Method of formation shift indexes vector by minimization of polynomials

© Liubov Berkman[1], © Serhiy Otrokh[2], © Valeriy Kuzminykh[2] and © Olena Hryshchenko[1]

[1] State University of Telecommunications, Kyiv, Ukraine
[2] National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
berkmanlubov@gmail.com, 2411197@ukr.net, vakuz0202@gmail.com, elena.grischenko1@gmail.com

Abstract. In this article, the 2 methods of formation shift indexes vector of the ring code are proposed. This article presents the matrix of the ring code in the form of binary elements (0,1) and in the form of polynomials, which consist of the sum of the arguments x of a certain category corresponding to the binary value of 1 code sequences of the ring code. With these two methods, shift indexes vectors are created using an example of a ring code of 7 × 7, each row containing 4 units and 3 zeros. The first method makes it possible to form VPS by implementing logical transformations (XOR, OR, or AND) over the binary elements of the generating matrix of the ring code. The second method makes it possible to form VPS by implementing logical transformations (XOR, OR, or AND) over the arguments x of a certain category corresponding to the binary value of 1 code sequences of the ring code. According to the proposed second method, as a result of the logical transformations of the XOR, OR, or AND arguments of polynomials of the matrix G, similar shift indexes vectors were formed. The first method allows developing an algorithm and implementing a program implementation of the formation shift indexes vector. The second method is more visual, by means of which you can mathematically describe the sequence of the formation shift indexes vector of the ring code.

Keywords: matrix of ring code, polynomial, logical transformations of the XOR, OR, or AND arguments, shift indexes vector.

1 Introduction

The ring code is constructed on the principle of block-cyclic codes [8], the rows of the forming matrices of which are linked by the condition of cyclicity.

Cyclic codes have been widely used due to their efficiency in detecting and correcting errors [1, 2].

Cyclic codes are a subset of linear block codes. Linear *(n, k)* - Code C is called cyclic if the cyclic shift of any word c is also a word of that code. The cyclic shift of the code word $c=(c_0, c_1, \ldots c_{n-1})$ corresponds to the shift of all elements of the word

one position to the right, resulting in a code word $c^{(1)}= (c_{n-1}, c_0, c_{1,}\dots c_{n-2})$. As a result of the *i*- multiple shift, we get the code word $c^{(i)}= (c_{n-i}, \dots, c_{n-1}, c_0, c_{1,}\dots c_{n-i-1})$.

The schematics of the encoding and decoding devices for these codes are extremely simple and are based on conventional shift registers. An example of a feedback register is shown in fig. 1.

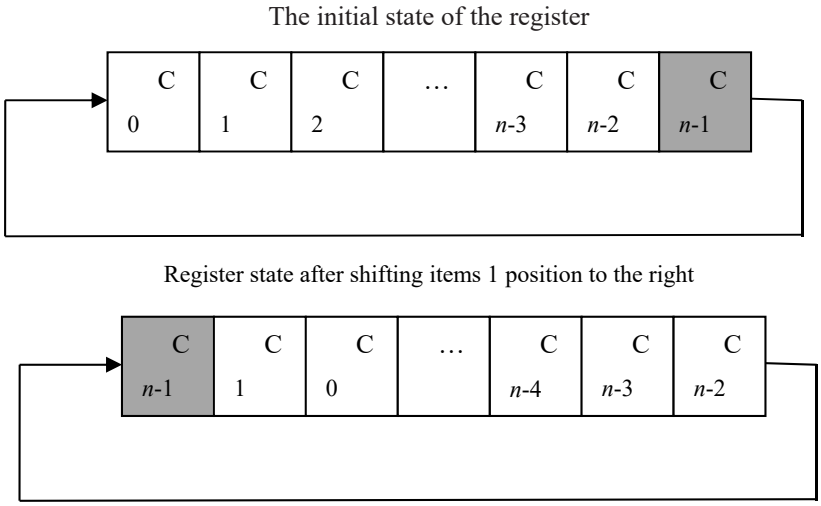

Fig. 1. The feedback register.

It is convenient to consider cyclic codes by presenting a combination of binary code not as sequences of zeros and ones, but as a polynomial from a dummy variable *x*.

The code word $c=(c_0, c_{1,}\dots c_{n-1})$ can be represented as the following polynomial:

$$c(x) = c_0x^0 + c_1x^1 +\dots+ c_{n-1}x^{n-1} = c_0 + c_1x +\dots+ c_{n-1}x^{n-1} \quad (1)$$

Where c_i – are the numbers of the given system of calculus (in binary system 0 and 1).

The above comparison of code words and polynomials is unambiguous: each word corresponds to a polynomial and each polynomial corresponds to a word that is determined by the coefficients of that polynomial.

If $c=(c_0, c_{1,}\dots c_{n-1})$ is a code word belonging to code *C*, then the corresponding polynomial c (x) is also called a code word C [3, 4].

According to the definition of the cyclic code for constructing the forming matrix $G_{n,\ k}$ it is sufficient to select only one initial n-degree combination $C_i(x)$. A cyclic shift can produce $(n - 1)$ different combinations, of which any k combinations can be taken as starting points By summing the rows of the forming matrix in all possible combinations, other code combinations can be obtained.

Thus, the cyclic (n, k) - code is a set of polynomials forming a k-dimensional linear subspace of the space of all polynomials of degree n-1 closed with respect to the cyclic shift operation [5].

The cyclic shift of the combination with the unit in the higher n-th digit is identical to the multiplication of the corresponding polynomial by x with the simultaneous subtraction from the result of the polynomial $(x^n - 1)$ or $(x^n + 1)$, since the operations are performed by module two. According to [6], the code combination of the cyclic (n, k)- code can be obtained in two ways:

1) by multiplying a simple code combination of degree $(k - 1)$ by the monomial x^{n-k} and adding to this multiplication the residue obtained from dividing the resulting multiplication by the polynomial $P(x)$ of degree $(n - k)$,

2) by multiplying a simple code combination of degree $(k - 1)$ by the forming polynomial $P(x)$ of degree $(n – k)$.

According to the first encoding method, the first k symbols of the resulting code combination match the corresponding symbols of the original simple code combination.

According to the second method, in the code combination obtained, the information symbols do not always coincide with the symbols of the original simple combination. This method is easy to implement, but because the resulting code combinations do not contain information symbols in explicit form, it complicates the decoding process.

In practice, the first method of obtaining a cyclic code is usually used.

The cyclic code can be represented as a formative matrix $P_{n,k}$, which consists of two submatrices: information U_k and additional H_p:

$$P_{n,k} = |U_k, H_p| \tag{2}$$

Information submatrix U_k, is a square unit matrix with the number of rows and columns equal to k. The additional sub-matrix H_p contains $p = n - k$ columns and k rows and is formed by residues $R(x)$[3].

The forming matrix allows k code combinations to be obtained. Other combinations are formed by summing the modulus of two rows of the forming matrix in all possible combinations.

The forming matrix $P_{n,k}$ can also be formed by multiplying the forming polynomial $P(x)$ of degree $\rho = n - k$ by the monomial x^{k-1} of the following

k-1 shifts of the resulting combination. The number of rows and columns in the formative matrix may be arbitrary.

The forming matrix of ring code unlike the cyclic code is always a square matrix of size $N \times N$, each row of which contains m units and, accordingly, $N – m$ zeros. Each row of the forming matrix of cyclic code has the same number of elements and the same structure of unitary and null character combinations. The first row of the forming matrix of cyclic code is called the initial vector, or the initial sequence, and the last row is the final vector of the cyclic shift of the code sequence elements. In this case, the rows of the matrix seem to form a ring of a complete cycle of shift of elements of code sequences [7].

2 An example of a matrix representation of a ring code

For example is a ring code, the shift of elements of the code sequences, which is carried out, from right to left, with the leftmost symbol being transferred to the right at the end of the code sequence each time. Below are the forming of ring code matrices that contain elements of code sequences in a binary number system. The forming matrix G has a size of 7×7 and contains 4 units and 3 zeros, and the forming matrix P has a size of 9 x 9 P contains 4 units and 5 zeros. The first rows of the forming matrices are called the initial sequence.

$$G = \begin{bmatrix} 0101011 \\ 1010110 \\ 0101101 \\ 1011010 \\ 0110101 \\ 1101010 \\ 1010101 \end{bmatrix} \tag{3}$$

$$P = \begin{bmatrix} 010010011 \\ 100100110 \\ 001001101 \\ 010011010 \\ 100110100 \\ 001101001 \\ 011010010 \\ 110100100 \\ 101001001 \end{bmatrix} \tag{4}$$

The above forming matrices can also be represented in the form of polynomials that correspond to the code sequences represented in the binary number system and where x – is an argument of a particular digit corresponding to the binary value of 1 ring code. The structure of the code sequences of the forming matrix G_1 corresponds to the structure of the code sequences in the binary number system the forming matrix G, the structure of the code sequences of the forming matrix P_1 corresponds to the structure of the code sequences in the binary number system the forming matrix:

$$G_1 = \begin{bmatrix} x^5 + x^3 + x^1 + x^0 \\ x^8 + x^5 + x^2 + x^1 \\ x^5 + x^3 + x^2 + x^0 \\ x^6 + x^4 + x^3 + x^1 \\ x^5 + x^4 + x^2 + x^0 \\ x^6 + x^5 + x^3 + x^1 \\ x^6 + x^4 + x^2 + x^0 \end{bmatrix} \tag{5}$$

$$P_1 = \begin{bmatrix} x^7 + x^4 + x^1 + x^0 \\ x^6 + x^4 + x^2 + x^1 \\ x^6 + x^3 + x^2 + x^0 \\ x^7 + x^4 + x^3 + x^1 \\ x^8 + x^5 + x^4 + x^2 \\ x^6 + x^5 + x^3 + x^0 \\ x^7 + x^6 + x^4 + x^1 \\ x^8 + x^7 + x^5 + x^2 \\ x^8 + x^6 + x^3 + x^0 \end{bmatrix} \quad (6)$$

Analysis of the structure of the polynomials of the G_1 and P_1 forming matrices indicates that the number of polynomials in the forming matrices depends on the number of elements in the code sequence, and, accordingly, on the number of code sequences. At that time, the number of members in each polynomial depends only on the number of units in the code sequence and does not depend on the number of elements in the code sequence.

Using the above matrices, we form the shift indexes vectors of the ring code by performing the binary transformations of *XOR*, *AND* and *OR* over the binary elements of the code sequences of the matrices G and P, as well as over the arguments of the polynomials of the matrices G_1 and P_1.

3 The method of formation of shift indexes vector by performing logical transformations over the binary elements of the matrices G and P

The shift indexes vector is the sequence of decimal numbers formed by summing the number of units resulting from the implementation of one of the binary transformations *XOR*, *AND*, *OR* (with or without negation) of the elements of the initial sequence (first line) of the ring code and the rest of its lines.

According to [8] the shift indexes vector (hereinafter referred to as SIV) is formed in three stages:

1) Alternatively, the binary logical transformation of *XOR*, *OR*, AND elements of the first line and the next rows of the matrix of the ring code, placed at the same positions of the code sequences, is performed.

2) The binary elements (0, 1), resulting from the logical transformation are written to the shift indexes matrix.

3) The number of units is calculated in each row of the resulting shift indexes matrix (hereinafter referred to as the SIM). In doing so, the SIM is transformed into SIV.

Tables 1 - 3 present the sequence of transformations of ring code a 7 × 7, each row containing 4 units and 3 zeros, in the SIV using logical transformations *XOR*, *OR*, *AND*.

Table 1. The sequence of the formation of SIV by logical transformation *XOR*.

Line number ring code	Structures of code sequences	Line numbers between elements of which the operation was performed *XOR*	Structures forming vectors	The sums of the single elements of the forming vectors
1	0 1 0 1 0 1 1	1 - 2	1 1 1 1 1 0 1	6
2	1 0 1 0 1 1 0	1 - 3	0 0 0 0 1 1 0	2
3	0 1 0 1 1 0 1	1 - 4	1 1 1 0 0 0 1	4
4	1 0 1 1 0 1 0	1 - 5	0 0 1 1 1 1 0	4
5	0 1 1 0 1 0 1	1 - 6	1 0 0 0 0 0 1	2
6	1 1 0 1 0 1 0	1 - 7	1 1 1 1 1 1 0	6
7	1 0 1 0 1 0 1			

Table 2. The sequence of the formation of the SIV by logical transformation *AND*.

Line number ring code	Structures of code sequences	Line numbers between elements of which the operation was performed *AND*	Structures forming vectors	The sums of the single elements of the forming vectors
1	0 1 0 1 0 1 1	1 - 2	0 0 0 0 0 1 0	1
2	1 0 1 0 1 1 0	1 - 3	0 1 0 1 0 0 1	3
3	0 1 0 1 1 0 1	1 - 4	0 0 0 1 0 1 0	2
4	1 0 1 1 0 1 0	1 - 5	0 1 0 0 0 0 1	2
5	0 1 1 0 1 0 1	1 - 6	0 1 0 1 0 1 0	3
6	1 1 0 1 0 1 0	1 - 7	0 0 0 0 0 0 1	1
7	1 0 1 0 1 0 1			

Table 3. The sequence of the formation of SIV using logical OR transformation.

Line number ring code	Structures of code sequences	Line numbers between elements of which the operation was performed *OR*	Structures forming vectors	The sums of the single elements of the forming vectors
1	0 1 0 1 0 1 1	1 - 2	1 1 1 1 1 1 1	7
2	1 0 1 0 1 1 0	1 - 3	0 1 0 1 1 1 1	5
3	0 1 0 1 1 0 1	1 - 4	1 1 1 1 0 1 1	6
4	1 0 1 1 0 1 0	1 - 5	0 1 1 1 1 1 1	6
5	0 1 1 0 1 0 1	1 - 6	1 1 0 1 0 1 1	5
6	1 1 0 1 0 1 0	1 - 7	1 1 1 1 1 1 1	7
7	1 0 1 0 1 0 1			

Tables 4 - 6 present the sequence of transformations of ring code a 9×9, each row containing 4 units and 5 zeros, in the SIV using logical transformations *XOR*, *OR* and *AND*.

Table 4. The sequence of the formation of the SIV by logical transformation *XOR*.

Line number ring code	Structures of code sequences	Line numbers that have binary transformations between elements	Structures forming vectors	The sums of the single elements of the forming vectors
1	0 1 0 0 1 0 0 1 1	1 - 2	110110101	6
2	1 0 0 1 0 0 1 1 0	1 - 3	011011110	6
3	0 0 1 0 0 1 1 0 1	1 – 4	000001001	2
4	0 1 0 0 1 1 0 1 0	1 – 5	110100111	6
5	1 0 0 1 1 0 1 0 0	1 – 6	011111010	6
6	0 0 1 1 0 1 0 0 1	1 – 7	001000001	2
7	0 1 1 0 1 0 0 1 0	1 – 8	100110111	6
8	1 1 0 1 0 0 1 0 0	1 – 9	111011010	6
9	1 0 1 0 0 1 0 0 1			

Table 5. The sequence of the formation of SIV using logical *AND* transformation.

Line number ring code	Structures of code sequences	Line numbers that have binary transformations between elements	Structures forming vectors	The sums of the single elements of the forming vectors
1	0 1 0 0 1 0 0 1 1	1 - 2	000000010	1
2	1 0 0 1 0 0 1 1 0	1 - 3	000000001	1
3	0 0 1 0 0 1 1 0 1	1 – 4	010010010	3
4	0 1 0 0 1 1 0 1 0	1 – 5	000010000	1
5	1 0 0 1 1 0 1 0 0	1 – 6	000000001	1
6	0 0 1 1 0 1 0 0 1	1 – 7	010010010	3
7	0 1 1 0 1 0 0 1 0	1 – 8	010000000	1
8	1 1 0 1 0 0 1 0 0	1 – 9	000000001	1
9	1 0 1 0 0 1 0 0 1			

Table 6. The sequence of the formation of SIV using logical *OR* transformation.

Line number ring code	Structures of code sequences	Line numbers that have binary transformations between elements	Structures forming vectors	The sums of the single elements of the forming vectors
1	0 1 0 0 1 0 0 1 1	1 - 2	110110111	7
2	1 0 0 1 0 0 1 1 0	1 - 3	011011111	7
3	0 0 1 0 0 1 1 0 1	1 – 4	010011011	5
4	0 1 0 0 1 1 0 1 0	1 – 5	110110111	7
5	1 0 0 1 1 0 1 0 0	1 – 6	011111011	7
6	0 0 1 1 0 1 0 0 1	1 – 7	011010011	5
7	0 1 1 0 1 0 0 1 0	1 – 8	110110111	7
8	1 1 0 1 0 0 1 0 0	1 – 9	111011011	7
9	1 0 1 0 0 1 0 0 1			

4 The method of forming a shift indexes vector by making logical transformations over the arguments of polynomials of the matrix G_1

Formation of the SIV by making logical transformations over the arguments of polynomials is carried out as follows:

1) Alternately the binary logical transformation *XOR*, *OR*, or AND of the arguments of the first polynomial and subsequent polynomials of the ring code G_1 matrix is performed.

2) Depending on the type of logical transformation (*XOR*, *OR* or *AND*) the following steps are performed:

a) For logical *XOR* transformation:

- paired arguments of the generated polynomial with identical digits are deleted;
- counts the number of arguments left after deletion paired arguments with equal digits;

b) For logical AND transformation:

- remove the odd arguments of the polynomial;
- paired arguments with equal digits are absorbed by one argument;
- counts the number of arguments left after deletion and absorbing arguments;

c) For OR logical transformation:

- paired arguments with equal digits are absorbed by one argument;
- counts the number of arguments left after the arguments are absorbed.

Tables 7 - 9 show the sequence of transformations of ring code a 9 × 9, each row containing 4 units and 5 zeros by performing logical transformations over the arguments of the polynomials of the forming matrix G_1.

Table 7. The sequence of the formation of SIV by logical transformation *XOR*.

Line number RC	The ring code matrix	Numbers rows matrix RC	Shift indexes matrix	Result addition modulo 2	Total number elements SIV
1	$x^5+x^3+x^1+x^0$	1 - 2	$x^5+x^3+\cancel{x^1}+x^0$ $x^6+x^4+x^2+\cancel{x^1}$	$x^5+x^3+x^0$ $x^6+x^4+x^2$	6
2	$x^6+x^4+x^2+x^1$	1 - 3	$\cancel{x^5}+\cancel{x^3}+x^1+\cancel{x^0}+$ $\cancel{x^5}+\cancel{x^3}+x^2+\cancel{x^0}$	x^1+x^2	2
3	$x^5+x^3+x^2+x^0$	1 - 4	$x^5+\cancel{x^3}+x^1+\cancel{x^0}+$ $x^6+\cancel{x^3}+x^2+\cancel{x^0}$	$x^5+x^1+x^6+x^2$	4
4	$x^6+x^3+x^2+x^0$	1 - 5	$\cancel{x^5}+x^3+x^1+$ $\cancel{x^0}+\cancel{x^5}+x^4+x^2+$ $\cancel{x^0}$	$x^3+x^1+x^4+x^2$	4
5	$x^5+x^4+x^2+x^0$	1 - 6	$\cancel{x^5}+\cancel{x^3}+\cancel{x^1}+x^0+$ $x^6+\cancel{x^5}+\cancel{x^3}+\cancel{x^1}$	x^0+x^6	2
6	$x^6+x^5+x^3+x^1$	1 - 7	$x^5+x^3+x^1+\cancel{x^0}$ $x^6+x^4+x^2+\cancel{x^0}$	$x^5+x^3+x^1+$ $x^6+x^4+x^2$	6
7	$x^6+x^4+x^2+x^0$				

Table 8. The sequence of the formation of SIV by logical transformation *AND*.

Line number RC	The ring code matrix	Numbers rows matrix RC	Shift indexes matrix	Result conjunctive absorption	Total number elements SIV
1	$x^5+x^3+x^1+x^0$	1 - 2	$\cancel{x^5}\wedge\cancel{x^3}\wedge x^1\wedge\cancel{x^0}\wedge$ $\cancel{x^6}\wedge\cancel{x^4}\wedge\cancel{x^2}\wedge x^1$	$x^1\wedge x^1=x^1$	1
2	$x^6+x^4+x^2+x^1$	1 - 3	$x^5\wedge x^3\wedge\cancel{x^1}\wedge x^0\wedge$ $x^5\wedge x^3\wedge\cancel{x^2}\wedge x^0$	$x^5\wedge x^5=x^5$ $x^3\wedge x^3=x^3$ $x^0\wedge x^0=x^0$	3
3	$x^5+x^3+x^2+x^0$	1 - 4	$\cancel{x^5}\wedge x^3\wedge\cancel{x^1}\wedge x^0\wedge$ $\cancel{x^6}\wedge x^3\wedge\cancel{x^2}\wedge x^0$	$x^3\wedge x^3=x^3$ $x^0\wedge x^0=x^0$	2
4	$x^6+x^3+x^2+x^0$	1 - 5	$x^5\wedge\cancel{x^3}\wedge\cancel{x^1}\wedge x^0\wedge$ $x^5\wedge\cancel{x^4}\wedge\cancel{x^2}\wedge x^0$	$x^5\wedge x^5=x^5$ $x^0\wedge x^0=x^0$	2
5	$x^5+x^4+x^2+x^0$	1 - 6	$x^5\wedge x^3\wedge x^1\wedge\cancel{x^0}\wedge$ $\cancel{x^6}\wedge x^5\wedge x^3\wedge x^1$	$x^5\wedge x^5=x^5$ $x^3\wedge x^3=x^3$ $x^1\wedge x^1=x^1$	3
6	$x^6+x^5+x^3+x^1$	1 - 7	$\cancel{x^5}\wedge\cancel{x^3}\wedge\cancel{x^1}\wedge x^0$ $\cancel{x^6}\wedge\cancel{x^4}\wedge\cancel{x^2}\wedge x^0$	$x^0\wedge x^0=x^0$	1
7	$x^6+x^4+x^2+x^0$				

Table 9. The sequence of the formation of SIV by logical transformation *AND*.

Line number RC	The ring code matrix	Numbers rows matrix RC	Shift indexes matrix	Result disjunctive absorption	Total number elements SIV
1	$x^5+ x^3+ x^1+ x^0$	1 - 2	x^5v x^3v x^1vx^0v x^6v x^4v x^2v x^1	x^1v x^1= x^1 (x^5vx^3vx^1vx^0vx^6vx^4v x^2)	7
2	$x^6+ x^4+ x^2+ x^1$	1 - 3	x^5v x^3v x^1v x^0v x^5v x^3v x^2v x^0	x^5v$x^{5=}$ x^5 x^3v$x^{3=}$ x^3 x^0v$x^{0=}$ x^0 (x^5vx^3vx^2vx^1vx^0)	5
3	$x^5+ x^3+ x^2+ x^0$	1 - 4	x^5^ x^3^ x^1 ^ x^0^ x^6 ^x^3 ^ x^2 ^ x^0	x^3^$x^{3=}$ x^3 x^0^$x^{0=}$ x^0 (x^5vx^6vx^3vx^1vx^2vx^0)	6
4	$x^6+ x^3+ x^2+ x^0$	1 - 5	x^5 ^ x^3^ x^1^ x^0^ x^5^ x^4 ^ x^2 ^ x^0	x^5^$x^{5=}$ x^5 x^0^$x^{0=}$ x^0 (x^5vx^4vx^3vx^1vx^2vx^0)	6
5	$x^5+ x^4+ x^2+ x^0$	1 - 6	x^5 ^ x^3^ x^1^ x^0^ x^6 ^ x^5 ^ x^3^ x^1	x^5^$x^{5=}$ x^5 x^3^$x^{3=}$ x^3 x^1^$x^{1=}$ x^1 (x^5vx^6vx^3vx^1vx^0)	5
6	$x^6+ x^5+ x^3+ x^1$	1 - 7	x^5^ x^3^ x^1^ x^0 x^6^ x^4^ x^2^ x^0	x^0^$x^{0=}$ x^0 (x^5vx^6vx^3vx^4vx^1vx^2v x^0)	7
7	$x^6+ x^4+ x^2+ x^0$				

As noted above, polynomials of the forming matrices G_1 and P_1 consist of arguments whose degrees correspond to the degrees of the binary elements of the code sequences of the forming matrices G and P of the ring code. The analysis of the above tables indicates that the algorithm for converting the ring code represented in the form of polynomials is similar to the algorithm for converting the ring code represented as binary elements.

5 Conclusions

1. The ring code can be represented in the form of a matrix, each line of which is a polynomial with arguments of a certain digit corresponding to the binary value of 1 ring code.
2. The shift indexes vector can be formed by two methods:
- by making logical transformations (*XOR*, *OR* or *AND*) over the binary elements of the ring matrix-forming matrix;
- by performing logical transformations (*XOR*, *OR* or *AND*) over the arguments of the forming matrix of polynomials.

3. The first method allows you to develop an algorithm and perform software implementation of the formation of the shift indexes vector.
4. The second method is more visual and allows you to mathematically describe the sequence of formation of the shift indexes vector of the ring code.

References

1. D. Augot, E. Betti, E. Orsini.: An introduction to linear and cyclic codes. Journal of Symbolic Computation, 47-68 (2009).
2. L. M. J. Bazzi and S. K. Mitter.: Some randomized code constructions from group actions. IEEE Trans. on In. Theory, vol. 52, pp. 3210–3219 (2006).
3. J. Yuan, C. Ding, Senior Member.: IEEE Secret Sharing Schemes from Three Classes of Linear Codes, IEEE Trans. on Inf. Theory, 52 (1), 206-212 (2006).
4. A. Vardy.: Algorithmic complexity in coding theory and the minimum distance problem, STOC ’97: Proceedings of the twenty-ninth annual ACM symposium on Theory of Computing, pp. 92–109 (1997).
5. A. Ashikmin, A. Barg.: Minimal vectors in linear codes, IEEE Transactions on Information Theory 44 (5) (1998).
6. G. Castagnoli, J. L. Massey, P. A. Schoeller, and N. von Seeman.: On repeated root cyclic codes, IEEE Trans. on Inf. Theory, vol. 37, pp. 337-342 (1991).
7. Gavrilko E.V., Otrokh S.I., Yarosh V.A., Grishchenko L.N.: Improving the quality of the functioning of the network of the future through the use of ring codes. Vesnik svyazi (2), 60-64 (in Russian) (2018).
8. S. Otrokh, V. Kuzminykh, O. Hryshchenko.: Method of forming the ring codes. CEUR Workshop Proceedings, vol. 2318, pp. 188-198 (2018).