Ministry of Education and Science of Ukraine
O.S. Popov Odessa National Academy of Telecommunications

A.G. Zuko Telecommunication Theory Department

Ivaschenko P., Rozenvasser D.

TELECOMMUNICATION THEORY

Textbook

Part 2

Odessa 2017

UDK 621.391 Plan of Education Materials Publication in 2017

Ivaschenko P. Telecommunication theory: textbook / P. Ivaschenko, D. Rozenvasser. – Odessa: O.S. Popov ONAT. – Part 2, 2017. – 110 p.

Editor of the English language version: Kuznetsova G.

The Telecommunication Theory textbook consists of two parts:

part 1 – «Communication Signals» and «Theory of Noise Immunity of Telecommunication Signals Reception»;

part 2 – «Transmission of Information in Telecommunication Systems» and «Bases of the Error-control Codes Theory».

This textbook (part 2) contains main theoretical positions, questions for checking the knowledge and English-Russian and Russian-English dictionaries of terms.

The textbook is intended for students majoring in 172 – Telecommunications and radio technology studying the Telecommunication Theory.

APPROVED at the meeting
of A.G. Zuko Telecommunication
Theory Department.
Protocol № 3, October 20, 2016

APPROVED
by O.S. Popov ONAT
Publishing community.
Protocol № 6, February 23, 2016

CONTENTS

1 INFORMATION CHARACTERISTICS OF MESSAGE SOURCES

1.1 Quantitative measure of the information

Transmission systems are created for transfering the information. It is necessary to be able to calculate the amount of the information, which is given out by a source and transferred by a channel. Also it is necessary to estimate the limiting possibility of the transmission channel for transfer of the information.

Therefore it is necessary to use some quantitative measure, which would allow to estimate objectively the amount of the information which the message contains in. Such measure has been introduced by K. Shannon in 1948.

The amount of the information in the message *a* (a sign, a word, a phrase) is defined as:

$$I(a) = \log_2 \frac{1}{P(a)} = -\log_2 P(a), \tag{1.1}$$

where $P(a)$ – probability of the message *a*. The relation (1) is an axiom.

The less is the probability of the message, the more information it contains.

The logarithmic function is used for realization of two obvious properties.

1. The message is beforehand known ($P(a) = 1$ – uncertainty is absent). Then the amount of the information in the message *a* is equal to zero: $I(a) = \log_2 1 = 0$.

2. The source consistently chooses two independent messages a_j and a_k. The probability of such choice $P(a_j, a_k)$ is joint probability of events a_j and a_k

$$\begin{aligned} I(a_j, a_k) &= -\log_2(P(a_j, a_k)) = -\log_2(P(a_j)P(a_k)) = \\ &= -\log_2(P(a_j)) - \log_2(P(a_k)) = I(a_j) + I(a_k). \end{aligned} \tag{1.2}$$

The amount of information in two messages is equal to the sum of amounts of information in each message (the measure is additive).

The units of measure of amount of information are binary unit or bit.

Example 1.1. The probability of message *a* is equal to 1/16. Define the amount of information in this message.

Solution. $I(a) = -\log_2 (1/16) = 4$ bits.

1.2 Models of discrete message sources

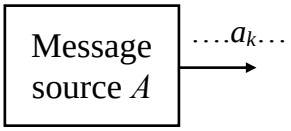


Figure 1.1 – Message source *A*

Let's consider a source of messages *A* (figure 1.1). The source gives out sequence of signs. The source chooses signs from the alphabet randomly $\{a_k\} = a_1, a_2, \ldots a_{M_A}$, where M_A – the size of the source alphabet. Signs may be individual letters, numbers, or their combinations, for example, words or even sentences.

For the source description it is necessary to specify probabilities of signs. There can be such cases.

1. Source without memory: the probability of a given sign does not depend on signs what were before it and will be after it. In this case the source is given by probabilities of signs $P(a_k)$, $k = 1, 2, \ldots, M_A$, while probabilities satisfying the condition

$\sum_{k=1}^{MA} P(a_k) = 1$. The amount of information in any sign a_k is determined by the formula (1.1)

$$I(a_k) = -\log_2 P(a_k). \quad (1.3)$$

Usually, the signs in messages are with different probabilities. Table 1.1 shows the probability distribution of meaningful letters in English texts. Hence, the amount of information in various marks is different.

Table 1.1 – The probability distribution of letters in English texts

Letter	Probability	Letter	Probability	Letter	Probability	Letter	Probability
Space	0,198	R	0,054	U	0,022	V	0,008
E	0,105	S	0,052	M	0,021	K	0,003
T	0,072	H	0,047	P	0,017	X	0,002
O	0,065	D	0,035	Y	0,012	J	0,001
A	0,063	L	0,029	W	0,012	Q	0,001
N	0,059	C	0,023	G	0,011	Z	0,001
I	0,055	F	0,022	B	0,010		

2. Source with memory: the probability of given sign depends on signs that were before it and will be after it. In this case a source is given by joint probabilities of signs. Thus, if the relationship between the signs extends within the two signs a_k and a_j, the source is given by probabilities $P(a_k/a_j)$, $k, j = 1, 2, ..., M_A$; $P(a_k/a_j)$ – the probability of a sign a_k provided that the previous sign a_j is known. In this case, the amount of information in sign a_k is defined by conditional amount of information

$$I(a_k/a_j) = -\log_2 P(a_k/a_j). \quad (1.4)$$

The amount of information in sign a_k is determined by averaging the conditional amount of information (1.4)

$$I(a_k) = -\sum_{j=1}^{MA} \left(P(a_j) \log_2 P(a_k/a_j)\right). \quad (1.5)$$

If the relationship between the signs extends within three signs a_k, a_j and a_l, the source is given by probabilities $P(a_k/a_j,a_l)$, $k, j, l = 1, 2, ..., M_A$ and so on.

The source of discrete messages is also given by the duration of each sign issuing T_k, $k = 1, 2, ..., M_A$. If the duration of issuing the signs is the same, the source is called synchronous, otherwise - asynchronous.

Let's consider the examples of calculating the amount of information of a source output.

***Example 1.2*.** Find the amount of information in a sequence of six signs from some sensor, provided the signs of source are equiprobable and independent and alphabet size $M_A = 16$.

Solving. Since the signs are equiprobable, the probability of each is $P(a_k) = 1/M_A = 1/16$, each sign by the formula (1.3) contains $I(a_k) = -\log_2 16 = 4$ bi-

nary unit of information. Signs are independent, so the amount of information in sequence $N = 6$ signs $I_{\text{seq}} = \sum_k I(a_k) = 6 \cdot 4 = 24$ binary units.

Example 1.3. Find the amount of information in the word Odessa, considering that letters in text are independent.

Solving. Probabilities of letters of the English text are given in Table 1.1. According to the formula (1.1) they contain the following values of amount of information: I(o) = 3,94 binary units; I(d) = 4,84 binary units; I(e) = 3,25 binary units; I(s) = 4,27 binary units; I(a) = 3,99 binary units. Thus, the amount of information in the word Odessa has additive measure because the information is

$$I_{\text{w}} = 3{,}94 + 4{,}84 + 3{,}25 + 2 \cdot 4{,}27 + 3{,}99 = 24{,}56 \text{ binary units.}$$

1.3 Entropy of an independent message source

The source is characterized by the average amount of information, fall on one sign. The average amount is called **source entropy**.

We shall consider a source without memory. It is described by probabilities of separate signs. If probabilities of signs are different, then signs contain different amount of information according to the formula (1.3)

Let's find source entropy

$$H(A) = \overline{I(a_k)} = -\sum_{k=1}^{M_A} P(a_k) \cdot \log_2 P(a_k). \tag{1.6}$$

Entropy properties:

1. Entropy is a real, limited and non-negative value. Property follows from a structure of expression for $H(A)$, and consequently that $0 \le P(a_k) \le 1$.

2. Entropy of determined messages source is equal to zero, $H(A) = 0$ because the probability of one of messages is equal to one and others are equal to zero.

3. Entropy is maximal, if all source messages are equiprobable. We shall prove it.

Let's consider a difference

$$\begin{aligned} H(A) - \log_2 M_A &= -\sum_{k=1}^{M_A} P(a_k) \cdot \log_2 P(a_k) - \log_2 M_A = -\sum_{k=1}^{M_A} P(a_k) \cdot \log_2 M_A P(a_k) = \\ &= \log_2 e \sum_{k=1}^{K} P(a_k) \cdot \ln \frac{1}{M_A P(a_k)}. \end{aligned} \tag{1.7}$$

Let's take the advantage of a known relation

$$\ln x \le x - 1, \tag{1.8}$$

which is correct for any positive x. Then

$$H(A) - \log_2 M_A \le \log_2 e \sum_{k=1}^{M_A} P(a_k) \cdot \left[\frac{1}{M_A P(a_k)} - 1 \right] =$$
$$= \log_2 \left[\sum_{k=1}^{M_A} \frac{P(a_k)}{M_A P(a_k)} - \sum_{k=1}^{M_A} P(a_k) \right] = \log_2 [1-1] = 0, \quad (1.9)$$

equality takes place only at $M_A P(a_k) = 1$, so:

$$P(a_1) = P(a_2) = ... = P(a_K) = 1/M_A$$

and then

$$H_{\max}(A) = -\frac{1}{M_A} \sum_{k=1}^{M_A} \log_2 \left(\frac{1}{M_A} \right) = \log_2 M_A . \quad (1.10)$$

Apparently from the last expression, in case of equiprobable messages entropy increases with increasing of source alphabet size (growth of signs number). At non equiprobable signs entropy, accordingly, decreases.

Example 1.4. The source A of discrete independent messages uses 4 signs with probabilities $P(a_1) = 0{,}7$; $P(a_2) = P(a_3) = P(a_4) = 0{,}1$. Define source entropy and maximal value of source entropy

Solution.

$$H(A) = -\sum_{k=1}^{K} P(a_k) \log_2 P(a_k) = -0{,}7 \cdot \log_2 0{,}7 - 3 \cdot 0{,}1 \cdot \log_2 0{,}1 = 1{,}36 \text{ bits};$$

$$H_{\max}(A) = \log_2 M_A = \log_2 4 = 2 \text{ bit}.$$

Properties of source entropy can be easyly calculated in case of a binary source ($M_A = 2$). Let probabilities of signs $P(a_1) = p$, and $P(a_2) = 1 - p$. Then binary source entropy will be written down

$$H(A) = -P(a_1) \log_2 P(a_1) - P(a_2) \log_2 P(a_2) = -p \log_2 p - (1-p) \log_2 (1-p). \quad (1.11)$$

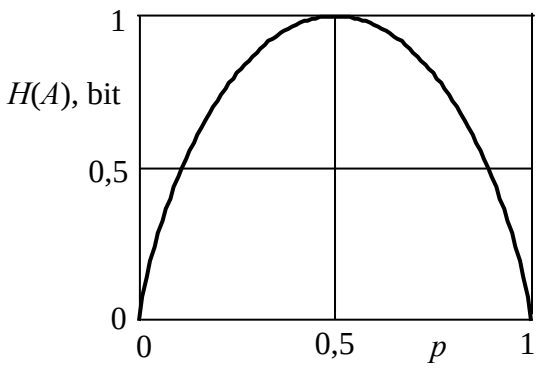


Figure 1.2 – Binary source entropy

In figure 1.2 dependence (1.11) is plotted. From figure it is visible, that entropy is equal to zero at $P(a_1) = 0$; $P(a_2) = 1$ or $P(a_1) = 1$; $P(a_2) = 0$; the maximum of entropy takes place, when $P(a_1) = P(a_2) = 0.5$, and the maximal value is equal 1 bit, as well as calculated under the formula (1.10). If probabilities of signs different and final, then entropy accepts values between zero and one.

1.4 Entropy of a dependent message source

Strong statistical dependence between signs appears at the source output in reality.

So, in texts, probabilities of separate letters depend on what letters preceded them. For example, let's consider the English sensible text. The letter "P" has appeared. The probability of that following will be "A", much more, than probability of the letter "X". The similar situation is observed in case of transfering images – the adjacent elements of the image have usually almost identical brightness and color.

For definition of source entropy when signs are statistically connected, it is necessary to take into account conditional probabilities of separate signs. Let two signs of the message following one after another are statistically connected. Using formula (1.4), we shall define the amount of information in sign a_k that appeared after (already known) sign received earlier a_j In this case entropy of dependent messages source is calculated by averaging on indexes k and j:

$$H_2(A) = \overline{I(a_k/a_j)} = -\sum_{j=1}^{M_A}\sum_{k=1}^{M_A} P(a_j, a_k)\log_2 P(a_k/a_j) = \\ = -\sum_{j=1}^{M_A} P(a_j)\sum_{k=1}^{M_A} P(a_k/a_j)\log_2 P(a_k/a_j) \quad . \qquad (1.12)$$

Calculation becomes complicated. The subscript n recorded entropy $H_n(A)$ indicates the number of signs within which the relationship takes into account. Taking into consideration the above mentioned signs of independent entropy - formula (1.6) can be designated $H_1(A)$.

Example 1.5. Signs at the output of the binary source are equiprobable: $P(a_1) = P(a_2) = 0{,}5$ and dependent within two signs described by probabilities: $P(a_1/a_1) = 0{,}9$; $P(a_2/a_2) = 0{,}8$. Find the source entropy.

Solving. Substituting the source data into formula (1.12), calculations give

$$H_2(A) = -0{,}5[0{,}9\cdot\log_2 0{,}9 + 0{,}1\cdot\log_2 0{,}1 + 0{,}2\cdot\log_2 0{,}2 + 0{,}8\cdot\log_2 0{,}8] = 0{,}595 \text{ bin. unit.}$$

Important that, a source entropy decreases if dependence between signs present

$$H_0(A) \geq H_1(A) \geq H_2(A) \geq \ldots H_n(A). \qquad (1.13)$$

The expression specifies how many signs have statistical dependence.

So, for the Russian text $H_0(A) = 5$ bit, $H_8(A) = 2$ bit.

1.5 Redundancy of a message source

Redundancy of a messages source is a property of a source to give out information by greater number of symbols, than it could be possible.

Example 1.6. Consider two messages:

– first message «On Monday, the second pair will be Laboratory work Telecommunication Theory»;

– second message «Mon. 2 p. LW TT».

Both messages contain the same information, but the first message is too redundant.

Two reasons of redundancy message sources – the relationship between signs of messages and/or different probabilities.

Quantitatively redundancy is defined by a coefficient of redundancy by the formula

$$K_{\text{red}} = \frac{H_{\max}(A) - H_n(A)}{H_{\max}(A)} = 1 - \frac{H_n(A)}{H_{\max}(A)}. \qquad (1.14)$$

This formula can be interpreted as follows: the numerator defines the average "underload" of information signs, and redundancy coefficient – relative (one sign) "underload" of information signs.

The redundancy coefficient is within: $0 \le K_{\text{red}} < 1$. Extreme values: $K_{\text{red}} = 0$ – there is no redundancy; $K_{\text{red}} \to 1$ – high redundancy.

Redundancy of messages source can be considered as negative feature when the source for dispensing a certain amount of information used more signs than you would excessively load through the communication channel to transmit or store messages. To reduce of redundancy of messages you should use effective (economical) coding of messages.

Redundancy of messages source can be considered as positive feature when due to this property one can analyze the message was transmitted communication channel with errors and detect and/or correct errors in it. To achieve this redundancy properties are specifically introduced by certain rules – the so called conversion is called error-control coding.

Example 1.7. Find the coefficient of redundancy of the source, which is set in Example 1.4: $H(A) = 1{,}36$ bit, $H_{\max}(A) = 2$ bit.

Decision. $K_{\text{red}} = (2 - 1{,}36)/2 = 0{,}32$.

1.6 Source rate

The source rate is the amount of information which is given out by a source on average per 1 second

$$R_{\text{s}} = \frac{H_n(A)}{T_{\text{sign}}}, \qquad (1.15)$$

where T_{sign} – the average time, spent by a source on delivery with one symbol.

Example 1.8. Calculate the performance of discrete sources of the messages given in Example 1.4, if durations of symbols are equale to 1, 2, 3 and 4 ms, respectively.

Solving. The entropy of the source calculated in Example 1.4 is equal to 1,36 binary unit, the average length of signs is calculated as expectation.

$$T_{\text{sign}} = 0{,}7 \cdot 1 + 0{,}1 \cdot 2 + 0{,}1\ 3 + 0{,}1\ 4 = 1{,}6 \text{ ms}.$$

Then by the formula (1.15) $R_{\text{s}} = 1{,}36/(1{,}6 \cdot 10^{-3}) = 850$ binary units/s.

1.7 Joint entropy and mutual information

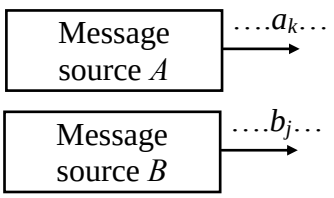


Figure 1.3 – Two message sources

Some problems require the simultaneous consideration of two message sources (figure 1.3). We assume that sources A and B are without memory, duration of signs at sources outputs is the same and equals T, the size of sources alphabets is also the same and equals M_{A}. Sources are set by alphabets and following probabilities:

– unconditional probabilities of signs $P(a_k)$ and $P(b_j)$, $k, j = 1, 2, ..., M_A$;

– conditional probabilities of signs $P(a_k/b_j)$, $P(b_j/a_k)$, $k, j = 1, 2, ..., M_A$;

– joint probabilities of signs $P(a_k, b_j) = P(a_k)P(b_j/a_k) = P(b_j)P(a_k/b_j)$, $k, j = 1, 2, ..., M_A$.

We are interested in the following information characteristics of two messages sources:

– ***joint entropy*** – total information from these sources;

– ***mutual information*** – the amount of information that can be obtained about source *A* by observing the source *B*, or vice versa.

These characteristics are obtained by calculating the joint entropy and mutual information – the average amount of relevant information in one sign.

Joint entropy is calculated

$$H_{JE}(A, B) = H(A) + H(B/A) = H(B) + H(A/B). \quad (1.16)$$

Mutual information is calculated

$$H_{MI}(A, B) = H(A) - H(A/B) = H(B) - H(B/A). \quad (1.17)$$

In relationships (1.16) and (1.17):

– $H(A)$ and $H(B)$ – entropies of sources *A* and *B*, determined by the formula (1.6);

– $H(A/B)$ – entropy of source *A*, provided that the messages are from known source *B*:

$$H(A/B) = -\sum_{k=1}^{MA}\sum_{j=1}^{MA} P(a_k, b_j)\log_2 P(a_k/b_j); \quad (1.18)$$

– $H(B/A)$ – entropy of source *B* provided that the messages are from known source *A*:

$$H(B/A) = -\sum_{k=1}^{MA}\sum_{j=1}^{MA} P(a_k, b_j)\log_2 P(b_j/a_k). \quad (1.19)$$

Exercise 1.1. Prove that the joint entropy of of two independent sources of messages *A* and *B* equal to the sum of their entropies, i.e.

$H_{JE}(A, B) = H(A) + H(B)$.

Solving. Since the joint probability of independent messages $P(a_k, b_j) = P(a_k)P(b_j)$, then $H(B/A) = H(B)$ from formula (1.16) it follows that

$H_{JE}(A, B) = H(A) + H(B)$.

Exercise 1.2. Prove that mutual information of two independent sources of messages *A* and *B* is equal to zero, i.e. $H_{MI}(A, B) = 0$.

Solving. Since for independent messages $P(a_k/b_j) = P(a_k)$ and $P(b_j/a_k) = P(b_j)$, then $H(B/A) = H(B)$ i $H(A/B) = H(A)$. Then from formula (1.17) it follows that $H_{MI}(A, B) = 0$.

***Example 1.9*.** Two messages sources *A* and *B* with entropies: $H(A) = 5{,}2$ binary units; $H(A/B) = 2{,}2$ binary units; $H(B) = 5{,}3$ binary units; $H(B/A) = 2{,}3$ binary units. Calculate joint entropy and mutual information of two sources.

Solving. According to formulas (1.16) and (1.17) joint entropy $H_{JE}(A, B) = H(A) + H(B/A) = H(B) + H(A/B) = 5{,}2 + 2{,}3 = 7{,}5$ binary units; mutual information $H_{MI}(A, B) = H(A) - H(A/B) = H(B) - H(B/A) = 5{,}3 - 2{,}3 = 3{,}0$ binary units.

1.8 Models of continuous message sources

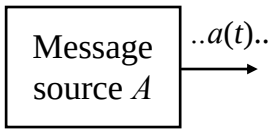


Figure 1.4 – Message source *A*

Let's consider a continuous messages source *A* (figure 1.4). The source gives out the continuous message $a(t)$. The message will be transformed by different transducers to a baseband (analogue) telecommunication signal $b(t)$: $b(t) = k \cdot a(t)$, where k is a factor of proportionality. We shall consider further, that the source gives out a signal $b(t)$.

To solve information tasks it is necessary to set probabilistic characteristics. We assume that the analysis and calculation of information characteristics of sources of continuous messages required the below mentioned characteristics: the signal $b(t)$ – is the implementation of a continuous by level stationary ergodic stochastic process $B(t)$ with characteristics: one-dimensional stochastic characteristics – probability distribution function $F(b)$ and probability density $p(b)$; average value $\overline{B(t)}$ and dispersion $D\{B(t)\}$; correlation function $K_B(\tau)$ and spectral power density $G_B(\omega)$; spectrum concentrated in a limited frequency band $0...F_{max}$.

1.9 Differential entropy of a continuous signal

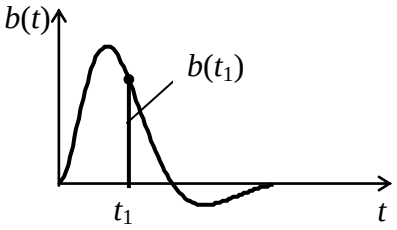


Figure 1.5 – Continuous signal

There is a signal at the source output (figure 1.5). We shall find the amount of information in one sample $b(t_1)$

$$I\left(b(t_1)\right) = \log_2 \frac{1}{P\left(b(t_1)\right)}. \qquad (1.20)$$

As $P\left(b(t_1)\right) \to 0$, then $I\left(b(t_1)\right) \to \infty$. The result is not unexpected – the amount of information in continuous signal $b(t)$ tends to infinity. And this is because continuous message has an infinite set of implementations, the probability of any of them tends to zero.

Intuitively, different signals contain different amount of information. Therefore, for numerical assessment of average amount of information of a continuous source (similar to a discrete source) introduced the concept of differential entropy**.** The term will come as follows. Let's execute quantization of sample (digitization on a level). Then sample will accept values $b_1, b_2, \ldots b_L$, where L – number of quantization levels.

Probabilities of these values are finite

$$P(b_k) = p(b_k)\Delta b, \;\; k = 1, 2, \ldots, L.$$

Entropy of samples

$$H(B) = \lim_{\Delta b \to 0} -\sum_{k=1}^{L} p(b_k)\Delta b \log_2\left(p(b_k)\Delta b\right) = \ldots = -\int_{-\infty}^{\infty} p(b)\log_2 p(b) - \lim_{\Delta b \to 0} \log_2 \Delta b. \qquad (1.21)$$

The first component is named as differential entropy

$$h(B) = -\int_{-\infty}^{\infty} p(b)\log_2 p(b). \qquad (1.22)$$

The second component $-\lim_{\Delta b \to 0} \log_2 \Delta b \to \infty$ (tends to infinity).

Thus, the quantity of the information in one sample tends to infinity. The second term is the same for all signals (not dependent on the characteristics of the signal), so it is not taken into account. For comparison of different messages differential entropy is used.

Differential entropy of sources of a continuous signal $b(t)$ is an analog of source entropy of discrete message. It is formally calculated as the mathematical expectation of the information amount at sample for the probability density that characterizes source uncertainty. The dimension of the differential entropy is binary unit/sample or simple binary unit.

Please note that the differential entropy:

– is calculated per sample and signal samples can be independent or dependent;

– depends on the signal $b(t)$ variance;

– does not show the average amount of information in the sample, but provides the opportunity to compare different sources of information.

Exercise 1.3. Derive the formula for calculating the differential entropy of signal $b(t)$, if it has a Gaussian probability distribution with zero average value and variance σ^2.

Solving. If we substitute the expression $p(b)$ in (1.22) for the Gaussian probability distribution, we get

$$h(B) = \int_{-\infty}^{\infty} p(b)\log_2 \frac{1}{p(b)}\,db = \int_{-\infty}^{\infty} p(b)\log_2 \left[\sqrt{2\pi\sigma^2}\exp\left(\frac{b^2}{2\sigma^2}\right)\right]db =$$

$$= \log_2\sqrt{2\pi\sigma^2}\int_{-\infty}^{\infty} p(b)db + \int_{-\infty}^{\infty} p(b)\log_2\left[\exp\left(\frac{b^2}{2\sigma^2}\right)\right]db.$$

As $\int_{-\infty}^{\infty} p(b)db = 1$, $\log_2 e^z = z\cdot\log_2 e$ and $\int_{-\infty}^{\infty} b^2 p(b)db = \sigma^2$,

then $h(B) = \log_2\sqrt{2\pi\sigma^2} + 0{,}5\log_2 e = \log_2\sqrt{2\pi e\sigma^2}$.

Table 1.2 contains formulas for differential entropy of common probability distributions.

Table 1.2 – Formulas for differential entropy

Signal $b(t)$ probability distribution		Differential entropy $h(B)$
Gaussian	$p(b) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{b^2}{2\sigma^2}\right)$	$\log_2\sqrt{2\pi e\sigma^2}$
Single-sided exponential	$p(b) = \begin{cases} \frac{1}{\sigma}\exp\left(-\frac{b}{\sigma}\right), & b \ge 0, \\ 0, & b < 0 \end{cases}$	$\log_2\sqrt{e^2\sigma^2}$

Two-sided exponential (Laplace)	$p(b) = \frac{1}{\sqrt{2\sigma^2}} \exp\left(-\frac{\sqrt{2}\lvert b\rvert}{\sigma}\right)$	$\log_2 \sqrt{2\,e^2\sigma^2}$
Uniform	$p(b) = \begin{cases} 1/(2b_{\max}), & \lvert b\rvert \le b_{\max} \\ 0, & \lvert b\rvert > b_{\max} \end{cases}$	$\log_2 \sqrt{12\sigma^2}$

Example 1.10. Calculate the differential entropy of signals $b(t)$ with probability distributions: a) Gaussian; b) one-sided exponential; c) two-sided exponential; d) uniform. Samples are independent. Signal variance $\sigma^2 = 10^{-4}$ V^2.

Solving. According to formulas from the table 1.2 we have:

a) for Gaussian distribution

$$h(B) = \log_2 \sqrt{2\pi e\sigma^2} = 0{,}5 \log_2(2\pi e \cdot 10^{-4}) = -4{,}60 \text{ binary units;}$$

b) for single-sided exponential distribution

$$h(B) = \log_2 \sqrt{e^2\sigma^2} = 0{,}5 \log_2(e^2 \cdot 10^{-4}) = -5{,}20 \text{ binary units;}$$

c) for two-sided exponential distribution

$$h(B) = \log_2 \sqrt{2e^2\sigma^2} = 0{,}5 \log_2(2e^2 \cdot 10^{-4}) = -4{,}70 \text{ binary units;}$$

d) for uniform distribution

$$h(B) = \log_2 \sqrt{12\sigma^2} = 0{,}5 \log_2(12 \cdot 10^{-4}) = -4{,}85 \text{ binary units.}$$

Calculations for Example 1.10 show that, firstly, differential entropy can be negative (determined by the variance); secondly, at the same variance differential entropy of signal is maximal, if the signal has normal (Gaussian) probability distribution.

1.10 Epsilon-entropy of a continuous signal

Any continuous signals suppose the approximate representation $\hat{b}(t)$. There is an error of approximate representation $\varepsilon(t) = \hat{b}(t) - b(t)$. Quantitatively the size of error is described by its average square $\overline{\varepsilon^2(t)}$. Signals $b(t)$ and $\hat{b}(t)$ are considered equivalent if the mean square error $\overline{\varepsilon^2(t)}$ does not exceed the set value ε_0^2.

Epsilon-entropy $H_\varepsilon(B)$ is the minimum average quantity of mutual information between $b(t)$ and $\hat{b}(t)$ in one sample at the given allowable error of its approximate representation $\overline{\varepsilon^2(t)} \le \varepsilon_0^2$:

$$H_\varepsilon(B) = \min h_{\text{mut}}\left(\hat{B}, B\right) = \min\left\{h(B) - h\left(B/\hat{B}\right)\right\} = h(B) - \max h(\text{E}). \qquad (1.23)$$

where $h(\text{E})$ – differential entropy of error $\varepsilon(t)$.

Examples of deriviation of formulas for epsilon entropy are given in Exercises 1.4, 1.5, final formulas are given in the table 1.3 for different probability distributions of signals.

Exercise 1.4. Derive the formula for calculating the epsilon entropy of source of continuous signal $b(t)$ in the general case.

Solving. We use the formula (1.23).

For a given error variance $D\{\mathrm{E}(t)\} = \varepsilon_0^2$ minimal value $h(\mathrm{E})$ takes place only for Gaussian distribution of error $\varepsilon(t)$. Then

$$H_\varepsilon(B) = h(B) - \log_2 \sqrt{2\pi e \varepsilon_0^2}\,. \qquad (1.24)$$

Exercise 1.5. Derive the formula for calculating the epsilon entropy of source of continuous signal $b(t)$ with Gaussian probability distribution.

Solving. If we substitute the expression $h(B)$ in (1.24) for Gaussian probability distribution (table 1.2), we get

$$H_\varepsilon(B) = \log_2 \sqrt{2\pi e D\{B(t)\}} - \log_2 \sqrt{2\pi e D\{\mathrm{E}(t)\}} = 0{,}5\log_2[D(B(t))/D(\mathrm{E}(t))] = 0{,}5\ \log_2 \rho_{b/er},$$

where $\rho_{b/er}$ – the ratio of signal and error variances.

Table 1.3 provided formulas for the epsilon entropy of common probability distributions.

Table 1.3 – Formulas for epsilon-entropy

Signal b(t) probability distribution	Epsilon entropy $H_\varepsilon(B)$
Gaussian	$0{,}5\cdot \log_2 \rho_{b/er}$
Single-sided exponential	$0{,}5\cdot \log_2[(e/2\pi)\rho_{b/er}]$
Two-sided exponential (Laplace)	$0{,}5\cdot \log_2[(2e/\pi)\rho_{b/er}]$
Uniform	$0{,}5\cdot \log_2((6/e\pi)\rho_{b/er})$
Explanation: $\rho_{b/er}$ – ratio of signal $b(t)$ and error $\varepsilon(t)$ variances	

Example 1.11. Calculate epsilon-entropy of a continuous signal sources for signals with probability distributions: a) Gaussian; b) one-sided exponential; c) two-sided exponential; d) uniform and ratio of signal and error variances $\rho_{b/er}$ = 43 dB.

Solving. According to formulas (table 1.3) we have:

a) for Gaussian distribution

$$H_\varepsilon(B) = 0{,}5\cdot\log_2 \rho_{b/er} = 0{,}5\cdot\log_2 10^{4{,}3} = 7{,}14 \text{ binary unit};$$

b) for single-sided exponential distribution

$$H_\varepsilon(B) = 0{,}5\cdot\log_2(e\,\rho_{b/er}/2\pi) = 0{,}5\cdot\log_2(e\cdot 10^{4{,}3}/2\pi) = 6{,}54 \text{ binary unit};$$

c) for two-sided exponential distribution

$$H_\varepsilon(B) = 0{,}5\cdot\log_2(e\,\rho_{b/er}/\pi) = 0{,}5\cdot\log_2(2e\cdot 10^{4{,}3}/\pi) = 7{,}54 \text{ binary unit};$$

d) for uniform distribution

$$H_\varepsilon(B) = 0{,}5\cdot\log_2(6\,\rho_{b/er}/\pi e) = 0{,}5\cdot\log_2(6\cdot 10^{4{,}3}/\pi \mathrm{e}) = 6{,}89 \text{ binary unit}$$

1.11 Redundancy of a continuous message source

The redundancy factor of continuous message source can be calculated according to the formula (1.14) at substitution of corresponding values of differential entropy $H_\varepsilon(B)$ and $\max H_\varepsilon(B)$.

$$K_{\text{red}} = \frac{\max H_{\varepsilon}(B) - H_{\varepsilon}(B)}{\max H_{\varepsilon}(B)}. \quad (1.25)$$

If the signal has normal distribution, then $K_{\text{red}} = 0$.

***Example 1.12*.** Calculate the redundancy of continuous signal source with epsilon-entropy calculated in Example 1.11, *b*, $H_{\varepsilon}(B) = 6{,}54$ binary units.

Solving. If we assume that the probability density of signal is Gaussian, then in Example 1.11, *a* max $H_{\varepsilon}(B) = 7{,}14$ binary units and source redundancy $K_{\text{red}} = 1 - 6{,}54/7{,}14 = 0{,}084$ were found.

1.12 Continuous message source rate

Epsilon rate $R_{\text{s}\,\varepsilon}$ of a continuous message source can be calculated according to the formula (1.15) in which it is necessary to substitute values of epsilon entropy $H_{\varepsilon}(B)$, and sampling interval T_{s}, in which samples are independent.

$$R_{\text{s}\,\varepsilon} = H_{\varepsilon}(B)/T_{s}. \quad (1.26)$$

Sampling interval at which samples are independent for signal with uniform spectrum is equal to $T_{\text{s}} = 1/(2F_{\max})$, and

$$R_{\text{s}\,\varepsilon} = 2F_{\max}H_{\varepsilon}(B). \quad (1.27)$$

Questions for section 1

1.1. Definition of "information".

1.2. Why is the logarithmic measure of information used?

1.3. What is called the entropy of the source in information theory and how is it determined for sources of independent and dependent messages?

1.4. Define the concept of "redundancy of the message source".

1.5. Define the concept of "message source rate".

1.6. Name the reasons for the presence of the redundancy of the source.

1.7. What is the maximum entropy of the binary source of independent messages equal to and under what conditions does it take place?

1.8. Define the concepts of "joint entropy" and "mutual information" of two sources of messages.

1.9. Name the basic properties of the joint entropy and mutual information of two sources of messages.

1.10. What is the amount of information from the source of a continuous message?

1.11. Define the concepts of "differential entropy" and "epsilon-entropy".

1.12. In what signals, in the fixed dispersion, is the maximum differential entropy?

1.13. Can the differential entropy acquire negative values?

1.14. What is the difference between differential entropy and the entropy of a discrete source?

1.15. Let $b(t)$ be a baseband signal with a limited spectrum. In the sampler, it is replaced by samplers taken from the Kotelnikov’s interval. Is it lost or not with this transformation of information?

1.16. Define the concept of the epsilon-rate of the source of a continuous signal.

Tasks

1.1. Find the amount of information in the word entropy, assuming that the letters are independent. The probabilities of letters in the English text are given in table 1.1.

1.2. The source alphabet consists of three letters a_1, a_2 and a_3 with probabilities $P(a_1) = 0{,}1$ and $P(a_2) = 0{,}3$. The length of the messages is 1,0 ms, 2,0 ms та 3,0 ms respectively. Calculate the amount of information in each sign, entropy, rate, and redundancy of the source.

1.3. The source of the discrete messages sends a message, using $M_A = 8$ signs. Entropy source equals 2,5 binary units. Calculate the source redundancy.

1.4. Calculate the rate of the source of discrete messages if its entropy is $H(A) = 2{,}25$ binary unit; average sign duration 10 ms.

1.5. Two sources of messages A and B have an entropy: $H(A) = 4{,}2$ binary units; $H(A/B) = 1{,}2$ binary units; $H(B) = 4{,}3$ binary units; $H(B/A) = 1{,}3$ binary units. Calculate the joint entropy and mutual information of two sources of messages.

1.6. Differential entropy of the source of the continuous signal $h(B) = 3{,}8$ binary unit and the differential entropy of error $h(\mathrm{E}) = -2{,}6$ binary units. Calculate the epsilon-entropy of this source.

1.7. Calculate the differential entropy of a continuous source of Gaussian (one-sided or two-sided exponential) probability distribution with a 0,25 V^2 variance.

1.8. Calculate the epsilon-entropy of a source of a continuous signal having a Gaussian probability distribution with a 2,5 V^2 variance. Error variance is 0,0025 V^2.

1.9. Calculate the epsilon-rate of the source of a continuous signal having a Gaussian probability distribution with a dispersion of 2,5 V^2, error variance 0,0025 V^2. The spectrum of the signal is uniform in the frequency range from 0 to 5,0 kHz.

1.10. Calculate the epsilon-rate of the source of a continuous signal having Gaussian probability distribution, the ratio of signal and error variances is 40 dB. The spectrum of the signal is uniform in the frequency range from 0 to 15,0 kHz.

2 EFFECTIVE CODING OF MESSAGES

2.1 Problem of message coding

As a rule, messages that are issued by the source are not adapted for their transmission to communication channels without certain transformations. Therefore, they are coded and, very often, repeatedly. The purpose of message coding is in representation of messages by a digital signal (binary symbols). Binary symbols are convenient for transmitting and storing. This section covers source encoders that provide:

- simple representation (binary symbols) of messages;
- reducing the redundancy of the message, which is typical.

At first we shall consider coding of discrete messages. Message coding is carried out on the basis of a certain code. The code is a rule or the table, according to which each sign in the message is put in conformity with the code word (a set of binary symbols). As a result of coding we receive a baseband digital signal. Inverse transformation of baseband digital signals to messages is called decoding, which is carried out by the decoder on the basis of the same code.

Inclusion of the source coder is shown in figure 2.1: M_A – the size of message source alphabet; m – the code base, as a rule, equals 2.

Coding is performed without information losses. Therefore the source rate is same, at the source output, and at the coder output. Figure 2.1 shows the inclusion of the source coder. The basic characteristic of a digital signal is its rate R.

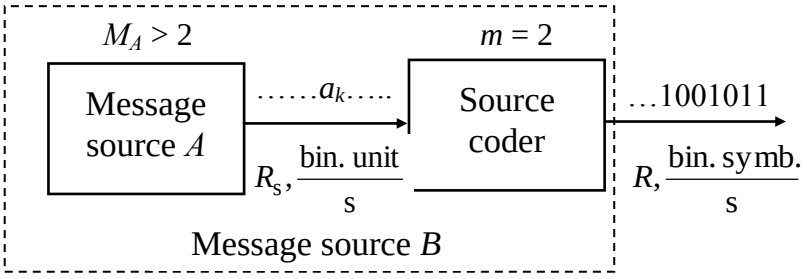


Figure 2.1 – Inclusion of the source coder

If lengths of everyone code words are identical and equal n the code is called uniform (fixed length) or primitive. The length of a binary code depends on the size of the source alphabet and is defined as:

$$n \geq \log_2 M_A. \qquad (2.1)$$

The length of code words can differ in non-uniform (variable-length) codes. At creating a non-uniform code it is necessary to take into account probabilities of coded signs: to give shorter code words to those signs which meet more often and vice versa. Therefore non-uniform codes are named statistical. The example of such a code is the Morse code (alphabet). Morse code was created nearby 170 years ago. Morse made his code intuitively. Further we shall formulate strict methods of non-uniform codes creation.

Exercise 2.1. Prove that when using uniform codes, the redundancy of source B is not less than the redundancy of source A.

The length of the uniform code $n \geq \log_2 M_A$.

Source redundancy factor A $K_{\text{red}}(A) = 1 - H(A)/\log_2 M_{\text{A}}$.

Source redundancy factor B $K_{\text{red}}(B) = 1 - H(B)/\log_2 m = 1 - H(B)$.

Obvious relationships: $H(B) = H(A)/n \le H(A)/\log_2 M_{\text{A}}$.

$$K_{\text{red}}(B) \ge 1 - H(A)/\log_2 M_{\text{A}} = K_{\text{red}}(A).$$

Consequently, as a result of encoding with a uniform code, the redundancy of messages can only remain unchanged or increase.

2.2 Primitive codes for discrete messages

Examples of a primitive code are the following basic codes: ITA-2, ASCII, EBCDIC. The typical feature of a primitive coding is that if the source message has redundancy, then redundancy is not reduced by the coder, it can even increase.

Code ITA-2 (International Telegraph Alphabet № 2) has length n = 5, that is code can be encoded by $M_A \le 2^n$ = 32 signs. How is it possible to code Latin and Russian letters, Arabian digits and moreover some signs? Practically, we have three code tables, and before we start coding the register, which determines what table should be used, and the decoder should use the same table accordingly. It is necessary to replace the register before transition to another table.

Codes ASCII (American Standard Code of Information Interchange) and EBCDIC (Extended Binary-Coded-Decimal Interchange Code) has length n = 8, where each sign is coded with a byte.

From the point of view of the information theory the use of primitive codes does not solve any problems.

2.3 Shannon theorem for the channel without noise

The Shannon theorem of coding for the channel without noise (for a source) approves, that the average length $\overline{n}$ of code words (binary sequences) can be arbitrary closely to the source entropy:

$$\overline{n} \ge H(A) + \varepsilon, \tag{2.2}$$

where $H(A)$ is a message source entropy;

ε – arbitrary small value.

It follows from entropy properties for a binary source (at the coder output) that entropy cannot exceed 1 binary unit. Hence, the average length of a code word $\overline{n}$ cannot be less then entropy.

Let's analyse, can an equality $\overline{n} = H(A)$ be there?

Let's define a source rate at the coder output, considering, that coded signs are independent

$$R_{\text{s}} = \frac{H(A)}{T_{\text{sign}}} = \frac{H(A)}{T_{\text{b}}\overline{n}} = \frac{-\sum_{k=1}^{M} P(a_k)\log_2 P(a_k)}{T_{\text{b}}\sum_{k=1}^{M} P(a_k) n_k}, \tag{2.3}$$

where T_{b} – duration of binary symbols at the encoder output.

We admit, that at coding sign a_k the length of a code word is:

$$n_k = -\log_2 P(a_k) \text{ for all } k. \quad (2.4)$$

Then the sums in numerator and denominator will be equal and the formula (2.3) passes to next:

$$R_s = \frac{1}{T_b} = R. \quad (2.5)$$

That is, each binary symbol transfers one binary unit of the information and equality is carried out $\bar{n} = H(A)$. Reallity equality (2.4) cannot be executed exactly for all k and then $\bar{n}$ come close to $H(A)$. For reduction ε in the ratio (2.2) it is necessary to code the big blocks of signs. It means we switch to the integrated alphabet.

2.4 Principles of discrete messages effective coding

Coding at which the average length of code word is minimized is called effective (economical) coding. At the fixed source rate R_s the rate of a digital signal at the coder output is minimized. Effective coding is called also the compression of information.

To use effective coding it is necessary to take off redundancy at coding. The reasons of redundancy are: dependence between signs of source and not equiprobable signs.

Accordingly, at coding 2 stages are carried out.

– Dependence between signs of a source is eliminated at coding by transition to the integrated signs, which will be independent. For example, switch from coding letters to coding words or phrases.

– The integrated independent signs are coded by one of the developed effective codes: Shannon-Fano or Huffman.

The numerical characteristics of the effective code are:

– *coefficient of code efficiency*

$$\eta = H(A)/\bar{n}, \quad (2.6)$$

that is, the ratio of the source entropy $H(A)$ to the average length of code words

– *average length of code words of non-uniform code*

$$\bar{n} = \sum_{k=1}^{M_A} n_k P(a_k). \quad (2.7)$$

According to Shannon's coding theorem (see Section 2.3) $\eta \le 1$;

– *compression ratio of messages*

$$\mu = n/\bar{n}, \quad (2.8)$$

that is, the ratio of the length of the uniform code n to the average length of code words $\bar{n}$. If acquires meaning $\mu \ge 1$, for better compression, the meaning μ is bigger.

2.5 Shannon-Fano code

A Huffman algorithm of compression of discrete messages with independent signs is optimum and a Shannon-Fano algorithm is nearly optimum. These algorithms

have much in common and provide practically identical degree of compression. The necessary operating condition of these algorithms is that signs probabilities $P(a_0)$, $P(a_1)$, … should be known. The principle of coding is that for more probable signs shorter words are appropriated, and to less probable signs longer words are appropriated. Besides, probabilities of symbols 1 and 0 at the coder output should any way be equal or almost equal.

The Shannon-Fano algorithm is the following. Signs are written down in decreasing order of their probabilities. Then signs are shared in two groups so that the sums of signs probabilities of each group were approximately identical. Zero is assigned to signs from the first group, as the first symbol of a non-uniform code, and the one is assigned to symbols of the second group. Each of the received groups containing more than one sign, is devided into two equiprobable groups and a rule of coding is repeated. This process proceeds until one coding sign will stay in each group.

Example 2.1. The source alphabet has the size $M_A = 6$, signs in messages are independent and have such probabilities: $P(a_1) = 0{,}4$; $P(a_2) = 0{,}2$; $P(a_3) = P(a_4) = 0{,}15$; $P(a_5) = P(a_6) = 0{,}05$, and the duration of issuing one sign is 1 ms. Create Shannon-Fano code. Calculate the source entropy, the average length of code words, the rate of the digital signal at the encoder output, compare the redundancy coefficients at the encoder input and output, calculate the efficiency coefficient and the compression coefficient of the received code.

Solving. Among the possible variants of constructing a code, consider the algorithm shown in Table 2.1.

Table 2.1 – Example of Shannon-Fano code construction

a_i	$P(a_i)$	Division into groups				Code words	n_j
a_1	0,4	0				0	1
a_2	0,2	1	0	0		100	3
a_3	0,15			1		101	3
a_4	0,15		1	0		110	3
a_5	0,05			1	0	1110	4
a_6	0,05				1	1111	4

Entropy of the source

$$H(A) = -\sum_{j=1}^{M_A} P(a_j)\cdot \log_2 P(a_j) = -\left(0{,}4\cdot \log_2 0{,}4 + 0{,}2\cdot \log_2 0{,}2 + 2\cdot 0{,}15\cdot \log_2 0{,}15 + 2\cdot 0{,}05\cdot \log_2 0{,}05\right) =$$

$$= 2{,}25 \text{ binary units.}$$

The average length of code words:

$$\bar{n} = \sum_{j=1}^{M_A} n_j \cdot P(a_j) = 1\cdot 0{,}4 + 3\cdot(0{,}2 + 0{,}15 + 0{,}15) + 4\cdot 2\cdot 0{,}05 = 2{,}3 \text{ bin. symb.}$$

Digital signal rate at the encoder output

$$R = \bar{n}/\bar{T} = 2{,}3/10^{-3} = 2300 \text{ bin.units/s (bps)}.$$

Source A redundancy factor

$$K_{\text{red}}(A) = 1 - \frac{2{,}25}{\log_2 6} = 0{,}13.$$

To calculate the source redundancy factor B (at the encoder output) it is necessary to calculate the entropy of this source $H(B)$. Find the probability of symbols "1" and "0" of source B output.

$$P(1) = 0{,}2 \cdot \frac{1}{3} + 2 \cdot 0{,}15 \cdot \frac{2}{3} + 0{,}05 \cdot \frac{3}{4} + 0{,}05 = 0{,}354;$$

$$P(0) = 0{,}4 + 0{,}2 \cdot \frac{2}{3} + 2 \cdot 0{,}15 \cdot \frac{1}{3} + 0{,}05 \cdot \frac{1}{4} = 0{,}646.$$

Entropy of the source B

$$H(B) = -(0{,}354 \cdot \log_2 0{,}354 + 0{,}646 \cdot \log_2 0{,}646) = 0{,}94 \text{ bin. un.}$$

Source B redundancy factor

$$K_{\text{red}}(B) = 1 - 0{,}94 = 0{,}06.$$

The values obtained indicate that encoding reduces redundancy.

To compare the received code with other codes we will calculate:

coefficient of code efficiency $\eta = \dfrac{H(A)}{\bar{n}} = \dfrac{2{,}25}{2{,}3} = 0{,}98;$

compression coefficient of non-uniform code $\mu = \dfrac{n}{\bar{n}} = \dfrac{3}{2{,}3} = 1{,}3,$

where n – length of a uniform code, which is defined as an integer that satisfies the condition

$$n \geq \log_2 M_A.$$

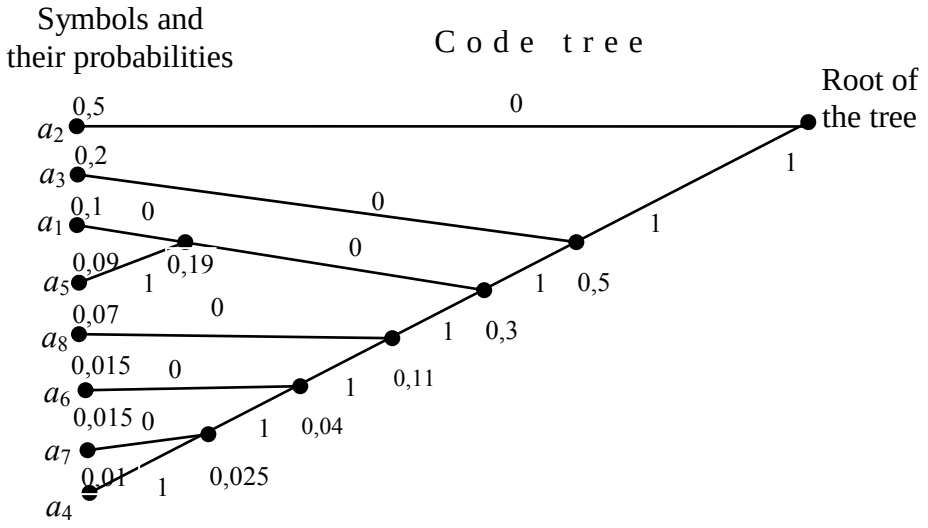
2.6 Huffman code

In case of Huffman coding algorithm the coding signs are written down in decreasing order of their probabilities. If some signs have identical probabilities, they are placed in any order. Then it is necessary to construct the probability tree: choose two signs with the least probabilities and form the first branch of the tree. The chosen signs are united into "intermediate" sign with probability, equal to the sum of probabilities of the chosen signs. Then among the remained signs (together with the intermediate sign) again find two signs with the least probabilities and do the same as at the first step. This procedure is carried out until all signs from the source alphabet and intermediate signs will be used and the root of tree with probability equals 1 will be received.

“Movement” up a tree from root to corresponding symbol is carried out by code words. Passing through a branch means we have to add a binary symbol to code word: if "movement" is going onthrough upper branch then one is added and if it is going through down branch, then zero is added.

Example 2.2. The source alphabet has the size $M_A = 8$. Signs in messages are independent and have such probabilities: $P(a_1) = 0{,}1$; $P(a_2) = 0{,}5$; $P(a_3) = 0{,}2$; $P(a_4) = 0{,}01$; $P(a_5) = 0{,}09$; $P(a_6) = P(a_7) = 0{,}015$; $P(a_8) = 0{,}07$. Construct Huffman code. Calculate the source entropy, the average length of code words, the rate of the digital signal at the encoder output, compare the redundancy coefficients at the encoder input and output, calculate the efficiency coefficient and the compression coefficient of the received code.

Solving. The construction of the Huffman code is clearly shown in figure 2.2. Regrouping is not reflected in the figure - in such a simple example, it is mentally done.



Code table

Symbol	$P(a_k)$	Code word
a_2	0,5	**0**
a_3	0,2	**10**
a_1	0,1	**1100**
a_5	0,09	**1101**
a_8	0,07	**1110**
a_6	0,015	**11110**
a_7	0,015	**111110**
a_4	0,01	**111111**

Figure 2.2 – Example of code tree constructing

$$\bar{n} = \sum_{i=1}^{M_A} n_i \cdot P(a_i) = 1\cdot 0{,}5 + 2\cdot 0{,}2 + 4(0{,}1 + 0{,}09 + 0{,}07) + 5\cdot 0{,}015 + 6(0{,}015 + 0{,}01) =$$

$$= 2{,}17 \text{ binary symbols.}$$

Entropy of the source

$$H(A) = -\sum_{k=1}^{M_A} P(a_k)\cdot \log_2 P(a_k) = -(0{,}1\cdot \log_2 0{,}1 + 0{,}5\cdot \log_2 0{,}5 + 0{,}2\cdot \log_2 0{,}2 + 2\cdot 0{,}01\cdot \log_2 0{,}01 +$$

$$+ 0{,}09\cdot \log_2 0{,}09 + 2\cdot 0{,}015\cdot \log_2 0{,}015 + 0{,}07\cdot \log_2 0{,}07) = 2{,}12 \text{ bin.un.}$$

Compare redundancy of messages at the encoder input and output. Source A redundancy factor

$$K_{\text{red}}(A) = 1 - \frac{2{,}12}{\log_2 8} = 0{,}29 .$$

To calculate the source redundancy factor B (at the encoder output) it is necessary to calculate the entropy of this source $H(B)$. Find the probability of symbols "1" and "0" of source B output.

$$P(1) = 0{,}2\cdot 1/2 + 0{,}1\cdot 2/4 + 0{,}09\cdot 3/4 + 0{,}07\cdot 3/4 + 0{,}015\cdot 4/5 + 0{,}015\cdot 5/6 + 0{,}01 = 0{,}305;$$

$$P(0) = 0{,}5 + 0{,}2\cdot 1/2 + 0{,}1\cdot 2/4 + 0{,}09\cdot 1/4 + 0{,}07\cdot 1/4 + 0{,}015\cdot 1/5 + 0{,}015\cdot 1/6 = 0{,}695.$$

Entropy of the source B

$$H(B) = -(0{,}305\cdot \log_2 0{,}305 + 0{,}695\cdot \log_2 0{,}695) = 0{,}89 \text{ binary unit.}$$

Source B redundancy factor

$$K_{\text{red}}(B) = 1 - 0{,}89 = 0{,}11.$$

So,

$$K_{\text{red}}(B) < K_{\text{red}}(A) -$$

efficient encoding reduces redundancy.

To compare the received code with other codes we will calculate:

coefficient of code efficiency $\eta = \frac{H(A)}{\bar{n}} = \frac{2{,}12}{2{,}17} = 0{,}98$;

compression coefficient of non-uniform code $\mu = \frac{n}{\bar{n}} = \frac{3}{2{,}17} = 1{,}38$,

where n – the length of a uniform code, which is defined as an integer that satisfies the condition

$$n \geq \log_2 M_A.$$

2.7 Application of discrete messages effective coding

It is necessary to note, that Huffman and Shannon-Fano codes are **prefix** codes. The codes are called prefix codes when there is no need to separate symbols between code words. From tables of Huffman and Shannon-Fano codes it is visible, that any of short code words is not the beginning of a longer word.

Let's code sequence of signs… $a_1\ a_6\ a_3\ a_1\ a_2\ a_0$ … with Huffman code

… 1 1 0 0 1 0 0 0 1 1 0 1 1 0 0 0 1 …

We use the same table for decoding and we shall receive

… $a_1\ a_6\ a_3\ a_1\ a_2\ a_0$ …

If a code is uniform (primitive), separating symbols between code words are required. For example, $M = 4$

a_i	a_0	a_1	a_2	a_3
Code word	00	01	10	11

Let's code sequence of signs… $a_1\ a_3\ a_2\ a_1\ a_2\ a_0$ … with the mentioned code

… 0 1 1 1 1 0 0 1 1 0 0 0 … It is necessary to know boundaries of code words for correct decoding. For this purpose enter a separating symbols.

… 1 0 1 1 1 1 1 0 1 0 1 1 1 0 1 0 0 1 …

Separating symbols are redundant, additional resource of a communication channel is spent for their transmission.

Huffman codes are used in some telecommunication technologies, for example, data compression in Rec. V.42, coding of images by method MPEG (Motion Picture Expert Group).

The universal algorithm of compression Ziv-Lempel-Whelch is used in modern computer archivers. This algorithm is updated by Huffman algorithm. For some types of files the factor of compression can achieve 5…10.

2.8 Digital methods of analog signals transmission

Methods of digital transmission of analog signals are widely used in modern telecommunications:

– the analog signal will be transformed in to digital (the sequence of binary symbols);

– the digital signal is transmitted by a digital communication channel;

– from digital signal the analog signal is recovered.

Any method of digital transmission is characterized by the rate of a digital signal R (bit per second) and accuracy of transmission – the signal/quantization noise ratio ρ_q. The problem is to satisfy the requirement to the ratio ρ_q at minimal value R.

There were developed a lot of digital methods of transmission. Among them the elementary method is pulse code modulation (PCM)[1].

2.9 Discretization of analog signals on time

At any method of digital transmission the analog signal $b(t)$, first of all, will be transformed to a discrete signal. It represents itself the sequence of samples $b(kT_s)$, taken through sampling interval $T_s \le 1/(2F_{max})$, where F_{max} is the maximal frequency of a signal spectrum $b(t)$. This transformation is named the sampling of a signal on time, and the device for its realization is called sampler. The sampling frequency should be not less than the double frequency F_{max}:

$$f_s = 1/T_s \ge 2F_{max}. \qquad (2.9)$$

According to Kotelnikov theorem, the performance of this ratio guarantees an opportunity of exact recovery of an analog signal on samples. Such recovery is carried out by LPF with a cut frequency F_{max} (figure 2.3).

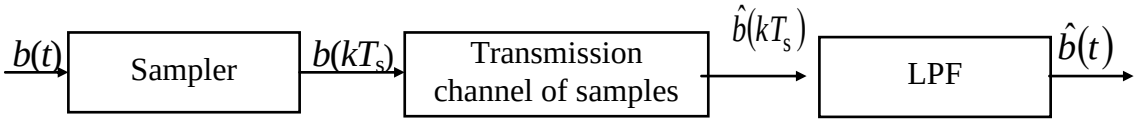


Figure 2.3 – Transmission of an analog signal by samples

All methods of digital transmission of analog signals differ in the ways of representation of discrete signals by digital signals. The converter of samples in a digital signal is called the coder of digital transmission system (DTS), and the converter of a digital signal in samples is called DTS decoder.

2.10 Methods of pulse code modulation

The typical feature of these methods is that each sample is represented by a digital signal independently of other samples. Methods differ by a used code. The diagram that describes the transmission of samples at PCM is represented at figure 2.4.

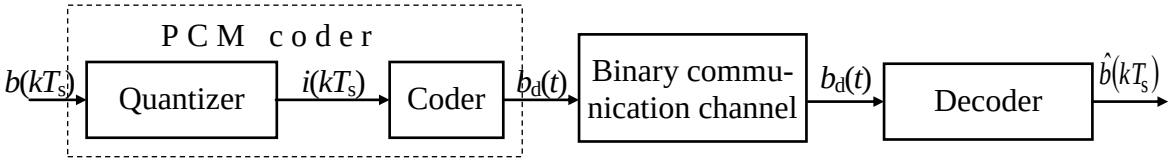


Figure 2.4 – PCM coder and decoder

The basic parameter of a quantizer is the number of quantization levels L. At uniform quantization a range of values b from $-b_{max}$ to b_{max} is broken into $L - 1$ intervals in size

$$\Delta b = 2b_{max}/(L - 1), \qquad (2.10)$$

[1] Despite the presence of a word "modulation", these methods of transmission have relation neither to analog, nor to digital modulation.

which is called quantization step. Figure 2.5 shows breakdown at $L = 8$. Discrete values b_i correspond to the middle of intervals. The index i accepts values 0, ±1, ±2, …, ±0,5L – 1. Discrete values are determined $b_i = i \cdot \Delta b$. At quantization each sample $b(kT_s)$ is approximated to the nearest discrete value b_i, and at a quantizer output the integer $i(kT_s)$ acts. Representation of sample $b(kT_s)$ by discrete value b_i brings an error

$$\varepsilon_q(kT_s) = i(kT_s)\cdot\Delta b - b(kT_s), \qquad (2.11)$$

which is called a quantization noise.

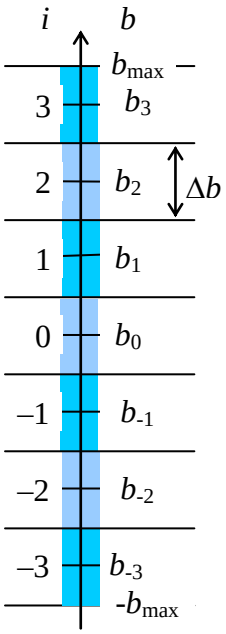


Figure 2.5 – To the explanation of quantization

In the coder included in PCM coder (figure 2.4), numbers $i(kT_s)$ are represented by the given binary code. The length of a code

$$n = \log_2 L. \qquad (2.12)$$

The digital signal at a coder output $b_d(t)$ has the rate

$$R = n \cdot f_s. \qquad (2.13)$$

The decoder from a digital signal $b_d(t)$ forms numbers $i(kT_д)$ on which quantizing samples are recovered: $b_q(kT_s) = i(kT_s)\Delta b$. They are recovered samples of a transmitted analog signal $\hat{b}(kT_s) = b_q(kT_s)$. From formula it follows, that samples are recovered with errors $\varepsilon_q(kT_s)$.

Assuming that the quantization noise has a uniform distribution of probabilities in the range $-\Delta b/2 \le \varepsilon \le \Delta b/2$, then the dispersion of quantization noise is the dispersion of the magnitude with a uniform distribution of probability equals $D\{E_q\} = (\Delta b)^2/12$. As the average value of the quantization noise is zero, the quantization noise power is $\overline{\varepsilon_q^2} = D\{E_q\}$ and

$$\overline{\varepsilon_q^2} = \frac{(\Delta b)^2}{12}. \qquad (2.14)$$

The ratio signal/quantization noise is $\rho_q = P_b/\overline{\varepsilon_q^2}$. The power of signal is $P_b = b_{max}^2 / K_A^2$, where K_A is an amplitude factor of an analog signal. By the formula (2.10) we calculate $(\Delta b)^2 = 4b_{max}^2/(L-1)^2$. Then the signal / (quantization noise) ratio will be

$$\rho_q = \frac{P_b}{\overline{\varepsilon_q^2}} = \frac{12 b_{max}^2 (L-1)^2}{4 K_A^2 b_{max}^2} = \frac{3(L-1)^2}{K_A^2}. \qquad (2.15)$$

The above analyzed transformation of an analog signal to a digital signal with uniform quantization step is called an analog-digital conversion (ADC); return transformation is called a digit-analog conversion (DAC). ADC and DAC diagrams are shown at figure 2.6.

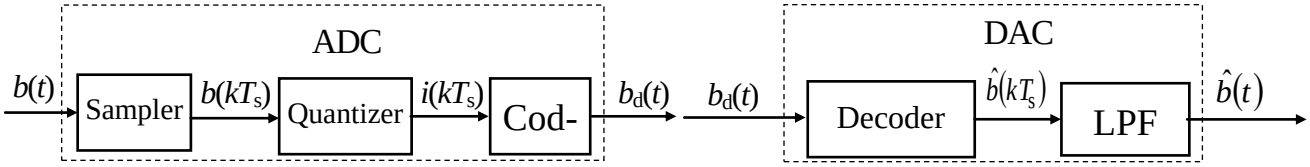


Figure 2.6 – ADC and DAC

The noise immunity of the transmission system from PCM is determined by the signal / noise ratio at the DAC output

$$\rho_{out} = \frac{P_b}{\overline{\varepsilon_q^2} + \overline{\varepsilon_{d\,ch}^2}}, \qquad (2.16)$$

where P_b – average power of the analog signal.

$\overline{\varepsilon_q^2}$ – quantization noise power at DAC output, caused by quantization in ADC;

$\overline{\varepsilon_{d\,ch}^2}$ – noise power at DAC output, caused by errors in the digital communication channel

$$\overline{\varepsilon_{d\,ch}^2} = p(\Delta b)^2 \cdot (4^n - 1)/3, \qquad (2.17)$$

where p – symbol error probability at DAC input (channel output).

When a DAC conversation signal is restorated, errors at the output of the digital communication channel result in a significant deviation of the received signal from the transmitted. It is perceived as a click. Therefore, often for digital conversation channels, standards are set not on the magnitude of signal power $\overline{\varepsilon_{d\,ch}^2}$, but the error probability which is based on the average time interval between clicks that is equal, for example, 30 seconds. In this case, the signal / noise ratio at the DAC output is taken $\rho_{out} = \rho_q$.

Example 2.3. Digital conversation signal rate is R = 64 kbps. Find the allowable error probability p_{all}.

Solving. If you assume that the average time interval between clicks is equal T_{cl} = 30 s, then the error probability

$$p_{all} = 1/(RT_{cl}) = 1/(64 \cdot 10^3 \cdot 30) = 5 \cdot 10^{-7}.$$

Exercise 2.2. Derive the formula of noise power of DAC output, caused by errors in the digital communication channel (formula (2.17)), according to known quantization step Δb, the number of quantization levels L, the length of the ADC code n, symbol error probability p.

Solving. Mistakenly accepted symbols in code words lead to a change in quantization level L_i to the value L_{i+m} with error $\varepsilon_{d\,ch} = L_{i+m} - L_i$, which depends on the digit k. In the code word where the error occurred, the error value (i.e. its "weight") is determined as $\varepsilon_{d\,ch}(k) = 2^{(k-1)} \cdot \Delta b$, k = 1, 2, 3,..., n.

Errors in digits of code words are equiprobable. In real transmission systems $p << 1$. Then the probability that the code word contains an error is np. Consequently,

the power of the noise caused by errors in the digital communication channel is determined

$$\overline{\varepsilon_{\text{d ch}}^2} = \frac{1}{n}\sum_{k=1}^{n}\varepsilon_{\text{d ch}}^2(k)\cdot n\cdot p = \sum_{k=1}^{n}2^{2(k-1)}(\Delta b)^2 .$$

If we take into account equality $\sum_{k=1}^{n}2^{2(k-1)} = (4^n - 1)/3$, we'll get the following formula $\overline{\varepsilon_{\text{d ch}}^2} = p(\Delta b)^2(4^n - 1)/3$.

Non-uniform quantization. The need for nonuniform quantization follows from the fact that at the step of quantization Δb = const signal/(quantization noise) ratio will be different for the large and small levels of the analog signal. Therefore, small signal levels should be quantified with a small step Δb, and large levels – with a bigger step Δb.

This method of transmission is technically implemented when a compressor combines an analog signal, a quantizer with a uniform step and an expander (Figure 2.7). It is frequently used for conversational messages (PCM for non-uniform quantization of conversational signals standardized by the ITU-T G.711 Recommendation: 8 kHz sampling frequency; code length at the PCM encoder output is 8 digits), because they have such features:

– probability distribution is Gaussian, that is, small signal levels occur more often than large;

– human ear perceives the sound nonlinearly – better distinguishes the value of sound pressure on small levels.

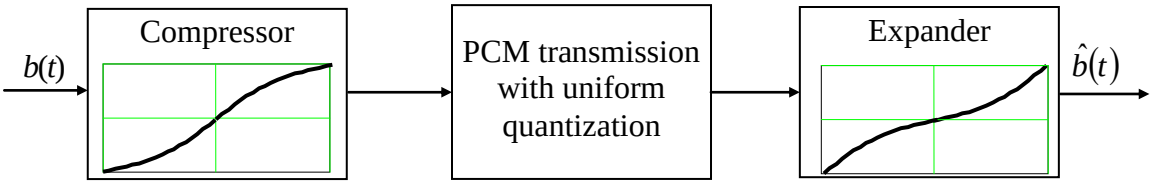


Figure 2.7 – PCM transmission with companding

Compression of the conversational signal provides reduction of the amplitude of the analog signal K_A. According to formula (2.15), the signal/(quantization noise) ratio (or, correspondingly, the digital signal rate) can be reduced.

The optimal companding law is the logarithmic compression law, but it is technically difficult to enforce it. Therefore, the laws of companding apply:

A-law (in Europe and Asia)

$$y = y_{\max}\frac{[A(|x|/x_{\max})]}{1+\ln A}\operatorname{sgn}x \text{ for } 0 < |x|/x_{\max} \le 1/A;$$

$$y = y_{\max}\frac{1+\ln[A(|x|/x_{\max})]}{1+\ln A}\operatorname{sgn}x \text{ for } 1/A < |x|/x_{\max} < 1,$$

where x and y – signal value at the input and output of the compressor, respectively;

x_{max} and y_{max} – maximum positive values of the signal at the input and output of the compressor, respectively;

A – positive constant, its default value A = 87,6;

$\operatorname{sgn} x = \begin{cases} +1 & \text{for} \quad x \geq 0, \\ -1 & \text{for} \quad x < 0 \end{cases}$ – sign function;

μ-law (in the USA, Canada, Japan and some other countries)

$$y = y_{max} \frac{\ln(1 + \mu(x / x_{max}))}{\ln(1 + \mu)},$$

where μ – positive constants, its typical value μ = 255.

Today the existing two algorithms of PCM coding with non-uniform quantization (A-law and μ-law) provide the necessary quality of digital signal transfer.

Technical execution of ADC and DAC at PCM. As it follows from the explanations above, calculation formulas, exercises and examples, ADC and DAC for PCM may have different parameters: sampling frequency, number of quantization levels and corresponding code length, different codes and laws of companding. Compressor (and accordingly - expander) can be turned on either in the circuit of the analog signal, or in the circuit of the digital signal.

From the technological point of view, especially in mass production, such diversity of ADC and DAC is inappropriate. Today, because of the massive introduction of digital technologies, there is such a compromise: firms produce a typical ADC (and, accordingly, DAC) with a minimum set of blocks (a sampler, a quantizer, an encoder) and code length, as a rule, n = 16 (two bytes)). After the ADC, another encoder is activated, which digitally performs all the necessary operations (code conversion, code length reduction, compression, non-uniform quantization, etc.) (figure 2.8).

Figure 2.8 – Generalized scheme of transfer of analog signals by PCM method

Procedure for calculating ADC and DAC parameters at PCM. The order and sequence of ADC and DAC parameters are calculated and to a large extent it depends at the output data what parameters are calculated. But the parameters of the analog signal and the requirements for the accuracy of the analogue signal recovery at the DAC output must be mandatory.

Typical initial data for ADC and DAC calculations at PCM:

– maximum frequency of analog signal spectrum F_{max};

– average power of analog signal P_b or its maximum value b_{max};

– amplitude coefficient of analog signal K_A;

– allowable signal/(quantization noise) ratio $\rho_{q.all}$;

– allowable signal/noise ratio at the DAC output $\rho_{out\ all}$;

– average time interval between clicks T_{cl};

– in ADC, uniform or non-uniform quantization is performed.

***Required calculations*:**

– sampling frequency f_s and sampling interval T_s;

– number of quantization levels L, length of the binary code n, digital signal rate R;

– signal / (quantization noise) ratio ρ_q with selected ADC parameters;

– allowable symbol (bit) error probability p at DAC output;

– operating parameters of the interpolating DAC filter.

You need to decide whether to enable or disable LPF before ADC, the so-called prefilter? The answer is simple:

– if the spectrum of the analog signal is already limited (F_{max} is set) and further decrease of F_{max} is not permitted, then there is no need to enable LPF before ADC;

– if the spectrum of the analog signal is unlimited, then in order to reduce the value of F_{max} the signal is passed through the LPF. Typically, the pre filter and interpolating (restoring) LPF parameters in the DAC are chosen identically.

Examples of calculations are given below.

***Example 2.4*.** Calculate ADC parameters with uniform quantization based on such input data:

– maximum frequency of analog signal spectrum F_{max} = 14 kHz;

– maximum value of analog signal b_{max} = 1,5 V;

– amplitude coefficient of analog signal K_A^2 = 9,0 dB or $K_A^2 = 10^{0,9} = 7{,}94$;

– allowable signal/(quantization noise) ratio $\rho_{q.all}$ = 48 dB or $\rho_{q.all} = 10^{4,8} = 63096$.

Solving. 1. According to Kotelnikov's (sampling) theorem, the sampling frequency $f_s = 1/T_s$ must satisfy the condition (formula (2.9)) $f_s \geq 2F_{max}$. For the given F_{max} = 14,0 kHz we can choose f_s = 32 kHz (reserve on simple LPF using in the DAC 10…15%).

2. Let's set the allowable signal/(quantization noise) ratio $\rho_{q.all}$ from a formula (2.15) number of allowable quantization levels

$$L_{all} \geq \sqrt{K_A^2 \rho_{q.all} / 3} + 1 = \sqrt{7{,}94 \cdot 63096 / 3} + 1 = 410.$$

It is chosen to implement ADC the number of quantization levels as the closest value from a number of powers of number 2 (2, 4, 8, 16, 32, 64, 128, 256, 512 and so on), in our example L = 512. Then, by the formula (2.12), the length (digit) of the ADC code $n = \log_2 512 = 9$.

3. Quantization step by the formula (2.10)

$$\Delta b = 2 \cdot 1{,}5/511 = 5{,}87 \cdot 10^{-3} \text{ V}.$$

4. Calculating and selecting values f_s, L and n, other ADC parameters will be as follows:

– signal/(quantization noise) ratio according to the formula (2.15)

$$\rho_q = 3 \cdot 511^2 / 7{,}94 = 98660 \text{ or } 49{,}9 \text{ dB};$$

– digital signal rate by the formula (2.13)

$$R = 9 \cdot 32 \cdot 10^3 = 288 \cdot 10^3 \text{ bps};$$

– power of quantization noise by the formula (2.14)

$$\overline{\varepsilon_q^2} = (5{,}87 \cdot 10^{-3})^2/12 = 2{,}87 \cdot 10^{-6} \text{ V}^2.$$

Example 2.5. Calculate the DAC parameters for such output data (some of which are obtained during ADC calculations in Example 2.4) and the requirements for the digital communication channel:

– maximum frequency of analog signal spectrum F_{max} = 14 kHz;

– sampling frequency f_s = 32 kHz;

– quantization step $\Delta b = 5{,}87 \cdot 10^{-3}$ V;

– signal/(quantization noise) ratio ρ_q = 49,9 dB or $\rho_q = 10^{4{,}99} = 98156$;

– power of quantization noise $\overline{\varepsilon_q^2} = 2{,}87 \cdot 10^{-6}$ V^2;

– allowable signal/noise ratio at the DAC output $\rho_{out\ all}$ = 45 dB or 31623.

Solving. 1. By the formula (2.16)

$$\overline{\varepsilon_{d\,ch}^2} = \overline{\varepsilon_q^2}\left(\rho_q/\rho_{out} - 1\right) = 2{,}87 \cdot 10^{-6} \cdot (98156/31623 - 1) = 6{,}04 \cdot 10^{-6} \text{ V}^2.$$

2. Power of the DAC output caused by errors in the digital communication channel depends on the probability of error p at the DAC input and is determined from the formula (2.17)

$$p_{all} = 3\overline{\varepsilon_{d\,ch}^2}/[(\Delta b)^2(4^n - 1)] = 3 \cdot 6{,}04 \cdot 10^{-6}/[(5{,}87 \cdot 10^{-3})^2 \cdot (4^9 - 1)] = 2 \cdot 10^{-6}.$$

3. The requirements for an interpolating LPF in the DAC are as follows:

– maximum frequency of passband $F_{pb} = F_{max}$ = 14 kHz with attenuation 3 dB;

– minimun frequency of stopband $F_{sb} = f_s - F_{max} = 32 - 14 = 18$ kHz with attenuation not less than 10…20 dB.

Remarks. 1. If it is specified that the errors in the channel are evaluated by clicks and given the average interval between clicks, then in the example 2.3 you can calculate the probability of a symbol error at the communication channel output.

2. Based on the data obtained, one can calculate the parameters of the realized interpolating LPF – analog passive or active.

Example 2.6. Calculate the parameters of the interpolating DAC filter for such initial data:

– maximum frequency of analog signal spectrum F_{max} = 3,4 kHz;

– sampling frequency f_s = 8,0 kHz;

– at maximum frequency of passband F_{pb} attenuation $A(F_{pb})$ = 3,0 dB;

– at minimun frequency of stopband F_{sb} attenuation not less than $A(F_{sb})$ = 15 dB;

– the Butterworth filter is used as the interpolating filter.

It is necessary to determine the order of the filter.

Solving.

1. Maximum frequency of passband $F_{pb} = F_{max}$ = 3,4 kHz.

2. Minimun frequency of stopband $F_{sb} = f_s - F_{max} = 8{,}0 - 3{,}4 = 4{,}6$ kHz.

3. Normalized AR of Butterworth filter is described by the expression

$$H(f) = \frac{1}{\sqrt{1+(f/F_{\text{p b}})^{2n}}}, \tag{2.18}$$

where n – order of the filter (integer positive number); F_{pb} – maximum frequency of passband at 3,0 dB.

4. The attenuation provided by the Butterworth filter at the frequency F_{sb},

$$A(F_{\text{sb}}) = 20\lg\frac{1}{H(F_{\text{sb}})} = 10\lg\left(1+\left(F_{\text{sb}}/F_{\text{pb}}\right)^{2n}\right). \tag{2.19}$$

5. The required order of the filter is determined by solving the equality (2.19) relative n

$$n \geq \frac{\lg\,(10^{0,1A(F_{\text{sb}})}-1)}{2\lg(F_{\text{sb}}/F_{\text{pb}})}. \tag{2.20}$$

An inequality sign appears, since n is an integer.

6. If we substitute value F_{sb} = 4,6 kHz, F_{pb} = 3,4 kHz, $A(F_{\text{sb}})$= 15 dB for n in the resulting formula, we will get

$$n \geq \frac{\lg\,(10^{1,5}-1)}{2\lg(4,6/3,4)} = 5,66, \text{ i.e. } n = 6.$$

At n = 6 Butterworth filter is quite complex, therefore, it is necessary to use Chebyshev filter as an interpolating filter, which have a smaller order, or to increase the sampling frequency to reduce the order of the filter.

Conclusion. PCM does not apply to the methods of low-rate coding of continuous messages and is used in systems where the task of economical use of frequency resources of communication channels is not set (for example, in cable and fiber-optic multichannel systems for the transmission of conversational signals), and the simplicity of the implementation of ADC and DAC prevails. Today, PCM is used also as a standard by which the quality indicators compared to all other methods of conversational messages transfered by digital communications channels.

2.11 Coding of analog signals with prediction

Samples of real analog signal are correlated. This fact allows to predict with any accuracy the value of the next sample of a signal basing on values of the previous samples. In the coder of transmission system with a prediction (figure 2.9) the error of a prediction is calculated

$$d(kT_{\text{s}}) = b(kT_{\text{s}}) - \tilde{b}(kT_{\text{s}}), \tag{2.21}$$

where $b(kT_{\text{s}})$ – the sample of an analog signal from sampler;

$\tilde{b}(kT_{\text{s}})$ – the predicted sample generated by the predictor on N basis of the previous samples $b((k-1)T_{\text{s}}),\ b((k-2)T_{\text{s}}),\ \cdots b((k-N)T_{\text{s}})$.

The error of a prediction is transferred to a communication channel by a digital signal. Therefore in the circuit of the coder of system (figure 2.9) the quantizer and the coder of the prediction of error samples are available. The decoder of samples recovers samples of a prediction error; in the system decoder there is precisely the same predictor, as in the system coder; the predicted samples are developed with the transferred sample of a prediction error and, thus, the sample of a transmitted analog signal is recovered.

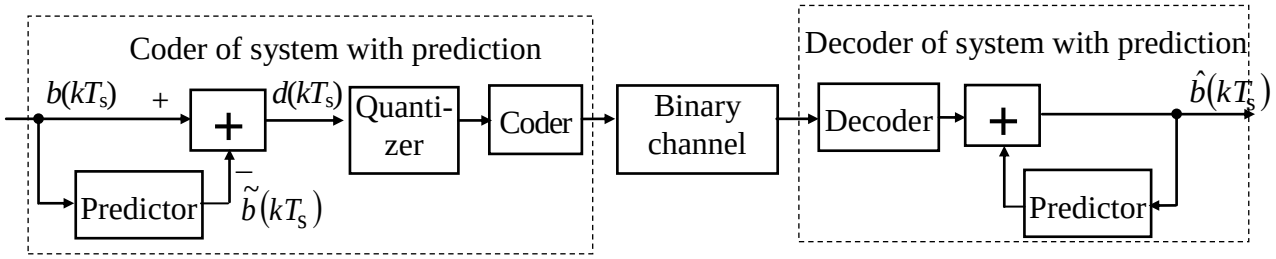


Figure 2.9 – Coder and decoder of system with prediction

Peak-to-peak of a discrete signal $d(kT_s)$ is smaller, than peak-to-peak of a signal $b(kT_s)$, therefore the number of quantization levels L at a constant quantization step Δd will be smaller than the transmission of samples $b(kT_s)$ by PCM method. Reduction of quantization levels number reduces the length of a code n and a digital signal rate $R = n \cdot f_s$. Or, at constant number of quantization levels L the step of quantization decreases $\Delta d = (d_{max} - d_{min}) / L$, the power of quantization noise decreases $\overline{\varepsilon_q^2} = \Delta d^2/12$, the ratio signal/quantization noise grows ρ_q.

2.12 Methods of differential PCM

In different variants of DPCM method the number of samples N on the basis of which predicted sample is defined. It is in limits from 1 up to 6. Predictor at $N \geq 2$ is carried out under the circuit of not recursive filter. At case $N = 1$ predicting sample $\tilde{b}(kT_s)$ is the previous sample $b((k-1)T_s)$.

Circuits of the DPCM coder and decoder, used in the real equipment, are resulted in figure 2.10. In the coder prediction an error acts on quantizer, similar quantizer systems with PCM, then the quantized error $d_q(kT_s)$ is transferred by a digital signal over a communication channel. Predictors in the coder and the decoder are completely identical.

As opposed to circuit resulted in figure 2.9, the predictor in the coder is included in a loop of a feedback. Therefore predicting samples $\tilde{b}(kT_s)$ in the circuit of the coder and in the circuit of the decoder are developed from the same samples $\hat{b}(kT_s)$ (if there were no errors in a channel at transferring).

In the decoder the predictor is included in circuits of a feedback and consequently at decoding quantization noise can be collected. Let's define a quantization error at DPCM according to the circuit in figure 2.10

$$\varepsilon_q(kT_s) = \hat{b}(kT_s) - b(kT_s) = [\tilde{b}(kT_s) + d_q(kT_s)] - [\tilde{b}(kT_s) + d(kT_s)] = d_q(kT_s) - d(kT_s). \quad (2.22)$$

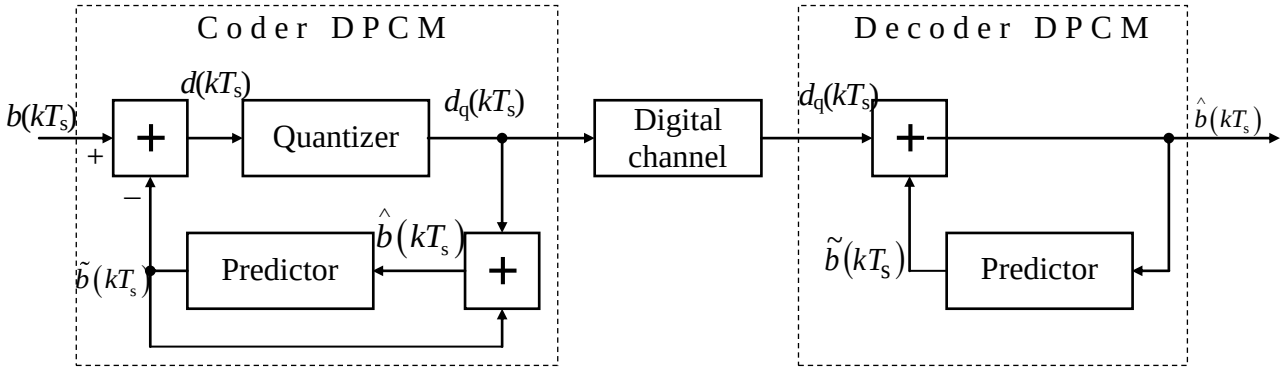


Figure 2.10 – Coder and decoder of system with DPCM

From the last ratio it is visible, that, due to inclusion of the predictor in the coder in a circuit of a feedback, the quantization error is defined only by quantizer parameters, and there is no effect of accumulation of quantization noise in the decoder.

2.13 Adaptive DPCM

Methods of adaptive DPCM (ADPCM) are widely applied. Adaptive parts of ADPCM coder are:

– predictor with $N = 4…6$ – its coefficients are automatically adjusted so that the dispersion of a signal $d(kT_s)$ was minimized, coefficients of the predictor are transferred over a communication channel. The predictors of coder and decoder have the same coefficients;

– quantizer – peak-to-peak of its characteristic (d_{max}, d_{min}) and accordingly a quantization step changes in accordance with peak-to-peak of a current signal $d(kT_s)$, data on a quantization step are transferred by a channel, so that the quantization step is same for the coder and the decoder.

2.14 Methods of delta modulation (DM)

Methods of DM are applied to transfering with a prediction. Methods of DM differ from PCM and DPCM because two-level quantizer ($L = 2$) is used. It becomes possible if the sampling frequency is greater, than $2F_{max}$. Then the adjacent samples from the sampler differ a little. In figure 2.11 circuits of coder and decoder, explaining one of DM methods, are presented.

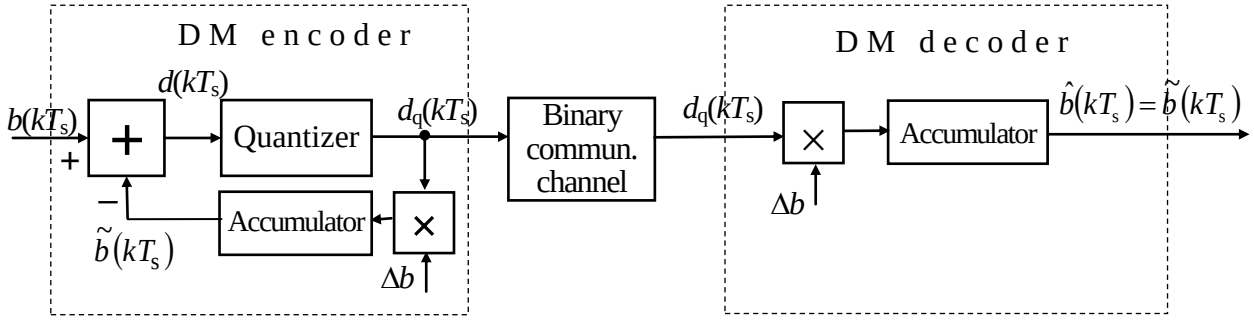


Figure 2.11 – DM encoder and decoder

The error of a prediction is calculated in the same way as for DPCM,

$$d(kT_s) = b(kT_s) - \tilde{b}(kT_s), \tag{2.23}$$

and the predicted samples are concentrated in the accumulator

$$\tilde{b}(kT_s) = \sum_{i=0}^{k-1} d_q(kT_s) \cdot \Delta b, \tag{2.24}$$

where Δb is coefficient;

$$d_q(kT_s) = \begin{cases} +1, & \text{if } d(kT_s) \geq 0, \\ -1, & \text{if } d(kT_s) < 0 \end{cases} - \tag{2.25}$$

The error of a prediction is quantized into two levels which are transferred by a binary communication channel.

The described method of coding is illustrated by time diagrams in figure 2.12. Here the predicted signal and a quantized prediction error signal are submitted by continuous time signals. It is visible, that the predicted signal $\tilde{b}(t)$ "traces" changes of an input signal. The essence of factor Δb follows from figure – it is a quantization step, as with this step a signal $\tilde{b}(t)$ quantizes. Two domains are visible in figure:

1) the domain where distortion of a slope overload is observed – the predicted signal $\tilde{b}(t)$ has not trace changes of an input signal;

2) the domain where subdivision noise is observed – at a constant input signal the predicted signal changes in peak-to-peak amplitude Δb.

Clearly, to reduce the first effect it is necessary to increase a step of quantization, and to reduce the second effect – to reduce a step of quantization. It is obvious, that there is an optimum step of quantization when the total effect from a slope overload and noise of subdivision at a signal $b(t)$ realizations is minimized.

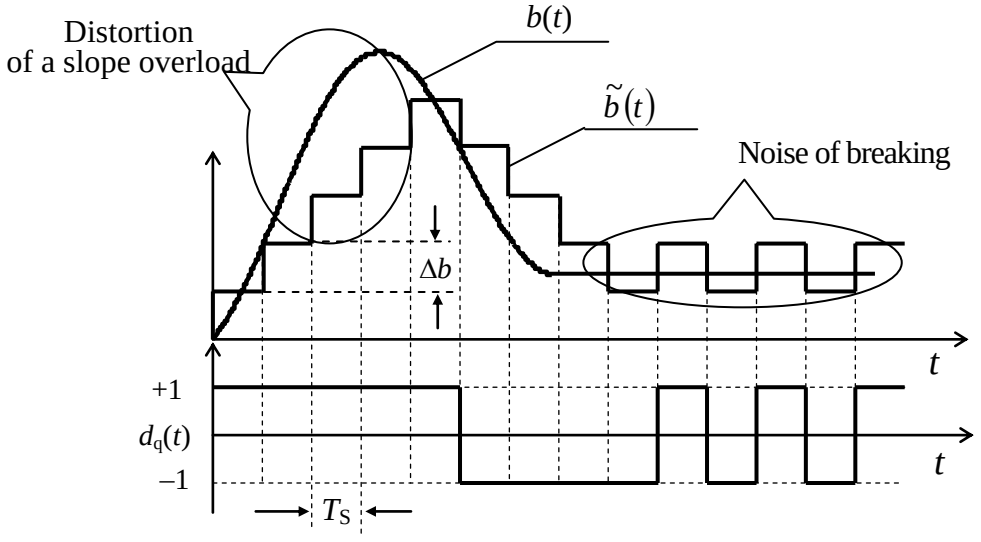


Figure 2.12 – Illustration of work of the DM coder

The work of the decoder of DM (figure 2.11) is calculate the predicted samples of the signal by the formula for $\widetilde{b}(kT_s)$.

It is possible to formulate features of DM methods:

– the sampling frequency f_s in some cases is more, than $2F_{max}$;
– as quantizer is two-level, so the code has length $n = 1$, and $R = f_s$;
– as $n = 1$, so necessity of the decoder synchronization disappears.

2.15 Adaptive DM (ADM)

At adaptive delta modulation (ADM) the quantization step can change. It is carried out as follows. At a coder output the analyzer of sequence of binary symbols is connected. If there was a sequence 111 or 000, the step of quantization increases to reduce distortions of a slope overload. If there was a sequence 101 or 010, the step of quantization decreases to reduce distortions from noise of subdivision.

The similar analyzer is connected at a decoder input and the step of quantization in the decoder changes in the same way.

2.16 Conclusion

The effective encoding of continuous signals is removed redundancy which is defined by statistical connection between samples (by correlation of samples). The effective encoding of continuous signals is based on the followings methods:

- static images compression (photo, fax) is subsampling (diminishing of frequency of sampling of color components), discrete Fourier cosine-transform, JPEG, GIF algorithms and other, vectorial compression (all curves on image are described by mathematical expressions);

- movable images compression (video) uses methods of prediction (in the first approaching - not samples, but the difference between them is transferred), subsampling, discrete Fourier cosine-transform, methods of motion compensation (information about direction of object element moves is formed), MPEG-1(2, 4) algorithms. These methods of encoding were succeessful in reducing television signal digital stream rate from 200 Mbit/s to 1,5 - 25 Mbit/s;

- voice message compression (telephony) uses methods of prediction - CELP algorithm, vocoders (design of human vocal organs). If at telephone signals transmission using a standard PCM method is used in the digital channel where $R = 64$ kbit/s, then the effective voice encoding methods allow to form the digital streams of 9,6 kbit/s (and even 4,8 kbit/s);

- musical message compression (sound broadcasting and television) is a method of the subband encoding (the signal spectrum is divided into bands, and the values of spectral density of different bands are quantized with the different quantization step) - MUSICAM algorithm. Higher class sound broadcasting signals and digitized records on disks by a PCM method require $R = 700$ kbit/s on one monophonic channel; the sound broadcasting signals effective encoding reduces rate to 100 kbit/s.

During last ten years there is rapid adoption of digital methods of television and sound broadcasting signals transmission. The main advantage is a high carrying capacity of the transmission systems, expressed by the number of programs.

At the compression of continuous messages converted into digital signals, it is possible to use the methods of discrete messages compression. However, at the continuous messages compression (video, photo, voice and music) the loss of information is admitted or, more exactly, the exception of unimportant information is admitted. It is possible because of human can sense of these messages types of his sense-organs are. In this case the degree of compression is limited to only the required quality of messages transmission.

Questions for section 2

2. 1. Why should the signs of a discrete source be encoded before they are transmitted or memorized?

2. 2. How are the main parameters of the code determined: the size (base) of the secondary alphabet, the length (bit) of the code?

2. 3. How many digits should be in the uniform code intended for encoding of the initial alphabet consisting of 128 symbols, with code m = 2, 4, 8, 16, 32.

2. 4. Explain the principle of effective coding.

2. 5. How to determine the coefficients of compression and efficiency of the nonuniform code?

2. 6. What purpose are digital methods used for transmitting analog signals?

2. 7. List the main advantages of digital continuous transmission systems compared to analog transmission methods.

2. 8. Give definitions: ADC, DAC, quantization noise, noise caused by errors in the digital communication channel.

2. 9. Give definitions: sampling, quantization of samples, encoding of quantized samples.

2. 10. How are sampling intervals and sampling frequences determined?

2. 11. What is the quantization step and how is it selected?

2. 12. How to increase the signal-to-noise ratio of quantization?

2. 13. What are the advantages of encoding with companding?

2. 14. How is the rate of a digital signal encoded by the PCM method determined?

2. 15. Explain the principle of digital systems with prediction.

2. 16. Explain the DPCM encoding principle.

2. 17. How is the sampling interval or sampling rate determined when encoding analog signals using the DPCM method?

2. 18. What does the length of the code depend on when DPCM?

2. 19. How is the rate of a digital signal encoded by the DPCM method determined?

2. 20. What is the difference between DPCM and PCM encoding?

2. 21. What is ADPCM?

2. 22. Explain the encoding by the DM method.

2. 23. How is the sampling interval and sample rate at DM determined?

2. 24. Explain the principle of operation, advantages and disadvantages of transmission with delta modulation.

2. 25. Explain the principle of operation and advantages of transmission with adaptive delta modulation.

2. 26. What is the difference between transmission systems using DPCM and DM?

2. 27. What is the distortion of slope overload? How to reduce it?

2. 28. What is the noise of subdivision? How to reduce it?

2. 29. How is the sampling interval and sampling rate in DM determined?

2. 30. Explain the principle of operation, advantages and disadvantages of transmission with delta modulation.

2. 31. Explain the principle of operation and advantages of transmission with adaptive delta modulation.

2. 32. What is the difference between transmission systems using DPCM and DM?

Tasks

2.1. Three codes are given

Number of code	Symbols							
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
1	000	001	010	011	100	101	110	111
2	0	1	00	01	10	11	110	111
3	00	01	100	101	1100	1101	1110	1111

It is necessary to: 1) determine the type of code; 2) determine which of these codes are prefixed and why; 3) encode the sequence $a_6a_0a_3a_3a_7a_2a_1$ with all three codes; 4) decode the binary sequence: a) code 1 – 111010011101000110; B) code 3 – 1111001011110.

2.2. Coded number 47 with natural codes with the base m = 2, 4, 8, 10. How does the number of digits of code words for different code bases change?

2.3. The volume of the source alphabet M_A = 18 symbols, the length of one symbol issue T_{sym} = 0,01 s. Determine the length (bit) of the uniform code and the speed of the digital signal (bps).

2.4. The source issues the message (symbols) with binary symbols of the uniform code: the length of the code is 8 digits, the duration of issuing one sign T_{sym} = 0,1 ms. Determine the maximum possible number of code words that can be encoded by this code, the duration of one symbol issue and the rate of the digital signal (bps).

2.5. Probability of messages (signs) of the source are: $P(a_1)$ = 0,04; $P(a_2)$ = 0,50; $P(a_3)$ = 0,10; $P(a_4)$ = 0,20; $P(a_5)$ = 0,06; $P(a_5)$ = 0,10. Construct the optimal non-uniform binary Huffman or Shannon-Fano codes. Calculate the average length of the code word and specify how much it is less than the length of the uniform code.

2.6. Decode sequence 01101011111, which was encoded by Huffman or Shannon-Fano code: a_1 – 11111; a_2 – 0; a_3 – 110; a_4 – 10; a_5 – 11110; a_6 – 1110.

2.7. Required signal/(quantization noise) ratio ρ_q = 50 dB for ADC with uniform quantization, designed to convert speech signal to digital. Determine the speed of the digital signal.

2.8. Length of the ADC code with uniform quantization was reduced by two digits. Determine how this changes the quantum signal/noise ratio at the output.

2.9. It is stated that when converting a continuous signal into digital using ADC with uniform quantization, eight levels of quantization were used. Determine the quantization signal/noise ratio ρ_q.

2.10. As a result of the sampling of the analog signal, the following sequence of readings is obtained: 0,28; 0,52; 1,23; 0,47; 0,02; –0,42; –0,96 V. Convert this sequence into a PCM signal in a symmetric code if the step of quantization Δb = 0,1 V.

2.11. It is stated that the ADC applied 8-bit code and the sampling frequency f_s = 16,0 kHz. Calculate the rate of the digital signal R at the ADC output.

2.12. Using compilation, they reduced the amplitude of the analog signal by 7 dB. How to change the signal/(quantization noise) ratio at the DAC output?

2.13. Determine the probability of an error at the DAC input to provide DAC output noise due to errors in the digital communication channel $\overline{\varepsilon^2_{d\,ch}} = 10^{-4}$ V^2 for the following ADC parameters: the number of quantization levels L = 256, the quantization step Δb = 0,015 V.

3 INFORMATION CHARACTERISTICS OF COMMUNICATION CHANNELS

3.1 Models of communication channels

Two models of communication channels are most distributed: a digital communication channel and a continuous communication channel.

The communication channel is called **digital** if it is intended for the transmission of a digital signal. Such channel is characterized by:

– the rate of a digital signal R (bit per second) which can be transmitted by a communication channel;

– the number of signal levels in a communication channel M;

– the symbol rate, symbol/s, which shows the quantity of channel symbols, transmitted over a communication channel per 1 second, $B = R/\log_2 M$;

– probabilities of symbol transitions $P(\hat{b}_j / b_k)$ for $k = 0, 1, 2, \ldots M-1$, $j = 0, 1, 2, \ldots M-1$, where b_k – a symbol at an input of the channel, $\hat{b}_j$ – symbol at an output of the channel.

The majority of digital communication channels are binary ($M = 2$), and for them it is carried out $P(\hat{b}_0 / b_1) = P(\hat{b}_1 / b_0) = p$, where p – the probability of a bit error in a communication channel. Such communication channel is called as a **binary symmetric communication channel** (BSC). It represent as the graph (figure 3.1). BSC is characterized by two parameters: B and p.

The digital communication channel is always constructed on the basis of the continuous channel. The communication channel is called **continuous** if it is intended for the transmission of a continuous signal. The most widespread model of the continuous channel is a Gaussian communication channel. Consider, that in such communication channel additive white Gaussian noise (AWGN) acts, i.e. the noise has Gaussian distribution of probabilities of instant values with zero average and a uniform spectrum in a passband of a communication channel. The model of a communication channel is shown in figure 3.2. – relationship between an output and an input is described by a ratio $z(t) = s(t) + n(t)$.

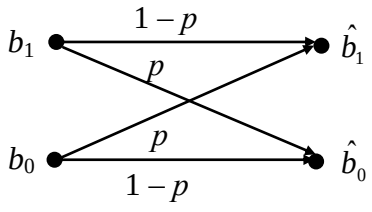


Figure 3.1 – Model of BSC

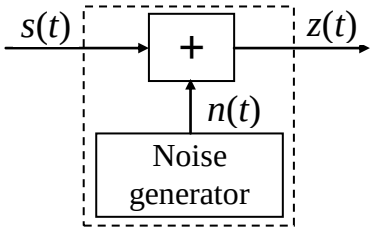


Figure 3.2 – Communication channel with AWGN

The Gaussian communication channel is described by the following parameters:

F_{ch} – passband of the channel;

P_s – average power of a signal at the channel output;

N_0 – power noise density at the channel output.

Consider, that noise takes place in a passband of a communication channel, and its average power at a channel output is defined as

$$P_n = N_0 F_{\text{ch}}. \tag{3.1}$$

3.2 Transmission rate of the information over a communication channel

Transmission rate of the information over a communication channel R_{ch} is defined as the average quantity of the mutual information between input and output of the channel per 1 second. The units of measure are bit/s (bps). In the information theory it is considered, that errors or noise acting in a communication channel result in a destruction of a part of the information.

Let's define transmission rate of the information in a digital communication channel. For this purpose we will take the concept of the mutual information of two sources of messages entered earlier. Let's consider here, that the output of one source is an input of a communication channel where the signal $b(t)$ operates, and the output of other source is an output of a communication channel where the signal $\hat{b}(t)$ operates. It is possible to write down

$$R_{\text{ch}} = \frac{I\left(B,\hat{B}\right)}{T_{\text{av}}} = \frac{H(B) - H\left(B/\hat{B}\right)}{T_{\text{av}}} = \frac{H\left(\hat{B}\right) - H\left(\hat{B}/B\right)}{T_{\text{av}}}, \tag{3.2}$$

where $H(B)$ – the entropy of a signal $b(t)$, describing average quantity of the information in one symbol at an input of a communication channel;

$H\left(B/\hat{B}\right)$ – the conditional entropy of a signal $b(t)$ provided that the signal $\hat{b}(t)$ is known, characterizes losses of the information in a communication channel (unreliability of the channel);

$H\left(\hat{B}\right)$ – the entropy of a signal $\hat{b}(t)$, describing average quantity of the information in one symbol at an output of a communication channel;

$H\left(\hat{B}/B\right)$ – the conditional entropy of a signal $\hat{b}(t)$ provided that the signal $b(t)$ is known, describing quantity of the extraneous information (generated by errors) at an output a communication channel;

T_{av} – the average duration of one symbol.

The similar approach is used for the definition of a transmission rate of the information in a continuous communication channel. At the input of a communication channel the signal $s(t)$ operates, and at the output of a communication channel the signal $z(t)$ operates. It is possible to write down

$$R_{\text{ch}} = \frac{h(S) - h(S/Z)}{T_s} = \frac{h(Z) - h(Z/S)}{T_s}, \tag{3.3}$$

где $h(S)$ – the differential entropy of a signal $s(t)$, describing average quantity of the information in one sample at an input of a communication channel;

$h(S/Z)$ – the conditional differential entropy of a signal $s(t)$ provided that the signal $z(t)$ is known. It characterizes losses of the information in a communication channel (unreliability of the channel);

$h(Z)$ – the differential entropy of a signal $z(t)$, describing average quantity of the information in one sample at an output of a communication channel;

$h(Z/S)$ – the conditional differential entropy of a signal $z(t)$ provided that the signal $s(t)$ is known, describing the quantity of the extraneous information (generated by noise) at an output of a communication channel;

T_s – sampling interval.

3.3 Capacity of binary symmetric communication channel

Communication channel capacity *C* is the greatest possible transmission rate of the information in communication channel at the set characteristics of a channel.

$$C = \max R_{\text{ch}}. \tag{3.4}$$

Use next formula to calculate the capacity of BSC

$$C = \max \frac{H(B) - H\left(B/\hat{B}\right)}{T_{\text{av}}}. \tag{3.5}$$

Symbols have duration $T_{\text{av}} = 1/B$ (since a communication channel is binary, then the symbol rate coincides with a digital signal rate).

Further it is necessary to search for the maximum of values: $\max\left[H(B) - H\left(B/\hat{B}\right)\right]$. The maximum of the first term takes place, when symbols at an input of the channel are independent and equiprobable: $\max\left[H(B)\right] = 1$ binary unit.

Let's calculate the second term

$$H\left(B/\hat{B}\right) = -\sum_{k=1}^{M}\sum_{j=1}^{M} P\left(b_k, \hat{b}_j\right)\log_2 P\left(b_k/\hat{b}_j\right). \tag{3.6}$$

After substitution of $M = 2$ and values of probabilities from figure 3.1 we shall receive

$$H\left(B/B\right) = -p\log_2 p - (1-p)\log_2(1-p). \tag{3.7}$$

Let's remind, that this conditional entropy reflects losses of the information in a communication channel.

Final expression looks like

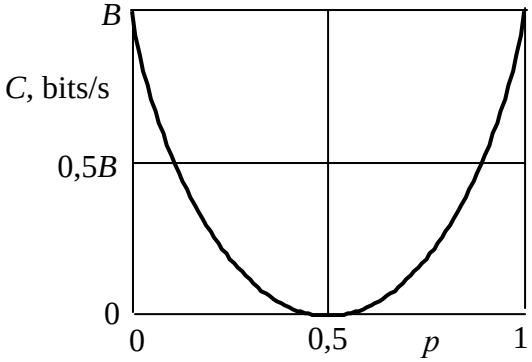


Figure 3.3 – Capacity of BSC

$$C = B\left[1 + p\log_2 p + (1-p)\log_2(1-p)\right].$$

Special cases:

$p = 0$ – losses of the information are equal to zero and $C = R$;

$p = 1$ – all symbols in a communication channel are inverted, there is no losses of the information and $C = R$;

$p = 0{,}5$ – losses of the information are equal to one and $C = 0$.

3.4 Capacity of communication channel with AWGN

To calculate a communication channel capacity with AWGN the next formula is used

$$C = \max \frac{h(Z) - h(Z/S)}{T_s}. \tag{3.8}$$

Let's search for a maximum of values: $\max[h(Z) - h(Z/S)]$. The maximum of the first term takes place, when the signal $z(t)$ has Gaussian probability distribution. As $z(t) = s(t) + n(t)$, and noise $n(t)$ has Gaussian probability distribution, than $z(t)$ will have Gaussian distribution, if a signal $s(t)$ has Gaussian distribution. If it is correct, then

$$h(Z) = \log_2 \sqrt{2\pi e(P_s + P_n)}. \tag{3.9}$$

Conditional differential entropy of a signal $z(t)$ (provided that the signal $s(t)$ is known) is equal to the differential entropy of a Gaussian noise $n(t)$

$$h(Z/S) = h(N) = \log_2 \sqrt{2\pi e P_n}. \tag{3.10}$$

Sampling interval $T_s = 1/(2F_{ch})$, thus samples of signals are independent.

Let's substitute the received values:

$$C = 2F_{ch}\left[\log_2 \sqrt{2\pi e(P_s + P_n)} - \log_2 \sqrt{2\pi e P_n}\right]. \tag{3.11}$$

After simple transformations we shall receive

$$C = F_{ch}\log_2\left(1 + \frac{P_s}{P_n}\right) = F_{ch}\log_2\left(1 + \frac{P_s}{N_0 F_{ch}}\right). \tag{3.12}$$

The ratio determining the capacity of a communication channel with AWGN, is known as Shannon formula. From Shannon formula it is visible, that the basic resources of a communication channel are the passband of a channel F_{ch} and the power of a signal P_s. If $F_{ch} \to 0$ or $P_s \to 0$, than $C \to 0$.

If a passband of a communication channel tends to infinity, the capacity of a Gaussian communication channel tends to final size

$$C_{F_{ch}\to\infty} = 1{,}443\frac{P_s}{N_0}. \tag{3.13}$$

3.5 Efficiency coefficients of a transmission system

To estimate how effectively the basic resources of transmission system are used, we apply the coefficient of frequency efficiency γ and the coefficient of power efficiency β, determined by ratios:

$$\gamma = R_{ch}/F_{ch}; \qquad \beta = R_{ch}/(P_s/N_0). \tag{3.14}$$

Coefficient γ is measured in units ((bit per second)/Hz) and shows, how many bits in a second are possible to transmit in a band of frequencies of 1 Hz with a considered method of transmission.

In a transmission system the exchange of frequency efficiency for power efficiency is possible at the fixed reliability of transmission and on the contrary. Such exchange for ideal transmission system is expressed by a ratio

$$\beta = \frac{\gamma}{\left(2^{\gamma} - 1\right)}. \qquad (3.15)$$

This ratio allows to find limit value of one efficiency coefficient at other fixed and is called **Shannon limit**. If to accept a degree of approach of parameters γ and β to limiting on Shannon as a measure of "perfection" of transmission system then it is possible to say, that the methods of coding of the channel developed for today (including modulation) are highly effective, and do not present significant reserves to increase the efficiency.

But the operating ratio of capacity of a communication channel is also used for the considered efficiency coefficients β and γ

$$\eta = R_{ch}/C. \qquad (3.16)$$

3.6 Shannon theorem for a communication channel with noise

The importance of Shannon's formula is connected with so well-known Shannon's theorem for a communication channel with noise: If a message source rate is less than the capacity of a communication channel ($R_s < C$), there is a way of the transmission, allowing to transfer all source messages as with any accuracy.

Hence, the noise in a communication channel doesn't limit the accuracy of message transfer, and the rate of the information transfer over a channel.

Questions for section 3

3.1. Why is real communication channel replaced by its mathematical model?

3.2. Provide the definition of concepts - the reliability of the channel, channel capacity.

3.3. In which cases is it important to know the bandwidth of the channel?

3.4. How does the bandwidth of the channel change from AWGN when expanding its bandwidth?

3.5. What do the following concepts mean - the breakdown of the channel, the entropy of the noise?

3.6. What explains the fact that the bandwidth of the DSC is maximal when the probability of the error is $p = 1$ (all symbols are false)?

3.7. What is the practical significance of the basic Shannon coding theorem in the defective channel? Is it possible, using the proof of this theorem, to construct real encoding and decoding schemes?

3.8. Formulate Shannon's theorem for digital channels without errors and with errors. Compare the contents and the essence of these theorems and the importance of their results for practice.

Tasks

3.1. The communication channel between points A and B is composed of a cascade connection of three partial channels with capacities: the first channel 1200 (binary unit)/s; second channel

1500 (binary unit)/s; third channel 2000 (binary unit)/s. Find the capacity of the composite communication channel.

3.2. The conditions are the same as in task 3.1, but the connection of three partial channels is parallel. Find the capacity of the composite communication channel.

3.3. Calculate the capacity of a binary symmetric channel without memory with a given probability of error $p = 0.001$ and a modulation rate $B = 600$ symb/s.

3.4. Consider the channel (almost not real), in which the noise power spectral density at a certain frequency band from f_1 to f_2 is zero ($f_1 \neq f_2$). What is its capacity equal to?

3.5. Calculate the capacity of Gaussian channel with a given bandwidth $F_{ch} = 4{,}0$ kHz and the average signal and noise power ratio $P_s/P_n = 33$ dB.

3.6. Calculate the capacity of Gaussian channel with an unlimited bandwidth and a noise power spectral density $N_0 = 10^{-9}$ V^2/Hz, which transmits a signal with a power $P_s = 0{,}015$ V^2.

3.7. Determine whether it is possible to transmit messages from a 1200 bps source in a high-quality over channel with a capacity of 1200 bps? Explain your answer.

3.8. Determine whether it is possible to transmit messages from a 1200 bps source in a high-quality over channel with a capacity of 1100 bps? Explain your answer.

4 PURPOSE, STRUCTURE AND CLASSIFICATION OF ERROR-CONTROL CODES

4.1 Error-control codes in transmission systems

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

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. The **Code** is the rule of forming code words. The importance 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 primary goal of any telecommunication system is the information transmition of given fidelity and rate. These requirements are in contradiction and the increase of information rate leads to decreasing the noise immunity and the reliability of the transmission. In accordance with well known Shannon's theorems, as is the considerable increase of information fidelity transfer desirable if a transmitting rate through channel R_{ch} does not exceed the channel capacity C, which is basically possible. It is reached by using the long enough error-control codes.

Error-control code is the code which allows to detect and/or correct **errors** arising from messages transition over communication channels. The typical block diagram of a digital transmission system with error-control code is presented in figure 4.1. A baseband signal $b(t)$ comes from any sources of digital signal or from source encoders which will be considered in section 2. Channel **encoder** and channel **decoder** (**codec** of error-control code) is shown in figure 4.1. A channel encoder inputs the redundancy into code words of an error-correcting code. A baseband signal $b_c(t)$ from the output of the channel encoder is transmitted by a channel formed by the modulator, the transmission line, and the demodulator. Since the signal at the input of the demodulator is distorted by interference, the demodulator restores the signal $\hat{b}_c(t)$ with errors.

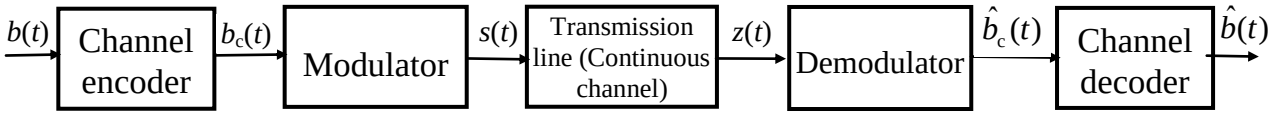


Figure 4.1 – Typical block diagram of a digital transmission system

Using the redundancy introduced during encoding, the channel decoder performs checks of codewords. Two decoding algorithms can be applied:

- if one or more errors are detected, the decoder sends a request to the encoder via the feedback channel to repeat the codeword; transmission system with ARQ decoding (ARQ – Automatic Request reQuest);
- if one or more errors are detected, the decoder attempts to correct errors by a certain algorithm; transmission system with FEC decoding (FEC – Forward Error Correction).

In real conditions the length of code words 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 in structure, construction principles and **error detection and correction capability**. In the subsequent sections the important classes of the codes with effective coding/decoding algorithms are considered.

4.2 Classification of error-control codes

The error-control codes can be classified in accordance to various features. The structure of codes classification is presented in figure 4.2. By a type of the error-control codes are subdivided on block and continuous codes formation. The formation of the **block codes** provides splitting of transferred digital sequences into separate blocks which are moved to encoder input. The block of code symbols which work is defined by the **coding algorithm** corresponds to each such block at an encoder output**.** The formation of the **continuous codes** is carried out continuously in time, without division into blocks as it can be seen from this code class name.

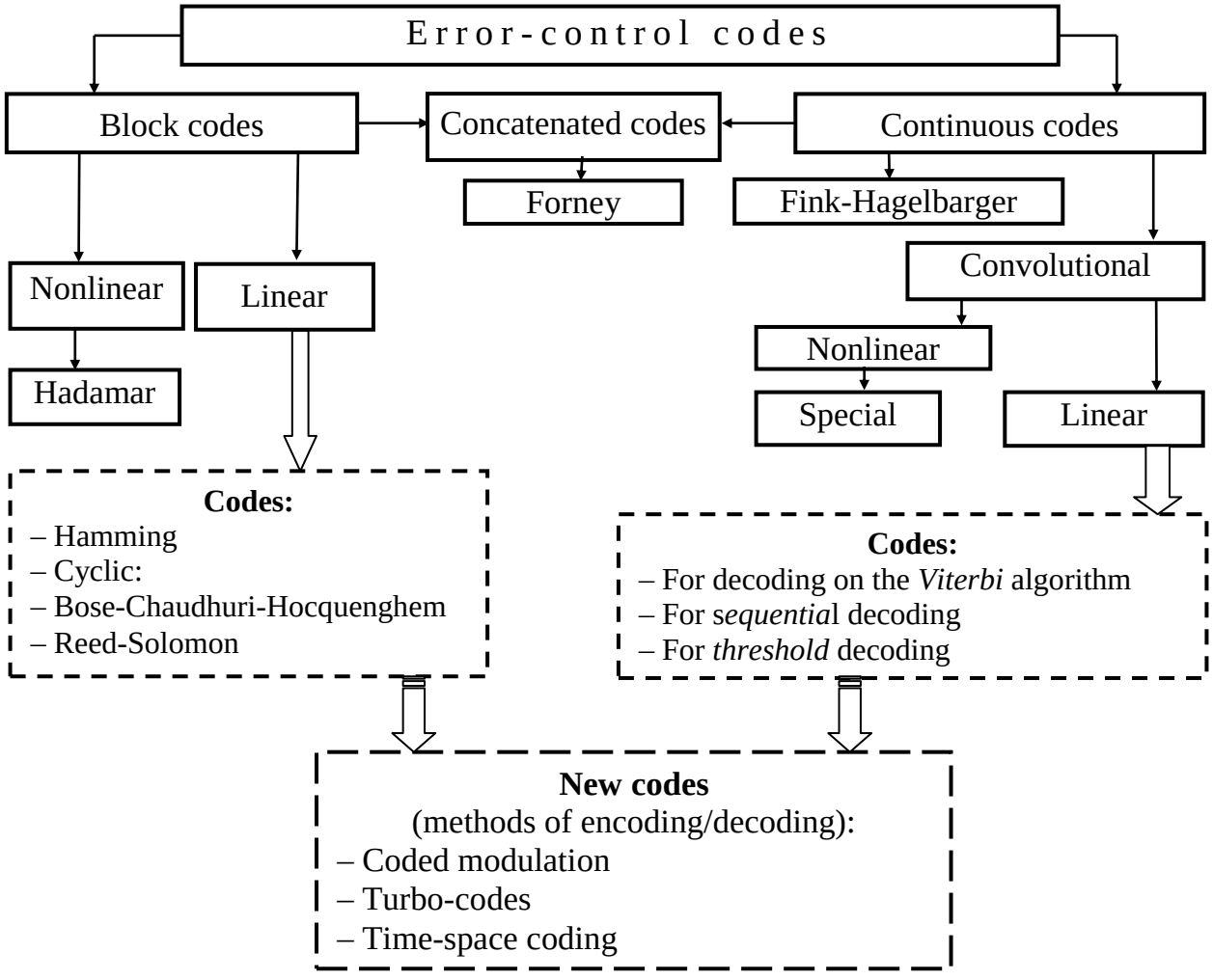


Figure 4.2– Structure of the code classification

Historically block codes where discovered and studied earlier, at beginning of the coding theory development. In a class of continuous codes it is necessary to note the **convolutional codes** which exceed in characteristics the block codes, and, for this reason, are widely applied in transmission systems.

To describe procedures of coding/decoding both block and convolutional codes usually use the corresponding mathematical apparatus. The well developed **linear algebra** is used to describe **linear codes**. The 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 the contrary the complexity of new decoding algorithms for convolutional codes (Viterbi algorithm, sequential decoding, threshold decoding) initiated searches of corresponding codes. Distinctive advantages of error-control codes forced searches for new approaches to find ways to increase noise immunity and efficiency of transmission systems. In figure 1.2 **new methods of encoding/decoding** (coded modulation, turbo-codes, time-space coding) are mentioned accordingly.

Questions for section 2

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

4.2. What elements does the codec of error-control code consist of?

4.3. What is the difference of coding procedures between block and continuous codes?

Tasks

4.1. Represent the block diagram of a digital transmission system and describe the purpose of its separate blocks.

4.2. Give the classification of error-control codes as for formation and structural properties.

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

5 BLOCK ERROR-CONTROL CODES

5.1 Parameters of block 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 in coding. In practice the codes with $m = 2$ are used. These are **binary codes**. To create a binary code word **binary alphabet** with symbols {0, 1} is used. Binary codes are widely practically implemented because of simplicity of binary logic elements construction in codec memory devices. The block code consists of a set of fixed length vectors named code words. The **code word length** is the number of elements in a code word and is denoted with n.

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

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

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 a **systematic code**. Each code word of length n symbols contains information symbols in a **systematic codes** k. Thus $r = n - k$ additional symbols are added to an information symbols which are dependent on information symbols and used in decoding for detection and correction of errors. In **nonsystematic codes** information symbols do not contain in a code word in an exact form.

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

$$M = m^n = 2^n. \tag{5.1}$$

These M code words are not completely used for an information transfer to have a possibility of detection and correction of errors. From these 2^n code words we may select $M_0 = 2^k$ code words ($k < n$) to form a code. Thus a 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 are named as a **permitted code word** as they are permitted for an information transfer. We call the resulting **block code** as an **(*n*, *k*) code**, and the ratio

$$R_{\text{code}} = \frac{k}{n} \tag{5.2}$$

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

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

$$R_{\text{code}} = (\log_2 M_0)/(\log_2 M). \tag{5.3}$$

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

$$R_{\text{code}} = 1. \tag{5.4}$$

The quantity of the permitted code words is equal to $M_0 = 2^k$.

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

$$K_{red} = 1 - R_{code} = (n - k)/n. \quad (5.5)$$

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

Exercise 5.1. As it is known, in a binary channels under the noises and distortions there are errors in form of transitions of transferred symbols to opposite symbols. For example, by transfering a symbol 1 transition (1→0) is possible and accordingly transitions (0→1) are also possible. Consider the possibilities of the binary error-control code construction intended for detecting the channel errors. Specify the encoding and decoding methods of such a code. For the developed algorithm of a coding define the rate and redundancy of such codes.

Instructions. The providing of errors detection in the transmitted code words will be possible if for the permitted code words forms will be given which are changed by errors in symbols of these words. Then the detection of errors (i.e. decoding) can be made by checking the conformity of received words to this previously known forms. When the first error-detecting codes were developed times of the the maintenance in the transmitted permitted words of «even number of 1s symbols» was considered to be simple. So the **«Code with even number of 1s»** has been invented.

Decision. Consider a construction variant of the binary systematic code with the code word length equals k. According to the 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 quantity of 1s in an information blocks can be both even and odd. It appears that it is convenient to use the procedure «**module-2 addition**» [2] for realization of encoding and decoding of such code words. This procedure defines the simple way to find the parity of 1s number in a code word. To every information block we will attribute one additional symbol ($r = 1$), so that the quantity of 1s in newly formed word would be **even**. The sequence of such an encoding is presented below:

1) Let information block be $\mathbf{a_1} = 101010$;

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

3) We form permitted code words, writinging an additional symbol at the end of block of information symbols $\mathbf{b_1} = 1010101$. It is clear, that the coding rule is carried out, since the number of 1s remains even;

4) By another information block $\mathbf{a}_2 = 101011$, $\boldsymbol{c} = 0$ and $\mathbf{b}_2 = 1010110$;

5) It is obvious, that any transition ((1→0) or (0→1)) changes the number of 1s in the words at a decoder input. It is possible to detect errors by calculating the 1s number in a decoding process.

Remark. It appears, such code allows detecting 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 number of 1s and by parameters (k, $r = 1$, $n = k+r = k + 1$) are defined by formulas:

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

It is visible, that for the information block $k >> 1$ of the big lengths the rate of such code is close to $R_{\text{code}} = 1$, and redundancy will be very small.

5.2 Error detection and correction capability of block codes

Let's establish dependence of detecting and correcting capability of the block codes from code parameters. It is useful to consider a binary code with parameters $n = 3$, $k = 2$. It is possible to divide all words of this code ($M = 8$) by sign «parity of 1s number in code words» into two groups:

– words with even number of 1s,

– words with odd number of 1s.

The code constructed by this principle is named “Code with even number of 1s” is considered in the Exercise 5.1.

Example 5.1. A binary code ($m = 2$, $n = 3$) with even number of 1s.

In table 5.1 the full set of binary words ($m = 2$, $n = 3$, $M = 8$) is divided into a set of permitted code words ($M_0 = 4$) containing words with even number of 1s (including the word 000 (number 0 – even)), and the set of the forbidden words with odd number of 1s. Their total quantity is equal to $M_{\text{forbid}} = M – M_0 = 4$.

Table 5.1 – Code with even number of 1s

<table>
<tr><td colspan="2">Full set of a words ($M = 8$):
{000, 001, 010, 011, 100, 101, 110, 111}</td></tr>
<tr><td>The permitted code words
(with even number of 1s),
M_0=4:
{000, 011, 101, 110}</td><td>The forbidden words
(with odd number of 1s),
$M_{\text{forbid}} = M – M_0 = 4$:
{001, 010, 100, 111}</td></tr>
<tr><td colspan="2">Code parameters: code rate R_{code} =2/3,
code distance $d_{\min} = 2$,
code can detect $q_{\text{det}} = 1$ error</td></tr>
</table>

Permitted code words are used for an information transfer through a channel (are permitted for transfer).

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

In the coding theory the concept of «distance between code words» plays an important role. Every binary block error-control code is 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 between the permitted code words. Let consider pairs of permitted code words from table 5.1. It is possible to establish that for this code a code distance is $d_{\min} = 2$. Such distance allows you to detect unitary errors in the channel. If the

transmitted code word is **b** = (1 1 0), and the channel error is characterized by a word (**error vector**) **e** = (0 1 0), the word at the decoder input $\hat{\mathbf{b}}$ with an error at the channel output 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 an error vector **e** that changes a corresponding symbol in a transmitted word **b** to an opposite symbol.

For the characteristic of quantity of channel errors the concept the multiplicity of errors is introduced. **Multiplicity of error** q is a quantity of the channel errors within a codeword. For example, for words from table 5.1 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 correct errors depends on code distance $d_{\min}$.

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

Error correction is the detection of errors in a certain symbols of a word $\hat{\mathbf{b}}$ and their subsequent correction by decoding.

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

1. **Error-detecting codes** allow to detect channel errors at decoding process.

2. **Error-control codes** allow to correct channel errors at decoding process and named in literature as codes with forward errors correction (i.e. with errors correction by code methods).

We will establish on an example of code with even number of 1s (see table 5.1) the relation between code distance $d_{\min}$ and error control ability of a code. It is convenient to use a geometrical representation of code words in figure 5.1. Let's represent code words by set from three symbols (x, y, z) and values of these symbols will be chosen 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. In figure 5.1 these tops are marked as follows:

– the permitted code words are marked by the sign "•";

– the forbidden code words are marked by the sign "✕".

It is visible, that code structure is between the permitted code words are forbidden words. They form the «**protective interval**». Therefore, the any unitary error translates any permitted word to the forbidden words nearest to them. This property leads to such a **decoding rule** of a code with even number of 1s and detection of any unitary error: reception from the channel output of the forbidden code words allows assert that there was any unitary error in the channel. It is easy to make conclusion that this code does not allow detecting double errors (because «protective interval» is nonsufficient). By induction it is possible to prove, that any binary code with even number of 1s allows to detect any error if their multiplicity is odd, and does not detect any errors if their multiplicity is even. The concept of «a protective interval» is easily

applied for a study of the relation between a code distance and code ability to correct errors. If the minimum distance between permitted code words (code distance) is d_{min}, that as is shown from figure 5.2 the protective interval contains (d_{min} - 1) forbidden words and it is necessary to make (d_{min} – 1) "steps" to "transfer" each permitted word to the nearest permitted word. Clearly, that all the errors with multiplicity q=1, 2, 3, ..., (d_{min} – 1) can be detected.

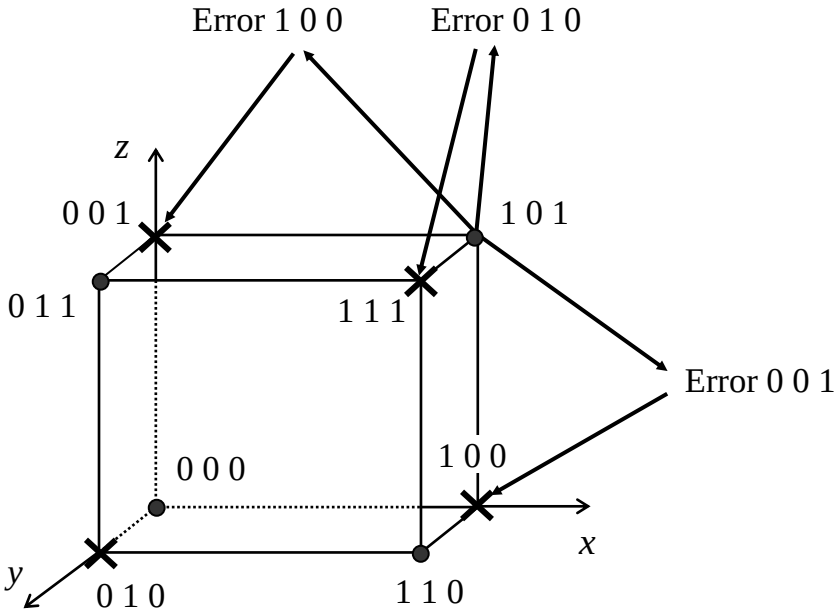


Figure 5.1 – Illustration of a code correction ability

From the above said it follows, that if code distance of a binary code is d_{min}, then the **code ability to detect errors** with multiplicity q_{det} is defined as:

$$q_{det} \leq (d_{min} - 1) \tag{5.6}$$

Let's take advantage of the similar representation to estimate the ability to correct errors. In figure 5.3 shows the layout of the permitted code words $\mathbf{b}_{allow.1}$ and $\mathbf{b}_{allow.2}$. Between them the forbidden words are allocated (d_{min} – 1). Let's divide all set of words into two permitted subsets as it is shown in figure 5.3. If, for example, the word $\hat{\mathbf{b}}$ is allocated into «permitted decoding subset of a word $\mathbf{b}_{allow.1}$» then during the decoding becomes decision about transmitting of the word $\mathbf{b}_{allow.1}$, so the error transitions of word $\mathbf{b}_{allow.1}$ to the nearest forbidden words are corrected. It is similarly possible to explain error control process by the transmitting the word $\mathbf{b}_{allow.2}$. It is visible, that the distance of each permitted subset is (d_{min} – 1)/2 (by d_{min} is odd). It defines the error-control code correction ability. For even values d_{min} the distance of each permitted subset is [(d_{min}/2) – 1], that also defines the error-control code correction ability.

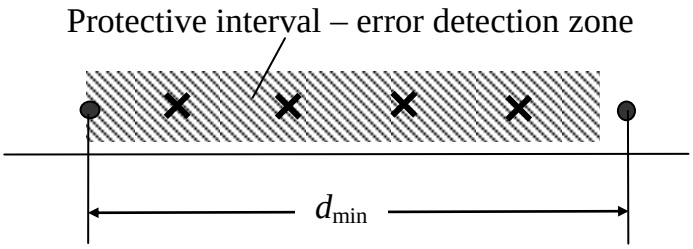


Figure 5.2 – Error detecting ability

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

$$q_{\text{corr}} \le \frac{d_{\min} - 1}{2}, \text{ if } d_{\min} \text{ is odd and } q_{\text{corr}} \le \frac{d_{\min}}{2} - 1, \text{ if } d_{\min} \text{ is even.} \qquad (5.7)$$

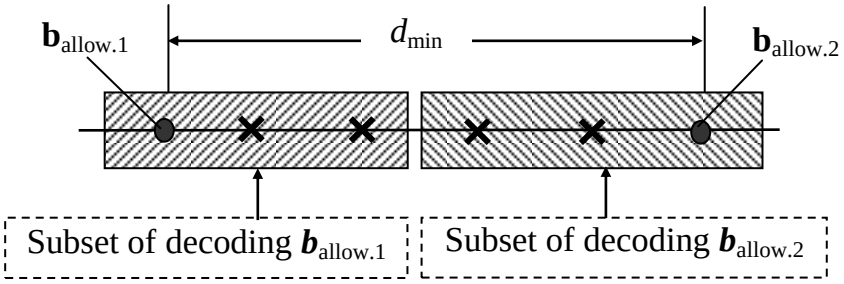


Figure 5.3 – Error correcting ability

5.3 Coding and decoding of a block code

We will use a mathematical apparatus of the general algebra to describe the linear block codes [2]. The concept of a **generator matrix** lies in the center of the block coding theory

$$\mathbf{G} = \begin{bmatrix} g_{01} & g_{02} & \cdots & g_{0,n} \\ g_{11} & g_{12} & \cdots & g_{1,n} \\ \vdots & \vdots & & \vdots \\ g_{k,1} & g_{k,2} & \cdots & g_{k,n} \end{bmatrix}. \qquad (5.8)$$

The **coding rule** of a block code is defined by the **product**

$$\mathbf{b} = \mathbf{aG}, \qquad (5.9)$$

where $\mathbf{a} = [a_1, a_2, ..., a_k]$ – row-matrix of informational symbols at encoder input;

$\mathbf{b} = [b_1, b_2, ..., b_n]$ – row-matrix of a block code word at encoder output;

G – generator matrix of linear (n, k) code.

Example 5.2. The encoder of code (7,4).

The encoder structure of a systematic code (7,4) is defined by generator matrix

$$\mathbf{G} = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{vmatrix} \qquad (5.10)$$

and coding rule (5.9). The row of informational symbols is $\boldsymbol{a} = (a_1, a_2, a_3, a_4)$ if at encoder input then the symbols of a permitted code word at its output $\boldsymbol{b} = (b_1, b_2, b_3, b_4, b_5, b_6, b_7)$ are defined by following equalities:

$$b_1 = a_1,\ b_2 = a_2,\ b_3 = a_3,\ b_4 = a_4,$$

$$b_5 = a_1 \oplus a_2 \oplus a_3 \oplus a_4,\ b_6 = a_1 \oplus a_2 \oplus a_4,\ b_7 = a_1 \oplus a_3 \oplus a_4. \qquad (5.11)$$

Figure 5.4 the diagram of a systematic code (7,4) shows the encoder with equalities (5.11). The coder operation algorithm:

1. k information bits in a parallel code or in a series code (the latter needs a shift register) at the coder input.

2. The checking symbols are calculated by $r = n – k$ adders.

3. k information bits and r checking symbols in a parallel or series code (the latter needs the converter of parallel in series code) at the coder output.

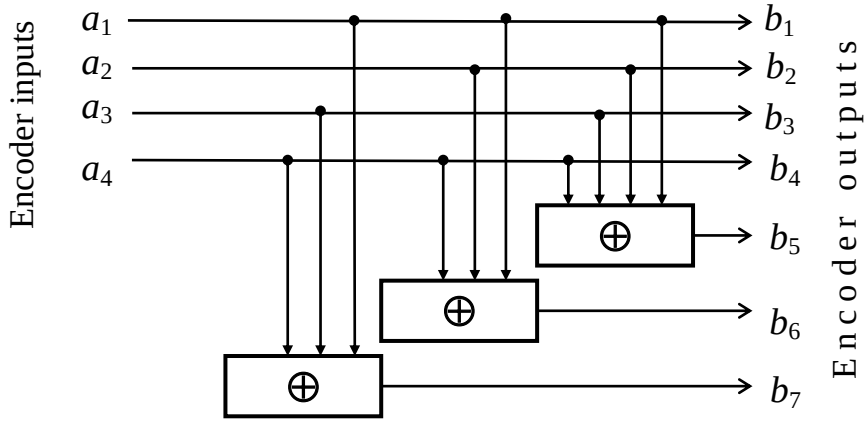


Figure 5.4 – Encoder diagram of systematic code

In general, the **G** matrix rows must satisfy the folowing conditions [1].

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

2. Every row must contain no less then $d_{\min}$ 1s.

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

In general, the generator matrix **G** can easily be constructed as the combination of the identity matrix **I** of order k (information matrix) and the additional matrix **P** of size $k \times r$.

$$\mathbf{G} = (\mathbf{I}|\mathbf{P}) = \begin{pmatrix} 1\ 0\ldots0\ \ g_{1,k+1}\ldots g_{1,n} \\ 0\ 1\ldots0\ \ g_{2,k+1}\ldots g_{2,n} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ 0\ 0\ldots1\ \ g_{k,k+1}\ldots g_{k,n} \end{pmatrix}. \qquad (5.12)$$

With such a construction of the matrix **G**, condition (3) is satisfied because the first k columns form an identity matrix. So, when constructing an additional matrix, we must satisfy conditions 1 and 2. If the code distance is $d_{\min} = 3$, then conditions 1 and 2 for the rows of the additional matrix can be formulated as follows:

1. The distance between any pair of rows must be not less than 1 (i.e. enough for the rows to be different).

2. Each line must contain at least two 1s.

Since these conditions are easily met, then for a code (7, 4) with $d_{\min} = 3$ the generator matrix has the form (5.10).

It can be shown that the rows of the generator matrix, as well as their linear combinations, are permitted codewords.

The concept weight of a code word plays an important role in the block codes theory. **Hamming weight w_H** of the binary code word is equal to a number of 1s in a code word.

Example 5.3. An evaluation of Hamming weights of code words.

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

Table 5.2 – 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

The structure of a generator matrix allows to define the code distance of the block codes. This position is illustrated by following exercises.

Exercise 5.2. Definition of the code distance by its generator matrix.

Generator matrix of the error-control codes by (5.10) is set. Show how to define code distance of a code by known generator matrix.

Instruction. To apply the method for code distance definition it is necessary to consider that zero combination $\mathbf{b_0}$ = (0000000) is also permitted.

Decision. It is noticed above, that permitted code words are defined by linear combinations of a generator matrix rows. As zero word $\mathbf{b_0}$ = (0000000) is also permitted, and rows of generator matrix **g1**, **g2**, **g3**, **g4** also are the permitted words, then Hamming distances from these words to a zero word $\mathbf{b_0}$ are defined by 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 follows from the above said.

The code distance as the value of a minimum distance between permitted code words is defined by the **least weight of rows** of a generator matrix.

Example 5.4. 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 5.3 (table 5.2). The result evaluations are shown in table 5.3.

Table 5.3 – The Hamming weights of generator matrix rows for systematic code

The generator matrix rows								Weights w_H(**gi**)
g1	1	0	0	0	1	1	0	3
g2	0	1	0	0	1	0	1	3
g3	0	0	1	0	0	1	1	3
g4	0	0	0	1	1	1	1	4

The analysis of data let us define the code distance of a systematic code from table 5.3 d_{min} = min{w_H (**gi**)} = 3.

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 code distance of block codes and weights of nonzero rows can be used for forming of a generator matrix of a block code with the beforehand define a code distance. This is illustrated by results of an Example 5.5.

Example 5.5. A generator matrix of a code with even number of 1s.

Let's form the generator matrix of the systematic (n, k) code which detects a unitary errors ($q_{\text{det}} = 1$). Such code should have a code distance $d_{\min} = q_{\text{det}} + 1 = 2$. Hence, the nonzero rows of a generator matrix of this code should have minimum weight $w_{\text{H}} = 2$. According to the standard form (5.12) each row of a systematic code matrix already contains a numeral 1 (defined by the submatrix $\mathbf{I}_k$), the weight should increase the weight of every row to 2 having added a numeral 1 (as a part of submatrix **P**) to the last numerals of every row.

For example, the generator matrix of such (5,4) codes with $k = 4$ will look like

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

The code word $\hat{\mathbf{b}}$ enters the decoder input from a communication channel:

$$\hat{\mathbf{b}} = \mathbf{b} \oplus \mathbf{e}, \qquad (5.14)$$

where $\mathbf{e} = [e_1, e_2, ..., e_n]$ – row-matrix of errors (error vector). For example, $\mathbf{e} = [0, 0, 1, 0, 0, 0, 0]$ – with the length of the code $n = 7$. The third symbol of the codeword in the communication channel was changed to the opposite (an error occurred).

By decoding the block codes the **check relations** are established with use of the parity **check matrix H**:

$$\mathbf{G} \cdot \mathbf{H}^{\text{T}} = 0. \qquad (5.15)$$

Here T – an index of the transposition.

If generator matrix is set in the form (5.12), the **check matrix** should look like:

$$\mathbf{H} = (\mathbf{P}^{\text{T}} \,|\, \mathbf{I}_{n-k}), \qquad (5.16)$$

where $\mathbf{P}^{\text{T}}$– transposed submatrix **P** of a generator matrix **G**;

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

Exercise 5.3. Find the check matrix of a systematic code (7,4).

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

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

Solution. We discover the submatrices entering the formula (5.16) further. The transposed submatrix by size is $(n - k) \times k$:

$$\mathbf{P}^{\text{T}} = \begin{vmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{vmatrix}.$$

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 combine submatrices in the uniform check matrix of a code:

$$\mathbf{H}=\begin{vmatrix}0 & 1 & 1 & 1 & 1 & 0 & 0\\ 1 & 1 & 0 & 1 & 0 & 1 & 0\\ 1 & 0 & 1 & 1 & 0 & 0 & 1\end{vmatrix}. \qquad (5.17)$$

From orthogonality condition of generating and parity check matrices of the linear code (5.15) it follows that each permitted word of linear code is generated by the rule **b** = **a**·**G** and also satisfies the orthogonality condition:

$$\mathbf{b}\cdot\mathbf{H}^{\mathrm{T}}=\mathbf{a}\cdot\mathbf{G}\cdot\mathbf{H}^{\mathrm{T}}=\mathbf{0}. \qquad (5.18)$$

By decoding we can calculate a **syndrome vector**

$$\mathbf{S}=\hat{\mathbf{b}}\cdot\mathbf{H}^{\mathrm{T}}=(s_1, s_2,..., s_{n-k}). \qquad (5.19)$$

The syndrome depends only on an error vector:

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

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

$$\mathbf{S}=\mathbf{e}\cdot\mathbf{H}^{\mathrm{T}}. \qquad (5.20)$$

From here the simple rule of the error detection follows:

1. If the **syndrome S = 0,** then an error vector **e** = **0**, i.e. the received word belongs to the set of the permitted code words and in the channel there were **no errors** or errors in the channel that converted the transmitted codeword to another permitted codeword.

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

This principle is underlied in the syndrome decoding. We will consider the principle of syndrome decoding on an example of a simple block code.

Example 5.6. The syndrome decoder of a systematic code (7, 4).

According to the rule (5.19) it is necessary to form the **transposed check matrix** of a code (7, 4) to realize of the syndrome decoder. The check matrix of this code looks like (5.17). A rule of a transposition of matrices is received as the result.

$$\mathbf{H} = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}; \qquad \mathbf{H}^{\mathrm{T}} = \begin{vmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}. \tag{5.21}$$

It is convenient noting a set of unitary errors in the transmission channel so:

$$\mathbf{e}_1 = (100\ldots0),\ \mathbf{e}_2 = (010\ldots0),\ \mathbf{e}_3 = (001..0),\ \ldots,\ \mathbf{e}_n = (000..1). \tag{5.22}$$

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

$$\mathbf{e} = \begin{vmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \\ \cdot \\ \mathbf{e}_i \\ \cdot \\ \mathbf{e}_n \end{vmatrix} = \mathbf{I}_n = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}, \tag{5.23}$$

where each row is the single error vector. Using properties of identity matrices, it is easy to show, that the matrix of syndromes coincides with the transposed parity check matrix of this code (5.21):

$$\mathbf{S} = \mathbf{e} \cdot \mathbf{H}^{\mathrm{T}} = \mathbf{I}_{\mathrm{n}} \cdot \mathbf{H}^{\mathrm{T}} = \mathbf{H}^{\mathrm{T}}. \tag{5.24}$$

By the syndrome decoding of a block code, the **matrix of syndromes S** coincides with the transposed check matrix of a code $\mathbf{H}^{\mathrm{T}}$. It allows to fill in the syndromes table**.** The more low reduced table 5.4 of syndromes for a code (7,4) is made according to rows of the transposed check matrix (5.21). In the table each vector of an error corresponds with the vector of the syndrome specifying a location of an error symbol in the received code word.

Table 5.4 – Syndromes for decoding of the code (7,4)

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

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

1. Forming of the transposed parity check matrix of a code $\mathbf{H}^{\mathrm{T}}$.
2. Tabling of syndromes for decoding of (n, k) code.

3. An evaluation of syndromes (see table 5.4) on the structure of transposed check matrix $\mathbf{H}^T$ code and error vector of a decoded codeword is done by the rule (5.24).

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: $\hat{\hat{\mathbf{b}}} = \hat{\mathbf{b}} \oplus \mathbf{e}_i$.

6. Dropping additional symbols in the code word $\hat{\hat{\mathbf{b}}}$ gives the decoded word $\hat{\mathbf{a}}$.

The diagram of a syndrome decoder of the code (7,4) showing this algorithm is in figure 5.5. According to the rule (5.24) received channel symbols are moved to modulo-2 adders. The connections with lines of channel symbols are available where the symbol 1 is arranged in rows of transposed parity check matrix. In the scheme of syndrome analyzer is given in accordance with the given table 5.4 there is transformation of syndrome vectors $\mathbf{S} = (s_0, s_1, ..., s_{n-k-1})$ in the corresponding error vectors $\mathbf{e}$ which then move to the error corrector is presented.

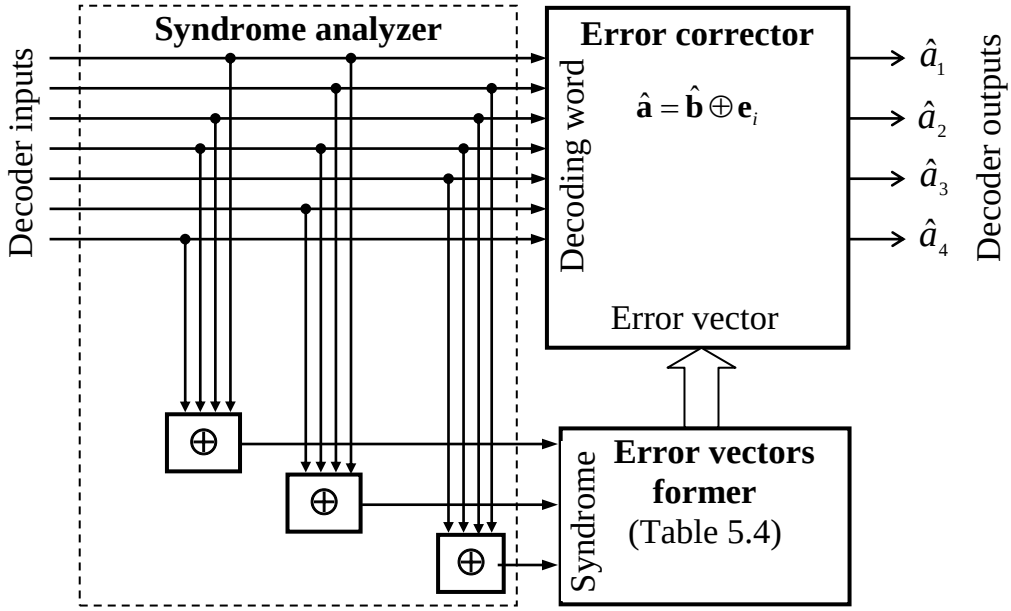


Figure 5.5 – The structure of the syndrome decoder of the code (7,4)

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

$$\mathbf{G} = \begin{vmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{vmatrix}. \qquad (5.25)$$

The transposed parity check matrix corresponds to this matrix:

$$\mathbf{H}^{\mathrm{T}} = \begin{vmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}. \qquad (5.26)$$

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

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

As the considered code is systematic, first three symbols (b_1, b_2, b_3) are information symbols. Using structural properties of this code, it is possible to form both trivial and compound estimations of information symbols during decoding, which are presented in table 5.5. On the basis of columns of parity check matrix (5.26) we will write down **verifying parities:**

$$b_1 \oplus b_3 \oplus b_4 = 0,\ b_1 \oplus b_2 \oplus b_3 \oplus b_5 = 0,\ b_1 \oplus b_2 \oplus b_6 = 0,\ b_2 \oplus b_3 \oplus b_7 = 0, \quad (5.27)$$

which allow us to form compound estimations. For example, on the basis of the first equality from (5.27) is followed the **compound estimation** of the first information symbol $b_1 = b_3 \oplus b_4$. The **trivial estimation** of this symbol is also used this symbol $b_1 = b_1$, as a code is systematic. The expressions for other information symbols are made similarly. They are presented in the table 5.5.

Table 5.5 – Majority decoding of the 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 to a **majority element** in which the decision on each information symbol is made taking into account «the **majority of voices**».

For example, if estimations of information symbol b_1 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 the quantity of estimations «0», the majority element passes the decision «to the majority»: $b_1 = 1$. The compound estimations are enumerated in table 5.5 and called «**orthogonal estimations**» as they

contain incoincident numerals. The number of orthogonal estimations N and a multiplicity of errors q_{corr}, corrected by majority decoding are presented by the ratio:

$$q_{corr.} \le (N-1)/2. \quad (5.28)$$

The code with generator matrix (5.25) allows to form $N = 3$ orthogonal estimations and, accordingly, to correct unitary errors in information symbols by considerable simplification of the decoding algorithm. It is necessary to notice that rules that form estimations can have cyclic properties that simplifie decoding procedure.

Example 5.7. The structure of the majority decoder for the code (7,3).

Let's create the structure of majority decoder of a code (7,3) on the basis of estimations system from table 5.5. It is easy to see that checks have cyclic properties.

For example, indeces 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 by 1 towards increase. Taking it into account the decoder structure of a code (7,3) you can see that the majority decoding algorithm looks like shown in figure 5.6. The decoder consists of the shift register, the switchboard at the input, operated from the system for block synchronization schemes for formation estimations, and the majority element. The decoder works as follows. At the beginning the switchboard at an input is established in position «1» and decoded code word **b** = (b_1, b_2, b_3, b_4, b_5, b_6, b_7) is entered in shift register cells. Thus at inputs of a majority element the compound estimations defined by table 5.5 execute both trivial and compound estimations. The decision about transmitted information symbol b_1 is read out from an output of a majority element. Then the switchboard is established in position «2» and there is on one symbol shift of word. At this step, the second information symbol is formed due to the cyclic properties of estimations and the decision of an information symbol b_2 is read out from output of majority element. The further process repeated up to reception of an output symbol b_3, etc.

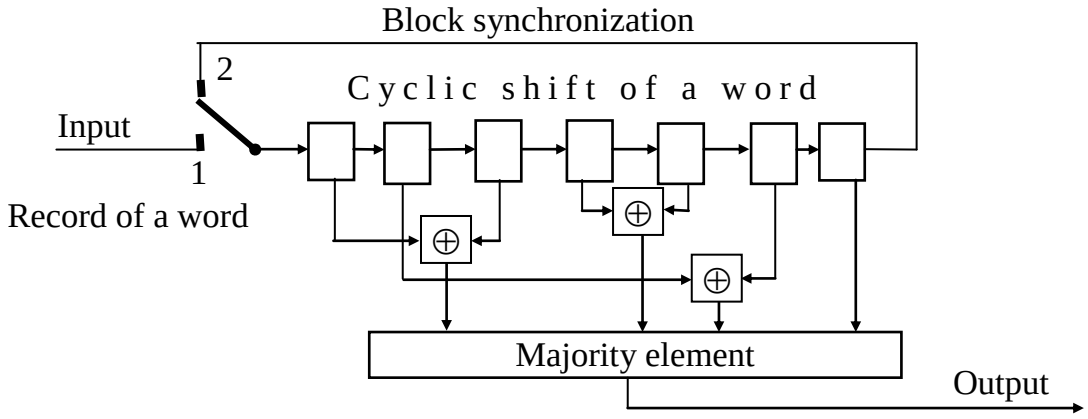


Figure 5.6 – Structure of the majority decoder of a code (7,3)

5.4 Boundaries of block codes parameters

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

Hamming upper bound. The derivation of the formula for the upper bound is based on reasons of spherical packing (bound of **spherical packing**) at given mini-

mum distance between the permitted code words $d_{\min}$. The greatest rate can be reached, if the spheres surrounding each word will be **most densely packed.** The volume of each sphere is equal to $\sum_{i=0}^{d_{\min}-1} C_n^i$ and the number of spheres (the number of code words) is equal to 2^k. For the best code the total quantity of spheres and the number of all possible words 2^n should coincide. Equality is reached for **densely packed (perfect) codes**. The area of each code word represents sphere with radius $(d_{\min} - 1)/2$, and these areas of such codes are not crossed densely and fill all n-dimensional space of code words themselves. From here follows the inequality:

$$2^{n-k} = 2^r \geq \sum_{i=0}^{(d_{\min}-1)/2} C_n^i . \qquad (5.29)$$

After simple transformations (5.29) it is possible to receive an obvious expression for the **code** rate:

$$R_{\text{code}} = \frac{k}{n} \leq \frac{1}{n}\log_2 \sum_{i=0}^{(d_{\min}-1)/2} C_n^i . \qquad (5.30)$$

Codes with parameters n, k and $d_{\min}$, for which the inequalities (5.29) and (5.30) become equalities are called perfect. The borders of Hamming are called:

- **upper,** if we are talking about inequality-equality (5.30), since the right side limits the value of the code rate;
- **lower,** if we are talking about inequality-equality (5.29), since the right side limits the value of additional symbols.

The dependence of Hamming upper bound is shown in figure 5.7 (curve «Hamming upper bound»). The meaning of this relationship is as follows: there is no code with parameters n, k and $d_{\min}$, which in these coordinates is indicated by a point above the dependence «Hamming upper bound».

Hamming bound is fair both for linear and nonlinear codes.

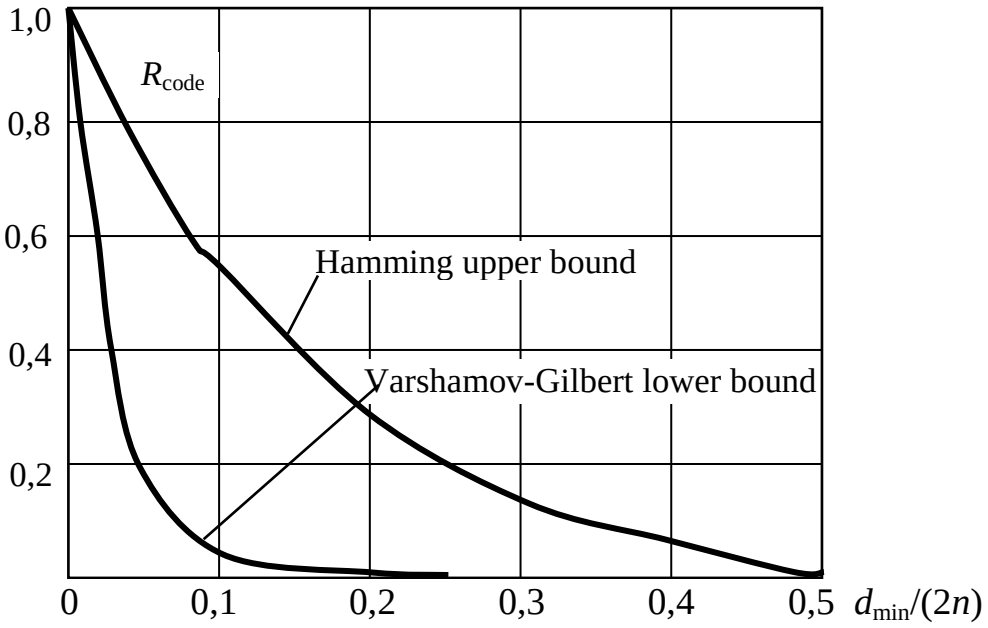


Figure 5.7 – Code boundaries of block codes

For block codes it is possible to get the **Varshamov-Gilbert lower bound** (figure 5.7). The asymptotic form (for long codes) of this bound looks like:

$$R_{\text{code}} \geq 1 - H(d_{\min}/n), \tag{5.31}$$

where $H(x)$ – binary entropy.

This bound defines the possibility of codes existence. We can find code that is shown by the point above the dependence «Varshamov-Gilbert lower bound».

The bound guarantees the existence of the codes with both parameters R_{code} and $d_{\min}$, which performances correspond to points arranged at least on a curve (or above it): we can find a code that is displayed by point above the dependence «Varshamov-Gilbert lower bound». The **search of the codes** ensuring the given code distance $d_{\min}$ and high enough rate R_{code} at $n \to \infty$, providing at the same time a possibility of algorithms decoding realization with low complexity is one of **important problems** of the theory of coding.

The using of code control ability depends on decoding algorithm. All possibilities of errors correction that follow from properties of a code are used by **full decoding**. According to Shannon fundamental theorem the error-control codes are used to correct channel errors and should be long enough. However, with the growth of a code word length n the complexity of realization procedures increases both for encoding and decoding. It causes difficulty to implement codecs in practice. In the coding theory one can estimate the complexity of realization of encoding/decoding procedures along with estimations of code error-control ability. They can be realized by software or hardware. Thus the length of a codeword n should act as an argument of the complexity function.

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

$$C_{\text{cod}} = nk = n^2(1 - R_{\text{code}}). \tag{5.32}$$

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

$$C_{\text{decod}} = m^n = 2^n. \tag{5.33}$$

It is said that the complexity of full-search decoding increases «as an exponent» with the growth of a code length. Clearly, that **full-search decoding algorithms are practically difficult for long codes to realize**.

5.5 Important classes of block codes

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

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

– Code word length $n = 2^r - 1$;

– Quantity of information symbols $k = 2^r - r - 1$; (5.34)

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

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

Hamming codes are perfect codes which **correct unitary errors**.

If parameters choice is $r = 2, 3, 4$, then according to the formula (5.34) 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 $k = 4$;

– Code distance $d_{\min} = 3$;

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

Generating and check matrices of this code have been considered earlier, in Section 5.3 (formulas (5.10) and (5.17)). As it has been noted before, this code also allows to **detect double errors**. Structures of an encoder and a syndrome decoder of Hamming code have been considered earlier in Section 5.3 (figures 5.4, 5.5). According to the formula (5.15) the transposed parity check matrix of this code looks like:

$$\mathbf{H}^{\mathrm{T}} = \begin{vmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}. \qquad (5.35)$$

From this formula it is seen that the size of the checking matrix of the Hamming code is $kn = n^2(1 - R_{\text{code}})$. This allows us to conclude that the complexity of the algorithm for decoding Hamming code is estimated by a power function of the form (5.33).

In the theory of block codes, the procedures for modifying the generating matrices of known codes are often used to find new codes. One such procedure is the procedure for shortening the code. In particular, any systematic (n, k) code can be shortened by going to $(n - b, k - b)$ code by discarding b information symbols in each permitted codeword at the encoder output (for example, by discarding the b bits). With this shortening, the discarded symbols are replaced with zeros that are not transmitted. But when decoding, the decoder restores them, so that the decoding is performed on the full length of the code. If the code distance of the initial code is $d^*_{\min}$, then the code distance of the shortened code is not less than $d^*_{\min}$. This is illustrated by the following example.

Example 5.8. Shortened Hamming code.

Let's consider the code (7, 4) with a generator matrix

$$\mathbf{G} = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{vmatrix}. \qquad (5.36)$$

The rate of this code is $R_{\text{code}} = 4/7$. We shorten the code by excluding the first row in this matrix ($b = 1$). The result of this shortening is the matrix:

$$\mathbf{G}^* = \begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{vmatrix}. \qquad (5.37)$$

It can be seen that the shortening procedure changed the code rate: $R^*_{\text{code}} = 3/6$. In this case, the code distances of the initial code (5.36) and the shortened code (5.37) coincide. Shortening is sometimes used to change the format of the block code (for example, to reduce the length of the block), if necessary, to "integrate" the combinations of the correction code into the symbol stream in the telecommunications system, or to match the format of this block code with the format of other code.

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 **b** = (b_0, b_1, ..., b_n) is the permitted code word of the cyclic code, its **cyclical shift** on arbitrary number of symbols is also the permitted code word. For example, a word **b1** = (b_n, b_0, b_1, ..., b_{n-1}) corresponds to the cyclical shift of a word **b** = (b_0, b_1, ..., b_{n-1}, b_n) by one symbol to the right. Thus according to the rule of a **cyclical permutation** combination symbols **b** are displaced on one numeral to the right, and the right numeral b_n takes place of the left numeral b_0. Properties of the cyclic code are convenient for studying and representing code words in the form of polynomials on powers 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) are made by the rules of polynomial algebra which were mentioned in Section 5 of the Manual [2]. If addition and multiplication of polynomials is made by the **modulo of polynomial ($x^n - 1$)** so all possible polynomials of power ($n - 1$) and less organize an **algebraic ring** of polynomials $\boldsymbol{R_n}$ with the properties stated in the Manual [2]. To construct a cyclic code in a ring $\boldsymbol{R_n}$ choose a subset of polynomials as an **ideal I.** The polynomial of the minimum power $\boldsymbol{g}(x)$ in this subset is called the **generator polynomial** of a cyclic code. Generator polynomials of the cyclic code choose the **prime polynomial**. In algebra of polynomials the prime polynomial plays the same role that the **prime number** plays in the algebra of integers. The detailed table of cyclical codes generator polynomials is presented in the Appendix A.1. Generator polynomials of short cyclic codes are given in table 5.6.

Table 5.6 – Generator polynomials of short cyclic codes

Maximum power 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 permitted code words of cyclic codes are divided into generator polynomial $\boldsymbol{g}(x)$ without remainder that allows to formulate the following coding rule:

The coding rule of nonsystematic cyclic code looks like:

$$\boldsymbol{b}(x) = \boldsymbol{a}(x)\cdot\boldsymbol{g}(x). \qquad (5.38)$$

The systematic cyclic codes are often used in practice.

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

$$\boldsymbol{b}(x) = \boldsymbol{a}(x)\cdot x^{n-k} + \boldsymbol{r}(x), \qquad (5.39)$$

where $\boldsymbol{r}(x)$ – remainder of division $\boldsymbol{a}(x)\cdot x^{n-k}$ on $\boldsymbol{g}(x)$. The coding rule (5.39) can be realized by such a **coding algorithm** for a 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 the polynomial multiplication $\boldsymbol{a}(x)$ on x^{n-k}.

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

3. The calculated remainder is added to the displaced combination $\boldsymbol{a}(x)\cdot x^{n-k}$. Therefore, the **permitted code word** is formed as (5.39).

Example 5.9. Forming of a code word of the 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). The polynomial representation of the primary code word will be $\boldsymbol{a}(x) = x^4 + x^2 + x$. A given cyclic code has parameters: $n = 10$, $k = 5$, $r = (n - k) = 5$. From the table 5.6 the generator polynomial $\boldsymbol{g}(x) = x^5 + x^4 + x^2 + 1$ is chosen as an example. Next we will do mathematical operations according to the algorithm (5.39):

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

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

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

3) The polynomial of a permitted code word is

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

A word of binary symbols $\boldsymbol{b}$ = (1011001101) corresponds to polynomial $\boldsymbol{b}(x) = x^9 + x^7 + x^6 + x^3 + x^2 + 1$ in which the first four symbols are informational and

the remaining – additional. The **property of divisibility** of permitted code words on the generator polynomial is widely used to detect of errors in transmission systems.

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

$$\hat{\boldsymbol{b}}(x)/\boldsymbol{g}(x) = \boldsymbol{q}(x) + \boldsymbol{s}(x). \qquad (5.40)$$

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

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

Example 5.10. Syndrome decoding of a cyclic code (7, 4).

The word of binary primary code **a** = (1010) as a subject for transmission via the channel with a unitary errors is set. Let's choose the cyclic code ensuring errorless transmission this word in these conditions. From table 5.6 we define, that the task can be solved by using 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 the syndrome decoding** for the correction of unitary errors is realized. Using encoding algorithm (5.39), we will generate the permitted 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

$$\hat{\boldsymbol{b}}(x) = \boldsymbol{b}(x) + \boldsymbol{e}(x) = x^6 + x^4 + 1 + x^6 = x^4 + 1. \qquad (5.41)$$

We use a rule (5.40) to determine 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 corresponding polynomials errors. To fill in such a table it is necessary to take advantage of the equality from (5.40) by $\boldsymbol{q}(x) = 0$:

$$\boldsymbol{s}(x) = \boldsymbol{e}(x)/\boldsymbol{g}(x). \qquad (5.42)$$

It outcomes of evaluations are presented in table 5.7 under this formula of polynomials syndrome $\boldsymbol{s}(x)$ for various polynomials of errors. Values of syndromes are presented in the form of binary words in table 5.7.

Table 5.7 – 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 looks 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 5.7 it is discovered, that the error polynomial $\boldsymbol{e}(x) = x^6$ corresponds to such

syndrome. The error correction includes the 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 permitted word $\boldsymbol{b}(x) = x^6 + x^4 + 1$. There a binary word **b** = (1010101) corresponds to it, where the first four symbols are errorless transmitted symbols of a 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 code distance $d_{\min} = 7$.

2 **Expanded Goley code** (24, 12) with code 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 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.

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

The effective using of cyclic properties of permitted words of cyclic codes allows to realize simple enough decoding algorithms. It is considered, that realization of the complexity of cyclic codes decoding algorithms is described by power function $C_{\text{decod}} = n^k$, where the k – a small number which size depends on concrete algorithm realization. Thus the mathematical apparatus of algebric polynomial and of description the discrete linear filters is presented in Sections 5 and 6 of the Manual [2] and widely used.

Example 5.11. The encoder structure of the systematic cyclic code.

Using algorithm (5.39) 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 5.6. The scheme of an encoder is presented in figure 5.9.

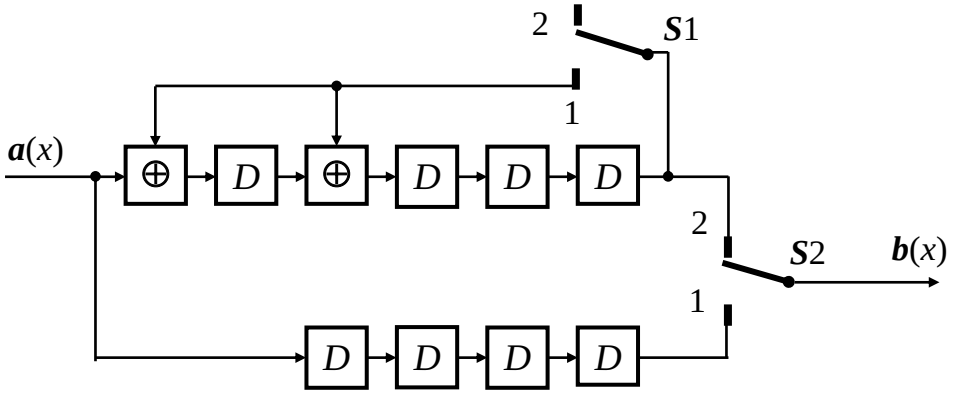


Figure 5.8 – Encoder of systematic cyclic code (15,11)

According to an algorithm (5.39), the encoder works as follows. Originally switches $S1$ and $S2$ are in position 1. Eleven information symbols of a coded prime word $\boldsymbol{a}(x)$ are entered at the left into the order of division into a polynomial $\boldsymbol{g}(x) = x^4 + x + 1$. Simultaneously they go through consistently connected delay ele-

ments and arrive to an encoder output, forming an information part of the permitted code word $\boldsymbol{a}(x)\cdot x^{n-k}$. The first four steps in register cells of the divider scheme on a generator polynomial the remainder of a division $\boldsymbol{r}(x)$ is formed. Then switches $S1$ and $S2$ are established in position 2, division process stops, and the remainder is read out from an output of a divider and finished in a checking part of the final code word $\boldsymbol{b}(x) = \boldsymbol{a}(x)\cdot x^{n-k} + \boldsymbol{r}(x)$.

Example 5.12. Encoder structure of the nonsystematic cyclic code

Using a coding rule (5.38) for a nonsystematic cyclic code, we will form the coder block diagram for generator polynomial $\boldsymbol{g}(x) = x^4 + x + 1$. The coding rule (5.38) provides the multiplication of polynomials $\boldsymbol{a}(x)$ and $\boldsymbol{g}(x)$. Using the structure of a multiplier for polynomials from section 6.1 of the Manual [2] we will present the encoder scheme in figure 5.9. The important element of coder's schemes for cyclic codes is the scheme of division of polynomial into a polynomial for an evaluation of a division remainder by coding of systematic code by the algorithm (5.39). It is also applied for the syndrome evaluation by syndrome decoding of an algorithm (5.40). The structure of such divider schemes is considered in Section 6.1 of the Manual [2].

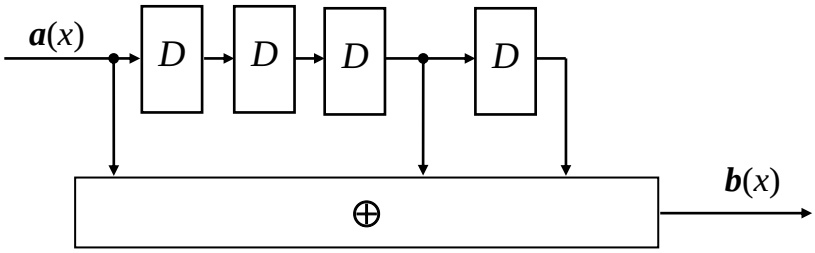


Figure 5.9 – Encoder of the nonsystematic cyclic code

5.6 Decoding noise immunity of block codes

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

$$P(q) = C_n^q p^q (1-p)^{n-q}. \tag{5.43}$$

Here C_n^q – the 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 at the decoder output, the word with not corrected errors (erroneous decoding) will be equal to

$$P_{\text{err dec}} \le \sum_{q=q_{\text{corr}}+1}^{n} P(q). \tag{5.44}$$

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

$$P_{\text{err dec}} \le \sum_{q=q_{\text{corr}}+1}^{n} C_n^q p^q (1-p)^{n-q}. \qquad (5.45)$$

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

The expression (5.45) allows to define the **upper estimation of error probability** of code words by decoding block codes in binary symmetric channel without memory. To calculate the probability of an error in concrete information (or additional) symbols it is necessary to know the used decoding algorithm and the structure of an error-control code (in particular, a set of distances from a transmitted code word to all permitted words). Such data for block codes are not presented in code tables and to calculate error probability decoding of code symbols (information or additional) the **approximated formula** [1] is use:

$$p_{\text{d}} = \frac{d_{\min}}{n} \sum_{q=q_{\text{corr}}+1}^{n} C_n^q p^q (1-p)^{n-q}. \qquad (5.46)$$

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

$$p = Q\left(\sqrt{2}h_{\text{s}}\right), \qquad (5.47)$$

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

$h_{\text{s}}^2 = h_{\text{b}}^2 \cdot k/n$; h_{b}^2 – the ratio of the energy spent on transfer of one bit E_{b} to power spectral density of noise N_0 at a demodulator input;

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

$$Q(z) = 0{,}65 \exp[-0{,}44(z + 0{,}75)^2]. \qquad (5.48)$$

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

$$\Delta F_{\text{code}} = \Delta F_s / R_{\text{code}}. \qquad (5.49)$$

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 be applicable. Therefore, when a transmission system is designed, the **problem of a code choice** lies in a search of a **compromise between desirable degree of a noise immunity and expansion of a frequency band of the coded signal**. Under formulas (5.46) and (5.47) is taken into account the expansion of a frequency band of coded signal according to the formula

(5.49). The following conclusions are made on the **efficiency of error-control coding application**:

1. With the growth of a code word the length n of the error probability of a decoding p_d goes down.

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

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

$$K_{\Delta F} = n/k\,. \tag{5.50}$$

In practice the application of error-control codes in telecommunications is important. This question is solved taking into account the following. The introduction of redundancy by encoding changes not only expands of a frequency band for the transmission of coded signals, but also requires the redundancy while calculating energy. In practice, according to the formula (5.47) the probability of 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 transferring of one information symbol (bit). It is followed from the equality $kE_b = nE_s$, i.e. $E_s = E_bR_{code}$. Therefore, as a rule in all energy calculations of systems with coding the value of the ratio of signal energy spent for transmitting one information binary symbol (bit) to noise power spectral density E_b/N_0 is defined. The probability of erroneous decoding of the block is defined by formulas (5.45) and (5.47) which in an argument of function $Q(z)$ include the value E_s – the energy of a signal spent for transmitting one binary signal (a code symbol) through the channel. In practice, according to formula (5.47) the probability of an error registration of channel signals (code symbols) is defined by their energy E_s, redundancy of a code is considered, it appears that is a little bit less energy E_b spent to transmit one information symbol (bit). Then the ratio of energy E_b to noise power spectral density N_0 can be designated as $h_b^2 = E_b/N_0$ used in power calculations of systems with coding. Taking into account the relation of signal energy E_s and bit energy E_b (5.47) the value in the formula will be $h_b^2 = R_{code}h_s^2$. Considering the expenses of energy for transmitting additional symbols of a redundancy code (5.47), it is possible to present the formula as follows:

$$p = Q(\sqrt{2R_{code}}h_b)\,, \tag{5.51}$$

and the bit error probability by expression (5.46). If it is necessary to define the probability of an error in the channel without coding it is enough to use the formula (5.47), with $R_{code} = 1$:

$$p = Q(\sqrt{2}h_b)\,. \tag{5.52}$$

Exercise 5.4. Decoding noise immunity of a block code.

Let's use the formula (5.46) for calculations of an error probability with optimum receiving of signals BPSK in the channel without coding. The results of calculations are given in table 5.8. The initial parameter for calculations is the signal/noise

ratio at 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 5.8 tabulate data is introduced that define by error probability by optimum reception of BPSK signal (formula (5.46)) including argument z of the function $Q(z)$.

The dependence curve $p = f(h_b^2, \text{dB})$ received on these data (BPSK) is presented in figure 5.10.

Under formulas (5.46), (5.51) we will define bit error probability by decoding words of cyclic code average length (31, 26) with parameters R_{code} =0,84, d_{min} = 3, q_{cor} = 1 in the channel with BPSK. The code is chosen from the table A.1.

The results of calculations are presented in figure 5.10 (curve «Code (31,26)»).

Table 5.8 – The 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 5.9 – The calculation of 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 the value of the ratio of the signal energy spent to transmit one information binary symbol (bit) to power spectral density of noise h_b^2 is used as a rule. It is considered **a uniform criterion of energy spent** for an information transfer through the channel with coding and without it. The

size change h_b^2 shows the **efficiency of application of an error-control code**. The effect of errors decreases at decoder output can be used differently.

The error-control coding provides the **reduction of the error probability** in the received messages. It is well visible from comparison of curves $p = f(E_b/N_0)$ in figure 5.10 for cases of an information transfer by BPSK uncoded method and with using of a cyclic code (31, 26). It is visible, that by using of an error-correcting code it is possible to agree with certain decrease in a channel signal/noise ratio and to receive accordingly a energy coding gain **CG** (dB). The **energy coding gain** from error-control coding CG is equal to a **difference** of values E_b/N_0 needed to maintain bit error probability in transmitted data by either by encoding or not. The value of the gain can be defined at various levels of bit error probability p at demodulator and decoder outputs. The above said is illustrated by the curves of a noise immunity presented in figure 5.10.

In particular, for the data presented in figure 5.10 the value coding gain is CG = 1,55 dB at $p = 10^{-5}$.

Gain values CG are widely used to **choose codes for designing** transmission systems. Values CG received by using of cyclic codes in channels with BPSK are presented in Appendix A.2.

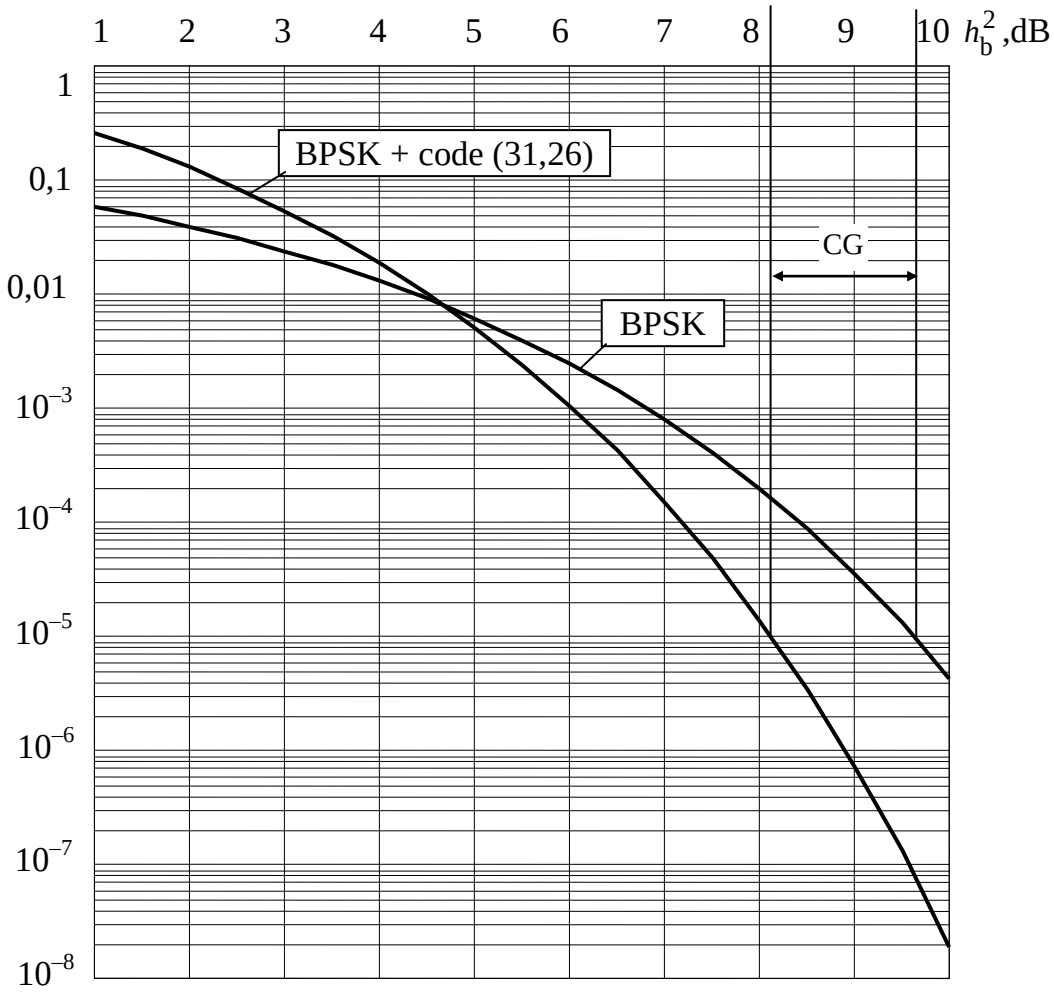


Figure 5.10 – Decoding noise immunity of a cyclic code

Example 5.13. The optimisation of cyclic code parameters.

Let's consider the optimisation procedure of cyclic codes parameters used in the binary symmetric channel with BPSK signals for the purpose of a maximum energy coding gain. The spectrum band expanding will not exceed $K_{\Delta F} = 2$ (double expansion of a signal frequency band in the channel). In Appendix A.2 we made a selection of cyclic codes which can meet the requirements of a band expansion factor ($K_{\Delta F} < 2$, $R_{\text{code}} > 0{,}5$). The results of such selection are shown in table 5.10. In table columns values of code rate are specified. The gain values (in dB) for various lengths of code word n are presented in the cells in the lines. Using Appendix A.1 we select the cyclic codes with block length $n = 255$, with rate which is closed to the optimum rate $R_{\text{code}} = 0{,}8$. It is visible, that the greatest value of gain CG = 4,0 dB is reached when long enough cyclic codes with word length $n = 255$ are used. In table 5.11 the parameters of the optimum cyclic code are shown.

Table 5.10 – The parameters of cyclic codes meeting the requirements of 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 5.11 – Characteristics of an optimum cyclic code

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

The selected code (255, 207) provides a power gain 4,0 dB at the rate $R_{\text{code}} = 0{,}811$. Band expansion factor $K_{\Delta F} = 1{,}23$ does not exceed the predefined value max $K_{\Delta F} = 2$.

Questions for section 5

5.1. What is the reason for wide application of binary codes in transmission systems?

5.2. Define a systematic block code.

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

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

5.5. What kind of double errors will a matrix have? How will it change in comparison with a matrix of unitary errors?

5.6. How are parameters of binary syndrome representation (see table 5.4) connected with the general number of possible configurations variants which detect and correct errors by syndrome decoding?

5.7. How will the syndrome format change if to apply a method of syndrome decoding is applied to decode double errors?

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

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

5.10. To what bound (upper or lower) is it necessary to move when developing of new block codes?

5.11. What are the key parameters of Hamming codes?

5.12. What are the advantages of cyclic codes?

5.13. Is it possible to use Hamming codes and cyclic codes to correct unitary errors? What will parameters of these codes be?

5.14. What are parameters of block error-control codes that define the error probability at decoder output (binary symmetric channel)?

5.15. How is the energy coding gain defined?

5.16. What are the reasons of signal frequency band expansion at coding?

Tasks

5.1. Represent structure of a systematic block code intended for detection of double errors with the generator matrix (4.6) by the principles given in the Example 5.1.

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 the permitted code word of this code **b** if the word at 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 the encoder structure of a code (7, 4) with the same generator matrix.

5.5. 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 permitted code word of this code ***b*** if a word of simple code at encoder input ***a*** = (1110) is set.

5.6. The 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.

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

5.8. By analogy according to formula (5.34) make the table of Hamming codes parameters for values r = 2, 3, 4. As these codes have identical code distance, compare their suitability for implementation in practical systems. Formulate the recommendation and a substantiation of application of the best (in your opinion) code from this list.

5.9. Form generating and check matrices for Hamming code recommended in the previous Section.

5.10. By the rules given in Exercise 5.2 define the value of a code distance by a generator the matrix of a code from the Task 5.8.

5.11. By a technique given in the Example 5.13 define parameters and generator polynomial of the cyclic code that provides the minimum expansion of a signal frequency band by energy coding gain CG > 3,0 dB.

5.12. Construct the dependence family of an energy coding gain CG from the code rate for various lengths of the code word for the cyclic code using data in Appendix A.2. Make the how the length of the block influences the gain value.

5.13. Construct dependences of a necessary code rate from a demanded energy coding gain CG for various lengths of the code word for the cyclic code using data in Appendix A.2. Make conclusions how a word length influences the exchange parities between a gain and the signal band expansion.

6 CONVOLUTIONAL ERROR-CONTROL CODES

6.1 Description methods of convolutional codes

Convolutional codes (CC) form a subclass of **continuous codes**. The name «convolutional code» occurs when the result of coding at the encoder output is formed as the convolution of coded information sequence with the pulse response of encoder. The encoder of CC contains one or several registers from delay elements and the converter of information sequences into code sequences. The coding process is made continuously. The scheme of simple encoder is shown in figure 6.1.

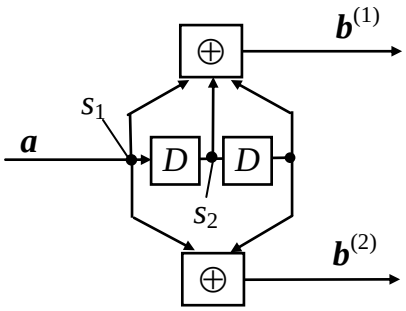


Figure 6.1 – Encoder of CC

Information binary symbols ***a*** arrive at the input of the register with *K* delay elements *D*. At outputs 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. During one input information symbol two output code symbols are formed.

The code rate is $R_{\text{code}} = k/n$, where k – the number of the information symbols simultaneously arriving at inputs of encoder, and n – the number of code symbols corresponding to them at encoder outputs. The code rate in this example is equal to $R_{\text{code}} = 1/2$. The coding with other speeds is possible. Convolutional encoder as a finite state machine with a final number of states can be described by the state diagram. A state is the symbol set at the inputs of register delay elements. For example, symbols (s_1, s_2) designate **encoder state** in figure 6.1.

The **state diagram** represents the **directed graph** that describes all possible transitions of an encoder from one state into another and also contains encoder output symbols which accompany these transitions.

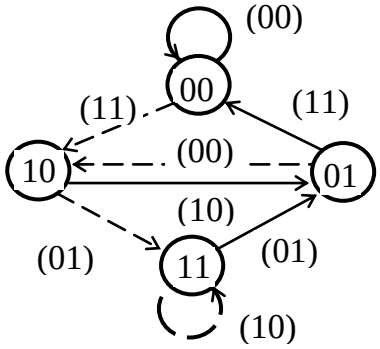


Figure 6.2 – State diagram

The example of the encoder state diagram is shown in figure 6.2. It contains four possible encoder states (s_1s_2) = 00, 10, 11 and 01 and possible transitions. Symbols above arrows designate symbols at an encoder output ($b^{(1)}b^{(2)}$) corresponding to the given transition. Continuous lines note the transitions made at the reception at encoder input of the information symbol 0 and dotted – by the reception of a symbol 1. Originally the encoder is in state 00, and the reception at its input of the information symbol $a = 0$ translates it also in state 00. Thus, at 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 reception $a = 1$ the encoder passes in state 10, thus, at 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, the reception at an input of the coder of information symbols 0 or 1 is possible. Thus, the coder passes into state 01 or 11, and symbols at the output will be 10 or 01

accordingly. The process of the diagram forming comes to an end, when all possible transitions from each state in all others will be understandable.

The **trellis diagram** (trellis) is representation 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 shift in one step to the right. The example of the trellis diagram is shown in figure 6.3. The trellis diagram gives evident representation of all permitted ways which are analogues of permitted code words of block codes. On them the encoder can move ahead by encoding. A **unique way through a trellis** corresponds to each information sequence at an encoder input.

In particular, a dotted line the way on a trellis shows …11100001… corresponding to input information sequence …1011… To describe the encoder work it is convenient to represent the sequence of input and output symbols with the use of the 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 – a number of encoder input, $1 \le i \le k$;

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

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

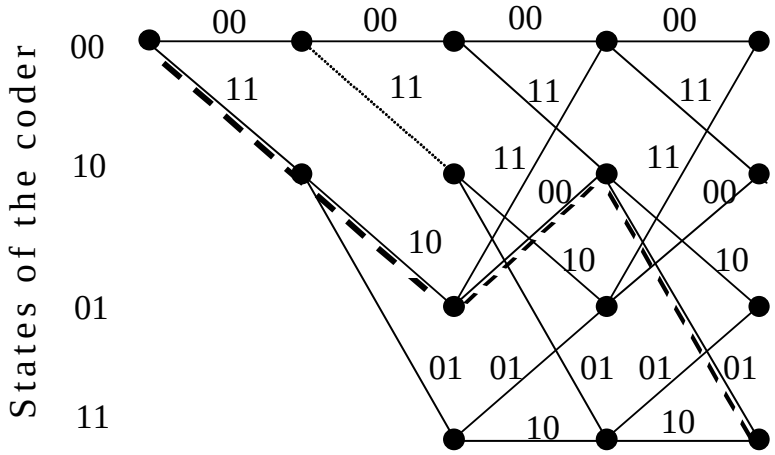


Figure 6.3 – Trellis diagram

Generator polynomial is used to represent convolution coding**.**

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

– the number of encoder inputs k;

– the number of encoder outputs n;

– the lengths of each registers K_i;

– connections of adders with register cells are specified in encoder scheme.

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:**

$$\boldsymbol{g}^{(j)}(D) = g_0^{(j)} + g_1^{(j)}D + g_2^{(j)}D^2 + ... + g_v^{(j)}D^v. \qquad (6.1)$$

Here $g_k^{(j)} = 1$, if a connection of j-th adder with k-th register cell outputs, and $g_k^{(j)} = 0$ if there is no such a connection.

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

$$\boldsymbol{b}^{(j)}(D) = \boldsymbol{a}_{(i)}(D)\boldsymbol{g}_{(i)}^{(j)}(D), \quad 1 \le i \le k, \quad 1 \le j \le n. \tag{6.2}$$

For example, the encoder in figure 6.1 is characterised by generator polynomials $\boldsymbol{g}^{(1)}(D) = 1 + D + D^2$ and $g^{(2)}(D) = 1 + D^2$ or, noting the sequence of factors g_k in the form of binary words, we receive $\boldsymbol{g}^{(1)} = (111)$ and $\boldsymbol{g}^{(2)} = (101)$. For long codes the **octal form** is often used**.** In this case the generator polynomials will be presented so:

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

The coding process can also be described by using generator matrices (accordingly, check matrices). It is possible to know this material more in detail in the Manual [1, Section 3.4, p. 114].

6.2 Key parameters and classification of convolutional codes

The code rate is defined as

$$R_{\text{code}} = k/n, \tag{6.3}$$

where k – the number of information symbols simultaneously arriving at k encoder inputs, n – the number of code symbols corresponding to them at n encoder outputs.

Several parameters are used to define the memory length by coding. The **length of encoder register** K is equal to the number of delay elements contained in the encoder scheme. The length of an encoder register is often used for the definition of memory by coding with rate $R_{\text{code}} = 1/n$, when an encoder has one register. The encoder represented in figure 6.1 has the register length $K = 3$. If encoder has some inputs ($k > 1$), then the lengths of registers connected to each input, can be various. In this case it is defined as a code constrained length.

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

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

The resulted code constrained length is defined by the sum:

$$\nu = \sum_{i=1}^{k} \nu_i \,. \tag{6.4}$$

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

$$\nu = K. \tag{6.5}$$

Complexity performance is used to compare a decoding algorithm complexity. As it was said above, a trellis diagram is made by a repetition of the same step (see figure 6.3). The diagram complexity is accepted to define a number of branches on a step of trellis diagram. The number of states of a trellis is defined by the number of variables $K = \nu$ at inputs of register elements. As a result **complexity** of one **trellis step** can be defined by the number of branches at this step

$$C = m^{(\nu + k)} \tag{6.6}$$

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

The free distance of a convolution code d_f – is the minimum distance between two arbitrary semi-infinite sequences at the encoder output which is 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 equal to minimum Hamming weight of a path in the diagram from a state 00 in the same state (cyclic loop at this state). In the diagram figure 6.2 is visible, that is free distance $d_f = 5$. By the value of free distance we can judge about correct properties of convolution codes. In particular, if two paths at the encoder output, going out from one state on the trellis diagram, differ in Hamming metric on the value d_f, then by decoding over a minimum distance. With the analogy to a case of block codes decoding (see Section 5.1), the **multiplicity of corrected errors** is defined by the expression

$$q_{\text{corr}} \le \frac{d_f - 1}{2}, \quad (d_f \text{ is odd}). \tag{6.7}$$

The free distance is used to estimate 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 at k (from n possible) encoder outputs there are information sequences of transmitted symbols, and on remaining $(n - k)$ outputs – 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** will 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 obtain at reception 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 block codes the use of convolution coding with rate $R_{\text{code}} = k/n$ leads to expansion of a signal frequency band in the channel. Thus the **band expansion factor** is defined by the expression:

$$K_{\Delta F} = \frac{n}{k}. \tag{6.8}$$

By small code rates the considerable band expansion becomes unallowable; therefore you should try to apply encoding with a high code rate. In practice, 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 6.1. 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 6.1, 6.2 and 6.3), we will estab-

lish 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 presented in figure 6.1. The diagram of states is constructed in figure 6.2.

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

2. The forming of a systematic code (1, 5).

Let modify the first polynomial of an initial code, having left one connection, as shown in figure 6.4. The state diagram will partially vary. The number of states remains the same as the structure of the encoder register is 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 belongs (figure 6.4). The code rate has not varied either.

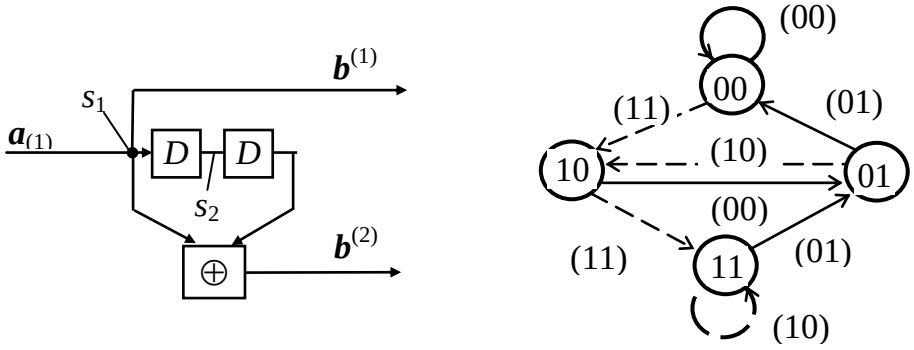


Figure 6.4 – Encoder of CC (1, 5) and its state diagram

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

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

3. The forming of a nonredundancy code (1,0).

This, apparently, the "exotic" example allows to reveal the role of nonzero generator polynomials forming additional symbols (figure 6.5).

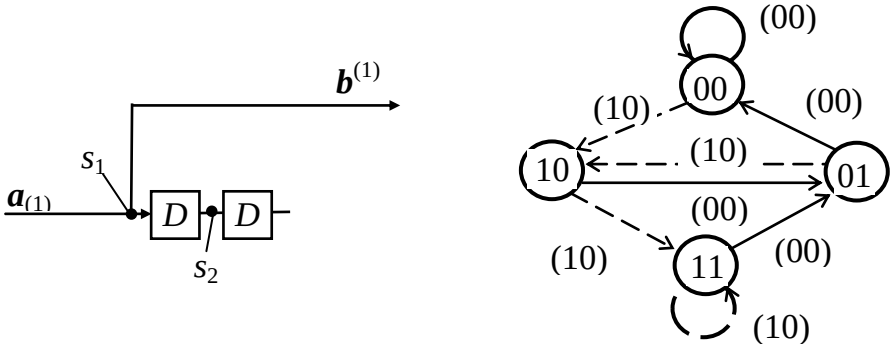


Figure 6.5 – Encoder of code (1, 0) and its state diagram

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

The encoder is systematic without additional symbols.

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

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

6.3 Classification of decoding algorithms

For the purpose of optimum solution it is necessary to compare the received sequence of symbols accepted from the channel with all possible transmitted sequences. As the number of possible sequences length N for binary code is equal to 2^N, the decoder becomes inadmissible complexity by the big sequence lengths (exponential decoding complexity, see Section 5.4), and the optimum decoding is practically difficult to realize. However, it is possible to substantial by increase transmission accuracy by big N as the noise averages on a long sequence. Therefore, the **problem** of **decreasing of decoding algorithms complexity** 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 simple realisations of codec. Such algorithms are not optimal, as algebraic decoding procedures are used and intended for concrete correction of configurations of channel errors (and not all). Algebraic methods are identified with **«element-by-element reception»** of sequences which for codes with redundancy, as it is known, yields the worse results, then «**reception in a whole**». Most simple algebraic algorithms are the algebraic decoding methods. This algorithm is so far from the optimum and consequently is seldom used, first of all, in systems with a high information rate. More detailed description of the threshold algorithm and its modification is presented in the Manual [1, Section 3.6.3].

2. **Probability decoding methods** are considerably nearer to the optimal "reception in a whole" as in this case decoder operates with values which are proportional to the posteriori probabilities. It 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 at an output of continuous channel is necessary to fulfil the **quantization** of a received signal with noise. Processes of elaborating of signals at an output of the demodulator for antipodal signals are shown in figure 6.6 where the following is presented:

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

b) – binary quantization by hard decision;

c) – octal quantization by soft decision.

In the simple case quantization of each channel symbol is made on two levels in the sample time (mentioned in the literature as **«a hard decision»**). Thus, hard de-

cision is presented by one binary symbol. It is shown in figure 6.6, *b*. By hard decision the number of quantization levels is $L = 2$. By a **soft decision** the number of quantization levels is $L > 2$ (figure 6.6, *c*). By soft decision the quantized output describes the magnitude of the decoded signal plus the more precisely noise that raises the noise immunity.

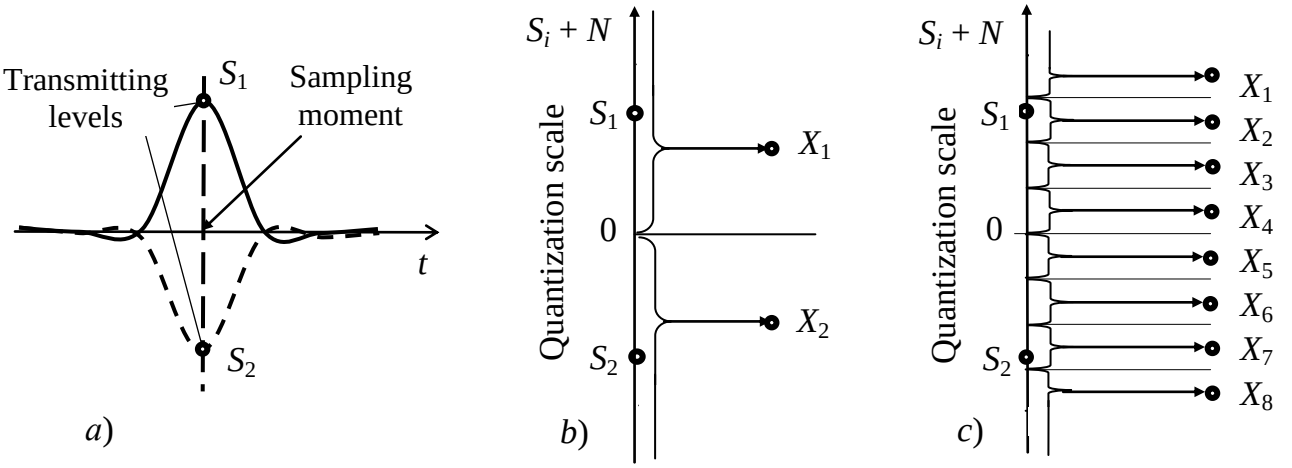


Figure 6.6 – Work of decision device in demodulator

Two basic probability algorithms for decoding 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 path through code trellis, corresponding to the transmitted informational sequence is made. Sequential decoding is used for decoding long convolutional codes. The detailed description of sequential decoding algorithm is presented in the book [3, Section 13.18]. Another variety of probability algorithms is the algorithm based on the principle of a dynamic programming, and known as Viterbi algorithm.

Dynamic programming principle was formulated in 1940 by R. Bellman. It has a wide application in control theory. In 1970 the dynamic programming in the form of a decoding algorithm for convolutional codes was applied by A. Viterbi to solve telecommunication problems (Viterbi algorithm).**Viterbi algorithm** finds wide application and allows search for the maximum probable path through code trellis with a rejection of a part of least probable variants of decoded paths. Viterbi algorithm is characterized by a constant of computing work. However, the complexity of Viterbi decoder grows because of all full search algorithms under the exponential law from code length. Therefore, Viterbi algorithm is used for decoding short convolutional codes.

6.4 Viterbi algorithm for decoding convolutional codes

Let's consider Viterbi algorithm on example of the code with rate $R_{\text{code}} = 1/n$. Let, since a moment $t = 0$, at encoder input the information sequence of length L symbols $\boldsymbol{a}_L = (a_0, a_1, \ldots, a_{L-1})$ moves. At an encoder output there will be a sequence of symbols $\boldsymbol{b}_L = (b_0, b_1, ..., b_{L-1})$. The encoder states at the moment t are defined as a set of ν information symbols $\boldsymbol{w}_t = (a_t, a_{t-1}, ..., a_{t-L+1})$. Trellis diagram of the code univalently connects the information sequence $\boldsymbol{a}_L$, the 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 the channel there corresponds a signal, which can be presented as 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 the channel an additive noise operates. Then arriving at decoder input the 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 is performed by tracing through a code trellis over a path with maximum of a posteriori probability. It is possible to specify the decoded path to one of kinds: by set of estimations of code branches $\boldsymbol{S}_L = (S_0, S_1, \ldots, S_{L-1})$ which make a path, 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 at 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 is chosen an estimation $\boldsymbol{S}_L$ for which a posteriori probability $P(\boldsymbol{S}_L/\boldsymbol{X}_L)$ is the maximum from all possible paths. The transmission of all variants of sequences $\boldsymbol{a_L}$ is considered equiprobable. In this case the decoding by criterion of a maximum a posteriori probability is equivalent to the decoding by criterion of the maximum of a probability when estimation $\boldsymbol{S}_L$ ensuring the 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 the sum of the 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 channel with white Gaussian noise and 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 the 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 $\boldsymbol{S}_L = (S_1, \ldots, S_{L-1})$ and sequence of branches univalently connected with it $\boldsymbol{S}_\text{L} = (S_0\, S_1 \ldots S_{L-1})$, which ensures a minimum sum

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

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 the received sum of a signal plus noise $\boldsymbol{X}_t$ and a vector of the signal $\boldsymbol{S}_t$ corresponding to branch of a code $\boldsymbol{A}_t$. In the discrete channel Hamming metric is used to estimate the distances. The periodic structure of the 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 chosen by random have enough long paths, and, as a rule, the common state. It is necessary to **compare** the segments of the paths entering into such states 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**, at each step of decoding.

In figure 6.7, *a, b, c, d* the development of decoding process of convolutional code (5, 7) is shown. Symbol pairs from channel arrive (…11,10,00,11,01...) at a decoder input (decoding with hard decision). Digits on branches designate branch metrics, figures about states designate **metric of states** (MS). In an initial moment of 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, then metrics of branches 00 and 11 are going out, this states will be MB (00) = 2 and MB (11) = 0. It is noted at the decoding first step. The similar picture takes place and at the following decoding step. The state metrics at this step are defined now as the sums of metrics of entering branches with the previous state metrics: MS (00) = 2 + 1 = 3; MS (10) = 2 + 1 = 3; MS (01) = 0 + 0 = 0 and MS (11) = 0 + 2 = 2.

Here the development of trellis diagram for the given code comes to an end. The **algorithm** consists of a **recurring** of **one basic step.** In each of the subsequent diagrams in figure 6.7, *a, b, c, d* this steps are represented explicitly. State metrics calculated at the previous stage are stored in memory of the decoder: $MS^{i-1}(00)$, $MS^{i-1}(10)$, $MS^{i-1}(01)$, $MS^{i-1}(11)$ to the beginning of *i*-th step.

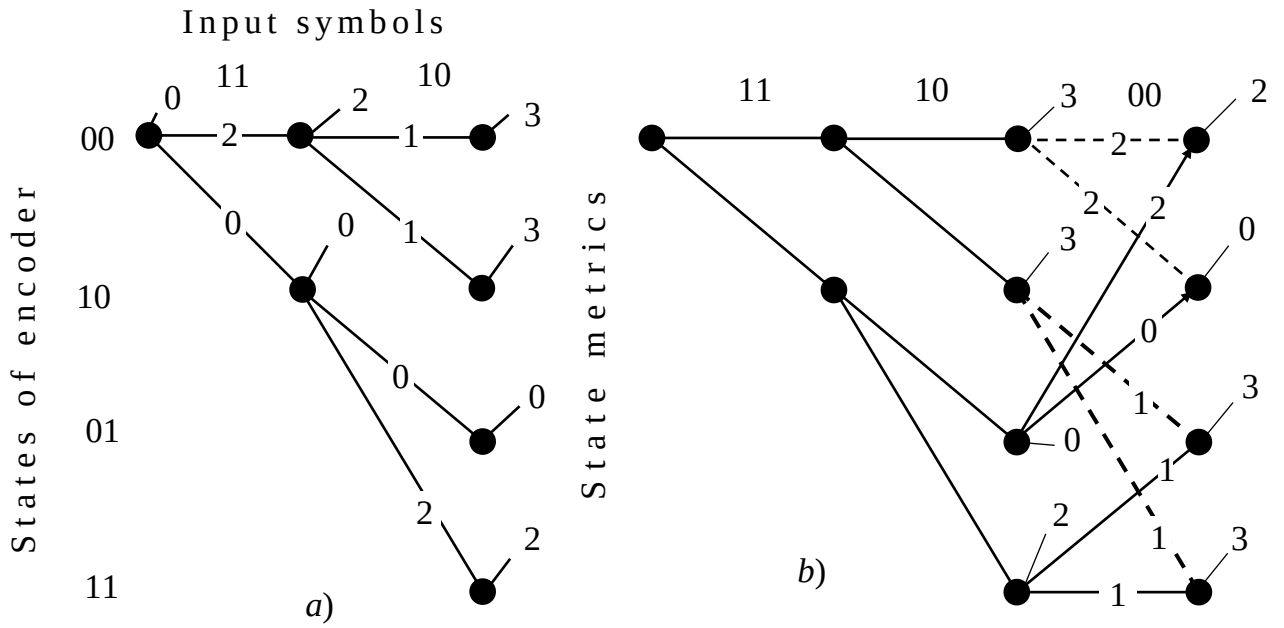


Figure 6.7, *a, b* – Decoding process by Viterbi algorithm

For the accepted channel symbols the evaluation of above branch metrics and shaping of four new states metrics is made: $MS^i(00)$, $MS^i(10)$, $MS^i(01)$ and $MS^i(11)$ by a following rule. Two paths lead to each new state. For example, state 00 conducts paths from the previous states 00 and 01. At *i*-th decoding step the decoder calculates metrics of paths as the sums of previous states metrics and entering branches metrics:

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

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

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

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

According to Viterbi algorithm at 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 paths is made in a random path.

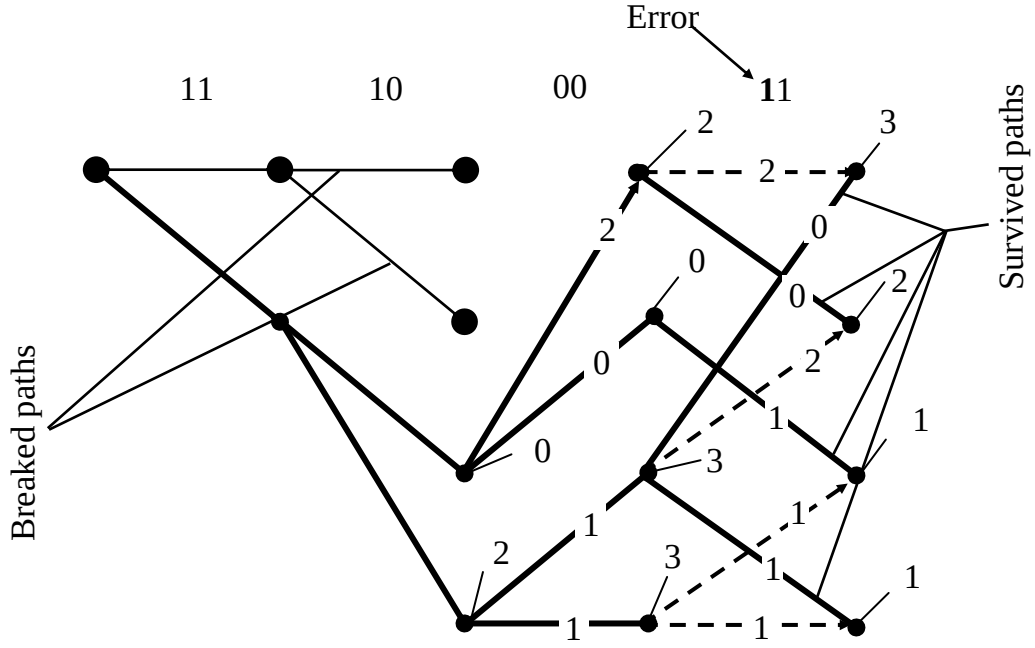


Figure 6.7, *c* – Decoding process on algorithm Viterbi

The complexity of Viterbi algorithm Realisation can be estimated by the number of branches of the code trellis created by the decoder at length of decoding *L*, tak-

ing into account the complexity of each step of a trellis (see formula (6.6)). The c**omplexity of decoder Viterbi realisation** can be estimated under the formula:

$$C = m^{(\nu + k)} \cdot L. \qquad (6.9)$$

In figure 6.8 the structure scheme of Viterbi decoder intended for work with the demodulator of signals QPSK is shown.

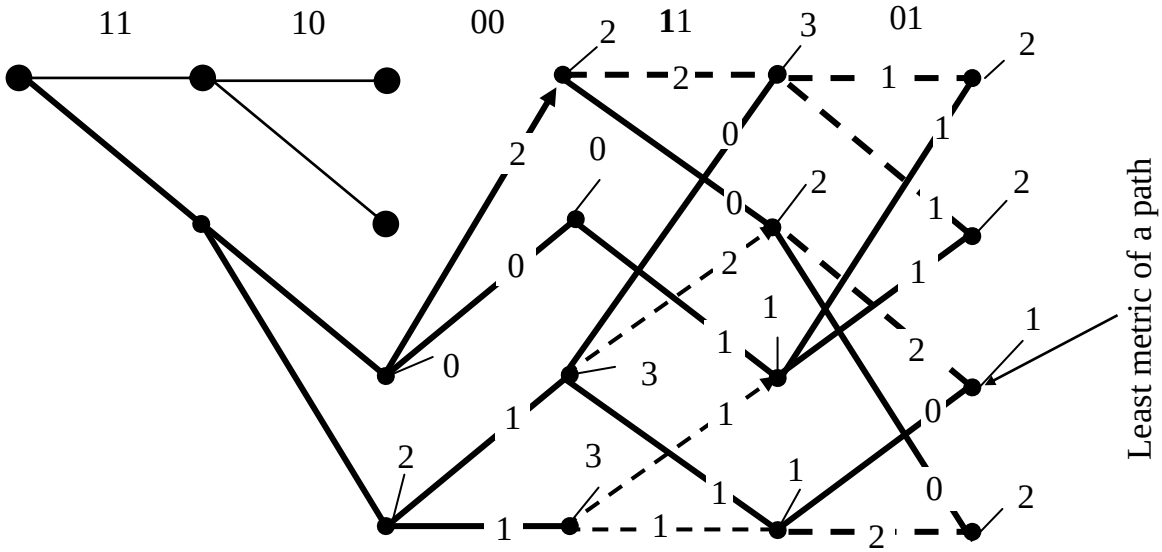


Figure 6.7, *d* – Decoding process by Viterbi algorithm

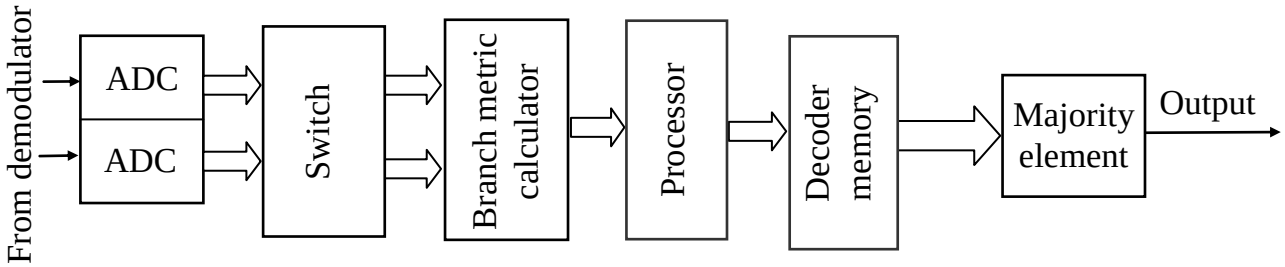


Figure 6.8 – Viterbi decoder structure scheme

The decoder consists of analog/digital converters (ADC) 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 of a signal/noise at an ADC input. More detailed description of assigning algorithms and the work of the Viterbi decoder block diagram elements are presented in the Manual [1, Section 3.8.2].

6.5 Decoding error probability of a convolutional code

The technique of a decoding of noise immunity estimation by convolutional codes is not differed from a technique stated in Section 5.6 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 play the main role. By using the probability decoding algorithm (Viterbi algorithm) the approximate expression for bit error probability looks like:

$$p_{\mathrm{d}} = \sum_{k=d_f}^{\infty} w_k P_k \,, \tag{6.10}$$

where P_k – the error probability of the path chosen on a code trellis;

w_k – spectrum of weights of erroneous paths;

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

$$P_k = Q\left(\sqrt{2kR_{\mathrm{code}}}h_{\mathrm{b}}\right). \tag{6.11}$$

The evaluations under formulas (6.10) and (6.11) show, that in the sum (6.10) by a big ratio signal/noise the first member (by $k = d_{\mathrm{f}}$) has the greatest value, and the remaining members of the sum with growth k decrease fast. Therefore, in practice we can use the simplified formula:

$$p_{\mathrm{d}} = w_{df} \cdot Q\left(\sqrt{2d_{\mathrm{f}} R_{\mathrm{code}}}h_{\mathrm{b}}\right). \tag{6.12}$$

As well as with 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 a bit error probability can be received from expression (6.11) having supposed $k = 1$, $R_{\mathrm{code}} = 1$:

$$p_{\mathrm{BPSK}} = Q\left(\sqrt{2}h_{\mathrm{b}}\right), \tag{6.13}$$

where $h_{\mathrm{b}} = E_{\mathrm{b}}/N_0$ – the ratio of the signal energy needed for the transmission of bit E_{b} to a power spectral density of a noise N_0 at an demodulator input.

Exercise 6.2. The analysis of a decoding noise immunity

Let's make calculations of a bit error probability at outputs demodulator of BPSK signals and Viterbi decoder in serial connection, using formulas (6.13) and (6.12) for the next codes:

1. Code (5, 7), $R_{\mathrm{code}} = 1/2$, $d_{\mathrm{f}} = 5$, $\nu = 2$.

2. Code (133, 171), $R_{\mathrm{code}} = 1/2$, $d_{\mathrm{f}} = 10$, $\nu = 6$.

The calculation results are given in table 6.1 and presented in figure 6.9. In the table the given values of the argument z are specified function $Q(z)$, used in formulas.

6.6 Energy coding gain

As well as in case of an estimation of a decoding noise immunity of the block codes (see Section 5.6) the concept of an energy coding gain is used in case of convolution codes.

The **energy coding gain** g is equal to a difference between values h_{b}^2 necessary to get the given error probability p by the absence and by using the code.

The values of error probability level at which the gain is defined depend on the requirements to fidelity of the transmitted digital information. For digital telephony systems an allowable level of a bit error probability usually makes $p_{\mathrm{all}} = (10^{-5} ... 10^{-6})$. In systems of digital TV transmission try to ensure $p_{\mathrm{all}} = (10^{-10} ... 10^{-11})$ which is allowable.

The value of the coding gain at the given bit error probability p_{all} can be defined by comparing the arguments of a function $Q(z)$ in formulas for error probability (6.12) and (6.13) for identical probabilities $p_d = p_{BPSK} = p_{all}$. Calculations show, that the gain depends on the level of error probability p_{all}, which defines it. It is well seen on the curves in figure 6.9 representing calculation results of the Exercise 6.2. The value of a gain with decreasing of a probability p_{all} tends to the limit which is named as **asymptotic coding gain** in the coding theory:

$$\text{ACG} = \lim g(p_{all} \to 0). \tag{6.14}$$

Comparing arguments in the expressions (11.5) and (11.4) we come to a widely used expression for ACG in logarithmic units for energy calculations in transmission systems:

$$\text{ACG} = 10\lg(R_{code} d_f) \text{ (dB)}. \tag{6.15}$$

Table 6.1 – 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	$4{,}2\cdot10^{-6}$				

As ACG is an upper bound of a gain CG, use ACG for a simple comparison and a choice of codes. Values of this ACG are often included in the code tables (see tables of the Appendix A.3). For example, in table 11.1 data about convolution codes with various lengths of a code length ν and rate R_{code} are given. The values of an ACG are shown. More detailed data are given in the tables of the Appendix A.3.

Table 6.2 – Characteristics of convolutional codes

Code rate R_{code}	Code constraint length ν = 4		Code constraint length ν = 6	
	Code	ACG, dB	Code	ACG, 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

The comparison of gain values is ensured by the cyclic coding (see table 5.10 and figure 5.11) with similar parameters for convolutional codes (see table 6.2 and figure 6.9). It shows that **convolution codes** in a combination with Viterbi decoding

algorithm **ensure considerably more gain in comparison with block codes**. It explains wide application of convolution codes in transmission systems for a noise immunity increase. The typical use of the code (133, 171) ensuring ACG = 6,99 dB at the rate R_{code} = 0,5, i.e. at twice expanding of a frequency band of the coded signal. The codecs of such code developed in the form of big integrated circuits are mass produced.

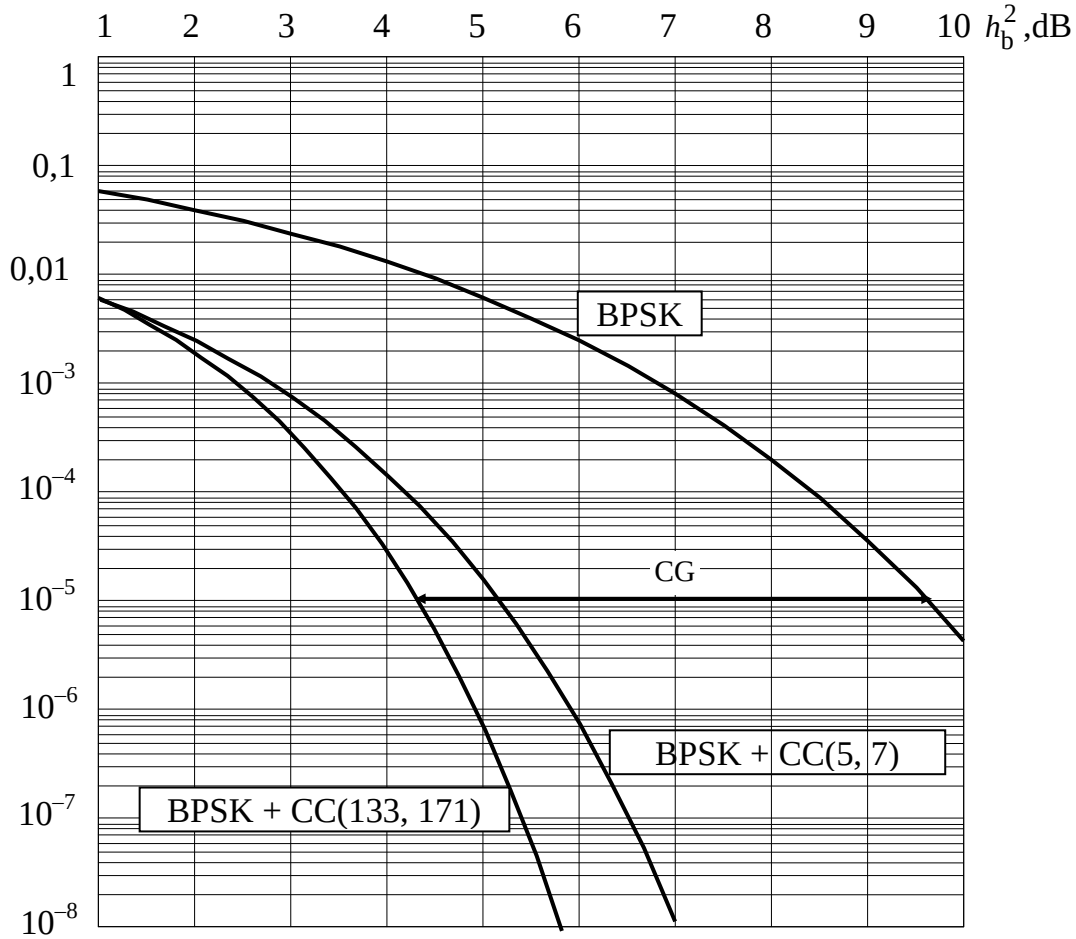


Figure 6.9 – Decoding noise immunity of a convolutional code

Questions for section 6

6.1. Name the key parameters of convolutional codes.

6.2. What are construction rules of the state diagram?

6.3. What is the connection between the state diagram and the trellis diagram?

6.4. How can we define a free distance by the state diagram?

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

6.6. How will the complexity of Viterbi decoder increase if a code constraint length is increased twice?

6.7. What is the reason of Viterbi decoder complexity raise at using a soft decision at a demodulator output?

6.8. How does the gain depend on the code constraint length?

6.9. How does the gain depend on the code rate?

Tasks

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

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

6.3. Construct the state and trellis diagrams of such a code. Show, how to define the free distance of a code on them. Find a line corresponding to this code in tables of convolutional codes from the Appendix A.3. By analogy with Exercise 6.1 analyse a connection of this code parameters with the value of free distance. Make generalising conclusions.

6.4. Prepare the trellis diagram of a code (1, 5) from the Exercise 6.1, necessary for the Viterbi algorithm analysis.

6.5. Prepare the trellis diagram of a code (1, 5) from Exercise 6.1 which is necessary for an illustration of Viterbi algorithm. As free distance of this code is $d_f = 3$ (according to the formula (6.7) the code corrects unitary errors) trace the decoding process by Viterbi algorithm if there is unitary error in the channel and establish the fact of its correction by the decoder.

6.6. Prepare the trellis diagram of a nonredundancy code (1, 0) from the Exercise 6.1 which is necessary to illustrate Viterbi algorithm. Try to explain the impossibility of error-correction by the form of the trellis diagram.

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

6.8. Using tables of the Appendix A.3 choose codes, ensuring ACG > 6 dB and specify parameters of these codes.

7 INCREASING OF DIGITAL TRANSMISSION SYSTEMS EFFICIENCY

7.1 Information, energy and frequency efficiency

Generally the result of work of a transmission systems is defined by 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 rate 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 Section 3.6). At the same time, the transmission rate cannot be above some informational resource named a channel capacity C_{ch}. A. Zuko suggested to consider the value of mean rate R_{ch} at which the given fidelity of an information transferring is ensured as one of indicators of a **system efficiency**. Thus the information system efficiency as a degree of use of a channel capacity is defined by relative value $\eta = R_{ch}/C_{ch}$. In real conditions the indicator η is always less than one. The more close η to one, the more absolute is the transmitting information system. Reaching the necessary 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 permitted to introduce the indicators: power $\beta = \dfrac{R_{ch}}{P_s / N_0}$ and frequency efficiency $\gamma = \dfrac{R_{ch}}{F_{ch}}$, which use of the mentioned resources characterizing degree. Here P_s/N_0 – the ratio of a signal power to a power spectral density of noise at a demodulate input). Thus, **efficiency indicators presented by A. Zuko** look like:

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

$$\eta = \frac{R_{ch}}{C_{ch}}; \tag{7.1}$$

Energy efficiency

$$\beta = \frac{R_{ch}}{P_s / N_0}; \tag{7.2}$$

Frequency efficiency

$$\gamma = \frac{R_{ch}}{F_{ch}}. \tag{7.3}$$

7.2 Limiting efficiency of transmission systems and Shannon bound

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

$$\eta = \frac{\gamma}{\log(1+\gamma/\beta)} \quad \text{and} \quad \gamma = \rho\beta \tag{7.4}$$

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

$$\beta = \frac{\gamma}{2^{\gamma}-1}. \tag{7.5}$$

This formula defines dependence between the energy efficiency and the frequency efficiency for the **ideal system** ensuring the equality of an information rate to a channel capacity. It is better to represent this equation in the form of a curve on a plane $\beta = f(\gamma)$ (figure 7.1, a curve «**Shannon bound**»). This curve is limiting and reflects the **best interchange between β and γ** in the continuous channel (CC).

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

$$\beta_{max} = \lim_{\gamma\to 0}\beta = \lim_{\gamma\to 0}\frac{\gamma}{2^{\gamma}-1} = \frac{1}{\ln 2} = 1{,}443. \tag{7.6}$$

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

$$\beta_{max} = 1{,}443. \tag{7.7}$$

Similar limiting curves can be constructed and for any other channels if in formulas (7.2) and (7.3) the substitute expressions for a channel capacity of the corresponding channel instead of a rate R_{ch}. So, in particular, in figure 7.1 the curve for a limiting equation $\beta = f(\gamma)$ for the discrete-continuous channel (D-CC) is shown.

It "is enclosed" in a curve of the continuous channel (CC) that confirms known result of an information theory. According to it channel capacity of D-CC is always 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 the final value and informational efficiency is less then a limiting value η_{max}. In these cases for the fixed error probability p = const it is possible to define the efficiency ratio β to γ and construct curves $\beta = f(\gamma)$.

In coordinates (β, γ) there will correspond a point on a plane to each variant of a transmission system**.** All these points (curves) **should be placed below a limiting curve** of «Shannon bound». The place of these curves depends on an aspect of signals (modulations), codes (coding methods) and a method of the elaborating of signals (demodulation/decoding). We can evaluate the system efficiency depending on a degree of approaching to the limiting values.

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

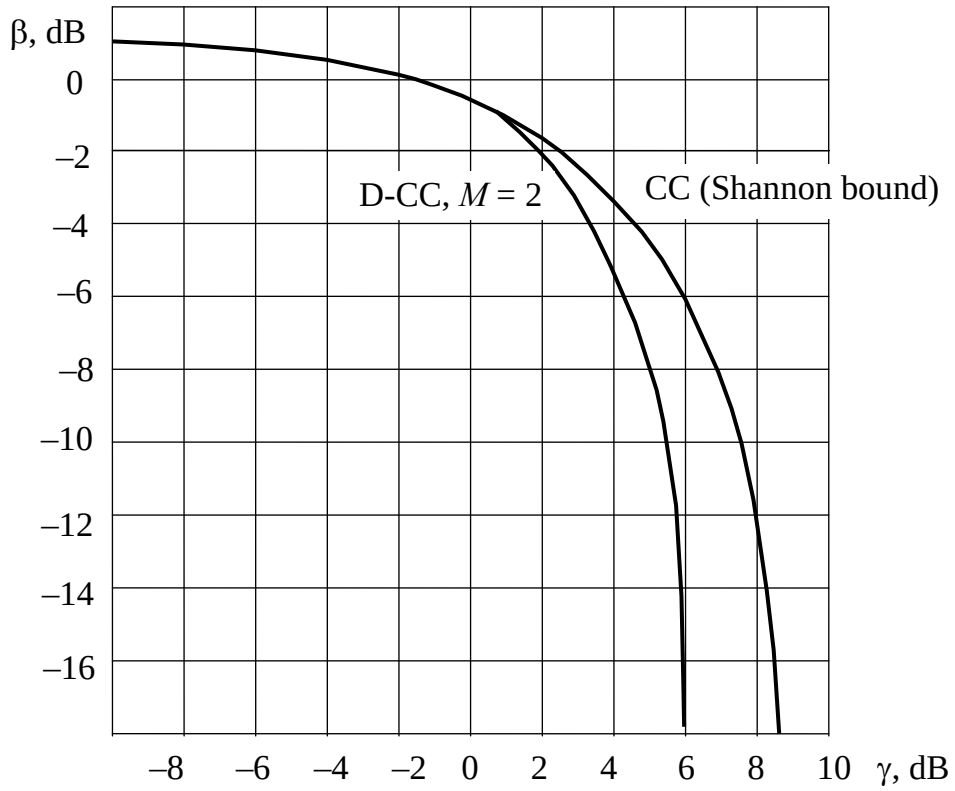


Figure 7.1 – Curves of communication systems limiting efficiency

7.3 Perspective ways of further efficiency increase

Using various methods of error-control coding considered in these Module, the designer of a transmission system having knowledge on optimisation can **flexibly change the efficiency indicators.** The designer can approach them to the limiting, potentially possible values which were obtained in the previous section. The 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 by their combinations. The choice of signals and codes in these cases is defined to construct **highly effective codems** (the codecs should correlate among themselves and modems as well). It is convenient to compare the efficiency of systems with *M*-ary signals and error-control codes with using the diagram $\beta = f(\gamma)$, presented in figure 7.1. Thus, the degree of efficiency of a modulation/coding methods can be estimated, comparing the efficiency with limiting values. The results of the efficiency analysis are presented in figure 7.2. At the same time, the various modulation/coding methods are convenient for making comparison of the "reference point" of the efficiency of transmission system with modulation QPSK (without error control coding). The method of modulation/coding with indicators $\gamma = 2$, $\beta = -9{,}6$ dB, $\eta = 0{,}47$ is the most effective and widely used among simple methods. The point representing in figure 7.2 values of efficiency QPSK is arranged in a central part of the diagram for convenient use. If an origin of coordinates is transfered to a point corresponding QPSK, in a new frame ($\Delta\beta$, $\Delta\gamma$) on a vertical axis, then the energy gain will be $\Delta\beta$ 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 in both energy and frequency efficiency.

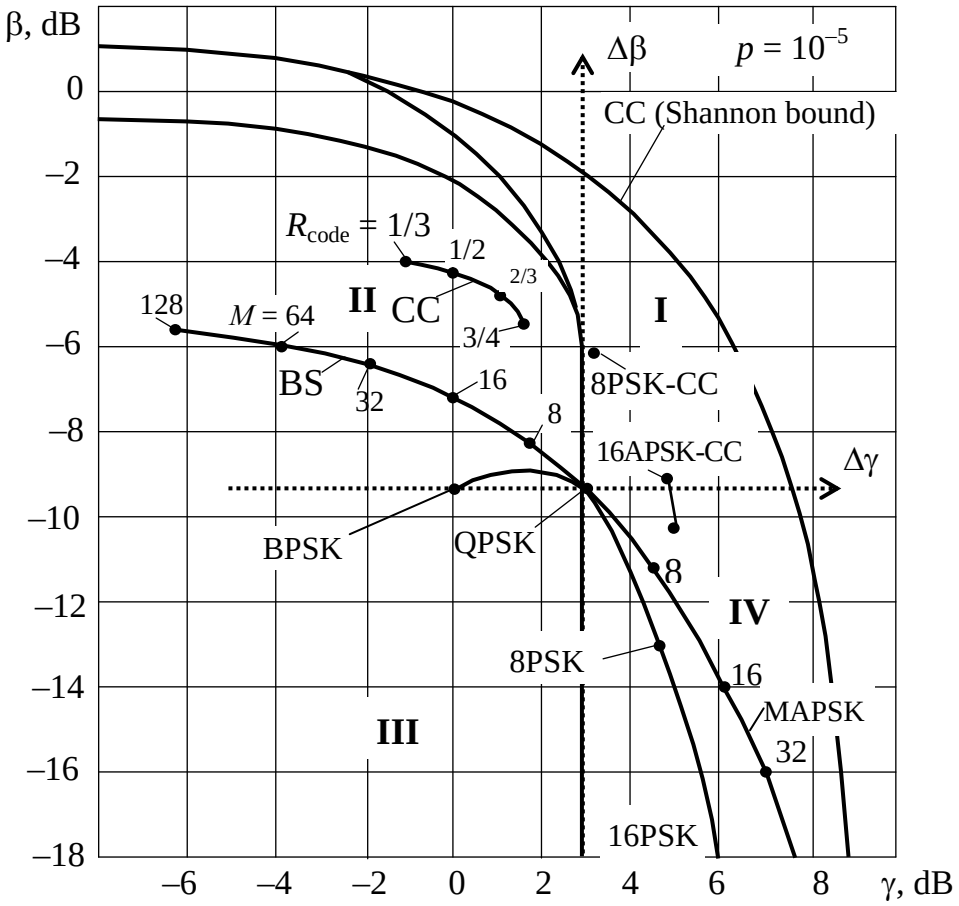


Figure 7.2 – Efficiency of *M*-ary signals and error-correcting codes

The outcomes of calculations show (figure 7.2, Quadrant I) that application of 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, the system 8PSK-CC ensures an energy gain $\Delta\beta = 2{,}8$ dB without a decreasing of a specific rate γ by the using of a convolution code with $R_{code} = 2/3$, and the system 16APSK-CC ensures a gain on a specific rate $\Delta\gamma = 2$ dB without a drop of energy efficiency β by the using of a convolution code with $R_{code} = 1/2$ and $\nu = 3$. The information efficiency of these systems is $\eta \approx (0{,}6\ldots0{,}7)$. The detailed analysis of such coded modulations is presented in Manual [2, Section 9.2]. Microelectronic progress over the last decade initiated attempts to realise the potentially possible effi-

ciency, despite the growth of decoding complexity. In 1993 **turbo-codes** were offered. Turbo-codes were described in details in Manual [2, Section 11.1] described. The intensive development of mobile transmission systems has led to the invention of a **time/space coding**, described in details in the Manual [2, Section 11.2].

APPENDIX A. PERFORMANCES OF ERROR-CORRECTING CODES

A.1 Presentation 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– number 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 cyclic codes

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

Table A.2 – Energy coding gain CG (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 Presentation 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; ACG– asymptotic coding gain (dB) by using code in channels with BPSK.

Table A.3 – Code rate R_{code} = 1/8

Code number	ν	Generating polynomials	d_{fm}	d_f	w_{df}	ACG, 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

Table A.4 – Code rate R_{code} = 1/4

Code number	ν	Generating polynomials	d_{fm}	d_f	w_{df}	ACG,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	Generating polynomials	d_{fm}	d_f	w_{df}	ACG,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	Generating polynomials	d_{fm}	d_f	w_{df}	ACG, 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

APPENDIX B. DICTIONARIES

B.1 English-Russian dictionary

accumulator	накопитель
ADC (analog-to-digital convertor)	АЦП (аналого-цифровой преобразователь)
additional symbol	дополнительный символ
ADPCM (adaptive differential PCM)	АДИКМ (адаптивная дифференциальная ИКМ)
algebraic description	алгебраическое описание
algebraic ring	алгебраическое кольцо
aliasing	наложение спектров
permitted code word	разрешенная кодовая комбинация
amplitude factor	коэффициент амплитуды
amplitude response (AR)	АЧХ (амплитудно-частотная характеристика)
antipodal signals	противоположные сигналы
a posteriori probability	апостериорная вероятность
AWGN (additive white Gaussian noise)	АБГШ (аддитивный белый гауссовский шум)
band expansion factor	коэффициент расширения полосы
binary channel	двоичный канал
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
code length	длина кода
coded modulation	сигнально-кодовая конструкция
code rate	скорость кода
code with even number of 1s	код с четным числом единиц
code word	кодовое слово (кодовая комбинация)
code word weight	вес кодового слова
coding algorithm	алгоритм кодирования
communication channel	канал связи

complexity	сложность
compound estimation	составная оценка
compression of information	сжатие информации
conditional entropy	условная энтропия
continuous code	непрерывный код
continuous source	источник непрерывных сообщений
convolutional code	сверточный код
cut-off frequency	частота среза
cyclic code	циклический код
cyclic property	циклическое свойство
DAC (digital-to-analog converter)	ЦАП (цифро-аналоговый преобразователь)
decoding method	метод декодирования
degree of compression	степень сжатия
detect	обнаружить
differential entropy	дифференциальная энтропия
digital transmission	цифровая передача
directed graph	направленный граф
discrete source	источник дискретных сообщений
DM (delta modulation)	ДМ (дельта модуляция)
double error	двукратная ошибка
dynamic programming	динамическое программирование
effective coding	эффективное кодирование
element-by-element reception	поэлементный прием
encoder state	состояние кодера
encoding method	метод кодирования
(energy) coding gain	энергетический выигрыш (от применения) кодирования
epsilon entropy	эпсилон-энтропия
epsilon rate	эпсилон-производительность
equal-length code	равномерный код
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 1s 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
frequency efficiency	частотная эффективность
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
Huffman code	код Хаффмана
hard decision	жесткое решение
hardware	аппаратное обеспечение
identity matrix	единичная матрица
information block of symbols	блок информационных символов
information characteristics	информационные характеристики
information amount	количество информации
information efficiency	информационная эффективность
instantaneous values	мгновенные значения
joint entropy	совместная энтропия
linear code	линейный код
linear combination	линейная комбинация
linear distortions	линейные искажения
low-pass filter (LPF)	фильтр нижних частот (ФНЧ)
majority decoding	мажоритарное декодирование
majority element	мажоритарный элемент
message source	источник сообщений
minimum distance of the code	кодовое расстояние
module-2 addition	сложение по модулю 2

multiplicity of errors	кратность ошибки
mutual information	взаимная информация
noise immunity	помехоустойчивость
nonlinear code	нелинейный код
nonsystematic code	несистематический код
non-uniform code	неравномерный код
odd number of 1s	нечетное число единиц
orthogonal	ортогональный
parity check matrix	проверочная матрица
path metric	метрика пути
power efficiency	энергетическая эффективность
predictor	предсказатель
primary code	первичный код
prime number	простое число
primitive code	примитивный код
probability decoding methods	вероятностные методы декодирования
protective interval	защитный интервал
pulse code modulation (PCM)	импульсно-кодовая модуляция (ИКМ)
quantization	квантование
quantization noise	шум квантования
quantization step	шаг квантования
Reed-Solomon code	код Рида-Соломона
reception in a whole	прием в целом
redundancy	избыточность
redundancy coefficient	коэффициент избыточности
row	строка
sampling	дискретизация
sequential decoding	последовательное декодирование
Shannon theorem	теорема Шеннона
Shannon-Fano code	код Шеннона-Фано
signal/(quantization noise) ratio	отношение сигнал/шум квантования
single error	однократная ошибка
slope overload	перегрузка по наклону
soft decision	мягкое решение
software	программное обеспечение

source entropy	энтропия источника
source rate	производительность источника
state diagram	диаграмма состояний
state metric	метрика состояния
statistical code	статистический код
subdivision noise	шум дробления
survived path	выживший путь
symbol rate	скорость модуляции
syndrome decoding	синдромное декодирование
systematic code	систематический код
threshold decoding	пороговое декодирование
time domain	временная область
time-space coding	пространственно-временное кодирование
transmission accuracy	точность передачи
transposed matrix	транспонированная матрица
trivial estimation	тривиальная оценка
trellis diagram	решетчатая диаграмма
turbo code	турбо код
uncorrelated messages	независимые сообщения
uniform quantization	равномерное квантование
variable-length code	неравномерный код
Viterbi algorithm	алгоритм Витерби

B.2 Russian-English dictionary

АБГШ (аддитивный белый гауссовский шум)	AWGN (additive white Gaussian noise)
автомат с конечным числом состояний	finite state machine
АДИКМ (адаптивная дифференциальная ИКМ)	ADPCM (adaptive differential PCM)
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

АЦП (аналого-цифровой преобразователь)	ADC (analog-to-digital convertor)
АЧХ (амплитудно-частотная характеристика)	amplitude response (AR)
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
взаимная информация	mutual information
временная область	time domain
выживший путь	survived path
граница	bound
двоичный канал	binary channel
двоичный код	binary code
двукратная ошибка	double error
диаграмма состояний	state diagram
динамическое программирование	dynamic programming
дискретизация	sampling
дифференциальная энтропия	differential entropy
длина кода	code length
длина кодового ограничения	code constrained length
длина регистра кодера	length of encoder shift-register
ДМ (дельта модуляция)	DM (delta modulation)
дополнительный символ	additional symbol
единичная матрица	identity matrix
жесткое решение	hard decision
запрещенная кодовая комбинация	forbidden code word
защитный интервал	protective interval
избыточность	redundancy
импульсно-кодовая модуляция (ИКМ)	pulse code modulation (PCM)
информационная эффективность	information efficiency
информационные характеристики	information characteristics
информационный блок	information block
источник дискретных сообщений	discrete source
источник непрерывных сообщений	continuous source

источник сообщений	message source
исчерпывающий поиск	exhaustive search
канал связи	communication channel
квантование	quantization
код Голея	Goley code
код Рида-Соломона	Reed-Solomon code
код с четным числом единиц	code with even number of 1s
код Хаффмана	Huffman code
код Хэмминга	Hamming code
код Шеннона-Фано	Shannon-Fano code
кодовая комбинация (слово)	code word
кодовое расстояние	minimum distance of the code
количество информации	information amount
конфигурация ошибок	errors configuration
корректирующая способность	adjusting ability
корректирующий код	error-control code
коэффициент амплитуды	amplitude factor
коэффициент избыточности	redundancy coefficient
коэффициент расширения полосы	band expansion factor
кратность ошибки	error multiplicity
линейная комбинация	linear combination
линейные искажения	linear distortions
линейный код	linear code
мажоритарное декодирование	majority decoding
мажоритарный элемент	majority element
метрика ветви	branch metric
мгновенные значения	instantaneous values
метод декодирования	decoding method
метод кодирования	encoding method
метрика пути	path metric
метрика состояния	state metric
мягкое решение	soft decision
накопитель	accumulator
наложение спектров	aliasing
направленный граф	directed graph

независимые сообщения	uncorrelated messages
нелинейный код	nonlinear code
непрерывный код	continuous code
неравномерный код	non-uniform, variable-length code
несистематический код	nonsystematic code
нечетное число единиц	odd number of 1s
обнаруживать	detect
однократная ошибка	single error
ортогональный	orthogonal
основание кода	size of a code alphabet
отношение сигнал/шум квантования	signal/(quantization noise) ratio
перегрузка по наклону	slope overload
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
предсказатель	predictor
прием в целом	reception in a whole
примитивный код	primitive code
проверочная матрица	parity check matrix
проверочное соотношение	check relation
программное обеспечение	software
производительность источника	source rate
пропускная способность канала	channel capacity
простое число	prime number
пространственно-временное кодирование	time-space coding
противоположные сигналы	antipodal signals
равномерное квантование	uniform quantization
равномерный код	equal-length, fixed length code
разрешенное кодовое слово	permitted code word
расстояние по Хэммингу	Hamming distance

решетчатая диаграмма	trellis diagram
сверточный код	convolutional code
свободное расстояние	free distance
сжатие информации	compression of information
сигнально-кодовая конструкция	coded modulation
синдромное декодирование	syndrome decoding
систематический код	systematic code
скорость кода	code rate
скорость модуляции	symbol rate
сложение по модулю 2	module-2 addition
сложность	complexity
совместная энтропия	joint entropy
способность исправлять ошибки	error correction capability
способность обнаруживать ошибки	error detection capability
составная оценка	compound estimation
состояние кодера	encoder state
статистический код	statistical code
степень сжатия	degree of compression
строка	row
теорема Шеннона	Shannon theorem
точность передачи	transmission accuracy
транспонированная матрица	transposed matrix
тривиальная оценка	trivial estimation
турбо код	turbo code
условная энтропия	conditional entropy
фильтр нижних частот (ФНЧ)	low-pass filter (LPF)
ЦАП (цифро-аналоговый преобразователь)	DAC (digital-to-analog convertor)
циклический код	cyclic code
циклическое свойство	cyclic property
цифровая передача	digital transmission
частота среза	cut-off frequency
частотная эффективность	frequency efficiency
четное число единиц	even number of 1s symbols
шаг квантования	quantization step
шум дробления	subdivision noise

шум квантования	quantization noise
энергетическая эффективность	power efficiency
энергетический выигрыш (от применения) кодирования	(energy) coding gain
энтропия источника	source entropy
эпсилон-производительность	epsilon rate
эпсилон-энтропия	epsilon entropy
эффективное кодирование	effective coding

REFERENCES

1. **Банкет В.Л.** Цифровые методы передачи информации в спутниковых системах связи: учебн. пособ. / Банкет В.Л., Иващенко П.В., Геер А.Э. – Одесса: УГАС, 1996. – 180 с.

2. **Банкет В.Л.** Дискретная математика в задачах теории цифровой связи: учебн. пособ. / Банкет В.Л. – Одесса: ОНАС, 2008. – 118 с.

3. **Питерсон У.** Коды, исправляющие ошибки / У. Питерсон, Э. Уэлдон, пер. с англ. под ред. Р.Л. Добрушина. – М.: Мир, 1976. – 594 с.

Education publication

Ivaschenko Peter Vasilyevich
Rozenvasser Denis Mikhailovich

TELECOMMUNICATION THEORY

Textbook
Part 2

Signed for printing 17.02.17
Format 60/88/16 Order № 5982
Number of copies 100. Number of press sheets 4.
Printed by the publishing RISO company equipment
Printed at the Editorial and Publishing Center of O. S. Popov ONAT