A survey on trellis termination alternatives for turbo codes

Hokfelt, Johan; Edfors, Ove; Maseng, Torleiv

Published in:
Proc. IEEE Vehicular Technology Conference

DOI:
10.1109/VETEC.1999.778457

1999

Citation for published version (APA):
Hokfelt, J., Edfors, O., & Maseng, T. (1999). A survey on trellis termination alternatives for turbo codes. In *Proc. IEEE Vehicular Technology Conference* (Vol. 3, pp. 2225-2229). IEEE - Institute of Electrical and Electronics Engineers Inc.. https://doi.org/10.1109/VETEC.1999.778457

Total number of authors:
3

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

A SURVEY ON TRELLIS TERMINATION ALTERNATIVES FOR TURBO CODES

Johan Hokfelt, Ove Edfors, Torleiv Maseng

Department of Applied Electronics, Lund University
Ole Römers väg 3, SE-221 00 Lund, SWEDEN
PH: +46 (46) 222 30 21 FAX: +46 (46) 222 47 18
e-mail: {Johan.Hokfelt, Ove.Edfors, Torleiv.Maseng}@tde.lth.se

Abstract – Numerous strategies and methods for trellis termination of turbo codes have been presented and proposed in the literature. In this paper the most common trellis termination methods are compared and their relative performances are investigated. An important observation is that the performance of a termination method is highly dependent on the particular interleaver choice. Conclusions should thus be drawn for the combination of interleaver and termination method, and not for the termination method alone. Another important conclusion is that the particular choice of termination method is not crucial for the error correcting performance, as long as certain precautions are taken regarding the choice of interleaver.

I. INTRODUCTION

Turbo codes are in general implemented as two convolutional encoders in parallel, where the input to the second encoder is an interleaved version of the original input sequence (1). The constituent encoders are *recursive*, implying that the next state of an encoder is dependent on both the next input symbol and the current state. Since the encoders are fed different input sequences, they will in general not be in the same state by the end of each input block, and hence different tail sequences are required in order to bring the encoders back to the zero states. Due to this need for separate tail sequences, a number of strategies for trellis termination of turbo codes have been proposed, see e.g. (2),(3),(4),(5) and (6).

The obvious reason for terminating a trellis is that it avoids poor decoding performance experienced near the end of the trellis, if it is truncated in an unknown state. However, the major concern regarding trellis termination for turbo codes is related to the distance spectrum of the code, and not to the decoding performance near the end of the trellis. This paper describes and compares commonly used trellis termination strategies for turbo codes, and investigate the performance differences achieved with the different methods.

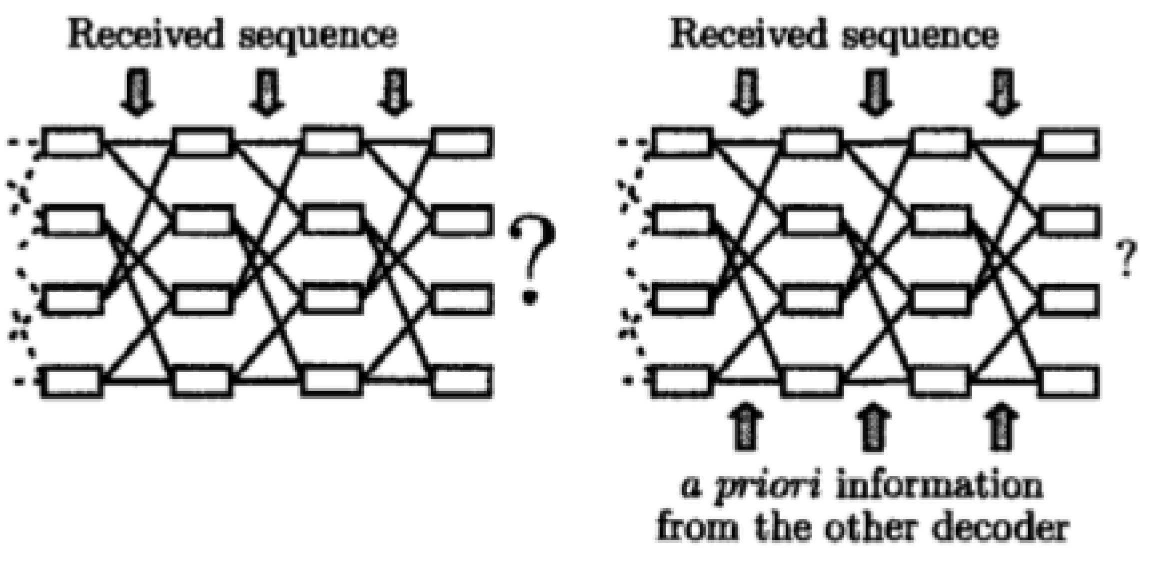
In Section II general differences between trellis termination of convolutional codes and turbo codes are discussed. The relevant issues in the case of turbo codes are elaborated in more detail, without specifying a specific termination method. In Section III these issues are applied to a number of distinct termination alternatives, discussing their respective advantages and disadvantages. Simulation results are also given, supporting the conclusions drawn. Throughout this paper, rate-1/3 turbo codes with two constituent encoders using generator polynomials $(1, 17/15)_{oct}$ are used. The interleaver size is 500 bits, and the constituent decoders use the MAP-algorithm with 15 decoding iterations.

II. TRELLIS TERMINATION ISSUES

The primary issues of trellis termination of turbo codes are (*i*) the decoding performance near to the end of a sequence, and (*ii*) the influence on the distance spectra of the code. Both these issues are discussed in the following subsections.

Decoding performance near truncated trellises

Consider the decoding situation near the end of a trellis, depicted in Figure 1 for both a convolutional code and a constituent code in a turbo code. If the convolutional code is truncated without tail bits, the only input to the decoder that improves the trellis state probabilities is the received symbols near the end of the sequence. The decoding performance is significantly harmed by such a truncation; using tail bits in this case considerably increase the decoding reliability. For the constituent turbo decoder the situation is different; it has access to *a priori* probabilities for each trellis transition. These *a priori* probabilities, if they have high quality, can significantly reduce the state uncertainty towards the end of the truncated trellis. As the iterative decoding proceeds, it is therefore expected that turbo codes become less and less dependent on the use of trellis termination.



(a) (b)

Figure 1. Truncated trellis for (a) an ordinary convolutional code, and (b) a constituent turbo code. The state uncertainty of the turbo code is smaller, as a result of the extrinsic information (*a priori* probabilities) from the other constituent decoder.

The above discussion intuitively justifies that trellis termination should be less important for turbo codes than for ordinary convolutional codes. Why, then, do turbo codes with different trellis termination methods have unequal error correcting performances? We argue next that this is often a result due to the impact on the distance spectra of the codes.

Interleaver edge effects

In general, low weight code words originate from low weight information sequences for which both parity sequences also have low weight. A Low weight parity sequence is in turn generated whenever the detour from the all-zero state path in the trellis is of limited length. Figure 2 depicts such a situation for an information sequence of Hamming weight three.

Interleaver edge effects refer to the implications on the distance spectrum resulting from the truncation of the encoder input sequences implied by the interleaver. Due to this truncation, a low weight parity word can be produced even though the encoder input does not force the encoder back to the zero state. The seriousness of the interleaver edge effects depends highly on the interleaver choice; with proper interleaver design its effects can be fully avoided. Using pseudo-random interleavers, the likelihood of severe distance spectrum degradation depends on the trellis termination method. Figure 3 depicts an interleaver edge effect example with a specific interleaver and no trellis termination. Even though neither of the encoder input sequences return the encoders to the zero-states, both trellis detours are of limited length and the parity words are of low weight. This particular interleaver choice produces a code word of Hamming weight 8, regardless of the length of the interleaver.

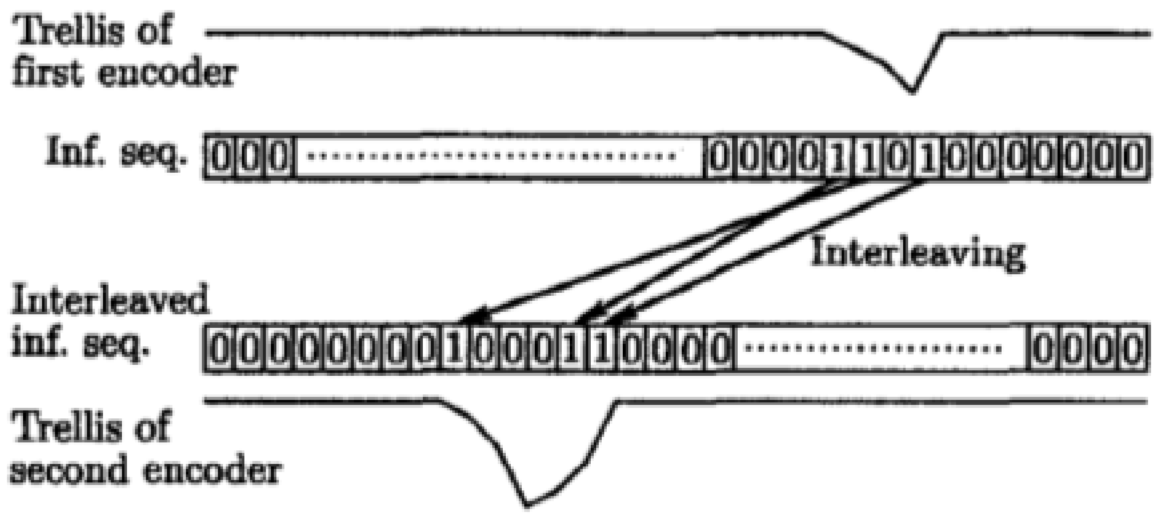


Figure 2. Example of an information sequence and an interleaver that result in short trellis detours and thus a low weight code word.

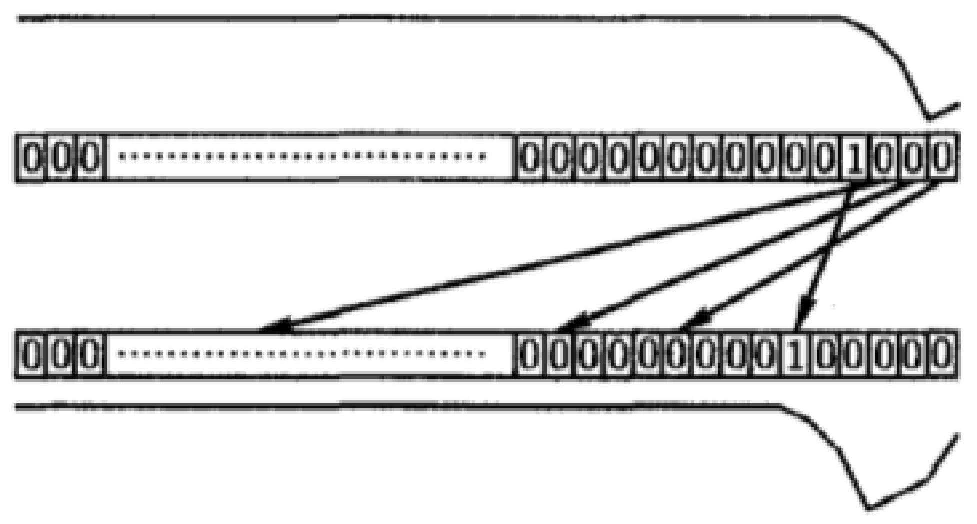


Figure 3. Example of interleaver edge effects when no trellis termination is employed.

III. TRELLIS TERMINATION ALTERNATIVES

The choice of trellis termination method affects both issues related to trellis termination described in the previous section. In the following subsections we investigate how this impact vary depending on the chosen termination strategy. The trellis termination methods are classified into four general strategies:

1. No termination of any constituent encoder, e.g. (5),
2. Termination of the first constituent encoder only, e.g. (7),
3. Termination of both encoders with individual tail sequences (6), and
4. Termination of both encoders with a single tail sequence, by imposing certain interleaver restrictions, e.g. (4),(8),(9) and (10).

The above methods are in the following evaluated using 500-bit interleavers. For the first three methods, which impose no restriction on the interleaver, we use an ordinary block interleaver and a reverse block interleaver (11), both with 20 rows and 25 columns. For the forth termination method we use block helical simile interleavers (4). In addition, we evaluate interleavers designed specifically for each terination method. These interleavers are based on the correlation criterion described in (10), with specific attention paid to the characteristics of each termination method.

No trellis termination

When using no termination at all, the trellises of both constituent encoders are left in unknown states. Naturally, this strategy results in the weakest decoding performance close to the end of the trellises. Further, it is very sensitive to the choice of interleaver, due to the relatively high probability of deteriorated performance resulting from interleaver edge effects. This was exemplified in Figure 3, illustrating the interleaver edge effects for a specific pseudo-random interleaver. Figure 4 depicts the situation for an ordinary block interleaver, which has extremely poor performance when no trellis termination is used. This is because a block interleaver, which write data in rows and read in columns, interleaves the last bit in the sequence to the exact same position. Hence the trellis detours for the error events corresponding to an all-zero sequence except for a one in the last position become extremely short. The resulting minimum distance of the code is 3 (regardless of the size of the interleaver), for a rate-1/3 code.

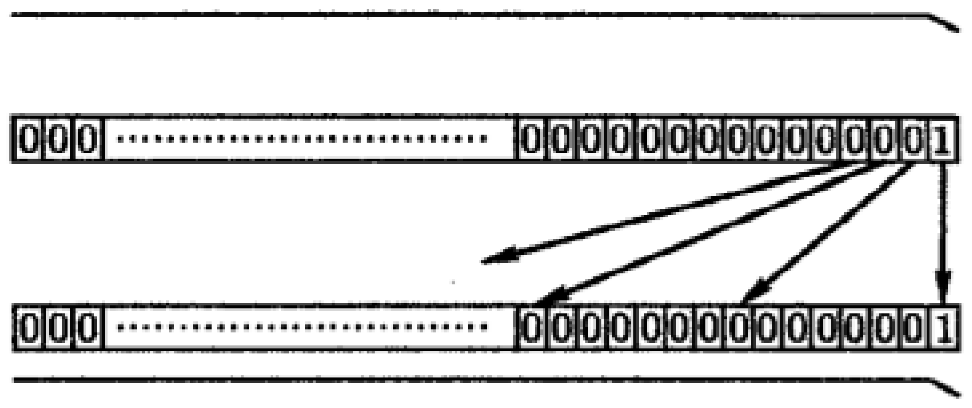


Figure 4. Turbo code with no trellis termination and an ordinary block interleaver. The interleaver edge effects result in a code with minimum distance 3.

The poor performance achieved with ordinary block interleavers can be avoided by the use of reverse block interleavers, in which bits are read in columns from the lower right corner instead of from the upper left. With more sophisticated interleaver design methods it is possible to further improve the performance. Figure 5 shows frame- and bit-error rates for turbo codes using a block interleaver, a reverse block interleaver, and an interleaver designed specifically for no trellis termination.

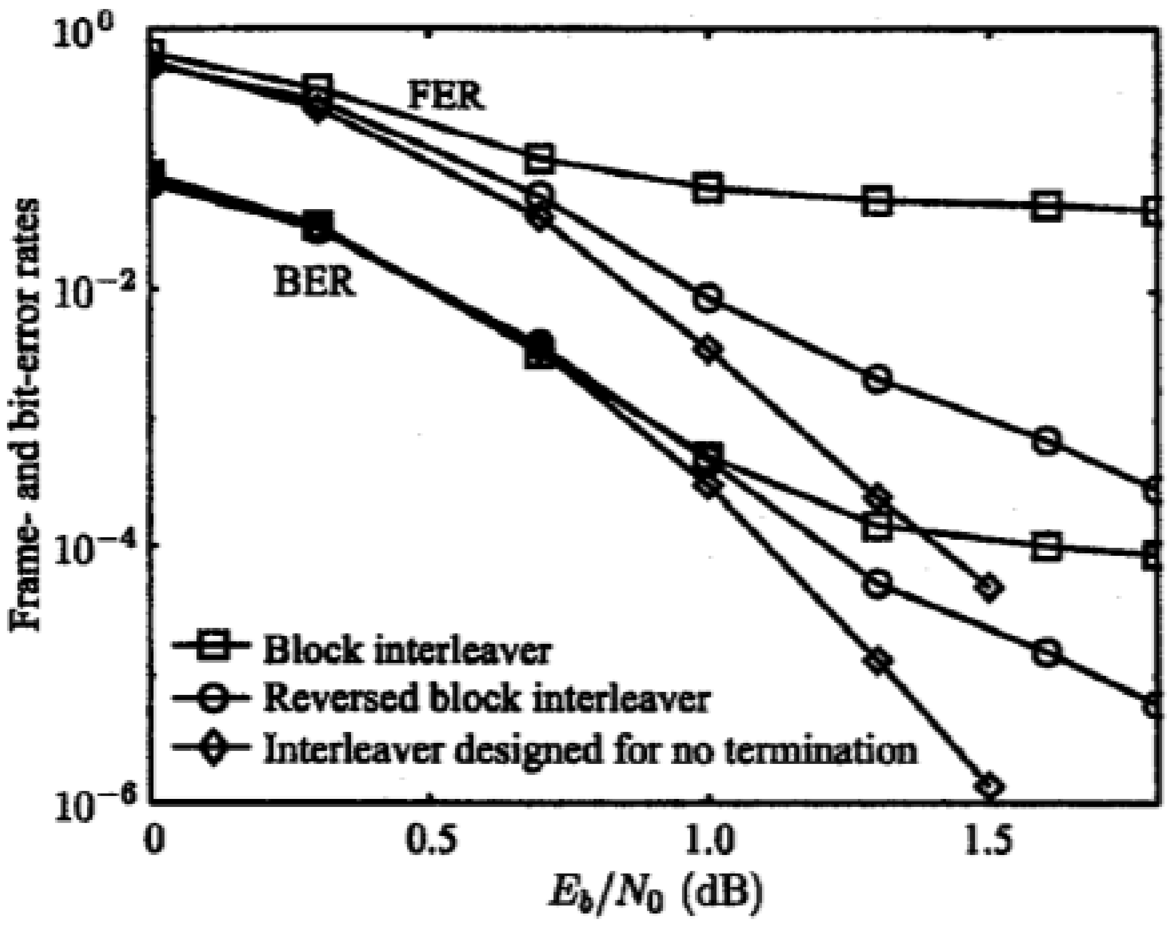


Figure 5. Simulated frame- and bit-error rates for Turbo codes with no trellis termination, for three different interleavers. The block interleaver has significantly worse performance than the reverse block interleaver, due to the interleaver edge effects.

Termination of the first constituent encoder

With this strategy tail bits are appended to the information sequence so that the first constituent encoder is terminated in the zero state. The tail bits are included in the sequence entering the interleaver, as shown in Figure 6. Termination of the first encoder only. The likelihood of unfortunate edge interleaver effects is significantly reduced.. The final state of the second constituent encoder after encoding the interleaved sequence is unknown to the decoder. The decoding performance near the end of the trellis is therefore somewhat worse for the second constituent decoder.

The likelihood of obtaining a code with poor minimum distance due to interleaver edge effects is substantially reduced when terminating the first trellis. This is because the error event in the first trellis contains at least two positions with ones. The probability (assuming a pseudo-random interleaver) that both these positions are interleaved to positions near the end of the interleaved sequence is significantly smaller than the corresponding probability for only one position, as was the case with no trellis termination at all. For the same reason, the very low minimum distance achieved with an ordinary block interleaver and no trellis termination is effectively removed by terminating the first trellis. It is therefore expected that the block- and reverse block interleavers will perform essentially the same when the first trellis is terminated. This is verified by the simulation results in Figure 7, which also shows the performance of an interleaver designed specifically for this termination method.

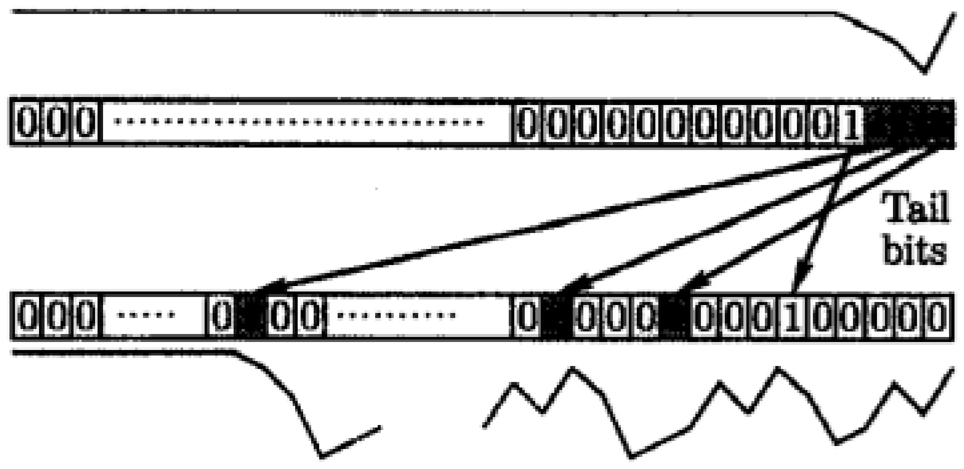


Figure 6. Termination of the first encoder only. The likelihood of unfortunate edge interleaver effects is significantly reduced.

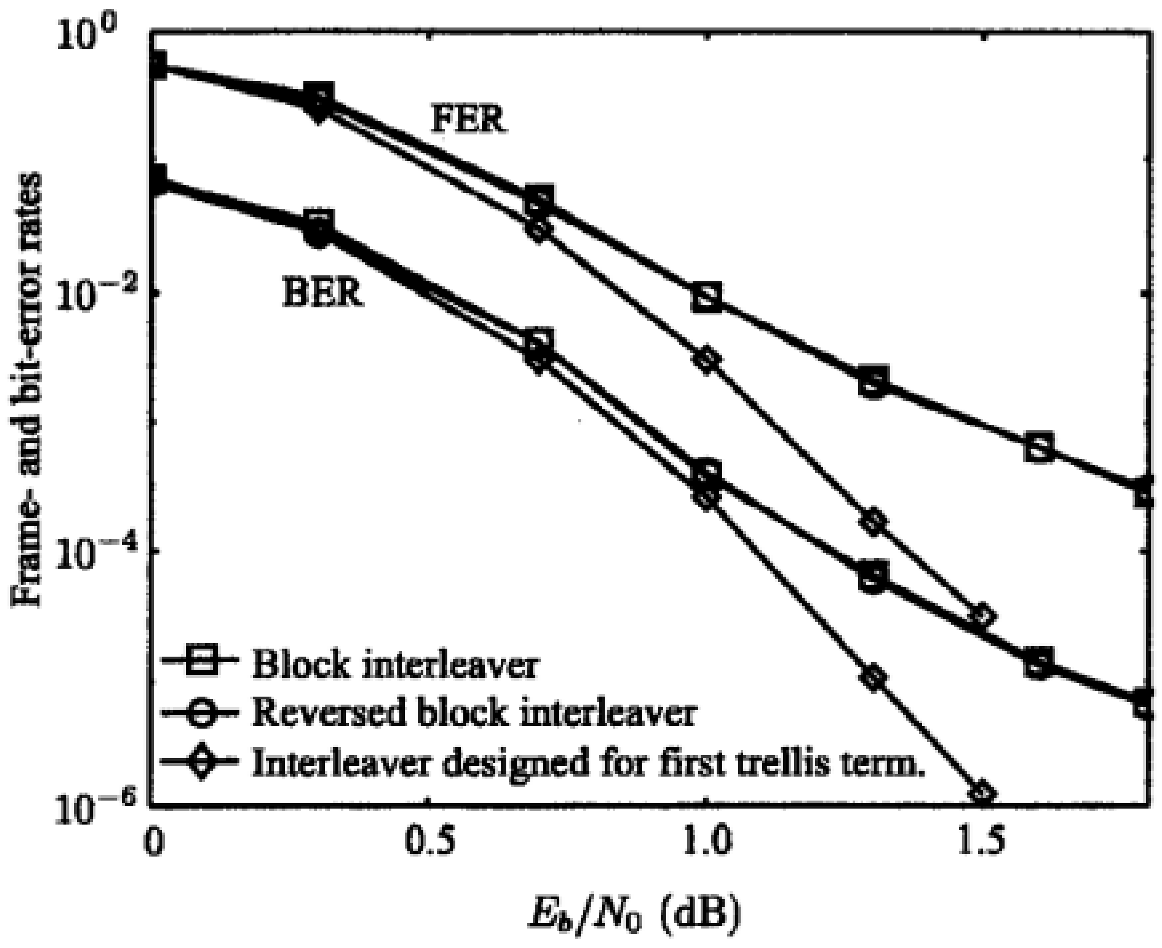


Figure 7. Simulated frame- and bit-error rates when the first trellis is terminated, for three different interleavers. The block- and reverse block interleavers now have the same performance.

Termination of both encoders with individual tail sequences

Separate tail sequences can be appended to each encoder input sequence. This method exhibits a similar risk of achieving a low minimum distance due to edge effects as when using no trellis termination at all. This is exemplified in Figure 8, showing the same pseudo-random interleaver as in Figure 3. Even though the behavior is same, the Hamming distance is increased due to the extra Hamming weight in the two tail sequences and their respective parity sequences.

The performances of the ordinary block-, reverse block- and the interleaver specifically designed for this termination method are shown in Figure 9. As expected, the performance of the block interleaver suffers somewhat from the edge effects.

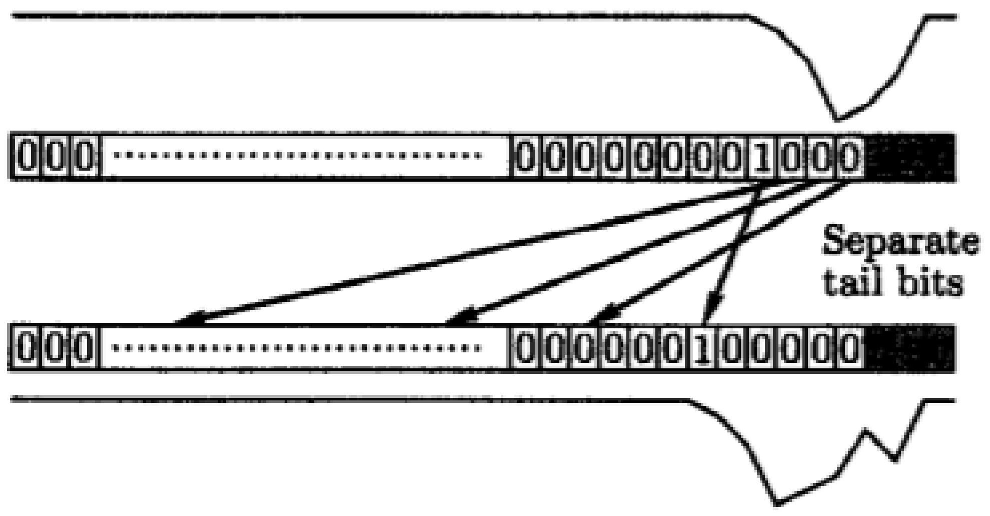


Figure 8. Trellis termination using separate tail sequences, with a high risk of unfortunate interleaver edge effects.

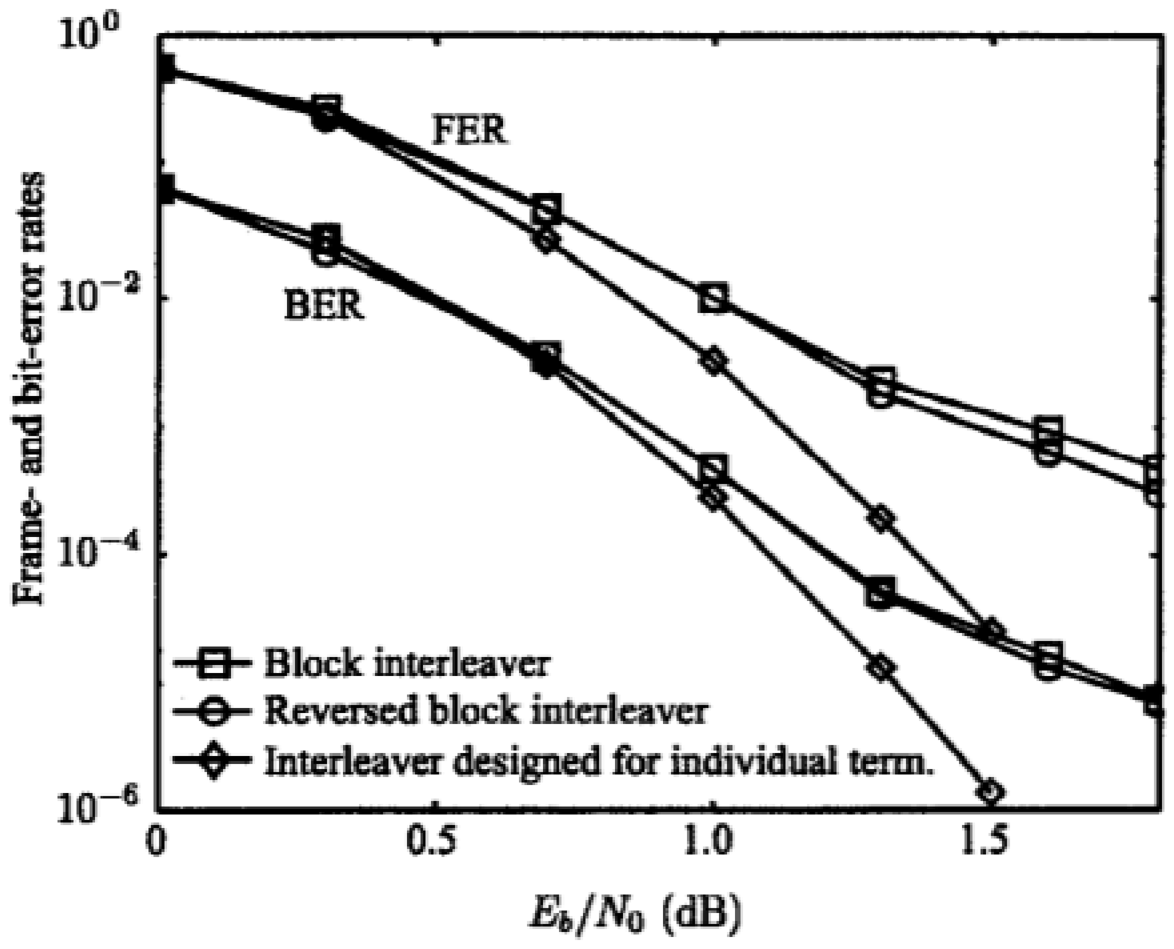


Figure 9. Simulated frame- and bit-error rates when both trellises are terminated with individual tail sequences. The block interleaver perform slightly worse than the reverse block interleaver, due to the interleaver edge effects.

Termination of both encoders with a single tail sequence

By imposing certain restrictions on the interleaver, it can be designed to terminate both constituent encoders in the same state, see e.g. (4),(9). Both encoders can thus be terminated in their zero-states by appending *one* set of tail bits to the information sequence, and including these bits in the sequence entering the interleaver. The interleaver restrictions allow interleaving only within certain subsets of the input sequence. Each subset consists of the positions that are separated from each other by a multiple of L bits, where L is the period of the feedback polynomial of the constituent encoders. Let $\pi(i)$ denote the position of input bit i after interleaving. The interleaving rule must obey $\pi(i) \bmod L = i \bmod L$, i=1, 2,…, N. A method for constructing such *self-terminating* interleavers called *block helical simile* interleavers is described in (4).

The drawback with designing self-terminating interleavers is that the restrictions impose less design freedom. Consequently, the possibilities of avoiding mappings that create low weight code words are reduced. This drawback is worst for small interleavers, typically up to a couple of hundred bits. The simulated performances of two block helical simile interleavers and a specifically designed self-terminating interleaver are shown in Figure 10.

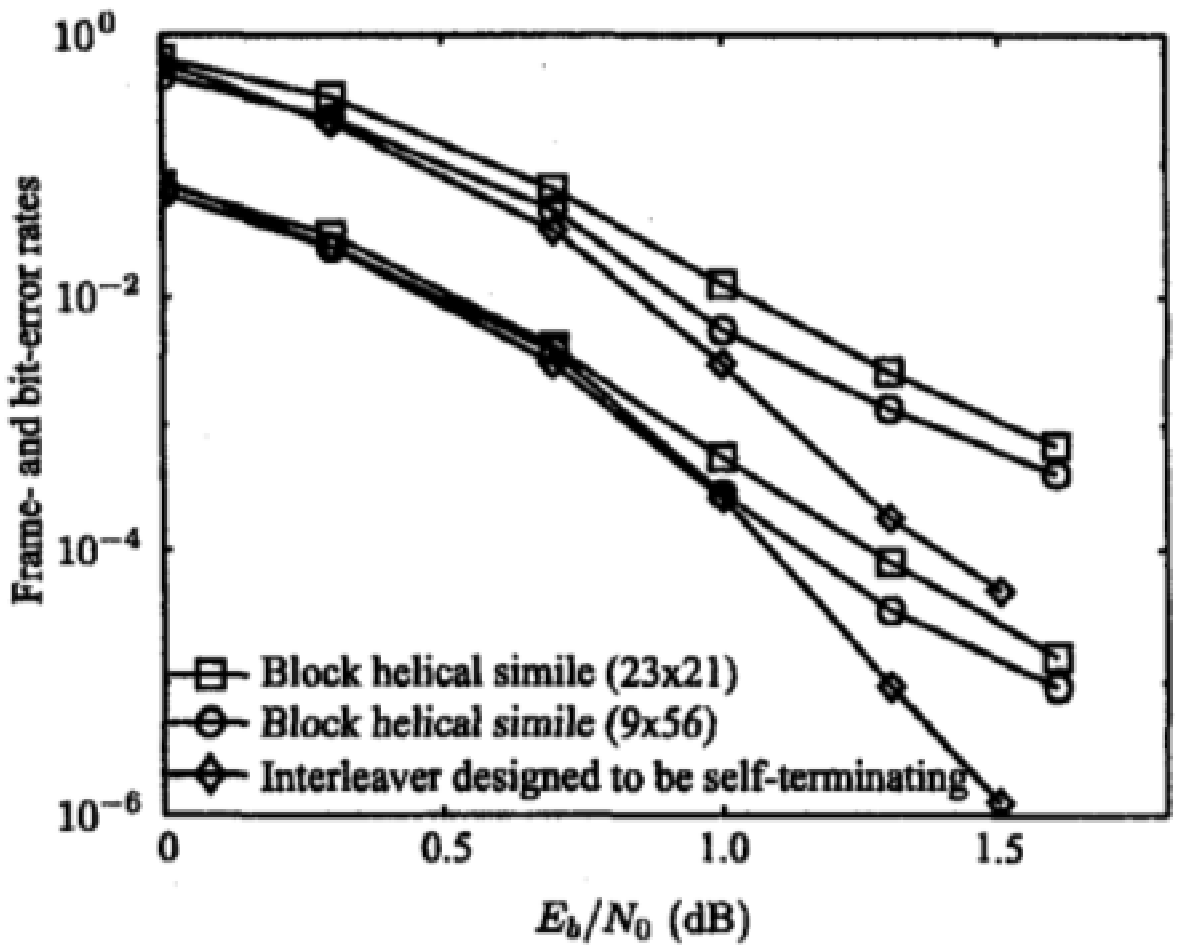


Figure 10. Simulated frame- and bit-error rates when both trellises are terminated using one tail sequence, accomplished with self-terminating interleavers.

IV. PERFORMANCE COMPARISON

Advantages and disadvantages of various trellis termination methods were described in the previous section. For each termination method, we evaluated interleavers designed specifically for each method. The error correcting performances of all these interleavers/termination methods are compared in Figure 11. The performance difference between the compared methods is evidently very small. Again, this conclusion is only valid for interleavers that are suitable for each specific method; as seen in the previous section, some interleavers perform fairly well with some termination methods, and perform bad with others.

V. CONCLUSIONS

We have compared four different trellis termination methods for turbo codes. For each termination method, it is important to use interleavers that are well suited for that specific method. The performance differences between different termination methods are very small, as long as the interleavers are suitable chosen.

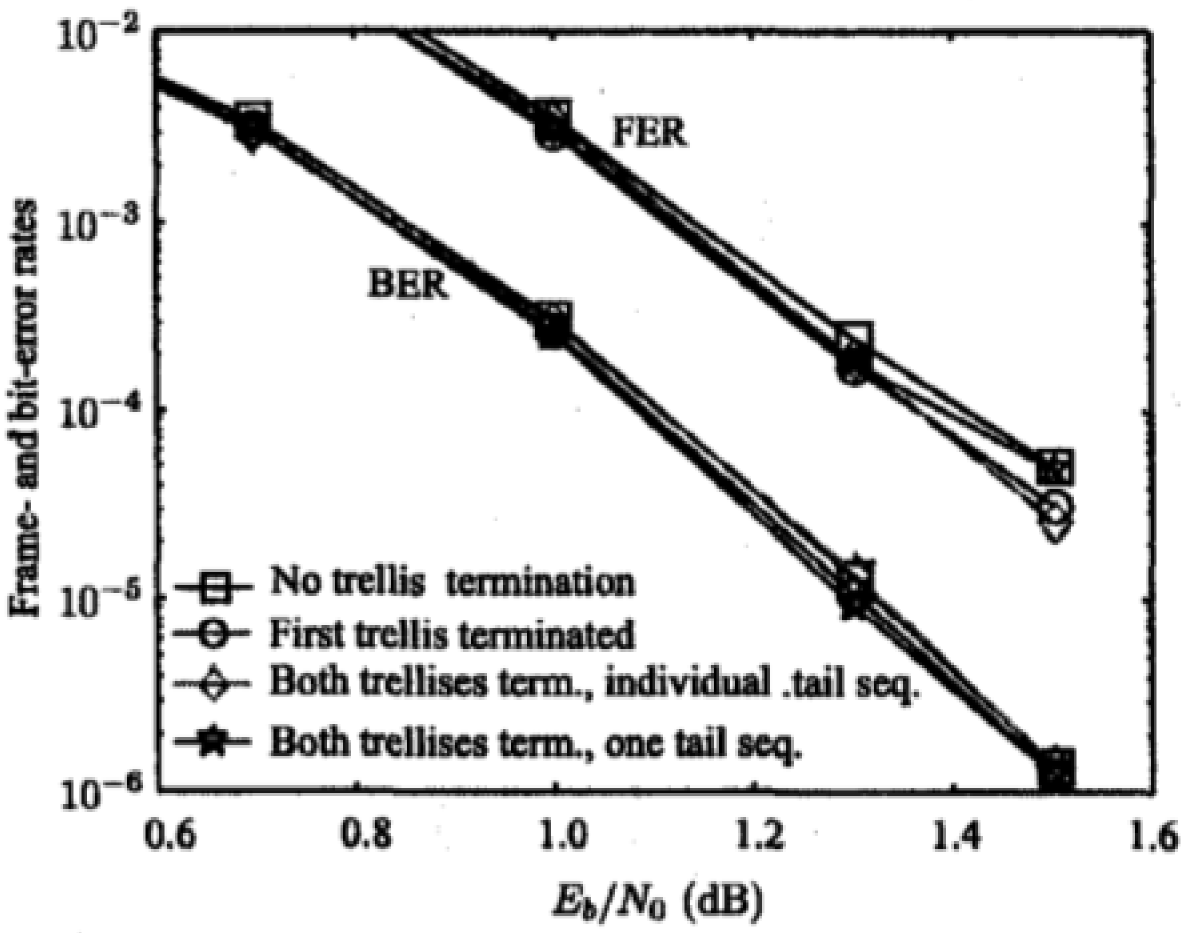


Figure 11. Performance comparison of the different trellis termination methods, using the best performing interleavers for each method.

REFERENCES

(1) C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," IEEE Transactions on Communications, vol. 44, pp. 1261-1271, October 1996.

(2) W. Blackert, E. Hall and S. Wilson, "Turbo code termination and interleaver conditions," Electronics Letters, vol. 31, pp. 2082-2084, November 1995.

(3) O. Joerssen and H. Meyr, "Terminating the trellis of turbo-codes," Electronics Letters, vol. 30, pp. 1285-1286, August 1994.

(4) A. S. Barbulescu and S. S. Pietrobon, "Terminating the trellis of turbo-codes in the same state," Electronics Letters, vol. 31, pp. 22-23, January 1995.

(5) M. C. Reed and S. S. Pietrobon, "Turbo-code termination schemes and a novel alternative for short frames," in Seventh IEEE International Symposium on Personal, Indoor and Mobile Communications, (New York, USA), 1996.

(6) D. Divsalar and F. Pollara, "Turbo codes for PCS applications," in IEEE International Conference in Communications, (New York, USA), 1995.

(7) P. Robertson, "Improving decoder and code structure of parallel concatenated recursive systematic (turbo) codes," in IEEE International Conference on Universal Personal Communications, (San Diego, USA), 1994.

(8) C. Berrou and M. Jézéquel, "Frame-oriented convolutional turbo codes," Electronics Letters, vol. 32, pp. 1362-1364, July 1996.

(9) M. Hattori, J. Murayama, and R. J. McEliece, "Pseudo-random and self-terminating interleavers for turbo codes," in Winter 1998 Information Theory Workshop, (San Diego, USA), 1998.

(10) J. Hokfelt, O. Edfors, and T. Maseng, "Interleaver design for turbo codes based on the performance of iterative decoding," in IEEE International Conference on Communication, (Vancouver, Canada), 1999.

(11) H. Herzberg, "Multilevel Turbo Coding with Short Interleavers," IEEE Journal on Selected Areas in Communication, Vol. 16, NO. 2, February 1998.