International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.10 (2016), pp. 421-430
http://dx.doi.org/10.14257/ijmue.2016.11.10.39

Designing the Puncturing Pattern for Turbo Codes with Strong Unequal Error Protection

Shao Xia[1] and Zhang Weidang[2]

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

Abstract

In multimedia transition systems, data usually is divided into levels according to their different importance. For these systems, unequal error protection (UEP) techniques are the powerful methods to guarantee lower BER for the more important classes. Puncturing patterns are very important for turbo codes to produce different code rates. Also, different puncturing patterns can give different BER distributions so that UEP can be obtained. In this paper we will study the affection of puncturing pattern to the BER distribution and present a new scheme of designing the puncturing patterns for turbo codes so that the strong unequal error protection is obtained. In this scheme the puncturing is only carried on the two parity check sequences and an idea of "gradually sparsing" is employed. Simulation results show that, compared with other puncturing schemes, this scheme can provide strong error protection for some bits without apparent degradation for other bits.

Keywords*: turbo codes, puncturing pattern, multimedia, unequal error protection (UEP)*

1. Introduction

In many communications systems, especially the multimedia transition systems, data usually is divided into levels according to their different importance. For these systems, unequal error protection (UEP) techniques are the powerful methods to guarantee lower BER for the more important classes. In particular, if the precise characteristics of the channel are not known, UEP can still be used to recover the more important classes even in very poor receiving conditions. Through hard researching and study by many researchers, some efficient schemes to achieve UEP and various applications of UEP have been proposed. Such as in references [1-4], the UEP are achieved by fountain codes. The UEP can also be achieved by using low-density parity-check codes (LDPC) [5-6], Because turbo codes have the flexible encoding structure and strong error correcting ability, turbo codes are also the important tools to realize UEP. In [7], the UEP is achieved by varying the code rate assigned to different portions of the scalable source bitstream. Varying the code rate is achieved by using rate compatible punctured convolutional (RCPC) codes. In [8], the authors propose a way of achieving different code rates within the same data block by puncturing some of the parity bits generated by one of the component convolutional encoders. In [9], UEP for Turbo Codes has been achieved by designing the interleaver in a special way. The proposed interleaver tends to produce lower error probability at the beginning of the data block than at the end of the block. In [10], the authors proposed a combination of Turbo Coding and modulation for UEP. In their scheme, data bits are first divided into classes of different importance. Each

ISSN: 1975-0080 IJMUE

class is then interleaved within itself and punctured by its own puncturing pattern, effectively creating a different Turbo Code for each class. In [11], the authors noticed that UEP is an inherent property of Turbo Codes, because the Bit Error Rate (BER) at various positions within the data block is not uniform. Then by simple reordering of the data bits that enter the Turbo encoder, the desired UEP pattern that matches the sensitivity of the individual data bits is achieved. More recently, the authors in [12] derived bounds on the performance of unequal error protecting turbo codes. These bounds served as an important tool in predicting the performance of these codes. In [13], the authors proposed a new unequal error protection (UEP) scheme for progressive image transmission by using rate-compatible punctured Turbo codes (RCPT) and cyclic redundancy check (CRC) codes only. There are still many other references considered about the UEP in Turbo codes, such as [14-19]. The subjects are about the interleaver design, image transmission, decoding algorithms, and so on.

Duo to more easily being implemented than other UEP schemes and high efficiency, using different puncturing patterns to obtain UEP has been studied in some references, such as [11,13,18,20]. In [11], the authors noticed that an uneven puncturing pattern (where the lowest weights for every position differ widely) will result in strong UEP. And two types of puncturing patterns were presented. But these puncturing patterns puncture not only on the parity check sequence, but also on the information sequence. In [20], a puncturing pattern noted by p7 gives very strong error protection on some bits but with serious degradation for some other bits. Inspired by the ideas proposed in these references, in this paper, we will focus on a special design of puncturing pattern of turbo codes to achieve strong unequal error protection based on a new idea called "gradually sparsing" that will be introduced in next section.

The rest of the paper is organized as follows. In Section 2, we explain the idea of "gradually sparsing". Two main steps of design scheme, basic puncturing pattern design and modified puncturing pattern design, are also introduced. An example is given to show the design procedures. In Section 3, some simulation results are showed and comparisons are made to show the efficiency of the strategy. In Section 4, we draw some conclusions.

2. Design of Puncturing Patterns

2.1. The Idea of "Gradually Sparsing"

In turbo codes, the puncturing pattern can change the BER distribution. Generally, a periodic puncturing pattern with a short periodic, such as the puncturing pattern p=[10; 01] that are commonly used to produced a turbo code with code rate 1/2, produces a uniformly distributed BER distribution with a nearly best average BER. Consequently, this puncturing pattern gives a nearly even error protection, that is, every bit has nearly the same error rate. On the other hand, a half puncturing pattern, noted as p_7 in [20] and expressed by

$$p_7 = \begin{bmatrix} 11...1100...00 \\ 11...1100...00 \end{bmatrix}$$

which produced a code rate 1/2 too, can give a very strong uneven error protection. The puncturing is carried on the two parity check sequences with second half of the two sequences are deleted totally. One of the curves in Figure 1 show the BER distribution produced by puncturing pattern p7 proposed in [20]. From Figure 1 it can be seen that it can give some bits very strong error protection, but with serious degradation for some other bits.

To overcome this drawback, tradeoff scheme are used in reference [11] and two new puncturing patterns, noted as p_2 and p_3, are proposed and expressed as

$$p_2 = \begin{bmatrix} 10...1011...11 \\ 11...1100...00 \\ 11...1110...10 \end{bmatrix}, p_3 = \begin{bmatrix} 11...1100...00 \\ 11...1111...11 \\ 11...1100...00 \end{bmatrix}$$

In these schemes, the puncturing is carried on not only on the parity check sequences, but also on the information sequence. For example, in p_2, the first half of the information sequence, as well as the second half of the second parity check sequence, are punctured alternatively. The second half of the first parity check sequence is deleted completely. The BER distributions produced by p_2 and p_3 are also displayed in Figure 1. Compared with the half puncturing pattern p_7 [20], the ability of strongest error protection for some bits is a little weaker, but the serious degradation appeared in half puncturing pattern is improved to some extents.

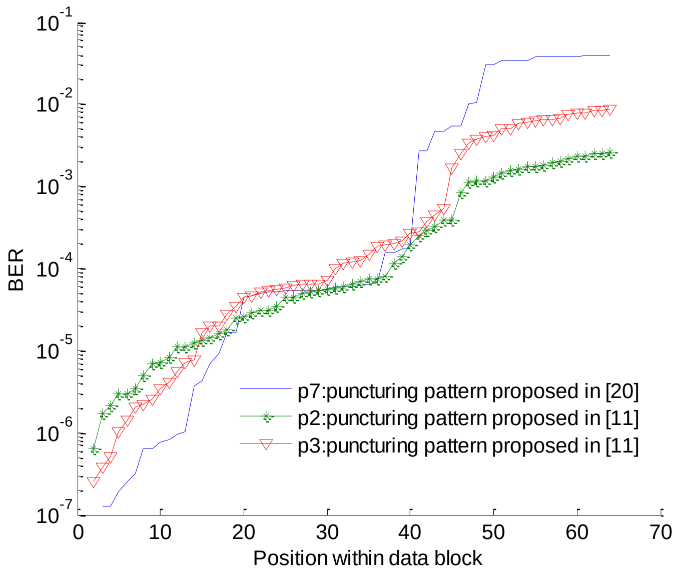


Figure 1. Comparison of the Reordering BER Distributions within a 64-Bit Data Block for a Turbo Code with Generator Polynomials (1,10001/10011), Using a 3gpp Interleaver. The Puncturing Patterns are P_7 Proposed in [20], P_2 and P_3 Proposed on [11] Separately. An Additive White Gaussian Noise (AWGN) Channel with a Signal-To-Noise Ratio (SNR) of 4 Db is Used. Turbo Decoding was Carried out Using 5 Iterations of the BCJR Algorithm. All the BER Distribution Curves are Obtained by Reordering the Bits in Ascending Order of BER

In order to improve the serious degradations for some other bits further as well as maintain the ability of strongest protection for some bits, a new scheme of puncturing pattern is presented in this paper with the idea of "gradually sparsing". "Gradually sparsing" means that, from the beginning to the end of the parity check sequence, more and more bits are punctured gradually.

2.2. Design Procedures of Puncturing Pattern

The design of puncturing patterns is composed of two main steps. Firstly, a basic puncturing pattern is constructed. Then the basic puncturing pattern is modified according to the interleaver length and code rate.

Let the interleaver size is L, and the required code rate is *Rate*. Selecting an integer k so that

$$2^k(k+1)+2^{k+1}-1 \le L \le 2^{k+1}(k+2)+2^{k+2}-1. \tag{1}$$

2.2.1. Basic Puncturing Sequence

The basic puncturing sequence for parity check one is constructed as

$$\underbrace{\underset{2^k}{1...1}0}_{2^0}\underbrace{\underset{2^{k-1}}{1...1}0\underset{2^{k-1}}{1...1}0}_{2^1}\underbrace{\underset{2^{k-2}}{1...1}0\underset{2^{k-2}}{1...1}0\underset{2^{k-2}}{1...1}0\underset{2^{k-2}}{1...1}0}_{2^2}\cdots\underbrace{\underset{2^1}{11}0\underset{2^1}{11}0...\underset{2^1}{11}0\underset{2^1}{11}0}_{2^{k-1}}\underbrace{1010...1010}_{2^k}$$

where "1" means the bits in the parity check sequence remain and "0" means the bits are deleted.

The basic puncturing sequence for parity check two is similar to one except that the "0" moves to the beginning in every subgroup. So the basic puncturing sequence for parity check two is as follow:

$$\underbrace{0\underset{2^k}{1...1}}_{2^0}\underbrace{0\underset{2^{k-1}}{1...1}0\underset{2^{k-1}}{1...1}}_{2^1}\underbrace{0\underset{2^{k-2}}{1...1}0\underset{2^{k-2}}{1...1}0\underset{2^{k-2}}{1...1}0\underset{2^{k-2}}{1...1}}_{2^2}\cdots\underbrace{0\underset{2^1}{11}0\underset{2^1}{11}...0\underset{2^1}{11}0\underset{2^1}{11}}_{2^{k-1}}\underbrace{01010...101}_{2^k}$$

Duo to the similar structures of sequence one and two, we will only consider basic puncturing pattern sequence one in next. The results can be easily applied to sequence two.

Apparently, the total number of "1" in the basic puncturing sequence is $l_{11}=(k+1)\times 2^k$. The total number of "0" in the basic puncturing sequence is $l_{10}=2^{k+1}-1$. So the total length of the basic puncturing sequence is

$$L_1 = l_{11}+l_{10} = 2^k(k+1)+2^{k+1}-1 \tag{2}$$

The basic rate is

$$rate1 = \frac{l_{11}}{l_{11}+l_{10}} = \frac{2^k(k+1)}{2^k(k+1)+2^{k+1}-1} \tag{3}$$

2.2.2. Modifying the Basic Puncturing Sequence

(a) If $L=2^k(k+1)+2^{k+1}-1$, and code *rate*1=*Rate*, the design is completed. Else, do the following.

(b) In this step, we will decide how many "1" and "0" shall be added to the basic puncturing sequence so that the total length is L and the code rate is *Rate* actually.

Because that the two parity check sequences are punctured similarly, that is, same number of bits are punctured in the two sequence, let l'_{11} be the number of "1" needed in the actual (modified) puncturing sequence, then the code rate is decided by the following

$$Rate = \frac{L}{L+2\times l'_{11}} \tag{4}$$

From above we get

$$l'_{11} = \frac{L(1-Rate)}{2Rate} \tag{5}$$

So more "1" should be added to the basic puncturing sequence, the number of extra "1" is

$$l''_{11} = l'_{11}-l_{11} = \frac{L\times(1-Rate)}{2\times Rate}-(k+1)\times 2^k \tag{6}$$

If $l'_{11} - l_{11} < 0$, the integer *k* should be decreased until $l'_{11} - l_{11} \geq 0$.

The number of "0" in the actual puncturing sequence is

$$l'_{10} = L - l'_{11} = L - \frac{L \times (1 - Rate)}{2 \times Rate} = L \times \frac{3Rate - 1}{2Rate} \quad (7)$$

Usually, more "0" should be added to the basic puncturing sequence, the number of extra "0" is $l''_{10} = l'_{10} - l_{10}$.

There are three cases about l''_{10} *and* l''_{11}.

Case1. If $l''_{10} = l''_{11}$, l''_{10} group of "10" should be added to the end of the puncturing sequence.

Case2. $l''_{10} > l''_{11}$, l''_{11} group of "10" should be added to the end of the puncturing sequence and $l''_{10} - l''_{11}$ "0" should be added to the end of the basic puncturing sequence.

Case3. $l''_{10} < l''_{11}$, l''_{10} group of "10" should be added to the end of the puncturing sequence and $l''_{11} - l''_{10}$ "1" should be added to the beginning of the basic puncturing sequence.

2.3. Example

Now, we give an example about the design of a puncturing pattern with interleaver size 64 and code rate 1/2.

Here, *L*=64, *Rate*=1/2. Firstly we construct the basic puncturing sequence. From the formula (1) we can get the integer *k*=3. So the basic sequence is

(111111110 1111011110 110110110110 1010101010101010)

There are 32 "1" and 15 "0" in the basic sequence. That is, $L_1 = 47$, $l_{11} = 32$ and $l_{10} = 15$. $l'_{11} = \frac{L(1 - Rate)}{2Rate} = 32$ and $l'_{10} = L - l'_{11} = 32$. Therefore, $l''_{11} = l'_{11} - l_{11} = 0$ "1" should be added to the beginning of the basic sequence and $l''_{10} = l'_{10} - l_{10} = 32 - 15 = 17$ "0" should be added to the end of the basic sequence. The final puncturing sequence for parity check one is (11111111011110111101101101101101010101010101010000000000000000 00) and the final puncturing sequence for parity check two is (0111111110111101111 011011011011 0101010101010101 00000000000000000)

3. Simulation Results and Comparisons with Other References

In this section, we will give several simulation results on turbo codes by employing the puncturing patterns designed in last section. The generating function of the turbo codes is the same, that is, g = (1,10001/10011), and we use the standard 3gpp interleavers for the turbo codes with different size. Turbo decoding is carried out using 5 iterations of the BCJR algorithm on AWGN channel. For comparison, the puncturing matrices proposed in [11], noted as p_2 and p_3, are also used in this section. We use the notion p_1 for the puncturing matrix proposed in this paper.

There are three figures in the followings according to the interleaver sizes of 64, 128 and 320 separately. In each figure, there are three curves to show the BER distributions produced by three different puncturing patterns. All the BER distribution curves are obtained by reordering the bits in ascending order of BER.

From the figures we can see that, compared with those produced by p_2 and p_3, the BER distributions produced by p_1 are nearly the same at the low BER region. But at high BER region, the improvements are significant.

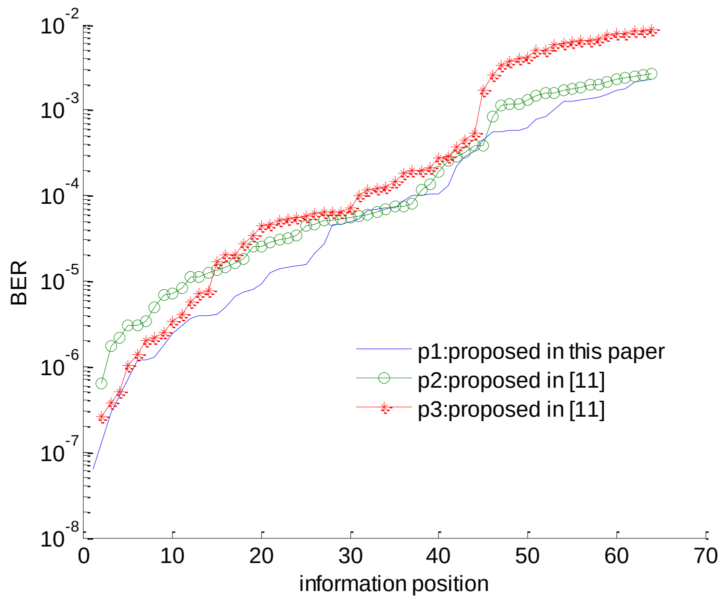


Figure 2. Comparisons of the Reordered BER Distributions for Puncturing Patterns P_1, P_2, P_3 with 3gpp Interleaver of Size 64 at E_b/N_0=4db

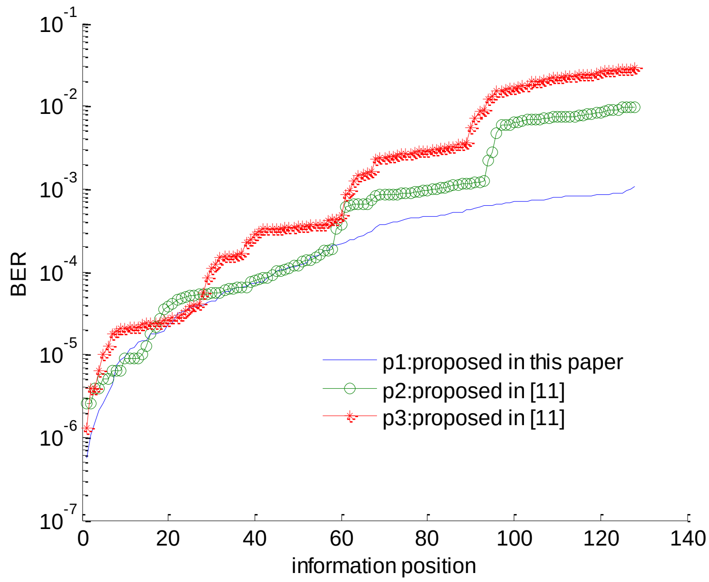


Figure 3. Comparisons of the Reordered BER Distributions for Puncturing Patterns P_1, P_2 ,P_3 with 3gpp Interleaver of Size 128 at E_b/N_0=3db

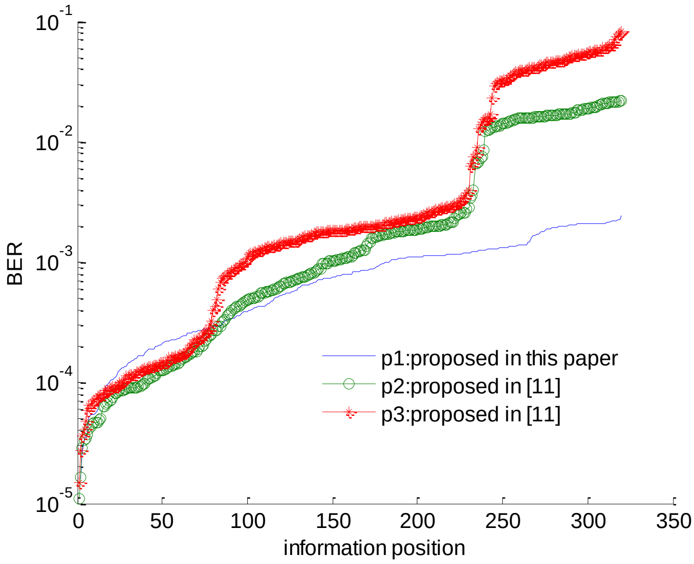


Figure 4. Comparisons of the Reordered BER Distributions for Puncturing Patterns P_1, P_2, P_3 with 3gpp Interleaver of Size 320 at E_b/N_0=2db

To see clearly the improvement of the unequal error protection, we divide the data block into 4 segments equally according to the unequal error protection level. Then, for every segment, we calculate the average BER. The results are displayed in Table 1. For example, the size of 64 is divided into four segments. The segment1 is from 1 to 16, segment2 from 17 to 32, segment3 from 33 to 48 and finally segment4 from 49 to 64. The average BER for these segments with puncturing pattern p1 are 2.07e-6, 2.83e-5, 2.48e-4 and 1.40e-3 respectively. From the table it can be seen that for every size, the average BER for segment1 are nearly the same no matter what the puncturing pattern is used. But from segment2 to segment4, the improvements produced by puncturing pattern p1 appear more and more noticeable. For example, the improvement of the average BER of segment4 for size 320 is more than 1 order of magnitude.

Table 1. The Four Segments Average Bers for the Three Cases

	64			128			320		
	P1	P2	P3	P1	P2	P3	P1	P2	P3
Segment1	2.07e-6	6.48e-6	4.72e-6	2.66e-5	3.49e-5	2.07e-4	1.71e-4	1.15e-4	1.42e-4
Segment2	2.83e-5	4.08e-5	6.22e-5	1.89e-4	4.17e-4	1.31e-3	5.41e-4	6.74e-4	1.31e-3
Segment3	2.48e-4	3.66e-4	9.60e-4	1.30e-3	4.70e-3	4.78e-2	1.10e-3	2.40e-3	3.50e-3
Segment4	1.40e-3	1.90e-3	6.60e-3	7.70e-3	2.28e-2	8.10e-2	1.80e-3	1.71e-2	4.71e-2

4. Conclusion

In this paper, in order to obtain strong unequal error protection for Turbo codes, we present a new scheme of designing the puncturing patterns for turbo codes. The new scheme is based on an idea of "gradually sparsing" which means that, from the beginning to the end of the parity check sequence, more and more bits are punctured gradually. Simulation results show that, compared with other puncturing schemes, this scheme can provide strong error protection for some bits without apparent degradation for other bits. In future work, we will consider the affections of the constructions of component encoders and interleavers to improve the puncturing performance further.

References

[1] J. H. Sørensen, P. Popovski and J. Østergaard, "UEP LT Codes with Intermediate Feedback", IEEE Communications Letters, vol. 17, no. 8, **(2013)**, pp. 1636- 1639.

[2] W. Yan and L. Rongke, "Enhanced unequal error protection coding scheme of Luby transform codes", IET Commun., vol. 9, no. 1, **(2015)**, pp. 33–41.

[3] W. Yeqing, K. Sunil, H. Fei, Z. Yingying and D. M. John, "Cross-Layer Forward Error Correction Scheme Using Raptor and RCPC Codes for Prioritized Video Transmission Over Wireless Channels", IEEE Transactions on Circuits & Systems for Video Technology, vol. 24, no. 6, **(2014)**, pp. 1047-1060.

[4] J. Yue, Z. Lin and B. Vucetic, "Distributed Fountain Codes With Adaptive Unequal Error Protection in Wireless Relay Networks", IEEE Transactions on Wireless Communications, vol. 13, no. 8, **(2014)**, pp. 4220-4231.

[5] K. Huang, C. Liang, X. Ma and B. Bai, "Unequal error protection by partial superposition transmission using low-density parity-check codes", IET Commun., vol. 8, no. 13, **(2014)**, pp. 2348–2355.

[6] X. Liangliang, W. Lin, H. Shaohua and W. Huihui, "New results on radiography image transmission with unequal error protection using photograph double LDPC codes", Firenze, Italy, Medical Information and Communication Technology (ISMICT), 2014 8th International Symposium on, **(2014)**, pp. 1-4.

[7] J. Hagenauer and T. Stockhammer, "Channel coding and transmission aspects for wireless multimedia", Proceeding IEEE, vol. 87, **(1999)**, pp. 1764-1777.

[8] A. S. Barbulescu and S. S. Pietrobon, "Rate compatible turbo codes", Electron. Lett., vol. 31, **(1995)**, pp. 535-536.

[9] M. Grangeto, E. Magli, and G. Olmo, "Embedding unequal error protection into turbo codes", in Proceeding Asilomar, vol. 1, **(2001)**, pp. 300-304.

[10] M. Aydinlik and M. Salehi, "Turbo coded modulation for unequal error protection", IEEE Trans. Commun., vol. 56, **(2008)**, pp. 555-564,

[11] W. Zhang, X. Shao, M. Torki, A. S. H. Mohammadi and I. V. Bajic¡ä, "Unequal error protection of JPEG2000 images using short block length turbo codes", IEEE Commun. Lett., vol. 15, no. 6, **(2011)**, pp. 659-661.

[12] M. Aydinlik, M. Semicond, M. A. Needham and M. Salehi, "Performance Bounds for Unequal Error Protecting Turbo Codes", Communications, IEEE Transactions on, vol. 57, no. 5, pp. 1215-1220.

[13] L. Yao and L. Cao, "Turbo codes-based image transmission for channels with multiple types of distortion", IEEE Transactions on Image Processing, vol. 17, no. 11, pp. 2112-2121.

[14] M. Qian and Q. Chuan, "A Novel Turbo Unequal Error Protection Scheme for Image Steganography", Informatica, vol. 24, no. 4, **(2013)**, pp. 561-576.

[15] D. Fang and A. G. Burr, "Quantization based Estimate-and-Forward with unequal error protection for Turbo coded Physical layer network coding on the TWRC", Wireless Communication Systems (ISWCS), 8th International Symposium on, **(2011)**, pp. 859-863.

[16] W. Abdul, K. David, H. Werner and N. H. Beltrao, "Leaking interleavers for UEP Turbo codes", Brest, France, 6th International Symposium on Turbo Codes and Iterative Information Processing, **(2010)**, pp. 191-195.

[17] T. Gnanasekaran, R. R. Ganapathi, S. U. Maheswari, K. Duraiswamy and A. P. Kapilan, "Unequal Error Protection Using Maximum A-Posteriori Probability (MAP) Algorithm, Soft Output Viterbi (SOVA) Algorithm and Modified SOVA", Nagpur, Maharashtra. Proceedings-1st International Conference on Emerging Trends in Engineering and Technology, ICETET, **(2008)**, pp. 1298-1301.

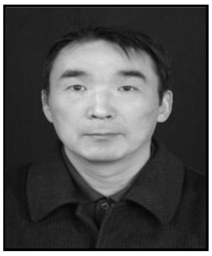
[18] A. L. Moulay, R. Méliani and M. Kandouci, "Research on unequal error protection with punctured turbo codes in jpeg image transmission system", Serbian Journal of Electrical Engineering, vol. 4, no. 1, **(2007)**, pp. 95-100.

[19] S. Vafi and T. A. Wysocki, "Application of convolutional interleavers in turbo codes with unequal error protection", Journal of Telecommunications and Information Technology, no. 1, **(2006)**, pp. 17-23.

[20] S. Xia and Z. Weidang, "Estimate BER Distributions of Turbo Codes", International Journal of Wireless and Microwave Technologies (IJWMT), vol. 2, no. 2, **(2012)**, pp. 53-58.

Authors

Shao Xia, She received her M.S. degree and B. S. degree from Zhengzhou University in 2007 and 1992 separately. Currently she is with the Department of Information Engineering, North China University of Water Resources and Electric Power. 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.