Asesorías y Tutorías para la Investigación Científica en la Educación Puig-Salabarría S.C.
José María Pino Suárez 400-2 esq a Lerdo de Tejada. Toluca, Estado de México. 7223898475
RFC: ATI120618V12

Revista Dilemas Contemporáneos: Educación, Política y Valores.
http://www.dilemascontemporaneoseducacionpoliticayvalores.com/
Año: VI Número: Edición Especial Artículo no.:55 Período: Marzo, 2019.

TÍTULO: Un enfoque lexicográfico para la decodificación de códigos polares.

AUTORES:

1. Nikolay Yu. Chilikhin.
2. Dmitrii V. Ganin.
3. Roman V. Klimov.
4. Aleksey A. Shamin.
5. Aleksey B. Kuzmichev.

RESUMEN. Los sistemas de información modernos requieren no solo el desarrollo de nuevos métodos y algoritmos para descodificar códigos redundantes existentes, sino también la creación de nuevas clases de códigos de corrección de errores. Este artículo examina un enfoque lexicográfico (incluidas sus modificaciones) para la decodificación de códigos polares. Los resultados muestran que se ha llevado a cabo una comparación de la ganancia de energía entre el enfoque lexicográfico propuesto para la decodificación de códigos polares y los esquemas de decodificación clásicos. El enfoque propuesto mostró los mejores resultados en términos de capacidades de corrección en el rango de bajas relaciones de señal a ruido.

PALABRAS CLAVES: grupo base, decodificación lexicográfica, métrica de Hamming, decodificación de decisión suave, decodificación suave, número de grupo.

TITLE: A lexicographic approach to the decoding of Polar codes.

AUTHORS:

1. Nikolay Yu. Chilikhin.

2. Dmitrii V. Ganin.

3. Roman V. Klimov.

4. Aleksey A. Shamin.

5. Aleksey B. Kuzmichev.

ABSTRACT: Modern information communication systems require not only the development of new methods and algorithms for decoding existing redundant codes, but also the creation of new classes of error-correcting codes. This article examines a lexicographic approach (including its modifications) to the decoding of polar codes. Results shows that a comparison of the energy gain between the proposed lexicographical approach to decoding polar codes and the classical decoding schemes have been carried out. The proposed approach showed the best results in terms of correction capabilities in the range of low signal-to-noise ratios.

KEY WORDS: base cluster, lexicographic decoding, Hamming metric, soft decision decoding, soft decoding.

INTRODUCTION.

Reasonably increasing requirements for management efficiency of modern and prospective management control systems (MCS) dictate the need for short-cycle management. Therefore, to protect information against errors in forward and reverse links of a MCS, it is expedient to use short error-correcting codes. Reducing the length of code sequences for given reliability requirements leads to the need for flexible synthesis of code and algorithmic methods for error correction based on applying soft decoding methods in combination with iterative data conversion. To a certain extent, this requirement is met by relatively new designs based on polar codes (PC). Application of the PC concept is stipulated by the following positive features of this class of block codes (Arikan, 2009; Arikan & Telatar, 2009; Gladkikh et al, 2016):

1. Reaching an asymptotically possible capacity for the binary channel;
2. An opportunity to freely select the required code distance within the Hamming metric due to the Bhattacharyya distance (BhD);
3. An acceptable level of error-correcting capability.

The concept of forming a PC is based on Arikan's kernel, which is a matrix $\boldsymbol{F} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$. $\boldsymbol{F}^{\otimes m}$ defines its m-th Kronecker index, where $m \in N$. To obtain the corresponding matrix, it is necessary to introduce a permutation matrix $\boldsymbol{B}_N$ based on BhD, which is defined as:

$$Z_{2 \cdot i+1, j} = \begin{cases} 2 \cdot Z_{i,\chi} - Z_{i,\chi}^2 & \text{для } j_{0,e}^{N-1} \wedge 0 \le \chi \le i \\ Z_{i,\chi}^2 & \text{для } j_{0,o}^{N-1} \wedge 0 \le \chi \le i \end{cases}$$

where $i = N/2 - 1$, $j = \{0, 1, 2, ..., N-1\}$, $j_{0,e}^{N-1}$, and $j_{0,o}^{N-1}$ are the components of the set with only even and odd numbers respectively, starting from zero, and $N = 2^m$ is the length of code combination. It is worth noting that the sets $\left| j_{0,e}^{N-1} \right| = \left| j_{0,o}^{N-1} \right| = \left| \chi \right|$ are equipotent. To obtain the system of equations (1) for different values of N, it is necessary to use such elements of the specified sets that are at the same positions. While the resulting generator matrix $\boldsymbol{G}_N$ is written as follows: $\boldsymbol{G}_N = \boldsymbol{B}_N \cdot \boldsymbol{F}^{\otimes m}$ (Arikan, 2009; Arikan, 2011; Ganin et al, 2017).

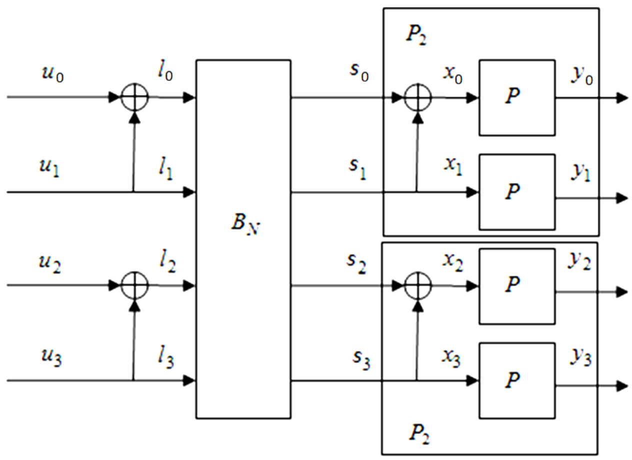
To perform a polarization operation, it is necessary to convert the scalar channel into a vector channel, identifying it with the function of the conditional probability density of the output symbol. This is achieved by recursively creating copies of the binary-symmetric channel, as shown in Figure

1. The scheme for constructing such a system is a multiple of degree 2, starting from zero, and its general form is formulated as follows (Arikan, 2009):

$$P_N\left(y_0^{N-1}\middle|u_0^{N-1}\right) = P^N\left(y_0^{N-1}\middle|u_0^{N-1}\cdot \boldsymbol{G}_N\right)$$

where u_0^{N-1} are the symbols formed by the source of information, and y_0^{N-1} are the symbols received from the communication channel.

Figure 1 – Recursive method for forming a code vector.



The polarization effect is achieved by applying a non-degenerate conversion of $\boldsymbol{B}_N\cdot \boldsymbol{F}^{\otimes m}$ into data. Thus, the polarizing conversion of the channel $P^{2\cdot N}\left(y_0^{2\cdot N-1}\middle|u_0^{2\cdot N-1}\right)$ is based on the polarization of vector channels of smaller dimension $P^i\left(y_0^{i-1}\middle|u_0^{i-1}\right)$, $i = N, N/4, N/2, \ldots, 1$. This decomposition for the conditional probability densities $P_N^{(i)}\left(y_0^{N-1}, u_0^{i-1}\middle|u_i\right)$ is expressed as [1, 4, 7, 8].

$$P_{2\cdot N}^{2\cdot i}\left(y_0^{2\cdot N-1},u_0^{2\cdot i-1}\left|u_{2\cdot i}\right.\right)=\frac{1}{2}\sum_{u_{2\cdot i+1}}P_N^{(i)}\left(y_0^{N-1},u_{0,e}^{2\cdot i-1}\oplus u_{0,o}^{2\cdot i-1}\left|u_{2\cdot i}\oplus u_{2\cdot i+1}\right.\right)\times$$
$$\times P_N^{(i)}\left(y_N^{2\cdot N-1},u_{0,o}^{2\cdot i-1}\left|u_{2\cdot i+1}\right.\right),$$
$$P_{2\cdot N}^{2\cdot i+1}\left(y_0^{2\cdot N-1},u_0^{2\cdot i}\left|u_{2\cdot i+1}\right.\right)=\frac{1}{2}P_N^{(i)}\left(y_0^{N-1},u_{0,e}^{2\cdot i-1}\oplus u_{0,o}^{2\cdot i-1}\left|u_{2\cdot i}\oplus u_{2\cdot i+1}\right.\right)\times$$
$$\times P_N^{(i)}\left(y_N^{2\cdot N-1},u_{0,o}^{2\cdot i-1}\left|u_{2\cdot i+1}\right.\right), \qquad (3)$$

where $u_{0,e}^{2i-1}$ and $u_{0,o}^{2i-1}$ denote the components of the vector only with even and odd numbers, respectively. When $N \to \infty$, the channels $P_N^{(i)}$ will be either completely noiseless or completely unreliable. In this regard, information symbols u_i transmitted through the channels with a low level of reliability, can be considered as ever fixed (frozen) (Gladkikh et al, 2016; Ganin et al, 2017).

The general expression for finding the output code vector X_{N-1} is defined by the following relation:

$$X_{N-1}=u_A\cdot(\boldsymbol{B}_N\cdot\boldsymbol{F}_{nfr}^{\otimes m})\oplus u_{A^{fr}}\cdot(\boldsymbol{B}_N\cdot\boldsymbol{F}_{fr}^{\otimes m})$$

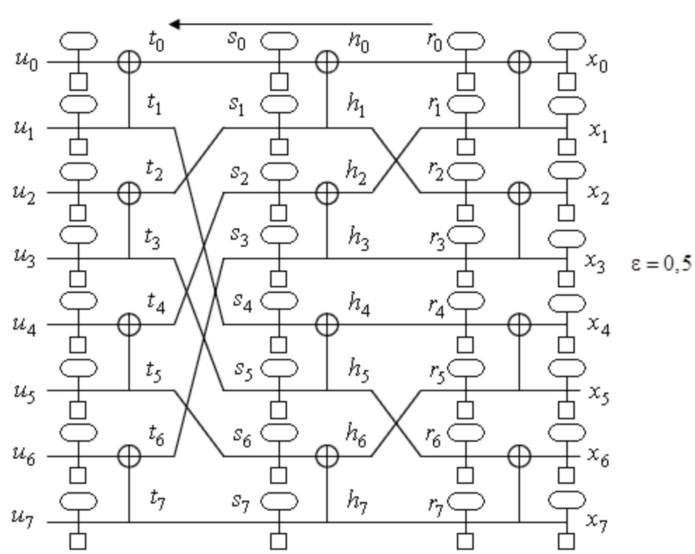
where u_A and $u_{A^{fr}}$ are reliable and unreliable symbols (positions of the code vector, which are assigned a value of 0, respectively. It is worth noting that the concept of unreliable symbols (symbols that are assigned a value of 0) is inextricably linked to the generator matrix of the code in question. In fact, unreliable symbols correspond to «weak» rows of the generating matrix (rows with a minimum weight) (Gladkikh et al, 2013; Ganin et al, 2017).

Consider a PC with lengths of $N=8$. The generator matrix without removing the unreliable channels («weak» rows of the generating matrix) for the specified code is formulated as $\boldsymbol{F}^{\otimes m}=\boldsymbol{F}^{\otimes 3}$ (its m-the Kronecker index is written as $\boldsymbol{F}^{\otimes m}$, where $m \in N$) and equal to (Gladkikh et al, 2016; Namestnikov & Chilikhin, 2017).

$$F^{\otimes 1} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad F^{\otimes 2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \quad F^{\otimes 3} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

where $F^{\otimes 1} = F = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ is the Arikan's kernel (matrix). After a structural analysis of the code, it can be noted that the number of permutations varies in the range of $1 \le M_{\text{perm}} \le m-1$, and the number of calculation levels – $1 \le L_{\text{calc\_lev}} \le m$, respectively. Permutation is such a conversion of a sequence of bits received from the output of an information source, in which the subsequent step differs from the previous one by the magnitude of the column of the original generator matrix. In this case, there is a redistribution of the erasure probability (uncertainty) at each calculation level, which follows the permutation phase. The functional scheme of forming a code vector is shown in Fig. 2 (Gladkikh et al, 2016; Namestnikov & Chilikhin, 2017).

Figure 2 – Functional scheme of forming a PC vector with a length of N = 8.



The symbol denotes the calculation of $2Z - Z^2$ type, and $\square$ is a calculation of Z^2.type. It is worth noting that evaluation of unreliability (degree of «frozenness») of a symbol or a row of the generator matrix (communication channel) occurs at the stage of code vector formation, where the system perceives a sequence of symbols $x_0, x_1, ..., x_{N-1}$ as the symbols, obtained from the communication channel, the erasure probability of which (uncertainty) is equal to $\varepsilon = 0,5$. The arrow shows the direction of the calculations (Gladkikh et al, 2016; Namestnikov & Chilikhin, 2017).

For the considered example, the Bhd distribution is obtained by applying expression, as presented in Table 1.

Table 1 – Bhd distribution for a PC with a length of N=8

Parameter	Z_{u0}	Z_{u1}	Z_{u2}	Z_{u3}	Z_{u4}	Z_{u5}	Z_{u5}	Z_{u7}
Value	0.9961	0.8789	0.8086	0.3134	0.6836	0.1914	0.1211	0.0039

The unreliable (frozen) symbols in the rows of the generator matrix are to be be struck out. It has been shown that if Bhd $Z \to 1$, the channel is considered unreliable (frozen or weak row of the generator matrix is crossed out and does not affect the formation of the code vector, i.e., the specified components of the sequence $\{u_0, u_1, ..., u_{N-1}\}$ are assigned a value of 0. Based on the ranking of the rows of the generator matrix (channels), the encoding rate can be flexibly controlled. Thus, each row of the generator matrix has its own rank based on Bhd (Gladkikh et al, 2016; Namestnikov & Chilikhin, 2017).

Accordingly, for the PC (8.4)

$$Z(8,4) = \begin{pmatrix} 0.9961 \\ 0.8789 \\ 0.8086 \\ 0.3134 \\ 0.6836 \\ 0.1914 \\ 0.1211 \\ 0.0039 \end{pmatrix} \quad \boldsymbol{F}^{\otimes 3} = \begin{pmatrix} \cancel{10000000} \\ \cancel{11000000} \\ \cancel{10100000} \\ 11110000 \\ \cancel{10001000} \\ 11001100 \\ 10101010 \\ 11111111 \end{pmatrix} \quad \to \quad \boldsymbol{F}^{\otimes 3} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Application of PCs in information exchange systems is an innovative area that contributes to increasing capacity of the communication channel by representing a binary symmetric channel in the form of a system of interconnected vector channels with given logical connections.

The PCs formation technology is based on the complete exclusion from the analysis of the received sequence of those channels in which the transmission of bits is notoriously unreliable. Capacity of unreliable channels is considered to be equal to zero (such channels are called «frozen»), and information recovery is carried based on a complex of received data from reliable channels. Thus, one can substantiate the advisability of introducing PC in design of new and modernization of existing data transmission systems (Arikan, 2009; Gladkikh et al, 2016; Korada, 2009; Ganin et al, 2017).

DEVELOPMENT.

Materials and methods

Equivalence of linear block (N, k, d) codes significantly extends the scope of application of redundant codes. It is known that the arrangement of the columns in the generator matrix $\boldsymbol{G}$ of $k \times n$ dimension is unessential, and then there are various options for choosing a basis for the same code. Consequently, various options of the matrix formation $\boldsymbol{G}$ will result in various vectors of the modular code representation. Modular representation of linear block codes allows obtaining the weighted structure of the code and to show that in the case of representation of all code combinations in the form of $\boldsymbol{C}$ matrix, any columns of this matrix will contain exactly 2^{k-2} units. Therefore, taking into account the zero vector, the number of zeros in the column is equal to the number of ones (Gladkikh et al, 2016).

Consider the matrix $M_{k\times(q^k-1)}$ containing all possible vectors as columns of k matrix of $GF(2^k)$ field elements, while the unit element of this field is excluded from the composition of the matrix with respect to the addition operation (zero vector of the field). For example, the $k=3$ matrix M can be given in the form $M=\begin{pmatrix}\alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6\end{pmatrix}$, where α^i – primitive elements of $GF(2^3)$, $i=0,\dots,2^k-2$ set. For binary code, each α^i element is represented as a binary vector $\alpha^0=100$; $\alpha^1=010$ etc. If necessary, individual elements of the field in the matrix M can be excluded.

For example, in conditions of transition to shortened codes. In the general case, a list of all possible non-zero code combinations is obtained in the form of a matrix K of $(q^k-1)\times n$ dimension, i.e. $K=M^T\times G$, which, in fact, is a set of all possible combinations of rows of the matrix G. All elements of the code are contained in the matrix C if M is considered as the generator matrix of the code: $C=M^T\times M$ (Gladkikh et al, 2016).

Assume that $k=3$, then a primitive polynomial for the formation of an extended field is $f(x)=1+x+x^2$, and matrix M takes the following form (Gladkikh et al, 2016):

$$M=\begin{pmatrix}0 & 0 & 1 & 0 & 1 & 1 & 1\\ 0 & 1 & 0 & 1 & 1 & 1 & 0\\ 1 & 0 & 0 & 1 & 0 & 1 & 1\end{pmatrix}.$$

$$C=M^T\times M=\begin{pmatrix}0 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 0\\ 0 & 1 & 1\\ 1 & 1 & 0\\ 1 & 1 & 1\\ 1 & 0 & 1\end{pmatrix}\times\begin{pmatrix}0 & 0 & 1 & 0 & 1 & 1 & 1\\ 0 & 1 & 0 & 1 & 1 & 1 & 0\\ 1 & 0 & 0 & 1 & 0 & 1 & 1\end{pmatrix}=\left(\begin{bmatrix}1 & 0 & 0 & 1 & 0 & 1 & 1\\ 0 & 1 & 0 & 1 & 1 & 1 & 0\\ 0 & 0 & 1 & 0 & 1 & 1 & 1\\ 1 & 1 & 0 & 0 & 1 & 0 & 1\\ 0 & 1 & 1 & 1 & 0 & 0 & 1\\ 1 & 1 & 1 & 0 & 0 & 1 & 0\\ 1 & 0 & 1 & 1 & 1 & 0 & 0\end{bmatrix}\right) \quad (5)$$

By applying in matrix C a generator matrix instead of M (for example, PC), it is possible to get a modular representation of this code, which can later be used to perform equivalent manipulations (permutations) with combinations of the selected code. Select the first 3 columns in 5 and represent the elements of these columns in terms of primitive elements α^i of $GF(2^3)$ field. Then to the selected first, second and third columns of the resulting matrix C add the 4th, after deleting the first. The obtained sets of binary elements can again be represented as a set α^i (Gladkikh et al, 2016).

While performing a similar cyclic shift of columns of the matrix C, it can generally be represented it in the form of a set of elements $\alpha^i \in GF(2^k)$. Since in the newly formed matrix C_{GF} , each column of which contains one sample of each α^i representative, then in the binary format, each column of the matrix C will contain exactly 2^{k-1} units. Matrix C is a non-degenerate one. Indeed,

$$C_{GF} \Rightarrow \begin{pmatrix} \alpha^2 & \alpha^0 & \alpha^1 & \alpha^6 & \alpha^3 & \alpha^5 & \alpha^4 \\ \alpha^1 & \alpha^6 & \alpha^3 & \alpha^5 & \alpha^4 & \alpha^2 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^6 & \alpha^3 & \alpha^5 & \alpha^4 & \alpha^2 \\ \alpha^4 & \alpha^2 & \alpha^0 & \alpha^1 & \alpha^6 & \alpha^3 & \alpha^5 \\ \alpha^3 & \alpha^5 & \alpha^4 & \alpha^2 & \alpha^0 & \alpha^1 & \alpha^6 \\ \alpha^5 & \alpha^4 & \alpha^2 & \alpha^0 & \alpha^1 & \alpha^6 & \alpha^3 \\ \alpha^6 & \alpha^3 & \alpha^5 & \alpha^4 & \alpha^2 & \alpha^0 & \alpha^1 \end{pmatrix} \quad (6)$$

The last two columns in the matrix C_{GF} are formed by the simultaneous cyclical shifting of the rows in the matrix C by 1, up to $k-1$ steps. Analysis of the resulting matrix C_{GF} shows that all α^i are ordered by rows and represent cyclic shifts of the first row. Therefore, while choosing any k columns from the matrix C_{GF} , the set of column elements remains unchanged, and the order of the columns makes no difference. This property of the modular representation of codes allows formulating an equivalence condition for codes: an equivalent code for C code is such a $C_{эк}$ code, which source code columns C are permuted preserving all properties of this code unchanged.

It is known that non-observance of the code equivalence condition results in an increase in the complexity of the computational process performed by the decoder processor by introducing an additional non-degeneracy property refinement procedure for the newly formed matrix. If this property conditions are not met, then the decoder carries out additional symbol permutations with mandatory verification of the non-degeneracy property in new matrices. Analysis shows that this is always true for a representation of $\boldsymbol{C} = \boldsymbol{M}^T \times \boldsymbol{M}$ type. If $\boldsymbol{M}$ matrix is replaced by $\boldsymbol{G}$ generator matrix of some block code, this condition can be violated by applying permutations. Thus, not all permutations of columns of the generator code matrix lead to the formation of equivalent codes. The appearance of such an unfavourable event is highly probable, and therefore, each permutation of columns requires a correctness verification of the equivalent code formation procedure (Gladkikh et al, 2016).

Based on the algebraic theory of groups and fields, the algorithms for non-algebraic decoding of block codes have been developed, including PCs that use lexicographic techniques of partitioning space of allowed code vectors into clusters (lists) with processing of these lists by permutations, using the equivalence property of systematic and non-systematic block codes. The algorithm of lexicographical processing of any received code vector is reduced to the following steps (Gladkikh et al, 2016; Ganin et al, 2017; Gladkikh et al, 2017):

Step 1. Receive a (N, k) soft-decision symbols (SDS) code vector from a communication channel.

Step 2. Next, it is necessary to process f symbols (digit positions) that are responsible for the cluster number. In this case, positions of the specified symbols (digit positions) are known beforehand on the receiver side. The decoder evaluates the level of their reliability and, in case of distortion performs their recovery using the received redundant symbols. The receiver decides on the cluster number.

Step 3. Next, it is necessary to call from the base of key combinations relevant both to the given cluster distribution, and the specific key combination, and use it to convert the received vector into a base (zero) cluster.

Step 4. Arrange the columns of the base cluster using the SDS ranking in descending order for the received vector.

Step 5. It is necessary to make sure that in the selected $k-f$ the positions of the columns of an ordered base (zero) cluster formed elements of the Galois field $GF(2^{k-f})$, which indicates the absence of linear dependence among the selected columns. Perform another sorting, if necessary.

Step 6. Recover the vector and reverse the permutation of its elements, highlighting the true error vector.

The essence of PC permutation decoding based on the lexicographic approach that $\boldsymbol{C}_{n,k}$ code with the generator matrix $\boldsymbol{G}$ and d Hamming metric comprising a set of allowed code vectors $\boldsymbol{C}_{n,k} = \{c_0, c_1, ..., c_{2^k}\}$ during the list decoding of the word received from $V_{acc} = V_{tr} \oplus e$ interference channel, where V_{acc} is the received vector, V_{tr} is transmitted vector, e is the error vector, and $V_{tr} \in \boldsymbol{C}_{n,k}$ is the error list for $\{S\} \in \boldsymbol{C}_{n,k}$ set of words that situated from the word V_{acc} at the distance of $d-1$ and closer. Based on the maximum likelihood criterion, the decoder replaces V_{acc} word with $\boldsymbol{C}_{np} \in \ {}_{n,k}$ vector from $\{S\}$ list, which has the largest number of coincident positions with V_{acc} word. This work shows the possibility of partitioning the space $\boldsymbol{C}_{n,k}$ into clusters using structural features (algebraic patterns) of noise-immune codes. Each cluster contains a closed set of $\{c_i\} \in \boldsymbol{C}_{n,k}$ combinations, where $0 \le i \le 2^f - 1$, f are the number of the same bit numbers for any combination of $\boldsymbol{C}_{n,k}$ space allocated to the cluster flag, while for systematic codes $f \le k$ and $\{f\} \in GF(2^{k-f})$

(Gladkikh et al, 2016; Ganin et al, 2017; Gladkikh et al, 2017; Chilikhin, 2014; Gladkikh & Chilikhin, 2014; Gladkikh & Chilikhin, 2013).

The ordering of i numbers is a lexicographic procedure that allows to reduce the time of the list formation by 2^f times due to the single-valued allocation from $C_{n,k}$ combinations of the cluster with i number, hence, $\{S_i\} = c_{i0}, c_{i1}, ..., c_{i2^k-1}$. Assuming that $2^k - 1 = \xi$ for the whole $C_{n,k}$ set, the following expression is obtained (Chilikhin, 2014; Gladkikh & Chilikhin, 2014; Gladkikh et al, 2013; Gladkikh et al., 2016):

$$
\begin{array}{lllll}
i = 0 & \{c_{00}, & c_{01}, & ..., & c_{0(2^k-1)}\}; \\
i = 1 & \{c_{10}, & c_{11}, & ..., & c_{1(2^k-1)}\}; \\
... & ... & ... & ... & ... \\
i = \xi & \{c_{\xi 0}, & c_{\xi 1}, & ..., & c_{\xi(2^k-1)}\}.
\end{array}
$$

A selection the same $f \le k$ digit positions of the systematic code for all $C_{n,k}$ allows to reduce the length of the list by 2^{k-f} times using partitioning of $C_{n,k}$ space into C_{2^f} clusters. A similar procedure is valid for non-systematic codes, such as PC, despite the absence of a pronounced structure in the allocation of information bits (Gladkikh et al, 2016).

Consider an example of applying the described algorithm for PC (8.4). All allowed combinations of the codebook are shown in Table 2. For the sake of visual simplicity, we highlighted the symbols of the code combination (high-order bits) forming the cluster number (Ganin et al, 2017).

Table 2 – Correspondence of information symbols and code combinations obtained after encoding.

K, bit	N, bit							
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
0000	0	0	0	0	0	0	0	0
0001	1	1	1	1	1	1	1	1
0010	1	0	1	0	1	0	1	0
0011	0	1	0	1	0	1	0	1
0100	1	1	0	0	1	1	0	0
0101	0	0	1	1	0	0	1	1
0110	0	1	1	0	0	1	1	0
0111	1	0	0	1	1	0	0	1
1000	1	1	1	1	0	0	0	0
1001	0	0	0	0	1	1	1	1
1010	0	1	0	1	1	0	1	0
1011	1	0	1	0	0	1	0	1
1100	0	0	1	1	1	1	0	0
1101	1	1	0	0	0	0	1	1
1110	1	0	0	1	0	1	1	0
1111	0	1	1	0	1	0	0	1

For the transition to cluster formation, it is necessary to structure the high-older bits of the allowed code combinations. The process of codebook clustering is presented in Table 3 (Ganin et al, 2017).

Table 3 – Correspondence of information symbols and code combinations obtained after encoding and broken into unique identifiers (clusters).

	K, bit	N, bit							
		a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
	0000	0	0	0	**0**	**0**	0	0	0
Cluster 00 (base cluster)	0101	0	0	1	1	0	0	1	1
	1001	0	0	0	0	1	1	1	1
	1100	0	0	1	1	1	1	0	0
	0011	0	1	0	1	0	1	0	1
Cluster 01	0110	0	1	1	**0**	**0**	1	1	0
	1010	0	1	0	1	1	0	1	0
	1111	0	1	1	0	1	0	0	1
	0010	1	0	1	0	1	0	1	0
Cluster 10	0111	1	0	0	1	1	0	0	1
	1011	1	0	1	**0**	**0**	1	0	1
	1110	1	0	0	1	0	1	1	0
	0001	1	1	1	1	1	1	1	1
Cluster 11	0100	1	1	0	0	1	1	0	0
	1000	1	1	1	1	0	0	0	0
	1101	1	1	0	**0**	**0**	0	1	1

The generation matrix G of PC (8,4) in a systematic form is written as follows (Gladkikh et al, 2016):

$$\boldsymbol{G}_{(8,4)} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Assume a vector at the output of the encoder in the following form:

$$V_{code} = \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0$$

The transmitter replaces the least significant bit (the rightmost) of the combination with the parity bit for two high-order bits $\{a_0, a_1\}$, defining the cluster number with the purpose to protect it. Consequently, the vector is transferred to the communication channel in the following form:

$$V_{tr} = \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1$$

The receiver takes the vector by identifying the reliability gradation for each symbol, according to some known principle (likelihood ratio test). Assume a correspondence of symbols and reliability gradation symbols (for the sake of simplicity, it is expedient to use the term SDS) as shown below (Gladkikh et al, 2016)

$$\begin{array}{lcccccccc} V_{tr} = & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ V_{acc} = & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ \text{MPC} & \mathbf{6} & \mathbf{2} & \mathbf{1} & \mathbf{3} & \mathbf{4} & \mathbf{7} & \mathbf{6} & \mathbf{7} \end{array}$$

Consequently, the error vector is represented by the following sequence

$$V_{er} = \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0$$

At the first step of decoding, having received a vector with errors, the decoder checks the cluster number for parity. In the present example, parity check has shown a negative result, so the decoder inverts the second bit in the cluster number, since it has the worst result of reliability estimation. It should be noted that this approach to protecting the cluster number is very trivial. At the moment, there is a whole complex of more effective methods for protecting cluster number, but implementation of these approaches leads to an increase in the redundancy of the code combination (Gladkikh et al, 2016).

The vector used for the subsequent analysis takes the following form:

$$V_{acc} = \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1$$

It is worth noting that initially, the error has been deliberately introduced while encoding, for the purpose of cluster number protection. Thus, the distribution of ranks will differ (bold and underlined symbols denote the maximum SDS values).

$$V_{tr} = \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1$$

$$V_{acc} = \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1$$

$$\text{SDS} \quad \underline{\mathbf{6}} \quad 2 \quad 1 \quad 3 \quad 4 \quad \underline{\mathbf{7}} \quad \underline{\mathbf{6}} \quad 0$$

The number of the restored cluster is 10. Thus, for transition to the truncated code combination, the decoder performs a modulo-2 addition with the keyword of this cluster, while the keyword for the second cluster is written in the following form:

$$K_{2\to 0} = \quad 1 \quad 0 \quad 1 \quad \underline{\mathbf{0}} \quad \underline{\mathbf{0}} \quad 1 \quad 0 \quad 1$$

After deleting the cluster number:

$$K'_{2\to 0} = \quad 1 \quad \underline{\mathbf{0}} \quad \underline{\mathbf{0}} \quad 1 \quad 0 \quad 1$$

Using the $V_{acc} \oplus K_{2\to 0} = V'_{acc}$ procedure, the vector with a_i symbols ranked in descending order is obtained.

$$V'_{acc} = \quad \oplus \begin{array}{cccccccc} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{array}$$

Thus, the vector V'_{acc} has the following form:

$$V'_{acc} = \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0$$

Next, discard the bits responsible for the cluster number, due to the above decision on the reliable transmission. Thus, the recourse vector takes the following form:

$V'_{acc} =$ 0 0 1 0 1 0

Hence, the SDS for each earlier identified symbol of this vector are as follows:

$V'_{acc} =$	0	0	1	0	1	0
SDS	1	3	4	**<u>7</u>**	**<u>6</u>**	0

Next, the vector in question should be considered in decreasing order of the reliability gradient. The last character is assigned the SDS below the minimum value (i.e. 0), since this symbol has been used to protect the cluster number (parity check), and a forced change of its value has been initially made. For this reason, according to the reliability gradient, this symbol occupies the last position, which is the most unreliable (Gladkikh et al, 2016).

SDS	**<u>7</u>**	**<u>6</u>**	4	3	1	0
$V_{rank} =$	0	1	1	0	0	0

The symbols of the maximum SDS values are highlighted in grey, while 7$\rightarrow$0, 6$\rightarrow$1. After the conversion into the base cluster (cluster ID $i = 00$), it is necessary to carry out a permutation of the columns, with regard for the reliability gradation. Correspondence of the SDS with the columns of the base cluster is shown in Table 4 (Gladkikh et al, 2016).

Table 4 – Correspondence of the SDS with the columns of the base cluster ($i = 00$)

Base cluster		MPC					
		1	3	4	**<u>7</u>**	**<u>6</u>**	0
0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	1
0	0	0	0	1	1	1	1
0	0	1	1	1	1	0	0

After the permutation, the correspondence takes the form shown in Table 5.

Table 5 – Correspondence of the SDS with the columns of the base cluster ($i = 00$) after the permutation, with regard for the reliability gradation.

Base cluster		MPC					
		<u>7</u>	**<u>6</u>**	4	3	1	0
0	0	0	0	0	0	0	0
0	0	**<u>0</u>**	**<u>1</u>**	0	1	1	1
0	0	1	1	1	0	0	1
0	0	1	0	1	1	1	0

Then, the keyword in the context of the base cluster is:

$$K_{0_2\,rank} = \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1$$

Vector $W_{rez(10)}$ is obtained by adding this word and the permuted vector $W_{short(p)}$:

$$K_{0_2\,rank} = \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1$$

$$V_{rank} = \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0$$

$$V_{er_{rank}} = \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1$$

Then, the original sequence of characters:

SDS	7 6 4 3 1 0
$V_{er_{rank}} =$	0 0 1 1 1 1
SDS source	1 3 4 7 6 0
$V_{er} =$	1 1 1 0 0 1

Adding to this vector the cluster number $i = 00$ and

$$V_{acc} = \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1$$

vector, results in the following sequences:

$$V_{acc} = \quad 1 \; 0 \; 1 \; 0 \; 1 \; 1 \; 1 \; 1$$

$$V_{er} = \quad 0 \; 0 \; 1 \; 1 \; 1 \; 0 \; 0 \; 1$$

$$V_{tr} = \quad 1 \; 0 \; 0 \; 1 \; 0 \; 1 \; 1 \; 0$$

$$V_{tr(source)} = \quad 1 \; 0 \; 0 \; 1 \; 0 \; 1 \; 1 \; 1$$

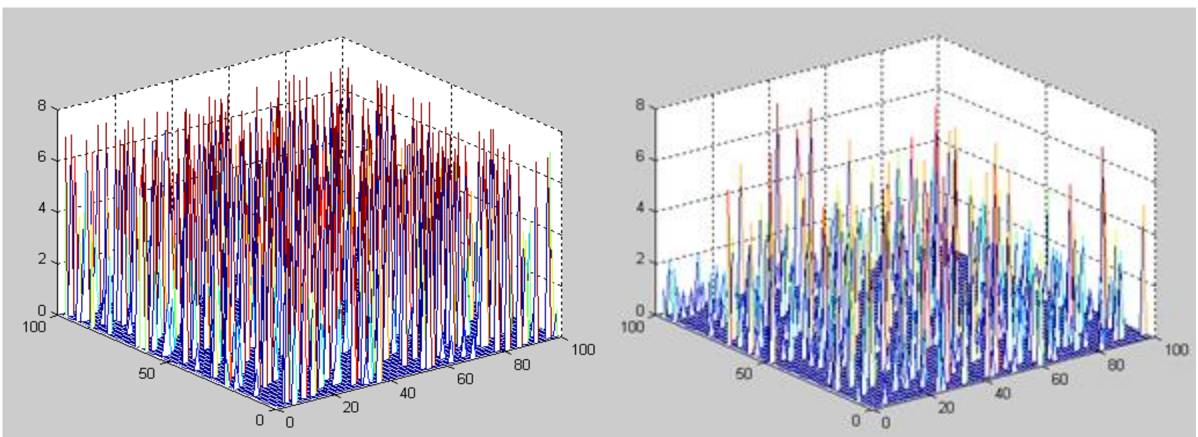
In so doing, the original code word is obtained immediately. In addition, during processing, the deliberate introduction of the error regarding the protection of the cluster has been taken into account. Thus, the specified algorithm allows to correct n-k+1 erasions. This fact suggests that the algorithm of permutation decoding based on the lexicographic approach makes maximum use of the redundancy introduced into the code. However, in the step of column permutation with regard to the SDS reliability gradation, a situation may arise when positions with erroneous symbols in the code combination have high SDS values. Such an effect leads to error propagation because the received vector will be incorrectly converted into the base cluster (Gladkikh et al, 2016; Gladkikh et al, 2017).

The erasure link properties are used while calculating SDS for a binary channel in the receiver decision circuit. Its parameters are chosen so that the erasure interval is wide and constant in value. Thus, all signals received outside the uncertainty zone of (in the neighbourhood of mathematical expectation of the random variable y), the first decision scheme assigns a maximum SDS gradation equal to $\lambda_{\max}$. Other values of $\lambda_i < \lambda_{\max}$ are formed on the basis of the linear characteristic, where $\lambda_{\max} = 7$ is the maximum estimate. The value of $\lambda_{\max} = 7$ is considered sufficient to receive symbols. The generic analytical expression for all types of modulation of characteristics within the erasure interval γ in the worst-case scenario is written as follows (Gladkikh et al, 2016; Gladkikh & Klimov, 2013; Gladkikh & Chilikhin, 2014; Klimov & Solodovnikova, 2014):

$$\lambda_i(y) = \left\lfloor \left| \frac{\lambda_{\max}}{\gamma \cdot M_{mp}} \times y \right| \right\rfloor \times a_s$$

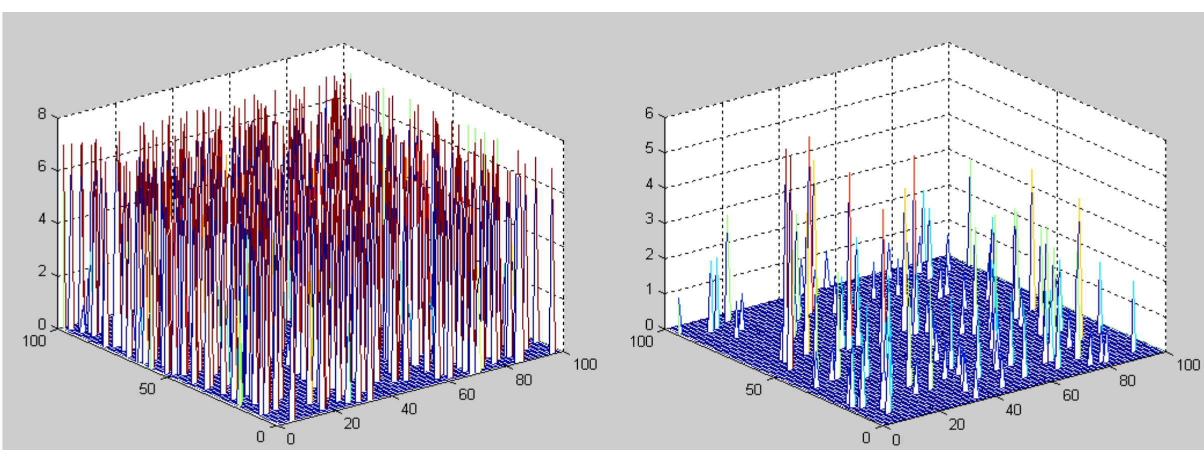
where M_{mp} is the mathematical expectation of the parameter being modulated, a_s is the multiplicative interference ratio. By manipulating angular brackets in (8), the receiver constructor can reduce the computational error $\lambda_i(y)$ or proceed to the rational indicators of this parameter, which corresponds to generally accepted norms (Gladkikh et al, 2016; Gladkikh & Klimov, 2013; Gladkikh & Chilikhin, 2013; Klimov & Solodovnikova, 2014).

Figure 3 – Comparative SDS data for $h = 0$ dB and $\rho = 0{,}9$



a) estimates of correct symbols. b) estimates of erroneous symbols.

Figure 4 – Comparative SDS data for $h = 3$ dB and $\rho = 0{,}9$



a) estimates of correct symbols. b) estimates of erroneous symbols.

Figures 3 and 4 show polygons of estimates for correctly and erroneously received symbols depending on the signal-to-noise ratio. For convenience, the code combination in the figures is shown in the form of 100×100 identity matrices with signal-to-noise ratios of 0 and 3 dB, which are of special scientific interest (Gladkikh et al, 2016; Gladkikh et al, 2013).

Based on the received characteristics, it is obvious that when the signal-to-noise ratio increases, the number of occurrences of erroneous symbols with high SDS values decreases. This is due to an improvement in the noise environment in the communication channel in question. Nevertheless, in the range of low values of the signal-to-noise ratio, erroneous symbols with high SDS values are observed.

This paper considers the design features of PC code combinations to eliminate this effect. The main structural feature of allowed PC combinations is their structural symmetry (or inverse structural symmetry) about n/2 combination symbols. This fact can be exemplified by $y = 1001\ 1001\ 1001\ 1001$ vector for the PC code (16,8). This code length has been chosen for a more vivid demonstration of the code design features. It is obvious that the sum of a modulo-2 addition of n/2 symbols of a combination with each other consists of Boolean zeros only. Accordingly, the sum n/2 symbols of $y = 1001\ 1001\ 0110\ 0110$ vector from the same set of allowed PC (16,8) combinations consists of boolean ones only [4, 6, 18] (Gladkikh et al, 2015; Ganin et al., 2017; Chilikhin, 2015).

$$1001\ 1001 \oplus 1001\ 1001 = 0000\ 0000$$
$$1001\ 1001 \oplus 0110\ 0110 = 1111\ 1111$$

Moreover, only boolean ones are formed when the code combination is anti-symmetric about n/2 symbols, and only boolean zeros – if n/2 symbols of the combination are identical to each other. Such a structural feature of the code (SFC) is preserved even when the code combination is decomposed to a minimum of 2 symbols. In general, a PC combination is divided by $n/2^j$, where j=1,2…m-1 is the decomposition step. Assume the following vector at the output of the encoder (an arbitrarily vector to understand the mechanism under consideration) (Chilikhin, 2015; Tal & Vardy, 2011; Tal & Vardy, 2013; Coetzee & Mammen, 2017):

$$V_{tr} = \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1$$

The receiver takes the vector form the communication channel as follows:

$$V_{acc} = \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad \mathbf{0} \quad 0 \quad 0 \quad 1$$

Firstly, to check the received combination for errors, it is necessary to carry out a modulo-2 addition of parts of this combination consisting of 2^{m-1} elements.

$$\begin{array}{c|cccccccc} \oplus & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ & 0 & 1 & 1 & 0 & \mathbf{0} & 0 & 0 & 1 \\ \hline & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \end{array}$$

Next, it is necessary to carry out a modulo-2 addition of parts of this combination consisting of 2^{m-2} elements.

$$\begin{array}{c|cccc} \oplus & 1 & 0 & 0 & 1 \\ & \mathbf{0} & 0 & 0 & 1 \\ & \mathbf{1} & 0 & 0 & 0 \end{array}$$

The 3rd decomposition step does not allow unequivocally identifying the error vector or confirming the hypothesis of error-free transmission of the original message. To restore the original message, it is necessary to perform a procedure of direct addition (see Figure 5) and a procedure of cross-addition (see Figure 6) for the received code sequence (Chilikhin, 2015; Tal & Vardy, 2011; Tal & Vardy, 2013; Coetzee & Mammen, 2017).

Figure 5 – Schematic representation of the direct addition operation by elements from the first and second half .

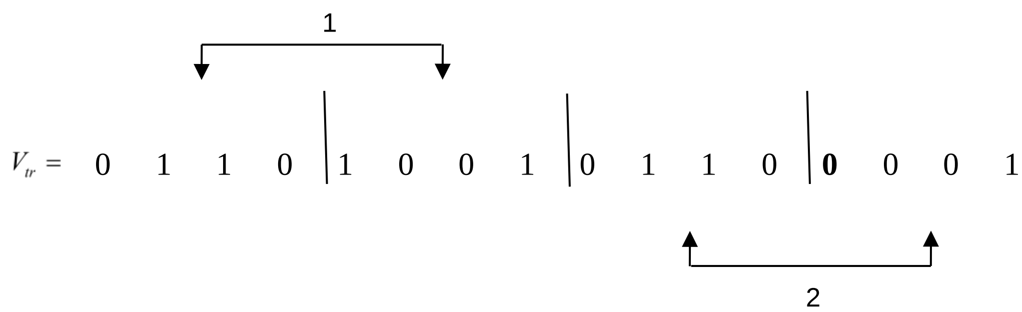
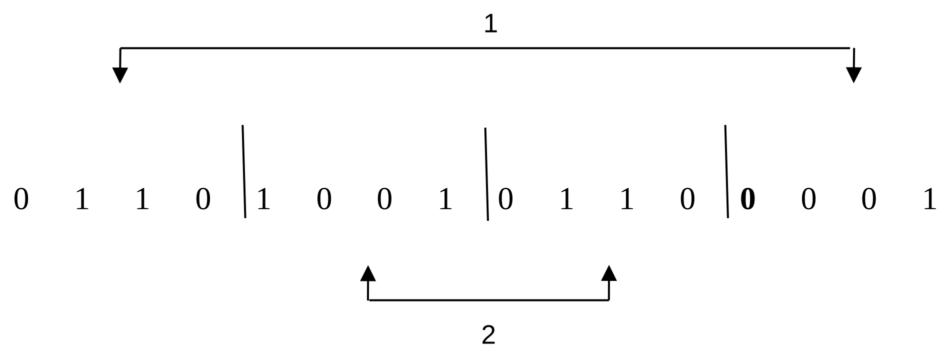


Figure 6 – Schematic representation of the cross-addition operation by elements of the combination



It is worth noting that the use of SFC PC allows not only assessing the correctness or falsity of SDS for all bits of the received code combination, but also providing protection of the cluster number without introducing additional redundancy. It is possible to create a number of rules allowing application of this methodology and providing an SFC-based effective protection of the cluster number. To do this, the following condition should be met (Gladkikh et al, 2016; Ganin et al, 2017; Chilikhin, 2015):

1. Size of the cluster number does not exceed the number of information symbols, $f \leq k$;
2. Size of the cluster number is not less than $f \geq 2$;
3. Size of the cluster number is a multiple of 2 due to SFC PC.

Modified algorithm with provision for the structural features of code combinations.

To understand the proposed modifications of the lexicographic decoding algorithm for polar codes, consider the clustering procedure for a set of allowed code combinations for PC lengths of $N=8$. Table 6 and 7 show the correspondences of information symbols and received code combinations with provision for the clustering using a unique identifier (Gladkikh et al, 2016; Ganin et al, 2017).

Table 6 – Correspondence of information symbols and code combinations obtained after encoding.

K, bit / K, bit	N, bit / N, bit							
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
0000	0	0	0	0	0	0	0	0
0001	1	1	1	1	1	1	1	1
0010	1	0	1	0	1	0	1	0
0011	0	1	0	1	0	1	0	1
0100	1	1	0	0	1	1	0	0
0101	0	0	1	1	0	0	1	1
0110	0	1	1	0	0	1	1	0
0111	1	0	0	1	1	0	0	1
1000	1	1	1	1	0	0	0	0
1001	0	0	0	0	1	1	1	1
1010	0	1	0	1	1	0	1	0
1011	1	0	1	0	0	1	0	1
1100	0	0	1	1	1	1	0	0
1101	1	1	0	0	0	0	1	1
1110	1	0	0	1	0	1	1	0
1111	0	1	1	0	1	0	0	1

The procedure of dividing into clusters is reduced to structuring according to the high-order bits of the allowed code combinations. A breakdown of the codebook into clusters is presented in Table 7.

Table 7 – Correspondence of information symbols and code combinations obtained after encoding and broken into unique identifiers (clusters).

	K, bit / K, bit	N, bit / N, bit								
		a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	
Cluster 00 (base) / Cluster 00 (base)	0000	0	0	0	**0**	**0**	0	0	0	Key combination (00) / Key combination (00)
	0101	0	0	1	1	0	0	1	1	
	1001	0	0	0	0	1	1	1	1	
	1100	0	0	1	1	1	1	0	0	
Cluster 01/ Cluster 01	0011	0	1	0	1	0	1	0	1	Key combination (01) / Key combination (01)
	0110	0	1	1	**0**	**0**	1	1	0	
	1010	0	1	0	1	1	0	1	0	
	1111	0	1	1	0	1	0	0	1	
	0010	1	0	1	0	1	0	1	0	
	0111	1	0	0	1	1	0	0	1	Key combination (10) / Key combination (10)
	1011	1	0	1	**0**	**0**	1	0	1	
	1110	1	0	0	1	0	1	1	0	
Cluster 11/ Cluster 11	0001	1	1	1	1	1	1	1	1	
	0100	1	1	0	0	1	1	0	0	
	1000	1	1	1	1	0	0	0	0	
	1101	1	1	0	**0**	**0**	0	1	1	Key combination (11) / Key combination (11)

However, the proposed modification has a certain functional limitation. The lexicographic approach to decoding described above is presented for a PC with a ratio of codeword bit length and information bit length formulated as. In the context of correcting ability (error correction) and encoding rate, the optimal ratio is obtained when the information bit length is equal to $(N, 1 \div N/2)$. However, with an increase of informational bit length, it is not always possible to perform the decomposition in the second and subsequent steps (Gladkikh et al, 2016; Ganin et al, 2017; Chilikhin, 2015).

In this case, the conceptual rule of the first decomposition step (structural symmetry about $N/2$) is valid for any information bit length. It can be said that the imposed constraint does not change some functionalities of the proposed approach, e.g. protection of the cluster number.

Consider the generator matrix PC (8.5), obtained as a Kronecker product of the Arikan matrix (kernel), prior to removing weak («frozen» or unreliable) rows (rows with low weight or rows with the smallest number of Boolean ones), using the Bhd technique.

A set of allowed code combinations for PC (8.5) is formed using this generator matrix. It is presented in Table 8. The combinations that do not have symmetry about the 2nd and subsequent SD are highlighted in dark grey (Ganin et al, 2017).

Table 8 – Correspondence of information symbols and obtained PC code combinations after encoding (k=5).

K, bit / K, bit	N, bit / N, bit							
00000	0	0	0	0	0	0	0	0
00001	1	1	1	1	1	1	1	1
00010	1	0	1	0	1	0	1	0
00011	0	1	0	1	0	1	0	1
00100	1	1	0	0	1	1	0	0
00101	0	0	1	1	0	0	1	1
00110	0	1	1	0	0	1	1	0
00111	1	0	0	1	1	0	0	1
01000	1	0	0	0	1	0	0	0
01001	0	1	1	1	0	1	1	1
01010	0	0	1	0	0	0	1	0
01011	1	1	0	1	1	1	0	1
01100	0	1	0	0	0	1	0	0
01101	1	0	1	1	1	0	1	1
01110	1	1	1	0	1	1	1	0
01111	0	0	0	1	0	0	0	1
10000	1	1	1	1	0	0	0	0
10001	0	0	0	0	1	1	1	1
10010	0	1	0	1	1	0	1	0
10011	1	0	1	0	0	1	0	1
10100	0	0	1	1	1	1	0	0
10101	1	1	0	0	0	0	1	1
10110	1	0	0	1	0	1	1	0
10111	0	1	1	0	1	0	0	1
11000	0	1	1	1	1	0	0	0
11001	1	0	0	0	0	1	1	1
11010	1	1	0	1	0	0	1	0
11011	0	0	1	0	1	1	0	1
11100	1	0	1	1	0	1	0	0
11101	0	1	0	0	1	0	1	1
11110	0	0	0	1	1	1	1	0
11111	1	1	1	0	0	0	0	1

It is obvious that with an increase in the number of information symbols within the range of $N/2+1 \div N-1$, the second and subsequent decomposition steps cannot be performed. This is because the number of combinations 00, 11, 01, 10 is finite. It is important to note that structural symmetry about $N/2$ is preserved. Thus, this aspect should be taken into account while implementing the theory of structural features of PC code combinations in decoding algorithms.

Modified algorithm based on cluster number, and with the largest number of soft symbol decisions.

An analysis of the set of allowed PC combinations allows concluding that it is not necessary to use only a_0 a_1 bits as cluster identifiers. The specificity of image compression code is in incomplete codebook (Ganin et al, 2017).

In this connection, it is necessary to use only such combinations of bits that have a consistent (identically equal) system of key combinations. Otherwise, the use of image compression code is impractical. The research shows that when using the pairwise combinations a_i a_{i+1}, the system of key combinations remains consistent, which opens the prospects of using the most «strong» bits as cluster identifiers in the context of SDS. Tables 9 and 10 show a similar correspondence.

Table 9 – Correspondence of information symbols and received code combinations after encoding with and without breaking into clusters (a_0 a_1 cluster number).

K, bit / K, bit	N, bit / N, bit							
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
0000	0	0	0	0	0	0	0	0
0001	1	1	1	1	1	1	1	1
0010	1	0	1	0	1	0	1	0
0011	0	1	0	1	0	1	0	1
0100	1	1	0	0	1	1	0	0
0101	0	0	1	1	0	0	1	1

K, bit / K, bit	N, bit / N, bit							
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
0000	0	0	0	**0**	**0**	0	0	0
0101	0	0	1	1	0	0	1	1
1001	0	0	0	0	1	1	1	1
1100	0	0	1	1	1	1	0	0
0011	0	1	0	1	0	1	0	1
0110	0	1	1	**0**	**0**	1	1	0

0110	0	1	1	0	0	1	1	0
0111	1	0	0	1	1	0	0	1
1000	1	1	1	1	0	0	0	0
1001	0	0	0	0	1	1	1	1
1010	0	1	0	1	1	0	1	0
1011	1	0	1	0	0	1	0	1
1100	0	0	1	1	1	1	0	0
1101	1	1	0	0	0	0	1	1
1110	1	0	0	1	0	1	1	0
1111	0	1	1	0	1	0	0	1

1010	0	1	0	1	1	0	1	0
1111	0	1	1	0	1	0	0	1
0010	1	0	1	0	1	0	1	0
0111	1	0	0	1	1	0	0	1
1011	1	0	1	**0**	**0**	1	0	1
1110	1	0	0	1	0	1	1	0
0001	1	1	1	1	1	1	1	1
0100	1	1	0	0	1	1	0	0
1000	1	1	1	1	0	0	0	0
1101	1	1	0	**0**	**0**	0	1	1

Table 10 – Correspondence of information symbols and received code combinations after encoding with and without breaking into clusters (high-order bits: a_4 a_5).

K, bit / K, bit	N, bit / N, bit							
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
0000	0	0	0	0	0	0	0	0
0001	1	1	1	1	1	1	1	1
0010	1	0	1	0	1	0	1	0
0011	0	1	0	1	0	1	0	1
0100	1	1	0	0	1	1	0	0
0101	0	0	1	1	0	0	1	1
0110	0	1	1	0	0	1	1	0
0111	1	0	0	1	1	0	0	1
1000	1	1	1	1	0	0	0	0
1001	0	0	0	0	1	1	1	1
1010	0	1	0	1	1	0	1	0
1011	1	0	1	0	0	1	0	1
1100	0	0	1	1	1	1	0	0
1101	1	1	0	0	0	0	1	1
1110	1	0	0	1	0	1	1	0
1111	0	1	1	0	1	0	0	1

K, bit / K, bit	N, bit / N, bit							
	a_4	a_5	a_0	a_1	a_2	a_3	a_6	a_7
0000	0	0	0	**0**	**0**	0	0	0
0101	0	0	0	0	1	1	1	1
1000	0	0	1	1	1	1	0	0
1101	0	0	1	1	0	0	1	1
0011	0	1	0	1	0	1	0	1
0110	0	1	0	1	1	0	1	0
1011	0	1	1	0	1	0	0	1
1110	0	1	1	**0**	**0**	1	1	0
0010	1	0	1	0	1	0	1	0
0111	1	0	1	**0**	**0**	1	0	1
1010	1	0	0	1	0	1	1	0
1111	1	0	0	1	1	0	0	1
0001	1	1	1	1	1	1	1	1
0100	1	1	1	1	0	0	0	0
1001	1	1	0	**0**	**0**	0	1	1
1100	1	1	0	0	1	1	0	0

Modified algorithm based on the cluster number spacing over the whole length of the code combination.

An interference analysis in the communication channel shows that influence of interferences is generally impulsive (except of anthropogenic factors resulting in complete signal suppression). Provided a sufficient length of the code structure, even an occurrence of errors makes it possible to void this influence.

Thus, it is advisable to distribute the bits responsible for the cluster number along the whole length of the combination. Table 11 presents such an opportunity (Ganin et al, 2017).

Table 11 – Correspondence of information symbols and received code combinations after encoding with and without breaking into clusters (high-order bits: a_1 a_6).

K, bit / K, bit	N, bit / N, bit							
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
0000	0	0	0	0	0	0	0	0
0001	1	1	1	1	1	1	1	1
0010	1	0	1	0	1	0	1	0
0011	0	1	0	1	0	1	0	1
0100	1	1	0	0	1	1	0	0
0101	0	0	1	1	0	0	1	1
0110	0	1	1	0	0	1	1	0
0111	1	0	0	1	1	0	0	1
1000	1	1	1	1	0	0	0	0
1001	0	0	0	0	1	1	1	1
1010	0	1	0	1	1	0	1	0
1011	1	0	1	0	0	1	0	1
1100	0	0	1	1	1	1	0	0
1101	1	1	0	0	0	0	1	1
1110	1	0	0	1	0	1	1	0
1111	0	1	1	0	1	0	0	1

K, bit / K, bit	N, bit / N, bit							
	a_1	a_6	a_2	a_3	a_4	a_5	a_0	a_7
0000	0	0	0	**0**	**0**	0	0	0
0111	0	0	1	0	1	1	0	1
1011	0	0	1	1	0	0	1	1
1100	0	0	0	1	1	1	1	0
0010	0	1	1	1	0	1	0	0
0101	0	1	0	1	1	0	0	1
1001	0	1	0	**0**	**0**	1	1	1
1110	0	1	1	0	1	0	1	0
0011	1	0	0	0	1	0	1	1
0100	1	0	1	**0**	**0**	1	1	0
1000	1	0	1	1	1	0	0	0
1111	1	0	0	1	0	1	0	1
0001	1	1	1	1	1	1	1	1
0110	1	1	0	1	0	0	1	0
1010	1	1	0	0	1	1	0	0
1101	1	1	1	**0**	**0**	0	0	1

Obviously, this approach helps counter grouping of errors and powerful impulse interference. However, such a spacing limit the choice of bits responsible for the cluster number with the highest SDS value, because the key combination system varies from combination to combination (Ganin et al, 2017).

Modified algorithm for all bit combination responsible for the cluster number.

The use of any combination of bits responsible for the cluster number greatly extends capabilities of lexicographic PC decoding. However, its implementation requires knowing the entire codebook. This completely breaks down the concept of image compression code. Nevertheless, to evaluate the proposed approaches, this modification will be realized in the context of simulation modelling (Ganin et al, 2017).

Results.

Based on the developed simulation models, the potential characteristics of the proposed algorithms for lexicographic PC decoding are studied in comparison with known decoding schemes for such codes. Codes with the lengths of $n=8$, $n=16$, $n=32$, which to the fullest extent meet the requirements of the management control system tested. Figure 7 shows testing results of simulation models in observance of the provision for acceptable statistical errors (Gladkikh et al, 2016; Gladkikh et al, 2017; Gladkikh et al, 2017).

With the block length of $n=8$ the proposed LD algorithm shows the best correcting capability. This is due to the small value of the code vector. The potential LD characteristics for the PC code (32,16) are much worse than for BP and AWF, because the model for breaking allowed codewords into 4-bit clusters has been used to reduce computational costs. But for non-systematic codes the cluster length is equal to $f > n/2$ in this case. With reduction of cluster length, the correcting capacity of LD becomes comparable with the relevant BP parameter. By varying length of the code vector and decreasing number of information bits, the LD PC (16.5) shows similar characteristics with BP. In this case, an increase in the correcting capability is associated with an increase in the number of redundant symbols. Changes in LD PC characteristic while preserving the length of the code vector and increasing in the number of information bits are stipulated by the expected trend towards degradation of correcting capability characteristic for all decoding schemes. All the considered methods and algorithms for their implementation turned out to be effective with respect to SCD and the Tal-Vardy algorithm in the range of parameter ratios from 0 to 3 dB. The poor corrective characteristics of SCD in the range of low signal-to-noise ratios are due to avalanche-like error propagation associated with the sequence of computational process. The Tal-Vardy algorithm shows poorer performance in the same range of communication channel due to appearance of false branches. At higher values of this parameter, the proposed algorithms increase their efficiency, and the reason for this is the manifestation of the effect of minimizing the occurrence of false/erroneous symbols of the previous steps and branches in SCD implementation procedures and the Tala-Vardy algorithm. (Gladkikh et al, 2016; Korada, 2009; Tal & Vardy, 2011; Tal & Vardy, 2013).

Figures 7 and 8 shows the comparative characteristics of the classical PC lexicographic decoding algorithm with its modified versions for PC (8-4) and PC (32-16) respectively.

1. LD is the PC permutation decoding algorithm based on the lexicographic approach;
2. LDsfc is the modified algorithm taking into account the structural features of code combinations;
3. LDmax is the modified algorithm with based on the cluster number with the largest value of soft symbol solutions;
4. LDdcn is the modified algorithm based on the cluster number spacing over the entire length of the code combination;
5. LDall is the modified algorithm for all combinations of bits responsible for the cluster number;

6. LDmix is the modified algorithm combining several modifications (SFC and cluster number with the highest SDS value).

Figure 7 – Results of PC (8,4) simulation modelling.

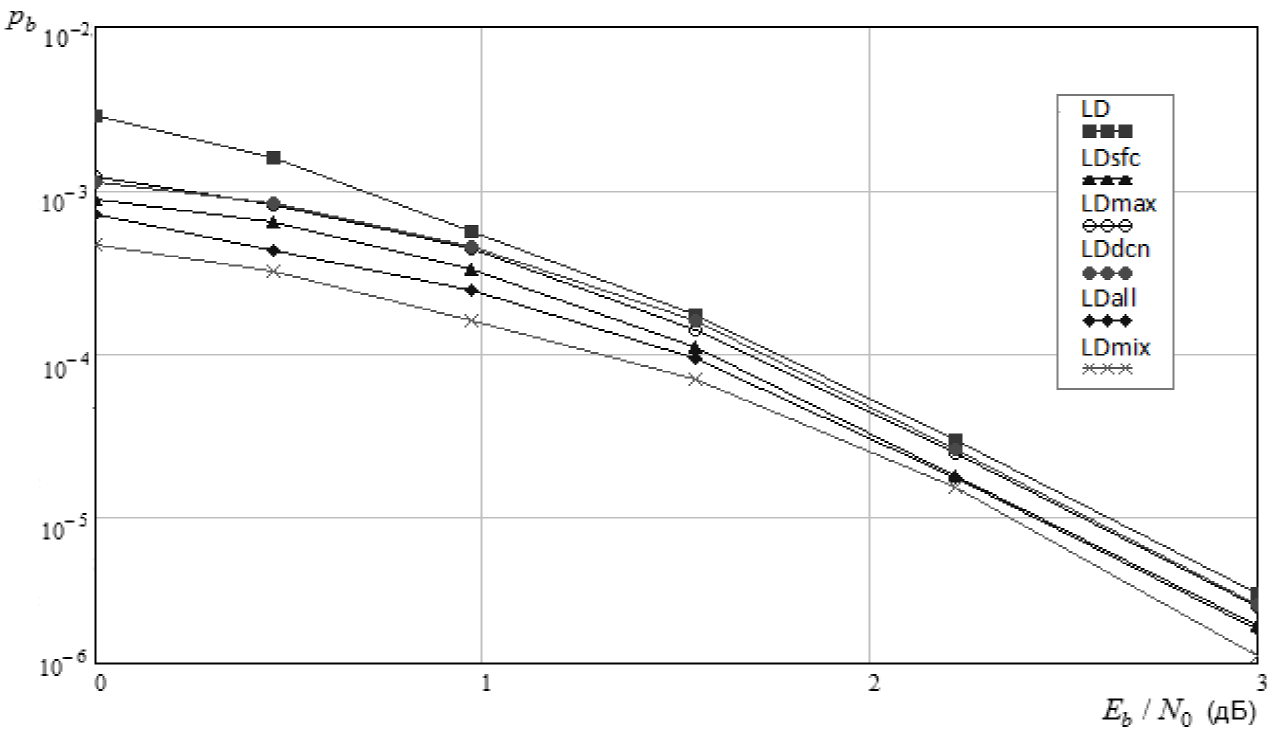
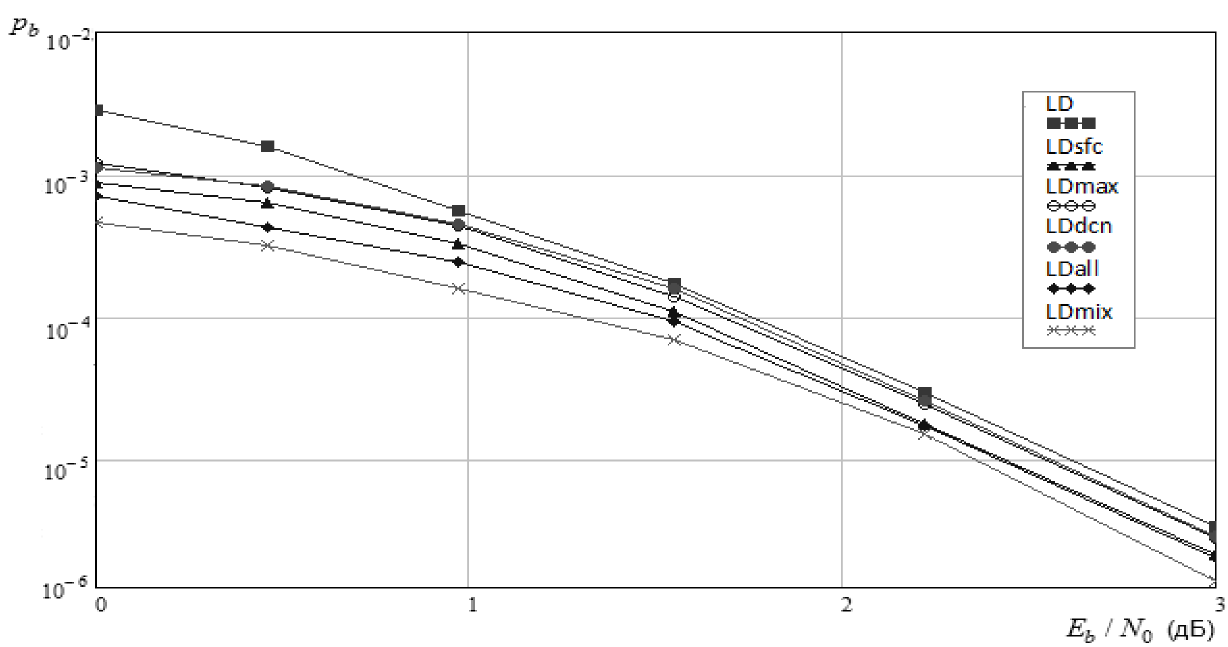


Figure 8 – Results of PC (32,16) simulation modelling.



As can be seen from above, the largest energy gain of about $0{,}7 \div 1$ dB is observed at signal-to-noise ratio $0 \div 1$ dB, while in the range of $1 \div 3$ dB the gain is about $0{,}2 \div 0{,}5$ dB. This is due to a reduction in the number of erroneous symbols with high values of SDS index and with improving interference environment. Based on the results of simulation modelling, the following energy modifications have the greatest energy gain:

1. LDall is a modified algorithm for all combinations of bits that are responsible for the cluster number – $0{,}2 \div 0{,}6$ dB, depending on parameters of a communication channel;
2. LDall is the modified algorithm for all combinations of bits that are responsible for the cluster number – $0{,}3 \div 0{,}75$ dB, depending on parameters of a communication channel (not suitable for code compression);
3. LDmix is a modified algorithm combining several cluster modifications (SFC and cluster number with the highest SDS value) – $0{,}5 \div 1$ dB, depending on parameters of a communication channel.

Discussion.

A comparative analysis of the above characteristics allows concluding that the proposed approach and its modifications in the range of low signal-to-noise ratio is prospective. The received data does not significantly change the decoder logic (Ganin et al, 2017).

The control unit is responsible for inclusion of additional blocks or their combinations to ensure a higher correcting capacity of the algorithm in question.

CONCLUSIONS.

Thus, it can be concluded that the use of the lexicographic approach to PC decoding makes it possible to improve correction characteristics (BER) of this class of block error correcting codes in the range of low signal-to-noise ratios. Implementation of various modifications provides an additional energy gain and significantly expands prospects and application scope of PC. Individual application of the PC corresponds to a great extend with the known data reliability requirements for modern and prospective information management complexes. However, the attractiveness of the proposed PC decoding algorithms in the range of small values reflects the expediency of using such codes in composition with 2D codes and above.

BIBLIOGRAPHIC REFERENCES.

1. Arikan, E. (2009). Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. IEEE Transactions on Information Theory, 55(7), 3051–3073.
2. Arikan, E. (2011). Systematic polar coding. IEEE communications letters, 15(8), 860–862.
3. Arikan, E., & Telatar, E. (2009, June). On the rate of channel polarization. 2009 IEEE International Symposium on Information Theory, 1493–1495.
4. Chilikhin, N.Y. (2014). Efficient decoding of Reed-Muller codes and polar codes based on the cluster approach. Modern problems of design, production and operation of radio engineering systems, UlGTU, 9, 79–82.
5. Chilikhin, N.Y. (2015). Structural features of codewords of polar codes and reed-Muller codes. Modern problems of design, production and operation of radio engineering systems, 1-2(9), 139–141.
6. Coetzee, J., & Mammen, K.J. (2017). Science and Engineering Students' Difficulties with Fractions at Entry-Level to University. International Electronic Journal of Mathematics Education, 12(3), 281–310.
7. Ganin, D.V., Namestnikov, S.M., & Chilikhin, N.Y. (2017). Modified algorithms for lexicographic decoding of polar codes in the image processing system. Vestnik NNSUEE, 11(78), 7–22.

8. Gladkikh, A.A., & Chilikhin, N.Y. (2013). The formation of soft solutions in broadband communication channel with QPSK-QAM. Automation of control processes, 3(33), 75–79.

9. Gladkikh, A.A., & Chilikhin, N.Y. (2014). Efficient decoding of binary block codes, Problems of telecommunications engineering and technology. Problems of telecommunications engineering and technology, 1, 71–73.

10. Gladkikh, A.A., & Klimov, R.V. (2013). Numerical modelling of the generalized procedure of forming the indexes soft solutions. Infocommunication technologies, 11(2), 22–28.

11. Gladkikh, A.A., Baskakova, E.S., Maslov, A.A., & Tamrazjan, G.M. (2013). Efficient decoding of nonbinary codes, with the provocation of the erased element. Automation of control processes, 2(32), 87–93.

12. Gladkikh, A.A., Chilikhin, N.Y., & Al-Tamimi, T.F.Kh. (2017). Ensuring the reliability of data in measuring systems based on sensory. Information-measuring and control systems, 15(12), 4–10.

13. Gladkikh, A.A., Chilikhin, N.Y., Namestnikov, S.M., & Ganin, D.V. (2015). Unification of algorithms of decoding of redundant codes in the system integrated information-control complexes. Automation of control processes, 1(39), 13–20.

14. Gladkikh, A.A., Klimov, R.V., & Chilikhin, N.Y. (2016). Metody effektivnogo dekodirovaniya izbytochnykh kodov i ikh sovremennye prilozheniya [Methods for efficient decoding of redundant codes and their modern applications]. Ulyanovsk: UlGTU.

15. Klimov, R.V., & Solodovnikova, D.N. (2014). Methods of formation of indexes of soft decision symbols based on a modification of the parameters of the channel with erasures. Radioengineering, 11, 90–93.

16. Korada, S.B. (2009). Polar codes for channel and source coding. PhD thesis. Ecole Polytechnique Federale de Lausanne.

17. Namestnikov, S.M., & Chilikhin, N.Y. (2017). Multidimensional code constructions using the Bhattacharya distance. Automation of control processes, 2(48), 28–34.

18. Tal, I., & Vardy, A. (2011). List decoding of polar codes. Proc. of the IEEE Intern. Symp. on Information Theory (ISIT), 1–5.

19. Tal, I., & Vardy, A. (2013). How to construct polar codes. IEEE Trans. Inform. Theory, 59(10), 6562–6582.

DATA OF THE AUTHORS.

1. Nikolay Yu. Chilikhin. Ph. D. (Engineering). Associate Professor of the chair Infocommunication Technologies and Communication Systems; Nizhny Novgorod State University of Engineering and Economics, Knyaginino (Russia).

2. Dmitrii V. Ganin. Ph.D. (Economy), Associate Professor; Vice-rector on scientific work and innovative development; Nizhny Novgorod State University of Engineering and Economics, Knyaginino (Russia).

3. Roman V. Klimov. Assistant Professor of the chair Infocommunication Technologies and Communication Systems; Nizhny Novgorod State University of Engineering and Economics, Knyaginino (Russia).

4. Aleksey A. Shamin, Ph.D. (Economy); Director of the Institute for Information Technologies and Communication Systems, Nizhny Novgorod State University of Engineering and Economics, Knyaginino (Russia).

5. Aleksey B. Kuzmichev. Ph. D. (Engineering); Associate Professor of the chair Applied Mathematics and Informatics, Togliatti State University, Tolyatti (Russia).

RECIBIDO: 6 de febrero del 2019. **APROBADO**: 25 de febrero del 2019.