the encoder input sequences do not force the encoders back to the zero-states. In terms of weight enumerating functions, this means that we require knowledge not only of the number of trellis paths that lead to the zero-state after the last transition, but also the number of paths that lead to other final states. This can be obtained by partitioning the IRWEF defined by (1) into a *state-dependent* counterpart $A_{t,s}(W,Z)$, which enumerates the number of trellis paths that lead to state s, having input weight w and parity weight j. An efficient method to find the state-dependent IRWEF of a convolutional encoder valid after t trellis transitions is to extend the IRWEF of the same encoder obtained for $t-1$ transitions. The state and time dependent IRWEF is defined as

$$A_{t,s}(W,Z) \triangleq \sum_{w=0}^{N}\sum_{j=0}^{J} A_{t,s,w,j} W^w Z^j \tag{5}$$

where $A_{t,s,w,j}$ is the number of paths with input weight w and parity weight j that lead to state s after t trellis transitions. Based on the encoder trellis, the coefficients of the state and time dependent IRWEF are calculated recursively in time as

$$A_{t,s,w,j} = \sum_{u=0}^{1} A_{t-1,S(s,u),w-u,j-P(S(s,u),u)}, \tag{6}$$

where $S(s,u)$ is the state that leads to state s when the input symbol is u, and $P(S(s,u),u)$ is the parity weight generated by the corresponding trellis

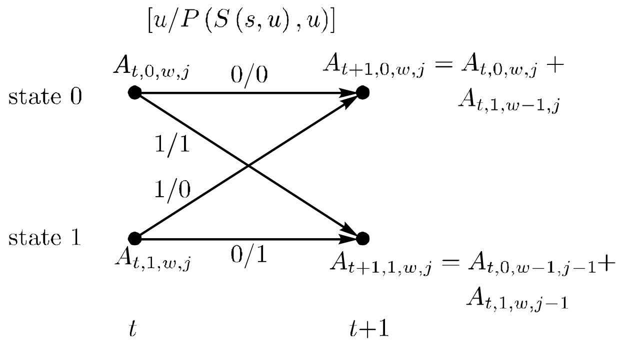

Fig. 1. Recursive calculation of distance spectrum for a two-state encoder. (French: Calcul récursif de la distribution de distances d'un codeur à deux états.)

transition. This recursion is illustrated in Figure 1, for a two-state encoder. At time $t = 0$, the recursive procedure is initialized with $A_{0,0,0,0} = 1$ and $A_{0,s,w,j} = 0$, $(s, w, j) \neq (0, 0, 0)$, which corresponds to an encoder initialized in the zero state.

Let $E_{w,j}^{C_1}$ and $E_{w,j}^{C_2}$ denote the multiplicities of codewords with input weight w and parity weight j that correspond to trellis paths that do not end up in the zero-state after encoding length-N input blocks, for component code C_1 and C_2 respectively. We denote such codewords *edge-effect codewords* and their multiplicities are the coefficients of the corresponding CWEFs, according to $E_w^{C_l}(Z) = \sum_{j=0}^{J} E_{w,j}^{C_l} Z^j$, $l = 1, 2$. The overall CWEFs, including both zero-terminating and edge-effect codewords, are then obtained as

$$\tilde{A}_w^{C_l}(Z) = A_w^{C_l}(Z) + E_w^{C_l}(Z), \tag{7}$$

and the resulting CWEF for the turbo code is

$$\tilde{A}_w^{TC}(Z) = \frac{\tilde{A}_w^{C_1}(Z)\,\tilde{A}_w^{C_2}(Z)}{\binom{N}{w}}. \tag{8}$$

Note that $A_w^{C_l}(Z)$ includes only trellis paths that end in the zero-state after

N transitions. Thus, $A^{C_l}_{w,j}$ is obtained from (6) for $t = N$ and $s = 0$. The difference between $\tilde{A}^{C_l}_{w}(Z)$ and $A^{C_l}_{w}(Z)$ equals $E^{C_l}_{w,j}$, which depends on how the trellises are terminated. In order to evaluate this quantity, we calculate below $E^{C_l}_{w,j}$ for four classes of trellis termination methods:

1. No termination of either component encoder.
2. Termination of the first component encoder.
3. Termination of both component encoders.
4. Post-interleaver flushing.

Class I. No trellis termination

With no termination of either component encoder, the multiplicities of codewords that stem from interleaver edge effects are calculated by summing the number of paths that end in the non-zero states after N trellis transitions. Thus,

$$E^{C_1}_{w,j} = \sum_{s=1}^{2^{m_1}-1} A^{C_1}_{N,s,w,j}, \quad (9)$$

$$E^{C_2}_{w,j} = \sum_{s=1}^{2^{m_2}-1} A^{C_2}_{N,s,w,j}, \quad (10)$$

where m_1 and m_2 are the number of memory elements in encoder 1 and 2, respectively. The overall distance spectrum including edge effect codewords, $\tilde{A}^{TC}_{w}(Z)$, is calculated using (7) and (8).

Class II. Termination of the first encoder

By appending m_1 tail bits to the input sequence so that the first encoder is terminated in the zero-state, the edge effect codewords are entirely removed from the first component code. Note that the tail bits are included in the sequence that enters the interleaver, and that their Hamming weight is included in the input weight w. For the second encoder, the situation is identical to the case of no trellis termination. Hence,

$$E_{w,j}^{C_1} = 0, \tag{11}$$

$$E_{w,j}^{C_2} = \sum_{s=1}^{2^{m_2}-1} A_{N,s,w,j}^{C_2}. \tag{12}$$

Class III. Termination of both encoders

It is also possible to terminate both component encoders in their zero-states. At least two different ways of achieving this have been reported in the literature:

1. By imposing interleaver restrictions, the second encoder can be forced to end up in the same state as the first encoder [5, 6]. It is then sufficient to append a single set of tail bits according to termination Class II in order to terminate both encoders in their zero-states.

2. By identifying specific, interleaver dependent, input positions it is possible to force the component encoders to their zero-states independently of each other [7]. This is achieved without any restrictions on the choice of interleaver, but with a slight increase in the number of input bits dedicated to trellis termination (m termination bits are required, $\max(m_1, m_2) \leq m \leq m_1 + m_2$).

With both encoders terminated in their zero-states, all edge-effect codewords are removed. Consequently, $E_{w,j}^{C_1} = E_{w,j}^{C_2} = 0$.

Class IV. Post-interleaver flushing

Trellis termination by post-interleaver flushing was proposed in [3]. With this method, both encoders are flushed independently of each other, after encoding their N-bit input sequences. The combination of the weight spectra of the component encoders is then similar to the case of no trellis termination, since the trellises are not terminated by the end of their length-N input sequences. However, extra codeword weight is added as a consequence of the encoder flushing. This is accounted for by adding the weight of the flush bits and the corresponding parity bits to the parity weight in the IRWEFs. More precisely,

$$E_{w,j}^{C_1} = \sum_{s=1}^{2^{m_1}-1} A_{N,s,w,j-F_1(s)}^{C_1}, \tag{13}$$

$$E_{w,j}^{C_2} = \sum_{s=1}^{2^{m_2}-1} A_{N,s,w,j-F_2(s)}^{C_2}, \tag{14}$$

where $F_l(s)$, $l = 1, 2$, is the sum of the weight of the flush bits and parity bits generated when forcing encoder l to the zero-state from state s.

III. EVALUATION

The distance spectra as such are not very useful when assessing the performance of turbo codes. However, in combination with proper bounding techniques a useful assessement can be made. In this section we use the presented method for calculating the distance spectra in combination with the tangential sphere bound [8, 9], which is an upper bound on the frame-error rate (FER) for ML-decoding of codewords transmitted over an additive white Gaussian noise channel. The tangential sphere bound is used since, in contrast with the union bound, it provides a useful bound on the error performance also below the cut-off rate of the channel. To verify the obtained bounds it would be of interest to present ML-decoding simulations as a comparison. Such comparisons are, however, not feasible due to the prohibitive ML-decoding complexity of turbo codes. We therefore compare with simulation results obtained when using standard suboptimal iterative decoding. The presented simulation results are obtained by 15 decoding iterations employing the modified BCJR (Bahl, Cocke, Jelinek, and Raviv) decoding algorithm [1, 10].

We have compared rate 1/3 turbo codes using interleavers of lengths 100 and 500 bits, and various feedback and feedforward polynomials. We have limited the investigation to the most common setup where two identical component encoders are used. Figures 2, 3 and 4 show the upper bounds (on ML-decoding) together with the simulated (iterative decoding) performances of a large number of randomly chosen interleavers, for a selection of the investigated codes. The simulated error-rates exceed the derived upper bounds, which is a direct result of the suboptimal iterative decoding. Disregarding the absolute values, the bounds give good indication on the relative performance of the different codes/termination methods. Thus, even though the distance spectra of different codes cannot be used for absolute perfomance prediction, they are useful when making intelligent design choices for turbo codes. These design choices are however beyond the scope of this paper.

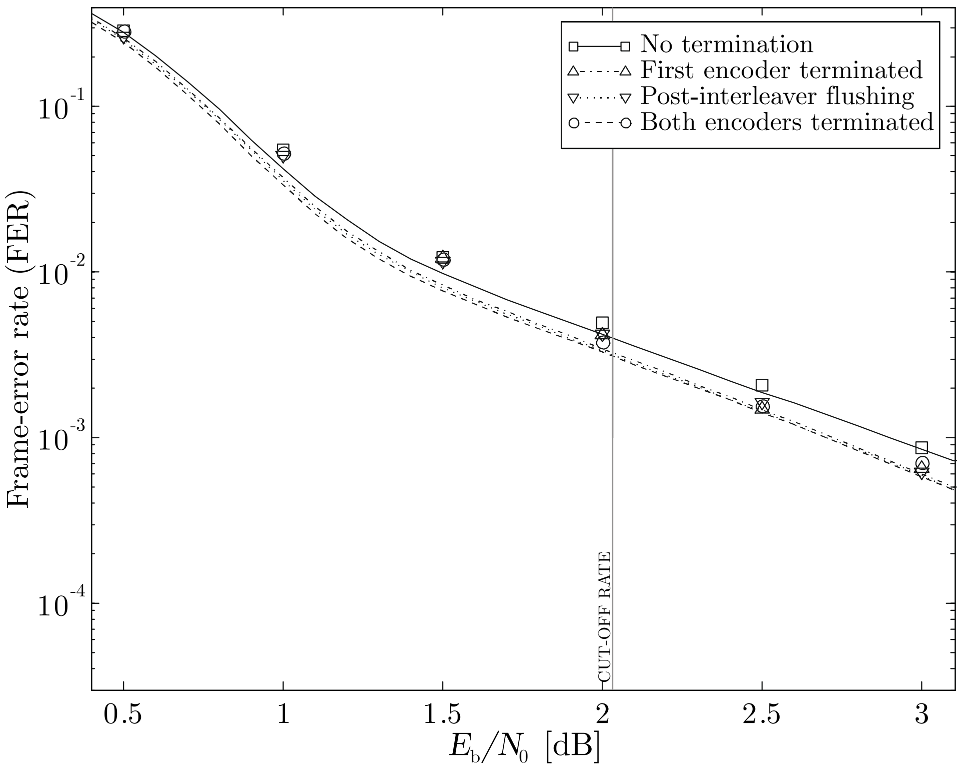

Fig. 2. Performance of turbo codes with 500-bit random interleaving. The feedforward and feedback polynomials are 5_8 and 7_8 respectively. The lines represent the calculated tangential sphere bounds, while the corresponding simulations are indicated by markers. (French: Performance de turbo codes à entrelaceur aléatoire de 500 bits. Les polynômes générateurs direct et récursif sont respectivement égaux à 5_8 et 7_8. Les courbes représentent la borne de Poltyrev et les points indiquent les résultats de simulation.)

IV. CONCLUSIONS

A method for deriving interleaver ensemble average distance spectra of turbo codes using different trellis termination methods has been presented. Using this method, we have investigated four principal classes of trellis termination: no termination, termination of the first encoder, termination of both encoders, and post-interleaver flushing. These methods have been evaluated using component encoders with constraint length 3 and 4, and two interleaver sizes; 100 and 500 bits.

In general, the performance differences between the termination methods are small, except for the case of *no termination.* Among the three methods that involve termination, the best distance spectra are observed with *post-interleaver flushing* and *both encoders terminated.* The performance degradation when no trellis termination is used show little dependence on the interleaver size but it is highly dependent on the choice of component encoders. Especially, the length of the period of the encoder impulse responses is crucial; the larger the period, the larger the performance loss of not using any trellis termination.

The large performance losses suffered when no trellis termination is used are results of the inferior average distance spectra achieved by the ensemble of turbo codes, corresponding to all possible interleavers. However, this performance degradation can be avoided by proper interleaver design, as discussed in [11].

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo Codes," in Proc. 1993 IEEE Int. Conf. on Comm. (ICC), pp. 1064–1070, Geneva, Switzerland, May 1993.

[2] S. Dolinar and D. Divsalar, "Weight distributions for turbo codes using random and non-random permutations," TDA progress report 42-122, Jet Propulsion Lab., Pasadena, CA, Aug. 1995.

[3] D. Divsalar and F. Pollara, "Turbo codes for PCS applications," in IEEE Int. Conf. on Comm., New York, USA, 1995.

[4] S. Benedetto and G. Montorsi, “Unveiling turbo codes: Some results on parallel concatenated coding schemes,” IEEE Trans. on Inf. Theory, vol. 42, pp. 409–428, Mar. 1996.

[5] A. S. Barbulescu and S. S. Pietrobon, “Terminating the trellis of turbo-codes in the same state,” Electronics Letters, vol. 31, pp. 22–23, Jan. 1995.

[6] M. Hattori, J. Murayama, and R. J. McEliece, “Pseudo-random and self-terminating interleavers for turbo codes,” in Winter 1998 Inf. Theory Workshop, San Diego, USA, Feb. 1998.

[7] P. Guinand and J. Lodge, “Trellis termination for turbo encoders,” in 17th Biennial Symp. On Communications, Kingston, Canada, May 1994.

[8] G. Poltyrev, “Bounds on the decoding error probability of binary linear codes via their spectra,” IEEE Trans. on Inf. Theory, pp. 1284–1292, Jul. 1994.

[9] I. Sason and S. Shamai. “Improved upper bounds on the ML decoding error probability of parallel and serial concatenated turbo codes via their ensemble distance spectrum,” IEEE Trans. on Inf. Theory, vol. 46, no. 1, pp. 24–47, Jan. 2000.

[10] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal decoding of linear codes for minimizing symbol error rate,” IEEE Transactions on Information Theory, vol. 20, pp. 284–287, Mar. 1974.

[11] J. Hokfelt, O. Edfors and T. Maseng, “On the theory and performance of trellis termination methods for turbo codes,” Accepted for publication in IEEE Journal on Selected Areas in Communication, 2001.

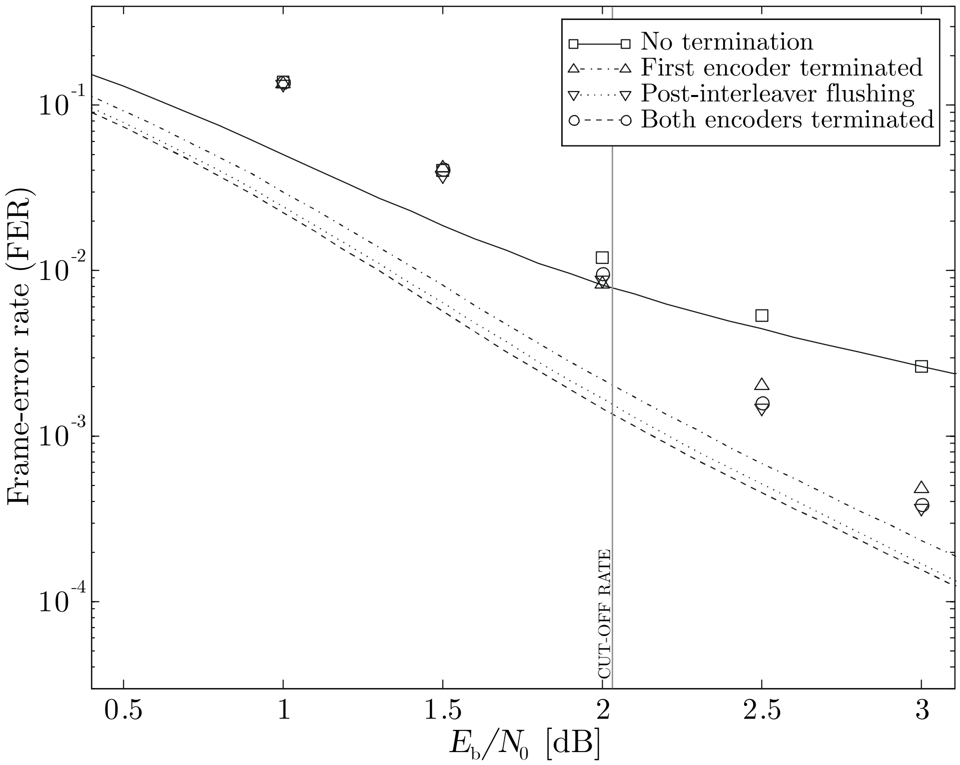

Fig. 3. Performance of turbo codes with 100-bit random interleaving. The feedforward and feedback polynomials are 17_8 and 15_8 respectively. The lines represent the calculated tangential sphere bounds, while the corresponding simulations are indicated by markers. (French: Performance de turbo codes à entrelaceur aléatoire de 100 bits. Les polynômes générateurs direct et récursif sont respectivement égaux à 17_8 et 15_8. Les courbes représentent la borne de Poltyrev et les points indiquent les résultats de simulation.)

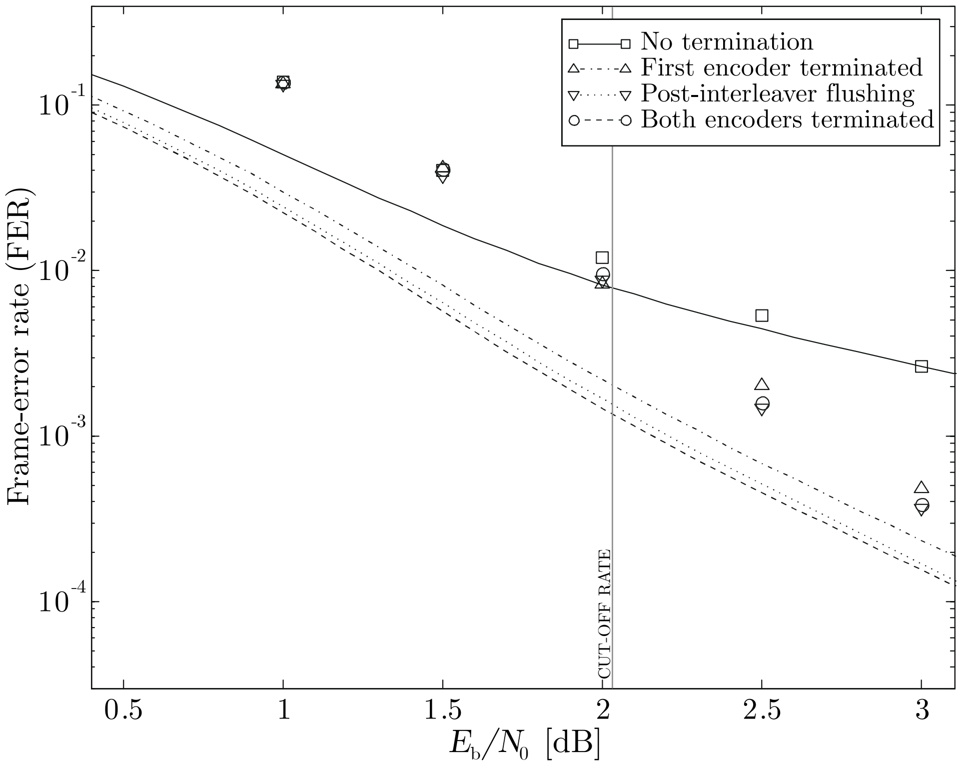

Fig. 4. Performance of turbo codes with 500-bit random interleaving. The feedforward and feedback polynomials are 17_8 and 15_8 respectively. The lines represent the calculated tangential sphere bounds, while the corresponding simulations are indicated by markers. (French: Performance de turbo codes à entrelaceur aléatoire de 500 bits. Les polynômes générateurs direct et récursif sont respectivement égaux à 17_8 et 15_8. Les courbes représentent la borne de Poltyrev et les points indiquent les résultats de simulation.)