LOWERING THE ERROR FLOOR FOR TURBO CODES

Mats Öberg and Paul H. Siegel
Center for Wireless Communications, ECE
University of California, San Diego
La Jolla, California 92093-0407
E-mail: moberg@ucsd.edu psiegel@ucsd.edu

Abstract— We present a simple method to improve the performance of turbo codes, using an analysis of the code distance spectrum. The sparseness of the spectrum permits explicit identification of the positions in the information frame that are affected by low-distance error events. A modified encoding is then used to avoid these error events and reduce the error floor that characterizes the code performance at moderate-to-high signal-to-noise ratios. For large interleaver sizes, the only cost is a very slight reduction of code rate. The expected performance improvement of this scheme is verified by empirical simulation results.

I. Introduction and Background

Two characteristic features of turbo code performance are the small bit-error-rate (BER) achieved even at very low signal-to-noise ratio (SNR) E_b/N_0 and the flattening of the error-rate curve – the so-called "error floor" – at moderate and high values of SNR.

Recently Perez et. al. [1] analyzed the codeword weight distribution and multiplicity – the distance spectrum – of turbo codes and offered an explanation for both of these phenomena. They showed that, at moderate-to-high SNR, the performance approaches the *free-distance asymptote*, P_{free}, given by

$$P_{free} = \frac{N_{free}\tilde{\omega}_{free}}{N} Q\left(\sqrt{d_{free}\frac{2RE_b}{N_0}}\right) \quad (1)$$

where R is the code rate, d_{free} is the code free distance, N_{free} is the multiplicity of minimum-weight codewords, N is the interleaver length, and $\tilde{\omega}_{free}$ is the average information weight of the input sequences causing free-distance codewords. They showed that the shallow slope of the error floor is a manifestation of the small free distance and the appearance of the floor only at low error rate reflects the very small effective multiplicity N_{free}/N.

Moreover, the analysis of "average" turbo codes in [1] shows that the very small multiplicity extends to low-weight codewords in general.

This research is supported by the Royal Swedish Academy of Sciences and the Center for Wireless Communications at UCSD

A consequence of this "spectral sparseness" is that the free-distance asymptote dominates the error-rate union bound even at very low SNR.

In this paper, we propose a simple method to lower the error floor of turbo codes. By examining the interplay between the constituent convolutional codes and a pseudo-random interleaver, we derive a list of the non-zero bit positions in the information frames that produce minimum-weight turbo codewords. A modified encoder places dummy bits in these positions, and after the turbo decoding is completed, the contents of these positions in the decoded frame are discarded. This process effectively removes the contribution to the BER of the free-distance error events, resulting in a lowering of the error floor and a change in its slope. By determining bit positions that correspond to other low-weight codewords, we can further improve code performance by applying this same procedure to those frame positions. The small multiplicity of low-weight codewords in the turbo code implies that the rate loss incurred by this encoder modification is negligible for large interleaver size.

II. Properties of the Constituent Encoders

We consider a turbo encoder consisting of two identical, rate 1/2, recursive systematic convolutional (RSC) encoders in parallel concatenation, along with an interleaver which permutes the input bits of the first encoder before applying them to the input of the second encoder. Input bits in a frame of length N are encoded by the first RSC encoder, producing what we call the *first dimension* codeword. The same information frame is permuted by the interleaver and encoded by the second RSC encoder, generating the *second dimension* sequence. The turbo codeword contains the first information frame, along with the parity bits from each encoder. To increase the overall rate of the encoder from 1/3 to 1/2, we follow the usual practice of puncturing every other parity bit in each dimension.

Berrou, *et al.* [2] proved a property of RSC encoders that plays an important role in the char-

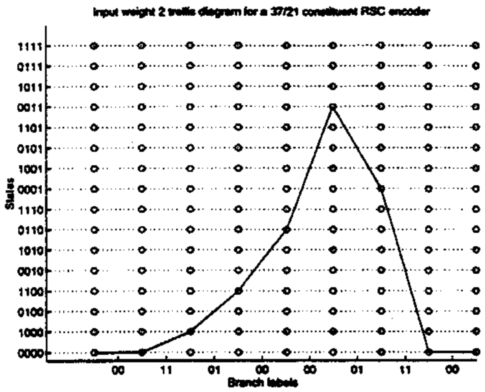

Fig. 1. Response of a octal (37,21) RSC encoder with input pattern 0100001000

acterization of minimum-distance error events in turbo encoders. They showed that if $h(D)$ is the feedback polynomial of a RSC encoder, and if $h(D)$ is a divisor of $1 + D^L$, then $h(D)$ is also a divisor of $1 + D^{pL}$ for any integer $p \geq 1$. Let 0^i denote a run of 0's of length i. The result implies that an input stream $1\,0^{pL-1}\,1$, where the two ones are separated by $pL - 1$ zeros, for any integer $p \geq 1$, will generate a trellis path that diverges from the all-zero path and then remerges after $pL + 1$ trellis steps. The encoder will leave the all-zero state in response to the first 1, and then will return to that state in response to the second 1. This is illustrated in Fig. 1 for $L = 5$ and $p = 1$.

For the rate 1/2, octal (37,21) encoder, this suggests that weight-2 input sequences in which the non-zero bits are separated by a small multiple of the code constraint length represent good candidates for producing minimum-weight RSC code sequences. The code has free distance $d_{free} = 6$, and Fig. 1 depicts a minimum-weight code sequence that is generated in response to a weight-2 input sequence in which the non-zero bit separation is one constraint length. Input sequences of this kind will play a role in analyzing the turbo code distance spectrum in the next section.

III. The Effect of the Interleaver

The analysis of turbo codes with random interleavers in [1], [3] shows that an "average interleaver" achieves excellent performance not by generating a large free distance, but rather, by drastically reducing the multiplicity of low-distance error events. Moreover, the analysis highlights the important role played by weight-2 information frames in generating low-weight codewords. Indeed, both of these characteristics have been observed empirically in turbo codes using particular pseudo-random interleavers [1].

Low-weight turbo codewords arise when the interleaver maps low-weight information frames that produce low-weight parity in the first dimension into frames that produce low-weight parity in the second dimension. The discussion in the previous section suggests that low-weight turbo codewords may arise from weight-2 information frames with non-zero bit positions $\{i, i + pL\}$ in the first dimension that get mapped by the interleaver to bit positions $\{j, j + rL\}$, where L is the parameter in the divisibility condition of the previous section; i, j are positive integers $\leq N$; and p, r are integers small enough that the corresponding information sequences produce low parity weight at the encoder output.

We analyzed the distance spectrum of a turbo code based upon the aforementioned rate 1/2 constituent encoder and a particular pseudo-random interleaver of size $N = 10000$. (In the analysis, we assume that, at the end of the information frame, the encoders in both dimensions are driven to the all-zero state by appending appropriate tails of length ν.) We found that the free distance of the turbo code is $d_{free} = 6$ with free-distance multiplicity $N_{free} = 4$. As was the case for the $N = 65536$ turbo code discussed in [1], all of the minimum-weight codewords correspond to weight-2 information sequences. In fact, the non-zero bit positions of these weight-2 sequences and their images under the interleaver mapping satisfied the condition described above.

We then determined the codewords of weight 16 or less that are generated by weight-2 inputs where the non-zero information bit positions in both dimensions are separated by small multiples of the constituent code constraint length. The codewords with weight 6 or 8 produced by this search are described in Table I. (It can be shown that the table includes all possible weight-6 codewords.) For each bit pair, the bit separation in the first dimension and the bit separation in the second dimension after permutation by the interleaver are indicated in the column labeled "Mapping". The bit separation is given in terms of the number of constraint lengths $L = 5$ that it spans.

The total numbers of non-zero information bit positions in the calculated codewords of weights 6, 8, 10, 12, 14, and 16 were found to be 8, 26, 38, 34, 24, and 8, respectively.

Mapping	Bit pairs for $d = 6$ (dimension 1)
$1 \mapsto 1$	(1979,1984)(5861,5866)
$1 \mapsto 2$	(785,790)(1383,1388)
Mapping	Bit pairs for $d = 8$ (dimension 1)
$1 \mapsto 2$	(5117,5122)(5827,5832)(9014,9019)
$1 \mapsto 3$	(449,454)(1280,1285)(5566,5571)
$2 \mapsto 1$	(4253,4263)
$2 \mapsto 2$	(1163,1173)
$3 \mapsto 1$	(3518,3533)(7132,7147)
$3 \mapsto 2$	(5804,5819)(9136,9151)
$4 \mapsto 2$	(3132,3152)

TABLE I

WEIGHT-2 INPUTS GENERATING TURBO CODEWORDS OF WEIGHTS 6 AND 8

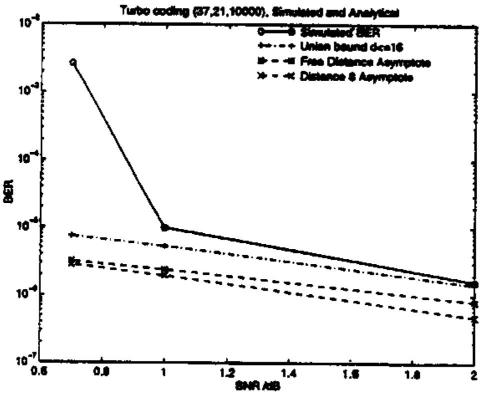

Fig. 2. Simulated BER and asymptotes for distances 6 and 8

IV. Improving BER by Discarding Spectral Lines

Fig. 2 shows the simulated BER as a function of SNR for the $N = 10000$ turbo code of the previous section. Also shown are the free-distance asymptote, the asymptote corresponding to the weight-8 codewords listed in Table I, and the curve reflecting the contributions to the union bound of all of the codewords of weight 16 or less found by the interleaver analysis in Section III.

Fig. 3 shows a histogram of the cumulative number of errors that occurred at each bit position in the information frame for SNR=2 dB. A total of 2,177,440,000 information bits were encoded and 3501 bits were decoded in error, yielding a BER of 1.61×10^{-6}. Of these bit errors, 2959 occurred at positions associated with the codewords of weight 16 or less that we found in our search. In particular, 1676 bit errors corresponded to $d = 6$ codewords, 952 to $d = 8$, 279 to $d = 10$, 44 to $d = 12$, 8 to $d = 14$, and 0 to $d = 16$.

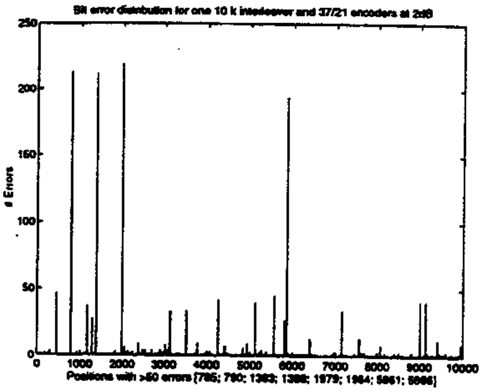

Fig. 3. Cumulative bit errors per bit position in frame

If we decompose the asymptotic performance bound into spectral lines as in [1], it is clear that the spectral line corresponding to $d_{free} = 6$ has the greatest impact on the performance. This suggests a simple modification to the turbo encoder that will improve BER performance. Using the results of the codeword search, the encoder can be modified so that it does not write information in the bit positions associated with minimum-weight codewords in Table I, treating the locations as dummy bit positions. If a free-distance error event occurs during decoding, only these positions in the information frame will be affected, so the actual information bits remain uncorrupted. Thus, the modified encoder effectively removes the contribution to the BER represented by the free-distance spectral line.

Fig. 4 compares the simulated BER curve of Fig. 2 to the BER curve obtained when the bit positions corresponding to minimum-weight codewords are ignored. The simulated BER is reduced by a factor of about 2 relative to the original BER. Also shown is the distance-8 asymptote reflecting the weight-8 codewords listed in Table I. The plot shows the expected changes to the error floor, namely a lowering and change in slope.

The modified encoder incurs a rate loss through this introduction of dummy bit positions. However, the impact is slight for large interleaver size. For the length $N = 10000$ interleaver studied in this paper, the loss is 8 information bits out of 10000, implying a reduction in rate of approximately .04 percent. If applied to the $N = 65536$ turbo code in [1],

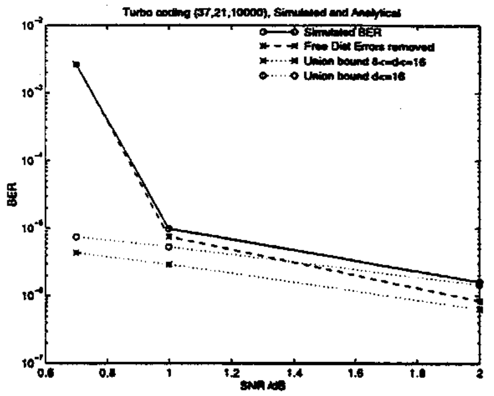

Fig. 4. Simulated BER with distance-6 events removed

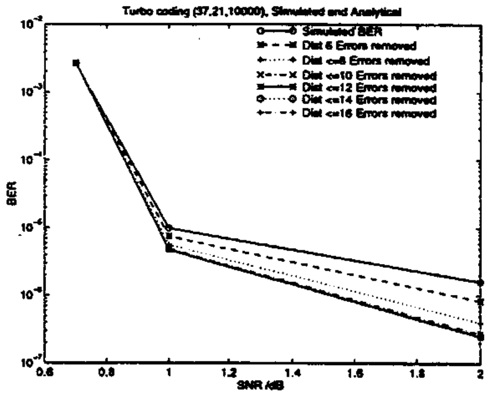

Fig. 5. Simulated BER with information weight 2 events of distance 16 or less removed

the modified encoding procedure would discard only 6 information bits out of 65536, causing a decrease in the code rate amounting to less than one hundredth of one percent.

Given a characterization of bit positions affected by other low-distance error events, the encoder can be further modified to avoid writing information into those locations, thereby providing additional improvement to the BER performance. By referring to Table I, we modified the encoder to avoid positions corresponding to codewords of weight 8, in addition to those of weight 6. The simulated BER is reduced by an additional factor of about 2 , yielding a total reduction factor of 4 relative to the original BER. The rate loss incurred in achieving this reduced BER was slight, amounting to only 34 information bits per frame, or about 0.17 percent.

Finally, the encoder was modified to successively avoid all of the positions affected by the codewords with weight 16 or less that we found in our search. Fig. 5 shows the progressive reduction in the error floor, culminating in a BER improvement that is almost an order of magnitude at an SNR of 2 dB. The rate penalty in this last case amounts to only 138 information bits out of 10000, or a reduction of about 0.7 percent.

V. Conclusions

We have presented a new method to lower the error floor of turbo codes with sparse distance spectrum. The technique involves identification of the bit positions associated with low-weight codewords, and modifying the encoder to avoid writing information in those locations. Simulation results for a particular turbo code using a pseudo-random interleaver of length $N = 10000$ show almost an order of magnitude improvement in BER at SNR=2 dB, with a code rate penalty of less than 1 percent.

In the simulations reported here, the decoder was not modified in any substantial manner, operating as usual and simply ignoring the final estimated bits in the dummy positions. We are investigating modifications to the decoder that will explicitly take into account the values inserted by the encoder into the dummy bit positions.

One can also increase the minimum distance by designing an interleaver that avoids certain permutations, such as those described in Section III. The spirit of this approach is similar to that of the "S-random" permutations introduced by Divsalar and Pollara [4]. It is possible that a combination of this constrained interleaver design and the modified encoding technique presented in this paper may further improve turbo code performance.

References

[1] L. Perez, J. Seghers, and D. J. Costello Jr., "A distance spectrum interpretation of turbo codes," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1698–1709, November 1996.

[2] C. Berrou, S. Evano, and G. Battail, "Turbo-block-codes," in *Turbo coding seminar*, Lund, Sweden, August 1996, pp. 1–8, Lund University.

[3] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 409–428, March 1996.

[4] D. Divsalar and F. Pollara, "Turbo codes for PCS applications," in *Proc. 1995 IEEE International Conference on Communication (ICC)*, Seattle, WA, June 1995, pp. 54–59.