Distance spectra of turbo codes using different Trellis termination methods

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

Published in:
Proc. International Symposium on Turbo codes

2000

Distance Spectra of Turbo Codes using Different Trellis Termination Methods

Johan Hokfelt, Carl Fredrik Leanderson, Ove Edfors and Torleiv Maseng

Dept. of Applied Electronics, 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: Johan.Hokfelt@tde.lth.se

Abstract: *This paper investigates the implications of the choice of trellis termination method for Turbo codes, and illuminates the origin of the performance degradation often experienced without trellis termination. Analythical 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. For 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.

1. 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 [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 the result of *interleaver edge effects* [2]. This paper describes interleaver edge effects 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 as an upper bound to the achievable maximum likelihood (ML) decoding performance. Already for relatively modest interleaver lengths, it is observed that the simulated error performance is close to the performance indicated by the tangential sphere bound.

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].

2. 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 already 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 were presented. In this section we summarize this method, and present a method to include the influences of different trellis termination methods.

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

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

for an (n, N)-code, where N is the length of the interleaver and n is the codeword length. $A_{w,j}$ is the number of codewords with input weight w and parity weight j, and W and Z are dummy variables.

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

$$A_w(Z) \triangleq \sum_{j=0}^{n-N} A_{w,j} Z^j, \quad w = 0, 1, \ldots, N,$$

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 with length N as

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

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 multiplicities a_d of codewords with Hamming weight d are achieved as

$$a_d = \sum_{w=1}^{N} A_{w,d-w}^{TC}, \tag{2}$$

where $A_{w,d-w}^{TC}$ are the coefficients in the Turbo code CWEF, i.e. $A_w^{TC}(Z) \triangleq \sum_{j=0}^{n-N} A_{w,j}^{TC} Z^j$.

When deriving the CWEF of the component codes of Turbo codes, it is common to take only the error events that end up in the zero-state into account. This corresponds to taking only zero-terminating input sequences into consideration. 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.

2.1. 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 of the number of paths that lead to other states. This can be obtained by partitioning the IRWEF defined by (1) into a *state-dependent* counterpart, 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 [5]. Let $A_{t,s,w,j}$ denote 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 state dependent IRWEF is recursively calculated as

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

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 transition. This recursive procedure 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.

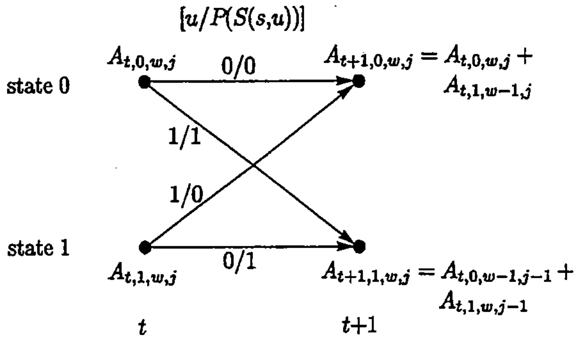

Figure 1: Recursive calculation of codeword multiplicities for a two-state encoder.

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. These multiplicities form the corresponding CWEFs according to $E_w^{C_l}(Z) = \sum_{j=0}^{n-N} E_{w,j}^{C_l} Z^j$, $l = 1,2$. The overall CWEFs, including both zero-terminating codewords and edge effects, are then obtained as

$$\tilde{A}_w^{C_l}(Z) = A_w^{C_l}(Z) + E_w^{C_l}(Z), \tag{3}$$

and the resulting CWEF for the Turbo code is

$$\tilde{A}_w^{TC}(Z) = \frac{\tilde{A}_w^{C_1}(Z)\, \tilde{A}_w^{C_2}(Z)}{\binom{N}{w}}. \tag{4}$$

Note that $A_w^{C_l}(Z)$ includes only trellis paths that end in the zero-state after N transitions. Thus, $A_{w,z}^{C_l} = A_{N,0,w,j}^{C_1}$. The difference in the $\tilde{A}_w^{C_l}(Z)$:s for different trellis termination methods is in the way the $E_{w,j}^{C_l}$:s are calculated. Below, we calculate the $E_{w,j}^{C_l}$:s 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 correspond to interleaver edge effects are calculated by summing the number of paths that end in non-zero states after N trellis transitions. Thus,

$$E_{w,j}^{C_1} = \sum_{s=1}^{2^{m_1}-1} A_{N,s,w,j}^{C_1}$$
$$E_{w,j}^{C_2} = \sum_{s=1}^{2^{m_2}-1} A_{N,s,w,j}^{C_2},$$

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, $\tilde{A}_w^{TC}(Z)$, is calculated using (3) and (4).

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 encoder. 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,

$$E_{w,j}^{C_1} = 0$$
$$E_{w,j}^{C_2} = \sum_{s=1}^{2^{m_2}-1} A_{N,s,w,j}^{C_2}.$$

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 has 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 [6,7]. It is then sufficient to append 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 [8]. This is achieved without any restrictions on the choice of interleaver, but with a slight increase in the number of input bits dedicated for 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 entirely 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,

$$E_{w,j}^{C_1} = \sum_{s=1}^{2^{m_1}-1} A_{N,s,w,j-F_1(s)}^{C_1}$$
$$E_{w,j}^{C_2} = \sum_{s=1}^{2^{m_2}-1} A_{N,s,w,j-F_2(s)}^{C_2},$$

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.

3. EVALUATION

The tangential sphere bound [9] is used for the comparisons of the error performances of the described trellis termination methods, for transmission over an additive white gaussian noise channel. The tangential sphere bound is used since it, in contrast to the union bound, provides a useful bound on the error performance for ML-decoding also below the cut-off rate of the channel. We have compared rate 1/3 Turbo codes using interleavers of lengths 100 and 500 bits, and various component encoders. Figures 2, 3 and 4 show the upper bounds together with the simulated performances of a large number of randomly chosen interleavers, for a selection of the investigated codes. The simulation results are achieved after 15 decoding iterations employing the full BCJR algorithm.

The simulated performances rank in the same order as predicted by the derived distance spectra; however, the simulation results are actually *above* the derived bounds. This is not an inconsistency, bearing in mind that the tangential sphere bound is valid for ML-decoding, while the simulation results are obtained with suboptimal iterative decoding.

4. 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 has been evaluated using component encoders with

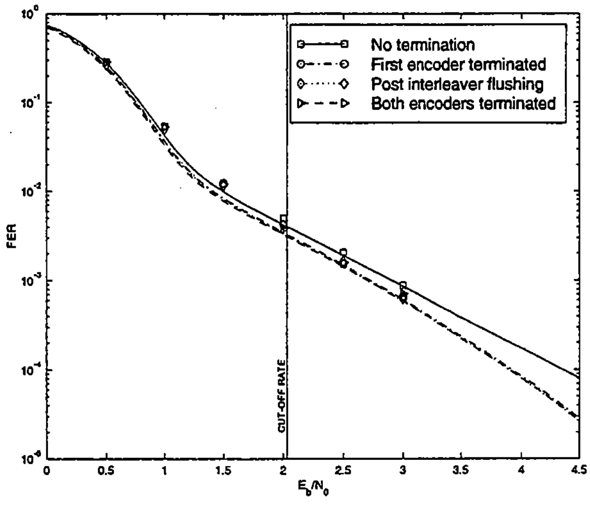

Figure 2: Performance of Turbo codes with 500-bit random interleaving. Parity- and feedback polynomials are equal to 5_8 and 7_8 respectively.

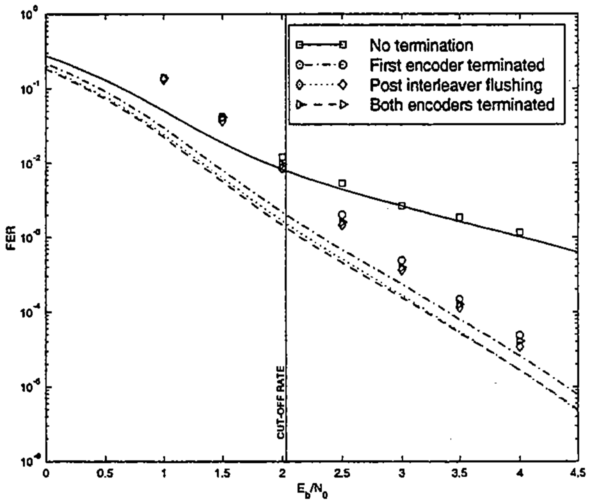

Figure 3: Performance of Turbo codes with 100-bit random interleaving. Parity- and feedback polynomials are equal to 17_8 and 15_8 respectively.

constraint length 3–5, 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 amount of performance degradation from not using any trellis termination is not very dependent of 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. Plots supporting this conclusion had to be omitted.

The performance loss experienced without trellis termination is the result of the inferior distance spectra achieved for the ensemble of interleavers. However, this deterioration can be avoided by proper interleaver design. Such investigations are under preparation.

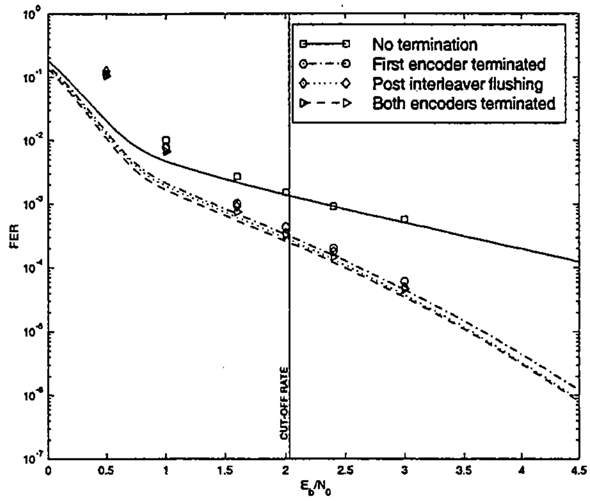

Figure 4: Performance of Turbo codes with 500-bit random interleaving. Parity- and feedback polynomials are equal to 17_8 and 15_8 respectively.

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 nonrandom permutations." TDA progress report 42-122, Jet propulsion Lab., Pasadena, CA, August 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, March 1996.

[5] W. Feng, J. Yuan, and B. Vucetic, "A code matched interleaver design for turbo codes." Submitted to IEEE Trans. on Comm.

[6] 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.

[7] 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.

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

[9] G. Poltyrev, "Bounds on the decoding error probability of binary linear codes via their spectra," *IEEE Trans. on Inf. Theory*, pp. 1284–1292, July 1994.