The Design of RA Code Pseudo-random Interleaver

Yanchun Shen, Guozhong Zhao*, Shengbo Zhang, Yashang Li, Shuai Li

Key Laboratory of Terahertz Optoelectronics, Ministry of Education, Physics Department, Capital Normal University, Beijing, 100048 China
cnsyc@126.com

Abstract. In the process of digital signal transmission, due to noise and other interference signals, the code element waveform will produce some changes. RA code can encode and decode in linear time, which is one of the key techniques of channel coding. Interleaver is a very important part of RA code encoder, so its performance affects the performance of RA code. A S-pseudo-random interleaver and deinterleaver is designed in this paper, and the model is verified by simulation. The experimental results show that the bit error rate (BER) of interleavers with different interleaver lengths is significantly different. The longer the interleaver length is, the better the performance of the interleaver is, and the closer to the Shannon limit is.

1. Introduction
Repeat Accumulate Code (RA code for short) combines the advantages of Turbo code and LDPC code and can be encoded and decoded in linear time. Block codes and convolutional codes can check and correct random independent errors, but for channels with memory, error codes are no longer randomly distributed independently. Due to the memory of the channel, the performance of checking and correcting errors of block codes and convolutional codes is degraded.

Using interleaving technique, the encoding operation can be performed without knowing the statistical characteristics of the channel with memory. Interleaving and deinterleaving play an important role in channel coding and decoding and reduce random errors, which is helpful to improve the performance of transmission system. In Turbo codes, the interleaver is used to reduce the correlation between the check bits, and then reduce the bit error rate in the iterative decoding process. C. Berrou, et al. gave the evaluation principles and characteristics of the interleaver design at the beginning of Turbo codes: (1) Increasing the interleaver length can improve the decoding performance. (2) The interleaver should make the input sequence as random as possible, that is to say, the free distance of Turbo codes should be as large as possible[1].

2. Working Principle of Interleaver
The design of interleaved coding is not to adapt to the channel, but to modify the channel. By interleaving and deinterleaving, a burst channel with memory is transformed into a random independent error channel without memory to correct the errors. The process of permutating code element in an error-correcting encoded code sequence in a predetermined manner is called interleaving.

If there is a burst of continuous error in the signal transmission process, these errors will appear on a continuous segment of the code sequence. After interleaving, the interleaved code becomes relatively discrete and randomized, and these continuous errors can be discretized into relatively short or independent error. At the receiving end, the interleaved code sequence is replaced by the original
code sequence in the opposite way, that is, deinterleaved. Finally, the error correction codes such as block codes or convolutional codes are used to correct the discrete short error or random independent error in the sequence. The response interleaver is also called pseudo-random interleaver because the input sequence is long and the approximate random mapping is needed.

Design process of pseudo-random interleaver:
To pick an integer \( i \) at random from \( S = \{1, 2, \cdots, N\} \) and delete \( i \) from \( S \) to obtain new set \( S_1 \).

Follow step 1 until \( N \) times are taken and the interleaving ends.

\[
\begin{align*}
  u_i &\in \{0,1\}, \quad i = 1, 2, \cdots, N, \text{ mapped output sequence after interleaving} \\
  u_j &\in \{0,1\}, \quad j = 1, 2, \cdots, N
\end{align*}
\]

In the design, the pseudo-random address interleaver is generated first, and then the corresponding sequence value is found according to the pseudo-random address, that is, a rearranged sequence is generated, and the new sequence is output. An ideal pseudo-random interleaver is used in this design.

3. Pseudo-random Interleave
An interleaver in the decoder corresponds to an interleaver in the encoder. The pseudo-random interleaver uses the random positive integer generated by a specific algorithm as the interleaver address to interleave the information sequence, and then uses the interleaver map to read the sequence data from the interleaver to complete the interleaving. For example, a random interleaver having an input sequence length of 6 writes the code sequence \([111000]\) to the matrix and reads it as \([010101]\) through the interleaver.

The block diagram of random interleaver is shown below:

![Fig.1 Working principle of pseudo-random interleaver](image)

In pseudo-random interleaver, the most typical one is \( S \) pseudo-random interleaver, whose random number is generated in the same way as other interleavers, but the distance between the information sequence symbols must be greater than \( S \) units before interleaving. This interleaver can better realize the discretization and randomization of code elements. \( S \) pseudo-random interleaver is an ideal random interleaver because of its high speed of interleaver generation.

Design principle of \( S \) pseudo-random interleaver:
1) The 0 and 1 code sequence which needs to be input is generated at the data transmitting end. Assuming the sequence length is \( N \), the sequence is written into the interleaved register (i.e. Interleaved matrix).

2) Generating an \( S \) value in the \( S \) pseudo-random interleaver according to the \( N \) value;

3) Starting from the head position, a random number between 1 and \( N \) with the same length as the input data sequence is generated by using a linear remainder method.
(4) The generated random number is compared with the previously generated random number, and if the absolute value of the difference between the two is greater than the positive integer S, the condition is met; otherwise, a random number is regenerated and the comparison is continued until the condition is met. In addition, each random number can only appear once until all the N random numbers are generated.

(5) The elements in the 0,1 code sequence are sequentially read out from the interleaving register to complete the interleaving by using the generated random number which meets the conditions as an address;

(6) When the receiving end performing deinterleaving, addresses are given to the generated random numbers, and data code elements corresponding to the positions in the interleaved sequence are read according to the addresses of the random numbers, and the deinterleaved matrix is written to complete the deinterleaving. The deinterleaver flow is shown in Fig.2:

![Fig.2 The program flow chart of S pseudo-random interleaver deinterleaving](image)

4. Simulation and Analysis of Interleaver and Deinterleaver

4.1. Simulation of S Pseudo-random Interleaver and Deinterleaver
In order to verify the function of interleaver, the information sequence which can set the length of sequence arbitrarily is added to the input of interleaver, the length of sequence is interleaver length, and random address is generated. The S distance is calculated by using the formula $S \leq \left\lfloor \sqrt{N/2} \right\rfloor - 1$ of S distance in S pseudorandom interleaver, then comparing the distance between the random addresses of the sequence until N addresses are selected. The sequence elements are read out according to their addresses to realize interleaving. The discrete state of the elements between the interleaved sequence and the original sequence is observed and deinterleaved finally.

When the receiver is deinterleaved, the random number is assigned to the address, and the address of the random number is taken as the reference at the receiver, and the interleaved sequence is read in turn to realize deinterleaved.

4.2. Simulation Result Analysis of S Pseudo-random Interleaver and Deinterleaver
The S pseudo-random interleaver and deinterleaver is simulated and analyzed, 20 points are selected, and the original input code sequence before interleaving is shown in Fig.3:
As shown in Fig. 3: Interleaved 0, 1 code sequence graphics. The task of the interleaver is to break up the connected "0" and "1" appearing in the original sequence.

From the above figure, it can be seen that S pseudo-random interleaver can realize the discretization of data sequence elements, and the parameters of S pseudo-random interleaver satisfy $S \leq \left\lfloor \sqrt{N/2} \right\rfloor - 1$, where N is the interleaver length. With the increase of S value, the performance of the interleaver is improved. To achieve better interleaver performance to select the appropriate S distance. The deinterleaved output sequence pattern is shown in Fig. 5:

Fig. 3 Original input code sequence

Fig. 4 Interleaved code sequence

Fig. 5 Deinterleaved code sequence
By comparison, the original input sequence is consistent with the deinterleaved sequence, which proves that the interleaver and deinterleaver can achieve the correct transmission of the sequence in the channel.

The interleaver length affects the performance of the interleaver, which in turn affects the bit error rate. Performance comparison of pseudorandom interleavers at different interleaving lengths, as shown in Fig.6:

![Fig.6 Comparison of performance under different interleaving lengths](image)

Interleavers with interleaver lengths of 100, 400, 900 are selected to compare the bit error rate results. Simulation results show that the longer the interleaving length, the better the performance of the interleaver, and the closer to the Shannon limit.

5. Conclusion

Based on the results and discussions presented above, the conclusions are obtained as below:

Interleaver is a very important part of RA code encoder/decoder architecture. The pseudo-random interleaver can interleave the information sequence randomly, which has higher confidentiality and fewer errors. The simulation results of interleaver model show that the bit error rate of interleaver with different interleaver lengths is obviously different. The longer the interleaver length is, the better the performance of interleaver is. However, with the increase of interleaver length, the interleaver will bring more system delay and equipment complexity. Therefore, the selection of appropriate interleaving length plays an important role in improving the system performance.

Acknowledgments

This work was financially supported by the National Natural Science Foundation of China (Grant Nos.61575130, 61575131, 50971094).

References

[1] YANG Zekun, JIN Dongli. Research and simulation of Turbo code for CCSDS standard[J]. Radio Communication Technology, 2017,49(06):132-141.
[2] He Xiangguo. Design and Optimization of Pseudo-random Interleaver in Turbo Code[D]. Hangzhou University of Electronic Science and Technology, 2014, 120-128.
[3] Cui Zhongpu, Gao Jun, Dou Gaoqi. Analysis and research of Turbo code decoding algorithm [J]. Communication Technology, 2016,37(09):23-27.
[4] Zhang Hongtai, Liu Hongli, Liu Shugang. Design of RA code interleaver based on full binary tree[J]. Computer Engineering and Science, 2015,57(04):103-108.
[5] Zhang Jingyi, Jin Minglu. An improved RA decoding algorithm [J]. Journal of Electronics and Information, 2011,49(11):46-50.
[6] Wen Wu, Xiao Hejing, Zhu Lianxiang. Design of multivariate RA code interleaver based on improved PEG algorithm [J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2010,37(02):69-73.
[7] Liu Donghua. Principle and Application Technology of Turbo Code [M], Beijing: Publishing House of Electronics Industry, 2004.