Програмний Комплекс для Дослідження і Порівняння Завадостійких Кодів

Юрій Гунченко
кафедра математичного забезпечення комп'ютерних систем
Одеський національний університет імені І.І.Мечникова
Одеса, Україна
7996445@mail.ru

Павел Ємельянов
кафедра математичного забезпечення комп'ютерних систем
Одеський національний університет імені І.І.Мечникова
Одеса, Україна
ipauler@gmail.com

Тетяна Щербакова
кафедра математичного забезпечення комп'ютерних систем
Одеський національний університет імені І.І.Мечникова
Одеса, Україна
ll_papilion@mail.ru

Вікторія Левчук
кафедра математичного забезпечення комп'ютерних систем
Одеський національний університет імені І.І.Мечникова
Одеса, Україна
levchuk.viktoria@stud.onu.edu.ua

Software for Research and Comparison of Noise-immune Codes

Yurii Gunchenko
department of mathematical support of computer systems
Odessa I.I. Mechnicov National University
Odessa, Ukraine
7996445@mail.ru

Pavel Emelyanov
department of mathematical support of computer systems
Odessa I.I. Mechnicov National University
Odessa, Ukraine
ipauler@gmail.com

Tetyana Shcherbakova
department of mathematical support of computer systems
Odessa I.I. Mechnicov National University
Odessa, Ukraine
ll_papilion@mail.ru

Viktoria Levchuk
department of mathematical support of computer systems
Odessa I.I. Mechnicov National University
Odessa, Ukraine
levchuk.viktoria@stud.onu.edu.ua

***Анотація*—В роботі запропоновано концепцію побудови програмного комплексу для дослідження завадостійких кодів з наявністю і без наявності помилок у повідомленні. Розглянуто необхідні модулі комплексу програмного забезпечення, які повинні здійснювати кодування, декодування повідомлень, внесення помилок, модулі зберігання, обробки та виводу даних, наведено умови і алгоритми їх функціонування. Показано доцільність використання OLAP-систем для зберігання і обробки отриманих даних. Наведено результати функціонування комплексу програмного забезпечення, створеного згідно концепції, що пропонується, на прикладі дослідження характеристик чотирьох кодів двох різних типів.**

***Abstract*—The paper introduces the concept of development a software system for research of noise-immune codes with and without errors in the message. This paper presents the necessary modules of software complex, that provide encoding, message decoding, introducing errors, data storage, processing and output modules and the operating conditions and algorithms as well. The expediency of using OLAP-systems for storing and processing of received data are shown in the work. The results of software complex operating, that was created under proposed concept, in terms of study the characteristics of four codes of two different types, is shown in the work.**

Ключові слова—завада, перешкода, завадостійкий код, програмне забезпечення, алгоритм побудови програмного модуля, база даних, OLAP-система

Keywords—noise; noise-immune code; software; software module creating algorithm; database; OLAP-system

I. Introduction

Obstacles and noises are undesirable factors of destructive character in different technical systems, because it interferes with the normal operation of such systems. It forces the creators of computer systems, to take certain actions to minimize the impact of noise on a particular system.

Because of the different nature of its appearance, noise can alter the data transmission between different computer systems or within the one system.

Noise immunity is a property of the system of performing its functions the most effectively if noise and distortion are available. Noise immunity is evaluated by the intensity of noises in which the dysfunction of the system (device) does not exceed the permissible limits. The stronger noise at which the system continues to operate properly and effectively is, the higher is its immunity.

II. The Review Of Noise-Immune Devices

A. Selecting a Template.

There is a wide range of outfits, devices and equipment, where resistance to interference and noise is essential for their proper operating. This may include personal computers, navigation systems, satellite communications, ballistic missiles.

Most often term “noise-immunity” is used to characterize the information transmission devices (for example, communication lines) or surveillance devices (for example, radars). For this reason in most cases, the term “signal” could be determined, and the noise-immunity evaluation can be made by considering the relation between interference and signal, at which the required quality of normal operation is provided. For example, in the radiolocation – the relation signal to the noise at which the specified detection reliability is provided (probability of correct detection at a certain probability of false alarm).

Building up of optimal devices that implement the required level of noise-immunity is usually very difficult, and their inevitable technical imperfections do not allow to reach this level in full. That is why the devices at the greatest simplicity which provide good approaching to the optimum level are applied for.

There are many ways to reduce the impact of noise on transmitted or stored data. The means of noise-immune coding based on certain noise-immunity codes are the most frequently used.

The work objective is to develop the concept and principles of design software to explore methods of encoding and decoding information in the presence and absence of errors. This software package will allow providing virtual research of systems that are being developed without their direct full-scale modeling or creation of prototypes.

III. Requirements Of Complex

Complex of software should be as a basis for analysis codes such as characteristics of code as follows: encoding, codeword size and length of the original character sequence. It should be noted that this system is designed to compare the different codes with each other. Its function capabilities should contain comparison of different codes in the same conditions. This approach allows getting adequate and correct information about the difference between the opportunities of codes, depending on their properties.

This complex is proposed to implement not as a monolithic system, but in the form of several discrete modules that are interconnected, where each module will perform its purpose. Interaction between the modules should be partially automated and partially supported by the operator. This solution allows you to make the software package more flexible, because such implementation facilitates rapid adjustments by changing the functionality in one or another part of the complex. Thus, the implementation of changes will cause relatively lower costs than the introduction of changes into a single monolithic system.

A. Algorithm Of Operation.

Let us describe the life cycle of a message that is processed by software complex, in the form of algorithm:

Message in a symbolic form is submitted in module that performs encoding (module "CODER").

Module "CODER" creates a binary representation of characters of the message based on their serial number in the ASCII table.

Module "CODER" performs apropriate to right code (for example Hemming code) algorithmic operations and writing data about the length of word that is encoded, and encode time in a log-file.

"CODER" sends the resulting encoded binary sequence to module "DECODER" through the communication channel.

Unit generating errors works where appropriate

Module "DECODER" processes received messages with necessary algorithms and writes data in log-file that are similar to data of encoding side.

Based on log-files of modules "CODER" and "DECODER", the recording into the database is implemented.

The database is processed using functional capabilities of OLAP-analyzer.

A graphical representation of the data is carried out.

Fig. 1 shows the diagram that demonstrates the interaction between the modules.

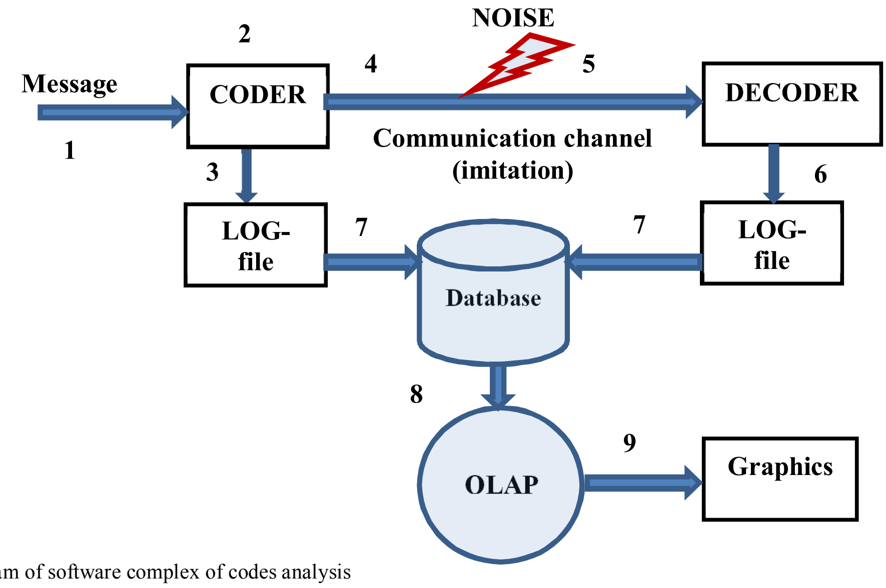

Fig. 1 Functional diagram of software complex of codes analysis

B. Design of "CODER" module

The module of coding messages for its intended use should perform the following functions:

translation of character sequences in a binary sequence consisting of symbols **1** and **0**;

coding of binary sequence using the appropriate algorithm and transferring encoded sequence through the communication channel that is imitated, precisely from the decoding side;

data output about the encoding sequence, and time spent on the process of coding this sequence.

Under the real conditions, encoding and decoding operations are often implemented using hardware. In the software complex, the analysis of different types of codes used for noise-immune coding will be investigated in software that imitates the hardware part.

It is assumed that the three above capabilities of this module will be implemented through the mechanism of using functions, such as writing discrete, logical coherent sections of a code, which can be called many times afterwards, transferring them the necessary data to process, or fully edit and modify.

This method will allow performing the whole spectrum of the module functionality in the form of software.

C. Design of generating errors module.

Under the conditions of software system, generating error module is an essential element for carrying out the code analysis. When transmitting information to the real conditions through the real communication channel, information can change due to interference, which in its turn can cause changes in the symbols in the transmitted sequence. As this complex does not involve information exchange between encoding and decoding parties by means of the communication channel, the coded sequence is devoid of risk to be changed.

It is good with hardware encoding, however, as the software system is developed for the analysis and comparison of codes and their capabilities, the wider range of information for analysis can be received having information about how codes correct errors.

Based on the above, there is a need to generate errors to test the time of algorithm work while correcting these errors. The module of generating errors will be considered as a generator of those errors. Processing of the transmitted coded message will be performed by generating errors block using the following algorithm:

an encoded sequence comes from the module "CODER" to block of generation of errors;

an error generating block changes the information bit of encoded message;

an error generating block transmits a coded binary sequence to the next module "DECODER"

This algorithm is quite simple, but we should note a few points. Using generation errors block will apply to encoded sequence strictly if it is necessary. So, it is assumed that each encoded sequence is transmitted without errors first, and then with error in the information bit that is generated as a result of the work of generating error module. Subsequently, the results of the code when decoding encoded binary sequence without errors and sequence with generated error will be recorded in the log-file at work of the module "CODER".

D. Design of "DECODER" module

The decoding module of encoded binary sequence is a module that receives data from the module "CODER" or generating error module.

Due to its appointment, module "DECODER" should perform the following functions:

receive the input encoded binary sequence from the generating error module or directly from the module "DECODER" passing generating error module;

implement decoding of the received encoded binary sequence consisting of symbols 1 and 0 according to the required decoding algorithm;

submit the output decoded sequence of characters and to input the required data in the log-file.

It should be noted that the decoding algorithm must comply with the encoding algorithm, not only by their type (e.g. Hemming code or CRC), but also by the key characteristics such as length codeword and the number of control symbols. Data recorded in the log file by module "DECODER" are completely the same as the data recorded in the log file of module "CODER" – precisely word length and decoding time. Subsequently, these data will be required to analyze and compare the effectiveness of various codes with different characteristics.

E. Design of database and its write action module

One of the key parts of the software complex is a database that will contain the information from log files of "CODER" and "DECODER" modules. One module should include both the database and the software that implements the data filling based on log-files.

The first step is to determine the database structure, such as its tables and fields in these tables. The database consists of five tables. These tables (for example) **Sequences**, **CodeTypes**, **Encodes**, **Decodes** and **DecodesEr**, intended for recording sequences data to encoding, algorithms and results of encoding, decoding and decoding with error, thereafter. The information required for the analysis should be entered into these tables.

The database structure is presented in Fig. 2.

Also, it is necessary to describe the functional capabilities of the program, intended to fill the database by available information. It can be made as an algorithm:

the program must read data from log files;

give the read data to the appropriate format to recording in a database;

initiate a connection to the database;

record in the appropriate fields of tables.

This algorithm will allow distributing data from log-files in appropriate table field most reliably. Also, the use of object-oriented language in this module will allow recording the necessary data in the database fast and on relatively high level.

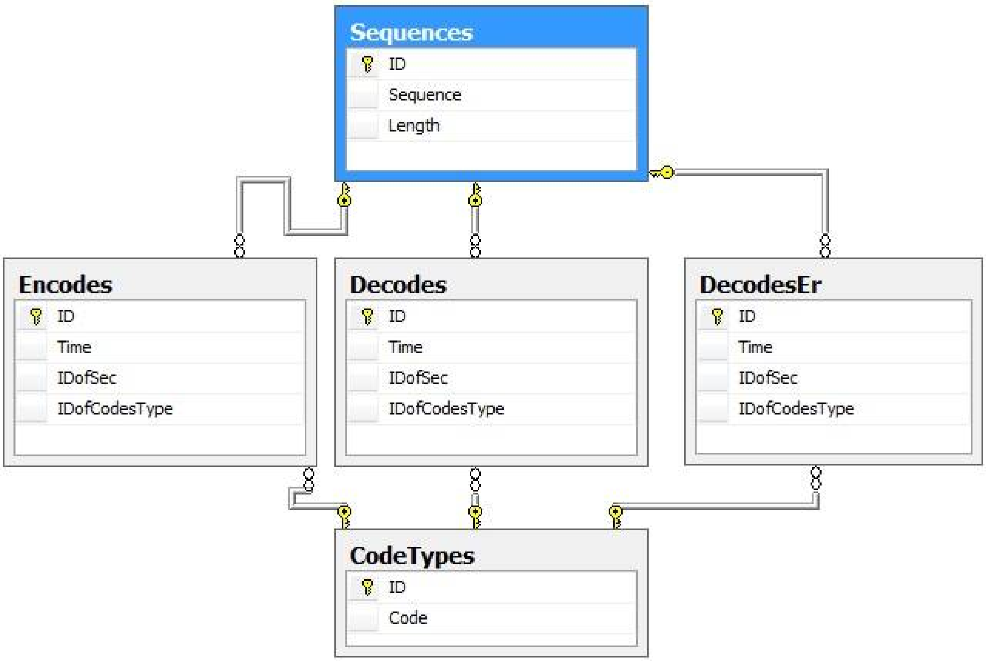

Fig. 2 The database structure

F. Design of OLAP tools

OLAP-systems allow working with data, precisely representing them in aggregated form for further convenience work with it. Also, OLAP-systems help to take the required decision that should be taken in the future, based on existing statistics. Often OLAP-systems are not separate solutions but integrated into other systems, such as decision support systems.

As data storage for OLAP-systems, data sources of different types can be used. The relational database MsSQL will be used for the proposed software module

G. Design of visual data presentation tools

The final step in the design of this software complex is the visual representation of data.

This step is the most simple in terms of implementation, because of the appropriate use of the finished software.

Part of the work that shows the different types of codes will be efficiently implemented by the form of graphs based on the received data.

Data that will be presented in graphs, will be received from the previous stage, where OLAP-component of this complex is implemented, which, in its turn, will handle the processing of information from the database.

This approach allowed implementing the software system, without negative affecting its quality, by the least costly way.

IV. Results of Operation and Test of Adequacy

To test the adequacy of the proposed approach of designing software complex, with its help, a research and analysis of the two types of codes that are widely used – Hamming code and Inverse code were conducted. Characteristics of the analyzed codes are shown in Table 6.

TABLE I. CHARACTERISTICS OF THE ANALYZED CODES

Code type	Hamming code	Hamming code	Inverse code	Inverse code
The number of information bits	8	16	8	16
General number of bits	12	21	16	32
Number of control bits	4	5	8	16
Specificity of work with errors	Corrects 1 error	Corrects 1 error	Checks for errors	Checks for errors
Denotation	HEM(8,12,4)	HEM(16,21,5)	INV(8,16,8)	INV(16,32,16)

Thus, four codes of two types are involved in comparison. Also, Hamming code has the ability to correct one error and inverse code operates on the principle of validation of the data.

General analysis of the code that is used further is a comparing of temporal characteristics of each code on such characteristics as time of encoding the information symbol sequence, decoding of coded sequence and time of decoding of symbol sequence with an error.

The result of the analysis becomes data shown graphically in Fig. 3 and Fig. 4.

Let us compare the four codes of two types, precisely Hamming code and Inverse code on parameters such as time of encoding, time of decoding and time of decoding with error. These operations were performed on eight information symbolic sequences of varying lengths – from 2 to 512 bits.

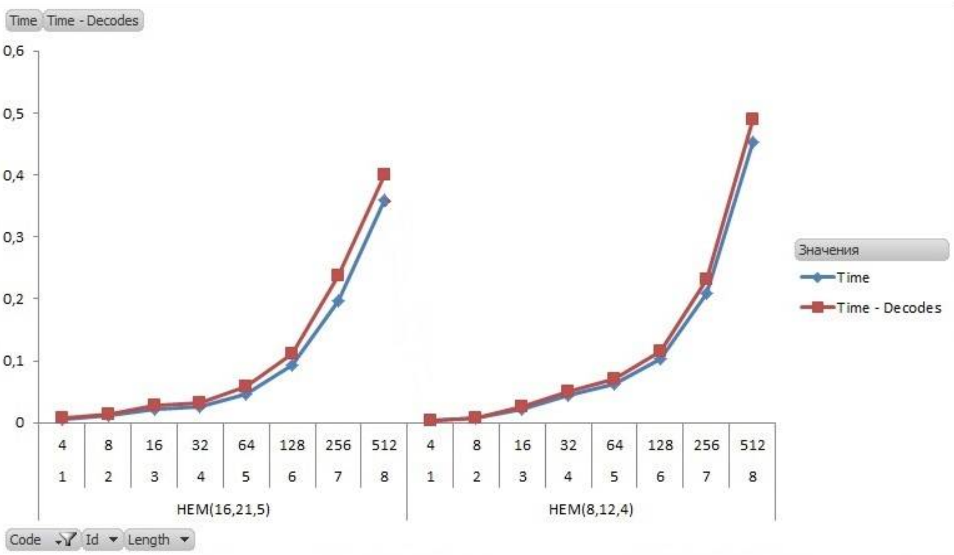

Fig. 3 Results of the study Hamming code

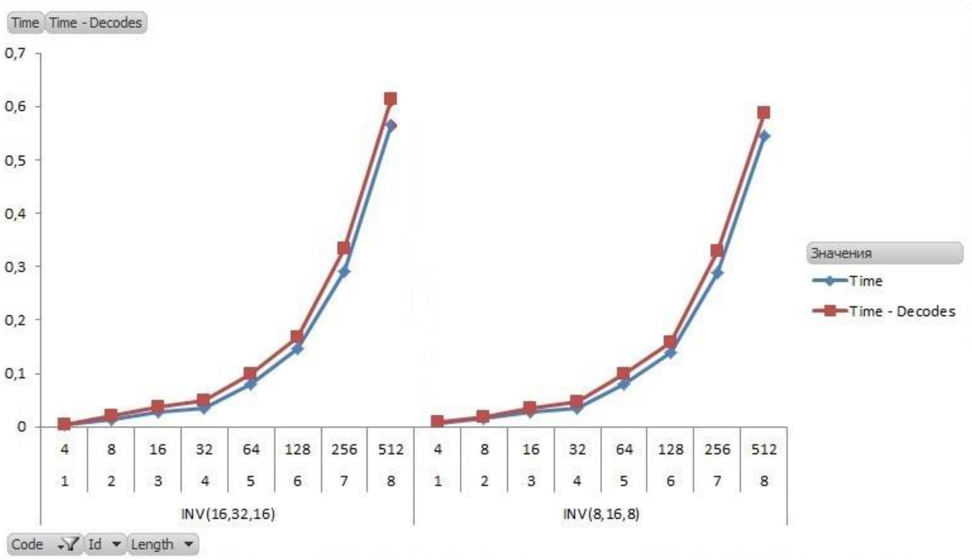

Fig. 4 Results of the study Inverse code

As a result of the software complex work, the advantages and disadvantages of each of the four codes were shown, such as code INV (16, 32, 16) by the number of bits of information and control bits twice those for the code INV (8, 16, 8). However, it performs the necessary operations in almost the same lap times as the code INV (8, 16, 8). This is caused by the unreasonableness of using this type of code in comparison with the Hamming code. The situation in which specific features of inversion code are necessary may be considered as the exception.

It was also determined that the Hamming code HEM (16, 21, 5) spends the least time on a work. This code in this situation can be called the most appropriate in terms of run-time encoding and decoding operations. It should also be noted that the code HEM (16, 21, 5) is a version of the code HEM (8, 12, 4), but with the increased numerical parameters. As the results of studies of these codes, code HEM (16, 21, 5) is much faster than code HEM (8, 12, 4). This allows suggesting that with increase characteristics in this type of code, the time of coding will be reduced. In other words, with the increase of number of information bits and control bits, productivity of the code will be increased too.

V. Conclusions

In this paper the concept of development was proposed and software complex to noise-immunity testing and analysis codes was implemented. The complex includes four components, and also two-party component in the form of software packages were involved.

This software complex allowed conducting a test comparison of the four codes of two types, precisely two implementations of the Hamming code and two implementations of Inversion code. During the testing and further analysis, the advantages and disadvantages of the investigated codes were identified. The advantage of this approach is the fact that the analysis was based on actual data obtained as a result of work of each code.

Also, it should be noted that this complex is only intended to emulate the actual processes of encoding and data transmitting, because encoding and decoding operations are implemented in software level. There is a possibility that the hardware implementation of encoding and decoding operations may vary from implementation presented in this paper.

However, with similar approaches in the implementation of both software and hardware, the similar nature of the relationships should be expected.

In addition, the results obtained in this work are based on the usage of software complex for code testing and analysis, are relative because they were obtained as a result of work of a specific computer with specific characteristics. Therefore it is fair to say that the time spent on encoding and decoding by similar codes, can change in one direction or another, depending on the technical characteristics of the platform.

One of the major advantages of the concept of software system is its versatility in terms of using other codes. It allows you to test, analyze and compare different implementations of different noise-immune codes under the same conditions. This result is achieved through the specific of call of modules functions that encode and decode: mechanism of function having almost nothing to do with the implementation of the function and allows you to give the same consistency to the input of many different functions.

Література References

[1] Lineynie blochnie kodi [Electronic resource] - Access mode: http://sernam.ru/book_tec.php?id=80.

[2] Kodi s obnarujeniem oshibok [Electronic resource] - Access mode: http://informkod.narod.ru/5_1item.htm.

[3] Kutuzaki A.S., Gunchenko Y.A. Analiz problem razbieniya na kadri dannih // Abstracts of the Tenth All-Ukrainian conference of students and young professionals "Informatics, information systems and technology". – Odessa: - 2013. – p. 193 – 195.