Unequal Error Protection on the Turbo-Encoder Output Bits *

Atousa H. S. Mohammadi, and A. K. Khandani
Dept. of Elec. and Comp. Eng., University of Waterloo

Abstract: Traditional turbo-codes with BPSK modulation scheme, use Equal Error Protection (EEP) for the turbo-encoder output bits. In this paper, it is shown that the role of the encoder output bits is not necessarily the same in determining the code performance. Imposing Unequal Error Protection (UEP) on the output bits can result in improvement of the turbo-code performance.

1 Introduction

Turbo-codes, introduced in 1993 [1], are composed of the parallel concatenation of two (or more) Recursive Systematic Convolutional (RSC) component codes, connected through an interleaver. The output bits corresponding to the kth input bit, d_k, are the systematic bit, x_k^{1s} (which is equal to d_k), and the two parity check bits, x_k^{1p} and x_k^{2p}, which are the outputs of the first and the second encoders, respectively. The encoders are terminated to the all-zero state at the end of each block of input data, and consequently turbo-code is equivalent to a linear block code. The Maximum Likelihood (ML) decoding of turbo-codes is highly complex for large interleaver lengths. Thus, turbo-codes employ a sub-optimum iterative decoding scheme with much less complexity [1, 2].

As usual, the performance of the code is determined by its distance properties. However, due to the presence of the interleaver, it is very difficult to enumerate the exact weight distribution of a turbo-code. The idea of averaging the performance of the code over the structure of the interleaver (uniform interleaver) is introduced in [3]. This is based on a structure in which a hypothetical interleaver produces all the possible permutations of the input with equal probabilities. Using this technique, one can compute an "Average Weight Enumerating Function (AWEF)" for the code, which is independent of the structure of the specific interleaver used.

Conventional turbo-codes assign an equal amount of energy to the three encoder output bits when BPSK modulation is employed for transmission. However, as we will see later, the role of these bits in the weight distribution of the code is not necessarily the same. In this article the problem of "Unequal Error Protection (UEP)" of turbo-encoder output bits is considered. This problem has been previously brought up in [4], where combined turbo-code and modulation was considered. In that case, UEP is imposed on the encoder output bits by the structure of the signal constellation.

In order to achieve UEP, for the BPSK case, the output bits are provided with different noise margins by dividing the bit energy unequally among them. As will be shown, UEP can result in an improvement in the code performance at no extra cost in energy, rate, or complexity.

The outline of the article is as follows. In Section 2, the theoretical aspects of UEP for turbo-encoder output bits are discussed. Section 3, includes some simulation results which show how UEP can be beneficial in performance improvement for turbo-codes, and finally, Section 4 is devoted to some concluding remarks.

2 Unequal Error Protection of Turbo-Encoder Output Bits

For any linear block code used over an AWGN channel, the minimum weight codewords (for the asymptotic case) or the first few lowest weight codewords (for lower SNR's) are dominant in determining the code performance. For a turbo-code, each codeword consists of two groups, the systematic and the parity check bits. The role of these two different groups is not necessarily the same in the weight distribution of the dominant codewords. If one group has a higher contribution in the weight of these dominant codewords, by increasing the energy assigned to this group (and decreasing the energy assigned to the other group), the distance properties, and consequently, the performance of the code can be improved. The following example further illustrates the above discussion.

Example: Consider the very simple case of a turbo-code with interleaver length $N = 3$ and two identical RSC component codes with one memory unit and generator polynomials (1,3). The interleaver maps the input block (a, b, c), to (b, c, a), where $a, b, c = 0$ or 1. Both RSC codes are terminated at the end of each input block by adding one tail bit. The equivalent (3,12) block code then includes the

*This work has been supported in part by the Information Technology Research Centre (ITRC) of Canada and in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

following codewords:

0000 0000 0000, 0011 0010 0110, 0101 0110 1110,
0110 0100 1000, 1001 1110 0010, 1010 1100 0100,
1100 1000 1100, 1111 1010 1010

where in each codeword the first four bits are the systematic and the last eight bits are the parity check bits. Suppose that one unit of energy is assigned to every encoded bit. The Euclidean distances of the 2nd through 8th codewords from the origin (the all-zero codeword) are, $\sqrt{20}$, $\sqrt{28}$, $\sqrt{16}$, $\sqrt{24}$, $\sqrt{20}$, $\sqrt{20}$, and $\sqrt{32}$, respectively. Now suppose that the systematic bits are protected more than the parity bits, such that, each systematic bit receives 2 units of energy and consequently each parity check bit receives 0.5 unit of energy. This time, the distances of the codewords will be, $\sqrt{22}$, $\sqrt{26}$, $\sqrt{20}$, $\sqrt{24}$, $\sqrt{22}$, $\sqrt{22}$, and $\sqrt{40}$, in the same order as before. As can be seen, except for the codeword 0101 0110 1110, the distance of each codeword has either increased, or remained the same. The minimum distance of the code has increased from $\sqrt{16}$ to $\sqrt{20}$, and the overall distance property of the code has improved. Furthermore, if all the energy is assigned to the systematic bits the distances of the first six codewords will become equal to $\sqrt{24}$ and the distance of the last codeword will become equal to $\sqrt{48}$. As is expected and simulation results confirm, the best performance of the code is achieved at this level of UEP, which means that the parity check bits are incurring damage in this code.

For the more general case, assume a turbo-code of rate $r = 1/3$, with two identical RSC codes and interleaver length N. The Weight Enumerating Function (WEF) of the code has the following form:

$$A(W,Z) = \sum_{i,j} A_{i,j} W^i Z^j, \tag{1}$$

where $A_{i,j}$ is the number of codewords having i and j 1's in the systematic and the parity check parts, respectively, and the dummy variables W and Z correspond to the systematic and parity bits, respectively.

If the systematic and parity check bits are equally protected in transmission through the channel, a 1 in the systematic part of a codeword will have the same effect in the distance of that codeword, as a 1 in the parity part. Thus, both W and Z in Eq. 1 can be replaced by the same variable, e.g. w, to result in:

$$A(w) = \sum_{i,j} A_{i,j} w^{i+j}, \tag{2}$$

where $(i+j)$ is the overall weight of the corresponding codeword.

Now, suppose that the bit-energy E_b is divided unequally between the systematic and the parity check bits, such that the energy assigned to each systematic bit is equal to $E_s = x\frac{E_b}{3}$ and the energies given to the first and second parity check bits are equal to $E_p = (\frac{3-x}{2})\frac{E_b}{3}$, where $x \in [0,3]$ (x=1 results in Equal Error Protection (EEP)) [1]. In this case, W and Z in Eq. 1 should be replaced with w^x and $w^{(3-x)/2}$, respectively. A codeword of the original form $W^i Z^j$ is now equivalent to a codeword of distance $ix + j(\frac{3-x}{2})$. This will result in the following weight distribution as a function of x:

$$\begin{aligned} A(w,x) &= \sum_{i,j} A_{i,j} w^{ix} w^{j(\frac{3-x}{2})} \\ &= \sum_{m} D_m w^m, \end{aligned} \tag{3}$$

where in the second equality D_m is defined as:

$$D_m \triangleq \sum_{i,j: ix+j(\frac{3-x}{2})=m} A_{i,j}. \tag{4}$$

It should be noted that in Eq. 3, m can take non-integer values for $x \neq 1$.

These weight distributions can be compared to each other to give a theoretical explanation for the performance of the code for different levels of UEP in practice. For large interleaver lengths, the WEF is replaced with the AWEF as in [3]. This is because it is practically very difficult to evaluate the WEF corresponding to a specific interleaver for large block lengths. The AWEF enumerates the codewords corresponding to every possible permutation in the same form as in Eq. 1. However, in this function the coefficient of each codeword $W^i Z^j$, is equal to the expected value of the number of codewords of this form. Thus, although the AWEF does not correspond to the specific interleaver which is employed, it can still be used to give an estimation of the code performance for different levels of UEP, on the average.

In order to show how the AWEF can be used in explaining the code behavior when UEP is employed, in the following, we consider two turbo-codes with different interleaver lengths. The codes consist of identical (5,7) RSC component codes and interleaver lengths 20 and 760. The multiplicities of the first few lowest weight codewords in the AWEF's of these two codes are shown in Table 1.

As can be seen from the table, for both codes, the expected value of the number of codewords in which the contribution of the systematic bits is higher ($j < 2i$, in a codeword the form $W^i Z^j$) is less than those in which the contribution of the parity bits is higher ($j > 2i$). This effect is much stronger for $N = 760$ as compared to $N = 20$. This agrees with the fact that a larger interleaver reduces the probability that a low weight input block of data results in low weight outputs in both component codes, simultaneously.

Next, we consider the effect of the UEP on the weight distribution of these codes. Fig. 1 shows the weight distribution of the first few codewords in AWEF's corresponding to three different levels of error protection for $N = 760$, and $N = 20$. In the diagrams corresponding to the same

[1] The exact bit energy is equal to $\frac{M+N}{N}E_b$, where M is the number of required terminating bits. However, this does not affect our discussion.

Code word	$N=$ 20	$N=$ 760	Code word	$N=$ 20	$N=$ 760
W^3Z^4	0.34	0.01	W^6Z^4	0.01	0.00
W^3Z^5	0.00	0.01	W^2Z^9	0.00	3.99
W^2Z^6	0.23	0.00	W^3Z^8	6.13	0.20
W^4Z^4	0.08	0.00	W^4Z^7	0.00	0.07
W^3Z^6	1.96	0.05	W^5Z^6	0.46	0.01
W^4Z^5	0.00	0.01	W^2Z^{10}	4.81	5.98
W^5Z^4	0.03	0.00	W^3Z^9	0.00	2.30
W^2Z^8	2.58	2.00	W^4Z^8	4.96	0.15
W^3Z^7	0.00	0.10	W^5Z^7	0.00	0.05
W^4Z^6	0.95	0.02	W^6Z^6	0.34	0.00
W^5Z^5	0.00	0.01			

Table 1: The multiplicities of the codewords of the form W^iZ^j, where $i+j \leq 12$, in the AWEF's corresponding to $N=20$ and $N=760$.

interleaver length, the total number of codewords is the same. For $N = 760$, it is easy to see that the distribution

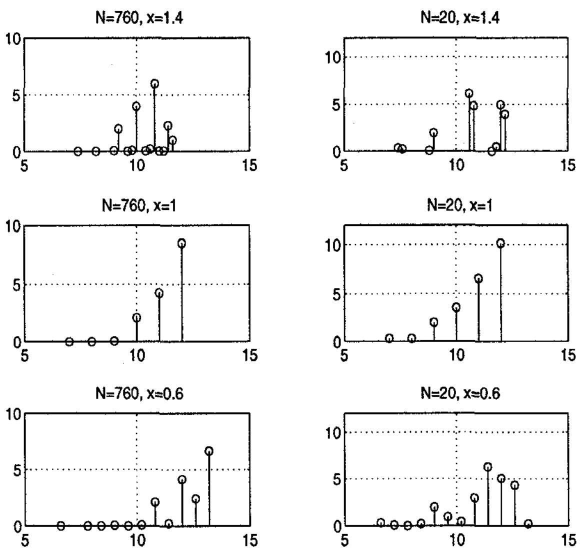

Figure 1: Weight distributions corresponding to different levels of error protection for $N = 760$ and $N = 20$.

of the codewords gets nearer to the origin for $x = 1.4$ and further from the origin for $x = 0.6$ (with respect to the distribution corresponding to EEP). For $N = 20$ however, in both levels of UEP the distribution gets diffusive in both directions.

We can also find the average performance bounds (union bounds), for different levels of UEP using the AWEF. This is obtained from following formula:

$$P_b(x) \leq \frac{1}{2} \sum_{m=m_{min}(x)}^{m(x)} \hat{D}_m \mathrm{erfc}\left(\sqrt{m\frac{rE_b}{N_0}}\right), \qquad (5)$$

where $P_b(x)$ is the probability of bit error and is a function of x, and $\hat{D}_m$ is defined as:

$$\hat{D}_m \triangleq \sum_{i,j:ix+j(\frac{3-x}{2})=m} \frac{i}{N} A_{i,j}. \qquad (6)$$

The factor $\frac{i}{N}$, where N is the interleaver length, is multiplied by each sentence in order to incorporate the measure of Bit Error Rate (BER). In Eq. 5, r is the code rate, here equal to 1/3, and $m_{min}(x)$ is the minimum value of m corresponding to x. In choosing $m(x)$, two different criteria were considered. The first criterion is to choose $m(x)$ such that the value $\sum_{m=m_{min}}^{m(x)} D_m$, is almost equal for different values of x, and the second criterion is to choose the same $m(x)$ for all values of x. The results for $N = 760$ with $E_b/N_0 = 2$ (dB) and for $N = 20$ with $E_b/N_0 = 3$ (dB) are shown in Table 2. In obtaining these results, $m(1)$ is chosen such that the bound is close enough to the corresponding simulation result. For $N = 760$, the WEF's

	$N=760, E_b/N_0=2$ (dB)		$N=20, E_b/N_0=3$ (dB)	
x	Criterion 1	Criterion 2	Criterion 1	Criterion 2
1.4	1.2×10^{-5}	1.7×10^{-5}	7.1×10^{-4}	7.2×10^{-4}
1.2	9.5×10^{-6}	9.5×10^{-6}	6.4×10^{-4}	6.1×10^{-4}
1	7.6×10^{-6}	7.6×10^{-6}	5.8×10^{-4}	5.8×10^{-4}
0.8	6.3×10^{-6}	3.4×10^{-6}	5.6×10^{-4}	5.8×10^{-4}
0.6	5.3×10^{-6}	2.9×10^{-6}	1.1×10^{-3}	1.1×10^{-3}
0.4	4.6×10^{-6}	1.1×10^{-6}	6.9×10^{-4}	6.7×10^{-4}
0.2	4.1×10^{-6}	1.3×10^{-6}	8.6×10^{-4}	9.4×10^{-4}
0	2.8×10^{-6}	1.5×10^{-6}	1.3×10^{-3}	1.8×10^{-3}

Table 2: Average performance bounds for $N = 760$ and $N = 20$.

corresponding to every branch in the hyper-trellis of the turbo code (see [3]) have been approximated by the WEF corresponding to the branch which connects the all-zero states in both component codes. Also, in order to limit the amount of computations in developing the AWEF, the transfer function corresponding to the component codes has to be truncated to codewords of Hamming weights less than a threshold. Since codewords of large weights are not incorporated in developing the average bound, this limitation does not affect this bound for $x = 1$. However, for $x \neq 1$, there might exist codewords which would have been encountered in the summation if they were included in the component transfer function. For this reason, the threshold should be appropriately selected to reduce this effect. Here, we've enumerated codewords of Hamming weights less than or equal to 75 in the component transfer function. Only the bounds corresponding to $x < 0.2$ will be affected by this limitation, thus in Table 2, the average bound for $x = 0$ is an under estimation. For $N = 20$, no approximation has been used and the component wise transfer function has been evaluated completely.

As can be seen from Table 2, for $N = 760$, performance improvement is achieved by reducing the protection of the systematic bits. The two criteria show slightly different behavior for $x < 0.4$. For $N = 20$, the best performance is achieved by EEP, or very near to EEP. In Section 3, the average of these two criteria is shown for comparison with the simulation results.

Before we proceed to the next section, it should be noted that although in this discussion the energies assigned to the parity check bits of both component codes are considered to be equal, the role of the two parity check bits is not necessarily the same in determining the performance of the code. However, when a uniform interleaver is considered these two bits will have the exact same role in the AWEF and the effect of unequally protecting them won't be shown by this function.

3 Simulation Results

Simulations are shown for turbo-codes of rate $r = 1/3$, employing two identical RSC codes. The channel is modeled as AWGN. Both encoders are terminated at the end of each block of input data. The component decoders employ the Bahl *et al* algorithm and the decoding procedure is similar to [2]. Figs. 2 and 3 show the logarithm of Bit Error Rate (BER) versus x, the factor already defined in Section 2. In Fig. 2, the simulation results correspond to a turbo-code with component codes (21,37), interleaver length $N = 16$, and SNR=2 (dB), and a code with component codes (5,7), interleaver length $N = 20$, and SNR=3 (dB). As can be seen, the performance for these two cases gets slightly better when the systematic bits are higher protected than the parity bits ($x > 1$). For $N = 20$, the theoretical bound is depicted in dashed line. This bound is the average of the bounds corresponding to the two criteria shown in Table 2.

Fig. 3 shows the results corresponding to a turbo-code consisting of component codes (5,7)and SNR=2 (dB), for $N = 380$, $N = 760$, and $N = 1000$. For larger interleaver lengths, better performance is achieved when the systematic bits are protected less than the parity check bits and this effect gets stronger as the block length increases. In this figure, the theoretical bound for $N = 760$ is depicted in dashed line. This bound is again the average of the bounds corresponding to the two criteria shown in Table 2. The theoretical bound agrees with the simulation results until $x = 0.2$.

For small values of x, there is a sudden rise in the BER in all three simulation curves. This degradation is due to the sub-optimum decoding scheme employed in turbo-decoding. The overall protection over the information given to the first component decoder (systematic and first parity check information) gets less than the case of EEP as x decreases, and consequently the extrinsic information passed to the second component decoder gets less reliable. For very small values of x, and consequently very unreliable extrinsic information, the iterative decoding procedure will no more converge to the ML solution, and this results in performance degradation.

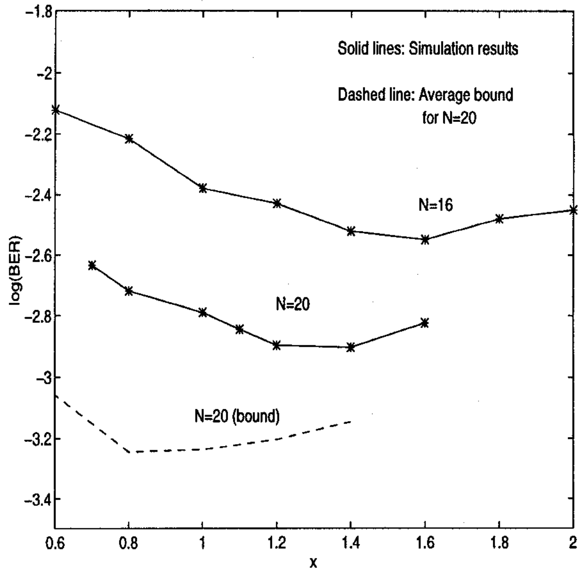

Figure 2: Results corresponding to turbo-code of component codes (21,37), $E_b/N_0 = 3$ (dB), and $N = 16$, and turbo-code of component codes (5,7), $E_b/N_0 = 3$ (dB), and $N = 20$.

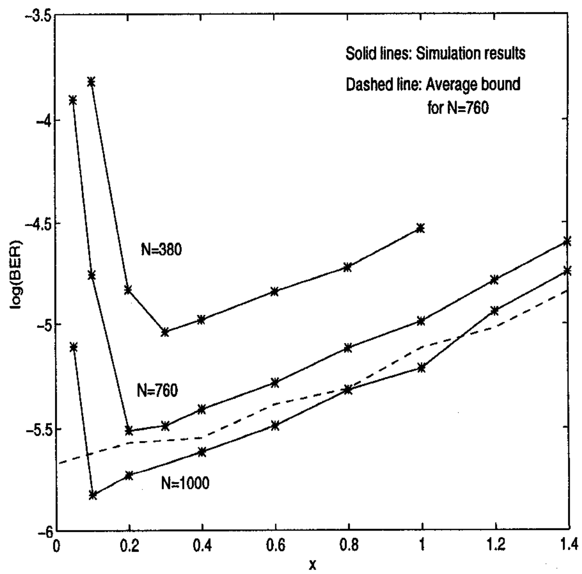

Figure 3: Results corresponding to turbo-code of component codes (5,7), $E_b/N_0 = 2$ (dB), with $N = 380$, $N = 760$, and $N = 1000$.

4 Conclusion

The role of the two groups of turbo-encoder output bits, namely the systematic and the parity check bits, in determining the code performance is studied in this article. It is shown that, when binary modulation is employed, the code performance can be improved by applying UEP to these two groups of information. For turbo-codes of very short interleaver lengths, the protection over the systematic information should be more than the parity information, for performance improvement. Studying the AWEF, suggests that as the interleaver length gets larger, the contribution of the parity check bits in the distances of the low weight codewords, increases. Thus, for large interleaver lengths, higher protection of these bits will improve the code performance. Some theoretical bounds have been obtained for different levels of UEP, using the AWEF, and it is shown that these bounds agree with the simulation results for a wide range of UEP. Improvements of about 0.5 in the $\log_{10}(\mathrm{BER})$ can be achieved by selecting the proper level of UEP,over EEP. This is approximately equivalent to 0.5 (dB) improvement in the Signal to Noise Ratio (SNR), in the range of the BER's of interest. This improvement is achieved at no extra cost in energy or rate.

References

[1] C. Berrou, A. Glavieux and P. Thitimajshima "Near Shannon limit error-correcting coding and decoding: Turbo-codes", Proc. of IEEE Int. Conf. on Comm. 1993 (ICC'93),Geneva, pp. 1064-1070, May 1993.

[2] P. Robertson "Illuminating the structure of code and decoder of parallel concatenated recursive systematic (turbo) codes", Proceedings of GLOBECOM'94, pp. 1298-1303, Dec. 1994.

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

[4] Atousa. H. S. Mohammadi, A. K. Khandani, "Effect of the labeling of the constellation points in a combined turbo-code and modulation," Proceedings of the 1996 Conference on Information Science and Systems (CISS'96), pp. 67-69, March 1996.