Turbo Codes with Non-Systematic Constituent Codes*

Oscar Y. Takeshita, Oliver M. Collins, and Daniel J. Costello, Jr.

9th NASA Symposium on VLSI Design
Albuquerque, New Mexico

Abstract

Turbo codes are usually constructed using systematic recursive convolutional codes (SRCC's) as constituent codes. In this paper, we introduce a new class of turbo codes that uses non-systematic recursive convolutional codes (NSRCC's) as constituent codes. A systematic constituent code then becomes a particular case of this more general class. The use of this larger class of constituent codes enhances the number of possible codes in the search space. We also introduce a modified iterative decoding method for this more general class of turbo codes. The decoding technique is no more complex than the standard iterative decoding algorithm.

1 Introduction

The usual view of turbo codes is two systematic recursive convolutional codes (SRCC's) linked by an interleaver [1][2][3][4]. The systematic bits that are identical to both constituent codes are transmitted only once, and the two decoders "share" the noisy received systematic symbols. Iterative decoding is then accomplished by exchanging extrinsic reliability information about the systematic bits between the two decoders.

In this paper we propose a new class of turbo codes that uses non-systematic recursive convolutional codes (NSRCC's) as constituent codes for turbo codes. This class of NSRCC's contains the usual SRCC's as a particular case. We also propose a modified iterative decoding method for these more general turbo codes.

*This work was supported by NSF Grants NCR95-22939, NCR96-96065 and CCR-0075221 and NASA Grant NAG5-8355. Oscar Y. Takeshita is with the Department of Electrical Engineering, Ohio State University, Columbus, OH 43210. Oliver M. Collins and Daniel J. Costello, Jr. are with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46656.

2 Non-Systematic Turbo Codes and Decoding

A non-systematic turbo code is shown in Fig. 1. Note that the systematic bits are not transmitted, unlike the original systematic turbo code.

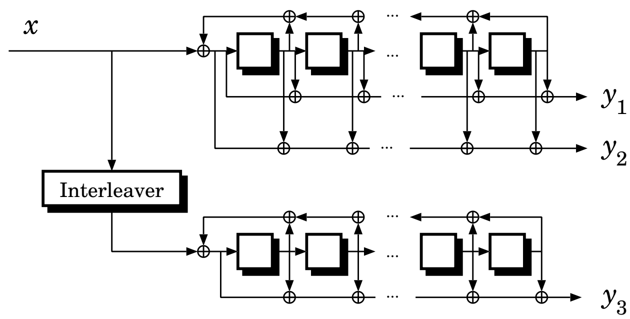

Figure 1: A non-systematic turbo code.

A block diagram of the modified iterative decoder is shown in Fig. 2. For each a posteriori probability (APP) decoder we use the standard BCJR [5] algorithm. With the BCJR algorithm, we can compute the a posteriori likelihood ratio of the information bits given the received noisy symbols and the a priori probabilities of the information bits. Decoding is performed by iterating the following algorithm:

1. Using the received noisy symbols $\tilde{y}_1$, $\tilde{y}_2$, and the a priori likelihood ratio of the information bit x_i^1 for the first encoder (which equals 1 for all bits in the first iteration, assuming equally likely and iid distributed information bits, and equals the deinterleaved version of the sequence x_e^2 from the previous iteration (see step 2) for subsequent iterations), compute x_c^1, which is the a posteriori likelihood ratio of the information bit x_t^1 divided by x_i^1. Note that this step does not assume that y_1 or y_2 are systematic bits.

2. Using $\tilde{y}_3$ and the interleaved version of the sequence x_c^1 as the a priori likelihood ratio of the information bit x_i^2 for the second encoder, compute x_e^2, which is the a posteriori likelihood ratio of the information bit x_t^2 divided by x_i^2.

3. If this is the last iteration, make a decision by using the sequence of deinterleaved likelihood ratios for x_t^2; otherwise, proceed to the next iteration starting at step 1.

Because we have lifted the restriction of using SRCC's as constituent codes, the number of possible constituent codes is now much larger than for systematic turbo codes. We have conducted a limited search for good turbo codes using NSRCC's as constituent codes. As a first step, we sorted the codes with respect to the average signal-to-noise ratio (SNR) of the estimated information bits at the end of one decoding iteration for 10 frames, using an information block size of 1024 bits and without any puncturing, i.e., for an overall code rate of 1/3. Since the first decoding iteration is crucial for successful convergence of the iterative

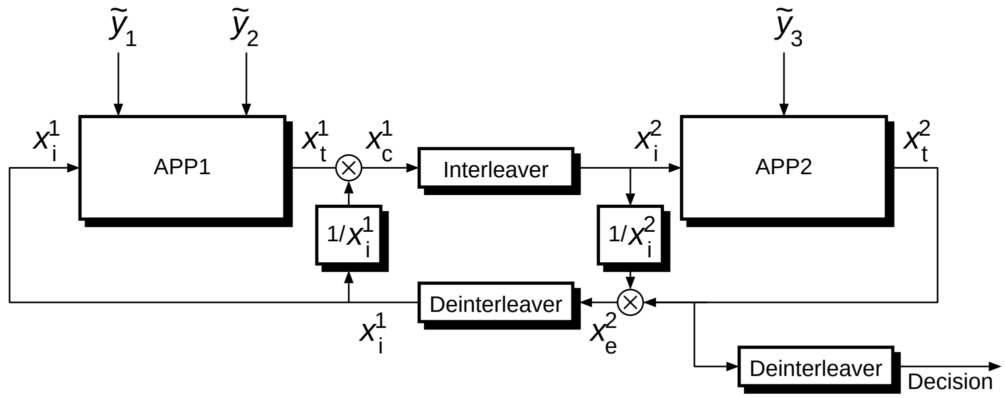

Figure 2: Block diagram of the decoder.

decoding process, this is a sensible strategy for selecting constituent codes. Second, we placed a restriction of 4 on the maximum degree of the polynomials in the constituent codes. In other words, each constituent code had at most 16 states. This left a total of about 500,000 codes (after eliminating some equivalent codes by symmetry). Then we selected every 1000th code of the best 100,000 codes resulting from the first step and simulated each of these up to 18 iterations for 5000 frames at an SNR of 0.56dB (Eb/No).

Finally, we sorted the codes simulated according to the bit-error-rate (BER) obtained after the 18th iteration. Among this limited group of 100 codes, we identified an asymmetric turbo code [6] with generators

$$[1 + D/1 + D + D^2 \quad 1 + D + D^3/1 + D + D^2]$$

and

$$[1 + D^3 + D^4/1 + D + D^2]$$

as the best.

The BER and frame-error-rate (FER) curves for this non-systematic turbo code (see Fig. 3) are very close to those of the original systematic turbo code reported in [1], which is known to have very good performance in the *waterfall* region, i.e., in the steep portion of the curve where the performance improves dramatically with only a small increase in SNR.

As in the case of systematic turbo codes, we can compute the *effective free distance* [7] of the non-systematic turbo code. The *effective free distance* is the minimum weight of a codeword caused by a weight-2 input sequence whose ones are separated by some multiple of the cycle-length of the feedback encoder of the constituent codes. The *effective free distance* of the new non-systematic turbo code is 10, which is identical to the *effective free distance* of the code in [1]. But note that this new code has a smaller state complexity (8-states for the first constituent code and 16-states for the second constituent code) than the code in [1] (16-states for both constituent codes).

Since the codes that we have simulated are just a small fraction of the total number of codes with 16 or fewer states for each constituent code, there is the potential for even

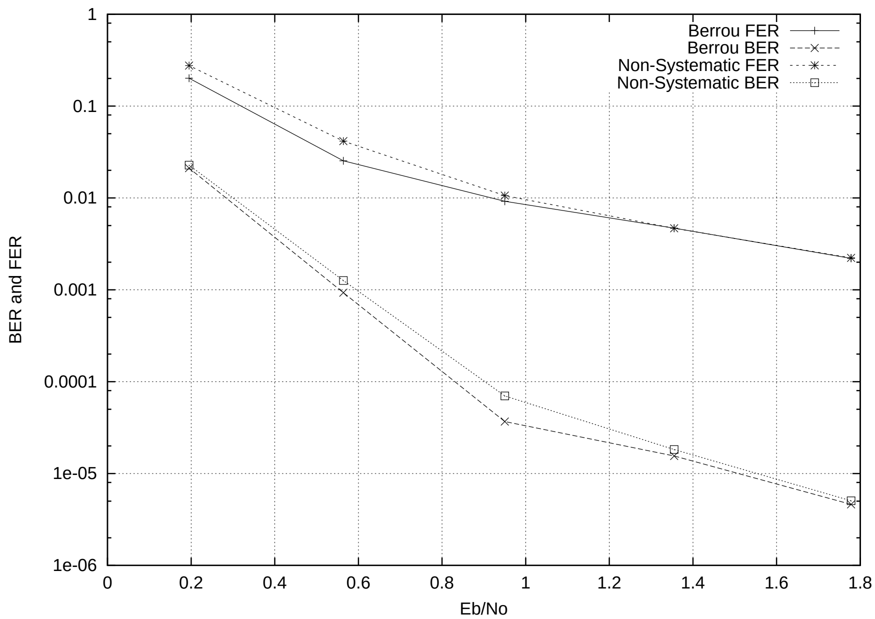

Figure 3: BER and FER simulations of a non-systematic turbo code and the original systematic turbo-code.

better codes to be found using a more extensive code search or a clever code construction algorithm. We are currently doing a systematic search for NSCRCC's with larger values of *effective free distance* and small effective multiplicities.

3 Bandwidth Efficient Non-Systematic Turbo Codes

Several bandwidth efficient turbo codes have been proposed in the literature [8][9][10][11]. In [11] a bandwidth efficient turbo code is obtained by mapping a binary turbo code to 16QAM symbols. In particular, a 2bits/sec/Hz scheme can be obtained by puncturing a rate 1/3 turbo code up to rate 1/2 and mapping the resulting binary symbols to 16QAM symbols. Note that several puncturing patterns and mappings are possible, all yielding different performance [12]. We have implemented a similar scheme as in [11] using the non-systematic turbo code described in the previous section. The scheme in [11] uses a systematic constituent code given by the generator $[1 \quad 1+D+D^2+D^4/1+D^3+D^4]$. Half of the parity bits for each of the binary constituent codes are punctured in [11]. For the non-systematic turbo code, we have chosen to puncture half of the parity bits corresponding to the generators $1+D+D^3/1+D+D^2$ and $1+D^3+D^4/1+D+D^2$. The number of information bits per frame is 4096, resulting in a frame size of 2048 16QAM symbols. BER and FER results for 10 decoding iterations per frame are shown in Fig. 4.

The results for the non-systematic turbo code are not as good as in [11]. We believe that this can be explained by the fact that the underlying binary non-systematic turbo code was designed for a good performance at low SNR's (similar to the original systematic turbo code). However, bandwidth efficient turbo codes operate at higher SNR's. It thus may be

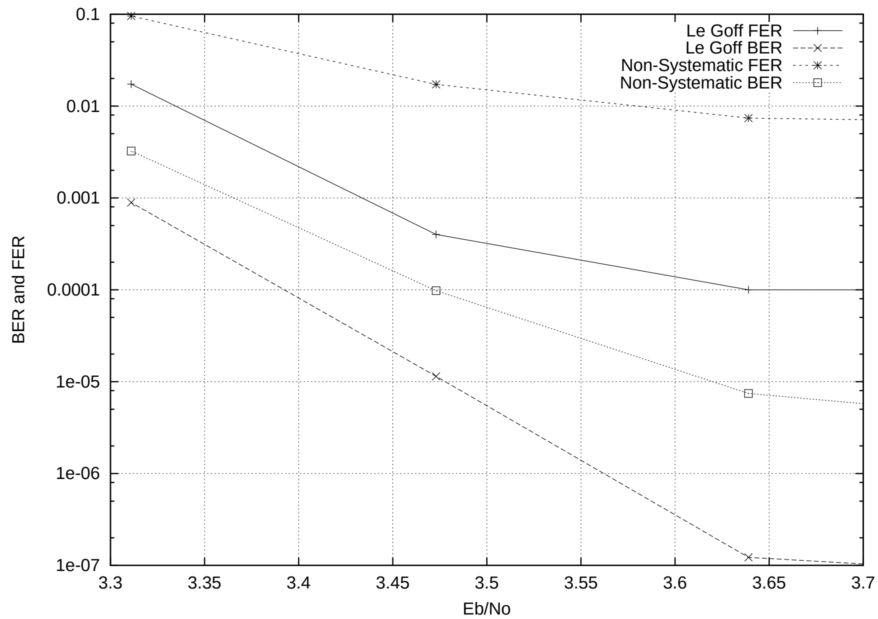

Figure 4: BER and FER simulations of bandwidth efficient non-systematic and systematic turbo codes.

preferable to use underlying binary turbo codes that perform better at higher SNR's, such as ones using primitive feedback polynomials and large cycle-lengths (the periodicity of the impulse response of a recursive convolutional code) in the constituent codes [4]. Indeed the codes in [11] use constituent codes with primitive feedback polynomials and a cycle-length of 15 symbols. We thus believe that we will be able to obtain good bandwidth efficient non-systematic turbo codes by using binary non-systematic turbo codes that perform well at high SNR's. We are currently investigating this approach to the design of bandwidth efficient non-systematic turbo codes.

4 Conclusions

We have shown that binary turbo codes using NSRCC's as constituent codes can be iteratively decoded and that they perform at least as well as turbo codes using SRCC's. In particular, a good NSRCC was found in a limited search, thus demonstrating the potential that this larger code space may contain even better codes. Also, this code has a smaller state complexity that the code in [1] with similar performance. We have used this binary non-systematic turbo code to implement a bandwidth efficient code using a similar construction as in [11]. The performance of the non-systematic scheme was worse than the systematic scheme in [11]. However, we note that the underlying binary non-systematic turbo code had good performance in the waterfall, i. e., at low SNR's. Bandwidth efficient turbo codes must operate at higher SNR's, and thus it is expected that the best underlying binary turbo codes may be ones that perform well at high SNR's. This may explain why in the systematic case [11] a constituent code with a primitive feedback polynomial (known to have good performance at high SNR's) was chosen. The next step in the design of good bandwidth efficient non-systematic turbo codes is to search for binary non-systematic turbo codes that perform well at high SNR's.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes," in *Proc. IEEE Int. Conf. on Commun.*, May 1993, pp. 1064–1070.

[2] S. Benedetto and G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated coding schemes," *IEEE Trans. on Inform. Theory*, vol. 42, no. 2, pp. 409–428, Mar. 1996.

[3] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. on Inform. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

[4] L. C. Perez, J. Seghers, and D. J. Costello, Jr., "A distance spectrum interpretation of turbo codes," *IEEE Trans. on Inform. Theory*, vol. 42, no. 6, pp. 1698–1709, Nov. 1996.

[5] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. on Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

[6] O. Y. Takeshita, O. M. Collins, P. C. Massey, and D. J. Costello, Jr., "A note on asymmetric turbo-codes," *IEEE Commun. Letters*, vol. 3, no. 3, pp. 69–71, Mar. 1999.

[7] D. Divsalar and R. J. McEliece, "Effective free distance of turbo codes," *Electronics Letters*, vol. 32, no. 5, pp. 445–446, Feb. 1996.

[8] S. Benedetto, D.Divsalar, G. Montorsi, and F. Pollara, "Bandwidth efficient parallel concatenated coding schemes," in *Electronics Letters*, 1995, vol. 31, pp. 2067–2069.

[9] Patrick Robertson and Thomas Woerz, "Novel bandwidth efficient coding scheme employing turbo codes," in *Proc. IEEE International Conference on Communications*, 1996, pp. 962–967.

[10] P. Robertson, "An overview of bandwidth efficient turbo coding schemes," in *Proc. International Symposium on Turbo codes*, Sept 1997, pp. 103–110.

[11] Stephane Le Goff, Alain Glavieux, and Claude Berrou, "Turbo-codes and high spectral efficiency modulation," in *ICC 94*, 1994, pp. 645–649.

[12] T. M. Duman and M. Salehi, "Performance bounds for turbo-coded modulation systems," *IEEE Trans. Commun.*, vol. 47, no. 4, pp. 511–521, Apr. 1999.