International Journal of Future Generation Communication and Networking
Vol. 8, No. 4 (2015), pp. 265-274
http://dx.doi.org/10.14257/ijfgcn.2015.8.4.26

Improve the BER Performance of Turbo Codes with Large Size Based on BER Distributions

Shao Xia[1] and Zhang Weidang[2]

[1] *Department of Information Engineering, North China University of Water Resources and Electric Power, Zhengzhou 450011, China.*
[2]*School of Information Engineering, Zhengzhou University, Zhengzhou, 450001, China, Corresponding author*
shaoxia@ncwu.edu.cn, zhangweidang@zzu.edu.cn

Abstract

A method for bitwise energy optimization of turbo codes with large size is developed. Unlike earlier methods, which relied on searching for minimum-weight codewords, the proposed method finds the required parameters using two bit error rate (BER) distributions obtained via simulation. Therefore, this method can be used not only for the codes with small codeword length, but also for the codes with large codeword length. Simulation results show BER improvements for turbo codes. Though the method is applied to turbo codes in this letter for concreteness, it can also be used for other kinds of error control codes because it does not rely on the code's structure.

Keywords: *turbo codes, bit error rate (BER), energy optimization.*

.

1. Introduction

Since their introduction by Berrou er al.[1], turbo codes have received tremendous attention. Even in the recent years, there are still many researchers studied on it. In reference[2], the authors discussed recent advances in the design and theory of turbo codes and their relationship to LDPC codes. Several new interleaver designs for turbo codes were presented which illustrated the important role that the interleaver played in these codes. The relationship between turbo codes and LDPC codes was explored. In reference [3], The thesis studied on the performance of turbo coded sampling based Noncoherent Binary Frequency Shift Keying (NCBFSK) and Noncoherent Quaternary Frequency Shift Keying (NCQFSK) under additive white Gaussian noise (AWGN) and given the detailed metric calculation. Adding security to turbo code is an attractive idea since it could reduce the processing cost of providing secure coded data and enjoys the advantages of high-speed encryption and decryption with high security, smaller encoder and decoder size and greater efficiency. In reference [4], secure channel coding schemes based on turbo codes were suggested for time reversal ultra wideband (TR-UWB) systems. In reference [5], the authors studied the application of turbo codes in advanced orbiting systems(AOS). Experiments showed that turbo codes were better than convolutional code especially under the condition of low Signal-to-Noise Ratio(*SNR*). In reference [6], the authors designed a turbo-coded noncoherent space-time modulation scheme based on the codeword-interleaving strategy. When comparing to the conventional turbo-coded scheme, the proposed scheme achieved a significant improvement in BER performance. And so on.

Although there are many literatures considered about various aspects of turbo codes, how to improve the bit error rate (BER) performance of turbo codes is still the most important task. There are many methods toward this destination. One of them is to

ISSN: 2233-7857 IJFGCN

reallocate the energy of the bit in the bit stream of the codeword. These schemes have previously been proposed in the literature [7-15]. In reference [7], the author assigned less and less power to the parity bits as the noise level increases to avoid the traditional negative "coding gain" associated with all error correcting codes at high noise levels. Reference [8] showed that the fraction of the total power that should be allocated for a systematic bit is usually lower than that of the parity bit. But the amount of improvement depends on the choice of the component codes, interleaver length and the signal to noise ratio. Reference [9] also pointed out that if different energies were assigned to two outputs of a turbo encoder, the information bit and parity bit, then the performance will be changed according to the ratio of the information bit energy to the parity bit energy. The optimum point of the ratio may not be 1. As the rate of the turbo code is changed, the optimum point will also be changed. In reference [10], it concluded that for turbo codes with short frames operating in very low signal-to-noise environments, more energy should be assigned to the systematic bits so that the performance was improved. At higher signal-to-noise ratios, allocating less energy to the systematic bits improved the performance. Reference[11] studied the effect of asymmetric energy allocations to the output bits of turbo codes. It showed that the error floor improves as more energy is given to the non-systematic bits. However, due to the degradation in the convergence threshold of the code, tradeoff between the error floor and the convergence threshold appeared. Reference [12] studied theoretically and empirically channels coding for nonuniform i.i.d. sequences using turbo codes with unequal energy allocation. It was shown that both systematic codes and non-systematic codes with unequal energy allocation improve on equal energy allocation schemes. Reference [13] introduced a method to reduce the error floor in parallel concatenated codes. It also pointed out that simple approached based on modifying just the energy of the systematic and coded bits seem very attractive. From above we can see that nearly all of them allocate the energies between the systematic bits and parity bits, but the merits of different strategies are sometimes not very clear, with different authors arriving to contradicting conclusions [11]. This is because that there is no theoretical base for the energy allocation between the information bits and parity check bits. The fraction of the total energy depends on the choice of the component codes, interleaver length, puncturing pattern and the signal to noise ratio.

In reference [14], the authors allocated the bits' energies among the codewords that have different weights instead of between the systematic and parity bits. In this scheme more energy is assigned to the codewords that have minimum (and second minimum) weight and the simulation results showed that the "error floor" of turbo codes was improved with no practical degradation in the waterfall region. In reference [15], based on Union Bound and by mathematical optimization and assisted with a BER distribution, the energy for every bit was optimized and the average BER achieves the minimum value. However, these methods were based on the minimum weight distributions. Searching for minimum-weight codewords is very time consuming even if the code length is no more than a thousand bits. Therefore, they cannot be used for the codes with large size.

In this paper, a method based on [15] is proposed to optimize the bitwise energy allocation. This method is based on the low-weight distribution of the code, which is estimated using two BER distributions obtained via simulation, rather than searching for minimum-weight codewords. Hence, this method is also applicable to codes with large codeword length. Case studies indicate that BER performance can be improved not only at a high *SNR*, but also at a low and moderate *SNR*. The remainder of this paper is organized as follows. In Section 2, a formula to estimate the BER distributions based on Union Bound is described. In Section 3, a formula to optimize the bit energy based on the BER distribution given in section 1 is derived. Numerical explanations are also given in this section. In Section 4, an adjustable parameter to modify the energy distribution is introduced. Section 5 gives several examples to show the efficiency of the scheme. Finally, concluding remarks are given in Section 6.

2. Bitwise Energy Allocation

Based on the Union bound, an estimate of the BER for each bit in the bit stream of the codeword has been derived as [17]

$$p_b(j) \approx \frac{1}{2} n_{\min}(j) e^{-d_{\min}(j) R_c E_b / N_0} \quad , \tag{1}$$

where R_c is the code rate, E_b is the bit energy and N_0 is the noise power spectral density. $d_{min}(j)$ is the minimum Hamming weight among the codeword(s) connecting to the j-th bit (i.e., those codewords that have a "1" in the j-th bit position) and $n_{min}(j)$ is the multiplicity. j=0,1,2,…,N-1 is the bit index within the codeword and N is the codeword length.

Equation (1) shows that, generally, due to the variability of $d_{min}(j)$ and $n_{min}(j)$ with j, the distribution of the BER is not uniform across the codeword. For example, if $d_{min}(j)$ is small for a specific bit index j, the BER for this bit, $p_b(j)$, will be high. The average BER of the code at high SNR is dominated by such bits that connect to the codewords with low weights. Therefore, if we allocate more energy to these bits, and consequently less energy to the bits with higher weights, so that the total energy of the code remains constant, the average BER will decrease. Inspired by this idea, we try to find an optimum bit energy allocation so that the average BER is minimized.

3. Optimizing the Energy Allocation

Replacing the constant parameter E_b with $E_b(j)$, which will denote the energy allocated for the j-th bit, and replacing $p_b(j)$ with $p_{ob}(j)$, which is the new BER for the j-th bit corresponding to $E_b(j)$, then (1) can be expressed as

$$p_{ob}(j) \approx \frac{1}{2} n_{\min}(j) e^{-d_{\min}(j) R_c E_b(j) / N_0} \tag{2}$$

The average of $p_{ob}(j)$ is

$$\bar{p}_{ob} = \frac{1}{N} \sum_{j=0}^{N-1} p_{ob}(j) \tag{3}$$

$E_b(j)$ must satisfies the constraint of energy conservation

$$\sum_{j=0}^{N-1} E_b(j) = N E_b \tag{4}$$

Using the Lagrange multiplier method and employing λ as the multiplier, $E_b(j)$ can be computed as[15]

$$\begin{aligned} E_b(j) &= \frac{N_0}{R_c d_{\min}(j)} \ln \frac{d_{\min}(j) n_{\min}(j) R_c}{2 N_0 \lambda} \\ &= \frac{N_0}{R_c d_{\min}(j)} \ln \frac{d_{\min}(j) n_{\min}(j) R_c}{2 N_0} - \frac{N_0}{R_c d_{\min}(j)} \ln\lambda \end{aligned} \tag{5}$$

where

$$\ln\lambda = \frac{\sum_{j=0}^{N\text{-}1} \left(\frac{N_0}{R_c d_{\min}(j)} \ln \frac{d_{\min}(j) n_{\min}(j) R_c}{2 N_0} \right) - N E_b}{\sum_{j=0}^{N\text{-}1} \frac{N_0}{R_c d_{\min}(j)}} \tag{6}$$

If the codeword length is small, there are some efficient algorithms, such as the methods presented in [19,20,21], to calculate $d_{min}(j)$ and $n_{min}(j)$. If the codeword length is large, the calculation of $d_{min}(j)$ and $n_{min}(j)$ is very time consuming. Next, we will give a novel method to overcome such difficulty. Let $p_{b1}(j)$, $p_{b2}(j)$ be two BER distributions produced by simulations under two different SNRs, denoted by $SNR1$ and $SNR2$, respectively. Then based on equation (1), $d_{min}(j)$ and $n_{min}(j)$ can be obtained from the following system of (approximate) equations

$$\begin{cases} p_{b1}(j) \approx \frac{1}{2} n_{\min}(j) e^{-d_{\min}(j) R_c SNR1} \\ p_{b2}(j) \approx \frac{1}{2} n_{\min}(j) e^{-d_{\min}(j) R_c SNR2} \end{cases} \tag{7}$$

The solution is

$$\begin{cases} d_{\min}(j) = \dfrac{\ln(p_{b2}(j) / p_{b1}(j))}{R_c(SNR1 - SNR2)} \\ n_{\min}(j) = 2 p_{b1}(j) \left(\dfrac{p_{b1}(j)}{p_{b2}(j)}\right)^{\frac{SNR1}{SNR2 - SNR1}} \end{cases} \tag{8}$$

Thus, (8) gives the estimates of $d_{min}(j)$ and $n_{min}(j)$. However, they are affected by the selection of $SNR1$ and $SNR2$.We compared several pairs of $SNR1$ and $SNR2$ and found that if one SNR, say as $SNR1$, takes moderate value and another, say as $SNR2$, takes a higher value, the estimates are more accurate. In this context, moderate SNR value means SNR that produces BER in the range 10^{-2} ~ 10^{-3}, while high SNR means SNR that leads to BER in the range 10^{-3} ~ 10^{-5}.

The following is a numerical illustration of this scheme. A turbo code with generating matrix g=(1,10001/10011) is used. The code rate is 1/3 and a random interleaver with size of 24 is employed. So the code length is N=(24+4)×3=84. The decoding algorithm is BCJR with five iterations. Binary antipodal signalling was used with AWGN channel model.

First, two BER distributions should be produced by simulation at two different SNRs. Here we choose $SNR1$=3dB and $SNR2$=5dB. Then by (5) we can get the optimized bit energy distribution with $d_{min}(j)$ and $n_{min}(j)$ produced by (8). Finally, we assign to each bit j its optimized energy $E_b(j)$. The simulation results are presented in Fig.1, where curve 1 shows bitwise BER before bit energy optimization and curve 2 shows bitwise BER after bit energy optimization. The bitwise BER before energy optimization varies considerably across the codeword and there are seven peaks, which matches the weight of the minimum-weight codeword. (The code contains a minimum-weight codeword with weight 7 whose non-zero bits are located at bit positions 13, 14, 22, 25, 26, 51, 75, exactly where the BER peaks are). In curve 2, the BER peaks are pulled down due to more energy being allocated to these bits. Therefore, this changes the energy of the codewords. For example, if we assume E_b=1, the energy of the codeword with minimum weight is increased from 7 to 12 at 5dB and the BER is decreased from 4.95×10^{-6} to 1.80×10^{-6}.

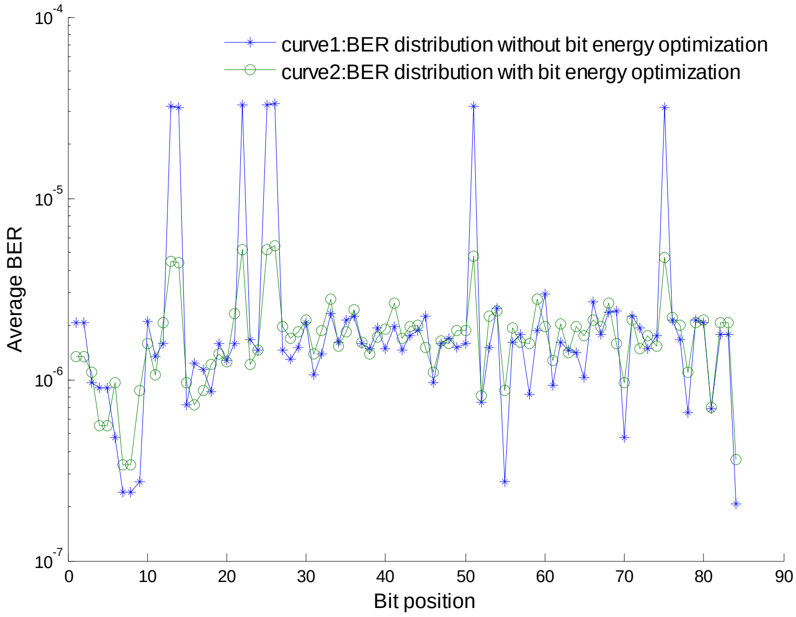


Figure 1. BER Distributions before and after Energy Optimization for a Turbo Code at SNR=5dB

Fig. 2 shows the comparisons of average BERs between theoretical results and simulation results for the code presented above. Theoretical curve 1 is calculated by (1) and the theoretical curve 2 is calculated by (2) with $E_b(j)$ given by (5). The simulation curve 1 is the simulation result without bit energy optimization ($E_b(j) = E_b$ = const.) and the simulation curve 2 is the simulation result after bit energy optimization, with $E_b(j)$ given by (5). All the parameters $d_{min}(j)$ and $n_{min}(j)$ required are estimated by (8).

From Fig. 2 we can see that the average BERs are improved at high *SNR* both theoretically and experimentally. However, at low *SNR*, even moderate *SNR*, degradation appears. This is because that $d_{min}(j)$ and $n_{min}(j)$ in (8) are only estimates derived from two specific BER distributions produced by simulation at two different *SNR*s, so they may deviate from the true values. Moreover, formula (1) is also an estimate of the BER distribution. Hence, the resulting bit energy allocation might not be optimal. To overcome this drawback, next we give a modification to the bitwise energy allocation those results in improved performance.

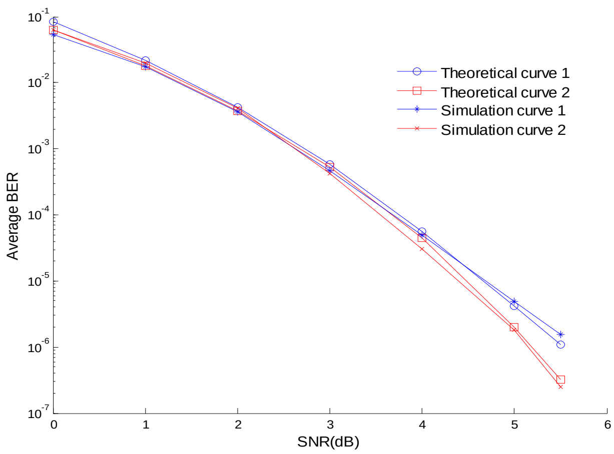


Figure 2. Comparisons of the Average BERs between Theoretical Results and Simulation Results

4. Modification

For simplicity, we assume E_b=1. Then let $E_b'(j) \geq 0$ be the modified optimized energy allocation, expressed as

$$E_b'(j) = E_b(j) + \rho(E_b(j)\text{-}1) \tag{9}$$

where ρ is a parameter that needs to be adjusted. Obviously,

$$\sum_{j=0}^{N-1} E_b'(j) = \sum_{j=0}^{N-1} (E_b(j) + \rho(E_b(j)\text{-}1)) = N \tag{10}$$

Equation (10) shows that $E_b'(j)$ still satisfies the energy conservation. Because we have assumed E_b=1, apparently, the maximum value of $E_b(j)$ is no less than 1 and the minimum value of $E_b(j)$ is no greater than 1. Then from (9) we can find the valid range for ρ as

$$\frac{\max E_b(j)}{1-\max E_b(j)} \leq \rho \leq \frac{\min E_b(j)}{1-\min E_b(j)} \tag{11}$$

If ρ =-1, $E_b'(j)=1$. In this case, every bit has the same energy consumption. However, with the increase of ρ , more energy is consumed by the bits with low weights. When ρ = 0, $E_b'(j) = E_b(j)$, the improved energy allocation matches that obtained in Section 3. Otherwise, if ρ <-1, the energy assigned to the low-weight codewords will be decreased and consequently, codewords with high weights will obtain more energy. Therefore, through modifying the parameter ρ , the energy allocation can be adjusted to ensure it is suitable for different channel states. Suitable value for ρ can be found through grid search.

Fig. 3 gives an example of comparison among the equal energy allocation, optimized energy allocation and modified optimized energy allocation for code 0, whose parameters are listed in Table 1. From Fig. 3 we can see that comparing with the curve, noted as "-*-", produced by equal energy allocation, the curve of average BER optimized by (5), noted as "-□-", is improved at high SNR region significantly. However, at low SNR, even moderate *SNR*, degradation appears. The curve, noted by "-o-", is obtained by modified optimized bit energy allocation (9) with best values of ρ . Clearly, BER performance is improved at low and moderate *SNR* regions, and it has the best performance over a wide range of *SNR*.

The best values of ρ for the code 0 and other four codes that will be used in next section are listed in Table 2. In each row, the values in the first line show *SNR* in dB and the values in second line show the best ρ. These values were obtained by grid search within the valid range of ρ, starting with step size of 3, down to the finest step size of 0.25.

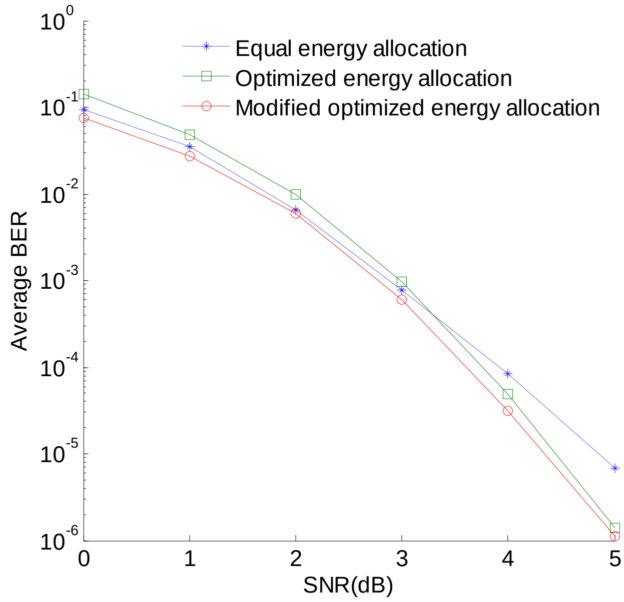


Figure 3. Comparison Among the Curves Produced by Equal Energy Allocation, Optimized Energy Allocation and Modified Optimized Energy Allocation for Code 0

5. Simulation Results

There are other four different turbo codes employed in this section for case studies. Their parameters are listed in Table 1.They have different interleaver types and sizes. The interleaver sizes are from 1024 to 16384. Two different puncturing patterns are used. The decoding algorithm is BCJR with 5 iterations and the two encoder components are both terminated. At high SNRs, the simulation is run until at least 300 errors are obtained.

The simulation results are showed in Fig.4. In the figure, there are two curves for every code. They are the curves for conventional (unoptimized) case and for bit energy allocated according to (9). For conciseness, the curves produced only by optimized energy allocation do not appear in the figure. The improvements are apparent even for the turbo codes with large codeword length cross the full range of SNR.

The best values for ρ in (9) for the four codes are listed in Table 2. In each row, the values in the first line show *SNR* in dB and the values in second line show the best ρ . These values were obtained by grid search, starting with step size of 3, down to the finest step size of 0.25.

Table 1. Five Different Turbo Codes used in This Section as Examples

	Generator Function	Punturing Pattern	Interleaver Type,zise	*SNR*1,*SNR*2 (dB)
Code0	(1,1101/1011)	(10;01)	Block, 8×8	3, 5
Code1	(1,1101/1011)	(11;11)	3gpp,1024	0.5, 1
Code2	(1,1101/1011)	(11;11)	3gpp,4096	0.3, 0.5
Code3	(1,10001/10011)	(10;01)	Random,16384	0.5, 1
Code4	(1,10001/10011))	(11;11)	Random,16384	0.2, 0.4

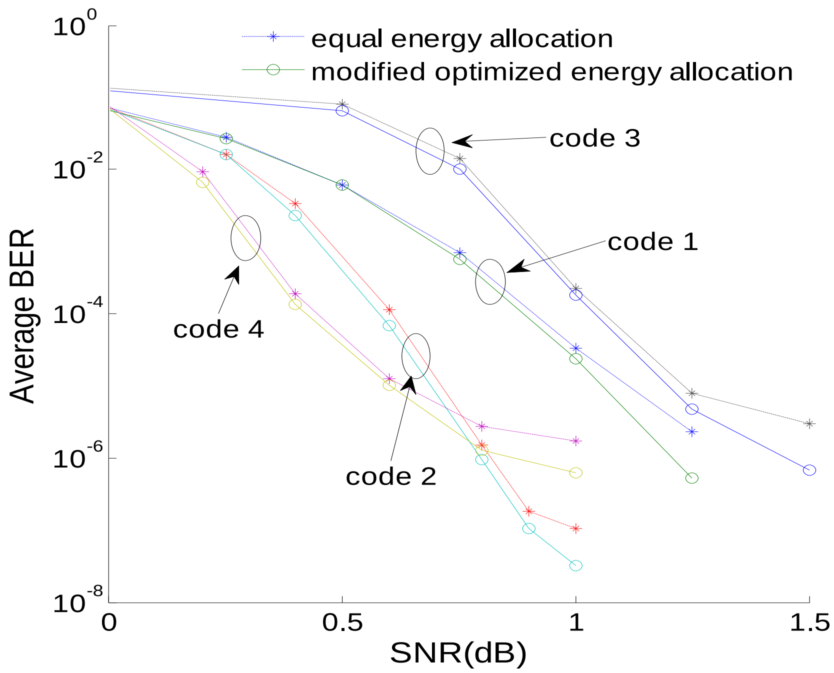


Figure 4. The Simulation BER Curves before and after Optimizing for Code 1, Code 2, Code 3 and Code 4

Table 2. The Best Values of *ρ*(bottom) for different *SNR*(in dB) (top) for the Five Codes

Code0	0 -2.5	1 -2.5	2 -2	3 -0.5	4 1.5	5 2	
Code1	0 -6	0.25 -3	0.5 -2.5	0.75 -1.5	1 0.75	1.25 2.75	
Code2	0 -5	0.25 -1.5	0.4 -2	0.6 -0.75	0.8 -0.5	0.9 1.25	1 1.25
Code3	0 -3	0.5 -1	0.75 0.	1 0.5	1.25 1	1.5 1.25	
Code4	0 -4	0.2 -3.75	0.4 -1.75	0.6 -0.5	0.8 0	1 0.25	

6. Conclusion

A new method to optimize the bitwise energy allocation within a codeword is presented in this paper. This method employs two different BER distributions and an adjustable parameter, instead of searching the minimum weight codewords. Getting the BER distributions by simulation is much easier than searching the minimum weight codewords, therefore, it can be applied to the codes with large sizes. Simulation results show that this method is efficient and effective for various types and large size of turbo codes. The BER improvements can be extended to a wide range of *SNR* by adjusting a parameter. Though the method was applied to turbo codes in this paper for concreteness, it can be used for other kinds of error control codes because it does not depend in the code's structure. Therefore, for any channel code, no matter what type it is and how large the size is, if it doesn't have uniform distribution of codeword weight, the proposed scheme can be used to improve its BER performance.

References

[1] C. Berrou, A Glavieux and A Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes", In Proc. 1993 IEEE Int. Conf. Communications, **(1993)**; Geneva, Switzerland.

[2] B. Vucetic, Y. Li, L. C. Perez and F. Jiang, "Recent advances in turbo code design and theory", Proceedings of the IEEE, **(2007)**.

[3] P. R. Pooja, "Sampling Based Turbo and Turbo Concatenated Coded. Noncoherent Modulation Schemes", M.S Dissertation, University of Toledo, **(2010)**.

[4] A-M. Dariush and T. V. Vahid, "Enhanced Secure Error Correction Code Schemes in Time Reversal UWB Systems", Wireless Personal Communications, vol.64, no.2, **(2012)**, pp 403-423.

[5] W. Chuanyu, W. Shanshan, T. Ye and M. Yufeng, "Research and simulation on the performance of turbo code and convolutional code in advanced orbiting systems", Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications (MAPE), 2013 IEEE 5th International Symposium on. Chengdu, **(2013)**.

[6] C. Yen-Ming, U. Yeong-Luh, S. Hau-Jung and H. Dung-Rung, "Noncoherent coded space-time modulation for a large number of transmit antennas", Communications in China (ICCC), 2012 1st IEEE International Conference on, **(2012)**; Beijing.

[7] J. Hokfelt and T. Maseng, "Optimizing the energy of different bitstreams of turbo codes", Turbo Coding Seminar Proc., **(1998)**; Lund, Sweden.

[8] T. M. Duman and M. Salehi, "On optimal power allocation for turbo codes", Ulm, Germany, **(1997)**, pp.104.

[9] Y. Choi and P. Lee, "Analysis of turbo codes with asymmetric modulation", Electron. Lett., vol.35, no.1, **(1999)**, pp.35–36.

[10] M. M. Salah, R. A. Raines, M. A. Temple and T. G. Bailey, "Energy allocation strategies for Turbo Codes with short frames", Proceedings International Conference on Information Technology: Coding and computing. ITCC **(2000)**, Las Vegas, NV, USA.

[11] F.e Cabarcas and J.r Garcia-Frias, "Asymmetric energy allocation strategies to improve Turbo Codes performance", Vehicular Technology Conference, 2001, VTC 2001 Fall, IEEE VTS 54th, **(2001)**, pp.1839 – 1842.

[12] G. I. Shamir, R. D. Souza and J. Garcia-Frias, "Unequal energy allocation with Turbo Codes for nonuniform sources", Turbo-Coding, Munich, **(2006)**, pp.1-6.

[13] J. Garcia-Frias and F. Cabarcas, "Reducing the error floor in turbo codes by using non-binary constituent encoders", Proc. VTC'00, **(2000)**; Boston, MA, USA.

[14] W. Zhang and X. Wang, "Optimal energy allocations for turbo codes based on distributions of low weight codewords", Electronics Letters, vol.40, no.19, **(2004)**, pp.1205 – 1206.

[15] S. Xia, P. Li and Z. Weidanng, "Improving the BER Performance by Optimizing Energy Allocation Based of Union bound", accepted by IET communications, doi: 10.1049/iet-com.2014.0160.

[16] S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes", IEEE Trans. Inform. Theory, vol.42, **(1996)**, pp.409–428.

[17] S. Xia and Z. Weidang, "Estimate the BER Distributions of Turbo Codes", I.J. Wireless and Microwave Technologies, vol.2, **(2012)**, pp.53-58.

[18] Z. Weidang , S. Xia, "Unequal Error Protection of JPEG2000 Images Using Short Block Length Turbo Codes", Communications Letters, IEEE, vol.15, no.6, **(2011)**, pp.659 – 661.

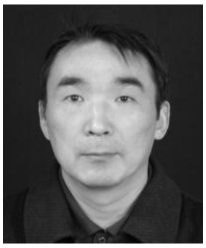
[19] R. Garello, P. Pierleoni and S. Benedetto, "Computing the Free Distance of Turbo Codes and Serially Concatenated Codes with Interleavers: Algorithms and Applications", IEEE Journal on Selected Areas in. Commun., vol.19, no.5, **(2001)**, pp.800-812.

[20] S. Scalise, Y-J Ko Bac and H. Ernst, "A Fast Algorithm to Estimate the Distance Spectrum of Turbo Codes", ICT`03, pp.90-95.

[21] S. Crozier, P. Guinand and A. Hunt, "Estimating the minimum distance of turbo codes using double and triple impulse methods", IEEE Commun. Letters, **(2005)**, pp.631–633.

Authors

Shao Xia, she received her M.S. degree and B. S. degree from Zhengzhou University in 2007 and 1992 separately. Her research focuses on key techniques for telecommunication theory and engineering.



Zhang Weidang , he received his Ph D from Xidian University in 2005. He academically has visited Simon Fraser University from 2008 to 2009. Currently he is with the School of Information Engineering, Zhengzhou University where he is a professor and his research interests include information theory and coding theory.