I.J. Wireless and Microwave Technologies, **2012, 2, 53-58**
Published Online April 2012 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijwmt.2012.02.08

Available online at http://www.mecs-press.net/ijwmt

Estimate BER Distributions of Turbo Codes

Shao Xia[a], Zhang Weidang[b]

[a]Department of Information Engineering , North China University of Water Conservancy and Electric Power, Zhengzhou, China, 450011
[b]School of Information Engineering , Zhengzhou University, Zhengzhou, China, 450001

Abstract

Based on the union bound, formulas to estimate the BER distribution of channel codes are derived. By using these formulas, the BER for every position in the information sequence can be estimated. Appling the formulas to Turbo codes, several examples were given, and the results are also compared with simulation results. The results show that the derived formulas can give out good estimations of the BER distributions for Turbo codes. Therefore this would be helpful for the BER analysis, especially the unequal error protection analysis of Turbo codes.

Index Terms: Channel coding; turbo codes; BER; unequal error protection

1. Introduction

For some cases in the data transmitting, such as when some bits of the input information are more important than others, unequal error protection is an efficient channel coding scheme. There are some examples to produce unequal error protections by using block codes and turbo codes[1]. But how to get the BER distribution of an unequal error protection code is not an easy work. Generally, for a given SNR and code, it can be got by simulation. However, the simulation maybe very time consuming, especially when the SNR changes the simulation shall be down again. In this paper, a theoretic method to estimate the BER distributions for turbo codes is tried to be derived without the disadvantages mentioned above

2. Union Bound And The Formulas Of The Ber Distribution

For an AWGN channel, the BER is bounded by the following formula [2][3],

$$P_b \le \sum_{i=1}^{2^k} \frac{w_i}{2k} erfc(\sqrt{d_i \frac{R_c E_b}{N_0}}) \tag{1}$$

* Corresponding author.
E-mail address: 67436622@163.com, iewdzhang@zzu.edu.cn

Where w_i and d_i are the information weight, and total Hamming weight, respectively, of the i-th codeword. k is the input length.

Due to the following formula,

$$Q(x) \le \frac{1}{2} e^{-x^2/2}$$

(1) can be simplified to

$$P_b \le \sum_{i=1}^{2^k} \frac{w_i}{2k} e^{-d_i \frac{R_c E_b}{N_0}} \tag{2}$$

Considering the weight enumerator function, (2) can be rewritten as

$$\begin{aligned} P_b &\le \frac{1}{2} \sum_{\omega=1}^{K} \frac{\omega}{K} e^{-\omega E_s/N_0} \sum_{z=1}^{N-K} A_{\omega,z} e^{-zE_s/N_0} \\ &= \frac{1}{2} \sum_{\omega=1}^{K} \frac{\omega}{K} \sum_{z=1}^{N-K} A_{\omega,z} e^{-(\omega+z)RcE_b/N_0} \end{aligned} \tag{3}$$

Where the ω is the weight of information and the z is the weight of parity check in a codeword.

The summations are extended over all the codewords of a systematic code C.

Let ci=(mi0,…,mik-1,pi0,…,pir-1) be a codeword of C, where mii is the ith bit of information and pij is the jth bit in the parity check bits of the codeword ci. Therefore the (3) can be written as

$$\begin{aligned} p_{jb} &= \sum_{c_i \in C} \frac{1}{2} m_{ij} e^{-d_i R_c E_b/N_0} \\ &= \sum_{i=1}^{2^k-1} m_{ij} \frac{1}{2} e^{-d_i R_c E_b/N_0} = \sum_{l} n_{jl} \frac{1}{2} e^{-d_l R_c E_b/N_0} \end{aligned} \tag{4}$$

Where $d_i = \sum_{j=0}^{k-1} m_{ij} + \sum_{s=0}^{r-1} p_{is}$ is the weight of ith codeword c_i.

And furthermore,

$$P_b \le \frac{1}{2}\sum_{c_i\in C}\sum_{j=0}^{k-1}\frac{m_{i\,j}}{k}e^{-d_iR_cE_b/N_0}$$

$$=\sum_{j=0}^{k-1}[\sum_{c_i\in C}\frac{1}{2}m_{i\,j}e^{-d_iR_cE_b/N_0}]/k \tag{5}$$

$$=\sum_{j=0}^{k-1}[p_{jb}]/k$$

Where njl is the number of the codewords with weight dl connected to ith bit of the information sequence. The quantity pb is a bound of C, therefore the pib is the bound for the i-th bit.

If we only consider the lowest weight codewords for every bit, we can get an estimate distribution of the bits' BERs by the following formula

$$p_b(j)\approx\frac{1}{2}n_{j\min}e^{-d_{j\min}R_cE_b/N_0} \tag{6}$$

where djmin is the lowest codeword(s)' weight connecting to the i-th bit and njmin is the number of such codewords.

Similarly, from (1) we can get

$$p_{jb}\le\sum_l n_{jl}\frac{1}{2}erfc(\sqrt{d_lR_cE_b/N_0}) \tag{7}$$

And

$$p_b(j)\approx\frac{1}{2}n_{j\min}erfc(\sqrt{d_{j\min}R_cE_b/N_0}) \tag{8}$$

From (6) and (8) we can see that if a bit connects to a codeword of lower weight, it will have a weaker error protection for error, and vice versa, if it only connects to the codewords of higher weights, it will have stronger protections.

For shorter lengths of codes, it is not difficult to get the lowest codeswords and their numbers for every bit. If the length is longer, it would be hard to calculate the required parameters. However, there have been some efficient algorithms to calculate the lowest weight of Turbo codes, such as [4]--[6]. They can be used here to get the required parameters only by slight modifications.

3. Simulation Results

The following examples are produced by turbo codes with generator matrix g = (1,1011/1101) and the interleaver size is 64. The simulations are carried on AWGN channel and the Eb/N0 is 4dB. The code rate is 1/3 or 1/2 according to different puncturing patterns. Here three type puncturing matrices are used. P0 is without any puncturing and the rate is 1/3. The puncturing patterns p4 and p7 are as the following

$$p4=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\cdot\quad p7=\begin{bmatrix}1 & \cdots & 1 & 0 & \cdots & 0\\ 1 & \cdots & 1 & 0 & \cdots & 0\end{bmatrix}.$$

Apparently p4 is a periodic puncturing pattern. For p7, the first 32 bits for both parity check sequences are kept and the last 32 bits for both parity check sequences are deleted. From the following Figs we can see that such a puncturing pattern can provide very strong error-protections for some bits. The number of BCJR iterations is 5. A block interleaver of 8×8 and a random interleaver are used. For comparison, the simulation results of BER distributions are also given.

To get the lowest weight codewords and their numbers for every bit, the full search is performed among the all codes with information weights less than or equal to 6.

For more accurate results, more terms in (7) can be considered. Here the second item is also considered and (8) becomes

$$p_b(j) \approx \frac{1}{2} n_{j\min} erfc(\sqrt{d_{j\min} R_c E_b / N_0}) \tag{9}$$

$$+\frac{1}{2} n_{j\,second\min} erfc(\sqrt{d_{j\,second\min} R_c E_b / N_0})$$

Where $n_{j\,second\min}$ and $d_{j\,second\min}$ are the number and the weight of second lowest weight codeword connecting to the j-th bit.

In the following figs, curve "----"denotes the results produced by simulations, "--*" produced by (8), and "-o" by (9).

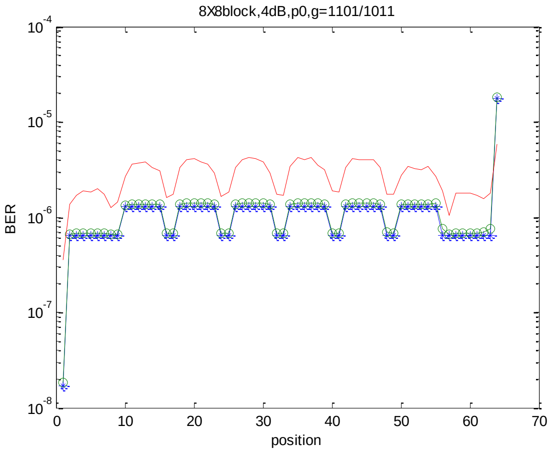

Fig 1 The curves for 8X8 block interleaver using p0 puncturing matrix under SNR=4dB.

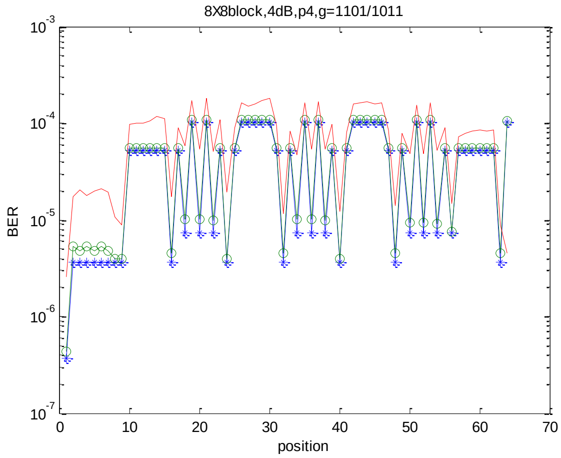

Fig 2 The curves for 8X8 block interleaver using p4 puncturing matrix under SNR=4dB.

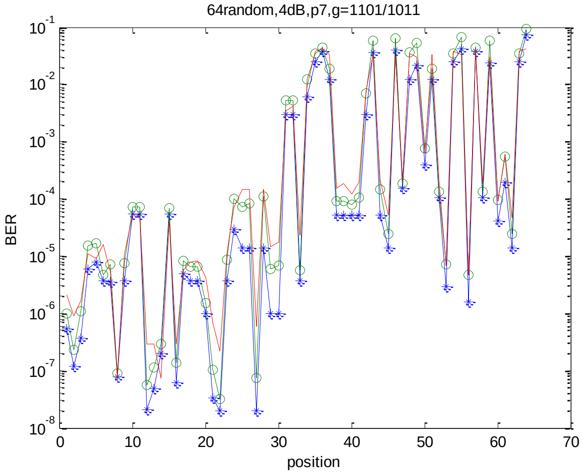

Fig 3 The curves for random interleaver using p7 puncturing matrix under SNR=4dB.

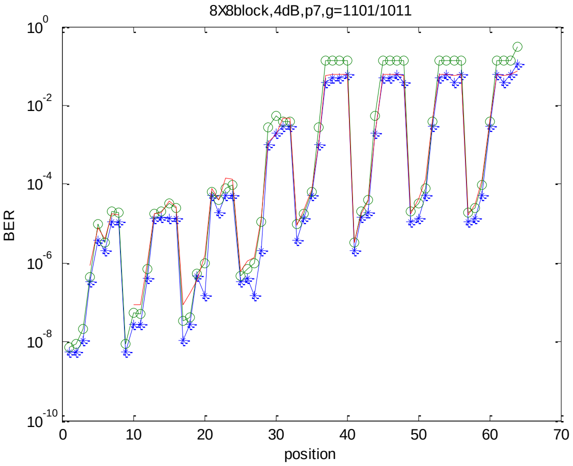

Fig 4 The curves for 8X8 block interleaver using p7 puncturing matrix under SNR=4dB.

From the above Figures we can see that the shapes of BER distribution curves produced by the formulas have a good match with those obtained by simulations, especially when the curves' vary over a wide range, such as in Fig 3. And the curves produced by (9) differ only slightly compared to those by (8). These also show that the bit's error rate of a particular bit is determined mainly by the lowest weight codeword connected to it.

4. Conclusion

Based on the union bound, formulas to estimate the BER distributions for turbo codes were explored, which can give a detailed description of the BER on every information bit. The formulas only need the weights of lowest (and second) weight codewords and their numbers for every position. So it can be used for various types of the constructions for turbo codes, no matter what kind of component encoders, interleavers, and puncturing schemes are used. Of course these formulas can also be applied to other kinds of channel codes with the required parameters.

Acknowledgements

The nature science project of Henan province education commission of China (2010B510020)

References

[1] M. Aydinlik and M. Salehi, “Turbo coded modulation for Unequal Error Protection,” *IEEE Trans. Commun.*, vol. 56, pp.555-564, April 2008.

[2] Perez, L.C., Seghers, J., and Costello, D.J. Jr.: ‘A distance spectrum interpretation of turbo codes’, *IEEE Trans. Inf. Thoery.*, 1996, IT-42, pp. 1698–1709

[3] Roberto Garello, *Member, IEEE,* Paola Pierleoni, and Sergio Benedetto, *Fellow, IEEE,* Computing the Free Distance of Turbo Codes and Serially Concatenated Codes with Interleavers: Algorithms and Applications, *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, VOL. 19, NO. 5, MAY 2001, pp800-812

[4] Oscar Y. Takeshita, *Member, IEEE,* Marc P. C. Fossorier, *Member, IEEE,* and Daniel J. Costello, Jr., *Fellow, IEEE,* A New Technique for Computing the Weight Spectrum of Turbo-Codes, *IEEE COMMUNICATIONS LETTERS*, VOL. 3, NO. 8, AUGUST 1999.pp251-253

[5] C. Berrou, S. Vaton, M. J ́ez ́equel, and C. Douillard, “Computing the minimum distance of linear codes by the error impulse method,” in *Proc.2002 GLOBECOM*, Nov. 2002, pp. 1017-1020.

[6] R. Garello and A. Vila Casado, “The all-zero iterative decoding algorithm for turbo code minimum distance computation,” in *Proc. ICC 2004,* Jun. 2004, pp. 361-364.