A joint design method of frame synchronization code and LDPC code

Runlu Tian¹, Kexian Gong¹, Peng Sun^{1,*}, Zhongyong Wang¹, and Mengke Ren¹

¹College of Information Engineering, Zhengzhou University, 100 Science Avenue, Zhengzhou City, Henan Prov, China

Abstract. Frame synchronization word (FSW) is an important basis in signal synchronization detection. In our work, the joint design of frame synchronization words and error correcting code are proposed which through regularly distributing the frame synchronization words among the information bits to improve the decoding performance of the LDPC. Frame synchronization word should be added to the information sequence corresponding to the position with the check matrix larger column weight which can guarantee that more Frame synchronization words are received by the check node in the decoding iteration. The frame synchronization word is known to the receiver which play an important role in decoding iteration. The results of simulation show that the algorithm makes an obvious improvement in decoding performance when the signal-noise ratio (SNR) is lower than 1 dB.

1 Introduction

In non-cooperative communication reception, frame synchronization is an indispensable part, and it relies on frame synchronization words for synchronization identification. The frame synchronization character is not only used for synchronization recognition, but also its certainty can be used to improve the performance of signal decoding. An algorithm based on frame synchronization word distribution is proposed to improve LDPC decoding performance. The frame synchronization words are distributed in the LDPC information sequence according to certain regulations, so that the frame synchronization words can maximize the decoding performance during the decoding iteration process. LDPC code is a reliable channel coding, which is widely used in modern communication systems due to its iterative decoding performance close to Shannon limit. In recent years, there have been an endless number of improved algorithms for LDPC code. Algorithms based on soft information iteration are all based on confidence propagation algorithm. They achieve the goal of reducing complexity or improving performance by changing some steps of the algorithm, but the performance improvement will be accompanied by an increase in the amount of calculation. However, it has not been proposed to the design method of combining frame synchronization word and LDPC, so there is a large research space in this aspect.

* Corresponding author: jepengsun@zzu.edu.cn

In recent years, there has been a relatively mature system for the study of Frame synchronization. Literature [1] proposes a multiple of frame length is estimated via the correlation and then other frame parameters are acquired using the mean of periodic samples with the estimate. A new algorithm based on the fluctuation of the correlation function of the received signals in time domain is proposed in [2]. The algorithm, employing the first-order fluctuation function of the correlation function as the detection function and combining with the SDM adaptive threshold estimation method to detect burst signals. However, the plenty of improved LDPC algorithm are based on the improvement of algorithm steps and the reduction of complexity. The grouping LDPC coding system is proposed in literature [3], and developed a semi-random method for block LDPC code construction. The check matrix is divided into several block matrices for coding with certain constraints, but the coding complexity is increased. Literature [4] has formed a simple and fast encoder by analyzing the parity matrix of quasi-cyclic LDPC (QC-LDPC) codes, but it is not suitable for random construction. Literature [5] optimizes the irregular LDPC code of the DE-LDPC encoding system through multi-symbol differential detection (MSDD) to obtain an optimized code with good performance.

Based on the above analysis, a joint design method of frame synchronization word and channel coding is proposed, and determines the location of frame synchronization words in the information sequence according to the check matrix of LDPC code. Since the frame synchronization word is fully known by the decoding party which is very helpful to other code words in the decoding iteration. There is an introduction to LDPC in the second section, a detailed derivation of the distribution law of frame synchronization words and the theoretical analysis of decoding performance are show in the third section. The fourth part verifies the influence of the frame synchronization word on the decoding performance through simulation experiments.

2 LDPC overview

2.1 Representation of LDPC codes

The LDPC code is a linear block code expressed as (n, k) , where 'n' is the code length and 'k' is the information sequence. The check matrix 'H' is sparse, and this sparseness ensures that the decoding complexity and minimum code distance only increase linearly with the code length. Each row of the check matrix 'H' corresponds to a check equation, and each column corresponds to a bit in the sequence. The number of non-zero elements in each column is called column weight. For a check matrix 'H' of dimension $m \times n$, the code word $c = [c_1, c_2, \dots, c_n]$, which is satisfied $H \cdot c^T = 0$.

2.2 Regular LDPC code and irregular LDPC code

If the number of non-zero elements in each row and column of the check matrix is the same, such LDPC code is called a regular code, otherwise it is called an irregular code.

The performance of irregular codes is better than that of regular codes through the analysis of the decoding process. In decoding, the greater degree of the variable node, the more check nodes connected to it, then the decoding performance is better[7]. In this way, variable nodes can obtain more confidence information from more check nodes, and can more accurately determine the correct value of variable nodes. On the contrary, the smaller the degree of the check node, the fewer the variable nodes connected to the check node, the higher message confidence to the connected variable nodes can be provide. For this contradiction, irregular codes can be a good compromise between the degree of variable

nodes and check nodes. In the decoding process of irregular codes, high degree variable nodes can receive more confidence messages. These variable nodes can achieve correct decoding faster as the number of iterations increases. In this way, it can send more reliable information to the connected check nodes. And these check nodes can give more information to the variable nodes with smaller degrees which connected to them, so that the decoding can converge quickly. In this way, variable nodes with high degrees are decoded correctly first, and then variable nodes with low degrees are decoded in turn until all variable nodes are completely decoded.

3 LDPC code algorithm innovation

3.1 Distribution of frame synchronization word

In channel transmission, the first task of the receiver is to detect the starting point of the received data. It usually sends a certain sequence as the preamble which is frame synchronization word used for sync detection. It will play a vital role in the decoding iteration because of the complete determinism of the frame synchronization word. It is an important problem that how to make the frame synchronization words to maximize the decoding performance. How many frame synchronization words are added, and how are the frame synchronization words arranged?

The purpose of decoding is to restore the received code word. Every time a frame synchronization word is added, a code word is determined. And the permutation and combination of the remaining code words are reduced accordingly, the probability of correct decoding is also improved. But the transmission efficiency decreases from the number of frame synchronization words increases. In this paper, we use the addition of 64-bit, 128-bit and 256-bit frame synchronization words for simulation comparison.

The distribution of frame synchronization words is also a question worth pondering. It can be seen from the Tanner graph of the check matrix that in the decoding iteration process, when each check equation receives a frame synchronization word from the message of the variable node, the variable node can be judged more accurately by the check equation. At this time, code performance has been improved to the greatest extent. Each row of the check matrix corresponds to a check equation, and which check equation is connected to the variable node is determined by the position of each row "1" in the check matrix. It can be seen that the column weight of the check matrix is the number of check equations connected to each variable node. Therefore, in order to the most obvious promotion of decoding performance, frame synchronization words should be added to the information sequence corresponding to the position with the largest column weight. In this way, the frame synchronization word information can be passed to the most check equations.

Through the above analysis, the frame synchronization word distribution can be determined by the check matrix. We use Gaussian elimination algorithm and DVB-S2 standard algorithm for simulation. The check matrix column weight of Gaussian elimination algorithm is 3. The column weights of the DVB-S2 standard algorithm are 8 and 3 respectively, and the column weight of information bits is 3. Therefore, the Gaussian elimination algorithm uses the frame synchronization word to be evenly distributed among the information bits with a code length of 512, while the DVB-S2 standard algorithm uses the frame synchronization word to be evenly distributed among the 8100 information bits.

3.2 Theoretical analysis of decoding performance

Suppose that the check node has k edges and the variable node has j edges in the decoding iteration. The channel transition probability of the binary channel is p_0 , and its range is 0 to 0.5. If the information sent by an information node has $j-1$ check equations are not satisfied, it is considered that this bit is wrong and the bit of this node is modified. Otherwise it is considered that the bit of this node is correct.

It is assumed that an error occurred during the first iteration. The receiving bit will fail if the other bits involved in the check have an even number of errors. And the probability of an even number of errors in the other $k-1$ bits is $\frac{1+(1-2p_0)^{k-1}}{2}$.

If $j-1$ check equations are not satisfied, it is considered that an error has occurred in the transmission and modify this node bit. Therefore, the $j-1$ check equations are not satisfied and the probability of bit receiving error is $p_0 \left[\frac{1+(1-2p_0)^{k-1}}{2} \right]^{j-1}$. So the probability that

the received bit is correct is $(1-p_0) \left[\frac{1-(1-2p_0)^{k-1}}{2} \right]^{j-1}$.

According to the above two expressions, the error probability p_1 of the first iteration is:

$$p_1 = p_0 - p_0 \left[\frac{1+(1-2p_0)^{k-1}}{2} \right]^{j-1} + (1-p_0) \left[\frac{1-(1-2p_0)^{k-1}}{2} \right]^{j-1} \quad (1)$$

In the same way, based on the error probability p_i of the i times iteration, the probability p_{i+1} of the $i+1$ times error can be obtained is:

$$p_{i+1} = p_0 - p_0 \left[\frac{1+(1-2p_i)^{k-1}}{2} \right]^{j-1} + (1-p_0) \left[\frac{1-(1-2p_i)^{k-1}}{2} \right]^{j-1} \quad (2)$$

where p_0 is the channel transition probability.

When the frame synchronization words are distributed among information bits, it is not affected by channel characteristics due to its complete Certainty. It can be regarded as a noise-free channel for frame synchronization words, so $p_0 = 0$. Suppose the channel error probability in this case is $p_{uw} = 0$. At this time, the probability of an even number of errors

in the other $k-1$ bits changes from $\frac{1+(1-2p_0)^{k-1}}{2}$ to $\frac{1+(1-2p_0)^{k-2}(1-2p_{uw})}{2}$. Then formula (1) becomes:

$$p_1 = p_0 - p_0 \left[\frac{1+(1-2p_0)^{k-2}(1-2p_{uw})}{2} \right]^{j-1} + (1-p_0) \left[\frac{1-(1-2p_0)^{k-2}(1-2p_{uw})}{2} \right]^{j-1} \quad (3)$$

Due to the $1-2p_{uw}=1$, so formula (3) becomes:

$$p_1 = p_0 - p_0 \left[\frac{1+(1-2p_0)^{k-2}}{2} \right]^{j-1} + (1-p_0) \left[\frac{1-(1-2p_0)^{k-2}}{2} \right]^{j-1} \quad (4)$$

The formula (1) minus formula (4) obtain the error probability difference before and after adding frame synchronization words is:

$$\delta = -p_0 \left[\frac{1+(1-2p_0)^{k-1}}{2} \right]^{j-1} + p_0 \left[\frac{1+(1-2p_0)^{k-2}}{2} \right]^{j-1} + (1-p_0) \left[\frac{1-(1-2p_0)^{k-1}}{2} \right]^{j-1} - (1-p_0) \left[\frac{1-(1-2p_0)^{k-2}}{2} \right]^{j-1} \quad (5)$$

And because the range of p_0 is $[0,0.5]$, so the range of $1-2p_0$ is $[0,1]$. Assume that the first to fourth terms of formula (17) are a_1, a_2, a_3, a_4 , then formula (5) becomes $\delta = -a_1 + a_2 + b_1 - b_2$. Analysis to $a_1 < a_2, b_1 > b_2$, so get $\delta > 0$. It is the error probability is reduced δ after adding frame synchronization words.

3.3 Density evolution theory

Density evolution theory [7] is an important method to analyze the performance of LDPC codes. Through the analysis of the decoding iteration process, the transfer of probability is replaced by the transfer of probability density function, and finally the theoretical bit error rate is calculated through integration.

In the BP decoding algorithm, the probability-likelihood ratio of the message is passed back and forth on the edge of the Tanner graph, and the decoding is achieved by iterative update of the message. According to the BP algorithm, v^l represents the LLR information from the variable node to the check node in the L times iteration, and u^l represents the LLR information from the check node to the variable node in the L times iteration.

$$v_i^l = \lg \frac{p(x=1|t)}{p(x=-1|t)} \quad (6)$$

$$u_i^l = \lg \frac{p(x=1|r)}{p(x=-1|r)} \quad (7)$$

where 't' is all the information received by this variable node in this iteration, and 'r' is all the information received by this check node in this iteration.

The iterative update process of the message during BP decoding is [8,9]:

$$v_i^{l-1} = v_i^0 + \sum_{j=1}^{d_v-1} u_j^l \quad (8)$$

$$\tanh \frac{u_j^l}{2} = \prod_{i=1}^{d_c-1} \tanh \frac{v_i^{l-1}}{2} \text{ or } u^l = 2 \tanh^{-1} \left[\prod_{i=1}^{d_c-1} \tanh \frac{v^l}{2} \right] \quad (9)$$

where d_v is the number of nodes connected to variable nodes, and d_c is the number of nodes connected to check nodes.

In the decoding process, the values of v^l and u^l are randomly distributed, so they are random variables with a certain probability distribution. Let $P^l(v)$ represent the probability density function of each v^l , and $Q^l(u)$ represent the probability density function of each u^l .

When the first iteration of decoding begins, the initial probability density of v^l and u^l is determined by the transition probability of the channel. With each iteration of the decoding algorithm, $P^l(v)$ and $Q^l(u)$ will change accordingly. The density evolution is to track the values of $P^l(v)$ and $Q^l(u)$ in each iteration to determine the bit error rate [10].

According to probability theory, if multiple independent random variables are added together, the probability density of the sum is equal to the convolution of the probability density function of the original random variable. So according to formula (8), the density evolution from variable node to check node is:

$$P^l(v) = P^0 * \otimes_{j=1}^{d_v-1} Q^l(u) \quad (10)$$

The density evolution from check node to variable node is more complicated. First define the operator $r(x, y) = 2 \tanh^{-1}(\tanh \frac{x}{2} \tanh \frac{y}{2})$, then formula (9) becomes

$u^l = r(v_1, r(v_2, \dots, r(v_{d_c-1}, v_{d_c-2}) \dots))$. Assume $R(P_x, P_y) = \sum_{(x,y)=r(x,y)} P_X(x)P_Y(y)$, Then the

density evolution from check node to variable node is:

$$Q^l(u) = R(P^l(v), R(P^l(v), \dots, R(P^l(v), P^l(v)) \dots)) \quad (11)$$

According to formula (10), the bit error rate can be expressed as:

$$p_e^l = \int_{-\infty}^0 P^l(v) dv \quad (12)$$

3.4 Gaussian approximation

Aiming at the memoryless binary input and output AWGN channel, approximate a simplified LDPC decoding method to estimate the theoretical performance of the code. The density evolution of multi-dimensional check messages and variable messages is simplified to the evolution of one-dimensional mean value. This method of reducing the computational complexity is called Gaussian approximation.

When the channel model is a Gaussian channel, the initial probability of decoding obeys the Gaussian distribution. In the process of density evolution, the variable node probability density function $P^l(v)$ still obeys the Gaussian distribution. Although the check node probability density function $Q^l(u)$ is not Gaussian, its waveform can be approximately Gaussian distribution [11].

When density evolution satisfies symmetry [12], we can only consider the mean and variance in the iterative process. Assuming that in the L times iteration, the density distribution of variable node LLR messages is $N(m_v, 2m_v)$, the density distribution of verification messages is $N(m_u, 2m_u)$, and the distribution of initial variable messages is $N(m_0, 2m_0)$. Since in the first iteration, variable nodes and check nodes are independently identically distribution, Since in the first iteration, variable nodes and check nodes are independently identically distribution, the mean of equations (8) and (9) is [13]:

$$m_v^l = m_0 + (d_v - 1)m_u^l \quad (13)$$

$$E[\tanh \frac{u_j^l}{2}] = E[\prod_{i=1}^{d_c-1} \tanh \frac{v_i^{l-1}}{2}] = E[\tanh \frac{v_i^{l-1}}{2}]^{d_c-1} \quad (14)$$

Define the function:

$$\phi(x) = \begin{cases} 1 - \frac{1}{\sqrt{4\pi x}} \int_{-\infty}^{\infty} \tanh \frac{u}{2} \exp(-\frac{(u-x)^2}{4x}) du, & x > 0 \\ 1, & x = 0 \end{cases} \quad (15)$$

where $\phi(x)$ decreases monotonically on $[0, \infty)$, $\phi(0) = 1$, $\phi(\infty) = 0$. Thus, the mean of the check message is:

$$m_u^l = \phi^{-1} \{ 1 - [1 - \phi(m_v^{l-1})]^{d_c-1} \} \quad (16)$$

Combining formula (16) and formula (13) can get the mean m_v^l of the variable node. According to the probability density distribution $P^l(v) = N(m_v^l, 2m_v^l)$ of the variable message in the L times iteration, the bit error rate is:

$$p_e^l = \int_{-\infty}^0 P^l(v) dv \quad (17)$$

where $P^l(v) = \frac{1}{\sqrt{4\pi m_v^l}} \exp[-\frac{(v-m_v^l)^2}{4m_v^l}]$.

Through numerical calculation, $\phi(x)$ can be obtained as:

$$\phi(x) = \begin{cases} \sqrt{\frac{\pi}{x}} (1 - \frac{10}{7x}) \exp(-\frac{x}{4}), & x \geq 10 \\ \exp(-0.4527x^{0.86} + 0.0218), & 0 < x < 10 \end{cases} \quad (18)$$

The following is the first iteration of the decoding process to analyze the changes when frame synchronization words are added. The initialization of the first iteration is $m_v^0 = m_0 = \frac{2}{\sigma^2}$, σ^2 is the noise variance of the AWGN channel. According to formula (16), the density evolution of the check node becomes:

$$m_u^1 = \phi^{-1} \{ 1 - [1 - \phi(m_0)]^{d_c - 1} \} \quad (19)$$

According to formula (11), the density evolution of the variable node becomes:

$$m_v^1 = m_0 + (d_v - 1)m_u^1 \quad (20)$$

Through the simulation of the first iteration process, the simulation results are shown in the figure 1. Describes that after three iterations under different SNR, the distribution diagram of the probability density function and the curve diagram of the decoding performance before and after the frame synchronization word is added. As shown in the figure, (a) describes that after a frame synchronization word is added in the decoding process, the mean value of its probability density function becomes larger and the graph moves to the right. And figure (b) is calculated by formula (16), which describes the graph of bit error rate and SNR after 3 iterations. It can be obtained that under the same SNR, the decoding error rate is significantly reduced after adding frame synchronization words.

Fig. 1. (a) The probability density distribution of variable nodes before and after the frame synchronization word is added. **(b)** Curve of decoding error rate before and after adding frame synchronization words.

4 MATLAB simulation realization of LDPC codec

In this paper, a large number of simulations have been carried out on the joint innovative algorithm of frame synchronization code and channel coding under AWGN channel. In this paper, Gaussian elimination and DVBS-2 standard coding algorithm are adopted to generate (1540, 512) codes with a code rate of 1/3 and (16200, 8100) codes with a code rate of 1/2. Distribute the frame synchronization words in the information bits according to the law, and do simulation analysis from the aspects of SNR, bit rate and the number of frame synchronization words.

Figure 2 shows a simulation diagram of the decoding performance of (1540, 512) codes and (16200, 8100) codes with different numbers of frame synchronization words as the SNR increases. Among them, figure (a) shows the (1540, 512) code distribution frame synchronization words have 32, 64, 128 bits. The decoding algorithm with 128-bit frame synchronization words reduces the bit error rate by about 0.1dB compared with the decoding algorithm without frame synchronization words. And figure (b) shows the (16200, 8100) code distribution frame synchronization words have 64, 128, 256, and 512 bits. When the frame synchronization word is 512 bits, the performance is improved by approximately 0.1dB. Figure (c) shows the simulation of this algorithm under the Rayleigh fading channel. When there are 128-bit frame synchronization words, the decoding performance is improved by about 3dB. It can be seen from the figure that the more frame synchronization words are added, the lower the bit error rate under the same SNR.

Fig. 2. Decoding performance under AWGN channel when (a) (1540, 512) code and (b) (16200, 8100) code have different frame synchronization words. (c) Decoding performance under Rayleigh fading channel when (1540,512) code have different frame synchronization words.

Figure 3 shows the simulation diagram of (1540, 512) code under 0.5dB SNR, and (16200, 8100) code under 0.78dB SNR after 1000 cycles, the decoding performance changes with the increase of frame synchronization words. As the number of frame synchronization words increases, the decoding bit error rate decreases. When the number of frame synchronization words added reaches a certain number, the error rate is zero. It can be seen from the figure that the bit error rate of decoding of (1540, 512) code drops to 10⁻⁴ when the frame synchronization word is added with 130 bits, and the frame synchronization word continues to increase the bit error rate drops to zero. (16200, 8100) code drops to 10⁻⁶ when the frame synchronization word is added with 780 bits, and the frame synchronization word continues to increase the bit error rate drops to zero.

Fig. 3. (a) (1540, 512) Code and (b) (16200, 8100) code decoding performance changes with the number of frame synchronization words added.

Figure 4 shows the (4100, 2048) code when the SNR is 0.8dB, and the (16200, 8100) code when the SNR is 0.8dB. As the number of iterations increases, the convergence performance of decoding when frame synchronization words are not added is added. It can be seen from the figure that the more frame synchronization words are added, the lower the bit error rate when decoding converges.

Fig. 4. The bit error rate of (a) (4100, 2048) code and (b) (16200, 8100) code when different frame synchronization words are added to the decoding performance convergence.

Figure 5 shows the effect of 128-bit frame synchronization words added in (16200,8100), (4100,2048) and (1540,512) codes on decoding performance. It can be seen from the simulation diagram that the most obvious improvement in decoding performance before and after the frame synchronization word is added is the (1540, 512) code, the (4100, 2048) code has a smaller improvement, and the smallest improvement is the (16200, 8100) code. It can be concluded that the greater the proportion of the frame synchronization word to the code length, the more obvious the improvement in decoding performance.

Fig. 5. The effect of the same frame synchronization word on the decoding performance of different code lengths.

Figure 6 simulates the conclusion in section 3.1 Namely, distributing frame synchronization words at the information sequence corresponding to the position with the largest column weight can maximize the decoding performance. As shown in the picture for the (16200, 8100) code, add 512-bit frame synchronization words to the parity check matrix to compare the decoding performance at the largest column weight and the smallest column weight.

Fig. 6. The influence of the distribution of frame synchronization words on decoding performance.

5 Conclusions

We proposes a joint innovative algorithm of frame synchronization and error correcting code, which effectively improves the decoding performance of error correcting code. In this algorithm, the frame synchronization word is not only used for synchronous detection, but also effectively improves the decoding performance of LDPC. It is not affected by code length, code rate and decoding mode. Simulation results show that the decoding performance is effectively improved. However, the non-centralized distribution of frame synchronization words makes the sync detection have to wait for the reception of the entire sequence to be completed.The detection delay increases and further research is needed.

Funding: This research was funded by the Development Program “Frontier Scientific and Technological Innovation”Special under grant 2019QY0302, the National Natural Science Foundation of China under Grant U1736107 and the National Natural Science Foundation of China under grants 61901417.

References

1. Y.S. Kil, H. Lee, S.H. Kim. Analysis of Blind Frame Recognition and Synchronization Based on Sync Word Periodicity. *IEEE Access*. **8**,147516-147532 (2020)
2. S. Dan, X. Xiaojian, W. Jing. A novel presence detector for burst signals based on the fluctuation of the correlation function. *IEEE 10th INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING PROCEEDINGS*. 1628-1631(2010)
3. H. Zhong, T. Zhang. Block-LDPC: a practical LDPC coding system design approach. *Circuits and Systems I: Regular Papers, IEEE Transactions on*. **52(4)**, 766-775 (2005)
4. J. Wang, S. Yuan, Y. Zhou, G. Zhang. Codec Implementation of QC-LDPC Code in CCSDS Near-Earth Standard. *2020 5th International Conference on Computer and Communication Systems (ICCCS)*. 575-579 (2020)
5. Y. Yu, Z. Jia, W. Tao. LDPC codes optimization for differential encoded LDPC coded systems with multiple symbol differential detection. *IEEE Global Conference on Consumer Electronics*. 1-2 (2016)
6. T. Etzion, A. Trachtenberg, A.Vardy. Which codes have cycle-free Tanner graphs. *IEEE Transactions on Information Theory*. **45(6)**, 2173-2181 (1999)
7. T.J. Richardson, R.L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*. **47(2)**, 599-618 (2001)
8. T.J. Richardson, M.A. Shokrollahi, R.L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory*. **47(2)**, 619-637 (2001)
9. S.Y. Chung, J.D.F.Jr, T.J. Richardson. On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit. *IEEE Communications Letters*. **5(2)**, 58-60 (2001)
10. F. Vatta, A. Soranzo, M. Comisso, G. Buttazzoni, F. Babich. New explicitly invertible approximation of the function involved in LDPC codes density evolution analysis using a Gaussian approximation. *Electronics Letters*. **55(22)**, 1183-1186 (2019)
11. D.J.C. MacKay, S.T. Wilson, M.C. Davey. Comparison of constructions of irregular Gallager codes. *IEEE Transactions on Communications*. **47(10)**, 1449-1454 (1999)
12. S.Y. Chung, T.J. Richardson, R.L. Urbanke. Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation. *IEEE Transactions on Information Theory*. **47(2)**, 657-670 (2001)
13. C.L. Ho. Analysis of threshold of regular and irregular LDPC codes using Gaussian approximation. *2009 Wireless Telecommunications Symposium*. 1-7 (2009)
14. G. Zhang, Y. Xu, D.He, J. Sun, W. Zhang. Performance Analysis of LDPC-BICM System Based on Gaussian Approximation. *IEEE Transactions on Broadcasting*. March. **65(1)**, 166-171 (2019)