Ministry of transport and telecommunications of the Ukraine
State Administration of Communication
Odessa National Academy of Telecommunication named after A.S. Popov

Department of the Telecommunication theory named after A.G. Zuko

BASES OF THE ERROR-CONTROL CODES THEORY

Module №4

Education manual
on telecommunication theory

Odessa 2010

УДК 621.391 Plane of Education Materials Publication in 2010

Bases of the error-control codes theory: [education manual] / V. Banket, P. Ivaschenko, L. Borschova, D. Rozenvasser. – Odessa: ONAT named after A.S. Popov, 2010. – 96 p.

Editor of English language Ocheretnaya O.Yu.

This education manual contains main theoretical positions of Telecommunication Theory, chapter «Bases of the error-control codes theory», questions and tasks for examination of knowledge, methodical instructions and input data for course work, methodical guidelines for fulfilling laboratory works, short English-Russian and Russian-English dictionaries.

The manual is intended for students training on the direction 050903 – Telecommunications studying the Module 4 of Telecommunication Theory

APPROVED on Chair of
Telecommunication Theory
named after A.G. Zuko meeting.
Protocol № 9, March 25, 2010

CONTENTS

1 PURPOSE, STRUCTURE AND CLASSIFICATION OF ERROR-CONTROL CODES

1.1 Error-control codes in transmission systems

In theory of modern transmission systems the considerable attention to coding methods of information is given.

Coding – the operation of an identification of the symbols or groups of symbols from one code by symbols or groups of symbols to other code. Necessity of coding arises, first of all, from requirement to adapt a message form to the given communication channel or to any other device intended for transformation or storage of the information. The typical block diagram of digital telecommunication system is resulted on figure 1.1. The source produces messages which it is necessary to transfer through the channel of a telecommunication systems. It can be discrete messages (data, cable messages etc.) or continuous messages (speech, audio, TV, etc.), transformed to the digital signals. The real messages contain redundancy and for matching of the information source with a transmitting channel usually the source encoder is used. Together with decoder they form source codec. The source coding methods were studied in Module №2. The primary goal of any telecommunication system is the information transmitting with given fidelity and rate. These requirements are in contradict and increasing of information rate leads to decreasing of the noise immunity and transmitting fidelity. In agree with well known Shannon theorems, as is wished considerable increase of fidelity of information transfer if a transmitting rate through channel R_{ch} does not exceed the channel capacity C_{ch} basically is possible. It is reached by using of the enough long error-correcting codes.

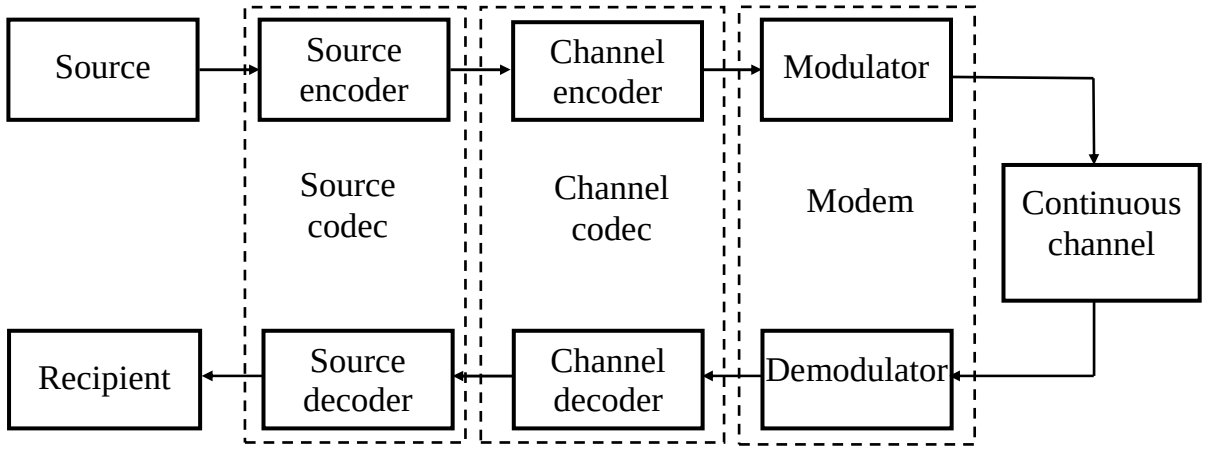

Figure 1.1 – Typical block diagram of a digital telecommunication system

Error-control code is the code which allow to **detect** and **correct errors** arising from messages transition in the communication channels. With this purpose the redundancy is entered into structure of an error-correcting codes. **Codec** of error-control code (channel **encoder** and **decoder**) is shown on figure 1.1.

In real conditions the length of a code is limited by admissible complexity of coding/decoding devices. Therefore the result from using of error-control codes depends on the code parameters and restrictions on realization of the channel codec.

The modern theory offers a wide set of error-control codes, various on structure, construction principles and **error detection and correction capability**. In the subsequent chapters the important classes of the codes with effective coding/decoding algorithms are considered.

1.2 Classification of error-control codes

The error-control codes can be classified to various signs. The structure of codes classification is resulted on figure 1.2. On a way of formation the error-control codes are subdivided on block and continuous codes. Formation of the **block codes** provides splitting of transferred digital sequences into separate blocks which move to encoder input. To each such block on an encoder output there corresponds the block of code symbols which work is defined by a rule named as the **coding algorithm.** Formation of the **continuous codes** is carried out continuously in time, without division into blocks as defines the name of this code class.

Block codes historically have been discovered and studied earlier, at beginning of coding theory development. In a class of continuous codes it is necessary to note the **convolutional codes** which exceed on characteristics of the block codes, and, for this reason, find wide application in a transmission systems.

Many codes carry names of scientists which have discovered and investigated them. Such example is the continuous Fink-Hagelbarger's code offered by Soviet scientist L. Fink and German expert R. Hagelbarger. Long time this code was in the literature as indicative example of a continuous code with simple encoding/decoding algorithms, but after elaborating the convolutional code has given way to them. For the description of procedures of coding/decoding both block and convolutional codes usually use an adequate mathematical apparatus. For the description of **linear codes** the well developed **linear algebra** is used. Formation of **nonlinear codes** is made with application of nonlinear procedures. Such approach allows to construct in some cases nonlinear codes with a number of special properties. In the error-control coding theory the problem of **realisation complexity** of encoding/decoding procedure and in particular decoding procedure is important. Therefore some classes of codes (Hamming codes, Bose-Chaudhuri-Hochquenghem codes, Reed-Solomon codes etc.) have been developed together with the decoding algorithms connected with structural properties of these codes. And on contrary the elaborating of a new decoding algorithms for convolutional codes (Viterbi algorithm, sequential decoding, threshold decoding) initiated searches of corresponding codes. Distinctive advantages of error-control codes induced searches of new approaches to realization of ways to increase of noise immunity and efficiency of transmission systems. On figure 1.2 **new methods of encoding/decoding** (coded modulation, turbo-codes, time-space coding) are noted accordingly.

Questions

1.1 What is a purpose of the error-control coding by transmitting of a digital signal?

1.2 What elements does the codec of error-control code consists?

1.3 What is difference of coding procedures between block and continuous codes?

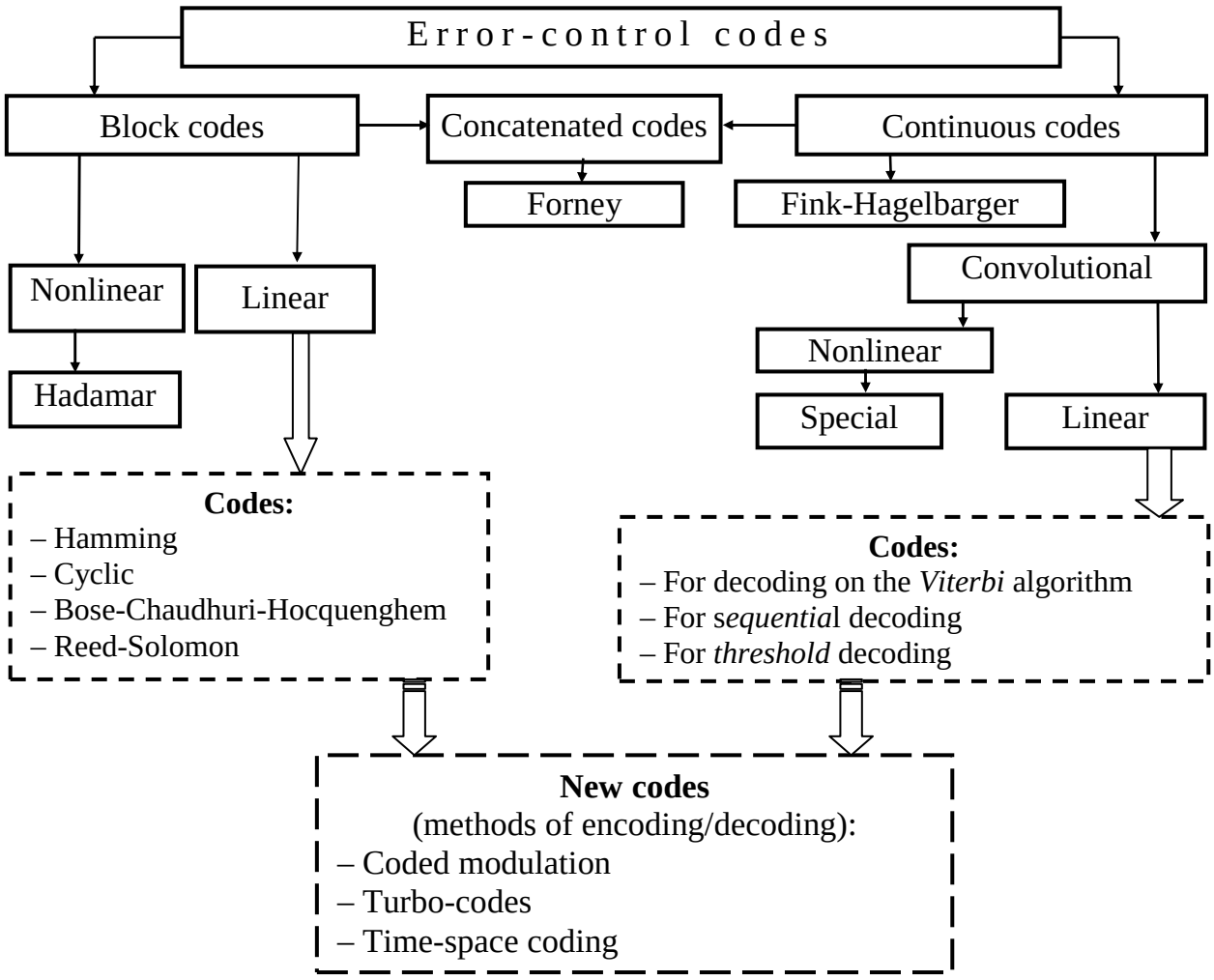

Figure 1.2– Structure of the code classification

Tasks

1.1 Represent the block diagram of telecommunication system and describe a purpose of its separate blocks.

1.2 Give classification of error-control codes by ways of formation and structural properties.

1.3 Give the scheme of inclusion of the encoder and decoder of a error-control code in the digital transmission system. Explain a purpose of scheme elements.

2 PARAMETERS OF BLOCK ERROR-CONTROL CODES

There are the following parameters of the block codes. The **size of a code alphabet** m is the number of the various symbols used by a coding. In practice the codes with $m = 2$ are used. These are **binary codes**. For construction of binary code word **binary alphabet** with symbols {0, 1} is used. Wide practical using of binary codes is defined for a reason of simplicity of binary logic elements construction in codec memory devices. The block code consists of set of fixed length vectors named code word. The **code word length** is the number of elements in the vector and is denoted with n.

Redundancy in the block code words can be entered as follows. Let on a block encoder input the block of **information symbols a** = $\{a_1, a_2, a_3, \ldots, a_k\}$ arrives. By the block coding **code word** on the encoder output can look like:

$$\mathbf{b} = \{a_1, a_2, a_3, \ldots, a_k, c_1, c_2, c_3, \ldots, c_{n-k}\},$$

where (c_1, c_2, c_3, …, $c_{n–k}$) – **additional symbols**. Values of additional symbols are defined by **coding rules.** Such code is called as **systematic code**. Each code word of length n symbols contains in a **systematic codes** k information symbols. Thus to an information symbols are added $r = n – k$ additional symbols which are depend on information symbols and used by the decoding for detection and correction of an errors. By **nonsystematic codes** information symbols in an explicit form in a code word do not contain.

The total quantity of the **possible code words** of the block error-control code is defined by the formula:

$$M = m^n = 2^n. \tag{2.1}$$

For a possibility of detection and correction of an errors these M code words not completely use for an information transfer. From these 2^n code words we may select $M_0 = 2^k$ code words ($k < n$) to the forming a code. Thus block of k information bits is mapped into a code word of length n selected from the set of $M_0 = 2^k$ code words. These words named as **allowed code word** as they are allowed for an information transfer. We refer the resulting **block code** as an **(*n*, *k*) code**, and the ratio

$$R_{\text{code}} = \frac{k}{n} \tag{2.2}$$

is defined to be the **code rate**.

The rate of a error-control code is defined also by the ratio

$$R_{\text{code}} = (\log_2 M_0)/(\log_2 M). \tag{2.3}$$

In **nonredundancy code** $M_0 = M$ (or $k = n$) and the rate is

$$R_{\text{code}} = 1. \tag{2.4}$$

Quantity of the allowed code words is equal $M_0 = 2^k$.

In the error-control code possible words are used not completely i.e. $M_0 < M$. It illustrates redundancy of a code. **Redundancy** of a systematic code K_{red} is a relative share of the number of additional symbols $n – k$ in a code word on its length n symbols:

$$K_{\text{red}} = 1 – R_{\text{code}} = (n–k)/n. \tag{2.5}$$

For simple (nonredundancy) code $n = k$, and $K_{\text{red}} = 0$.

Exercise 2.1 As it is known, in a binary channels under the noises and distortions there are an errors in form of transitions of a transferred symbols to an opposite symbols. For example, by transfer of a symbol 1 transition (1→0) is possible and accordingly transitions (0→1) are possible also. Consider the possibilities of the binary error-control code construction intended for transfer of messages with symbols

from alphabet with volume of M_A, and allowing by the receiving to detect the channel errors. Specify the encoding and decoding methods of such code. For the developed algorithm of a coding define rate and redundancy of such codes.

Instructions. The providing of an errors detection in the transmitted code words will be possible if for the allowed code words to give a forms which are changed by errors in symbols of this words. Then detection of errors (i.e. decoding) can be made by check of conformity of received words to this in advance known forms. At the first development times of the error-detecting codes the maintenance in the transmitted allowed words of «even number of unit symbols» was considered as simple way. So the **«Code with even number of units »** has been invented.

Decision. Consider a construction variant of the binary systematic code intended for transferring letters, chosen from the alphabet of volume M_A. According to above considered rule the information block $\mathbf{a} = \{a_1, a_2, a_3, a_4, \ldots, a_k\}$ of each word should contain k binary symbols a_i. The total quantity of information blocks should be precisely equal to volume of the source alphabet M_A. That is the equality $M_A = 2^k$ guarantees transfer of each source symbol and the corresponding to it code words of systematic code. The quantity of units in an information blocks depends from a primary simple code and can be both even and odd. It appears that for realization of encoding and decoding of such code words it is convenient to use the procedure «**module-2 addition**» [3]. This procedure defines the simple way to find of the parity of units number in a code word. To everyone information block we will attribute one additional symbol ($r = 1$) so that the quantity of units in again formed word was **even**. Encoding it is made in such sequence:

1 Let information block **a** is represented by a primary code: $\mathbf{a}_1 \rightarrow 101010$;

2 By consecutive module-2 addition of the primary code symbols defines an additional symbol $c = 1$;

3 We form allowed code words, finishing an additional symbol to the block of information symbols **b** = 1010101. It is visible, that the coding rule is carried out, since the number of units remains even;

4 By the other form of a primary code it is received: $\mathbf{a}_2 \rightarrow 101011$, $c = 0$ and $\mathbf{b}_2 = 1010110$;

5 It is obvious, that any transition ((1→0) or (0→1)) changes number of units in the received words. If by decoding to use procedure of calculation of units number it is possible to detect errors.

Remark. It appears, such code allows to detect not any **errors configurations.** The simple analysis shows, that two-multiple change of symbols cannot change parity and such errors in this code are impossible to detect. It is recommended to make such analysis for other variants of error combinations independently.

The rate and redundancy of a code with even units number and by parameters (k, $r = 1$, $n = k+r = k + 1$) are defined by formulas:

$$R_{\text{code}} = \frac{k}{n} = \frac{k}{k+1} \text{ and } K_{\text{red}} = \frac{n-k}{n} = \frac{1}{k+1}.$$

It is visible, that for the big lengths of the information block $k>>1$ rate of such code is close to $R_{code} = 1$, and redundancy by transfer for example letters from the Russian text with alphabet volume $M_A = 32$ ($k = 5$) will be small $K_{red} = 1/6$.

Questions

2.1 What is the reason of wide application of binary codes in transmission systems?

2.2 Is the placing of additional symbols in front of the block of information symbols in a systematic code possible? Will it change redundancy of a code?

3 ERROR DETECTION AND CORRECTION CAPABILITY OF BLOCK CODES

Let's establish dependence of detecting and correcting capability of the block codes from a code parameters. It is useful to consider a binary code with parameters $n = 3$, $k = 2$. All words of this code ($M = 8$) it is possible to divide by sign «parity of units number in a code words» on two groups:

– words with even number of units,

– words with odd number of units.

The code constructed by this principle named “Code with even number of units” is considered in the Exercise 2.1.

Example 3.1 A binary code ($m = 2$, $n = 3$) with even number of units.

In table 3.1 the full set of binary words ($m = 2$, $n = 3$, $M = 8$) is divided into a set of the allowed code words ($M_0 = 4$) containing words with even number of units (including the word 000 (number 0 – even)), and the set of the forbidden words with odd number of units. Their total quantity is equal to difference $M_{forbid} = M – M_0 = 4$.

Table 3.1 – Code with even number of units

Full set of a words ($M = 8$): {000, 001, 010, 011, 100, 101, 110, 111}	
The allowed code words (with even number of units), M_0=4: {000, 011, 101, 110}	The forbidden words (with odd number of units), $M_{forbid} = M – M_0 = 4$: {001, 010, 100, 111}
Code parameters: code rate R_{code} =1/2, code distance $d_{min} = 2$, code can detect $q_{det} = 1$ error	

Allowed code words are used for an information transfer through channel (are allowed for transfer).

Forbidden code words are not used for an information transfer through channel (are forbidden for transfer).

In the coding theory the concept «distance between code words» plays the important role. Everyone binary block error-control code are characterized by a parameter code distance. The code distance d_{min} is one of the major parameters of error-control codes.

The **code distance** of the binary error-control code $d_{\min}$ is the minimal Hamming distance [3] between the allowed code words. Let consider a pairs of allowed code words from table 3.1. It is possible to establish that for this code a minimal distance is $d_{\min} = 2$. Such distance allows to detect a single errors in the channel. If the transmitted code word is **b** = (1 1 0), and channel error is characterized by a word (**error vector***)* **e** = (0 1 0) the received word $\hat{\mathbf{b}}$ with error on the channel exit is defined by module-2 addition:

$$\mathbf{b} = 1\ 1\ 0,$$
$$\mathbf{e} = 0\ 1\ 0,$$
$$\hat{\mathbf{b}} = (\mathbf{a} \oplus \mathbf{e}) = 1\ 0\ 0.$$

From this it is visible, that the symbol «1» in error vector **e** changes a corresponding symbol in transmitted word **b** to an opposite symbol.

For the characteristic of quantity of channel errors enter concept the multiplicity of an errors. **Multiplicity** of an errors q is a quantity of the channel errors within a codeword. For example, for words from table 3.1 the error vector variants with multiplicity $q = 1$ are: **e** = 100, 010, 001. And the double errors are: 110, 011, 101.

The code capability to detect and to correct of errors depends from code distance $d_{\min}$.

Error detection is the fixing by decoding of an error presence of certain multiplicity in received word $\hat{\mathbf{b}}$.

Error correction is the detection by decoding of an errors in certain symbols of received words and their subsequent correction.

According to these definitions error-control codes are subdivided into following classes:

1 **Error-detecting codes** which detect a channel errors.

2 **Error-control codes** which correct a channel errors and named in literature as codes with direct correction of errors (i.e. with errors correction by a code methods).

The relation between code distance $d_{\min}$ and error control ability of a code we will establish on an example of code with even number of units (see table 3.1). It is convenient to use a geometrical representation of code words on figure 3.1. Let's represent a code words by set from three symbols (x, y, z) and values of these symbols will choose from the binary alphabet {0, 1}. It is possible to represent all possible code words by the points in the Cartesian system with coordinates (x, y, z). Thus words will form tops of a three-dimensional cube. On figure 3.1 these tops are marked as follows:

– By the sign "•" notes the allowed code words,

– By the sign "×" notes the forbidden code words.

It is visible, that code structure is that between the allowed code words are forbidden words. They form the «**protective interval**». Therefore the action of any single error translates any allowed word to the nearest forbidden words. This property leads to such **decoding rule** of a code with even units number and detection of any single errors: reception from the channel output of the forbidden code words allows

to assert that in the channel there was any single error. It is easy to be convinced that this code does not allow to detect double errors (because «protective interval» is nonsufficient*)*. By induction it is possible to prove, that any binary code with even number of units allows to detect any errors if their multiplicity is odd and does not detect any errors if their multiplicity is even. The concept of «a protective interval» is easily applicable for a study of the relation between code distance and code ability to correct of an errors. If the minimum distance between allowed code words (code distance) is $d_{\min}$, that as is shown from figure 3.2 the protective interval contains ($d_{\min}$ - 1) forbidden words and for "transfer" of each allowed word to nearest allowed word it is necessary by errors to make ($d_{\min}$ – 1) "steps". Clearly, that all errors with multiplicity q=1, 2, 3, ..., ($d_{\min}$ – 1) can be detected.

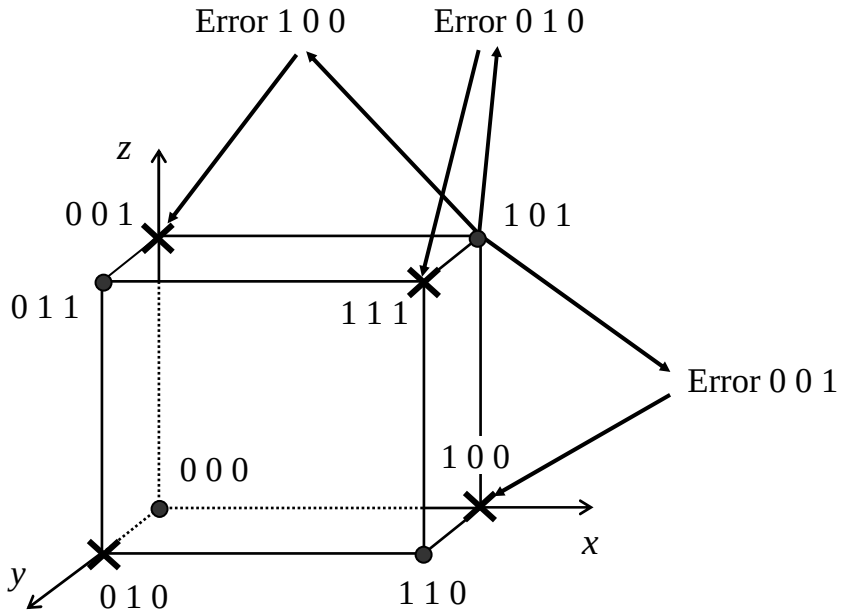

Figure 3.1 – Illustration of a code correction ability

From here follows, that if code distance of a binary code is $d_{\min}$ **code ability to detect of errors** with multiplicity q_{det} is defined as:

$$q_{\text{det}} \le (d_{\min} - 1) \tag{3.1}$$

Let's take advantage of similar representation for estimations of ability to correct of errors. On figure 3.3 layout of the allowed code words $\mathbf{b}_{\text{allow.1}}$ and $\mathbf{b}_{\text{allow.2}}$ is shown. Between them are allocated ($d_{\min}$ – 1) the forbidden words. Let's divide all set of the words on two allowed subset as is shown in a figure 3.3. If for example the received word $\widehat{\mathbf{b}}$ is allocated into «allowed decoding subset of a word $\mathbf{b}_{\text{allow.1}}$» that during the decoding becomes decision about transmitting of the word $\mathbf{b}_{\text{allow.1}}$, i.e. thereby the error transitions of word $\mathbf{b}_{\text{allow.1}}$ to the nearest forbidden words are corrected. It is similarly possible to explain error control process by the transmission of the word $\mathbf{b}_{\text{allow.2}}$. It is visible, that distance of each allowed subset is ($d_{\min}$ – 1)/2 (by $d_{\min}$ is odd). It defines code error control ability. For even values $d_{\min}$ the distance of each allowed subset is [($d_{\min}$/2) – 1], that also defines error-control ability of a code.

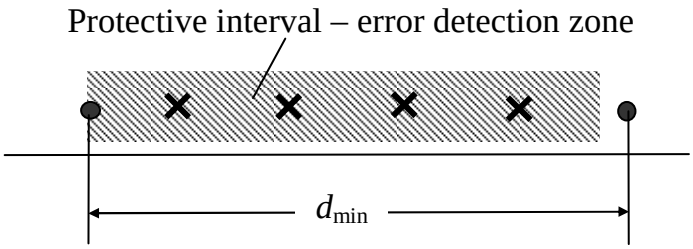

Figure 3.2 – To illustration of an error detecting ability

Thus, if code distance of a binary error control code is $d_{\min}$ **code ability to correct of errors** is defined by expressions:

$$q_{\text{corr}} \le \frac{d_{\min} - 1}{2}\text{, if } (d_{\min} - \text{odd}) \text{ and } q_{\text{corr}} \le \frac{d_{\min}}{2} - 1\text{, if } (d_{\min} - \text{even}). \qquad (3.2)$$

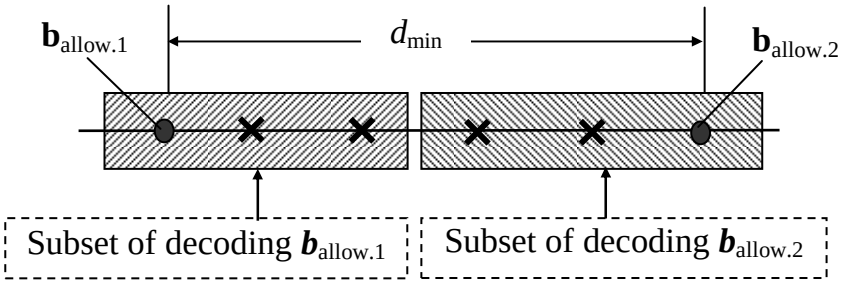

Figure 3.3 – Illustration of an error-correcting ability

4 ALGEBRAIC DESCRIPTION OF BLOCK CODES

For description of the linear block codes use a mathematical apparatus of the general algebra [3]. By the block coding form code words $\boldsymbol{b} = (b_1, b_2, ..., b_n)$. Choose symbols of binary codes from the **Galois field GF**(2). The set of words forms n-dimensional vector space over field **GF**(2). For elements of this space (vectors) the addition and multiplication operations and operation of multiplication of a vector and also scalar product of a vectors are defined. Some vectors subset of the space $\mathbf{B}_n$ which satisfy to the vector space axioms organizes subspace $\mathbf{A}_k$.

The binary block code with block length n and 2^k allowed code words is called as the **linear (*n*, *k*) code** if its code words form k-dimensional vector subspace $\mathbf{A}_k$ of n-dimensional space $\mathbf{B}_n$. Subspace $\mathbf{A}_k$ is generated by the **basis** from k linearly independent vectors, which organize the lines of a **generator matrix** of the (n, k) code:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_k \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & g_{02} & \cdots & g_{0,\,\text{n}} \\ g_{10} & g_{11} & g_{12} & \cdots & g_{1,\,\text{n}} \\ \vdots & \vdots & \vdots & & \vdots \\ g_{k,\,0} & g_{k,1} & g_{k,\,2} & \cdots & g_{k,\,\text{n}} \end{bmatrix}. \qquad (4.1)$$

It is possible to present code words in the **systematic form**, forming separately **informational part** from k numerals and **check part** from $r = (n - k)$ additional numerals.

The generator matrix of a systematic code looks like:

$$\mathbf{G}_{\text{syst}} = |\mathbf{I}_k \mathbf{P}| = \left| \underbrace{\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}}_{\mathbf{I}_k} \underbrace{\begin{matrix} g_{1,n-r} & \cdots & g_{1,n} \\ g_{2,n-r} & \cdots & g_{2,n} \\ \vdots & \cdots & \vdots \\ g_{k,n-r} & \cdots & g_{k,n} \end{matrix}}_{\mathbf{P}} \right| \qquad (4.2)$$

Matrix $\mathbf{G}_{\text{syst}}$ contains **identity matrix** $\mathbf{I}_k$ which defines the information part of code words and matrix **P** defines the **additional symbols**. Transition to the systematic form is made by a linear combination of rows from the matrix (4.1). Such transition is illustrated by a following example.

Example 4.1 Matrix transformation of the nonsystematic code. The nonsystematic block code (7,4) is set by the generator matrix:

$$\mathbf{G}_{\text{nonsyst}} = \begin{vmatrix} \mathbf{g1}_{\text{ns}} \\ \mathbf{g2}_{\text{ns}} \\ \mathbf{g3}_{\text{ns}} \\ \mathbf{g4}_{\text{ns}} \end{vmatrix} = \begin{vmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{vmatrix}. \qquad (4.3)$$

Using a method of the linear combination of rows from matrix (4.3) we will transform it to the systematic form (4.2). For forming of a systematic generator matrix the rows of an initial matrix (4.3) it is convenient to present in the form of a table 4.1, in which rows $\mathbf{g1}_{\text{ns}}$, $\mathbf{g2}_{\text{ns}}$, $\mathbf{g3}_{\text{ns}}$ and $\mathbf{g4}_{\text{ns}}$ are shown.

Table 4.1 – Rows of the nonsystematic generator matrix

$\mathbf{g1}_{\text{ns}}$	1	0	1	1	0	1	0
$\mathbf{g2}_{\text{ns}}$	0	1	0	0	1	0	1
$\mathbf{g3}_{\text{ns}}$	0	0	1	0	0	1	1
$\mathbf{g4}_{\text{ns}}$	0	0	0	1	1	1	1

Using modulo-2 addition rules the elements of these rows by exhaustive search of rows in various combinations it is established that by the most suitable variants for forming of matrixes rows for the systematic code are following:

$$\mathbf{g1}_{\text{syst}} = (\mathbf{g1}_{\text{ns}} \oplus \mathbf{g3}_{\text{ns}} \oplus \mathbf{g4}_{\text{ns}}),\ \mathbf{g2}_{\text{syst}} = (\mathbf{g2}_{\text{ns}} \oplus \mathbf{g3}_{\text{ns}}),\ \mathbf{g3}_{\text{syst}} = \mathbf{g3}_{\text{ns}},\ \mathbf{g4}_{\text{syst}} = \mathbf{g4}_{\text{ns}}.$$

The outcome of an evaluation of the matrixes rows of the systematic code is reduced in table 4.2.

Table 4.2 – The matrixes rows of the systematic code

$\mathbf{g1}_{syst}$	1	0	0	0	1	1	0
$\mathbf{g2}_{syst}$	0	1	0	0	1	0	1
$\mathbf{g3}_{syst}$	0	0	1	0	0	1	1
$\mathbf{g4}_{syst}$	0	0	0	1	1	1	1

The matrix of the systematic code in the standard form low given:

$$\mathbf{G}_{syst} = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{vmatrix}. \qquad (4.4)$$

The concept weight of a code word plays the important role in the block codes theory.

Hamming weight w_H of the binary code word is equal to an amount of units in a code word.

Example 4.2 An evaluation of Hamming weights of a code words.

We will define values of Hamming weights for the code words by table 4.3:

Table 4.3 – The Hamming weights of code words

Binary code words							Weight $w_H(\mathbf{b_i})$
$\mathbf{b_1}$	1	0	1	1	0	1	4
$\mathbf{b_2}$	0	1	0	0	0	1	2

Structure of a generator matrix allows to define the minimum distance of the block codes. This position is illustrated by following exercises.

Exercise 4.1 Definition of the code distance by its generator matrix.

Generator matrixes of the error-control codes by (4.3) or (4.4) are set. Show how to define code distance of a codes by known generator matrix.

Instruction. By elaborating of the method for definition of code distance it is necessary to consider that zero combination $\mathbf{b_0}$ = (0000000) is also allowed.

Decision. It is above noticed, that allowed code words are defined by linear combinations of a generator matrix rows. As zero word $\mathbf{b_0}$ = (0000000) also is allowed, and rows of a generator matrix **g1**, **g2**, **g3**, **g4** also are the allowed words then Hamming distances from these words to a zero word $\mathbf{b_0}$ it is defined their weights d_H (**gi**, $\mathbf{b_0}$) =w_H(**gi**), i = (1,…k). Further it is necessary to find the minimum weight, i.e. the minimum distance. Such conclusion from here follows.

Code distance as the value of minimum distance between allowed code words is defined by the **least weight of rows** of generator matrix.

Example 4.3 Definition of Hamming weights of the generator matrix rows of a systematic code.

Define values of row weights of a generator matrix from the Example 4.1 (table 4.2). Outcomes of evaluations are reduced in table 4.4.

Table 4.4 – The Hamming weights of generator matrix rows for systematic code

The generating matrix rows								Weights w_H(**gi**)
$\mathbf{g1}_{syst}$	1	0	0	0	1	1	0	3
$\mathbf{g2}_{syst}$	0	1	0	0	1	0	1	3
$\mathbf{g3}_{syst}$	0	0	1	0	0	1	1	3
$\mathbf{g4}_{syst}$	0	0	0	1	1	1	1	4

The analysis of data makes definition of the minimum distance of systematic code from table 4.4 $d_{min\ syst} = \min\{w_H(\mathbf{gi})\} = 3$.

Exercise 4.2 Define by the same way the code distance of nonsystematic code from an Example 4.1 (table 4.1).

Instruction. The statement about code distance of block code from Exercise 4.1 is fair both for systematic nonsystematic codes.

Decision. We will apply a technique from the Example 4.3. Outcomes of evaluations the weights of rows are reduced in table 4.5.

Table 4.5 – The Hamming weights of the generator matrix rows for nonsystematic code

Generator matrix rows								Weights w_H(**gi**)
$\mathbf{g1}_{ns}$	1	0	1	1	0	1	0	4
$\mathbf{g2}_{ns}$	0	1	0	0	1	0	1	4
$\mathbf{g3}_{ns}$	0	0	1	0	0	1	1	3
$\mathbf{g4}_{ns}$	0	0	0	1	1	1	1	4

The analysis of these data makes definition of the minimum distance of nonsystematic code from table 4.5 $d_{min\ ns} = \min\{w_H(\mathbf{gi})\} = 3$. The received outcomes allow to state that systematic and nonsystematic codes from the Example 4.1 on the value of code distance are equivalent.

Thus, the **code distance** of a block code is a **least weight of nonzero rows** from the code generator matrix. The above noted dependence between the minimum distance of block codes and weights of nonzero rows can be used for forming of a generator matrix of block code with the beforehand set code distance. This is illustrated by outcomes of Examples 4.4 and 4.5.

Example 4.4 A generator matrix of a code with even number of units.

Let's form the generator matrix of the systematic (n, k) code which detect a single errors ($q_{det} = 1$). Such code should have code distance $d_{min} = q_{det} + 1 = 2$. Hence the nonzero rows of generator matrix of this code should have minimum weight $w_H = 2$. According to the standard form (4.4) each row of systematic code matrix already contains a numeral 1 (defined by the submatrix $\mathbf{I}_k$), the weight should be increased the weight of every rows to 2 having added in last numerals every rows (as a part of submatrix **P**) a numeral 1.

For example the generator matrix such (7,4) codes with $k = 4$ will look like

$$\mathbf{G} = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{vmatrix}, \qquad (4.5)$$

and unit in submatrix **P** can be in any place of a line.

Exercise 4.5 Generator matrixes of codes which can detect double errors.

Form generator matrix of the systematic code which can detect double errors.

Instruction. From the theory it does not follow that such code there can be only one. It is recommended to consider at first a principle construction of a matrix at least one code and then on this basis to give generalization and to find matrixes of several more codes.

Decision. The code can detect errors with multiplicity $q_{\text{det}} = 2$ should have the minimum distance $d_{\min} = q_{\text{det}} = 2 + 1 = 3$. Hence rows of generator matrix of such code should have minimum weight $w_{\text{H}} = 3$. From general view of generator matrix of a systematic code (4.2) follows what to get such weight it is possible by choice of rows of the additional symbols submatrix **P** and one of row from this submatrix should have the weight equal 2. Following variants of submatrix **P** are possible:

$$\mathbf{P1} = \begin{vmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{vmatrix}; \quad \mathbf{P2} = \begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{vmatrix}, \quad \mathbf{P3} = \begin{vmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{vmatrix}, \qquad (4.6)$$

which differ by rows permutations. As the minimum of each weight rows of these matrixes is equal to 2, they can be used for forming of systematic codes with minimum distance $d_{\min} = 3$. In particular, the generator matrix of one of such codes looks like:

$$\mathbf{G} = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{vmatrix}. \qquad (4.7)$$

Questions

4.1 Codes with generator matrixes (4.3) and (4.4) have the identical minimum distance. What is it explained by?

4.2 Are error-control properties of a block code varied by permutation of columns of a generator matrix?

4.3 Is it the only one method of transformation from nonsystematic generator matrix to the systematic code matrix for forming which is considered in the Example 4.1?

Task

4.1 Follow method stated in Exercise 4.3 form generator matrix of the systematic code which correct triple errors. Is this code the only one?

5 CODING AND DECODING OF BLOCK CODES

5.1 Coding and decoding of block code

In the center of the block coding theory is the concept of a generator matrix (4.1) and (4.2). If **a** = $|a_0, a_1, ..., a_k|$ – row-matrix of a primary code the **coding rule** of a block code is defined by the **product**

$$\mathbf{b} = \mathbf{aG}, \tag{5.1}$$

where $\boldsymbol{a} = |a_0, a_1, ..., a_k|$ – row-matrix of primary code at encoder input,

$\boldsymbol{b} = |b_0, b_1, ..., b_n|$ – row-matrix of block code word at encoder output,

$\boldsymbol{G}$ – generator matrix of linear (n, k) code.

Example 5.1 The encoder of code (7,4).

The encoder structure of a systematic code (7,4) is defined by generator matrix (4.4) and coding rule (5.1). If on encoder input is the symbols row of a primary code $\boldsymbol{a} = (a_1, a_2, a_3, a_4)$ then symbols of allowed code word on its output

$\boldsymbol{b} = (b_1, b_2, b_3, b_4, b_5, b_6, b_7)$ are defined by following equalities:

$$b_1 = a_1,\ b_2 = a_2,\ b_3 = a_3,\ b_4 = a_4,$$
$$b_5 = a_1 \oplus a_2 \oplus a_3 \oplus a_4,\ b_6 = a_1 \oplus a_2 \oplus a_4,\ b_7 = a_1 \oplus a_3 \oplus a_4. \tag{5.2}$$

On figure 5.1 the structure of encoder of systematic code (7,4) with equalities (5.2) is shown.

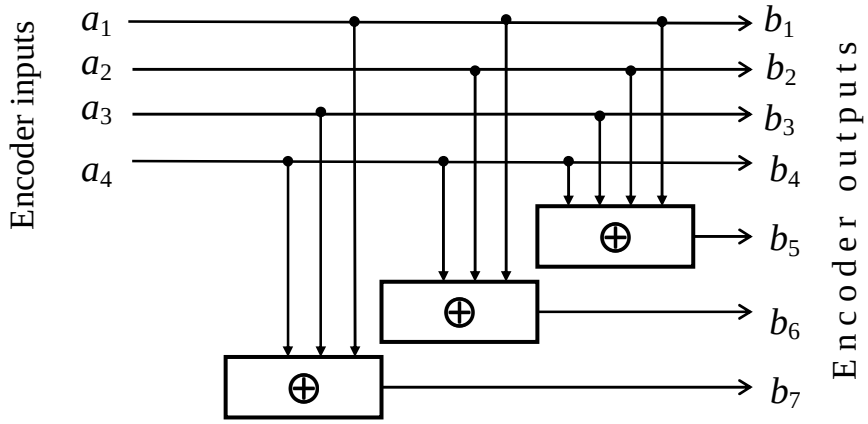

Figure 5.1 – Encoder structure of systematic code

By the decoding of block codes the **check relations** establish with use of the parity check matrix **H** which space of rows is **orthogonal** to space of rows of generator matrix, that is:

$$\mathbf{G} \cdot \mathbf{H}^{\mathrm{T}} = 0. \tag{5.3}$$

Here T– an index a transposition.

If generator matrix is set in the form (4.2) for performance of a orthogonality condition the **parity check matrix** should look like:

$$\mathbf{H} = (\mathbf{P}^{\mathrm{T}} \,|\, \mathbf{I}_{n-k}), \tag{5.4}$$

where $\mathbf{P}^{\mathrm{T}}$– transposed submatrix **P** of generator matrix **G**,

$\mathbf{I}_{n-k}$ – identity matrix a size $(n - k) \times (n - k)$.

Exercise 5.1 The parity check matrix of a systematic code (7,4).

The generator matrix of systematic code (4.7) is set:

$$\mathbf{G}_{\text{syst}} = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{vmatrix}.$$

According to rule (5.4) form parity check matrix of this code.

Solution. Sequentially we discover the submatrixes entering into the formula (5.4). The transposed submatrix by size $(n - k) \times k$:

$$\mathbf{P}^{\mathrm{T}} = \begin{vmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{vmatrix},$$

The identity submatrix by size $(n - k) \times (n - k)$:

$$\mathbf{I}_{n-k} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}.$$

We unite submatrixes in the uniform parity check matrix of code:

$$\mathbf{H}_{\text{syst}} = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}. \qquad (5.5)$$

From orthogonality condition of generator and parity check matrixes of linear code (5.3) follows that each allowed word of linear code generated by rule $\mathbf{b} = \mathbf{a}\cdot\mathbf{G}$ also satisfies to the orthogonality condition:

$$\mathbf{b} \cdot \mathbf{H}^{\mathrm{T}} = \mathbf{a} \cdot \mathbf{G} \cdot \mathbf{H}^{\mathrm{T}} = \mathbf{0}. \qquad (5.6)$$

By transmission through the channel code symbols are distorted. The received words look like $\widehat{\mathbf{b}} = \mathbf{b} \oplus \mathbf{e}$, where $\mathbf{b} = (b_0, b_1, ..., b_n)$, and an error vector $\mathbf{e} = (e_0, e_1, ..., e_n)$. By decoding calculate a **syndrome vector**

$$\mathbf{S} = \widehat{\mathbf{b}} \cdot \mathbf{H}^{\mathrm{T}} = (s_0, s_1, ..., s_{n-k-1}). \qquad (5.7)$$

The syndrome depends only on an error vector:

$$\mathbf{S} = \widehat{\mathbf{b}} \cdot \mathbf{H}^{\mathrm{T}} = (\mathbf{b} \oplus \mathbf{e})\, \mathbf{H}^{\mathrm{T}} = \widehat{\mathbf{b}} \cdot \mathbf{H}^{\mathrm{T}} \oplus \mathbf{e} \cdot \mathbf{H}^{\mathrm{T}}.$$

As the condition of orthogonality $\widehat{b} \cdot \mathbf{H}\mathrm{T} = \mathbf{0}$ is satisfied, the **syndrome** is equal:

$$\mathbf{S} = \mathbf{e} \cdot \mathbf{H}^{\mathrm{T}} \qquad (5.8)$$

From here the simple rule of the error detection follows:

1 If the **syndrome S = 0** then an error vector **e = 0**, i.e. in the channel there were **no errors** and the received word belongs to set of the allowed code words.

2 If $\mathbf{S} \neq \mathbf{0}$ word $\hat{\mathbf{b}}$ **contains errors**. It is possible by the syndrome symbols to define a configuration of the error vector.

This principle underlies syndrome decoding.

5.2 Syndrome decoding of the block codes

The pprinciple of syndrome decoding we will consider on an example of simple block code.

Example 5.3. The syndrome decoder of systematic code (7, 4).

According to a rule (5.8) for realization of the syndrome decoder it is necessary to form the **transposed parity check matrix** of a code (7, 4). The parity check matrix of this code looks like (5.5). Applying to it a rule of a transposition of matrixes it is received:

$$\mathbf{H}_{\text{syst}} = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}; \qquad \mathbf{H}_{\text{syst}}^{\text{T}} \begin{vmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}. \tag{5.9}$$

It is convenient to note the single errors in transmission channel so:

$$\mathbf{e}_1 = (100\ldots0),\ \mathbf{e}_2 = (010\ldots0),\ \mathbf{e}_3 = (001..0),\ \ldots,\ \mathbf{e}_n = (000..1). \tag{5.10}$$

In such form the error vector $\mathbf{e}_i$ represents a symbol set from n elements in which on a place with number i the symbol of an error 1 (at the left) is arranged and on remaining places zero symbols are arranged. Error vectors can be presented in the form of an identity matrix:

$$\mathbf{E} = \begin{vmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \cdot \\ \cdot \\ \mathbf{e}_i \\ \cdot \\ \mathbf{e}_n \end{vmatrix} = \mathbf{I}_n = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}, \tag{5.11}$$

which each row is the single error vector. Using properties of identity matrixes, it is easy to show, that the matrix of syndromes coincides with the transposed parity check matrix of this code (5.9) is:

$$\mathbf{S} = \mathbf{E} \cdot \mathbf{H}^{\text{T}} = \mathbf{I}_{\text{n}} \cdot \mathbf{H}^{\text{T}} = \mathbf{H}^{\text{T}}. \tag{5.12}$$

By the syndrome decoding of a block code the **matrix of syndromes S** coincides with the transposed parity check matrix of a code $\mathbf{H}^{\text{T}}$. It is the foundation

for tabling of syndromes**.** The more low reduced table 5.1 of syndromes for a code (7,4) is made according to rows of the transposed parity check matrix (5.9). In the table to each vector of an error there corresponds the vector of the syndrome specifying a location of an error symbol in the received code word.

Table 5.1 – Syndromes for decoding of the code (7,4)

Syndromes	011	110	101	111	100	010	001
Errors	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$	$\mathbf{e}_6$	$\mathbf{e}_7$

It allows to formulate of **syndrome decoding algorithm**:

1 Forming of the transposed parity check matrix of a code $\mathbf{H}^T$.

2 Tabling of syndromes for decoding of (n, k) code.

3 An evaluation of syndromes (as table 5.1) on structure of code transposed parity check matrix $\mathbf{H}^T$ and error symbols vector of a decoded codeword by rule (5.12).

4 Forming of a vector of an error $\mathbf{e}_i$ on the basis of the syndromes table.

5 Error correction in the received code word by a rule: $\hat{\mathbf{a}} = \hat{\mathbf{b}} \oplus \mathbf{e}_i$.

The structure of syndrome decoder of code (7,4) realizing this algorithm is reduced on figure 5.2. According to rule (5.12) received channel symbols move to modulo-2 adders. The connections with lines of channel symbols are available there where in rows of transposed parity check matrix the symbol 1 is arranged. In the scheme of syndrome analyzer with according to given table 5.1 there is transformation of syndrome vectors $\mathbf{S} = (s_0, s_1, ..., s_{n-k-1})$ in the corresponding error vectors **e** which then move to the error corrector.

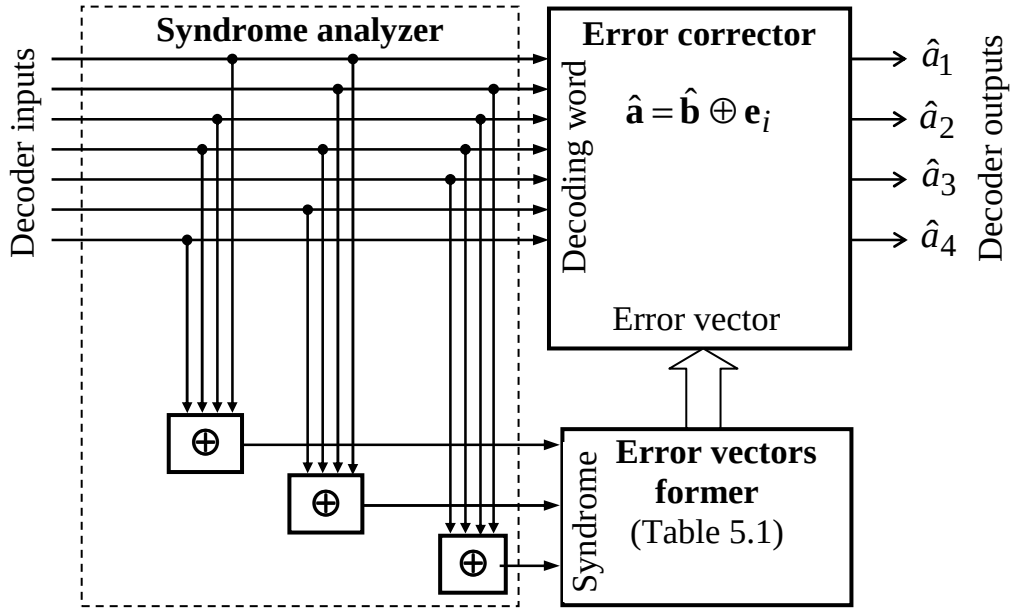

Figure 5.2 – The structure of the syndrome decoder of the code (7,4)

5.3 Majority decoding of block codes

Some block codes suppose realization of simple **majority algorithm** which is based on a possibility to express each information code symbol of a word by several ways through other received symbols. Let's consider a systematic code (7,3):

$$\mathbf{G} = \begin{vmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{vmatrix}. \qquad (5.13)$$

To this matrix corresponds transposed parity check matrix:

$$\mathbf{H}^{\mathrm{T}} = \begin{vmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}. \qquad (5.14)$$

Let's designate the received from the channel code word as

$$\mathbf{b} = (b_1, b_2, b_3, b_4, b_5, b_6, b_7).$$

As considered code – systematic, first three symbols (b_1, b_2, b_3) are information symbols. Using structural properties of this code, it is possible to form during decoding both trivial and compound estimations of information symbols which are presented to table 5.2. On the basis of columns of parity check matrix (5.15) we will write down **verifying parities:**

$$b_1 \oplus b_3 \oplus b_4 = 0,\ b_1 \oplus b_2 \oplus b_3 \oplus b_5 = 0,\ b_1 \oplus b_2 \oplus b_6 = 0,\ b_2 \oplus b_3 \oplus b_7 = 0, \quad (5.15)$$

which allow to form compound estimations. For example, on the basis of the first equality from (5.15) follows the **compound estimation** of the first information symbol $b_1 = b_3 \oplus b_4$. The **trivial estimation** of this symbol also is, actually, this symbol $b_1 = b_1$, as a code is systematic. Expressions for other information symbols are made similarly. They are presented in the table5.2.

Table 5.2 – Majority decoding of block code

Estimations of information symbols		
Symbol b_1	Symbol b_2	Symbol b_3
T r i v i a l		
$b_1 = b_1$	$b_2 = b_2$	$b_3 = b_3$
C o m p o u n d		
$b_1 = b_3 \oplus b_4$ $b_1 = b_5 \oplus b_7$ $b_1 = b_2 \oplus b_6$	$b_2 = b_4 \oplus b_5$ $b_2 = b_6 \oplus b_1$ $b_2 = b_3 \oplus b_7$	$b_3 = b_5 \oplus b_6$ $b_3 = b_7 \oplus b_2$ $b_3 = b_4 \oplus b_1$

After formation of estimations they move on a **majority element** in which the decision on each information symbol is taken out on «the **majority of voices**».

For example, if estimations of information symbol $b1$ look like:

$$b_1 = b_1 = 1,\ b_1 = b_3 \oplus b_4 = 1,\ b_1 = b_5 \oplus b_7 = 1,\ b_1 = b_2 \oplus b_6 = 0,$$

in which the quantity of estimations «1» exceeds quantity of estimations «0» the majority element passes the decision «on the majority»: $b_1 = 1$. The compound estimations enumerated in table 5.2 are called as **orthogonal estimations** as incoincident numerals enter into them. The number of orthogonal estimations N and a multiplicity of errors q_{corr}, corrected by majority decoding are in the ratio:

$$q_{\text{corr.}} \le (N - 1)/2. \qquad (5.16)$$

The code with generator matrix (5.13) allows to form $N = 3$ orthogonal estimations and, accordingly, to correct unitary errors in information symbols by considerable simplification of decoding algorithm. It is necessary to notice that formation rules of estimations can have cyclic properties that simplifies decoding procedure.

Example 5.4 Structure of the majority decoder for the code (7, 3).

Let's generate structure of majority decoder of code (7, 3) on the basis of estimations system from table 5.2. It is easy to see, that checks have cyclic properties.

For example, indexes in compound estimations $b_1 = b_3 \oplus b_4$, $b_2 = b_4 \oplus b_5$ and $b_3 = b_5 \oplus b_6$ change on 1 towards increase. Taking it into account the decoder structure of code (7, 3) realizing majority decoding algorithm looks like shown on figure 5.3. The decoder consists of the shift register, the switchboard on the input, operated from system for block synchronization, schemes of estimations formation and the majority element. The decoder works as follows. At the beginning the switchboard on an input is established in position «1» and decoded code word $\mathbf{b} = (b_1, b_2, b_3, b_4, b_5, b_6, b_7)$ is entered in shift register cells. Thus on inputs of majority element the compound estimations defined by table 5.2 operate both trivial and compound estimations. The decision about transmitted information symbol b_1 is read out from an exit of majority element. Then the switchboard is established in position «2» and there is on one symbol shift of word. On this step, owing to cyclic properties of estimations the second information symbol are formed and the decision on an information symbol b_2 is read out from exit of majority element. Further process repeats up to reception on a output symbol b_3 etc.

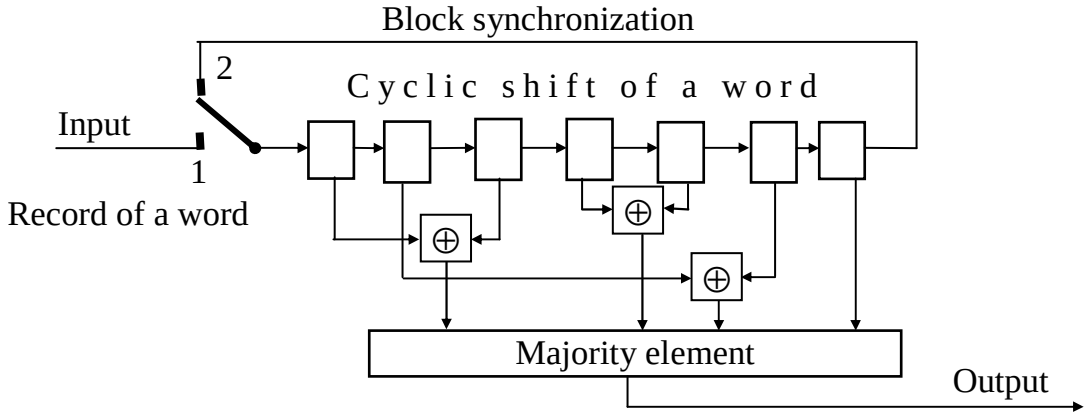

Figure 5.3 – Structure of the majority decoder of a code (7, 3)

Questions

5.1 What kind will a matrix of double errors have? How will it change in comparison with a matrix of single errors (5.11)?

5.2 How are parameters of binary syndrome representation (see table 5.1) connected with general number of possible configurations variants which detected and corrected errors by syndrome decoding?

5.3 How will the syndrome format change if to apply a method of syndrome decoding to decoding double errors?

5.4 Give the generalized block diagram of syndrome decoder of a block (n, k) code. What is the function of syndrome analyzer?

Tasks

5.1 By the principles stated in the Example 5.1 represent structure a systematic block code intended for detection of double errors with the generating matrix (4.6).

5.2 The generator matrix of a code (7,4) is set:

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

Define allowed code word of this code **b** if the word of a primary code on a coder input **a** = (1110) is set.

5.3 Define code distance of a code (7, 4) with a generator matrix from the Task 5.2.

5.4 Represent a encoder structure of a code (7, 4) with the same generator matrix.

6 BOUNDARIES OF BLOCK CODES PARAMETERS

The problem of the coding theory is the search of codes which at given block length n and rate R_{code} provides a maximum of code distance $d_{\min}$. Limits of these parameters are defined by the code boundaries.

6.1 Hamming upper bound

The conclusion of the upper bound is based on reasons of spherical packing (bound of **spherical packing**). At given minimum distance between the allowed code words $d_{\min}$. The greatest rate can be reached, if the spheres surrounding each word will be **most densely packed.** Volume of each sphere is equal $\sum_{i=0}^{d_{\min}-1} C_n^i$ and the number of spheres (number of code words) is equal 2^k. For best code the total quantity of spheres and number of all possible words 2^n should coincide. Equality is reached for **densely packed (perfect) codes**. Area of each code word represents sphere with radius $(d_{\min} - 1)/2$, and these areas of such codes not being crossed densely fill with themselves all n-dimensional space of code words. The inequality from here follows:

$$2^k \sum_{i=0}^{d_{\min}-1} C_n^i \le 2^n .$$

After simple transformations it is possible to receive obvious expression for rate of the **perfect code**:

$$1 - R_{\text{code}} \ge \frac{1}{n} \log_2 \sum_{i=0}^{d_{\min}-1} C_n^i \ . \qquad (6.1)$$

The dependence of Hamming upper bound is shown on figure 6.1 (curve «Hamming upper bound»). Hamming bound is fair both for linear and nonlinear codes.

6.2 Varshamov-Gilbert lower bound

For block codes it is possible to get the **Varshamov-Gilbert lower bound** which defines the possibility of codes existence with both parameters R_{code} and $d_{\min}$. The asymptotic form (for long codes) of this bound looks like:

$$R_{\text{code}} \ge 1 - H(d_{\min}/n), \qquad (6.2)$$

where $H(x)$ – binary entropy.

Dependence of **Varshamov-Gilbert lower bound** for binary codes is shown on figure 6.1 (curve «Varshamov-Gilbert lower bound»). The bound guarantees existence of the codes which performances correspond to points arranged at least on a curve (or above it). **Search of the codes** ensuring the given minimum distance $d_{\min}$ and high enough rate R_{code} at $n \to \infty$, ensuring at the same time a possibility of algorithms decoding realization with low complexity is one of **important problem** of the theory of coding.

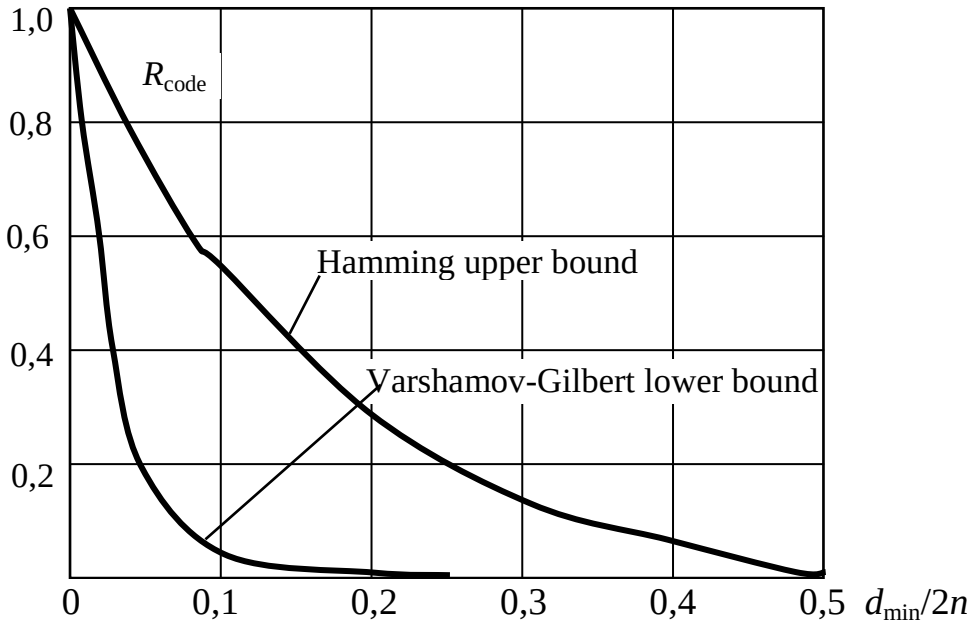

Figure 6.1 – Code boundaries of block codes

6.3 Complexity of coding and decoding algorithms

The using of code control ability depends on decoding algorithm. By **full decoding** use all possibilities to correct errors following from properties of a code. According to Shannon fundamental theorem the error-control codes used for correction of channel errors should get out long enough. However with growth of a code word length n increases complexity of realization procedures both of encoding and decoding that causes difficulty of practical realization of codecs. In the coding theory of along with estimations of code error-control ability of can estimate complexity of realization of encoding/decoding procedures which can be realized by software or hardware. Thus as argument of complexity function the length of a codeword n should act.

Coding complexity of a block codes C_{cod} with use of generator matrix (n, k) code with a size $nk = n^2(1 - R_{\text{code}})$ usually estimate in the value which is proportional to number of elements of the generator matrix

$$C_{\text{cod}} = nk = n^2(1 - R_{\text{code}}). \qquad (6.3)$$

The decoding algorithms appear more difficult. Among them it is considered to be the most difficult **full-search algorithm** according to which the decoder by the full searching compares the received code word with the set of all possible words and the decision on that transmitted from the allowed word which appears on the minimum distance from the received word (**decoding by a minimum distance**) passes. It is considered to be complexity of full-search decoding algorithm proportional to quantity of all possible code words to volume of full search**:**

$$C_{\text{decod}} = m^n = 2^n. \qquad (6.4)$$

It is sUSBC˜â^□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ã□ã decoding increases «as an exponent» with growth of code length. Clearly, that **full-search decoding algorithms are practically difficult for realising for long codes**.

Questions

6.1What is practical significance of use of Hamming upper bound and Varshamov-Gilbert lower bound for an estimation of block error-control codes performances?

6.2 To what bound (upper or lower) is it necessary to aspire by elaborating of new block codes?

7 IMPORTANT CLASSES OF BLOCK CODES

The big number of codes, various on structure, construction principles and control ability is known. In this Chapter the classes of effective block codes with simple decoding algorithms are considered.

7.1 Hamming codes

Hamming codes (*by R. Hamming*) – systematic block codes with parameters:

– Code word length $n = 2^r - 1$;

– Quantity of information symbols $k = 2^r - r - 1$; (7.1)

– Number of additional symbols $r = n - k$;

– Minimum distance $d_{\min} = 3$, $r = 2, 3, 4$.

Hamming codes – perfect codes which **correct single errors**.

By parameters choice $r = 2, 3, 4$ according to formulas (7.1) it is possible to set all known binary Hamming codes. For example, at $r = 3$ the parameters of a code (7, 4) will be the following:

– Length of the code word $n = 3$;

– Quantity of information symbols $k = 4$;

– Minimum distance $d_{\min} = 3$;

– Code rate $R_{\text{code}} = (2^r - r - 1)/(2^r - 1) = 4/7$.

Generator and check matrixes of this code have been considered earlier, in Section 4.1 (formulas (4.4) and (4.7)). As it has been noted earlier, this code allows to **detect double errors** also. Structures of encoder and syndrome decoder of Hamming code have been considered earlier in Section 5.1 (figures 5.1, 5.2). According to the formula (3.5) transposed parity check matrix of this code looks like:

$$\mathbf{H}^{\mathrm{T}} = \begin{vmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}. \qquad (7.2)$$

7.2 Cyclic codes

The considerable part of block codes belongs to the class of cyclic codes. It defines a simplification of both encoding and decoding procedures on the basis of a **cyclical properties** of code words. If $\mathbf{b} = (b_0, b_1, ..., b_n)$ – the allowed code word of the cyclic code so its **cyclical shift** on arbitrary number of symbols also is the allowed code word. For example, a word $\mathbf{b1} = (b_n, b_0, b_1, ..., b_{n-1})$ corresponds to cyclical shift of a word $\mathbf{b} = (b_0, b_1, ..., b_{n-1}, b_n)$ on one symbol to the right. Thus according to a rule of **cyclical permutation** combination symbols **b** are displaced on one numeral to the right, and the right numeral b_n takes a place of a left numeral b_0. Properties of the cyclic code are convenient for studying, representing code words in the form of polynomials on degrees of a formal variable x which factors are symbol numerals in a code word $\boldsymbol{b}(x) = b_0 + b_1 \cdot x + b_2 \cdot x^{`} + ... + b_n \cdot x^n$. Mathematical operations (addition, multiplication and division of polynomials) make by rules of polynomials algebra which stated in Section 5 of manual [3]. If addition and multiplication of polynomials is made by the **modulo of polynomial ($\boldsymbol{x^n - 1}$)** so all possible polynomials of degree $(n - 1)$ and less organize an **algebraic ring** of polynomials $\boldsymbol{R_n}$ with the properties stated in the manual [3]. For construction of a cyclic code in a ring $\boldsymbol{R_n}$ choose a subset of polynomials an **ideal I.** The polynomial of the minimum degree $\boldsymbol{g}(x)$ in this subset is called as the **generator polynomial** of cyclic code. As generator polynomials of the cyclic code choose the **prime polynomial**. In algebra of

polynomials of the whole degree prime polynomial play the same role what **prime number** play in the algebra of integers. The detailed table of generating polynomials of cyclical codes is reduced in Attachment A.1. Generator polynomials of short cyclic codes are given in table. 7.1.

Table 7.1 – Generator polynomials of short cyclic codes

Maximum degree of a generator polynomial	Generator polynomial $\boldsymbol{g}(x)$		
3	$x^3 + x^2 + 1$	$x^3 + x + 1$	
4	$x^4 + x + 1$	$x^4 + x^3 + 1$	
5	$x^5 + x^2 + 1$	$x^5 + x^3 + 1$	$x^5 + x^4 + x^2 + 1$
6	$x^6 + x + 1$	$x^6 + x^5 + 1$	$x^6 + x^5 + x^3 + x^2 + 1$

All polynomials of the ideal **I** corresponding to the allowed code words of cyclic codes are divided on generator polynomial $\boldsymbol{g}(x)$ without remainder that allows to formulate a following coding rule:

Coding rule of nonsystematic cyclic code looks like:

$$\boldsymbol{b}(x) = \boldsymbol{a}(x)\cdot\boldsymbol{g}(x). \qquad (7.3)$$

Systematic cyclic codes are used often in practice

Coding rule of systematic cyclic code (n, k) looks like:

$$\boldsymbol{b}(x) = \boldsymbol{a}(x)\cdot x^{n-k} + \boldsymbol{r}(x), \qquad (7.4)$$

where $\boldsymbol{r}(x)$ – remainder of division $\boldsymbol{a}(x)\cdot x^{n-k}$ on $\boldsymbol{g}(x)$. The coding rule (7.4) can be realized by such **coding algorithm** for systematic cyclic code:

1 To the word of a primary code $\boldsymbol{a}$ on the right $(n - k)$ zeros are added. It is equivalent to polynomial multiplication $\boldsymbol{a}(x)$ on x^{n-k}.

2 Product $\boldsymbol{a}(x)\cdot x^{n-k}$ divides on the generator polynomial $\boldsymbol{g}(x)$. As a result of division remainder $\boldsymbol{r}(x)$ is defined.

3 The calculated remainder is summarized with the displaced combination $\boldsymbol{a}(x)\cdot x^{n-k}$. Therefore the **allowed code word** is formed as **(**7.4).

Example 7.2 Forming of code word of cyclic code (10, 5).

For the given primary code word $\boldsymbol{a}$ = (10110) we will generate a code word of a cyclic code (10, 5). Polynomial representation of the primary code word will be $\boldsymbol{a}(x) = x^4 + x^2 + x$. Given cyclic code has parameters: $n = 10$, $k = 5$, $r = (n - k) = 5$. From the table 7.1 for example the generator polynomial $\boldsymbol{g}(x) = x^5 + x^4 + x^2 + 1$ is chosen. Next we will fulfill mathematical operations according to algorithm (7.5):

1) $\boldsymbol{a}(x)\cdot x^{(n-k)} = (x^4 + x^2 + x)\cdot x^5 = x^9 + x^7 + x^6$;

2) Division $\boldsymbol{a}(x)\, x^{(n-k)} / \boldsymbol{g}(x)$

$$\begin{array}{l} x^9 + x^7 + x^6 \quad \big| \, x^5 + x^4 + x^2 + 1 \\ \oplus \underline{x^9 + x^8 + x^6 + x^4} \quad\quad x^4 + x^3 + 1 \\ x^8 + x^7 + x^4 \\ \oplus \underline{x^8 + x^7 + x^5 + x^3} \\ x^5 + x^4 + x^3 \\ \oplus \underline{x^5 + x^4 + x^2 + 1} \\ x^3 + x^2 + 1 = \boldsymbol{r}(x) \end{array}$$

3) Polynomial of allowed code word is

$$\boldsymbol{b}(x) = \boldsymbol{a}(x)\cdot x^{n-k} \oplus \boldsymbol{r}(x) = x^9 + x^7 + x^6 + x^3 + x^2 + 1.$$

To polynomial $\boldsymbol{b}(x) = x^9 + x^7 + x^6 + x^3 + x^2 + 1$ there corresponds a word of binary symbols $\boldsymbol{b}$ = (1011001101) in which first four symbols are informational and remaining – additional. **Property of divisibility** of allowed code words on the generator polynomial is widely used for detection of errors in transmission systems.

If $\hat{\boldsymbol{b}}(x) = \boldsymbol{b}(x) + \boldsymbol{e}(x)$ – the received code word containing the errors polynomial $\boldsymbol{e}(x) = e_0 + e_1x + ... + e_n x^n$ as a result of division it is received:

$$\hat{\boldsymbol{b}}(x)/\boldsymbol{g}(x) = \boldsymbol{q}(x) + \boldsymbol{s}(x). \qquad (7.5)$$

Here $\boldsymbol{q}(x)$ – an arbitrary polynomial ("whole"), $\boldsymbol{s}(x)$ – the polynomial of a syndrome equal to remainder of division $\hat{\boldsymbol{b}}(x)$ on $\boldsymbol{g}(x)$. It has degree not above $(n - k - 1)$.

By absence of errors a syndrome $\boldsymbol{s}(x) = 0$. On syndrome form it is possible **to establish** a **location of errors** in the received code word and to use this information for **decoding with error-correction.**

Example 7.3 Syndrome decoding of a cyclic code (7, 4).

The word of binary primary code **a** = (1010) as subject to transmission via the channel with single errors is set. Let's choose the cyclic code ensuring errorless transmission this word in these conditions. From table A.1 we define, that the task can be solved by using of the cyclic code with a generator polynomial $\boldsymbol{g}(x) = x^3 + x^2 + 1$ and parameters $n = 7$, $k = 4$, $q_{\text{cor}} = 1$. We will show, how the **method of syndrome decoding** for correction of single errors is realized. Using algorithm of encoding (7.4), we will generate allowed word $\boldsymbol{b}(x) = (x^6 + x^4 + 1)$. Suppose, that in the channel the single error $\boldsymbol{e}(x) = x^6$ operates. In this case the received word looks like $\hat{\boldsymbol{b}}(x) = \boldsymbol{b}(x) + \boldsymbol{e}(x) = x6 + x^4 + 1 + x6 = x^4 + 1$. We use a rule (7.5) for determination of a syndrome. By syndrome decoding on the syndrome form it is possible to establish an error location (i.e. to fulfill **syndrome decoding**). For this purpose it is necessary to make **the table of syndromes** and of errors polynomials corresponding to them. For compiling of such table it is necessary to take advantage of the equality implying from (7.5) by $\boldsymbol{q}(x) = 0$:

$$\boldsymbol{s}(x) = \boldsymbol{e}(x)/\boldsymbol{g}(x). \qquad (7.6)$$

Outcomes of evaluations are presented to table 7.2 under this formula of syndrome polynomials $\boldsymbol{s}(x)$ for various polynomials of an errors. With a view of presentation a value of syndromes are presented in the form of binary words.

Table 7.2 – Correspondence between syndromes and error polynomials

Error polynomial $\boldsymbol{e}(x)$	x^6	x^5	x^4	x^3	x^2	x	1
Syndrome $\boldsymbol{s}(x)$	$x^2 + x$	$x + 1$	$x^2 + x + 1$	$x^2 + 1$	x^4	x^2	1
Binary syndrome representation	110	011	111	101	100	010	001

Let the polynomial of the received from the channel word look like $\hat{\boldsymbol{b}}(x) = x^4 + 1$. We will fulfill operation of division $\hat{\boldsymbol{b}}(x)/\boldsymbol{g}(x)$:

$$\begin{array}{rl} & x^4 + 1 \quad \big| \, x^3 + x^2 + 1 \\ \oplus & \underline{x^4 + x^3 + x \qquad\quad x + 1} \\ & x^3 + x + 1 \\ \oplus & \underline{x^3 + x^2 + 1} \\ & x^2 + x = \boldsymbol{s}(x) \end{array}$$

From table 7.2 it is discovered, that to such syndrome there error polynomial $\boldsymbol{e}(x) = x^6$ corresponds. Error correction consists of addition of the received code word with an error polynomial

$$\hat{\boldsymbol{b}}(x) + \boldsymbol{e}(x) = x^4 + 1 + x^6 = x^6 + x^4 + 1$$

that coincides with the transmitted allowed word $\boldsymbol{b}(x) = x^6 + x^4 + 1$. To it there a binary word **b** = (1010101) corresponds, in which first four symbols are errorless transmitted symbols of primary code $\hat{\mathbf{a}}$ = (1010) (as the used code is systematic).

Such codes with cyclic properties find application in practice:

1 **Goley code** (23, 12) – perfect cyclic code with a generator polynomial $\boldsymbol{g}(x) = x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$ and minimal distance $d_{\min} = 7$.

2 **Expanded Goley code** (24, 12) with minimal distance $d_{\min} = 8$ which is received by addition of the general parity checking.

3 **Bose-Chaudhuri-Hochuenghem codes** (**BCH** codes) which form extensive class of a cyclic codes. Bynary BCH codes have parameters: $n = 2^m - 1$, $(n - k) \geq mt$, $d_{\min} = 2t + 1$, where m ($m \geq 3$) and t – any positive integers. Theoretical data on BCH codes is given in Section 10.4 of the textbook [1].

4 **Reed-Solomon codes** (**RS** codes) – a subclass of nonbinary BCH codes with parameters: code symbols are got out of field **GF**(q), $q = 2^m$, m – integer; length of word $n = (q - 1)$, quantity of information symbols $k = (n - 2q_{\text{corr}})$, the minimum distance $d_{\min} = (2q_{\text{cor}} + 1)$. The extension of a code to $n = q$ or to $n = (q + 1)$ is also possible.

The effective using of cyclic properties of allowed words cyclic codes allows to realize enough simple decoding algorithms. It is considered, that realization complexity of cyclic codes decoding algorithms is described by power function $C_{\text{decod}} = n^k$, where the k – small number which size depends on concrete algorithm realization. Examples of encoding/decoding algorithms are more low resulted. Thus the mathematical apparatus of sedate polynomials algebra and the description the discrete linear filters, is presented in Sections 5 and 6 of the manual [3] is widely used.

Example 7.4 Encoder structure of the systematic cyclic code.

Using algorithm (7.5) we will form the block diagram of a cyclic encoder (15, 11), with a generator polynomial $\boldsymbol{g}(x) = x^4 + x + 1$ which is chosen from table 7.1. The scheme of encoder is resulted on figure 7.1.

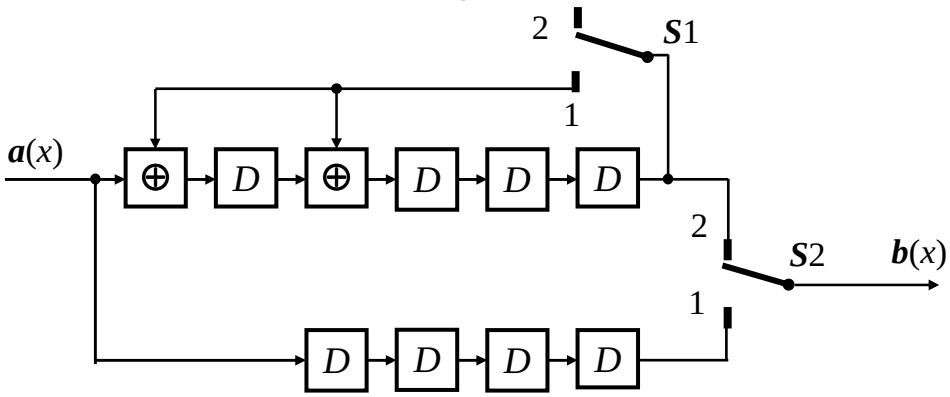

Figure 7.1 – Encoder of systematic cyclic code(15,11)

According to an algorithm (7.4) encoder works as follows. Originally switches *S*1 and *S*2 are in position 1. Eleven information symbols of a coded prime word $\boldsymbol{a}(x)$ are entered at the left into chain of division into a polynomial $\boldsymbol{g}(x) = x^4 + x + 1$. Simultaneously they through consistently connected delay elements arrive on a encoder exit, forming an information part of the allowed code word $\boldsymbol{a}(x)\cdot x^{n-k}$. On first four steps in register cells the divider scheme on a generator polynomial the remainder of a division $\boldsymbol{r}(x)$ is formed. Then switches *S*1 and *S*2 are established in position 2, division process stops, and remainder is read out from an exit of divider and finished in a checking part of final code word $\boldsymbol{b}(x) = \boldsymbol{a}(x)\cdot x^{n-k} + \boldsymbol{r}(x)$.

Example 7.5 Encoder structure of the nonsystenatic cyclic code

Using a coding rule (7.3) for nonsystematic cyclic code we will form the coder block diagram for generator polynomial $\boldsymbol{g}(x) = x^4 + x + 1$. The coding rule (7.3) provides multiplication of polynomials $\boldsymbol{a}(x)$ and $\boldsymbol{g}(x)$. Using structure of a multiplier for polynomials from section 6.1 of manual [3] the encoder scheme we will present on figure 7.2. The important element of coders schemes for cyclic codes is the scheme of division polynomial on a polynomial for an evaluation of a division remainder by coding of systematic code by algorithm (7.4) and also for syndrome evaluation by syndrome decoding on algorithm (7.5). The structure of such divider schemes is considered in Section 6.1 from the manual [3].

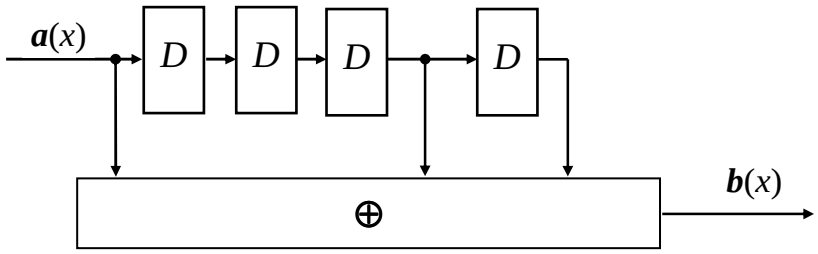

Figure 7.2 – Encoder of the nonsystematic cyclic code

Questions

7.1 What are the key parameters of Hamming codes?

7.2 What are the advantages of cyclic codes?

7.3 Is it possible to use Hamming codes and cyclical codes for correction of single errors? What will parameters of these codes be?

Tasks

7.1 The generator matrix of a code (7, 4) is set:

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

Define the allowed code word of this code $\boldsymbol{b}$ if word of simple code on encoder input $\boldsymbol{a}$ = (1110) is set.

7.2 The parity check matrix of a code (7, 4) is set:

$$\mathbf{H} = \begin{pmatrix} 1\,0\,1\,1\,1\,0\,0 \\ 1\,1\,1\,0\,0\,1\,0 \\ 0\,1\,1\,1\,0\,0\,1 \end{pmatrix}.$$

Give a functional chart of the decoder of this code.

7.4 Consider an example of formation of a allowed code word if a word of a simple code is **a** = (10010).

7.5 By analogy to an example of Section 7.1 make the table of Hamming codes parameters for values r = 2, 3, 4. As these codes have identical minimum distance, compare them on suitability for realization in practical systems. Formulate the recommendation and a substantiation of application of the best (in your opinion) code from this list.

7.6 For Hamming code recommended in the previous Section, form generator and parity check matrixes.

7.7 By the rules stated in Exercise 4.1 define value of minimum distance by a generator matrix of code from the Task 7.5.

8 DECODING NOISE IMMUNITY OF BLOCK CODES

8.1 Decoding noise immunity of block codes

Let's define of an error probability by decoding of block codes in the binary symmetric channel. We will consider code (n, k) with minimal distance $d_{\min}$. In such channel errors in sequentially transmitted code symbols (signals) occur independently with probability p (decoding in discrete channel without memory). Then the probability of that on length of the block n will occur a error multiplicity q, will be equal

$$P(q) = C_n^q p^q (1-p)^{n-q}.$$

Here C_n^q – number of combinations from n elements on q. If the code corrects all errors of multiplicity $q_{\text{corr}} = (d_{\min} - 1)/2$ ($d_{\min}$ – odd) and less then the probability of reception on decoder output the word with not corrected errors will be equal

$$P_{\text{err word}} = \sum_{q=q_{\text{corr}}+1}^{n} P(q).$$

Hence, the probability of erroneous decoding of block will satisfy to an inequality:

$$P_{\text{err word}} \le \sum_{q=q_{\text{corr}}+1}^{n} C_n^q p^q (1-p)^{n-q}. \qquad (8.1)$$

In this expression equality takes place, if the perfect code is used. Parities between parameters n, k and q_{corr} are defined by the concrete chosen code.

Expression (8.1) allows to define the **upper estimation of error probability** of a code words by decoding of block codes in binary symmetric channel without memory. For calculation of probability of an error in concrete information (or

additional) symbols it is necessary to know used decoding algorithm and structure of an error-control code(in particular, a set of distances from a transmitted code word to all allowed words). Such data for block codes are not published in a code tables and for calculations of error probability decoding of code symbols (information or additional) use the **approximated formula** [1]:

$$p_{\mathrm{d}} = \frac{d_{\min}}{n} \sum_{q=q_{\mathrm{corr}}+1}^{n} C_n^q p^q (1-p)^{n-q}. \qquad (8.2)$$

For channels with coherent receiving of signals BPSK the probability of an signal error reception is defined by the formula:

$$p = Q\left(\sqrt{2} h_{\mathrm{s}}\right), \qquad (8.3)$$

where $h_{\mathrm{s}}^2 = E_{\mathrm{s}} / N_0$ – the ratio of the energy spent on transfer of one binary symbol E_{s} to power spectral density of noise N_0 on a demodulator input;

$Q(z) = \frac{1}{\sqrt{2\pi}} \int_z^{\infty} \exp(-\frac{t^2}{2}) dt$ – gaussian Q-function (probability integral) which tables contain the handbooks on probability theory and statistical calculations. For practical calculations it is convenient to use enough **exact approximation**:

$$Q(z) = 0{,}65 \exp[-0{,}44(z + 0{,}75)^2]. \qquad (8.4)$$

The introduction of redundancy by using of error-control coding leads to expansion of a frequency band that occupied with coded signal. If the frequency band in system without coding is ΔF_s (Hz), the using of the code with a rate $R_{\mathrm{code}} = k/n$ demands **expansions of a frequency band**

$$\Delta F_{\mathrm{code}} = \Delta F_s / R_{\mathrm{code}} \ \ (\mathrm{Hz}). \qquad (8.5)$$

I.e. there is an expansion of a frequency band in $K_{\Delta F} = n/k$ time. For codes with low code rate ($n/k > 1$) such expansion can appear appreciable. Therefore the **problem of a code choice** by designing of transmission system consists of search of a **compromise between desirable degree of a noise immunity and expansion of a frequency band of the coded signal**. Under formulas (8.2) and (8.3) taking into account expansion of a frequency band of coded signal according to the formula (8.5) following conclusions allow to draw on **efficiency of application error-control coding**:

1 With growth of a code word length n the error probability of an decoding p_{d} goes down.

2 Codes with the big redundancy (small code rate R_{code}) provide considerable decreasing of a decoding probability error.

3 By using of error-control codes in transmission systems **as a payment for noise immunity increasing is expansion of frequency band** of transmitted signal, caused by the redundancy entered by coding on size:

$$K_{\Delta F} = \frac{n}{k}. \qquad (8.6)$$

8.2 Energy coding gain

For practice the question about expediency of application of error-control codes in telecommunications is important. This question is decided with the taking into account the following. Introduction of redundancy by encoding changes not only expenses of a frequency band for transmitting of coded signals, but also demands the account of a redundancy by energy calculations. Really, according to the formula (8.3) for probability of error registration of channel signals (code symbols) is defined by their energy E_s which with taking into account redundancy of a code appears a little bit less energy E_b spent for transfer of one information symbol (bit). It is followed from equality $kE_b = nE_s$, i.e. $E_s = E_b R_{code}$. Therefore in all power calculations of systems with coding using as a rule the value of the ratio of signal energy spent for transmitting of one information binary symbol (bit) to noise power spectral density E_b/N_0. The probability of erroneous decoding of the block is defined by formulas (8.1) and (8.3) which in argument of function $Q(z)$ include value E_s – the energy of a signal spent for transmitting through the channel of one binary signal (a code symbol). Really, according to formula (8.3) probability of an error registration of channel signals (code symbols) is defined by their energy E_s which taking into account redundancy of a code, appears a little bit less energy E_b spent for transmitting of one information symbol (bit). Then used in power calculations of systems with coding the relation of energy E_b to noise power spectral density N_0 can be designated as $h_b^2 = E_b / N_0$ .Taking into account relation of signal energy E_s and bit energy E_b (8.3) the value entering into settlement formula will be $h_b^2 = R_{code} h_s^2$. Then taking into account expenses of energy for transmitting of additional symbols of a redundancy code (8.3) it is possible to present the formula as follows:

$$p = Q(\sqrt{2R_{code}} h_b), \qquad (8.7)$$

and bit error probability by expression (8.2). If necessary to define probability of an error in channel without coding it is enough to take advantage of the formula (8.3), having put $R_{code} = 1$:

$$p = Q(\sqrt{2} h_b). \qquad (8.8)$$

Exercise 8.1 Decoding noise immunity of block code.

Let's take advantage of the formula (8.8) for calculations of an error probability with optimum receiving of signals BPSK in channel without coding. Results of calculations are resulted in table 8.1. Initial parameter for calculations is the relation a signal/noise on demodulator input h_b^2. The used in practice value h_b^2 (dB) is defined by formula $h_b^2\,(\text{dB}) = 10 \lg h_b^2$. In table 8.1 settlement data by definition of error probability by optimum receiving of signals BPSK (formula (8.8)) including argument z of the function $Q(z)$.

The dependence curve $p = f(h_b^2$, dB) constructed on these data (BPSK) is resulted on figure 8.1.

Under formulas (8.2), (8.7) we will define of an bit error probability by decoding in the channel with BPSK words of cyclic code average length (31, 26) with parameters R_{code} =0,84, d_{min} = 3, q_{cor} = 1. The code is chosen from table A.1.

Results of calculations are presented on figure 8.1 (curve «Code (31,26)»).

Table 8.1 – Calculation of signals BPSK noise immunity

h_b^2, dB	h_b	R_{code}	z	p
1	1,12	1	1,59	$5{,}8\cdot10^{-2}$
2	1,26	1	1,18	$3{,}9\cdot10^{-2}$
3	1,41	1	2,00	$2{,}4\cdot10^{-2}$
4	1,59	1	2,24	$1{,}3\cdot10^{-2}$
5	1,78	1	2,52	$6{,}1\cdot10^{-3}$
6	2,00	1	2,82	$2{,}4\cdot10^{-3}$
7	2,51	1	3,17	$7{,}9\cdot10^{-4}$
8	2,82	1	3,99	$2{,}0\cdot10^{-4}$
10	3,16	1	4,47	$4{,}2\cdot10^{-6}$

Table 8.2 – Calculation of a decoding noise immunity of the cyclic code

Modulation method BPSK, cyclic code (31, 26)					
h_b^2, dB	h_b	R_{code}	z	C_{31}^2	p_d
1	1,122	0,84	1,454	465	0,26
2	1,259	0,84	1,632	465	0,13
3	1,413	0,84	1,831	465	$5{,}4\cdot10^{-2}$
4	1,585	0,84	2,054	465	$1{,}9\cdot10^{-2}$
5	1,778	0,84	3,305	465	$5{,}1\cdot10^{-3}$
6	1,995	0,84	2,586	465	$1{,}0\cdot10^{-3}$
7	2,239	0,84	2,902	465	$1{,}5\cdot10^{-4}$
8	2,512	0,84	3,562	465	$1{,}4\cdot10^{-5}$
9	2,818	0,84	3,653	465	$7{,}2\cdot10^{-7}$
10	3,162	0,84	4,099	465	$1{,}9\cdot10^{-8}$

In all energy calculations of systems with coding it is used as a rule value of the relation of energy of the signal spent for transmitting of one information binary symbol (bit) to power spectral density of noise h_b^2 which is considered as **a uniform criterion of power expenses** for an information transfer through channel with coding and without it. Size change h_b^2 shows **efficiency of application of error-control code**. The effect of errors decrease on decoder exit can be used on a miscellaneous.

Error-control coding provides **reduction of the error probability** in the received messages. It is well visible from comparison of curves $p = f(E_b/N_0)$ on figure 8.1 for cases of an information transfer by method uncoded BPSK and with using of a

cyclic code (31, 26). It is visible, that by using of an error-correcting code it is possible to admit certain decrease in a channel signal/noise ratio and to receive accordingly a energy coding gain ***g*** (dB). The **energy coding gain** from error-control coding g is equal to a **difference** of values E_b/N_0 necessary for maintenance of bit error probability in transmitted data by both absence and using encoding. The value of gain can be defined at various levels of bit error probability p on demodulator and decoder outputs. Told it is illustrated by the curves of a noise immunity presented on figure 8.1.

In particular, for the data resulted on figure 8.1 value gain is g = 1,55 dB at $p = 10^{-5}$.

Gain values g is widely used for a **choice of codes by designing** of transmission systems. Values g received at use of cyclic codes in channels with BPSK are resulted in table A.2 of Attachment A.1.

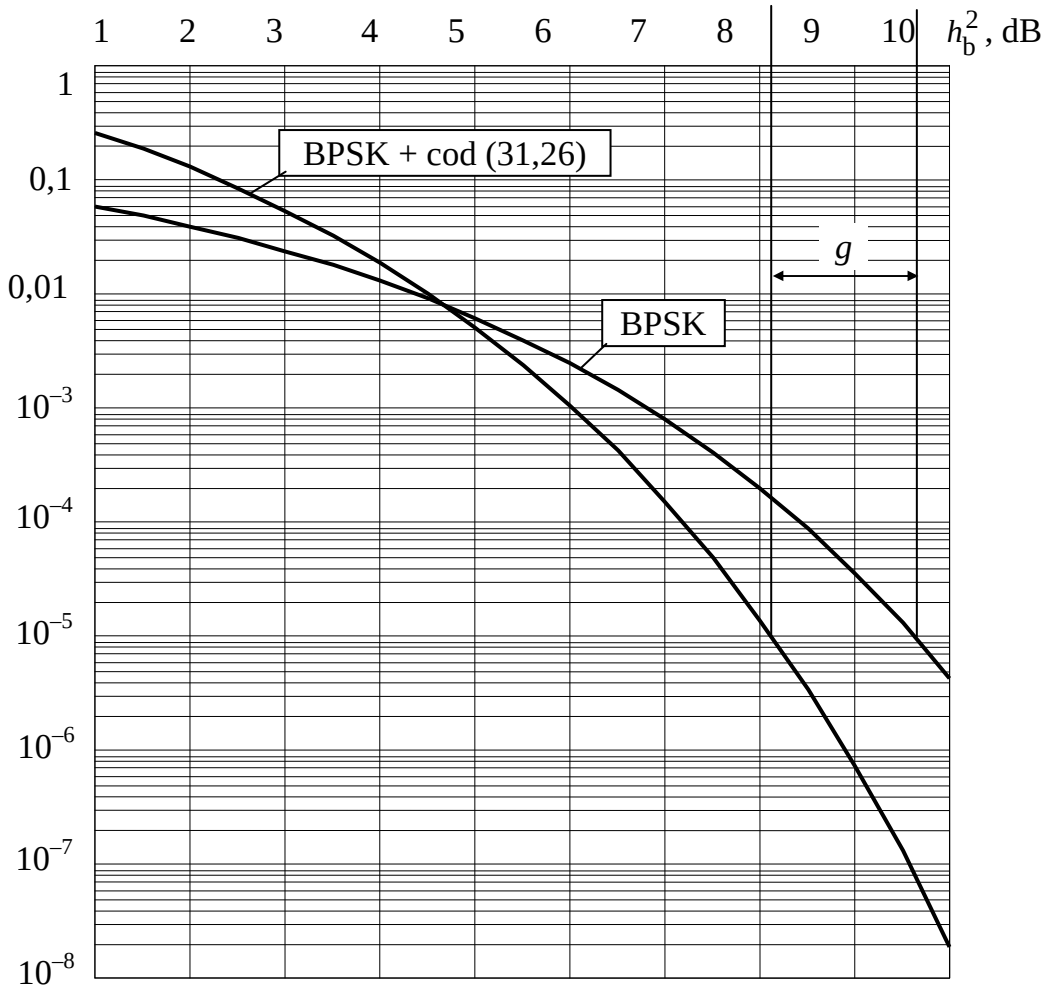

Figure 8.1 – Decoding noise immunity of cyclic code

Example 8.1 Optimisation of a cyclic code parameters.

Let's consider the optimisation procedure of cyclic codes parameters used in the binary symmetric channel with signals BPSK for purpose of a maximum energy coding gain from error-control coding provided by factor of signal spectrum band expanding will not exceed $K_{\Delta F}$ = 2 (double expansion of a signal frequency band in the channel). Preliminary, under table A.2 of Attachment A.1 we will make selection

of cyclic codes which can meet requirements on band expansion factor ($K_{\Delta F} < 2$, $R_{code} > 0,5$). Results of such selection are shown in table 8.3. In table columns values of code rate are specified. In cells in the lines the gain values (in dB) for various lengths of code word n are presented. Under table A.1 of Attachment A.1 we select the cyclic codes with block length $n = 255$ with rate which is closed to optimum rate $R_{code} = 0,8$. It is visible, that the greatest value of gain $g = 4,0$ dB is reached at using enough long cyclic codes with word length $n = 255$. In table 8.4 parameters of the optimum cyclic code are shown.

Table 8.3 – Parameters of a cyclic codes meeting requirements on a code rate

Word length n	Code rate R_{code}			
	0,5	0,6	0,7	0,8
63	2,7	2,8	2,7	2,1
127	3,4	3,5	3,3	2,8
255	3,9	4,0	3,8	3,3

Table 8.4 – Characteristics of an optimum cyclic code

n	k	q_{corr}	Code rate R_{code}	Gain g, dB
255	207	6	0,811	4,0

The selected code (255, 207) provides a power gain 4,0 dB at rate $R_{code} = 0,811$. Factor band expansion $K_{\Delta F} = 1,23$ not exceeding preset value max $K_{\Delta F} = 2$.

Questions

8.1 What parameters of block error-control codes is the error probability by decoding in binary symmetric channel defined?

8.2 How is the energy coding gain defined?

8.3 What are the reasons of signal frequency band expansion with coding?

Tasks

8.1 By a technique stated in the Example 8.1 define parameters and generator polynomial of the cyclic code providing the minimum expansion of a signal frequency band by energy coding gain $g > 3,0$ dB.

8.2 By data from table A.2 Attachment A.2 construct the dependence family of a energy coding gain g from code rate for various lengths of the code word for the cyclic code. Draw conclusions on influence of length of the block on the gain value.

8.3 By data table A.2 of Attachment A.2 construct dependences of a necessary code rate from a demanded energy coding gain g for various lengths of the code word for the cyclic code. Draw conclusions on influence of a word length on the exchange parities between gain and factor of signal band expansion.

9 STRUCTURE AND CHARACTERISTICS OF CONVOLUTIONAL CODES

9.1 Description methods of convolutional codes

Convolutional codes (CC) form a subclass of **continuous codes**. The name «convolutional code» occurs that the result of coding on encoder exit is formed as

convolution of coded information sequence with pulse response of encoder. Encoder of CC contains one or several registers from delay elements and converter of information sequences into code sequences. Coding process is made continuously. The scheme of simple encoder is shown on figure 9.1.

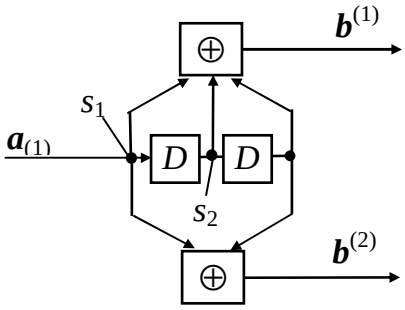

Figure 9.1 – Encoder of CC

Information binary symbols ***a*** arrive on input of the register with *K* delay elements *D*. On exits of Module-2 adders code symbols $\boldsymbol{b}^{(1)}$ and $\boldsymbol{b}^{(2)}$ are formed. Inputs of adders are connected to certain inputs of encoder register elements. The switch *K* on a encoder exit establishes send sequence of code symbols to channel. During one input information symbol it is formed two output code symbols.

Code rate is $R_{code} = k/n$, where k – number of the information symbols simultaneously arriving on inputs of encoder, and n – number of code symbols corresponding to them on encoder exits. Code rate in this example is equal $R_{code} = 1/2$. Coding with other speeds is possible. Convolutional encoder as a finite state machine with final number of states can be described by state diagram. It is considered to be state as symbol set on the inputs of register delay elements. For example, symbols (s_1, s_2) designate **encoder state** on figure 9.1. The **state diagram** represents the **directed graph** who describes all possible transitions of encoder from one state into another and also contains encoder output symbols of the which accompany these transitions.

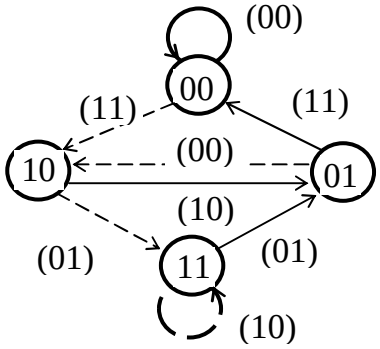

Figure 9.2 – State diagram

Example of the encoder state diagram is shown on figure 9.2. It contains four possible encoder states (s_1s_2) = 00, 10, 11 and 01 and possible transitions.

Symbols about arrows designate symbols on a encoder output ($b^{(1)}b^{(2)}$) corresponding to the given transition. Continuous lines note the transitions made at receipt on encoder input of the information symbol 0 and dotted – by the receipt of a symbol 1. Originally the encoder is in state 00, and receipt on its input of information symbol $a = 0$ translates it also in state 00. Thus on an encoder output there will be symbols ($b^{(1)}b^{(2)}$) = 00. On diagram this transition is designated by loop "00" leaving a state 00 and again coming back in this state. Further, at symbol receipt $a = 1$ the encoder passes in state 10 thus on output there will be symbols $b^{(1)}b^{(2)} = 11$. This transition is designated by dashed line from a state 00 into state 10. Further, receipt on an input of the coder of information symbols 0 or 1 is possible. Thus the coder passes into state 01 or 11, and symbols on output will be 10 or 01 accordingly. Process of the diagram forming comes to an end, when all possible transitions from each state in all others will be seen. The **trellis diagram** (trellis) is development of the state diagram in a time. On trellis the states are shown by knots. The **states** are connecting by lines. After each transition from one state into another there is a displacement on one step to the right. Example of trellis diagram is shown on figure 9.3. Trellis diagram gives evident representation of all allowed ways

which are analogues of allowed code words of a block codes. On them the encoder can move ahead by encoding. A **unique way through a trellis** corresponds to each information sequence on a encoder input.

In particular by a dotted line the way on a trellis …11100001… is shown corresponding to input information sequence …1011… For description of encoder work the sequence of input and output symbols it is convenient to representing with use of delay operator D in the form of infinite series

$$\boldsymbol{a}_{(i)}(D) = a_{(i)0}D^0 + a_{(i)1}D^1 + a_{(i)2}D^2 + ...,$$
$$\boldsymbol{b}^{(j)}(D) = b_0^{(j)}D^0 + b_1^{(j)}D^1 + b_2^{(j)}D^2 + ...$$

Here indexes in brackets designate:

i – number of encoder input, $1 \le i \le k$;

j – number of encoder output, $1 \le j \le n$.

Indexes without brackets (0, 1, 2, ...) designate discrete time moments.

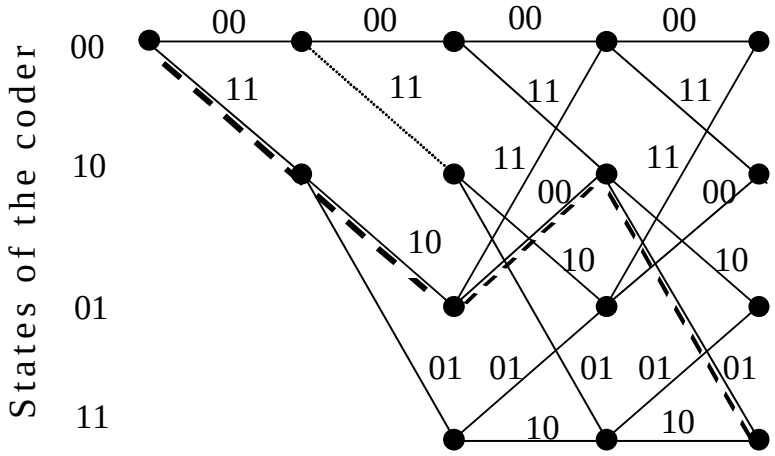

Figure 9.3 – Trellis diagram

For exposition of convolution coding it is used concept of the generator polynomial.

The convolution code will be completely set, if encoder scheme is known:

– An amount of encoder inputs k;

– An amount of encoder outputs n;

– Lengths of each registers K_i;

– Connections of adders with register cells are specified.

For codes with rate $R = 1/n$ connection of j-th adder ($1 \le j \le n$) with cells of shift register is described by the representation of **generator polynomial:**

$$\boldsymbol{g}^{(j)}(D) = g_0^{(j)} + g_1^{(j)}D + g_2^{(j)}D^2 + ... + g_\nu^{(j)}D^\nu. \qquad (9.1)$$

Here $g_k^{(j)} = 1$ if connection of j-th adder with k-th register cell exists, and $g_k^{(j)} = 0$ if such connection is absent.

Coding process can be presented as multiplication of a generator polynomial $\boldsymbol{g}_{(i)}^{(j)}(D)$ on an input information sequence $\boldsymbol{a}_{(i)}(D)$:

$$\boldsymbol{b}^{(j)}(D) = \boldsymbol{a}_{(i)}(D)\boldsymbol{g}_{(i)}^{(j)}(D), \quad 1 \le i \le k, \quad 1 \le j \le n. \qquad (9.2)$$

For example, the encoder on figure 9.1 is characterised by generator polynomials $\boldsymbol{g}^{(1)}(D) = 1 + D + D^2$ and $g^{(2)}(D) = 1 + D^2$ or, noting sequence of a factors g_k in the form of binary words, we receive $\boldsymbol{g}^{(1)} = (111)$ and $\boldsymbol{g}^{(2)} = (101)$. For long codes it is used often the **octal form.** In this case generator polynomials will be presented so:

$$\boldsymbol{g}^{(1)} = (7) \text{ and } \boldsymbol{g}^{(2)} = (5), \text{ or } \boldsymbol{G} = (\boldsymbol{g}^{(1)}, \boldsymbol{g}^{(2)}) = (7, 5).$$

Coding process can be described also with using of generator matrixes (accordingly, check matrixes). It is possible to familiarise with this material under the manual [2, Section 3.4, p. 114] more in detail.

9.2 Key parameters and classification of convolutional codes

Code rate is defined as

$$R_{\text{code}} = k/n, \tag{9.3}$$

where k – an amount of information symbols simultaneously arriving on k encoder inputs, n – an amount of code symbols corresponding to them on n encoder outputs.

It is used several parameters for definition of memory length by coding. The **length of encoder register** K is equal to an amount of delay elements containing in encoder scheme. Length of encoder register often apply to memory definition by coding with rate $R_{\text{code}} = 1/n$, when encoder contains one register. The encoder represented on figure 9.1 has register length $K = 3$. If encoder contains some inputs ($k > 1$) so lengths of registers connected to each input, can be various. In this case it is defined a code constrained length.

The **code constrained length** on each input is defined by the higher degree of corresponding generator polynomials

$$\nu_i = \max [\deg \boldsymbol{g}_{(i)}^{(j)}(D)].$$

The resultant code constrained length is defined by the sum:

$$\nu = \sum_{i=1}^{k} \nu_i \,. \tag{9.4}$$

For codes with one register ($k = 1$) the values ν and K are connected by a simple relation

$$\nu = K. \tag{9.5}$$

For comparison of a decoding algorithm complexity it is used complexity performance. As it was marked earlier, development of trellis diagram consists in a repetition of the same step (see figure 9.3). Diagram complexity is accepted to define an amount of branches on a step of trellis diagram. The number of states of a trellis is defined by number of variables $K = \nu$ on inputs of register elements. As a result **complexity** of one **trellis step** can be defined an amount of branches on this step

$$C = m^{(\nu + k)} \tag{9.6}$$

The decoding noise immunity depends on distance properties of code sequences on encoder input. Thus for binary codes the distance between sequences is often estimated in Hamming metric.

Free distance of a convolution code d_f – is the minimum distance between two arbitrary semi-infinite sequences on the encoder output which differing from the first branch. For short codes free distance can be defined under the state diagram. If the binary code diagram is set free distance is equally to minimum Hamming weight of a way under the diagram from a state 00 in the same state (excepting a loop at this state). On the diagram figure 9.2 it is visible, that free distance $d_f = 5$. On the value of free distance judge about control properties of convolution codes. In particular, if two ways on encoder output, going out from one state on the trellis diagram, differ in Hamming metric on the value d_f, that by decoding on a minimum distance (with analogy to a case of block codes decoding (see Section 3.1)) the **multiplicityof corrected errors** is defined by expression

$$q_{\text{corr}} \le \frac{d_f - 1}{2}, \quad (d_f \text{ is odd}). \tag{9.7}$$

The free distance is used for an estimation of a noise immunity of convolution codes decoding with decoding algorithms by a maximum a posterior probability or close to them (Viterbi algorithm etc.). In a systematic code on k (from n possible) encoder outputs there are information sequences of transmitted symbols, and on remaining $(n - k)$ exits – the sequences of the additional symbols formed as linear combination of information symbols. By rate $R_{\text{code}} = 1/2$ **generator polynomials** of a **systematic code** look like

$$\boldsymbol{g}^{(1)}(D) = 1 \text{ and } \boldsymbol{g}^{(2)}(D) = g_0^{(2)} + g_1^{(2)}D + g_2^{(2)}D^2 + ... + g_\nu^{(2)}D^\nu.$$

Systematic codes allow to receive on a receiving site an estimation of information symbols, without decoding or any other processing of received symbols. Nonsystematic codes do not possess such property. As well as in case of a block codes the using of convolution coding with rate $R_{\text{code}} = k/n$ leads to expansion of a signal frequency band in the channel. Thus the of **band expansion factor** is defined by expression:

$$K_{\Delta F} = \frac{n}{k}. \tag{9.8}$$

By small code rates the considerable band expansion becomes unacceptable, therefore try to apply encoding with a high code rate. Practically, a choice of code parameters is made on the basis of the **compromise**, proceeding from demanded level energy coding gain and admissible value of frequency band expansion factor.

Exercise 9.1 The analysis of code parameters connections.

Using consecutive modification of the structure of initial encoder (7, 5) and corresponding to it state diagram and a trellis (figures 9.1, 9.2 and 9.3) we will establish connections of the encoder parameters k, n, R_{code}, S and generator polynomial with code free distance d_f. We will consider some variants of the codes:

1. Initial code (7, 5). Its scheme is resulted in figure 9.1. The diagram of states is constructed in figure 9.2.

Parameters of the code (7, 5) : $k = 1$, $n = 2$, $K = 2$, $R_{code} = 1/2$, since code is binary (m=2) then $S = 2^K = 4$, free distance $d_f = 5$, code is nonsystematic.

2. Forming of a systematic code (1, 5).

Let modify the first polynomial of an initial code, having left one connection, as shown in figure 9.1. The state diagram will partially vary. The number of a states remains former as the structure of encoder register has not varied. Nonzero branches vary: according to a modification of the first generator polynomial on a place of the first branch numeral it is necessary to write down the first numeral of a state to which this branch is directed (figure 9.4). The code rate also has not varied.

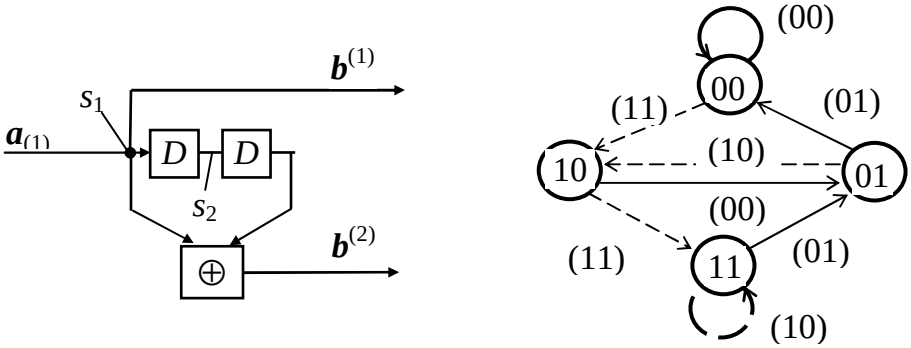

Figure 9.4 – Encoder of CC (1, 5) and its state diagram

Parameters of the code (1, 5) : $k = 1$, $n = 2$, $K = 2$, $R_{code} = 1/2$, since code is binary (m=2) then $S = 2^K = 4$, free distance has decreased $d_f = 3$, code is systematic.

This example illustrates the general conclusion of a coding theory: on the free distance the systematic code appear worse of a nonsystematic codes from which they are organised. Therefore **in practice it is preferred to use the nonsystematic codes**.

3. Forming of a nonredundancy code (1,0).

This, apparently, the "exotic" example allows to reveal a role of the nonzero generator polynomials forming additional symbols (figure 9.5).

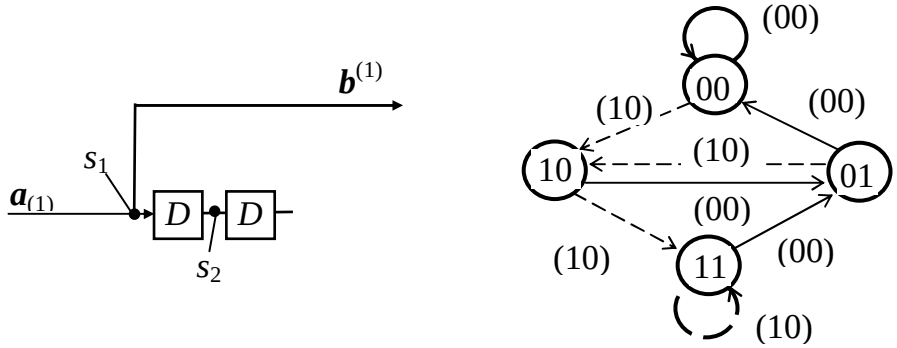

Figure 9.5 – Encoder of code (1, 0) and its state diagram

Parameters of the code (1, 0) : $k = 1$, $n = 1$, $K = 2$, $R_{code} = 1$, since code is binary (m=2) then $S = 2^K = 4$, free distance has considerably decreased $d_f = 1$.

Encoder is systematic without a additional symbols.

Actually, nonredundancy coding is present (memory of the encoder is not used). Therefore the code free distance is equally $d_f = 1$, also corresponds to a rate of the nonredundancy code $R_{code} = 1$. All increment of free distance in the a code is considered in variant 2 spoke presence of nonzero additional symbols.

In Attachment A.3 performances of binary convolutional codes with maximum free Hamming distance for various code rates are given.

Questions

9.1 Name key parameters of convolutional codes.

9.2 What are construction rules of the state diagram?

9.3 What is the connection between the state diagram and the trellis diagram?

9.4 How can we define a free distance under the state diagram?

Tasks

9.1 Generator polynomials ($g^{(1)}$, $g^{(2)}$) = (1101, 1111) are set. Define parameters such code. What are the octal and polynomial representations ($g^{(1)}(D)$, $g^{(2)}(D)$) this code?

9.2 Form a functional scheme of code with such set of the generator polynomials.

9.3 Construct the state diagram and the trellis diagram of such code. Show, how on them to define the free distance of a code. Find a line corresponding to this code in tables of convolutional codes from Attachment A.3. By analogy with Exercise 9.1 analyse a connection of this code parameters with value of free distance. Make generalising conclusions.

9.4 Prepare the trellis diagram of a code (1, 5) from the Exercise 9.1, necessary for an analyse of the Viterbi algorithm.

10 DECODING ALGORITHMS OF CONVOLUTIONAL CODES

10.1 Classification of decoding algorithms

By the receiving for purpose of optimum solution the received sequence of symbols accepted from the channel it is necessary to compare with all possible transmitted sequences. As the number of possible sequences length N for binary code is equally to 2^N by the big sequence lengths the decoder becomes inadmissible complexity (exponential decoding complexity, see Section 6.3), and optimum decoding practically difficultly realizing. However by big N substantial increasing of transmission fidelity as the noise averages on a long sequence is possible. Therefore the **problem** of **decoding algorithms complexity decreasing** is important. Two groups of decoding methods for convolution codes are known:

1. **Algebraic decoding methods** are based on the use of algebraic properties of code sequences. In some cases these methods lead to a simple realisations of codec. Such algorithms are not optimal, as used algebraic decoding procedures are intended for correction concrete (and not all) configurations of channel errors. Algebraic methods are identified with **«element-by-element reception»** of sequences which for codes with redundancy, as is known, yields the worst results, than «**reception in a whole**». Most simple of algebraic algorithms is the algebraic decoding methods. This

algorithm is so far from optimum and consequently is seldom used, first of all, in systems with a high information rate. More detailed description of threshold algorithm and its modification can be discovered in the manual [2, Section 3.6.3].

2. **Probability decoding methods** are considerably is nearer to optimal "reception in a whole" as in this case decoder operates with values which proportional to a posteriori probabilities, estimates and compares probabilities of various hypotheses and on this basis carries out decision about transmitted symbols.

Algebraic algorithms operate with limited alphabet of input data for which deriving on an exit of continuous channel it is necessary to fulfil **quantization** of received signal with noise. Processes of elaborating of signals in an exit of the demodulator for antipodal signals are shown on figure 10.1 where are presented:

a) – forms of antipodal signals in sampling time on input of decision device of demodulator;

b) – binary quantization by hard decision;

c) – octal quantization by soft decision.

In the simple case it is made quantization of each channel symbol in sample time on two levels (named in the literature as **«a hard decision»**). Thus hard decision is presented by one binary symbol. It is shown on figure 10.1, *b*. By hard decision number of quantization levels is $L = 2$. By a **soft decision** number of quantization levels is $L > 2$ (figure 10.1, *c*). By soft decision the quantized output describes magnitude of decoded signal plus noise more precisely that raises a noise immunity.

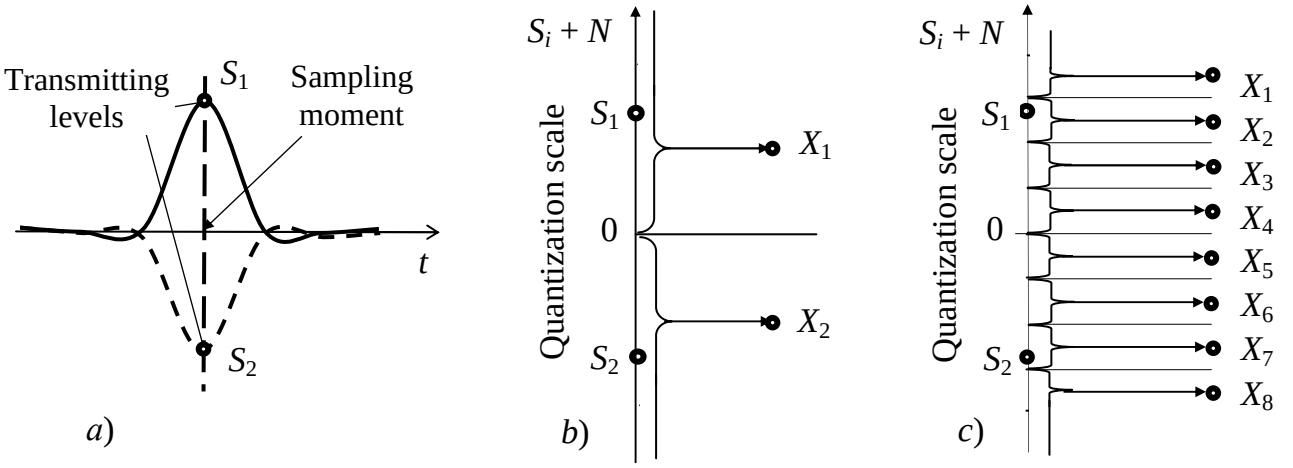

Figure 10.1 – Work of decision device in demodulator

Two basic probability algorithms for decoding of convolution codes, and also their various modifications are known **.**

Sequential decoding algorithm ensures arbitrarily small error probability by nonzero messages transmission rate through the channel. By sequential decoding the search of way through code lattice, corresponding to the transmitted informational sequence is made. Sequential decoding is used for decoding of long convolutional codes. The detailed description of sequential decoding algorithm has presented in the book [4, Section 13.18]. Other variety of probability algorithms is the algorithm based on a principle of dynamic programming, and known as Viterbi algorithm.

Dynamic programming principle has been formulated in 1940 by R. Bellman. It has wide application in control theory. In 1970 the dynamic programming

in form of decoding algorithm for convolutional codes has been applied by A. Viterbi to solving of telecommunication problems (Viterbi algorithm).**Viterbi algorithm** finds wide application and realizes search of maximum probable way through code trellis with rejection of a part of least probable variants of decoded paths. Viterbi algorithm is characterized by a constant of computing work, however complexity of decoder Viterbi grows, as by all full search algorithms under the exponential law from code length. Therefore Viterbi algorithm is used for decoding of short convolutional codes.

10.2 Viterbi algorithm for decoding of convolutional codes

Let's consider Viterbi algorithm on example of the code with rate $R_{\text{code}} = 1/n$. Let, since a moment $t = 0$, on encoder input the information sequence of length L symbols $\boldsymbol{a}_L = (a_0, a_1, \ldots, a_{L-1})$ moves. On encoder output there will be a sequence of symbols $\boldsymbol{b}_L = (b_0, b_1, ..., b_{L-1})$. Encoder states at the moment t define as a set of ν information symbols $\boldsymbol{w_t} = (a_t, a_{t-1}, ..., a_{t-L+1})$. Trellis diagram of code univalently connects the information sequence $\boldsymbol{a_L}$, sequence of the encoder states $\boldsymbol{w_L}$ and the sequence of the output symbols $\boldsymbol{b_L}$. To each branch $\boldsymbol{b}_t$ in channel there corresponds a signal, which can be presented a set of coordinates $\boldsymbol{S}_t = \left(S_t^{(1)}, S_t^2, ..., S_t^{(N)}\right)$, where N – dimension of a signal space. In channel an additive noise operates. Then arriving on decoder input receiving signal sequence will be equal to $\boldsymbol{X_L} = \boldsymbol{S_L} + \boldsymbol{n_L}$, where $\boldsymbol{S}_L = (S_0, S_1, ..., S_{L-1})$ and $\boldsymbol{n_L} = (n_0, n_1, \ldots, n_{L-1})$, $\boldsymbol{n}_t = \left(n_t^{(1)}, n_t^{(2)}, ..., n_t^{(N)}\right)$ is N-dimensional vector of a noise.

Decoding consists in tracing through a code trellis of a way with maximum a posteriori probability. It is possible to specify the decoded way to one of kinds: by set of estimations of code branches $\boldsymbol{S}_L = (S_0, S_1, \ldots, S_{L-1})$ which making a way, by the sequence of estimations of the encoder states $\boldsymbol{w}_L = (w_0, w_1, \ldots, w_{L-1})$, by the sequence of estimations of information symbols on the encoder input $\boldsymbol{A}_L = (a_0, \ldots, a_{L-1})$ which coincide with the first symbols of state estimations $\boldsymbol{S} = (s_1, \ldots, s_{t-\nu+1})$. The sequence $\boldsymbol{X_L}$ will be decoded with the minimum error probability if from all possible ways to choose estimation $\boldsymbol{S}_L$ for which a posteriori probability $P(\boldsymbol{S}_L/\boldsymbol{X}_L)$ is maximum. Transmission of all variants of sequences $\boldsymbol{a_L}$ considers equiprobable. In this case decoding by criterion of a maximum a posteriori probability is equivalent to decoding by criterion of maximum of a probability when estimation $\boldsymbol{S}_L$ ensuring performance of condition $P(\boldsymbol{S}_L/\boldsymbol{X}_L)$=max gets out. In the channel without memory conditional probability $P(\boldsymbol{S}_L/\boldsymbol{X}_L)$ is proportional to product of conditional densities of sum of signal and noise:

$$P(\boldsymbol{X_L}/\boldsymbol{S_L}) = \prod_{t=0}^{L-1} P(\boldsymbol{X}_t / \boldsymbol{S}_t) = \prod_{t=0}^{L-1} P(X_t^{(1)}, X_t^{(2)}, \ldots, X_t^{(N)} / S_t^{(1)}, S_t^{(2)}, \ldots, S_t^{(N)}).$$

In Gaussian channel by the white noise with an one-side power spectral density N_0 each factor of this product looks like:

$$p(\boldsymbol{X}_L / \boldsymbol{S}_L) = (1/\sqrt{\pi N_0})^N \exp\{-[\sum_{i=1}^{N} (X_t^{(i)} - S_t^{(i)})^2]/(2N_0)\}$$

For maximum search we will take the logarithm:

$$\ln P(\boldsymbol{X}_L / \boldsymbol{S}_L) = \ln \prod_{t=0}^{L-1} (1/\sqrt{\pi N_0})^N \times$$

$$\times \exp\{-[\sum_{i=1}^{N} (X_t^{(i)} - S_t^{(i)})^2]/(2N_0)\} = NL \ln(1/\sqrt{\pi N_0}) - \sum_{t=0}^{L-1} \sum_{i=1}^{N} (X_t^{(i)} - S_t^{(i)})^2 /(2N_0).$$

By decoding choose sequence of signals $\boldsymbol{S}_L$ = (S_1, …, S_{L-1}) and sequence of branches univalently connected with it $\boldsymbol{S}_\text{L}$ =(S_0 S_1 … $S_{L\text{-}1}$) which ensures a sum minimum

$$\text{MP} = \sum_{t=0}^{L-1} \sum_{i=1}^{N} (X_t^{(i)} - S_t^{(i)})^2 = \min,$$

which is called as the **metric of the decoded path** (MP). The path metric contains the **metric of branches** (MB)

$$\text{MB} = \sum_{i=1}^{N} (X_t^{(i)} - S_t^{(i)})^2$$

In Gaussian channel the branch metric is proportional to quadrate Euclidean distances between a vector of received sum of a signal plus noise $\boldsymbol{X}_t$ and a vector of signal $\boldsymbol{S}_t$ corresponding to branch of a code $\boldsymbol{A}_t$. In the discrete channel for an estimation of distances it is used Hamming metric. The periodic structure of trellis diagram essentially simplifies comparison and a choice of paths according to decoding rules. The number of states on a trellis is limited, and two by random chosen enough long paths have, as a rule, the common state. Segments of the paths entering into such states it is necessary to **compare** and **choose** a path with the **least metric**. Such **path** is called as **survived**. According to Viterbi algorithm such c**omparison** and **rejection** of segments of path is made **periodically**, on each step of decoding.

On fig. (10.2, *a*, *b*, *c*, *d*) the development of decoding process of convolutional code (5, 7) is shown. On a decoder input symbol pairs from channel arrive (…11,10,00,11,01...) (decoding with hard decision). Figures on branches designate branch metrics, figures about states designate **metric of states** (MS). In an initial time it is supposed, that the decoder is in state 00 and initial metric of this state is MS (00) = 0. If the channel symbols are 11 so metrics of branches 00 and 11 going out this states will be MB (00) = 2 and MB (11) = 0. It is noted on the decoding first pitch. The similar picture takes place and on a following decoding step. The state metrics on this step are defined now as the sums of metrics of entering branches with previous state metrics: MS (00) = 2 + 1 = 3; MS (10) = 2 + 1 = 3; MS (01) = 0 + 0 = 0 and MS (11) = 0 + 2 = 2.

On it the development of trellis diagram for the given code comes to an end. The **algorithm** consists in a **recurring** of **one basic step.** On each of the subsequent diagrams figure 10.2, *a*, *b*, *c*, *d* this steps is represented explicitly. To the beginning of *i*-th step state metrics calculated at the previous stage are stored in memory of the decoder: $\text{MS}^{i-1}(00)$, $\text{MS}^{i-1}(10)$, $\text{MS}^{i-1}(01)$, $\text{MS}^{i-1}(11)$.

On the accepted channel symbols the evaluation of above branch metrics and shaping of four new states metrics is made: $\text{MS}^{i}(00)$, $\text{MS}^{i}(10)$, $\text{MS}^{i}(01)$ and $\text{MS}^{i}(11)$

by a following rule. To each new state lead two ways. For example, to state 00 conduct ways from the previous states 00 and 01. On *i*-th decoding step the decoder calculates metrics of paths as the sums of previous states metrics and entering branches metrics:

$$MS^i(00)\begin{cases}MP^i(00)=MS^{i-1}(00)+MB^i(00),\\ MP^i(00)=MS^{i-1}(01)+MB^i(11);\end{cases}$$

$$MS^i(01)\begin{cases}MP^i(01)=MS^{i-1}(10)+MB^i(10),\\ MP^i(01)=MS^{i-1}(11)+MB^i(01);\end{cases}$$

$$MS^i(10)\begin{cases}MP^i(10)=MS^{i-1}(00)+MB^i(11),\\ MP^i(10)=MS^{i-1}(01)+MB^i(00);\end{cases}$$

$$MS^i(11)\begin{cases}MP^i(11)=MS^{i-1}(10)+MB^i(01),\\ MP^i(11)=MS^{i-1}(11)+MB^i(10).\end{cases}$$

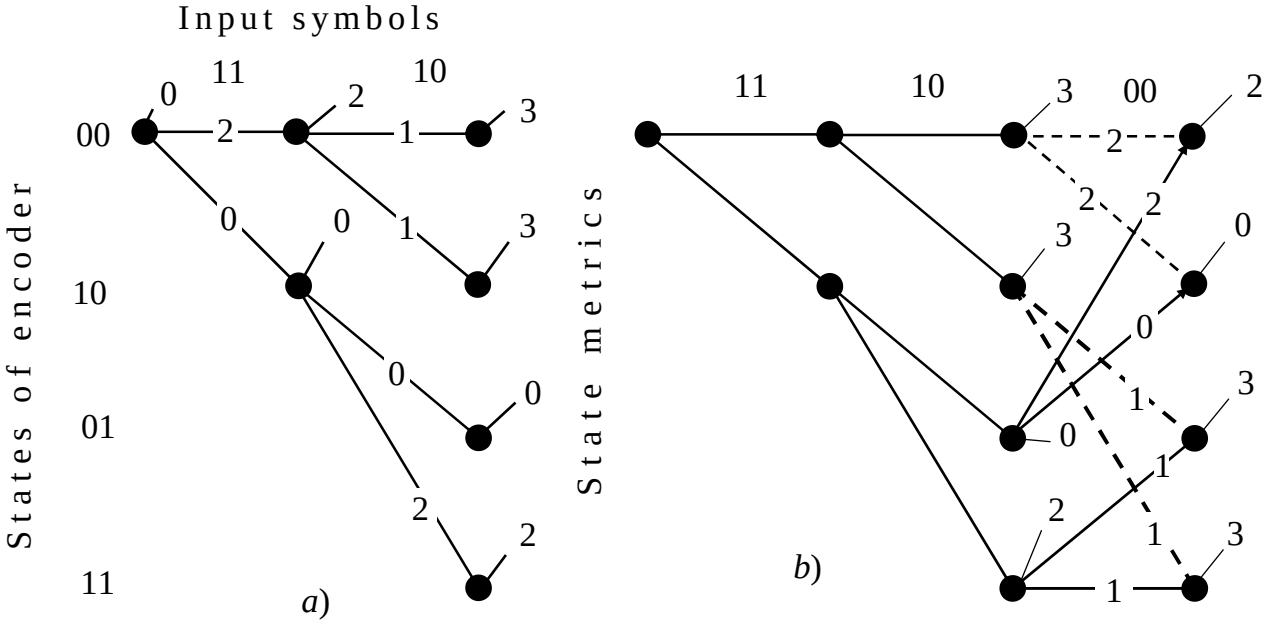

Figure 10.2, *a*, *b* – Decoding process on algorithm Viterbi

According to Viterbi algorithm on each decoding step in each of trellis states the same type operations are made**:**

1) **Addition of metrics** of the previous states with metrics of corresponding branches;

2) **Comparison of metrics** of entering paths;

3) **Choice of paths** with the least metrics which values are used as the metric of the states on the subsequent decoding step. If metrics of compared paths are identical, the choice of one of two path is made in a random way.

Realisation complexity of Viterbi algorithm can be estimated by the amount of branches of the code trellis treated by the decoder at length of decoding *L*, taking into

account complexity of each step of a trellis (see formula (9.6)). **Complexity of decoder Viterbi realisation** can be estimated under the formula:

$$C = m^{(v + k)} \cdot L. \qquad (10.1)$$

On figure 10.3 the structure scheme of Viterbi decoder intended for work with the demodulator of signals QPSK is shown.

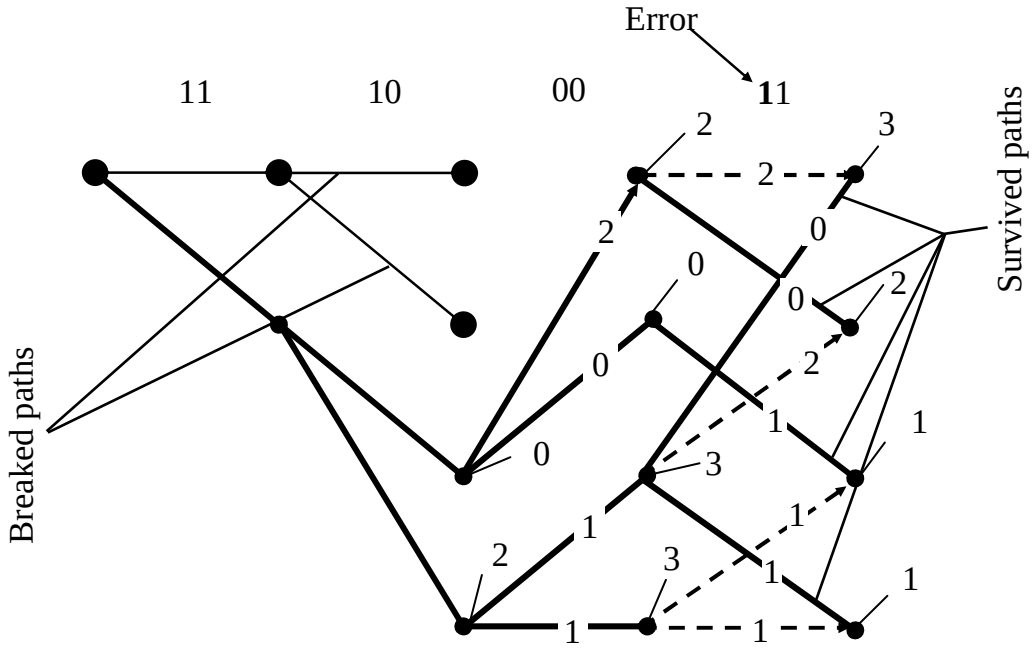

Figure 10.2, *c* – Decoding process on algorithm Viterbi

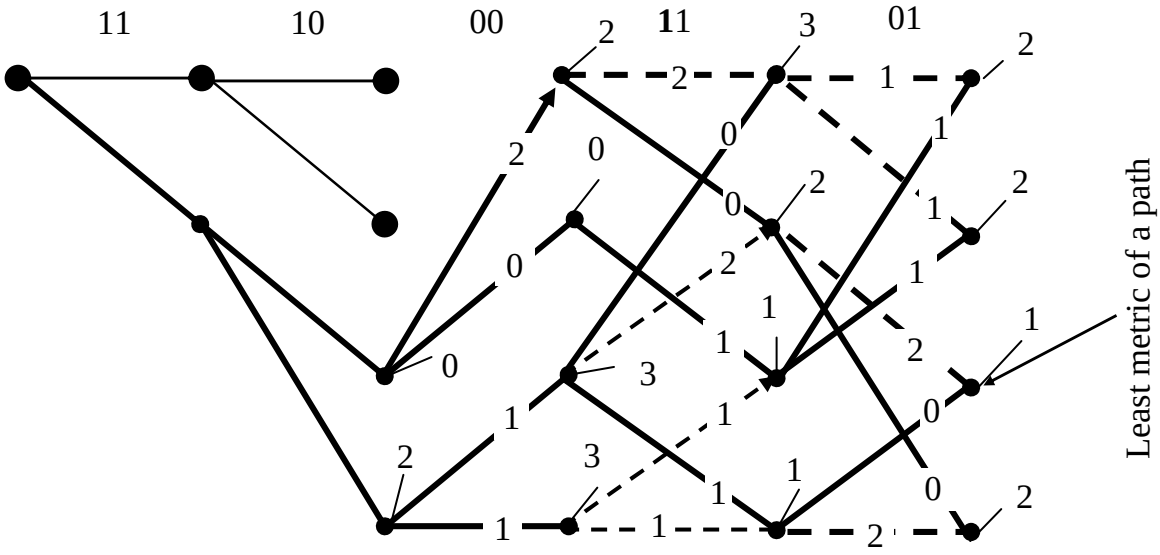

Figure 10.2, *d* – Decoding process on algorithm Viterbi

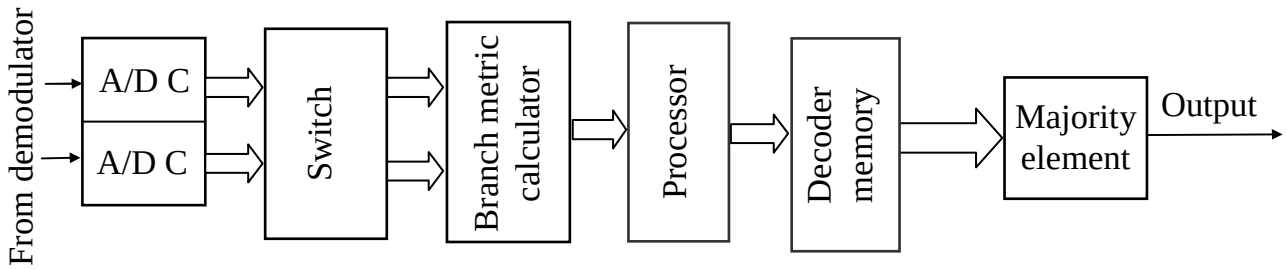

Figure 10.3 – Viterbi decoder structure scheme

The decoder consists from analog/discrete converters (A/D C) in channels *X* and *Y*, the calculator of branch metrics and processor in which operations of **addition**, **comparison** and a **choice** are made, **memory device** of a survived paths, and majority element in which the path with the least metric gets out. The best value of a quantization levels depends on the ratio a signal/noise on input A/D C. By eight quantization levels of losses minimum is ensured at the ratio of a signal magnitude to the quantization step is equal to (4,5...5,5). More detailed description of assigning and work algorithms of the decoder Viterbi block diagram elements of are reduced in the manual [2, Section 3.8.2].

Questions

10.1 Does the realisation complexity of the Viterbi algorithm depend on length of free distance?

10.2 How will increase complexity of Viterbi decoder increase by increasing a code constraint length twice?

10.3 What is the reason of Viterbi decoder complexity raise at using of a soft decision on a demodulator exit?

Tasks

10.1 Prepare the trellis diagram of a code (1, 5) from Exercise 9.1 which necessary for an illustration of Viterbi algorithm. As free distance of this code $d_f = 3$ (according to formula (9.7) the code corrects single errors) trace decoding process by Viterbi algorithm if in the channel is single error and establish the fact of its correction by the decoder.

10.2 Prepare the trellis diagram of a nonredundancy code (1, 0) from the Exercise 9.1 which necessary for an illustration of the Viterbi algorithm. Try to explain by the form the trellis diagram impossibility of error-correction.

11 NOISE IMMUNITY OF CONVOLUTIONAL CODE DECODING

11.1 Decoding error probability of convolutional code

The technique of a decoding noise immunity estimation by convolutional codes is not differed from a technique stated in Section 8.1 for a case of block codes. Here the code rate R_{code}, code distance properties (in a case of the convolutional codes – the free distance d_f), and decoding algorithm is played the main role. By using of probability decoding algorithm (Viterbi algorithm) the approximately expression for bit error probability looks like:

$$p_d = \sum_{k=d_f}^{\infty} w_k P_k \,, \tag{11.1}$$

where P_k – error probability of the way choice on a code trellis;

w_k – spectrum of weights of erroneous ways;

At transmission of code symbols through a channel with BPSK with white noise power spectral density $N_0/2$ is defined so:

$$P_k = Q\left(\sqrt{2kR_{\text{code}}}h_b\right). \tag{11.2}$$

Evaluations under formulas (11.1) and (11.2) show, that in the sum (11.1) by a big ratio signal/noise the first member (by $k = d_f$) has the greatest value, and remaining members of the sum with growth k fast decrease. Therefore in practice they are limited to use of the simplified formula:

$$p_d = w_{df} \cdot Q\left(\sqrt{2 d_f R_{code}} h_b\right). \quad (11.4)$$

As well as by the block coding, comparison of a decoding noise immunity can be made with a noise immunity of coherent receiving of signals with binary phase modulation BPSK. Thus the calculation formula for bit error probability can be received from expression (11.2) having supposed $k = 1$, $R_{code} = 1$:

$$p_{BPSK} = Q\left(\sqrt{2} h_b\right), \quad (11.5)$$

where $h_b = E_b / N_0$ – the ratio of the signal energy expended on transmission of bit E_b to a power spectral density of a noise N_0 on an input of the demodulator.

Exercise 11.1 The analysis of a decoding noise immunity

Let's make a calculations of a bit error probability on exits of the demodulator of signals PM-2 and Viterbi decoder included after it, using formulas (11.5) and (11.4) for next codes :

1. Code (5, 7), $R_{code} = 1/2$, $d_f = 5$, $\nu = 2$;
2. Code (133, 171), $R_{code} = 1/2$, $d_f = 10$, $\nu = 6$.

The calculation results are given in table 11.1 and presented on figure 11.1. In the table given values of argument z are specified function $Q(z)$, used in formulas.

11.2 Energy coding gain

As well as by an estimation of a decoding noise immunity of the block codes (see Section 8) in a case of convolution codes use concept of an energy coding gain.

The **energy coding gain** g is equal to a difference between of values h_b^2 necessary to get the given error probability p by the absence and by the coding use.

Values of error probability level at which the gain is defined depends on the requirements to fidelity of the transmitted digital information. For digital telephony systems a acceptable level of a bit error probability usually makes $p_{acc} = (10^{-5} ... 10^{-6})$. In systems of digital TV transmission try to ensure $p_{acc} = (10^{-10} ... 10^{-11})$. acceptable

The value of coding gain at the given bit error probability p^* can be defined by comparing the arguments of function $Q(z)$ in a formulas for error probability (11.4) and (1.5) for identical probabilities $p_d = p_{BPSK} = p_{acc}$. Calculations show, that gain depends from level of error probability p_{acc} on which it is defined. It is well visible on the curves figure 11.1 representing calculation results from Exercise 11.1. Value of a gain with decreasing of a probability p_{acc} aspires to the limit which in the coding theory name as **asymptotic coding gain**:

$$\text{A-gain} = \lim g(p_{acc} \to 0). \quad (11.6)$$

Comparing arguments in the expressions (11.5) and (11.4) we come to wide used in energy calculations of transmission systems to expression for A-gain in logarithmic units:

$$\text{A-gain} = 10\lg(R_{\text{code}} d_{\text{f}}) \text{ (dB)}. \quad (11.7)$$

Table 11.1 – Calculation of decoding noise immunity

E_b/N_0, dB	Bit error probability on demodulator BPSK output		Bit error probability on decoder output			
			Code (5, 7)		Code (133, 171)	
	z	p_{BPSK}	z	p_{d}	z	p_{d}
1	1,59	$5{,}8\cdot10^{-2}$	2,51	$6{,}1\cdot10^{-3}$	3,55	$6{,}9\cdot10^{-3}$
2	1,78	$3{,}9\cdot10^{-2}$	2,82	$2{,}4\cdot10^{-3}$	3,98	$1{,}2\cdot10^{-3}$
3	2,00	$2{,}4\cdot10^{-2}$	3,16	$7{,}8\cdot10^{-4}$	4,47	$1{,}5\cdot10^{-4}$
4	2,24	$1{,}3\cdot10^{-2}$	3,54	$1{,}9\cdot10^{-4}$	5,01	$1{,}1\cdot10^{-5}$
5	2,52	$6{,}1\cdot10^{-3}$	3,98	$3{,}5\cdot10^{-5}$	5,62	$4{,}1\cdot10^{-7}$
6	2,82	$2{,}4\cdot10^{-3}$	4,46	$4{,}2\cdot10^{-6}$		
7	3,55	$7{,}9\cdot10^{-4}$	5,62	$1{,}1\cdot10^{-8}$		
8	3,99	$2{,}0\cdot10^{-4}$				
10	4,47	$4{,}2\cdot10^{-6}$				

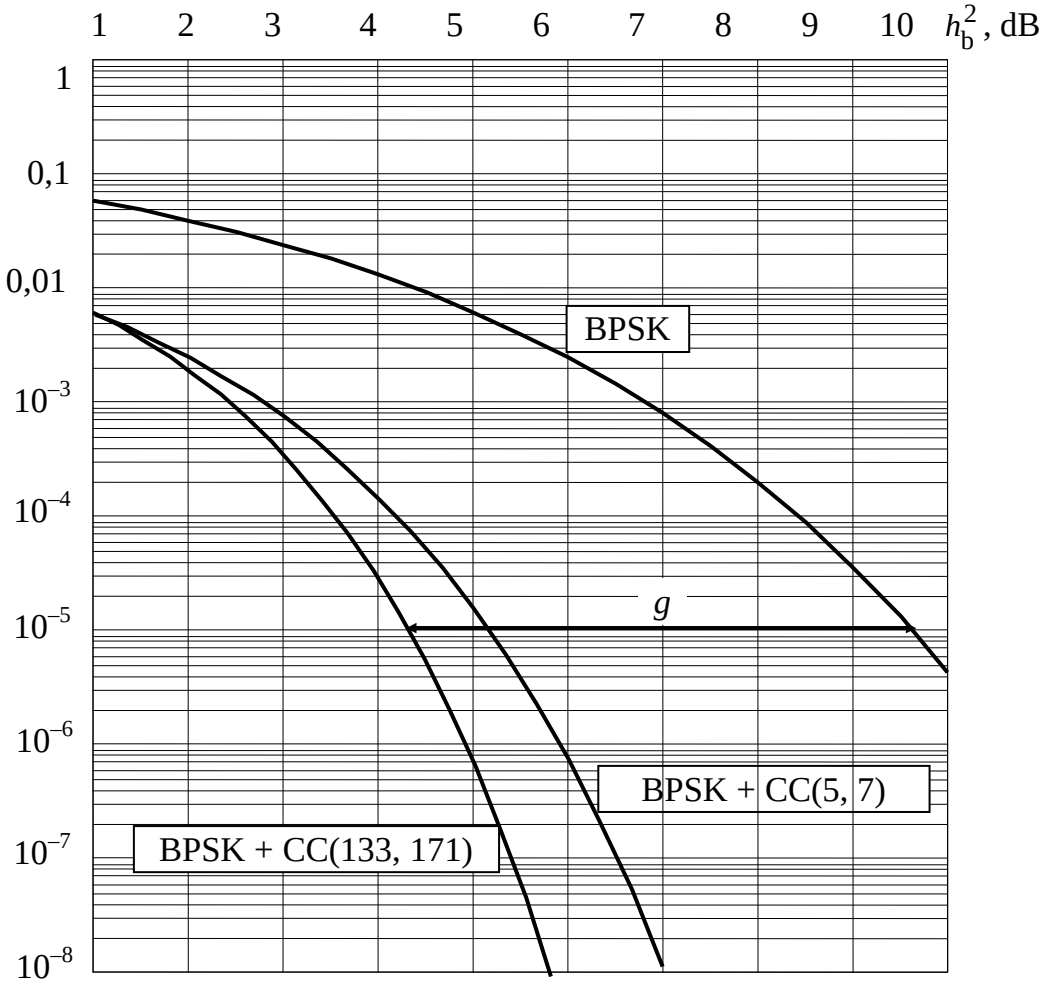

Figure 8.1 – Decoding noise immunity of cyclic code

As A-gain is upper bound of a gain *g* for fast comparison and a choice of codes use A-gain. Values of this A-gain often include in the code tables (see tables of Attachment A.3). In table 11.1 for an example data about convolution codes with various lengths of a code length ν and rate R_{code} are cited. Values of a A-gain are shown. More detailed data are given in tables A.3…A.6 from the Attachment A.3.

Table 11.1 – Characteristics of a convolutional codes

Code rate R_{code}	Code constraint length ν = 4		Code constraint length ν = 6	
	Code	A-gain, dB	Code	A-gain, dB
1/3	25, 33, 37	6,02	133, 145, 175	6,99
1/2	31, 33	5,44	133, 171	6,99
2/3	31, 33, 31	5,23	133, 171, 133	6,02
3/4	25, 37, 37, 37	4,78	135, 163, 163, 163	6,73

Comparison of a gain values is ensured by the cyclic coding (see table 8.3 and figure 8.1) with similar parameters for convolutional codes (see table 11.1 and figure 11.1) shows, that **convolution codes** in a combination to Viterbi decoding algorithm **ensure considerably more gain in comparison with block codes**. It explains wide using of convolution codes in transmission systems for a noise immunity increasing. Typical a using of the code (133, 171) ensuring A-gain = 6,99 dB by the rate R_{code} = 0,5 here is, i.e. at two-multiple expansion of a frequency band of the coded signal. The codecs of such code are developed in the form of the big chips are serially emitted.

Questions

11.1 How does the gain depend from code constraint length?

11.2 How does gain depend from code rate?

Tasks

11.1 Using tables of a convolutional codes from the Attachment A.3 construct the dependence of a gain from a code rate by the fixed values a constraint code length. Explain tendencies of a progress of these dependences.

11.2 Using tables of the Attachment A.3 choose a codes, ensuring A-gain> 6dB and specify parameters of these codes.

12 INCREASING OF DIGITAL TRANSMISSION SYSTEMS EFFICIENCY

12.1 Information, energy and frequency efficiency

Generally the result of work of a transmission systems is defined by an quantity and quality of the transmitted information. The quantity is estimated by an information transmitting rate through a channel R_{ch} (bit per second), and quality – by the values of an error. According to Shannon theorems, the error with a corresponding choice of a transmission method (i.e. modulation/coding) can be made arbitrarily small (see explicitly the materials of the Module 2). At the same time, the transmission rate cannot be above some informational resource named a channel capacity C_{ch}. A. Zuko has suggested to consider as one of indicators of a **system efficiency** the value of mean rate R_{ch} at which the given fidelity of an information

transferring is ensured. Thus the information system efficiency as degree of use of a channel capacity of the channel is defined by relative value $\eta = R_{\text{ch}} / C_{\text{ch}}$. In real conditions the indicator η always is less than unit. The more close η to unit, the more absolutely transmitting information system. Reaching necessary for a transmission rate and fidelity is accompanied by certain expenditures of other **major resources**: **signal power** P_{s} and a **channel frequency band** F_{ch}. Such approach has allowed to introduce the indicators: power $\beta = \dfrac{R_{\text{ch}}}{P_s / N_0}$ and frequency efficiency $\gamma = \dfrac{R_{\text{ch}}}{F_{\text{ch}}}$, uses of the mentioned resources characterizing degree. Here P_s/N_0 – the ratio of a signal power to a power spectral density of noise on a receiver input). Thus, **efficiency indicators by A. Zuko** look like:

Information efficiency of system which define the degree of channel capacity using

$$\eta = \frac{R_{\text{ch}}}{C_{\text{ch}}}; \tag{12.1}$$

Energy efficiency

$$\beta = \frac{R_{\text{ch}}}{P_s / N_0}; \tag{12.2}$$

Frequency efficiency

$$\gamma = \frac{R_{\text{ch}}}{F_{\text{ch}}}. \tag{12.3}$$

12.2 Limiting efficiency of transmission systems and Shannon bound

Indicators β and γ make sense a specific rates, and inverse values $\beta' = 1/\beta$ and $\gamma' = 1/\gamma$ define specific expenses of corresponding resources on an information transferring with unity rate (1 bit per second). For the Gaussian channel with frequency band F_{ch}, the ratio of signal to noise $\rho = P_s/P_n$ and channel capacity $C_{\text{ch}} = F_{\text{ch}} \log(\rho + 1)$ it is possible to establish, that these efficiency indicators are connected by the relation:

$$\eta = \frac{\gamma}{\log(1 + \gamma/\beta)} \quad \text{and} \quad \gamma = \rho\beta \tag{12.4}$$

For ideal system ($\eta = 1$) limiting equation can be defined. According to Shannon theorem by the corresponding transmission methods (coding and modulation) and receiving (demodulation and decoding), the value η can be as much as close to unit. Thus the error can be made as much as small. In this case by a condition $\eta = 1$ it is received limiting equation between β and γ:

$$\beta = \frac{\gamma}{2^{\gamma} - 1}. \tag{12.5}$$

This formula defines of energy efficiency from the frequency efficiency for the **ideal system** ensuring equality of a information rate to a channel capacity. It is convenient to represent this equation in the form of a curve on a plane $\beta = f(\gamma)$ (figure 12.1, a curve «**Shannon bound**»). This curve is limiting and reflects the **best interchanging between β and γ** in the continuous channel (CC).

It is necessary to notice, that frequency efficiency γ varies in limits from 0 to ∞, **energy efficiency is bounded above** by magnitude:

$$\beta_{\max} = \lim_{\gamma \to 0} \beta = \lim_{\gamma \to 0} \frac{\gamma}{2^{\gamma} - 1} = \frac{1}{\ln 2} \approx 1{,}443. \qquad (12.6)$$

Differently, **energy efficiency** of any information transmitting system in a Gaussian channel **can not exceed the magnitude**

$$\beta_{\max} = 1{,}443. \qquad (12.7)$$

Similar limiting curves can be constructed and for any other channels if in formulas (12.2) and (12.3) instead of a rate R_{chan} to substitute expressions for a channel capacity of the corresponding channel. So, in particular, on fig. 12.1 the curve for limiting equation $\beta = f(\gamma)$ the is discrete-continuous channel (D-CC) is shown.

It "is enclosed" in a curve of the continuous channel (CC) that confirms know result of an information theory according to which DN channel capacity of D-CC always is less a than channel capacity of the continuous channel (CC) which is a basis for construction of corresponding D-CC. In real digital systems error probability p always has a final value and informational efficiency is less then a limiting value $\eta_{\max}$. In these cases for the fixed error probability p = const it is possible to define efficiency ratio β to γ and to construct curves $\beta = f(\gamma)$.

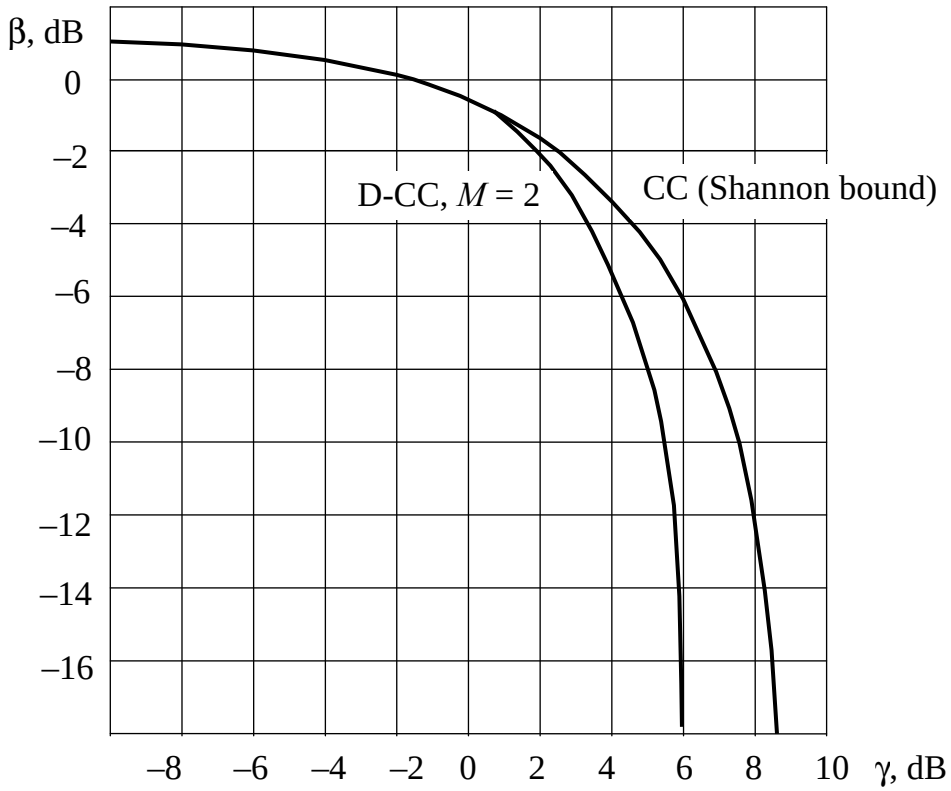

Figure 12.1 – Curves of communication systems limiting efficiency

In coordinates (β, γ) to each variant of a transmission system there will corresponds a point on a plane**.** All these points (curves) **should place below a limiting curve** of «Shannon bound». The place of these curves depends on an aspect of signals (modulations), a codes (coding methods) and a method of the elaborating of a signals (demodulation/decoding). About perfection of the digital telecommunication methods judge on a degree of placing of real efficiency of to the limiting values.

Concrete data about the efficiency of various modulation/coding methods and also their combinations are given in following section.

12.3 Perspective ways of further increasing efficiency

Using the various methods of error-control coding considered in these Module the designer of transmission system owning art of optimisation can **flexibly change of the efficiency indicators** approaching them to the limiting, potentially possible values which are established in the previous section. Efficiency of the digital transmission systems of transmission can be essentially increased by the application of *M*-ary signals and error-control codes, and also their combinations. The choice of signals and codes in these cases is defining for construction **highly effective codems** (the codecs matching among themselves and modems). Comparison of efficiency of systems with *M*-ary signals and error-control codes is convenient for making with using of diagram $\beta = f(\gamma)$, presented on figure 12.1. Thus degree of perfection of a modulation/coding methods and can be estimated, comparing efficiency with limiting values. Results of the efficiency analysis are presented on figure 12.2. At the same time, comparison various modulation/coding methods is convenient for making comparison taking for "reference point" the efficiency of transmission system with modulation QPSK (without error control coding*)*. From among simple methods it is the most effective and widely used method of modulation/coding with indicators $\gamma = 2$, $\beta = -9{,}6$ dB, $\eta \approx 0{,}47$. Conveniently as well that the point representing on figure 12.2 values of efficiency QPSK is arranged in a central part of the diagram. If an origin of coordinates to transfer to a point corresponding QPSK, in a new frame ($\Delta\beta$, $\Delta\gamma$) on a vertical axis the energy gain $\Delta\beta$ in comparison with QPSK, and on a horizontal axis a gain on a specific rate $\Delta\gamma$ will be counted. Let's notice, that all possible modulation/coding methods can be divided into four groups corresponding to four quadrants on the diagram $\beta = f(\gamma)$:

Quadrant III in which the **low efficiency methods** are arranged having rather QPSK loss on β and γ;

Quadrant II including methods with high energy efficiency, ensuring a gain on β in exchange for loss on γ (systems with error-control codes);

Quadrant IV including modulation methods ensuring a gain on γ in exchange for loss on β (systems with *M*-ary *M*-PSK and *M*-APSK signals);

Quadrant I including perspective modulation/coding methods ensuring a simultaneous gain both energy and frequency efficiency.

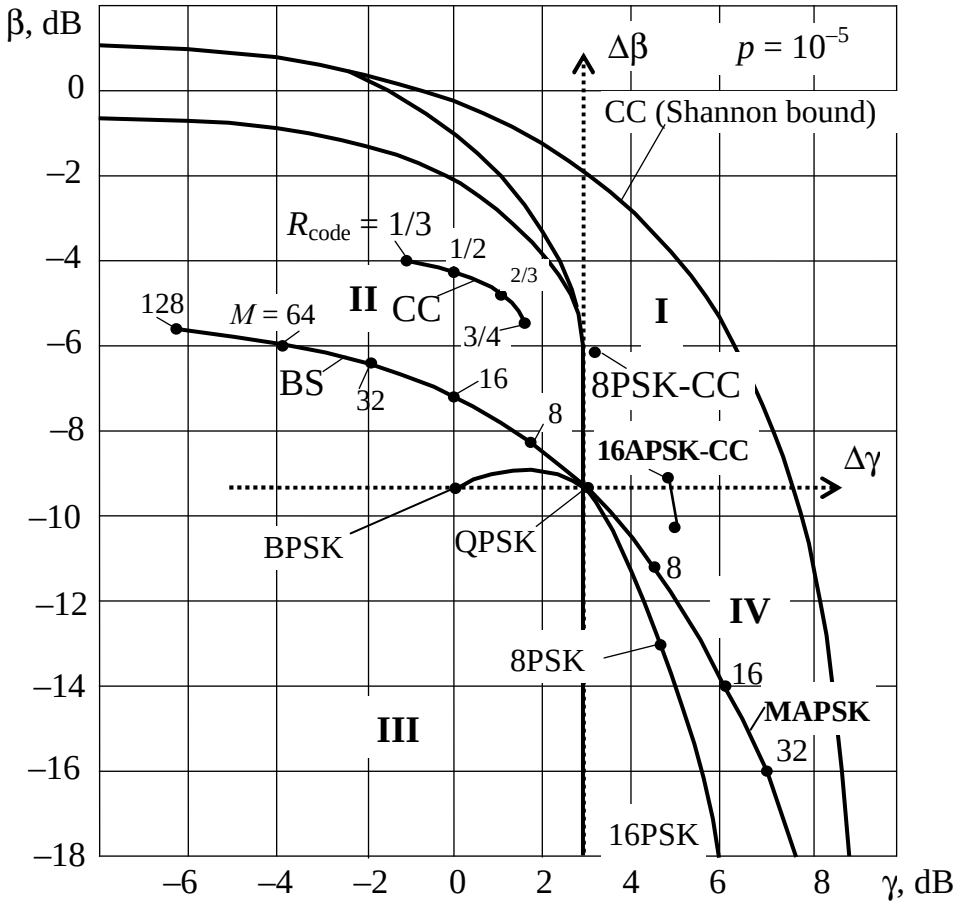

Figure 12.2 – Efficiency of *M*-ary signals and error-correcting codes

Outcomes of calculations show (figure 12.2, Quadrant I), that application such **coded modulations** allows to receive simultaneously a **gain both in energy, and frequency efficiency** and, anyway, to get a gain on one indicator, not worsening another. So, system 8PSK-CC by the using of a convolution code with R_{code} = 2/3 ensures a energy gain $\Delta\beta$ = 2,8 dB without a decreasing of a specific rate γ, and system 16APSK-CC by R_{code} = 1/2 and ν = 3 a gain on a specific rate $\Delta\gamma$ = 2 dB without a drop of energy efficiency β. Information efficiency of these systems is $\eta \approx (0{,}6\ldots0{,}7)$. The detailed analysis such coded modulations is reduced in manual [3, Section 9.2]. Microelectronic progress last decade initiated attempts to realise the potentially possible efficiency, despite of growth of decoding complexity. In 1993 **turbo-codes** have been offered. Turbo-codes has been in details in manual [3, Section 11.1] described. Intensive development of mobile transmission systems has led to the invention of a **time/space coding**, in details described in the manual [3, Section 11.2].

Attachment A. Performances of error-correcting codes

A.1 Performances and generator polynomials of cyclic codes

In table A.1 the short table of performances and generator polynomials of binary cyclic codes is presented. Generator polynomials of codes are given in the octal form where:

n – word length;

R_{code} – code rate;

k– amount of information symbols in the word;

q_{corr} – multiplicity of corrected errors.

Example A.1. Octal representation of generator polynomials.

The code with parameters $n = 7$, $k = 4$, $q_{corr} = 1$ has a polynomial (13) $\rightarrow$ (001.011) $\rightarrow$ (1011) $\rightarrow x^3 + x + 1$.

Table A.1 – Performances and generator polynomials of cyclic codes

n	k	q_{corr}	R_{code}	Generator polynomials
7	4	1	0,57	13
15	11	1	0,73	23
	7	2	0,47	721
31	26	1	0,84	45
	21	2	0,68	3551
	16	3	0,52	107657
	11	5	0,35	5423325
63	57	1	0,9	103
	51	2	0,81	12471
	45	3	0,71	1701317
	39	4	0,62	166623567
	36	5	0,57	1033500423
127	120	1	0,95	211
	113	2	0,89	41567
	106	3	0,84	11554743
	99	4	0,78	624730022327
	92	5	0,97	435
255	239	2	0,94	267543
	231	3	0,91	156720665
	223	4	0,87	75626641375

A.2 Energy coding gain by using of the cyclic codes

In table A.2 the values of energy coding gain g (dB) are given for using of cyclic codes in channels with BPSK.

TableA.2 – Energy coding gain g (dB) by using of the cyclic codes

Block length, n	Code rate R_{code}					
	0,3	0,4	0,5	0,6	0,7	0,8
31	1,2	1,6	1,9	2,0	1,9	1,6
63	2,0	2,4	2,7	2,8	2,7	2,1
127	2,6	3,1	3,4	3,5	3,3	2,8
255	3,2	3,6	3,9	4,0	3,8	3,3

A.3 Performances of binary convolution codes

In tables (A.3…A.6) performances of binary convolution codes with maximum free Hamming distance and rates (1/8…1/2) are given. Generator polynomials are given in the octal form. Labels: ν – code constrained length; d_f – free Hamming distance; d_{fm} – upper bound of free distance; w_{df} – quantity of erroneous ways with weight d_f; A-gain– asymptotic coding gain (dB) by using code in channels with BPSK.

Table A.3 – Code rate R_{code} = 1/8

Code number	ν	Generator polynomials	d_{fm}	d_f	w_{df}	A-gain, dB
1	4	25,27,33,35, 37,25,33,37	32	32	8	6,02
2	5	45,55,57,65, 67,73,77,47	36	36	3	6,53
3	6	115,127,131,135, 157,173,175,123	40	40	1	6,99

TableA.4 – Code rate R_{code} = 1/4

Code number	ν	Generator polynomials	d_{fm}	d_f	w_{df}	A-gain,dB
4	2	5,7,7,7	10	10	2	3,98
5	3	13,15,15,17	13	13	4	5,12
6	4	25,27,33,37	16	16	8	6,02
7	5	51,55,73,77	18	18	5	6,53
8	5	53,67,71,75	18	18	6	6,53
9	6	135,135,147,163	20	20	37	6,99
10	7	235,275,313,357	22	22	11	7,40
11	8	463,535,733,745	27	27	4	8,29

Table A.5 – Code rate R_{code}=1/3

Code number	ν	Generator polynomials	d_{fm}	d_f	w_{df}	A-gain,dB
12	2	5,7,7	8	8	3	4,26
13	2	5,6,7	8	7	1	3,68
14	3	13,15,17	10	10	6	5,23
15	3	11,15,17	10	9	1	4,77
16	3	10,15,17	10	8	3	4,26
17	4	25,33,37	12	12	12	6,02
18	5	47,53,75	13	13	1	6,37
19	5	47,55,75	13	13	4	6,37
20	5	45,55,75	13	12	3	6,42
21	6	133,145,175	15	15	11	6,99
22	6	127,155,165	15	13	3	6,37
23	7	255,331,367	16	16	1	7,27
24	8	557,663,711	18	18	10	7,78

Table A.6 – Code rate R_{code} = 1/2

Code number	ν	Generator polynomials	d_{fm}	d_f	w_{df}	A-gain, dB
25	2	5,7	5	5	1	3,98
26	3	15,17	6	6	2	4,77
27	3	13,15	6	6	4	4,77
28	4	23,35	8	7	4	5,44
29	4	31,33	8	7	4	5,44
30	4	25,37	8	6	2	4,77
31	5	53,75	8	8	2	6,02
32	5	61,73	8	8	6	6,02
33	5	43,75	8	8	6	6,02
34	5	45,73	8	8	5	6,02
35	5	71,73	8	8	10	6,02
36	6	133,171	10	10	36	6,99
37	6	135,163	10	10	46	6,99
38	7	247,371	10	11	2	6,99

Attachment B. Methodical manual for the course work

Topic of the course work **«Optimisation methods of error-control coding for transmission system»**

Introduction. In Chapter 12 it is shown, that error-control coding is effective means for the optimisation of transmission systems. In practice engineer-designer should solve **optimisation problems** on the basis of numerical calculations and corresponding comparison of a coding methods and a choice of concrete coding methods and corresponding to them the codes. The solution of such problem was underlay in basis of the course work.

Input data are set in the table of variants (table B.2):

1. The digital binary signal with rate R is subject to transfer.

2. The transmission channel is the channel with constant parameters and additive white noise.

3. Signal to noise ratio on a demodulator input is h_b^2.

4. Methods of modulation are BPSK or QPSK.

5. Mode of reception is coherent.

6. Pass band of transmission channel is F_{ch}.

7. Probability of an error on an output of transmission system no more p_{acc}.

8. Permissible complexity of a code trellis – no more C_{perm}.

It is necessary:

1. To choose and justify a choice of a error-control code for projected system, ensuring demanded bit error probability level p_{acc} under condition of a following *restrictions:*

1.1. The bandwidth of the modulated signal ΔF_s should not exceeds of a Pass band of transmission channel F_{ch} ($\Delta F_s < F_{ch}$).

1.2. By using of convolution codes the code trellis complexity should be no more magnitude C_{perm}.

2. To develop and give detailed exposition structural and function schemes of the encoder and the decoder for the chosen code and to justify their parameters.

3. To analyze of energy and frequency efficiency of a projected transmission system and to compare them to limiting values of efficiency.

4. To make a conclusion on the done work.

The content of the executed work

1. The introduction and input data.

2. Exposition of the block diagram of designed transmission system with indication of inclusion places of the error-control encoder, modulator, demodulator and the decoder with detailed explanations of functions fulfilled by them.

3. An application substantiation in the work of convolution codes.

4. A substantiation of a Viterbi algorithm choice for decoding of a convolutional code.

5. Calculation of a bandwidth occupied with a modulated signal ΔF_s at code rates 1/8, 1/4, 1/3, 1/2, 1.

6. Definition of an admissible code rate R^*_{code} by a condition 1.1 ($\Delta F_s < F_{ch}$).

7. Definition of the enumeration of codes with the rates that are not exceeding admissible rate R^*_{code}, which can be used for a task in view solution.

8. Choice codes from this enumeration ensuring given bit error probability level by the Condition 1.1 and restriction satisfying to the requirement on decoder complexity (by Condition 1.2).

9. Checking calculation of a bit error probability for the decoding of the chosen code.

10. Elaborating and exposition of a structural and function schemes of the encoder and the decoder chosen code.

11. A conclusion with summarising of the performed work.

12. The list of the used literature.

Methodical instructions

Calculation of a signal BPSK (QPSK) bandwidth should be made under recommendations from the Module 1. Using of an error-control codes with code rate R_{code} leads to increasing of a occupied frequency band of the signal $K_{\Delta F} = 1/R_{\text{code}}$ times (see Chapter 9). On the other hand, a control ability of a code increases with a decreasing of a code rate. Therefore the problem of a code parameters optimisation consists in a choice of a code with a rate at which the frequency band of the coded signal **does not exceed the given pass band of transmission channel** F_{ch}. If the demanded pass band for the transmission of a coded MPSK signal with rate R is equal to ΔF_{MPSK}, and the code rate is chosen equal to R_{code} the pass band of channel which is necessary for transmission a coded MPSK signal will be equal

$$\Delta F_{\text{s}} = \frac{\Delta F_{\text{MPSK}}}{R^*_{\text{code}}}.$$

Then from a inequality ($\Delta F_{\text{s}} < F_{\text{ch}}$) it is received a simple **condition for choice of code rate:**

$$R_{\text{code}} > R^*_{\text{code}}. \qquad \text{(B.1)}$$

The told is illustrated by figure B.1. The expanding of a frequency band of a coded signal is proportional to factor of a band expansion. By the process of a decreasing of a code rate (increasing of $K_{\Delta F}$) the frequency band extends and reaches values of a given channel frequency band F_{ch}. On the same figure the line A-gain = $f(R_{\text{code}})$ is shown. Intersection of a curve band with boundary given value F_{ch} defines the admissible factor of a channel frequency band expansion $K_{\Delta F} = 1/R_{\text{code}}$ and, accordingly, of code rate R^*_{code}.

The first stage of a choice of a error-control code is the choice of a class of codes (a class block or convolution codes). Using materials of Chapters 8 and 11 it is recommended to justify with deep arguments for a choice of convolution codes for using in project. Among decoding algorithms on a latitude of practical application the in the lead place occupies Viterbi algorithm. It is recommended to apply Viterbi algorithm in project. In section of the project with a substantiation of application of this algorithm it is necessary to give information about realisation complexity of an algorithm. Among the codes selected by criterion of a rate according to the formula

(B.1) there can be codes with various length of code constraint length (and, accordingly, with various decoder complexity). The noise immunity of decoding is characterised by A-gain. In code tables of a values A-gain are not reduced at certain level of error probability. At the same time, magnitude of A-gain is upper estimation of a real gain. Therefore at choice of a codes it is recommended to use A-gain which values are available from the Attachment A. Among the selected candidates of codes it is necessary to apply code ensuring maximum A-gain and meeting maximum requirements on code rate and minimum complexity of decoder.

Definitive data about error probability on decoder output it is necessary to get by the calculations the decoding bit error probability for chosen code from the signal to noise ratio. In case of representation failure to meet requirements it is recommended to apply a code with more value of A-gain.

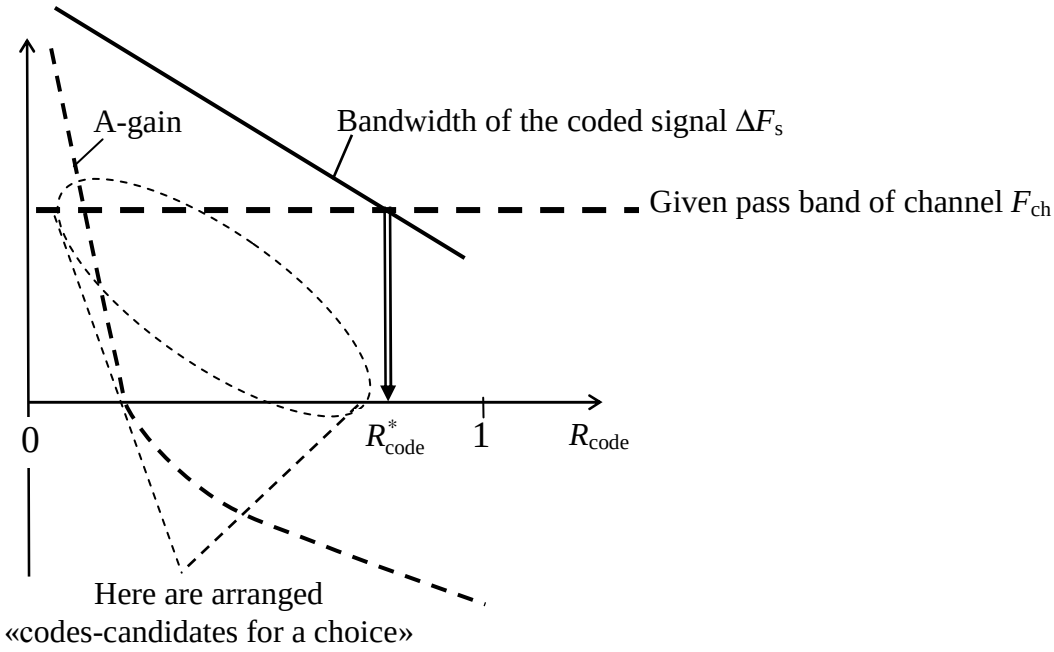

Figure B.1– Procedure of code optimization

Example of calculations and code optimisation procedure

Input data:

1 The rate of digital signal R = 64 kbit/s.

2 S/N ratio h_b^2 = 4,5 dB.

3 A modulation method is QPSK.

4 Mode of reception is coherent.

5 Pass band of transmission channel F_{ch} = 100 kHz.

6 Acceptable bit error probability $p_{acc} = 10^{-5}$.

7 Admissible code trellis complexity no more C_{perm} = 150.

Solution

1 Calculation of a necessary channel pass band for transmission with the method QPSK is made under formula $\Delta F_{QPSK} = [R(1 + \alpha)]/2$, where α is roll-off factor of spectrum . Being set by value α = 0,4, we receive

$$\Delta F_{\text{QPSK}} = [R(1 + \alpha)]/2 = [64\ (1+0{,}4)]/2 = 44{,}8 \text{ kHz}.$$

2 According to the formula (B.1) it is defined limiting value of code rate

$$R^{*}_{\text{code}} = \frac{\Delta F_{\text{s}}}{F_{\text{ch}}} = \frac{44{,}8}{100} = 0{,}448.$$

3 Under code tables we select the codes, satisfying to the requirement on a rate. Data about these codes are shown in table B.1. From the table it is visible, that for the solving given task can be used codes with the rate R_{code} = 1/2 which ensure enough big A-gain. In the table data the code with generator polynomials (133, 171) which at rate R_{code} = 0,5 ensures A-gain =6,99 dB is chosen for the project. Data of bit error probability calculation is given on figure 8 1 (signals BPSK and QPSK have the same noise immunity). It is visible, that the using of a such code ensures such performance: by the ratio signal/noise h_{b}^2 = 4,5 dB the bit error probability is less than 10^{-5}. Comparison with curves for uncoded QPSK shows that by $p = 10^{-5}$ this code ensures coding gain 6 dB.

Table B.1 – Performances for a code choice

Code rate R_{code}	Generator polynomials	Code length ν	Trellis complexity C	A-gain, dB
1/8	25,27,33,35, 37,25,33,37	4	32	6,02
1/8	115,127,131,135, 157,173,175,123	6	128	6,99
1/4	25,27,33,37	4	32	6,02
1/4	463,535,733,745	8	512	8,29
1/3	47,53,75	5	64	6,42
1/3	557,663,711	8	512	7,78
1/2	53,75	5	64	6,02
1/2	61,73	5	64	6,02
1/2	71,73	5	64	6,02
1/2	133,171	6	128	6,99
1/2	247,371	7	256	6,99

Table B.2 – Input data for the course work

Variant number for an elaborating of course work should correspond to the number of student surname in the academic group register						
Variant number	S/N ratio h_b^2, dB	Modulation method	Rate of signal R, kbit/s	Bandwidth of channel F_{ch}, kHz	Bit error probability p_{acc}	Trellis complexity C_{perm}
1	4,0	QPSK	64	80	10^{-6}	150
2	5,0	QPSK	16	25	10^{-4}	160
3	6,0	BPSK	256	800	10^{-5}	170
4	6,5	BPSK	64	200	10^{-6}	180
5	4,0	QPSK	16	25	10^{-4}	250
6	7,0	QPSK	128	200	10^{-5}	350
7	5,0	BPSK	2400	7000	10^{-8}	560
8	6,0	QPSK	32	50	10^{-6}	200
9	5,0	BPSK	24	70	10^{-4}	300
10	4,5	QPSK	256	400	10^{-5}	250
11	5,5	BPSK	300	1200	10^{-8}	550
12	4,0	QPSK	48	70	10^{-6}	150
13	4,0	QPSK	32	50	10^{-4}	250
14	5,0	BPSK	256	800	10^{-5}	300
15	4,0	QPSK	450	1300	10^{-9}	550
16	7,0	QPSK	56	90	10^{-6}	150
17	5,0	BPSK	24	70	10^{-4}	160
18	4,5	QPSK	256	400	10^{-5}	200
19	5,5	QPSK	500	1400	10^{-9}	550
20	6,0	BPSK	64	200	10^{-6}	150
21	7,5	QPSK	32	400	10^{-4}	250
23	6,5	QPSK	16	50	10^{-5}	150
24	6,0	QPSK	64	150	10^{-6}	150
25	4,5	BPSK	16	25	10^{-4}	200
26	5,0	BPSK	6000	16000	10^{-9}	550
27	6,0	QPSK	384	600	10^{-5}	250
28	4,5	QPSK	56	100	10^{-6}	150
29	5,0	BPSK	16	50	10^{-5}	250
30	5,5	BPSK	5500	32000	10^{-9}	560
31	4,5	QPSK	64	200	10^{-5}	150
32	5,0	QPSK	64	300	10^{-5}	250

Attachment C. Education manual for laboratory works

LW 4.1 Studying of block error-control Hamming code codecs structure

1 Objectives

1.1 Studying of Hamming systematic code (7, 4) codec structure.

1.2 Research of the code (7, 4) control ability.

2 Main principles

The systematic code is error-control code, which code word contain k information bits and $r = n - k$ checking symbols (checking symbols are linear combination of information bits). Systematic codes are denoted as (n, k) or (n, k, $d_{\min}$). In this work code (7, 4) or (7, 4, 3) is studied.

Error-control codes with code distance $d_{\min} = 3$, allowing to correct first order errors at decoding, name as Hamming codes [5, p. 149]. We will determine connection between error-control code parameters n and k. It is known that for any natural number r the Hamming code of lengths $n = 2^r - 1$ or $k + r = 2^r - 1$ exists [5, p. 149]. These equalities can be used and as inequalities $k \le 2^r - r - 1$. The last expression allows choosing n and r at given k.

The matrix method of linear block codes coding and decoding processes description is most useful (see Sections 5, 6). So, coding by systematic code (n, k) consist in addition to code words checking symbols and can be described by matrix equality

$$\mathbf{A}\cdot\mathbf{G} = \mathbf{B}, \qquad (1)$$

where $\mathbf{A} = (b_1\ b_2\ \dots\ b_k)$ is row matrix size k, correspond to information code word;

$\mathbf{B} = (b_1\ b_2\ \dots\ b_{\mathrm{k}}\ b_{\mathrm{k+1}}\ \dots\ b_{\mathrm{n}})$ is row matrix size n, correspond to error-control code word.

G – generator matrix $k \times n$, the elements of which g_{ij} take on values 1 or 0.

The **G** matrix rows must satisfy next conditions [6, p. 86…88].

1 Distance between any two rows must not be less $d_{\min}$.

2 Every row must contain no less then $d_{\min}$ units.

3 All rows must be linearly independent, i.e. none of rows can be got by adding (XOR) of some other rows.

For example, for a code (7, 4) generator matrix looks like formulas (4.4), (4.7).

The coder operation algorithm:

1 k information bits in a parallel code or in a series code (in last case a shift register is needed) on the coder input.

2 The checking symbols $r = n - k$ by adders with r calculates.

3 k information bits and r checking symbols in a parallel or series code (in last case the converter of parallel in series code is needed) on the coder output.

On the figure 5.1 the code (7, 4) with generator matrix (4.4) encoder functional diagram is resulted. Input and output code words are represented in a parallel code.

The process of decoding includes the syndrome calculation. In matrix form is written down as :

$$\mathbf{S} = \mathbf{H}\cdot\hat{\mathbf{B}}, \qquad (2)$$

where **H** is check matrix $r \times n$.

$\hat{\mathbf{B}}$ is matrix-column, size n, correspond to the code word on the decoder input;

S is syndrome, matrix-column, size r.

The decoder algorithm is following:

1 n symbols of the code word come on the decoder input.

2 Using (6) a syndrome calculates.

3 Syndromes analyzer, built on the syndromes table basis, forms signals for error symbols correction.

4 Error symbols correction consist in their inverting: XOR for error symbol and unit (lets $\hat{b}_i$ is some symbol, then $\hat{a}_i = \hat{b}_i \oplus 1$).

5 After correction of error the information code word of k symbols come on the decoder output.

On the figure 5.2 the code (7, 4) decoder functional diagram is resulted. Input and output code words are represented in a parallel code.

A code, encoder and decoder of which, is built on multiinput adders (with XOR operation), name as Hamming code or even parity check code. Using the indicated principles it is possible to build encoder and decoder for different values of n and k.

3 Questions

3.1 Give the definition of error-control codes.

3.2 What is code distance?

3.3 Write down expressions for determining of code control ability with given code distance.

3.4 Give the definition of systematic error-control codes.

3.5 How to define the checking symbol number at set number of k, if $d_{\min} = 3$?

3.6 What is generator matrix?

3.7 How to build a generator matrix for a systematic code?

3.8 What is check matrix?

3.9 What is syndrome?

3.10 Explain principle of Hamming code decoder construction.

4 Home task

4.1 Study main principles.

4.2 A code (7, 4) is set by a generator matrix

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

Write down the number of brigade N in the binary number. Considering these four digits as information word on the encoder input, calculate code word on the coder output. Form the given code check matrix and code syndromes table. Put first order error into symbol b_N of formed code word; calculate a syndrome for the code word on the decoder input. Make sure, that a syndrome corresponds to error symbol $\hat{b}_N$.

4.3 Prepare for discussion on key questions.

5 Laboratory task

5.1 Start the program on a computer and using mouse cursor study the method of input data setting.

5.2 Enter the information bits got in the home task, after coding implementation, make sure in correct results of home work.

5.3 Put an error into b_1, b_2, b_3, b_4, b_5, b_6, b_7 symbols by turns, make sure in correctness of decoding and syndromes table which were calculated in the home task.

5.4 Put a double error into arbitrary two symbols, make sure, that a decoder tries to correct errors in accordance with a syndrome and puts the third error. Repeat the experiment for two-three other double errors.

5.5 Put a triple error into symbols b_1, b_2, b_3. Make sure, that a syndrome is equal to the zero – it shows that $d_{\min} = 3$. Repeat the experiment putting a triple error into symbols b_1, b_4, b_5. On the basis of check matrix you will define, what another triple errors result in the permitted words.

6 Description of laboratory model

A laboratory model is executes on the personal computer. A code (7, 4) is described by generator matrix

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

The control by model is produced by the left mouse button. The putting of errors produces by setting in «1» digit position (positions) of errors block e_1, e_2, e_3, e_4, e_5, e_6, e_7 (which must contain an error). A model produces conversion of code word on the encoder output to code word on the decoder input by the rule $\hat{a}_i = \hat{b}_i \oplus e_i$, for $i = 1, 2, \ldots, 7$.

7 Requirements to the report

7.1 Title and objectives of laboratory work.

7.2 Block diagrams of coder and decoder of the used code (7, 4).

7.3 Results of homework performing.

7.4 Results of implementation of laboratory task (tables and graphs).

7.5 Conclusions on every item of laboratory task, with analysis of the got results.

LW 4.2 Cyclic codes coding and decoding studying

1 Objectives

1.1 Error-control code coding principles studying.

1.2 Experimental research of cyclic code encoder and decoder operation principles.

2 Main principles

2.1 One of methods of digital signal transmission by communication channel with errors quality increasing is error-control codes using. Error-control codes allow to detect or to correct errors which appear in a communication channel. In this work binary block error-control codes study.

2.2 General principle of error-control codes construction easy enough. From the possible code words of length n number $M = 2^n$ not all are used for the transmission, only $M_0 = 2^k$ ($M_0 < M$). The used code words are named permitted. Other $M - M_0$ code words is considered forbidden, they can not appear at the communication channel input, their appearance on the channel output indicates about errors. Thus, due to the forbidden code words presence the possibility of error detection appears. So, any error-control code is a code with redundancy ($r = n - k$ redundant (checking) symbols in every code word transmits by communication channel).

2.3 For error-control codes description next parameters are used.

Hamming distance d_{ij} shows the order of difference between i-th and j-th code words. For any two binary code words distance equals the noncoincident symbols number in them.

Code distance $d_{\min}$ is the minimum Hamming distance for the given code. Enumerating all possible pair of the permitted code words and calculating distances d_{ij} for them, it is necessary to find minimal among them, $d_{\min} = \min d_{ij}$.

Code rate R shows the relative number of information symbols k in code words of length n and R is calculated as $R = k/n$.

Code control ability is determined by order of detected errors q_{det}, and order of corrected errors q_{corr}.

The order of detected errors q_{det} is number of errors in code word, which are assured detected at decoding, – it is determined as: $q_{\text{det}} < d_{\min}$.

The order of corrected errors q_{corr} is the number of errors in code word, which are corrected at decoding, – it is determined: $q_{\text{corr}} < d\text{min}/2$.

2.4 At the error-control coding use in the communication channel structure error-control code encoder and decoder is included, it is resulted on the figure 1.

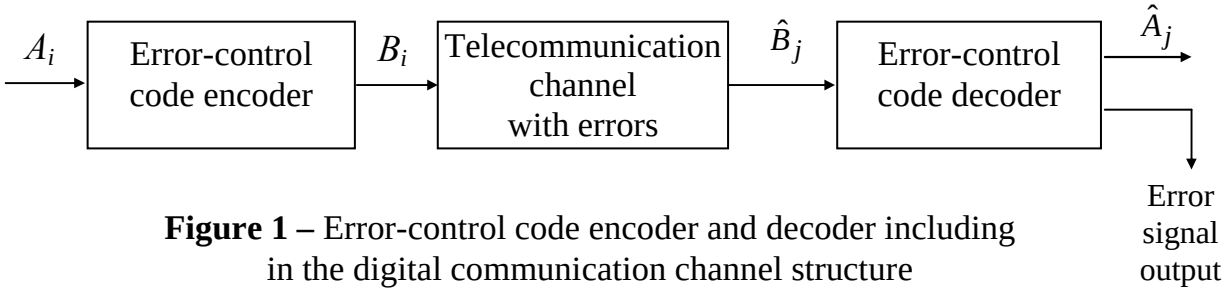

Figure 1 – Error-control code encoder and decoder including in the digital communication channel structure

Encoder and decoder destination consists in the following. Code word $\mathbf{A}_i$, length k on the encoder input, encoder will transform it in error-control code word $\mathbf{B}_i$, length n in accordance with the coding rule, and $n > k$. Code word, length n from a communication channel on the decoder input:

$$\hat{\mathbf{B}}_j = \mathbf{B}_i \oplus \mathbf{E}, \quad (1)$$

where **E** is error word. For example, $\mathbf{B}_i$ = 101000; let an error appear in the second and third symbols, then **E** = 011000, and $\hat{\mathbf{B}}_j$ = 110000.

Depending on code control ability and purpose of its application the error-control code decoder can work in the detection mode or in the error correction mode. In the error detection mode a decoder analyses: is word $\hat{\mathbf{B}}_j$ permitted or forbidden? If word is permitted, a decoder in accordance with the decoding rule forms on the output information code word $\mathbf{A}_j$, length *k*. If word $\hat{\mathbf{B}}_j$ is forbidden, a decoder does not decode it, on the decoder output no any code word, and a certain signal appears on the error signal output (figure 1) (for example, "1"). In the error correction mode a decoder instead of the forbidden code word decodes the permitted code word the nearest to it in accordance with the decoding rule and gives out information code word length *k*.

2.5 In transmission systems systematic codes the most distributed, code word of systematic code contain *k* information symbols (symbols on the encoder input) and $r = n - k$ checking symbols formed by encoder from information code word. In the case of linear codes checking symbols are linear combinations of information code word.

Among systematic block codes wide distribution was got by cyclic codes, due to easy encoder and decoder construction. For cyclic codes description by polynomials is most useful - for example, the code word $\mathbf{A}_i$ =10111 corresponds to polynomial $a_i(x) = x^4 + x^2 + x + 1$ (code word symbols are coefficients at the proper dummy variable *x* order of, thus to symbol 1, which is written down first, the most high order x in a polynomial corresponds).

Any cyclic code is set by not only the numbers of *n* and *k* but also generate polynomial *g*(*x*) order *r*. A cyclic (*n*, *k*) code names a code, all code words of which appear by the polynomials order of *n* - 1 and less, which are divided without remainder on generate polynomial. In table 1 generate polynomials are resulted for *r* = 3, 4 and 5.

Table 1 – Generate polynomials

r	*g*(*x*)
3	$x^3 + x^2 + 1$
	$x^3 + x + 1$
4	$x^4 + x^3 + 1$
	$x^4 + x + 1$
5	$x^5 + x^4 + x^2 + 1$

Cyclic code (*n*, *k*) encoder operation method consists in the following. Lets *a*(*x*) is polynomial which corresponds to information code word on the encoder input. Polynomial $a(x)\cdot x^r$ corresponds to addition *r* zeros to information code word on the right. The polynomial $a(x)\cdot x^r$ on the generate polynomial *g*(*x*) dividing with the purpose of determination of remainder from the dividing *r*(*x*) is executed. Remainder from the dividing r(x) is the checking symbols. Polynomial, corresponding to code word on encoder output, it is determined:

$$b(x) = a(x)\cdot x^r + r(x), \quad (2)$$

the *r* zeros, is change by combination proper to the remainder from the dividing.

It is easy to show that the polynomial $b(x)$ is divided without a remainder on the polynomial of g(x):

$$\frac{b(x)}{g(x)} = \frac{a(x)x^r}{g(x)} + \frac{r(x)}{g(x)} = p(x) + \frac{r(x) \oplus r(x)}{g(x)} = p(x),$$

where $p(x)$ is integer part from the division $a(x) \cdot x^r / g(x)$.

You should remember that addition of polynomials is executed as XOR operation of coefficients at the equal orders of x.

We will consider the example of code (10, 5) code word forming with the generate polynomial $g(x) = x^5 + x^4 + x^2 + 1$ Lets $\mathbf{A}_i = 10110$, then $a_i(x) = x^4 + x^2 + x$, and $a_i(x) \cdot x^5 = x^9 + x^7 + x^6$. We will execute the division with the purpose of remainder determination.

$$\begin{array}{l} x^9 + x^7 + x^6 \quad | \; x^5 + x^4 + x^2 + 1 \\ \oplus \underline{x^9 + x^8 + x^6 + x^4} \quad x^4 + x^3 + 1 \\ \quad x^8 + x^7 + x^4 \\ \quad \oplus \underline{x^8 + x^7 + x^5 + x^3} \\ \quad\quad x^5 + x^4 + x^3 \\ \quad\quad \oplus \underline{x^5 + x^4 + x^2 + 1} \\ \quad\quad\quad x^3 + x^2 + 1 = r(x) \end{array}$$

Acording to (2) $b_i(x) = x^9 + x^7 + x^6 + x^3 + x^2 + 1$ or $\mathbf{B}_i = 1011001101$.

In the cyclic code decoder the code word on decoder input division on generate polynomial is produced. Polynomials of the code word on encoder output $b(x)$, code word on decoder input $\hat{b}(x)$ and errors $e(x)$ connected by expression like (1): $\hat{b}(x) = b(x) \oplus e(x)$.The result of division on generate polynomial can be represented

$$\frac{\hat{b}(x)}{g(x)} = \frac{b(x)}{g(x)} \oplus \frac{e(x)}{g(x)} = p(x) \oplus \frac{e(x)}{g(x)} = p(x) \oplus v(x) \oplus \frac{s(x)}{g(x)},$$

From these expression follows that remainder $s(x)$ depends only on the error polynomial and does not depend on the code word on encoder output ($v(x)$ is integer part from the division $e(x)$ on $b(x)$). Remainder from the division $s(x)$ on $g(x)$ is a syndrome. A nonzero remainder indicates that the accepted code word is forbidden (with errors). If a decoder works in the error correction mode, the error symbol number (or numbers of error symbols) is determined on the syndrome analysis base. For code (10, 5) with the generate polynomial $g(x) = x^5 + x^4 + x^2 + 1$ we will make the syndromes table for all single errors, executing the divisions $e(x)$ on $g(x)$ and writing down remainders from the division in table 2.

Table 2 – Syndromes for first order errors

Error polynomial $e(x)$	Syndrome $s(x)$
x^9	$x^4 + x^3 + 1$
x^8	$x^4 + x^2 + x$
x^7	$x^3 + x + 1$
x^6	$x^4 + x^3 + x^2 + x + 1$
x^5	$x^4 + x^2 + 1$
x^4	x^4
x^3	x^3
x^2	x^2
x	x
1	1

From table 2 follows that in the case of single errors (first order errors) all syndromes are different, therefore every syndrome simply specifies the error symbol number. Error correction by a decoder is executes by decipherer built according to table 2, and inverting element which executes the error symbol inversion.

The researched code (10, 5) has code distance $d_{\min} = 4$ and allows to correct first order errors.

Questions

3.1 What codes are named as error-control codes?

3.2 Explain error-control code encoder and decoder destination.

3.3 What is redundancy and code rate?

3.4 What is the Hamming distance between code words, code distance, and error order?

3.5 How to explain detecting and control code ability?

3.6 Explain error detecting and error correction general principles.

3.7 What codes are named cyclic?

3.8 How to write code word as polynomials?

3.9 Explain cyclic codes encoding and decoding principles.

4 Home task

4.1 Study "Error-control codes" from the compendium of lectures and literature [5, p. 137...150; 2, p. 287...297].

4.2 Write down the number of brigade $N + 8$ in the binary number. Considering these number as information word, length $k = 5$ on the encoder input, form cyclic code (10, 5) code word, using generate polynomial $g(x) = x^5 + x^4 + x^2 + 1$.

4.3 For three errors words $e_1(x)$, $e_2(x)$ and $e_3(x)$, given in table 3, calculate syndromes, and then, using table 2, define decoder work result. If the calculated syndrome is wrote in table 2, a decoder inverts code word symbol which is considered as error. If the calculated syndrome is absent in table 2, a decoder does not change code word symbols which are decoded. Code word on the output of decoder is code word on decoder input first k symbols.

Table 3 – Polynomials of errors for the home task

Number of work place N	$e_1(x)$ – single error	$e_2(x)$ – doubled error	$e_3(x)$ – triple error
1, 11	x^9	$x^9 + 1$	$x^7 + x^6 + 1$
2, 12	x^8	$x^6 + x^8$	$x^9 + x^8 + x^4$
3	x^7	$x^6 + x$	$x^9 + x^8 + x^2$
4	x^6	$x^9 + x$	$x^7 + x^6 + x$
5	x^5	$x^7 + x^6$	$x^8 + x^7 + x^2$
6	x^4	$x^7 + x^4$	$x^8 + x^7 + x^3$
7	x^3	$x^9 + x^2$	$x^8 + x^7 + x$
8	x^2	$x^6 + x^3$	$x^8 + x^6 + x$
9	x	$x^7 + x^2$	$x^9 + x^6 + x^4$
10	1	$x^8 + x$	$x^8 + x^6 + x^4$

4.4 Draw a shame error-control code encoder and decoder including in the digital communication channel structure.

4.5 Be ready to discuss key questions.

5 Laboratory task

5.1 *Acquaintance with a virtual model*. Start the program 4.2, using the icon TT (English) on a desktop. It is necessary to study the structure of a virtual model using its description in part 6 of this LW. Coordinate the plan of performance of the laboratory task the teacher.

5.2 *Encoding process research.* The cyclic code (10, 5) from home task is researched. For this purpose:

- choose in a menu "That do we research?" point "Encoding";
- choose a code (10, 5) and set a proper generate polynomial;
- enter information code word determined in home task.

Run the program and compare the code word on the encoder output with calculated in the home task.

5.3 *Transmission by a communication channel process research.* For this purpose:

- choose in a menu "That do we research?" point "Transmission by a channel";
- enter zero error word, length n.

Run the program at the same settings (from item 5.2). Make sure, that in the case of zero error code word on the output "Telecommunication channel" coincides with code word on the input.

Set error word, which corresponds to the single error $e_1(x)$ from Table 2 for your variant. Run the program, compare code words on the telecommunication channel input and output and make sure in the telecommunication channel with errors correct work.

5.4 *Research of process of decoding.* For this purpose:

- choose in a menu "That do we research?" point "Decoding";
- enter zero error word.

Run the program at the same settings. Make sure, that a syndrome is zeroes, and code word on the decoder output coincides with code word on the encoder input.

Set errors word which corresponds to the single error $e_1(x)$ from the home task. Run the program at the same settings in encoder. Make a table with the decoding results according to the sample Tabl. 4.

Table 4 – Cyclic code (10, 5) for the code word $N + 8 = 22$ (encoder input – 10110, encoder output - 1011001101) researches results

Errors word $e(x)$	Encoder input $\hat{b}(x)$	Encoder output $\hat{a}(x)$	Syndrome $s(x)$	Error symbol number, which decoder define
x^9	0011001101	10110	$x^4 + x^3 + 1$	x^9
…	…	…	…	…

Repeat research of decoding at double $e_2(x)$ and triple $e_3(x)$ errors from the home task. Compare got results with calculations in the home task results.

Repeat research of decoding at the arbitrary fourfold error $e_4(x)$. For certain error words, for example, $e_4(x) = x^6 + x^5 + x^3 + x$ syndrome is zeroes, that confirms that a code (10, 5) has code distance $d_{\min} = 4$.

5.5 Other cyclic code (n, k) research.

Teacher gives a cyclic code (n, k). Repeat research item 5.4.

6 Description of laboratory model

Laboratory work is executed on a computer in the HP VEE environment with using a virtual model, the block diagram of which is resulted on the figure 2. A model

is intended for cyclic codes: (7, 4), (10, 5), (10, 6), (11, 7), (12, 8), (13, 9), (14, 10), (15, 11), encoding and decoding processes research.

A model allows research consistently encoding, transmission by a telecommunication channel and decoding. It contains the marked by a red color options which a student must set. The blue color windows are used for the model work results indication.

Coder forms the permitted code word by the $r = n - k$ checking symbols calculation and add to them information symbols. The got code word is indicated on the encoder output.

For the transmission by a telecommunication channel implementation it is necessary to enter errors word, length n, consisting of 1 and 0. The symbol 1 is set in those positions, which an error at the transmission by a telecommunication channel must be in. In a telecommunication channel code word from encoder is added with combination of errors by XOR. The got code word is indicated on the telecommunication channel output.

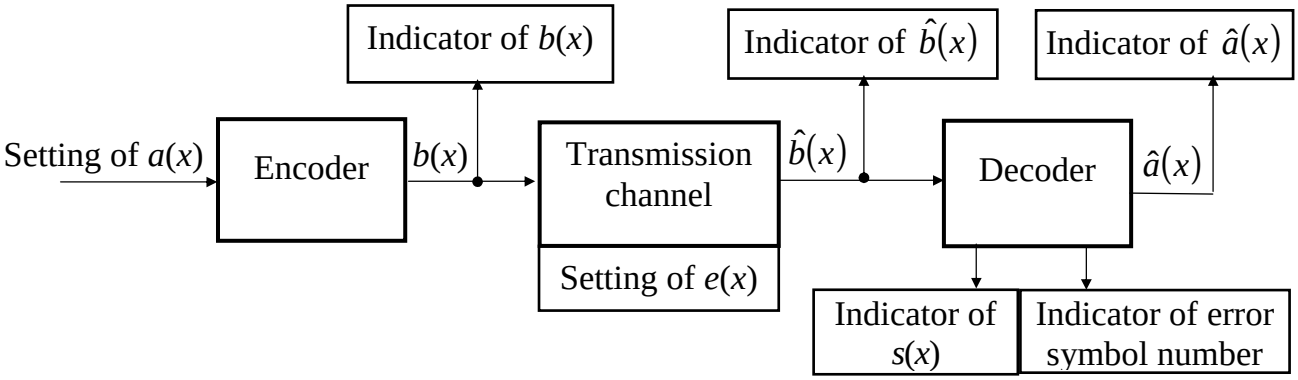

Figure 2 – Virtual model block diagram

A decoder divides code word on his input, on a generate polynomial. A window under a decoder shows the code word syndrome (in binary presentation); a window left shows the decoder decision about the error symbol number (all codes researched in a model allow to correct single errors).

If a syndrome is zero, a decoder gives message "No errors". If a syndrome is not zero, on the syndromes table a decoder determines the error symbol number and "Error in x^p " message, where p is error symbol number.

The syndromes table contains only n syndromes which correspond to n symbols of the decoded code word. The possible syndromes number is equal 2^{n-k}. If $n < 2^{n-k} - 1$, a syndrome which is absent in the syndromes table can appear in a decoder. In this case a decoder gives out the message "Unknown error".

If a decoder defined the error symbol number it gives out the message "Error in x^p", it corrects this symbol and takes away $n - k$ last symbols. If syndrome is zero or it is absent in the syndromes table, a decoder only takes away $n - k$ last symbols. The information code word appear on the decoder output.

7 Requirements to the report

7.1 Title and objectives of laboratory work.

7.2 Results of the homework execution

7.3 Results of execution of items 5.2...5.5 of laboratory task.

7.4 Conclusions on every item of the laboratory task, with analysis of the got results (coincidence of experimental and theoretical information).

7.5 Signature of student about the laboratory work execution, teacher's signature for the laboratory work defend with estimation and date.

LW 4.3 Noise immunity of block error-control codes researching

1 Objectives

1.1 Study of block diagram of (n, k) block error-control codes decoder with the errors correction.

1.2 Experimental researches of noise immunity of (n, k) block error-control codes with the errors correction.

1.3 Calculations of coding gain (CG) from data of experimental researches of noise immunity of (n, k) block error-control codes.

2 Main positions

2.1 **Correcting ability of error-control codes.** Correcting ability of error-control code is expressed by the guaranteed correctable errors value $q_{\text{cor}} \le (d_{\min} - 1)/2$ and by the guaranteed detectable errors value $q_{\text{det}} \le d_{\min} - 1$, where $d_{\min}$ – code distance.

Noise immunity of transmission system with a error-control code is described by the $p_d = f(E_b/N_0)$, where p_d is bit error probability at the output of the decoder, E_b/N_0 is the ratio of signal energy expended on the transfer of one bit to the power density of noise (SNR) at the input of the demodulator.

Frequently the noise immunity of the transmission system with a error-control code it is convenient to describe by the gain coding (CG) value (see below).

2.2 **Syndromes decoding.** For (n, k) block error-control codes today the syndromes decoding is basic for a discovery and errors correction.

Syndromes method of errors correction is based on a simple rule: on the syndrome of code word the errors location is determined. Therefore under the syndrome of code word understand the result of decoder calculation on the set rules of number $\mathbf{s} = (s_1, s_2, \ldots, s_r)$, $r = n - k$, which testifies to the errors detected and determines their placing (configuration) in code word. In binary codes a syndrome is written down in the binary number system, that its digits $s_1, s_2, \ldots, s_r$ take on a value 0 and 1.

A zero syndrome specifies on that the received code word is permitted, that the detected errors are not present. Nonzero syndrome is corresponding to some errors configurations. Thus possible such situations:

– the syndrome received by calculation corresponds to some configuration of errors which meets at $q \le q_{\text{cor}}$, in this situation the decoder "corrects" errors, but correction not true, and, most likely, the quantity of erroneous symbols in a code word grows;

– the syndrome received by calculation does not correspond to any of configurations of errors which can be at $q \le q_{\text{cor}}$, in this situation the decoder "refuses" decoding and passes such code word on an output with errors.

Thus, the syndromes decoding of binary codes is taken to the calculation of syndrome and search that or by other method errors configurations which errors are corrected on. Correction of an error in some binary symbol is reduced to its inversion and carried out by addition of an erroneous symbol with 1 on the module 2.The generalized block diagram of syndromes decoder is resulted in figure 1.

In the scheme of figure 1 “Syndrome analyzer” is the most complex device. In it on the calculated syndrome errors configuration, on which then the errors corrected with corrector, is set. As a syndrome can be calculated only after reception of all code word (block), for the errors correction in a code word needed to delay on *n* symbols, which is carried out *n*-digit shift register. After establishment of errors configuration an analyzer symbols of the decoding word are consistently "pushed" from a shift register, and from an analyzer symbols "1" act in those moments of time, when it is necessary to invert erroneous symbols.

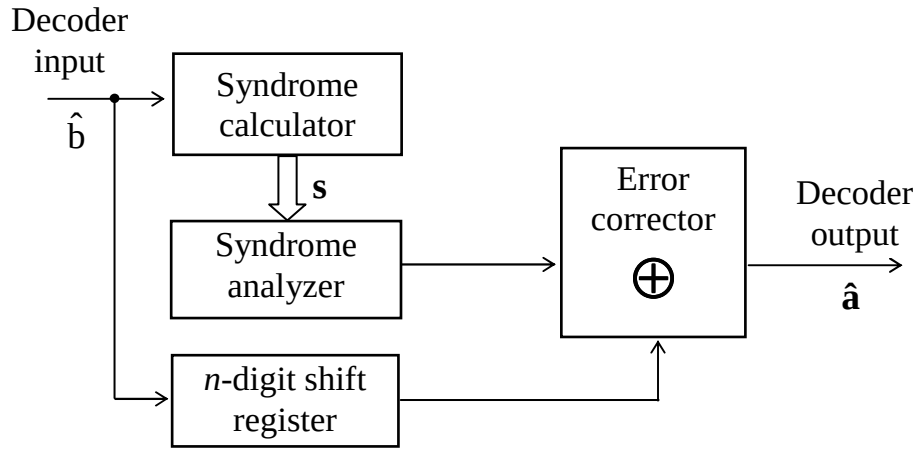

Figure 1 – (*n*, *k*) block code syndromes decoder generalized block-diagram

2.3 Coding gain

Coding gain (CG) shows how many decibels lower required SNR at the input of the demodulator in a transmission system with a error-control code, rather than in a transmission system without error-control code for a given value p_d.

Let SNR (dB) at the input of the demodulator in a transmission system without error-control code h_{b1}^2, and in a transmission system with error-control code h_{b2}^2 with the same symbol error probability. Then, if the signal to noise ratio expressed in decibels,

$$\text{CG} = h_{b1}^2 - h_{b2}^2 . \quad (1)$$

We remind that a ratio h_{b2}^2 is determined as

$$h_{b2}^2 = \frac{E_b}{N_0} = \frac{P_s T_b}{N_0}, \quad (2)$$

where P_s/N_0 is ratio of average power of signal to noise power density at the demodulator input; T_b is duration of information binary symbol (bit) at the transmission system input.

CG can be calculated in theory or measured experimentally. For example (figure 2), in a transmission system is required to ensure the symbol error probability $p = 10^{-5}$. In a transmission system without error-control code SNR $h^2_{b1} = 9{,}6$ dB is required, and in a transmission system with error-control code SNR $h^2_{b2} = 6{,}3$ dB is required, then CG = 9,6 – 6,3 = 3,3 dB.

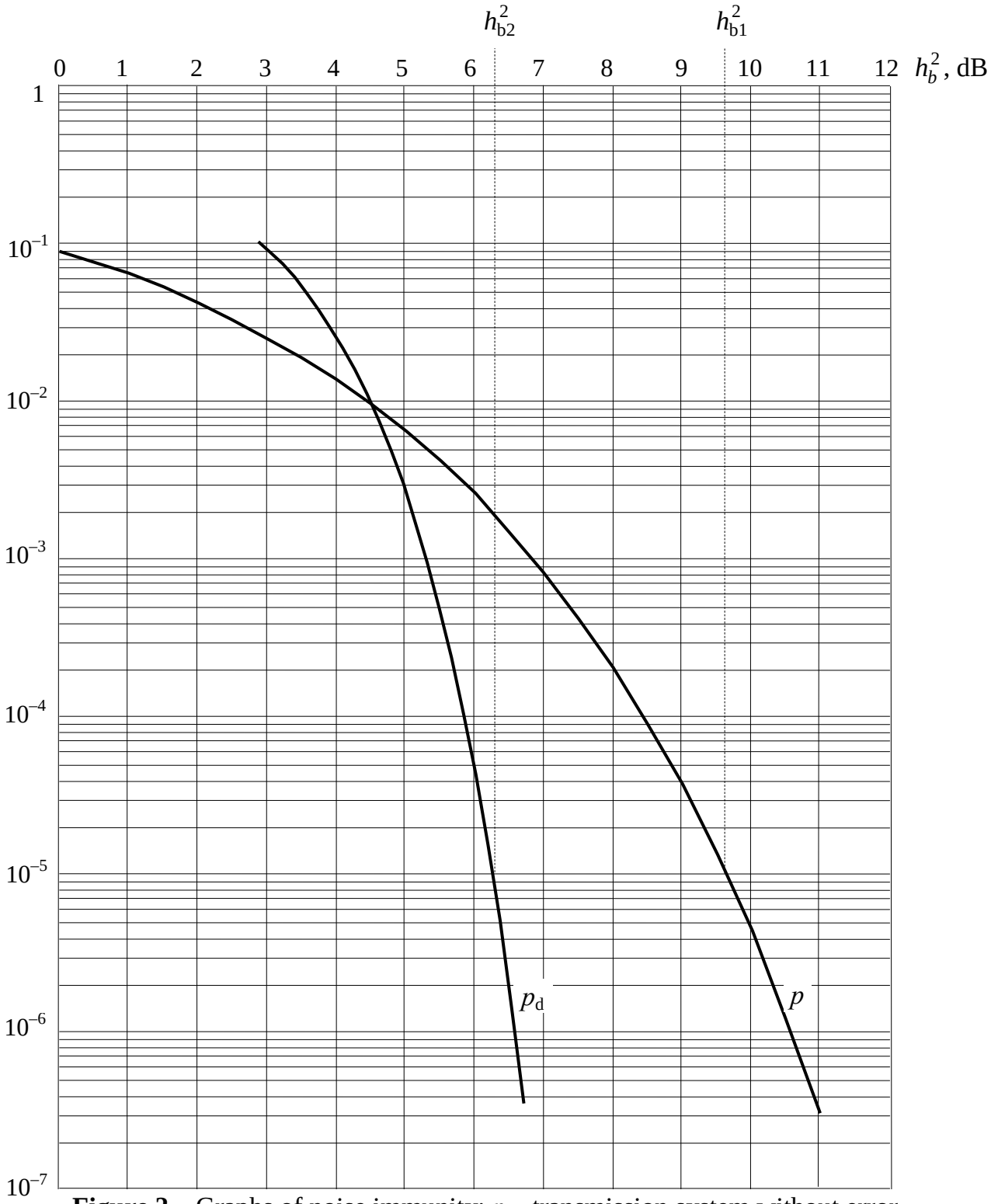

Figure 2 – Graphs of noise immunity: p – transmission system without error-control code; p_d – transmission system with error-control code

If CG > 0 dB (on figure 2 at $p_d < 10^{-2}$), such error-control code allows to decrease the signal/noise ratio at the demodulator input, if CG ≤ 0 dB, such error-control code does not allow to decrease the signal/noise ratio at the demodulator input, its application is worsened by quality of reception.

Dependencies $p = f(h_b^2)$ and $p_d = f(h_b^2)$ can be obtained experimentally by the method described in the laboratory work 3.3. In this laboratory we study noise immunity of transmission system with error-control code using a model of digital communication channel with errors (figure 3).

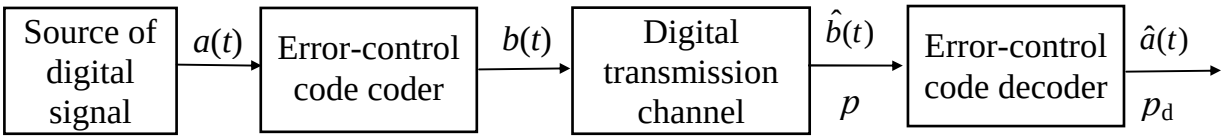

Figure 3 – Experimental studies of noise immunity transmission system with error-control code

The order of experimental determination of the CG when using the model of digital communication channel with errors.

1. A series of values of error probability at the channel output p in a certain range of values, such as, $p = 10^{-1}$–10^{-3} is setting (model of the digital communication channel with errors should allow such settings). For each value of p the error probability at the output of the decoder is determined p_d.

2. Must be specified modulation type which is used in digital communication channel. For transmission system without error-control code and with this type of modulation $p = f(h_b^2)$ is plotted. Figure 2 this dependence is built for BPSK and QPSK signals.

3. Graphing $p_d = f(h_b^2)$ is produced in the following order. For one of the values of p on plot $p = f(h_b^2)$ determine the h_b^2 value, which for the transmission system with a error-control code will be equal to the ratio signal/noise

$$h^2 = \frac{E_s}{N_0} = \frac{P_s T_s}{N_0}, \qquad (3)$$

where T_s is duration of the binary symbols at the modulator input in a transmission system with error-control codes (the output of the error-control code encoder). As $T_s = T_b k/n$, so in a transmission system with an error-control code $h_b^2 = h^2 n/k$. In the presentation in decibels $h_b^2 = h^2 + 10\lg(n/k)$. This conversion takes into account the fact that at the error-control code encoding in nonredundant code word $n - k$ checking symbols are introduced which lead to a decrease in a duration of all symbols in the code word and a corresponding decrease in signal energy.

In the resulting value h_b^2 and the corresponding value p_d point depending is set $p_d = f(h_b^2)$. The same procedure is repeated for all other values of p. Points are connected by a smooth curve and get dependence $p_d = f(h_b^2)$.

4. At a given level of error probability determine the values h_{b1}^2 and h_{b2}^2. CG is calculated by the formula (1).

Example calculation of points on the graph $p_d = f(h_b^2)$. (31, 26) code is used; BPSK modulation; error probability $p = 5 \cdot 10^{-3}$ and $p_d = 10^{-5}$. On the value of p and the graph in figure 2 $h^2 = 5{,}5$ dB is defined. Calculate

$$h_b^2 = 5{,}5 + 10\lg(31/26) = 6{,}3 \text{ dB}.$$

Lay off the point schedule $p_d = f(h_b^2)$, whose coordinates is $(6{,}3;\ 10^{-5})$.

3 Questions

3.1 Specify the methods of error-control codes decoding which are known by you.

3.2 What is syndrome of code word and what purpose is it used for?

3.3 What (n, k) block error-control codes the syndrome decoding is mainly used for?

3.4 What function is executed by the analyzer of syndrome in the decoder of error-control code?

3.5 How errors are corrected in code words of error-control code?

3.6 How is the syndrome of code words calculated for cyclic codes?

3.7 That does determine CG of error-control code?

3.8 What parameters of error-control code does CG depend on?

4 Home task

4.1 In the workbook, prepare a graph $p = f(h_b^2)$ for BPSK and QPSK signals (redraw from figure 2 or calculate).

4.2 Use the graph to calculate the CG, provided by error-control block (n, k) code, at a given probability p_d according to table 1.

4.3 Prepare to the discussion of questions.

Table 1 – Error-control code parameters for home task calculations

Brigade number	Error probability at the decoder output p_{out}	Error probability at the decoder input p_{in}	Code rate $R_{code} = k/n$
1	10^{-4}	$5 \cdot 10^{-2}$	0,5
2	10^{-5}	$2 \cdot 10^{-2}$	0,6
3	10^{-4}	10^{-2}	0,7
4	10^{-5}	$5 \cdot 10^{-3}$	0,8
5	10^{-4}	$2 \cdot 10^{-3}$	0,9

5 Laboratory task

5.1 *Acquaintance with the computer model of decoder*. For this purpose to start the program, using an icon “Laboratory works” on a desktop, and then folder of TT-2. To master a conducting method research of correcting ability of (n, k) block code, that by introduction of basic data, start of the program, reading of results. To bring the chart of researches to the LR report.

Table 2 – Parameters entered in the study

Denotation	Comments
n	Code word length (bit)
k	Number of information bits
$d_{\min}$	Code distance
$g(x)$	Generate polynomial
prob_err	p – decoder input error probability
num_rand	N – number of code words

At each new dimension prob_err and num_rand must be changed.

5.2 *Experimental research of correcting ability of decoder*. The experimentally determined values p and p_d and recorded in the table 4. Research is conducted for codes (n, k):

- brigade №1: (31, 26), (23, 12) codes;
- brigade №2: (30, 25), (22, 11) codes;
- brigade №3: (29, 24), (21, 10) codes;
- brigade №4: (28, 23), (20, 9) codes;
- brigade №5: (27, 22), (19, 8) codes.

On the teacher task can be researched other (n, k) codes for $n \le 32$.

Generator polynomials for researches get out from table 3.

Table 3 – Generator polynomials of cyclic codes (n, k)

Order $n - k$	Generator polynomials
4	$x^4 + x^3 + 1$; $x^4 + x + 1$
5	$x^5 + x^2 + 1$; $x^5 + x^3 + 1$;
11	$x^{11} + x^{10} + x^6 + x^5 + x^4 + x^2 + 1$; $x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$
Example of polynomial input: x^4+x^3+1	

During research to set error probability at the decoder input $p < 0{,}01$ on diminishing (5–6 points) until error probability at the decoder output p_d will not attain a value, near by 10^{-5}.

Table 4 – The results of measurements (specify code)

Measu-ring number	Code words number N	**Decoder input**		**Decoder output**		
		$N_{\text{er. in}}$	p	$N_{\text{er. out}}$	p_d	Fault in decoding
1						
2						
3						
4						
5						

Symbol error probability at the decoder input and output are calculated on formulas:

$$p = \frac{N_{\text{er in}}}{N \cdot n} \quad \text{and} \quad p_d = \frac{N_{\text{er out}}}{N \cdot k}, \tag{4}$$

where $N_{er\ in}$ and $N_{er\ out}$ – number of word errors at the decoder input and output in times of supervision.

According to Table. 4 shall be calculated dependence $p = f(h_b^2)$, then the CG is determined at levels $p_d = 10^{-4}$ and $p_d = 10^{-5}$.

6 Description of the computer program of (*n*, *k*) code correcting ability research

6.1 **Decoder**. For researches the realized programmatic model of Meggitt decoder of cyclic (*n*, *k*) code is for $n \leq 32$, the block diagram of which is resulted in a figure 3. A Meggitt decoder is a tabular decoder in which entered the following of decoding [8]:

1. Syndrome analyzer includes pre-computing “Table of syndromes” that contains all errors configuration that can be corrected by this (*n*, *k*) code and their corresponding syndromes.

2. For the accepted code word a concrete syndrome is calculated by ordinary rule is dividing of the accepted code word on generator polynomial. Calculator syndrome is based on an (*n* – *k*)-bit shift register. Syndrome is the remainder of the division.

3. The "Syndrome analyzer" is performed by a search in the tables of the calculated concrete syndrome, reading of errors configuration and presentation of the proper sequence of "1" to "Errors corrector" for the errors correction in the accepted code word.

6.2 **Research of correcting ability of (*n*, *k*) code.** Conducted on a block diagram, resulted in a figure 4.

As a generator of errors symbols is used generator of pseudo noises numbers 0 and 1, in which probability 1 equals probability of error of input symbols *p*.

Meters fix:

- number of code words which are analyzed in times of supervision – decod_suc, let's designate *N*;
- number of code word errors at the decoder input – input_err, let's designate $N_{er.\ in}$;
- number of code word errors at the decoder output – output_err, let's designate $N_{er.\ out}$;
- number of decoding fault (the calculated syndrome is not found in tables) – decod_err.

6.3 **Order of work with computer program.**

1. Measuring of noise immunity is conducted the start of file of **meg32n.exe** (it is in a folder “Laboratory works of TT-2, laboratory work 4.3).

2. On the requirements of the computer program necessary for work basic data are entered.

3. In the program foreseen also step-by-step mode of operation - parameter /t in a command line and conclusion (seal) of table of syndromes is a parameter /s in a command line.

4. Probability of error can be set, both in an ordinary and in scientific format, for example, 0,025 or 2.5e-2.

Intermediate results indicate on a display each 3 seconds.

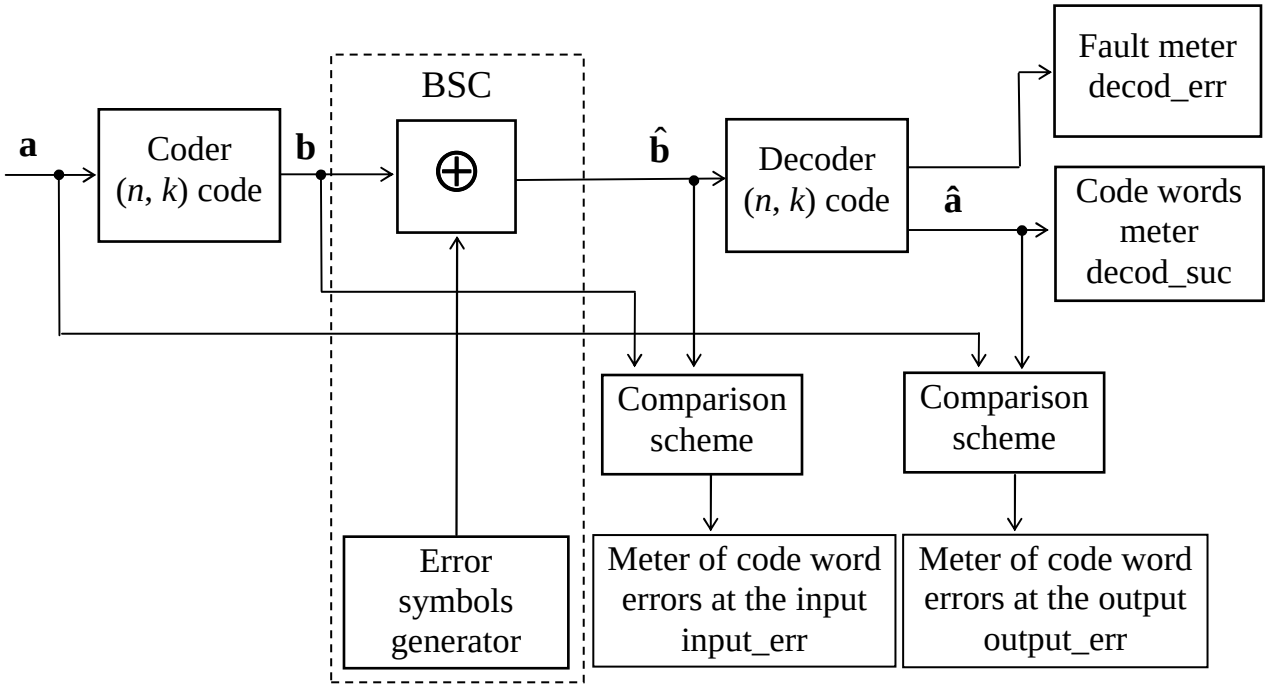

Figure 4 – Block-diagram of (*n, k*) code correction ability researching

7 Requirements to the report

7.1 Name of laboratory work.

7.2 Objectives of laboratory work.

7.3 Results of the home work.

7.4 Block diagram of researches, list of devices (indicators) which are used in LR.

7.5 Results of implementation of items 5.2, 5.3 laboratory task (measured and calculation numerical values etc.).

7.6 Conclusions on every point of laboratory task, in which to give the analysis of the got results is a coincidence of theoretical and experimental information etc.

7.7 Date, signature of student, visa of teacher with an estimation on the 100-mark scale of evaluation.

LW 4.4 Studying of coding and decoding by error-control convolution codes

1 Objectives

1.1. Studying of convolution codes codecs structure.

1.2. Researching of convolution code control ability.

2 Main principles

2.1. **Definition and description of convolution codes.** As is known, in the case of block codes the sequence of information symbols (in future bits) is divided on separate blocks which in future are encoded independent of each other. Thus, the coded sequence is the sequence of independent code words of equal length.

For convolution codes a principle is other. The coding process is continuous and symbols on the encoder output (so-called code symbols) are one semi-infinite code word.

Convolution codes (CC) are the special case of continuous codes. They got the name through its property. The sequence of code symbols on the encoder output is calculated as mathematical operation of digital convolution of information bits on the input with encoder pulse response.

The structure of convolution code encoder and process of coding (decoding) are set by the generator polynomials $g\,(^{i})$, where $i = 1, 2, \ldots, n$; D is delay. As a rule, polynomials are written down briefly, three binary coefficients of polynomial designated as one octal number. For example:

$g^{(1)} = 7$ means $g^{(1)} = 111$, i.e. $g^{(1)} = D^2 + D + 1$ or $1 + D + D^2$;

$g^{(2)} = 21$ means $g^{(2)} = 010101$, i.e. $g^{(2)} = D^4 + D^2 + 1$ or $1 + D^2 + D^4$.

2.2 **Main parameters of convolution codes.** *Code rate* is determined as $R_{\text{code}} = k/n$, where k is the number of encoder inputs, n is number of encoder outputs. Code rate shows that on k input information bits encoder gives out n code symbols.

Constraint length ν characterizes encoder memory and equals the number of memory cells which encoder contains.

Encoder pulse response is the response of CC encoder on one information bit as “1”, which passes through encoder from the i-th input to the j-th output, encoder has kn pulse responses.

Free distance of code d_{f} is the minimum Hamming distance between the sequence of zero code symbols and all other sequences of code symbols. Free distance d_{f} characterizes the correcting ability of CC, i.e. number of errors q_{corr}, which are corrected by CC on length of the accepted code symbols sequence $N = (5–6)\cdot\nu$. Connection between d_{f} and q_{corr} is the same, as for code distance of block error-control code:

$$d_{\text{f}} \geq 2\, q_{\text{corr}} + 1. \qquad (1)$$

2.3. **Convolution codes encoder.** Convolution code encoder contains the clocked memory register for saving of certain number of information symbols and transformer of input information sequence to output code sequence. Block diagram of CC (7, 5) encoder (code rate $R_{\text{code}} = 1/2$) is shown on the figure 1. Encoder contains a shift register with the three memory cells D, modulo-two adder $\oplus$ and multiplexer MX. Inputs of modulo-two adders connected with those cells of register, in which the coefficients of generator polynomials equal to unit.

Information bits $\boldsymbol{a}$ on the input of register. In every clock interval on the adders outputs code symbols $b^{(1)}$ and $b^{(2)}$ appear, i.e. on one information bit there will be two code symbols on output.

For mathematical description of convolution encoding, calculation of digital convolution, a few methods is used: state diagram, tree graph and trellis diagram. The trellis diagram which is considered below is most evident.

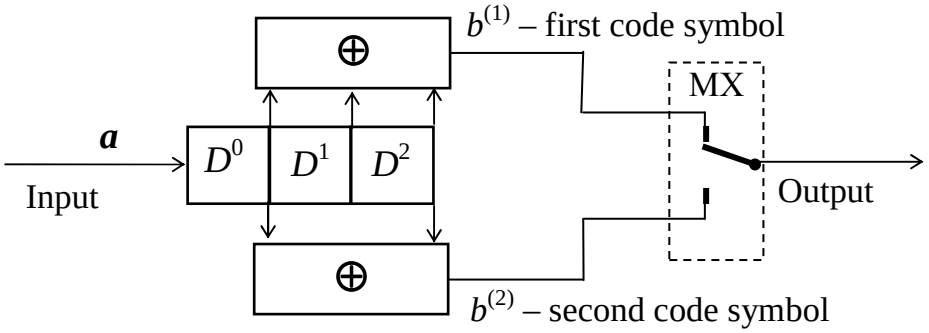

Figure 1 – Block diagram of convolution code (7, 5) encoder

2.4. **Trellis diagram of CC.** The *trellis diagram* of CC called the directed graph with structure of "cells" which repeat periodically. Every cell consists of columns with the identical number of *nodes* connected by graph *branches* (figure 2, *a*). Between procedure of the CC encoding and trellis diagram there is univocal correspondence which is prescribed by such rules:

- every node corresponds to the encoder internal state, as a rule, it is content of two last memory cells in shift register;
- every branch represents encoder transition from one state to other after new information symbol reception in encoder: upper branch correspond to 0, and lower to 1;
- when encoder passes from one state to other, on every branch links this states initial code symbols which appear on encoder output are written down;
- the sequence of branches which is determined by the sequence of information bits and identically gives the code symbols sequence proper to it is called a *path* on a trellis;

So, for CC (7, 5) encoder represented on a figure 1, a trellis will have four states (00, 10, 01 and 11) and it is shown on the figure 2. The evident rule of calculation of initial code symbols on branches showed on figure 3 for the encoder initial state 00 and information bits 0 and 1 on the input. The calculation of initial code symbols of other branches is similar as for other encoder states.

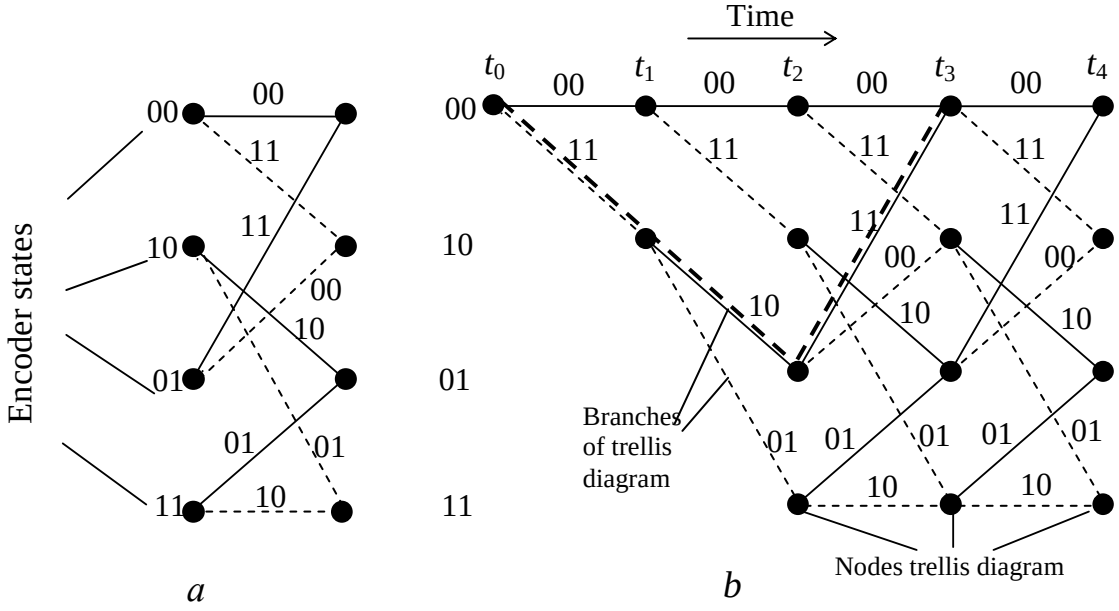

Figure 2 – Trellis diagram of convolution code (7, 5):
a – trellis diagram cell; *b* – trellis diagram evolution on time

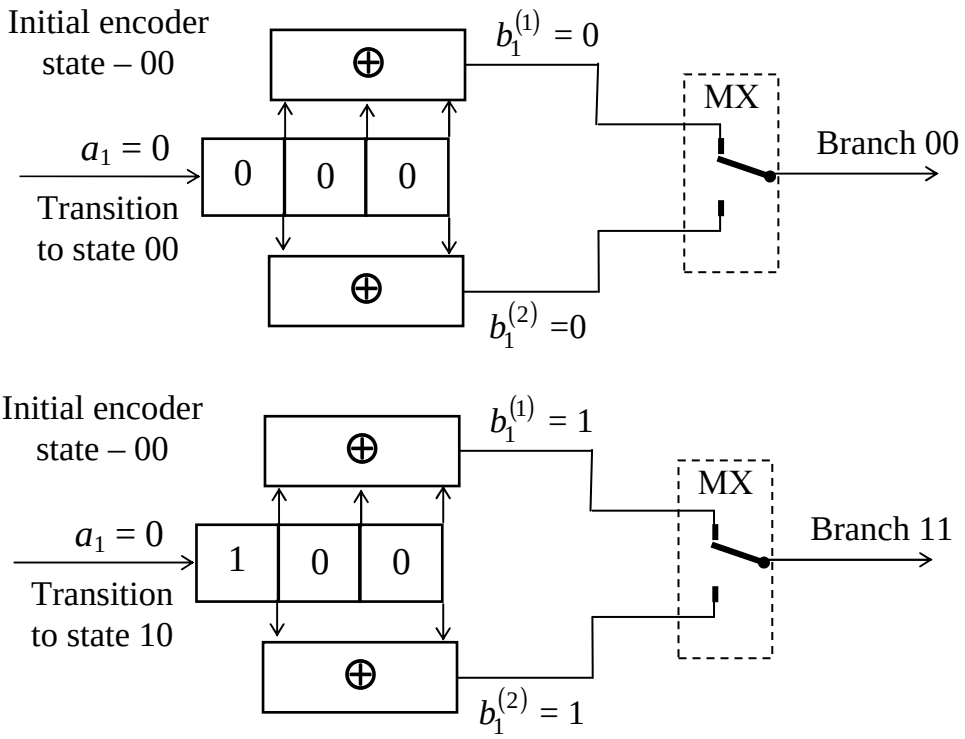

Figure 3 – Convolution coding process for code (7, 5)

Using trellis diagram code *free distance* d_f is calculated as weight (quantity of units) of shortest nonzero path which begins and ends in zero state (on a figure 2 for CC (7, 5) it is the dotted path).

Example 1. For CC (7, 5) encoder represented on a figure 1, find the sequence of code symbols, if sequence of information bits a = 01101000. Accept that in the initial state a register contains zeros.

Solution. Coding sequence is given in table 1, on its base on figure 4 represented trellis diagram with coding path, under coding path understand the sequence of branches which passes during coding process.

Table 1 – Coding process of information bits sequence 01101000, CC (7, 5)

k	Information bit a_k	Encoder contains	Encoder state in moment t_{k-1}	Encoder state in moment t_k	Output code symbols in moment t_k	
					$b_k^{(1)}$	$b_k^{(2)}$
1	0	000	00	00	0	0
2	1	100	00	10	1	1
3	1	110	10	11	0	1
4	0	011	11	01	0	1
5	1	101	01	10	0	0
6	0	010	10	01	1	0
7	0	001	01	00	1	1
8	0	000	00	00	0	0

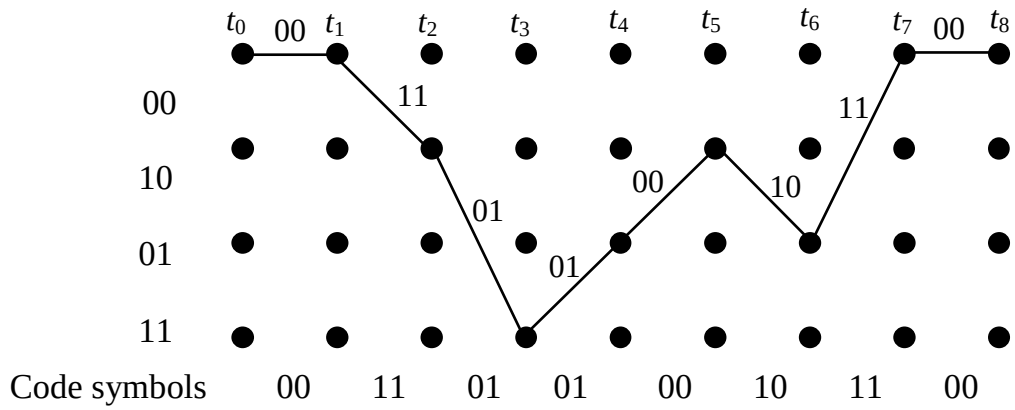

Figure 4 – Encoder trellis diagram for CC (7, 5) and coding path

2.5. **Decoding of convolution codes.** Typical decoding algorithm, based on the probabilistic characteristic of the received signals, is the Viterbi algorithm [9–12], that uses the structure of certain trellis diagram of CC.

On any *k* time interval the Viterbi algorithm provides decoding stages given below.

1) *Calculation* of distance between received symbols and possible symbols, which correspond to all branches of trellis, trellis branches are included in every state in the moment t_k. This distance is called the *metric of branch*.

2) *Construction* of decoder trellis diagram, which similar to encoder trellis diagram, on which represent all possible branches with their metrics. The number of branches and correspond path on a trellis increases at the increase of trellis cells number, which are in point (the *decoding depth*, which depends on the capacity of decoder memory and takes value less then 10 constraint lengths).

3) *Collapsing* of trellis diagram on every step of its construction. Collapsing of trellis diagram it is procedure of exclusion of one from two paths which are included in every decoder state, according to the rule: a path with a greater metrics is excluding, a path with a less metrics stays (if metrics are identical, any path is excluding). *Metric of path* (or *metric of state* of trellis M_{ij}, where *ij* is number of decoder state) represents total metric of branches which a concrete path in the moment t_k passes to the concrete state. Collapsing of trellis diagram is necessary for decreasing of decoder paths number and decreasing of memory capacity.

4) Finding of optimum path on trellis after ending of decoding and making decision about transmitted information bits. A path with the less metric is optimum is called as *surviving path*. Decoding is carrying out on a surviving path: if it passes on the upper trellis branch, information bit is "0", lower – "1".

On the Viterbi algorithm we will consider procedure of decoding on a concrete example for a binary symmetric channel, demodulator gives out the "hard" decision as the sequence of code symbols with errors $\hat{\mathbf{b}}$.

Example 2. Decode using Viterbi algorithm, sequence of the received code symbols: $\hat{\mathbf{b}}$ = 00 11 00 01 00 10. Convolution code (7, 5). Take, that at the beginning of decoding the decoder register is in the zero state.

Note. The received sequence $\hat{\mathbf{b}}$ is the fragment of sequence of encoder code symbols from example 1.

Solution. Procedures of decoding 1) and 2) on the Viterbi algorithm, combine into one during construction of decoder trellis diagram, resulted on figure 5 for the time moments t_0 – t_4. The metric of branches on any moment k is calculated as Hemming distance between a pair of received code symbols $\hat{b}_k^{(1)}\hat{b}_k^{(2)}$ and code symbols of trellis branches. The calculated distance (0, 1 or 2) is shown near every branch on the figure 5. From figure 5 it is visible that from the moment t_2 the number of branches is equal to eight in every trellis cell, and the number of possible paths increases exponentially with the increase of decoding depth.

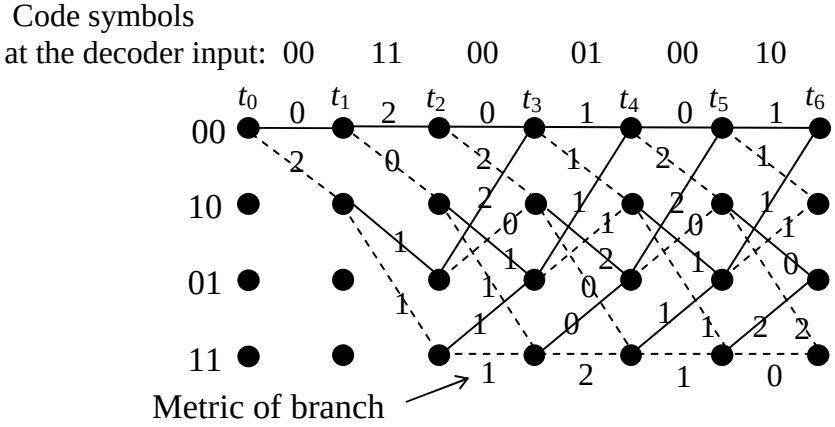

Figure 5 – Fragment of decoder trellis diagram, CC (7,5)

Procedure 3) collapsing of trellis diagram resulted on a figure 6 for the time moments t_3 and t_4.

Notes. 1. For moments t_1 and t_2 there no collapsing of trellis diagram, because in any node which are taken up, one branch enters only.

2. It is obvious, that in errors absence the metric of one path will be zero, because this path repeats the path of encoding.

On a figure 6, *a* for the moments t_2 and t_3 shown all branches and paths, and on a figure 6, *b* only with a less metric. As none from the metrics of paths (states) equals to zero, it means that in the received sequence of code symbols error is present. Making decision about a surviving path is impossible, as two paths (states) have identical metrics.

The process of decoding in a trellis must be continued. For moments t_3 and t_4 collapsing of trellis diagram is shown on a figure 6, *d*. Again in nodes in the moment t_4 paths with less metric are chosen. For the node 11 there are two paths with the metric $M_{11} = 3$, one is chosen arbitrarily.

If to complete decoding on trellis, a path with the metric $M_{01} = 1$ is optimum, shown on a figure 6, *d* by bold dotted line, and the decoded sequence will be $\hat{\mathbf{a}} = 011010$, which coincides with the sequence of information bits in example 1. Conclusion: the error is corrected.

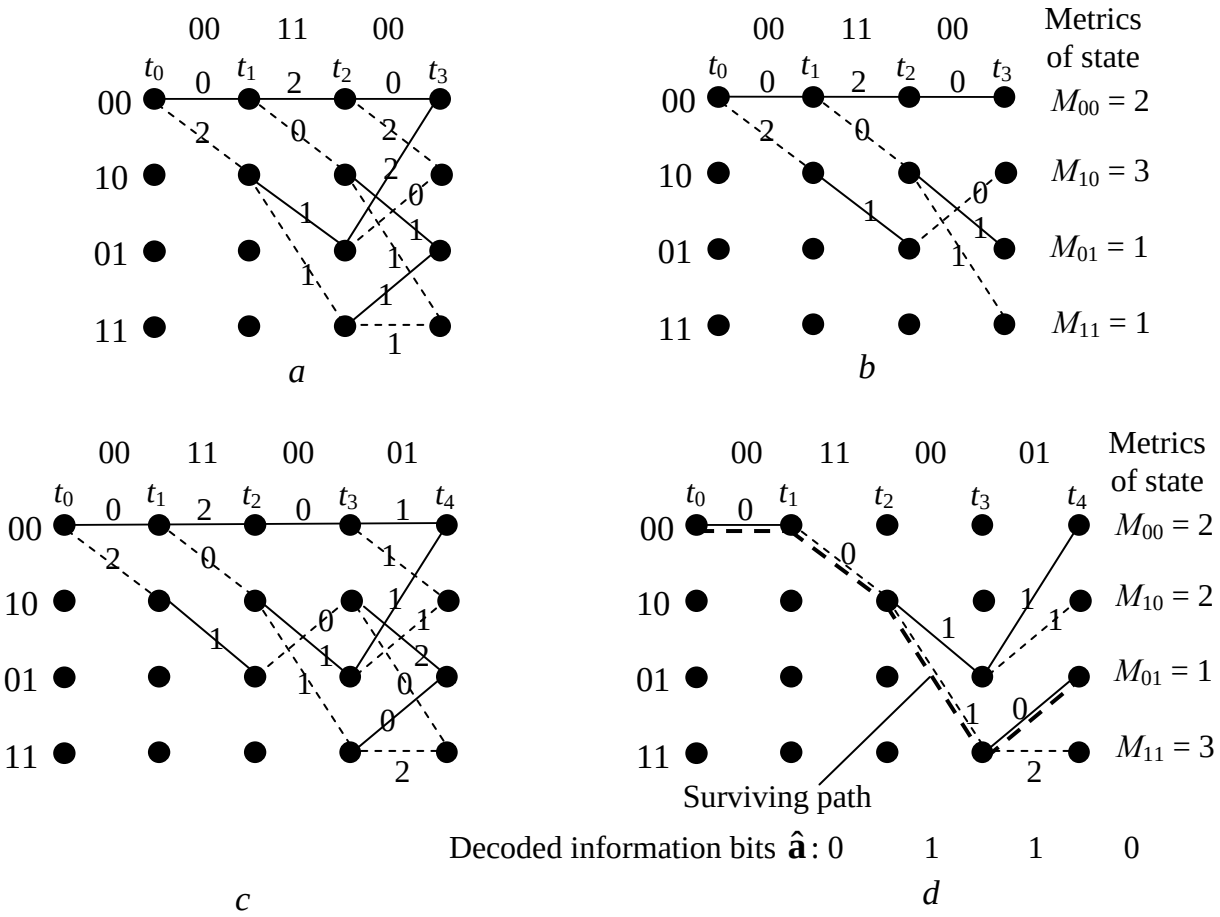

Figure 6 – Surviving path choice: *a* – metrics comparison in moment t_3; *b* – surviving paths in moment t_3; *c* – metrics comparison in moment t_4; *d* – surviving paths in moment t_4

2.6 **Soft decoding.** It is simple modification of the just expounded procedure. At the soft decoding samples from the output of the demodulator matched filter act on the input of decoder. On the first stage of decoding it is needed to replace the Hamming metric on the Euclid metric. All other stages of decoding do not change. So complication of decoder realization with the soft decision not strongly differs from complication of decoder realization with the hard decision. It is one of important advantages of Viterbi decoding algorithm.

3 Questions

3.1 Give definition of the error-control codes in general and convolution in particular.

3.2 What is free distance of convolution code and what does it characterize?

3.3 Give definition to CC encoder pulse response?

3.4 What is CC constraint length?

3.5 Give definition for metrics of: branch, path and state.

3.6 How is it possible to describe work of convolution code encoder?

3.7 How to build the CC encoder trellis diagram?

3.8 Explain the principle of Viterbi decoder work for the CC decoding.

3.9 What surviving path on a trellis determines and how to find it?

3.10 What is decoding depth and how it is determined?

4 Home task

4.1 Study item 2 this LW.

4.2 CC with generator polynomial $g^{(i)} = (7, 5)$ is given. Write down number $(12\,N + 900)$ in binary number system, where N is number of your work place.

Code the got binary sequence by CC (7, 5), build the trellis diagram of this CC encoder and mark the encoding path on it.

On trellis diagram (figure 2, *b*) define free distance of code (7, 5) and multiplicity of corrected errors of this code.

4.3 Be ready to answer questions.

5 Laboratory task

5.1 *Acquaintance with a virtual model.* Start the program 4.4, using the icon "TT (English)" on the desktop. In researches the sequence of information bits (symbols) from home task uses.

5.2 *Research of coding process.* It is necessary to enter information bits **a** got in the home task on the CC encoder panel. Using button “Step by Step” to carry out the coding process (while the button will be active). Write into report the sequence of information bits, content of encoder register and initial sequence of code symbols on every step. Make sure in the rightness of the home task result.

5.3 *Research of decoding process at errors absence.* It is checking of decoder work ability if on the decoder input the sequence of code symbols, which got during encoding (item 5.2). For this purpose it is necessary to press the button “Step by Step” on decoder panel while it will be active. After that push the button “Decision” and surviving path will appear. Compare it with encoding path and make a conclusion about decoder work ability.

5.4 *Research of decoding process at presence of errors.* At first it is necessary to clean the memory register using the button “Clear” on the panel of CC decoder. Put a single error into one of the first six received code symbols by pressure of mouse left button on code symbol in received sequence, in which you want to put an error. Repeat the procedure described in item 5.3. Put into report the fragment of got trellis diagram for the first four steps of decoding ($t_0 - t_4$). Draw all possible collapsed paths with their metrics, surviving path and recovered sequence of information bits. Make conclusions about error correction.

5.5 *Research of decoder control ability.* For this purpose put two errors in a row and separately into any received code symbols and repeat item 5.4, i.e. decode. Put three errors in a row and separately and decode received code symbols. Make conclusions about order of corrected errors.

Record research results as it is shown below. Error code symbols and initial information bits are underlined.

Received code symbols	11	01	0<u>0</u>	01	01	0<u>0</u>	<u>1</u>0	00	10	11	00	11	00
Decoded information bits	1	0	1	0	0	<u>1</u>	<u>1</u>	1	0	0	1	1	0

6 Description of laboratory model

A laboratory model is performed on a computer program. Convolution code is set by the generator polynomials $g^{(i)} = (7, 5)$.

The work of virtual model manages by the left mouse button. Putting of errors is makes by the inversion of symbols from "Received". For the repeating of encoding process it is recommended to clear of encoder memory by using button “Clear” on the CC encoder panel. Similar for a decoder to push button “Clear” on the panel CC decoder. There is also the global clearing which is recommended to use before new research. For this purpose it is necessary to push the button “Clear All” on a panel Code Parameters.

7 Requirements to the report

7.1 Title of laboratory work.

7.2 Objectives of laboratory work.

7.3 Results of the home work performing.

7.4 Encoder block diagram that is used in LW.

7.5 Results of performing of laboratory work items (trellis diagrams, numerical values of code sequences, etc.).

7.6 Conclusions on every item of laboratory task, in which to make the analysis of the got results (coincidence of theoretical and experimental information, control ability of CC (7, 5), etc.)

7.7 The date, signature of student, visa of the teacher with mark.

ATTACHMENT D. DICTIONARIES

D.1 English-Russian dictionary

additional symbol	дополнительный символ
algebraic description	алгебраическое описание
algebraic ring	алгебраическое кольцо
allowed code word	разрешенная кодовая комбинация
antipodal signals	противоположные сигналы
a posteriori probability	апостериорная вероятность
band expansion factor	коэффициент расширения полосы
binary code	двоичный код
block code	блоковый код
Bose-Chaudhuri-Hochquenghem code	код Боуз-Чоудхури-Хоквенгема (БЧХ)
bound	граница
branch metric	метрика ветви
channel capacity	пропускная способность канала
checking relation	проверочное соотношение
code constrained length	длина кодового ограничения
coded modulation	сигнально-кодовая конструкция
code rate	скорость кода
code with even number of units	код с четным числом единиц

code word	кодовое слово (кодовая комбинация)
code word weight	вес кодового слова
coding algorithm	алгоритм кодирования
complexity	сложность
compound estimation	составная оценка
continuous code	непрерывный код
convolutional code	сверточный код
cyclic code	циклический код
cyclic property	циклическое свойство
detect	обнаружить
directed graph	направленный граф
double error	двукратная ошибка
dynamic programming	динамическое программирование
element-by-element reception	поэлементный прием
encoder state	состояние кодера
(energy) coding gain	энергетический выигрыш (от применения) кодирования
error	ошибка
error-control code	корректирующий код
error correction capability	способность исправлять ошибки
error detection capability	способность обнаруживать ошибки
error vector	вектор ошибки
errors configuration	конфигурация ошибок
even number of unit symbols	четное число единиц
exhaustive search	исчерпывающий поиск
fidelity	точность, верность
finite state machine	автомат с конечным числом состояний
forbidden code word	запрещенное кодовое слово
free distance	свободное расстояние
full search algorithm	алгоритм полного перебора
Galois field	поле Галуа
generator matrix	порождающая матрица
generator polynomial	порождающий многочлен
Goley code	код Голея
Hamming code	код Хэмминга
Hamming distance	расстояние по Хэммингу

Hamming upper bound	верхняя граница Хэмминга
hard decision	жесткое решение
hardware	аппаратное обеспечение
identity matrix	единичная матрица
information block of symbols	блок информационных символов
length of encoder shift-register	длина регистра кодера
linear code	линейный код
linear combination	линейная комбинация
majority decoding	мажоритарное декодирование
majority element	мажоритарный элемент
minimum distance of the code	кодовое расстояние
module-2 addition	сложение по модулю 2
multiplicity of errors	кратность ошибки
noise immunity	помехоустойчивость
nonlinear code	нелинейный код
nonsystematic code	несистематический код
odd number of units	нечетное число единиц
orthogonal	ортогональный
parity check matrix	проверочная матрица
path metric	метрика пути
primary code	первичный код
prime number	простое число
probability decoding methods	вероятностные методы декодирования
protective interval	защитный интервал
Reed-Solomon code	код Рида-Соломона
reception in a whole	прием в целом
redundancy	избыточность
row	строка
quantization	квантование
sequential decoding	последовательное декодирование
single error	однократная ошибка
soft decision	мягкое решение
software	программное обеспечение
state diagram	диаграмма состояний
state metric	метрика состояния

survived path	выживший путь
syndrome decoding	синдромное декодирование
systematic code	систематический код
threshold decoding	пороговое декодирование
time-space coding	пространственно-временное кодирование
transposed matrix	транспонированная матрица
trivial estimation	тривиальная оценка
trellis diagram	решетчатая диаграмма
turbo code	турбо код
Viterbi algorithm	алгоритм Витерби

D.2 Russian-English dictionary

автомат с конечным числом состояний	finite state machine
алгебраическое кольцо	algebraic ring
алгебраическое описание	algebraic description
алгоритм Витерби	Viterbi algorithm
алгоритм кодирования	coding algorithm
алгоритм полного перебора	full-search algorithm
апостериорная вероятность	a posteriori probability
аппаратное обеспечение	hardware
блоковый код	block code
Боуз-Чоудхури-Хоквенгема (БЧХ) код	Bose-Chaudhuri-Hochquenghem(BCH) code
вероятностные методы декодирования	probability decoding methods
верхняя граница Хэмминга	Hamming upper bound
вес кодового слова (комбинации)	code word weight
выживший путь	survived path
граница	bound
двоичный код	binary code
двукратная ошибка	double error
диаграмма состояний	state diagram
динамическое программирование	dynamic programming
длина кодового ограничения	code constrained length
длина регистра кодера	length of encoder shift-register
дополнительный символ	additional symbol
единичная матрица	identity matrix

жесткое решение	hard decision
запрещенная кодовая комбинация	forbidden code word
защитный интервал	protective interval
избыточность	redundancy
информационный блок	information block
исчерпывающий поиск	exhaustive search
квантование	quantization
код Голея	Goley code
код Рида-Соломона	Reed-Solomon code
код с четным числом единиц	code with even number of units
код Хэмминга	Hamming code
кодовая комбинация (слово)	code word
кодовое расстояние	minimum distance of the code
конфигурация ошибок	errors configuration
корректирующая способность	adjusting ability
корректирующий код	error-control code
коэффициент расширения полосы	band expansion factor
кратность ошибки	error multiplicity
линейная комбинация	linear combination
линейный код	linear code
мажоритарное декодирование	majority decoding
мажоритарный элемент	majority element
метрика ветви	branch metric
метрика пути	path metric
метрика состояния	state metric
мягкое решение	soft decision
направленный граф	directed graph
нелинейный код	nonlinear code
непрерывный код	continuous code
несистематический код	nonsystematic code
нечетное число единиц	odd number of units
обнаруживать	detect
однократная ошибка	single error
ортогональный	orthogonal
основание кода	size of a code alphabet

поле Галуа	Galois field
первичный код	primary code
помехоустойчивость	noise immunity
пороговое декодирование	threshold decoding
порождающая матрица	generator matrix
порождающий многочлен	generator polynomial
последовательное декодирование	sequential decoding
поэлементный прием	element-by-element reception
прием в целом	reception in a whole
проверочная матрица	parity check matrix
проверочное соотношение	check relation
программное обеспечение	software
пропускная способность канала	channel capacity
простое число	prime number
пространственно-временное кодирование	time-space coding
противоположные сигналы	antipodal signals
разрешенное кодовое слово	allowed code word
расстояние по Хэммингу	Hamming distance
решетчатая диаграмма	trellis diagram
сверточный код	convolutional code
свободное расстояние	free distance
сигнально-кодовая конструкция	coded modulation
синдромное декодирование	syndrome decoding
систематический код	systematic code
скорость кода	code rate
сложение по модулю 2	module-2 addition
сложность	complexity
способность исправлять ошибки	error correction capability
способность обнаруживать ошибки	error detection capability
составная оценка	compound estimation
состояние кодера	encoder state
строка	row
транспонированная матрица	transposed matrix
тривиальная оценка	trivial estimation
турбо код	turbo code

циклический код	cyclic code
циклическое свойство	cyclic property
четное число единиц	even number of unit symbols
энергетический выигрыш (от применения) кодирования	(energy) coding gain

REFERENCES

1. **Теорія** електричного зв’язку / В.К. Стеклов, Л.Н. Беркман; За ред. В.К. Стеклова. – К.: Техніка, 2006. – 552 с.

2. **Банкет В.Л.**, Иващенко П.В., Геер А.Э. Цифровые методы передачи информации в спутниковых системах связи: Учебн. пособ. – Одесса: УГАС, 1996. – 180 с.ё

3. **Банкет В.Л.** Дискретная математика в задачах теории цифровой связи: Учебн. пособие. – Одесса: ОНАС, 2008. – 118 с.

4. **Питерсон У.**, Уэлдон Э. Коды, исправляющие ошибки / Пер. с англ. под ред. Р.Л. Добрушина. М.: Мир, 1976. – 594 с.

5. **Теория** передачи сигналов: Учебник для вузов / А. Г. Зюко и др. – М.: Радио и связь, 1986. – 304 с.

6. **Основы** теории информации и кодирования / Н.В. Кузьмин, В.А. Кедрус. – К.: Вища шк., 1986. – 238 с.

7. **Теорія** електричного зв’язку: Підручник для студентів вузів I та II рівнів акредитації / І.П. Панфілов, В.Ю. Дирда, А.В. Капацін. – К.: Техніка, 1998. – 328 с.

8. **Блейхут Р.** Теория и практика кодов, контролирующих ошибки: Пер. с англ. – М.: Мир, 1986. – 576 с.

9. **Банкет В.Л.** Сверточные коды в системах передачи информации: Учебн. пособие. – Одесса: ОЭИС, 1986. – 57 с.

10. **Теория** электрической связи: Учебник для вузов / А.Г. Зюко и др.; Под ред. Д.Д. Кловского. – М.: Радио и связь, 1998. –432 с.

11. **Скляр Б.** Цифровая связь. Теоретические основы и практическое применение. Изд. 2-е, испр.: Пер. с англ. – М.: Издательский дом "Вильямс", 2003. –1104 с.

12. **Кларк Дж., мл.**, Кейн Дж. Кодирование с исправлением ошибок в системах цифровой связи: Пер. с англ. – М.: Радио и связь, 1987. – 392 с.

Education publication

Banket Victor Leonidovich
Ivaschenko Peter Vasilyevich
Borschova Lesya Mikhailovna
Rozenvasser Denis Mikhailovich

BASES OF THE ERROR-CONTROL CODES THEORY

Education manual

Здано в набір 25.09.2009 Підписано до друку 6.10.2009
Формат 60/88/8 Зам. № 4009
Тираж 100 прим. Обсяг 4,5 друк. арк.
Віддруковано на видавничому устаткуванні фірми RISO
у друкарні редакційно-видавничого центру ОНАЗ ім. О.С. Попова
ОНАЗ, 2010