sciendo

Transport and Telecommunication, 2024, volume 25, no. 3, 300-310
Transport and Telecommunication Institute, Lauvas 2, Riga, LV-1019, Latvia
DOI 10.2478/ttj-2024-0022

ALGORITHMIZATION AND HARDWARE IMPLEMENTATION OF POLAR CODING FOR 5G TELECOMMUNICATIONS

Ilya Pyatin[1], Juliy Boiko[2], Oleksander Eromenko[3]

[1]Department of Computer Engineering, Faculty of Computer Engineering, Khmelnytskyi Polytechnic Professional College by Lviv Polytechnic National University
10, Zarichanska str., Khmelnytskyi, 29019, Ukraine
ilkhmel@ukr.net
[2]Department of Telecommunications, Media and Intelligent Technologies, Faculty of Information Technologies, Khmelnytskyi National University
11, Instytuts'ka str., Khmelnytskyi, 29016, Ukraine
boiko_julius@ukr.net
[3]Department of Physics and Electrical Engineering, Faculty of Information Technologies, Khmelnytskyi National University
11, Instytuts'ka str., Khmelnytskyi, 29016, Ukraine
yeromenko_s@ukr.net

The article explores a recursive algorithm for determining Bhattacharyya parameters, which is a measure of the degree of channel polarization for telecommunications with polar coding. A method is given for determining the positions of information and fixed bits in a polar code. The principles of constructing a polar encoder and a polar decoder successive cancellation are considered. The Field-Programmable Gate Array implementation of the successive cancellation list decoder is considered. Experimental studies of the construction of a polar code using Gaussian approximation have been carried out. The influence of the design signal-to-noise ratio on the bit error rate of telecommunications with polar codes has been studied. The use of multi-position modulation in telecommunications with polar codes has been studied. It is expected that the results obtained will be useful in hardware and software implementation of 5G telecommunications with polar coding.

Keywords: 5G, polar code; FPGA; telecommunication system; decoder

1. Introduction

The conceptual innovation of the 5G NR 15 standard was the description of the directive to change the physical layer channel coding formats. The main emphasis was placed on the use of low-density parity check (LDPC) codes (Bae *et al.* 2019; Boiko *et al.,* 2024) and polar codes (PC) (Su *et al.,* 2022). The main goal of this innovation concerned a radical replacement of convolutional and turbo codes that make up the essence of channel coding in 4G technology (LTE - Long-Term Evolution).

In general, new scenarios for 5G implementation have been developed within the framework of the new concept. In particular, technology adoption should be directed towards the deployment of Enhanced Mobile Broadband (eMBB); Ultra Reliable Low Latency Communications (URLLC) and Massive Machine Type Communications (mMTC) (Ghosh *et al.,* 2019). Consequently, such a wide range of scenarios imposes high demands on ensuring throughput, minimizing latency, and increasing reliability indicators compared to the previous generation.

In connection with the listed capabilities, the main requirements for 5G channel codes can be clearly identified. Among such requirements, it is necessary to highlight their ability to work with changes in rate and code length, support for setting up a hybrid automatic repeat request (HARQ). Moreover, these capabilities should concern both control information and user data. That is why the implementation of new channel coding schemes, especially in complex conditions caused by noise, interference, low signal level (Pyatin *et al.,* 2023) in order to localize and correct transmission errors, is a very important and urgent scientific task.

A significant part of the available literature does not take into account the specifics of codes for 5G scenarios, as well as the process of encoding them, taking into account their further widespread use in the design of encoding circuits using Field-Programmable Gate Array (FPGA).

In article (Yuan and Parhi, 2014), the authors propose a new SCL decoding algorithm (LLR-SCL) - Successive cancellation list based on the log-likelihood ratio (LLR). Work (Ochiai *et al.,* 2021) focuses on

the investigation of the improved Gaussian approximation technique, assessing the asymptotic behavior of the function in order to track the evolution of the LLR. Important issues are discussed in the article (Bocherer *et al.*, 2017). The authors, by introducing the apparatus of information theory, evaluate the performance of PC based on the concept of sequential inverse transformation. The paper presents the results of a study of the performance of PC inverse mapping devices. Reliable scenarios for assessing the performance of PC based on bit error rate monitoring are given in (Benkhouya *et al.*, 2020). The authors simulate various scenarios for changing the length of a code block and their performance. Research on a PC concatenation scheme to reduce burst errors is discussed in (Jin *et al.*, 2020). New algorithms for constructing PC are described in (Tahir and Rupp, 2017; Qin *et al.*, 2017). To implement and simulate polar encoders in (Parmar and Jain, 2021), the authors suggest using Xilinx ISE 14.7 software. Important performance improvements of the proposed code are presented. Certain aspects of FPGA implementation using Xilinx Virtex VI for implementing advanced encoding and decoding schemes for PC are reflected in (Oommen and Ravishankar, 2015). The capabilities of FPGAs in the standard Intel DE10 development kit for developing PC are described in (Krasser *et al.*, 2021). The authors focus on describing a deployed encoder and decoder architecture to achieve high performance, increased throughput for short PC. Current issues of implementing PC are also discussed in the works (Boiko *et al.*, 2020; Chiu, 2022).

The proposed work is intended to complement the cutting-edge research discussed in the review of the authors' works above. We focused on important issues that have been overlooked by cutting-edge research regarding the implementation of polar encoding/decoding. A study of a recursive algorithm for determining the Bhattacharyya parameters (PB) for PC is presented. The algorithmic apparatus for identifying bit positioning in the PC structure and the hardware for implementing the SCL decoder are considered. The results of an experimental study of the performance of PC in channels with multi-position modulation are presented. Based on the research, specific recommendations are given to developers of 5G PC schemes regarding the selection of parameters (rate, code length, addition of redundant code (CRC), etc.) of signal-code construction (SCC) to improve the efficiency of telecommunication channels.

2. Polar Code Concept

2.1. Algorithmic aspects of constructing polar codes

The analysis of articles and works of well-known authors carried out above allows us to assert that PCs are used to increase noise immunity in wireless telecommunications, correct errors when working with solid-state drives, increase the reliability of data transmission in satellite communications, optical communication systems, the Internet of things and other applications.

Thus, the process of constructing a code directly precedes the encoding of information bits with PC. This process is based on the technique of selecting positions for placing the information and frozen bit set when forming a code word. We will consider a synthesized Simulink model of a telecommunications system from a PC, Figure 1.

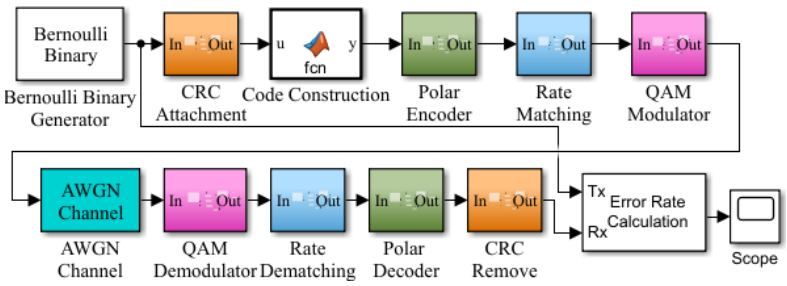


Figure 1. Simulink–model of a telecommunication system with polar codes

The PC is constructed using the algorithm in Figure 2. Taking into account the code rate (R) and frame length (M), the number of information bits and CRC (K) is determined; the number of punctured bits P is calculated; frozen bits F and fixed positions U according to the algorithm in Figure 2. Each bit of the PC word is transmitted over a separate synthesized channel. Such channels should be sorted by reliability. To do this, the PB is calculated using the algorithm in Figure 3.

In Figure 2 we used the following notation: K is the number of information bits and CRC, $K=N*R$; M is the frame size (block length); N is the number of encoded bits at the output of the PC encoder; F is the

number of frozen bits *F=N-K*; *P* is the positions of punctured bits in the vector: 1:*N-M*; *U* is the positions of frozen bits in the vector: 1:*N-K*; Z is the PB parameters. To determine the upper bound on the error probability of a maximum likelihood solution for a given channel, we used PB.

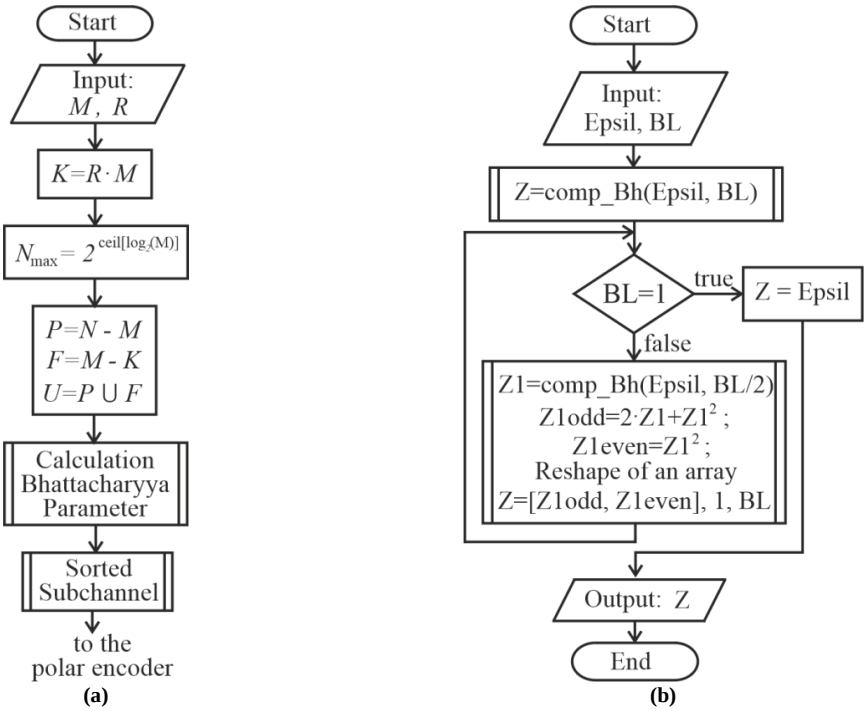


Figure 2. Algorithm: (a) for constructing a polar code, (b) calculating the Bhattacharyya parameter

$$Z(W) = \sum_{y \in Y} \sqrt{W(y|0)W(y|1)}\ , \tag{1}$$

where $W(y|0)$, $W(y|1)$ is the probability of the value 0 or 1 at the output (y) of the synthesized channel W.

Note that we examined PB in the case of a binary channel with erasure (BEC) (Boiko *et al.*, 2021) and took into account the erasure probability at level ε = 0.35. Thus, based on the recursive algorithm (Figure 3), each bit contained in the codeword forms its own synthesized transmission channel with its own error probability (1).

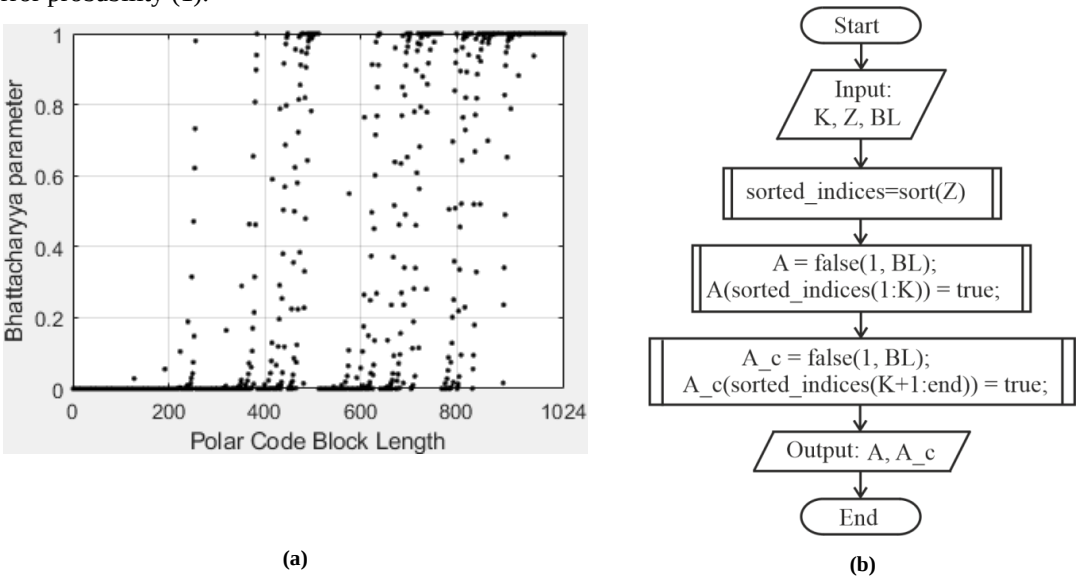


Figure 3. Graphic interpretation of parameters Bhattacharyya: (a) for polar code *M=N*=1024, (b) algorithmization of the subchannel selection process (the positions of the information bits A_c and the frozen bits is A were taken into account)

In Figure 3, we presented a recursive method for determining PB. The next step is to implement sorting of subchannels according to reliability criteria (Figure 3b). Here the process of identifying the corresponding informative and fixed bits takes place, which are then used as input data for the operation of the PC.

2.2. Specifics of polar encoding and decoding

We denoted as G_N the generator matrix of the PC, which is formed in accordance with the expressions:

$$G_N = B_N F^{\otimes n};\ F = \begin{vmatrix} 1 & 0 \\ 1 & 1 \end{vmatrix}, \tag{2}$$

where $F^{\otimes n}$ is the n-th power of the Kronecker matrix F, $n = \log_2 N$, B_N is the is a bit-reversal permutation matrix of size $M \times N$.

Figure 4 shows a graphical interpretation of the Kronecker matrix.

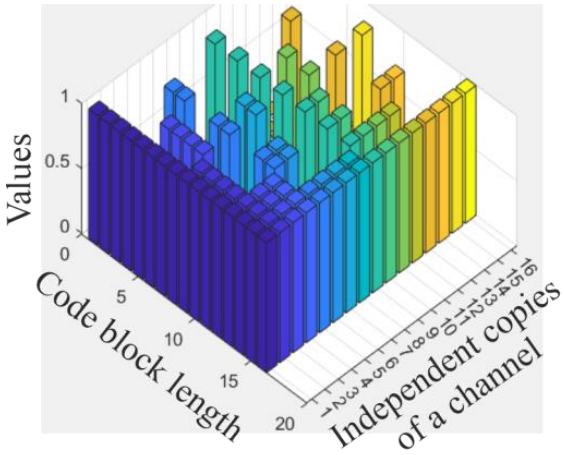


Figure 4. Graphical representation of the Kronecker matrix for $n = 4$

The output signal $x = (x_0,\ x_1,\ \ldots,\ x_{N-1})$ of the polar encoder is given by:

$$x = uG_N, \tag{3}$$

where $u = \{u_1, u_2, \ldots u_N\}$ is the input signal generated at the code construction stage.

We presented the signal received through the telecommunications channel in the following form:

$$r(t) = G_a \sum_k a(k)\, p(t - kT_s - \tau) + w(t), \tag{4}$$

where $a(k)$ is the k-th received signal symbol; T_s is the period of arrival of symbols; τ is the delay; $p(t)$ is the shape of a unit energy pulse; G_a is the totality of all amplitude gains and losses accompanying the signal on the path from the transmitter to the receiver; $w(t)$ is the additive white Gaussian noise (AWGN).

The resulting symbol y_i will be determined by a form of the form: $y_i = s_i + z_i$ where z_i is AWGN with zero mean and level variance $N_0/2$.

On the receiver side we used successive cancellation (SC) decoding. Given a sequence of input symbols y_n, each input bit u_i ,i=0,1,…,N-1 is successive decoded based on its corresponding LLR:

$$L_n^{(i)}(y_n) = \begin{cases} \lg \dfrac{p(y_n \mid u_i = 0)}{p(y_n \mid u_i = 1)}, & i = 0 \\ \lg \dfrac{p(y_n, \hat{u}_{i-1} \mid u_i = 0)}{p(y_n, \hat{u}_{i-1} \mid u_i = 1)}, & i = 1, 2, \mathrm{K}, 2^n - 1 \end{cases}, \tag{5}$$

where $p(\,|\,)$ denotes the conditional probability density function, and $\hat{u}_{i-1} = (\hat{u}_0,\ \hat{u}_1,\ \ldots,\ \hat{u}_{i-1})$ is a vector containing the estimated bits that were already determined when decoding the i-th bit; the subscript n on

$L_n^{(i)}$ reflects the fact that the LLR is calculated based on observing 2^n the received symbols in addition to the previously estimated input bits.

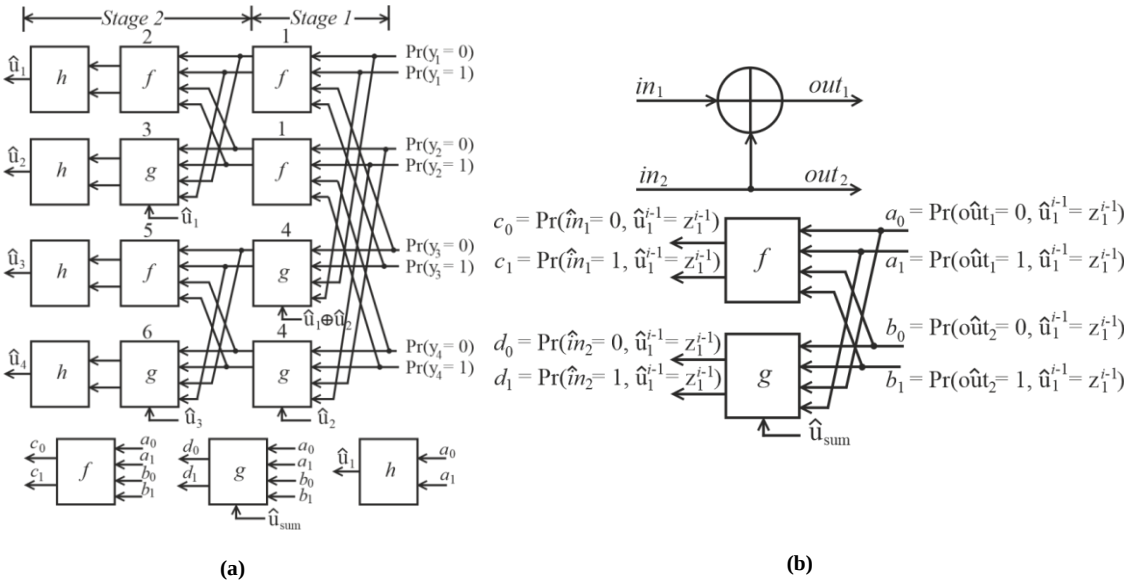
Note that at the receiver input the signal has a Gaussian distribution. We work with LLR and use the Gaussian approximation to simplify calculations.

In SC decoding, received symbols are sequentially decoded based on the corresponding LLRs. LLRs are assumed to have a Gaussian distribution with mean γ and variance 2γ. For the AWGN channel, the LLRs are given by the expression:

$$L(y_i) = \log \frac{p(y_i \mid x_i = 0)}{p(y_i \mid x_i = 1)} = 4 \frac{\sqrt{E_s}}{N_0} y_i, \; i = 0, 1, \text{K}, \; N-1. \tag{6}$$

For PC with bit rate *R=K/N*, the *K* channel indices are selected from the *N* indices. We can argue that Gaussian approximation allows tracking the average values of LLRs instead of their exact values, which improves performance and optimizes FPGA resources.

In Figure 5a shows an SC decoding circuit with $n = 4$. Here, the SC decoder consists of $\log_2 n = 2$ stages, where each stage consists of main computing units *f* and *g* with 4 inputs and 2 outputs. At the end of the last stage, a hard decision block (*h*) is used to determine the decoded bit $\hat{u}_i$. In Figure 7, block *f* or *g* is characterized by its activation time. In Figure 5b basic blocks of polar encoding and SC decoding, respectively.



(a) **(b)**

Figure 5. The block diagram: (a) of the SC decoder, (b) basic block of polar encoding and decoding

Accordingly, the block functions *f* and *g*, as the basic computational units of the polar decoder, represent the estimates from right to left from $o\hat{u}t_1$ and $o\hat{u}t_2$ to $\hat{i}n_1$ and $\hat{i}n_2$, where $\hat{i}n_1$, $\hat{i}n_2$, $o\hat{u}t_1$, $o\hat{u}t_2$ denote the estimates in_1, in_2, out_1 and out_2, respectively. If we assume that the previous decoded bits $\hat{u}_1, \; \hat{u}_2, \text{K}, \hat{u}_{i-1}$ are equal to $\hat{z}_1, \; \hat{z}_2, \text{K}, \hat{z}_{i-1}$, and denote this event as $\hat{u}_1^{i-1} = z_1^{i-1}$, then we can derive the functions *f* and *g* of the modules like so:

$$c_0 = \Pr\left(\hat{i}n_1 = 0, \hat{u}_1^{i-1} = z_1^{i-1}\right) = a_0 b_0 + a_1 b_1, \tag{7}$$

$$c_1 = \Pr\left(\hat{i}n_1 = 1, \hat{u}_1^{i-1} = z_1^{i-1}\right) = a_0 b_1 + a_1 b_0, \tag{8}$$

$$d_0 = \Pr\left(\hat{i}n_2 = 0, \hat{i}n_1 = \hat{u}_{sum}, \hat{u}_1^{i-1} = z_1^{i-1}\right) = a_{\hat{u}_{sum}} b_0, \tag{9}$$

$$d_1 = \Pr\left(\hat{i}n_2 = 1, \hat{i}n_1 = \hat{u}_{sum}, \hat{u}_1^{i-1} = z_1^{i-1}\right) = a_{1-\hat{u}_{sum}} b_1, \tag{10}$$

where $a_0 = \Pr\left(o\hat{u}t_1 = 0, u_1^{i-1} = z_1^{i-1}\right)$; $a_1 = \Pr\left(o\hat{u}t_1 = 1, u_1^{i-1} = z_1^{i-1}\right)$; $b_0 = \Pr\left(o\hat{u}t_2 = 0, u_1^{i-1} = z_1^{i-1}\right)$; $b_1 = \Pr\left(o\hat{u}t_2 = 1, u_1^{i-1} = z_1^{i-1}\right)$ is the terms describing the four inputs of blocks *f* and *g*.

Block *h* in Figure 4 describes making a hard decision about the received bit using the expression:

$$u_1 = \begin{cases} 0, & a_0 \geq a_1 \text{ or } \hat{u}_i \text{ is frozen bit} \\ 1, & a_0 < a_1 \text{ and } \hat{u}_i \text{ is free bit} \end{cases}. \quad (11)$$

Note that, using the path metric, the SC decoder performs a locally optimal search strategy to find the path of length *n* with the largest metric. At each layer, the SC decoder first processes two child nodes associated with the current survival path. By comparing the corresponding path metrics, the SC decoder selects the path with higher metrics as the updated survival path (Yu *et al.*, 2018).

The determination of $\hat{u}_i$ in the SCL algorithm is done by comparing the metrics of all candidate paths. We used formulas (7-10) and represented the incoming messages in LLR form:

$$c = 2arctgh\left(tgh\left(\frac{a}{2}\right)tgh\left(\frac{b}{2}\right)\right) \approx sign(a)sign(b)\min\left(|a|,|b|\right), \quad (12)$$

$$d = a(-1)^{u_{sum}} + b, \quad (13)$$

where $c = \ln\left(c_0/c_1\right)$, $d = \ln\left(d_0/d_1\right)$, $a = \ln\left(a_0/a_1\right)$ and $b = \ln\left(b_0/b_1\right)$.

Determination of the SLC path metric is performed in an atomic format by blocks *f* and *g* using the metric calculation block (MCU).

The function of the MCU is to compute the path metrics $M_{i,0}$ and $M_{i,1}$ (corresponding to the path metrics for $\hat{u}_i = 0$ and $\hat{u}_i = 1$) with the known output *c* or *d* from block *f* or *g* in the last step. The format for block *f* is:

$$c = \ln\frac{c_0}{c_1} = \ln\frac{\Pr\left(\hat{u}_i = 0, \hat{u}_1^{i-1} = z_1^{i-1}\right)}{\Pr\left(\hat{u}_i = 1, \hat{u}_1^{i-1} = z_1^{i-1}\right)} = M_{i,0} - M_{i,1}. \quad (14)$$

The logarithmic domain metric $M_{i-1,z_{i-1}}$ for a path of length (*i*-1) has the form:

$$M_{i-1,z_{i-1}} = \ln\left(\Pr\left(\hat{u}_{i-1} = z_{i-1}, \hat{u}_1^{i-2} = z_1^{i-2}\right)\right) = \ln\left(e^{M_{i,0}} + e^{M_{i,1}}\right). \quad (15)$$

Taking into account mathematical models (13) and (14), we write the following formulas:

$$M_{i,0} = M_{i-1,z_{i-1}} + c - \ln\left(1 + e^{c}\right), \quad (16)$$

$$M_{i,1} = M_{i-1,z_{i-1}} - \ln\left(1 + e^{c}\right). \quad (17)$$

Thus, from a message in LLR format with *c* or *d*, based on the calculated metric of the path of length (*i*-1) $M_{i-1,z_{i-1}}$, the metric - *i* (16), (17) is calculated.

2.3. Hardware architecture of the SCL decoder

We present the general view of the architecture of the SCL decoder, taking into account the size of the list *L*, in Figure 5. The design of such a decoder is formed by the *L*-th set of SC decoders. We showed above that after SC decoders calculate *c* and *d* messages in LLR form, including previous path metrics, within block *L* the MCUs are calculated to form 2*L* candidate path metrics. After sorting the best *L* results among the number of 2*L* candidates, *L* – survival paths are formed.

The design of the SCL decoder also includes components of a memory bank. The first bank contains messages in the form of LLR, which are then used in a number of *L*-SC decoders. The next two banks form a bank of metrics and paths of survival and are the source of updating the *L*-th number of paths of survival and the corresponding metrics of paths.

From Figure 6a, in particular, it follows that the component structure of the SC decoder is characterized by a number of processing elements (PE). Accordingly, each PE contains blocks of type *f* and *g* in LLR format (formulas (12) and (13)). Then we present the architecture of the *q*-bit PE in Figure 6b. The C2S and S2C blocks used perform the role of a complementary code and sign-magnitude form converter.

Now let's pay attention to formulas (16) and (17). In Figure 7a, we used a form of the form inputLLR to describe the result as the decoded bit $\hat{u}_i$ to highlight the possibility of inputting *c* and *d* into the MCU. Thus, based on (16), (17), a *q*-bit MCU design is formed.

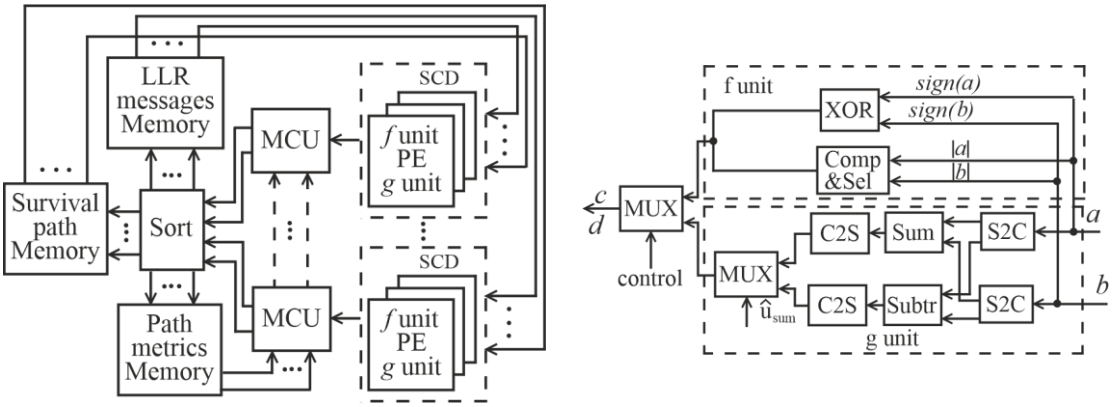


Figure 6. Architectural design: (a) LLR-SCL decoder, (b) PE based on a *q*-bit LLR

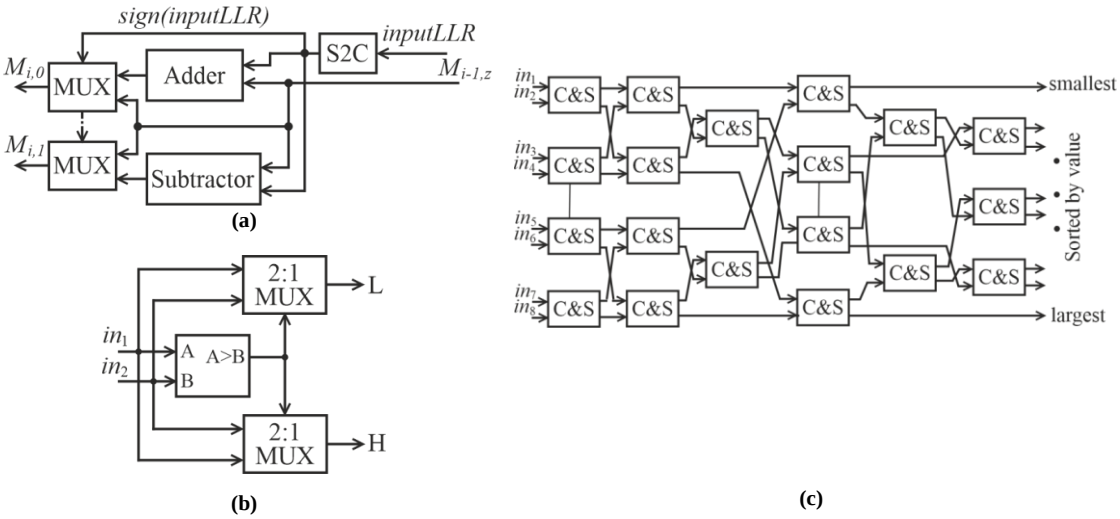


Figure 7. Architectural design: (a) LLR-SCL decoder, (b) PE based on a *q*-bit LLR, (c) 8-input sort format

In Figure 7b we present the structure of the C&S block, which performs the functions of comparison and replacement. Such a block forms a sorting network of size 2. This block has two inputs, compares them and swaps them if $in_1 > in_2$. Then in Figure 7c we can imagine the architectural concept of sorting which is formed by 8 inputs. The main content of such a network is the implementation of a parallel processing algorithm, which is so important for hardware implementation forms. Here, procedures for merging odd and even arrays are performed and, ultimately, a path *L* with large metrics is outlined by comparing metrics.

3. Experimental Results

The Simulink model of the telecommunication system under study is shown in Figure 1. First, the PC is constructed, the variables are defined: frame length *M*; number of information bits and CRC(*R*); the size of the code word at the output of the polar encoder is calculated in the form of the result of the operation

$N = 2^{\log_2(M)}$ rounded up to an integer; the presence of a bit reversal operation at the output of the polar encoder is determined; the design signal-to-noise ratio (SNRdes) is determined.

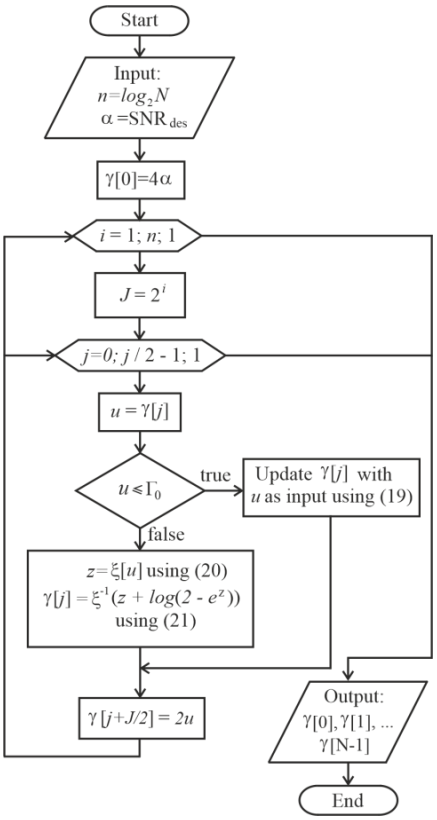
Next, the channel polarization procedure is performed according to the algorithm shown in Figure 8. To construct a PC, we will use the Gaussian approximation, which allows us to track the average value of the LLR, which can be represented as (Ochiai *et al.*, 2021):

$$\gamma_k^{(2i)} = \xi^{-1}\left(\xi(\gamma_{k-1}^{(i)}) + \log_2\left(2 - e^{\xi(\gamma_{k-1}^{(i)})}\right)\right). \tag{18}$$

For the practical use of Gaussian approximation, empirical expressions (19), (20) (Ochiai *et al.*, 2021) were obtained, which make it possible to represent the LLR of the receiver signal at different intervals by the corresponding nonlinear functions. Carrying out Gaussian approximation can be presented in the form of an algorithm (Figure 8).

$$\Xi(\gamma) = \frac{1}{2}\gamma^2 - \frac{1}{2}\gamma^3 + \frac{2}{3}\gamma^4,\ \gamma \le \Gamma_0, \tag{19}$$

$$\xi(z) = \begin{cases} -\frac{1}{2}\gamma + \frac{1}{8}\gamma^2 - \frac{1}{8}\gamma^3, & \gamma \le \Gamma_0 \\ a_0 + a_1\gamma + a_2\gamma^2, & \Gamma_0 < \gamma \le \Gamma_1 \\ a\gamma^c + b, & \Gamma_1 < \gamma < \Gamma_2 \\ \frac{2,29-\gamma}{4} - \frac{1}{2}\log_2\gamma + \log_2\left(1 - \frac{\pi^2}{4\gamma} + \frac{\kappa_0}{\gamma^2}\right), & \Gamma_2 \le \gamma \end{cases}. \tag{20}$$



$a_0 = $-0.002706; $a_1 = 0.476711$; $a_0 = 0.0512$
$a = $-0.4527; $b = 0.0218$; $c = 0.86$;
$\Gamma_0 = 0.2$; $\Gamma_1 = 0.7$; $\Gamma_2 = 10$; $\kappa_0 = 8.554$.
The inverse function of $z = \xi(\gamma)$, i.e. $\xi^{-1}(z)$, is determined by the expression:

$$\xi^{-1}(z) = \begin{cases} -2z + z^2 + z^3, & z \ge Z_0 \\ \frac{-a_1 - \sqrt{a_1^2 - 4a_2(a_0 - z)}}{2a_2}, & Z_1 \le z \le Z_0 \\ \left(\frac{z-b}{a}\right)^{\frac{1}{c}}, & Z_2 < z < Z_1 \end{cases} \tag{21}$$

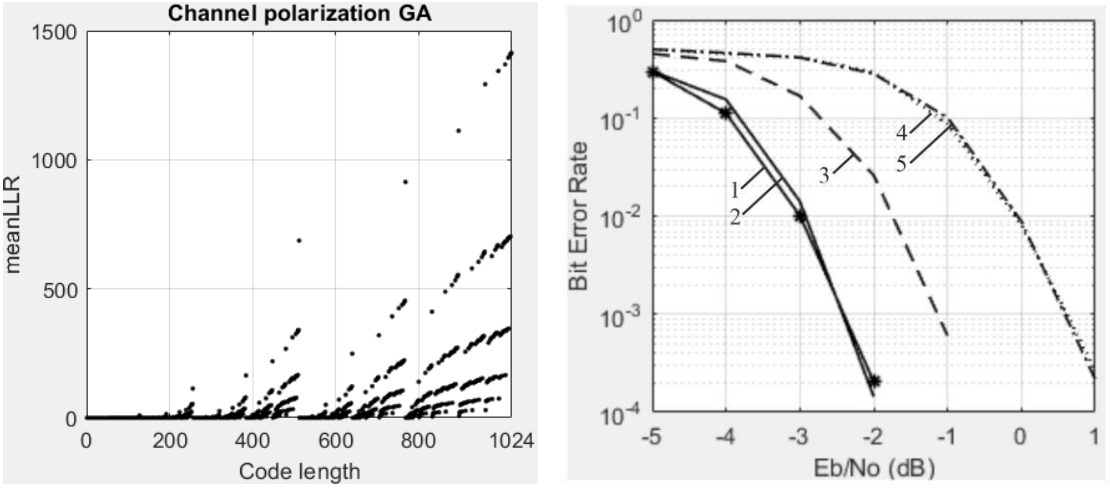
where Z_0, Z_1, Z_2 is the thresholds defining areas of use of the corresponding function.
$a_0 = $-0.002706; $a_1 = $-0.476711; $a_2 = 0.0512$;
$a = $-0.4527; $b = 0.0218$; $c = 0.86$;
$z_0 = \xi(\Gamma_0)$;
$z_1 = \xi(\Gamma_1)$;
$z_2 = \xi(\Gamma_2)$;

Figure 8. Gaussian approximation algorithm

The average LLR during PC development should range from very small to very large values, as shown in Figure 9a. Channel polarization involves calculating the LLR of the worst and best channels. In AWGN channels, due to changes in channel quality (the operating SNR changes), it is necessary to

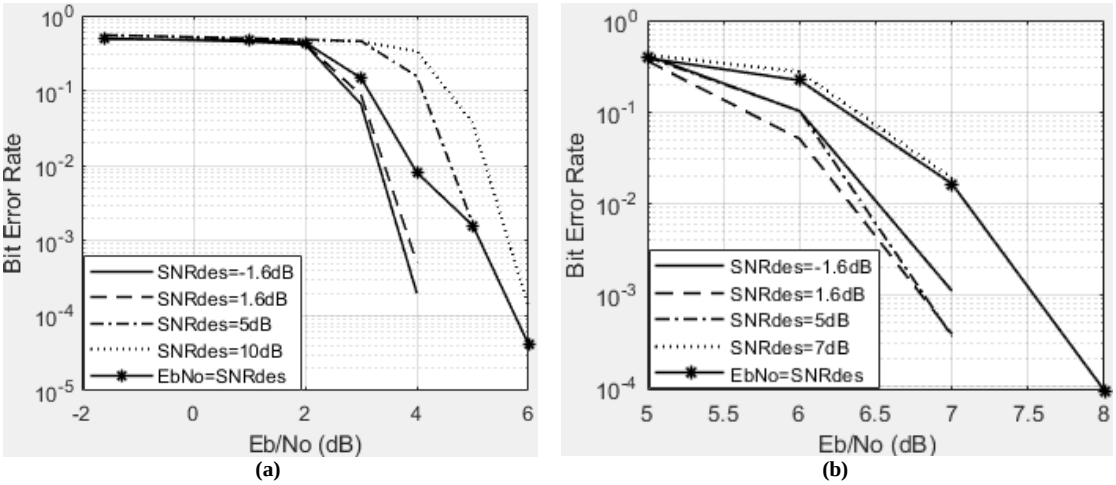
construct PCs to select optimal bit channels for each SNR value, which leads to the high complexity of constructing PC.

According to the Simulink model presented in Figure. 1, studies were carried out on the bit error rate (BER) of a telecommunication system with PC with different code rates. Different code rates and different design values of the SNRdes were used. For comparison, the bit error rate for the current signal-to-noise ratioSNR (Eb/N0), equal to the design one (SNRdes) (Figure 9b), is constructed. Such a graph was constructed on a continuous basis and the procedure for constructing a PC was repeated for each value of Eb/N0=SNRdes. Simulation parameters: QPSK modulation; M=N=1024; set encoder mode with bit reversal; type CRC="CRC16". From the obtained dependencies we can conclude that for low code rates it is preferable to choose the design SNR on the Shannon edge. For code rates greater than R=2/3, a higher design SNR value, determined by the desired bit error rate, becomes preferable. For example, for a code rate of R=8/9, QPSK modulation and a codeword length of N=1024 bits, the design SNR can be selected from 1.6 dB to 5 dB. SNRdes=-1.6dB gives a 0.25dB worse bit error rate. In Figure 10 shows the experimentally obtained noise immunity graphs for PC for the conditions (a) is the K = 683; R = 2/3; (b) - modulation is the QPSK; K = 911; R = 8/9; BitRev: 'yes'; CRCType: "CRC16"; decType: 'sscl'; L = 4. In Figure 11a shows the dependences of the error rate in the frame for a telecommunication system with a code rate of R = 8/9, and Figure 11b is the noise immunity graph for a telecommunication system (Zhurakovskyi *et al.*, 2020) with PC and 1024QAM modulation.



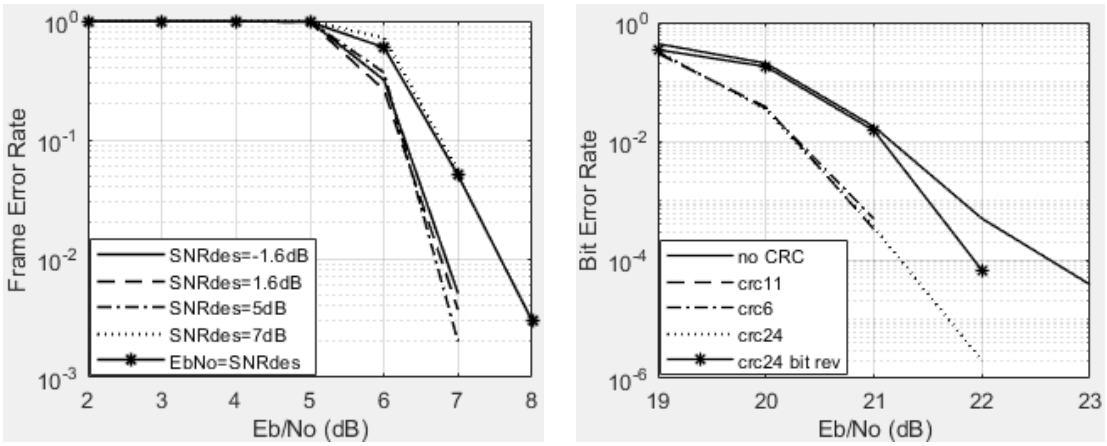
(a) **(b)**

Figure 9. The estimate of the average LLR value: (a) using the Gaussian approximation, (b) the noise immunity graph is for the conditions K = 205; R = 1/5; 1 is the Eb/N0 = SNRdes; 2 is the SNRdes = -1,6 dB; 3 is the SNRdes = 5 dB; 4 is the SNRdes = 10 dB; 5 is the SNRdes = 5 dB



(a) **(b)**

Figure 10. The noise immunity graph: (a) for the conditions K = 683; R = 2/3, (b) for the condition's modulation is the QPSK; K = 911; R = 8/9; BitRev: 'yes'; CRCType: "CRC16"; decType: 'sscl'; L = 4



(a) **(b)**

Figure 11. The noise immunity graph: (a) for the conditions the error rate in the frame K = 911; R = 8/9, (b) for the conditions modulation is the 1024QAM

4. Conclusions

Based on the results presented in the research, the article contains a description of the algorithmic principles of PC design, features of PC calculation, a technique for selecting subchannels for transmitting information and fixed bits, a technique for performing a Gaussian approximation of average LLR values. The principle of formation of the Kronecker matrix for n=4 has been studied. An FPGA implementation of a sequential elimination list polar decoder is presented, and the circuit implementations of some polar decoder blocks are characterized. The bit error rates of PC telecommunications have been studied. From the obtained dependencies we can conclude that for low code rates it is preferable to choose the design SNR on the Shannon boundary. For code rates greater than R=2/3, a larger design SNR value, determined by the desired bit error rate, becomes preferable. For example, for a code rate of R=8/9, QPSK modulation and a codeword length of N=1024 bits, the design SNR can be selected from 1.6 dB to 5 dB. SNRdes=-1.6dB gives a 0.25dB worse bit error rate. Increasing the modulation position from 1 to 10 bits/symbol requires an increase in SNR from 3 dB to 6 dB for every 2 bits/symbol. Adding CRC allows you to increase the noise immunity of the system by 1 dB. Bit reversal at the PC output degrades the bit error rate for high positionality communication systems.

References

1. Bae, J. H., Abotabl, A., Lin, H. P., Song, K. B., and Lee, J. (2019) An overview of channel coding for 5G NR cellular communications. *APSIPA transactions on signal and information processing*, 8, e17. https://doi.org/10.1017/ATSIP.2019.10.
2. Benkhouya, R., Idriss, C. and Youssef, H. (2020) Study of the operational SNR while constructing polar codes. *International Journal of Electrical and Computer Engineering*, 10(3), 3200-3207. http://doi.org/10.11591/ijece.v10i3.pp3200-3207.
3. Bocherer, G., Prinz, T., Yuan, P. and Steiner, F. (2017) Efficient Polar Code Construction for Higher-Order Modulation. In: *Proceedings of the 2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, San Francisco, CA, USA, 19-22 March 2017, 1-6. IEEE Press. http://doi.org/10.1109/WCNCW.2017.7919039.
4. Boiko, J., Pyatin, I., Eromenko, O., and Karpova, L. (2024) Evaluation of the Capabilities of LDPC Codes for Network Applications in the 802.11ax Standard. *IoT Based Control Networks and Intelligent Systems. Lecture Notes in Networks and Systems*, 789. Singapore: Springer. https://doi.org/10.1007/978-981-99-6586-1_25.
5. Boiko, J., Pyatin, I., and Eromenko, O. (2020) Simulation of the Transport Channel with Polar Codes for the 5G Mobile Communication. In: *Proceedings of the 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T)*, Kharkiv, 06-09 October 2020, 182-186. IEEE Press. http://doi.org/10.1109/PICST51311.2020.9468013.
6. Boiko, J., Pyatin, I., and Eromenko, O. (2021) Design and evaluation of the efficiency of channel coding LDPC codes for 5G information technology. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, 9(4), 867-879. https://doi.org/10.52549/ijeei.v9i4.3188.

7. Chiu, M.-C. (2022) Analysis and Design of Polar-Coded Modulation. *IEEE Transactions on Communications*. 70(3), 1508-1521. https://doi.org/10.1109/TCOMM.2022.3142280.
8. Ghosh, A., Maeder, A., Baker M., and Chandramouli, D. (2019) 5G Evolution: A View on 5G Cellular Technology Beyond 3GPP Release 15. *IEEE Access*, 7, 127639-127651. http://doi.org/10.1109/ACCESS.2019.2939938.
9. Jin, J., Deng, R., Liu, T., Li, L. (2020) On Error Performance and Concatenated Coding of Polar Codes in AWGN Channels. *IOP Conference Series: Materials Science and Engineering*, 768, 072075. http://doi.org/10.1088/1757-899X/768/7/072075.
10. Krasser, F. G., Liberatori, M. C., Coppolillo, L., Arnone, L., and Moreira, J. C. (2021) Fast and efficient FPGA implementation of Polar Codes and SoC test bench. *Microprocessors and Microsystems*, 84, 104264. https://doi.org/10.1016/j.micpro.2021.104264.
11. Ochiai, H., Mitran, P. and Vincent Poor, H. (2021) Capacity-Approaching Polar Codes with Long Codewords and Successive Cancellation Decoding Based on Improved Gaussian Approximation. *IEEE Transactions on Communications*, 69(1), 31-43. http://doi.org/10.1109/TCOMM.2020.3030057.
12. Oommen, M.S., and Ravishankar, S. (2015) FPGA implementation of an advanced encoding and decoding architecture of polar codes. In: *Proceedings of the 2015 International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA)*, Bengaluru, 08-10 January 2015, 1-6. IEEE Press. http://doi.org/10.1109/VLSI-SATA.2015.7050456.
13. Pyatin, I., Boiko, J., Eromenko, O., and Parkhomey, I. (2023) Implementation and analysis of 5G network identification operations at low signal-to-noise ratio. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 21(3), 496-505. http://doi.org/10.12928/telkomnika.v21i3.22893.
14. Parmar, R.N., and Jain, A. (2021) A Verilog Implementation and Performance Analysis of Polar Encoder and Decoder for Next Generation Communication Application. *Journal of Emerging Technologies and Innovative Research*, 8(2), 1751-1756. https://www.jetir.org/view?paper=JETIR2102212.
15. Qin, M., Guo, J., Bhatia, A., Fàbregas, G. A. and Siegel, P. H. (2017) Polar Code Constructions Based on LLR Evolution. *IEEE Communications Letters*, 21(6), 1221-1224, June 2017, http://doi.org/10.1109/LCOMM.2017.2656126.
16. Su, B.-Sh., Lee, Ch.-H., Chiueh, T.-D. (2022) A 58.6/91.3 pJ/b Dual-Mode Belief-Propagation Decoder for LDPC and Polar Codes in the 5G Communications Standard. *IEEE Solid-State Circuits Letters*, 5, 98-101. https://doi.org/10.1109/LSSC.2022.3167423.
17. Tahir, B. and Rupp, M. (2017) New construction and performance analysis of Polar codes over AWGN channels. In: *Proceedings of the 2017 24th International Conference on Telecommunications (ICT)*, Limassol, Cyprus, 03-05 May 2017, 1-4. IEEE Press. http://doi.org/10.1109/ICT.2017.7998250.
18. Yuan, B. and Parhi, K. K. (2014) Successive cancellation list polar decoder using log-likelihood ratios. In: *Proceedings of the 2014 48th Asilomar Conference on Signals, Systems and Computers, Pacific Grove*, CA, USA, 02-05 November 2014, 548-552. IEEE Press. http://doi.org/10.1109/ACSSC.2014.7094505.
19. Yu, Q., Shi, Z., Li, X., Du, J., Zhang, J. and Rabie, K. M. (2018) On the Concatenations of Polar Codes and Non-Binary LDPC Codes. *IEEE Access*, 6, 65088-65097 https://doi.org/10.1109/ACCESS.2018.2877178.
20. Zhurakovskyi, B., Boiko, J., Druzhynin, V., Zeniv, I., and Eromenko, O. (2020). Increasing the efficiency of information transmission in communication channels. *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, 19(3), 1306-1315. http://doi.org/10.11591/ijeecs.v19.i3.pp1306-1315.