МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Кваліфікаційна наукова
праця на правах рукопису

Харін Олександр Олександрович

ДИСЕРТАЦІЯ

МЕТОДИ ТА ЗАСОБИ ІНТЕГРОВАНОГО ЗАХИСТУ ІНФОРМАЦІЇ В ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМАХ МНОЖИННОГО ДОСТУПУ НА ОСНОВІ ФАКТОРІАЛЬНОГО КОДУВАННЯ ДАНИХ

123 – комп'ютерна інженерія

Подається на здобуття ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

О.О. ХАРІН

Науковий керівник:
Фауре Еміль Віталійович
доктор технічних наук, доцент

Черкаси – 2020

АНОТАЦІЯ

Харін О.О. Методи та засоби інтегрованого захисту інформації в телекомунікаційних системах множинного доступу на основі факторіального кодування даних. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 123 «Комп'ютерна інженерія». – Черкаський державний технологічний університет, Черкаси, 2020.

Дисертаційна робота спрямована на вирішення актуальної науково-технічної задачі, що полягає в підвищенні достовірності передавання інформації, забезпеченні її конфіденційності та цілісності на основі використання факторіального кодування даних. Ця задача передбачає необхідність удосконалення існуючих методів факторіального кодування, створення методу побудови телекомунікаційних систем на основі нероздільного факторіального кодування з динамічним розподілом ресурсів між користувачами системи в режимі реального часу.

У роботі проведено аналіз існуючих методів факторіального кодування, зокрема факторіального коду з відновленням даних за перестановкою, який показав, що вони вразливі до помилок парної кратності, таких, що призводять до трансформації однієї перестановки з дозволеної множини в іншу перестановку, що належить цій же множині. З метою підвищення стійкості факторіальних кодів до помилок парної кратності, сформульовано задачу роботи, яка полягає у розробці методів підвищення достовірності передавання даних у системах з нероздільним факторіальним кодуванням.

Ще одним напрямком досліджень, пов'язаним з факторіальним кодуванням, є процес формування сигнально-кодової конструкції для нероздільних методів факторіального кодування, зокрема факторіального кодування з відновленням даних за перестановкою, що дозволило б виявляти та виправляти помилки малої кратності. З метою забезпечення бажаного рівня достовірності передавання даних та максималізації швидкості коду, було сформульовано наступну задачу дисертаційної роботи, яка полягає у розробці

методу формування сигнально-кодової конструкції для систем *з* факторіальним кодування з відновленням даних за перестановкою.

Розглянуто існуючі методи кодування мовленнєвих сигналів. Проаналізовано методи кодування на основі згорткових кодів та методи на основі каскадних кодів. Аналіз згорткових кодів показав, що сформована кодова послідовність містить інформаційні символи у відкритому вигляді, що накладає обмеження на використання цих кодів під час передавання конфіденційної інформації, а також мають швидкість коду, яка не перевищує 1/2, тобто одному інформаційному символу ставиться у відповідність два символи кодової послідовності. У той же час, каскадні коди, за рахунок послідовного використання декількох кодів, дозволяють вирішити проблему захисту інформації від несанкціонованого читання, але вносять додаткову надлишковість, а також збільшують апаратні витрати. Виконаний аналіз дає змогу чітко сформулювати наступну задачу дисертаційного дослідження, яка полягає у розробці методів декодування факторіальних кодів з відновленням даних за перестановкою для систем передавання мовленнєвої інформації реального часу.

Виконано аналіз існуючих методів побудови телекомунікаційних систем з множинним доступом на базі мультиплексування пакетів та часового розподілу. Аналіз цих методів показав, що більшість з них передбачають наявність службових пакетів для реалізації механізму резервування / звільнення часових інтервалів під передачу даних, що зменшує загальну пропускну здатність каналу. Також у цих методах не реалізовано єдину процедуру забезпечення інтегрованого захисту інформації. Виходячи з цього, сформульовано ще одну задачу даної роботи, яка полягає в розробці методу побудови телекомунікаційних систем множинного доступу з використанням нероздільного факторіального кодування.

В рамках дисертаційного дослідження вперше розроблено методи підвищення достовірності передавання даних у системах з факторіальним кодування з відновленням даних за перестановкою, які за рахунок доповнення

перестановки бітами ознак та комбінування завадостійких кодів дозволяють підвищити здатність нероздільних факторіальних кодів до виявлення помилок. Розроблено алгоритм факторіального кодування з додатковими перевірними бітами. Реалізація алгоритму дає змогу зменшити ймовірність невиявленої кодом помилки, за незалежних бітових помилок, у порівнянні з факторіальним кодом з відновленням даних за перестановкою з доповненням на 2-4 порядки для $M = 8$ у залежності від ймовірності бітової помилки за рахунок незначної втрати в швидкості коду. Також розроблено алгоритм факторіального каскадного кодування. Реалізація алгоритму дає змогу на порядок зменшити ймовірність невиявленої кодом помилки, за незалежних бітових помилок, у порівнянні з ФКВД для $k = 15$ у залежності від ймовірності бітової помилки, при цьому за довжини блоку даних $k \geq 128$ швидкість коду не змінюється.

Вперше розроблено метод формування сигнально-кодової конструкції для систем з факторіальним кодування з відновленням даних за перестановкою, який за рахунок побудови решіток з високою щільністю розташування сигнальних векторів дозволяє максимізувати швидкість коду для заданого рівня достовірності передавання даних та відповідної мінімальної відстані між вузлами решітки. Побудовано решітки, вузлами якої є перестановки, що дозволяють виявляти помилки кратності $t \leq 5$ і виправляти помилки кратності $t \leq 2$ з максимальною швидкістю коду для $M = 5$, $k = 3$, $N_{sv} = 10$ та $M = 6$, $k = 5$, $N_{sv} = 60$.

Крім того, отримав подальший розвиток метод факторіального кодування з відновленням даних за перестановкою, який за рахунок відновлення даних з втратами на основі метрики Хеммінга або методу лінійної інтерполяції дозволяє забезпечити захист мовленнєвої інформації від помилок каналу зв’язку та несанкціонованого доступу в телекомунікаційних системах реального часу. Розроблено структурну схему пристрою декодування та алгоритм виправлення помилок з втратами у нероздільних факторіальних кодах в метриці Хеммінга, що забезпечує рівень шуму декодування, який не перевищує значень рівня комфортного шуму за ймовірності бітової помилки

$p_0 \leq 10^{-2}$ у каналах з незалежними бітовими помилками. Розроблена схема дозволяє апаратно реалізувати пристрій декодування факторіальних кодів в метриці Хеммінга. Розроблено структурну схему пристрою декодування та алгоритм виправлення помилок з втратами у нероздільних факторіальних кодах методом лінійної інтерполяції, що забезпечує рівень шуму декодування, який не перевищує значень рівня комфортного шуму за ймовірності бітової помилки $p_0 \leq 10^{-2}$ у каналах з пакетуванням бітових помилок. Розроблена схема дозволяє апаратно реалізувати пристрій декодування факторіальних кодів методом лінійної інтерполяції.

В заключній частині дисертаційного дослідження вперше розроблено метод побудови систем множинного доступу на основі нероздільного факторіального кодування, який за рахунок використання факторіального коду з заданим числом інверсій та виділення кожному з користувачів підмножини перестановок з певним набором властивостей забезпечує інтегрований захист інформації від несанкціонованого доступу та помилок каналу зв'язку, а також циклову синхронізацію без застосування окремого каналу синхронізації. Розроблено структурну схему та алгоритм роботи мультиплексора каналів зв’язку на основі факторіального коду з заданим числом інверсій, що не потребує організації синхронізуючого каналу та підтримує циклову синхронізацію за робочим сигналом. Розроблена схема дозволяє апаратно реалізувати мультиплексор каналів зв’язку. Розроблено структурну схему телекомунікаційної системи з множинним доступом з використанням мультиплексора каналів зв'язку на основі факторіального коду з заданим числом інверсій, розроблено алгоритм динамічного розподілу ресурсу пропускної здатності каналу залежно від значення навантаження, що створюється кожним з користувачів системи. Розроблена схема дозволяє апаратно реалізувати телекомунікаційну систему з множинним доступом на основі факторіального кодування.

Ключові слова: захист інформації, конфіденційність, цілісність, достовірність передавання даних, факторіальне кодування, сигнально-кодова

конструкція, система множинного доступу.

SUMMARY

Kharin O. Methods and means of integrated protection of information in telecommunication systems with multiple access based on data factorial coding. – Qualified scientific work on the rights of the manuscript.

Thesis for a Doctor of Philosophy degree in specialty 123 "Computer engineering". – Cherkasy State Technological University, Cherkasy, 2020.

The dissertation is aimed at solving the actual scientific and technical problem of increasing the reliability of data transmission and ensuring their integrity using factorial codes that provide integrated protection of information (simultaneous protection against communication channel errors and protection against unauthorized modification and / or unauthorized access). This task includes the factorial codes improvements, create a method of construction of telecommunication systems that use these codes and have new features such as error correction with losses and dynamic allocation of resources between users of the system in real-time.

The paper analyzes the existing methods of factorial coding, including factorial code with data recovery by permutation, which showed that they are vulnerable to errors pair multiplicity which leading to a transformation of a permutation of the allowed set to another permutation, owned by the same set. To increase the robustness of factorial codes to even multiplicity errors, the task of the research is to develop the methods for improving the reliability of data transmission in systems with inseparable factorial coding.

Another area of research related to factorial coding is the process of forming a signal-code structure for inseparable methods of factorial coding, in particular, factorial code with data recovery by permutation. The following task of the dissertation is to develop the method of construction of signal-code structure for systems with factorial code with data recovery by permutation.

The existing methods of encoding speech signals are considered. Methods of coding based on convolutional codes and methods based on cascading codes are

analyzed. Analysis convolutional codes showed that formed coding sequence contains unprotected information symbols, which restricts the use of these codes for confidential information exchange. Also, convolutional codes have speed code which no more than 1/2. At the same time, cascading codes, through the consistent use of several codes, can solve the problem of protecting information from unauthorized reading, but they add additional redundancy and also increase hardware costs. The performed analysis allows to clearly formulate the next task of the research, which is to develop a methods of decoding of factorial code with data recovery by permutation for real-time telecommunication systems.

The analysis of existing methods of construction of telecommunication systems with multiple access based on multiplexing of packets and time distribution is made. An analysis of these methods showed that most of them involve the availability of service data to implement a reservation/ release mechanism of time slots for data transmission, which reduces the overall bandwidth of the channel. Also, these methods do not implement a single procedure for providing integrated information security. The results of the analysis made it possible to define clearly the tasks of the research to develop a method of construction of multiple access systems based on inseparable factorial coding.

As part of the research a new methods of improving the reliability of data transmission systems with factorial coding with data recovery by permutation were developed, which allows to increase the capacity of inseparable factorial codes to detect errors by adding of additional bits signs to the permutation and combining of the noise immunity codes. The algorithm of factorial coding with additional bits has been developed. Implementation of the algorithm allows reducing the undetected error probability, for independent bit errors, compared to the factorial code with data recovery by permutation with addition by $2 - 4$ orders of magnitude at $M = 8$ depending on the bit error probability due to a small loss in code speed. Also, the algorithm of factorial cascade coding has been developed. Implementation of the algorithm allows to reduce the undetected error probability, for independent bit errors, compared to the factorial code with data recovery by permutation by order of

magnitude for $k = 15$ depending on the bit error probability with same code speed if the length of the data block is $k \geq 128$.

At first time a method of forming a code signal design for systems with factorial coding of data recovery permutation has been developed. It allowed to maximize speed of code for a given level of reliability data and the corresponding minimum distance between the nodes by constructing arrays of high density arrangement of signal vectors. A lattice is constructed, the nodes of which are permutations that allow to detect errors of multiplicity $t \leq 5$, and correct errors of multiplicity $t \leq 2$ with a maximum speed of code for $M = 5$, $k = 3$, $N_{sv} = 10$ and $M = 6$, $k = 5$, $N_{sv} = 60$.

In addition, the method of factorial coding with permutation data recovery has been further developed, which by data recovery with loss based on Hemming's metric or linear interpolation method allows protecting speech information from communication channel errors and unauthorized access in real-time telecommunications systems. The block diagram of the decoding device and the algorithm for error correction with losses in inseparable factorial codes in the Hamming metric are developed, which provides a decoding noise level that does not exceed the values of comfort noise level for bit error probability $p_0 \leq 10^{-2}$ in channels with independent bit errors. The developed scheme allows implementing a device for decoding factorial codes in the Hamming metric. The block diagram of the decoding device and the algorithm of error correction with losses in inseparable factorial codes by the method of linear interpolation are developed, which provides the level of decoding noise that does not exceed the values of comfort noise level in bit error probability $p_0 \leq 10^{-2}$ in channels with packets of errors. The developed scheme allows implementing a device for decoding factorial codes by linear interpolation.

In the final part of the dissertation research at first time a method of building multiple access systems based on inseparable factorial coding has been developed, which provides integrated protection of information from unauthorized access and as well as cyclic synchronization without the use of a separate synchronization

channel. The structural scheme and algorithm of operation of the multiplexer of communication channels on the basis of the factorial code with the given number of inversions which does not required the organization of the synchronizing channel and supports cyclic synchronization on a working signal are developed. The developed scheme allows implementing multiplexer of communication channels on hardware basis. The block diagram of telecommunications system with multiple access using multiplexer channels based on a factorial code with given number of inversions, the algorithm of dynamic resource allocation bandwidth based on the load generated by each of the users also have been developed The structural scheme allows to implement telecommunication in systems with multiple access based on factorial code on hardware basis.

Keywords: information security, confidentiality, integrity, reliability of data transmission, factorial coding, signal-code construction, multiple access systems.

Список основних публікацій здобувача

[1] О. О. Харін, «Оцінка властивостей каскадного коду, що поєднує факторіальний та рівноважний код», *Вісник Черкаського державного технологічного університету*, № 2, с. 86-90, 2017.

[2] О. О. Харін, «Порівняльна оцінка факторіальних кодів», *Вісник Черкаського державного технологічного університету. Серія: технічні науки*, № 4, с. 88-93, 2017.

[3] E. V. Faure, A. I. Shcherba and A. A. Kharin, «Factorial Code with a Given Number of Inversions», *Radio Electronics, Computer Science, Control*, vol. 2, p. 143-153, 2018.

[4] E. V. Faure, V. V. Shvydkyi, A. O. Lavdanskyi and O. O. Kharin, «Methods of factorial coding of speech signals», *Radio Electronics, Computer Science, Control*, vol. 4, p. 186-198, 2019.

[5] J. Al-Azzeh, B. Ayyoub, E. Faure, V. Shvydkyi, O. Kharin and A. Lavdanskyi, «Telecommunication Systems with Multiple Access Based on

Data Factorial Coding», *International Journal on Communications Antenna and Propagation (IRECAP)*, vol. 10, issue 2, p. 102-113, 2020.

[6] Е. В. Фауре та О. О. Харін, «Дослідження ймовірності виникнення помилки декодування під час використання факторіального коду з відновленням даних», в *Актуальні задачі та досягнення у галузі кібербезпеки: Тези доповідей Всеукраїнської науково-практичної конференції, Кропивницький, 23-25 листопада 2016 р.*, Кропивницький, 2016, с. 178-179.

[7] Е. В. Фауре та О. О. Харін, «Факторіальне кодування з відновленням даних і виправленням помилок», в *Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку: Тези доповідей Всеукраїнської науково-практичної Internet-конференції, Черкаси, 13-19 березня 2017 р.*, Черкаси: ЧДТУ, 2017, с. 74–76.

[8] О. О. Харін, «Підвищення достовірності передачі даних за допомогою факторіального кодування з виявленням транспозицій», в *Надзвичайні ситуації: безпека та захист: Тези доповідей VII Всеукраїнської науково-практичної конференції з міжнародною участю, Черкаси, 20-21 жовтня 2017 р.*, Черкаси: ЧІПБ ім. Героїв Чорнобиля НУЦЗ України, 2017, с. 162-163.

[9] А. А. Харин и А. И. Щерба, «Организация замкнутой группировки абонентов в открытой сети коллективного пользования», в *Інформаційні технології в освіті, науці і техніці (ІТОНТ-2018): Тези доповідей IV Міжнародної науково-практичної конференції, Черкаси, 17-18 травня 2018 р.*, Черкаси: ЧДТУ, 2018, с. 109-111.

[10] О. О. Харін, «Формування сигнально-кодової конструкції на основі теорії решіток», в *Наука України – погляд молодих вчених крізь призму сучасності: Тези доповідей II Всеукраїнської науково-практичної конференції з міжнародною участю, Черкаси, 26 вересня 2019 р.*, Черкаси: ЧДТУ, 2019, с.42-44.

[11] О. О. Харін, Е. В. Фауре та А. О. Лавданський, «Оцінка захищеності мовного сигналу в системах з факторіальним кодуванням», в *Інформаційні технології в освіті, науці і техніці (ІТОНТ-2020): Тези доповідей V Міжнародної науково-практичної конференції, Черкаси, 21-23 травня 2020 р.*, Черкаси: ЧДТУ, 2020, с. 85-87.

[12] Е. В. Фауре, О. О. Харін, В. В. Швидкий та А. О. Лавданський, «Ефективність виявлення помилок факторіальними кодами», в *Інформаційні технології в освіті, науці і техніці (ІТОНТ-2020): Тези доповідей V Міжнародної науково-практичної конференції, Черкаси, 21-23 травня 2020 р.*, Черкаси: ЧДТУ, 2020, с. 94-95.

[13] Е. В. Фауре, О. О. Харін, В. В. Швидкий та А. І. Щерба, «Спосіб факторіального кодування з виявленням і виправленням помилок», Україна. Пат. 121361, 11.12.2017.

[14] Е. В. Фауре та О. О. Харін, «Пристрій кодування та декодування факторіальних кодів з виявленням і виправленням помилок», Україна. Пат. 123640, 12.03.2018.

[15] О. О.Харін та А. І. Щерба, «Спосіб факторіального кодування в метриці Хеммінга», Україна. Пат. 130458, 10.12.2018.

[16] Е. В. Фауре, О. О. Харін, В. В. Швидкий та А. І. Щерба, «Спосіб факторіального кодування з відновленням даних», Україна. Пат. 117004, 12.06.2017.

[17] А. О. Лавданський, Е. В. Фауре, О. О. Харін та В. В. Швидкий, «Спосіб декодування факторіального коду з відновленням вибірок мовного сигналу реального часу в метриці Хеммінга», Україна. Пат. 137722, 11.11.2019.

[18] А. О. Лавданський, Е. В. Фауре, О. О. Харін та В. В. Швидкий, «Спосіб декодування факторіального коду з відновленням вибірок мовного сигналу реального часу методом лінійної інтерполяції», Україна. Пат. 139760, 27.01.2020.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

LTE – Long Term Evolution;

PRMA – Packet Reservation Multiple Access;

TDMA – Time-division multiple access;

ВЗЗ – вирішувальний зворотній зв’язок;

КФК – комбінований факторіальний код;

МПД – мультиплексор передавання даних;

ПФК – повний факторіальний код;

СКК – сигнально-кодова конструкція;

СМД – система множинного доступу;

СПД – система передавання даних;

ТС – телекомунікаційна система (телекомунікаційні системи);

ФК – факторіальний код (факторіальні коди);

ФКВД – факторіальний код з відновленням даних по перестановці;

ФКВДвп – факторіальний код з відновленням даних і виправленням помилок;

ФКВДд – факторіальний код з відновленням даних за перестановкою з доповненням;

ФКДКСн – нероздільний факторіальний код з декількома контрольними сумами;

ФКДКСр – роздільний факторіальний код з декількома контрольними сумами;

ФКЗЧІ – факторіальний код з відновленням даних за перестановкою з заданим числом інверсій;

ФКП – факторіальний код з проріджанням;

ФСЧ – факторіальна система числення;

ФКДБ – факторіальний код з додатковими перевірними бітами;

ФКК – факторіальний каскадний код.

ВСТУП

Актуальність теми дослідження. Інформаційно-комунікаційні технології та системи відіграють значну роль в житті сучасного суспільства, в якому відбувається активна комп'ютеризація всіх сфер його діяльності. Як наслідок – стрімке збільшення кількості інформації, яка отримується, обробляється, передається та зберігається в електронному вигляді, в тому числі інформації з обмеженим доступом. Це особливо проявляється в фінансовій та комерційній діяльності, сфері дистанційного керування автоматизованими виробничими комплексами і процесами, а також в сферах діяльності силових і охоронних відомств. Все це призводить до інтенсифікації досліджень і розробок нових методів обробки та захисту інформації в комп'ютерних системах і мережах.

У цій дисертаційній роботі загальновідомі теоретичні основи і методи комп'ютерної криптографії використовуються для створення нових методів і засобів інтегрованого захисту [1] інформації, які поєднують на одному теоретичному базисі, на одній апаратній платформі, в єдиній процедурі наступні функції:

- захисту інформації від помилок, що виникають у каналі зв'язку під час її передавання;
- захисту інформації від нав'язування хибних даних;
- захисту інформації від несанкціонованого читання.

Забезпечення інтегрованого захисту інформації також передбачає розв'язання супутніх задач, пов'язаних зі створенням нових методів кодування для телекомунікаційних систем реального часу, в тому числі для кодування мовленнєвих сигналів.

Питаннями розробки нових і вдосконалення існуючих методів і засобів інтегрованого захисту інформації в комп'ютерних системах, підвищення ефективності кодування інформації та пошуку ефективних комбінацій різних кодів займалися такі вчені, як R.J. McEliece [2], F.J. MacWilliams та N.J.A. Sloane [3], О.А. Борисенко [4] – [8], І.Д. Горбенко [9], [10], В.І. Грабчак [11] –

[13], О.О. Кузнецов [14] – [16], Е.Л. Онанченко [17], [18], С.О. Осмоловский [19] – [21], О.П. Стахов [22] – [26], В.Я. Чечельницький [27] – [30]. Питанням кодування мовленнєвих сигналів та побудови телекомунікаційних систем реального часу присвячені роботи D. W. Hagelbarger [31], [32], W. L. Kilmer [33], C. Berrou [34], [35], A. Stefanov та T. M. Duman [36], [37], L. Bahl [38], D. J. Goodman [39], [40], E.N. Gilbert [41], E.O. Elliott [42], Ф. М. Возенкрафта [43], Л. М. Фінка [44], [45] та ін.

Серед існуючих на сьогодні підходів до поєднання завадостійкого кодування та криптографічного захисту використання перестановок, як формату запису числа в факторіальній системі числення (ФСЧ), є відносно новим та найменш поширеним, але перспективним підходом [1], [46] – [59]. Цей підхід являє собою розумний компроміс між достовірністю передавання даних, з одного боку, та відносною швидкістю передавання й швидкістю коду, з іншого.

Разом з тим, недостатньо досліджено питання використання факторіальних кодів в спеціалізованих комп’ютерних і телекомунікаційних системах, в яких є необхідність забезпечення заданого рівня достовірності передавання даних в умовах погіршення якості каналу зв’язку.

Крім того, актуальною залишається задача забезпечення інтегрованого захисту інформації, в тому числі мовленнєвої, в телекомунікаційних системах реального часу з урахуванням їх ресурсних обмежень.

У дисертаційній роботі вирішується важлива науково-технічна задача, що полягає в підвищенні достовірності передавання інформації, забезпеченні її конфіденційності та цілісності на основі використання факторіального кодування даних. Ця задача передбачає необхідність удосконалення існуючих методів факторіального кодування, створення методу побудови телекомунікаційних систем на основі нероздільного факторіального кодування з динамічним розподілом ресурсів між користувачами системи в режимі реального часу.

Виходячи з цього, тема дисертаційної роботи «Методи та засоби інтегрованого захисту інформації в телекомунікаційних системах множинного доступу на основі факторіального кодування даних» є актуальною.

Зв'язок роботи з науковими програмами, планами, темами. Дослідження, результати яких представлено в дисертаційній роботі, відповідають пріоритетному напряму розвитку науки і техніки України «Інформаційні та комунікаційні технології» та його тематичному напряму «Технології та засоби захисту інформації» і виконувалися відповідно до програм і планів науково-дослідних робіт Черкаського державного технологічного університету, у тому числі в рамках держбюджетної науково-дослідної теми «Розробка мобільної системи захищеного інформаційного обміну для військових і цивільних підрозділів державних структур» (номер державної реєстрації 0120U102607), в яких автор брав участь як виконавець.

Мета роботи полягає в підвищенні ефективності передавання даних у телекомунікаційних системах множинного доступу шляхом розробки методів і засобів інтегрованого захисту інформації на основі факторіального кодування даних.

Для досягнення поставленої мети вирішуються наступні задачі:

- розробка методів підвищення достовірності передавання даних у системах з нероздільним факторіальним кодуванням;
- розробка методу формування сигнально-кодової конструкції для систем з факторіальним кодуванням з відновленням даних за перестановкою (ФКВД);
- розробка методів декодування факторіальних кодів з відновленням даних за перестановкою для систем передавання мовленнєвої інформації реального часу;
- розробка методу побудови телекомунікаційних систем множинного доступу з використанням нероздільного факторіального кодування;

Об'єктом дослідження є процеси інтегрованого захисту інформації у комп'ютерних і телекомунікаційних системах і мережах.

Предметом дослідження є методи та засоби інтегрованого захисту інформації на основі факторіального кодування даних.

Методи досліджень. Для вирішення задачі розробки методів підвищення достовірності передавання даних у системах з нероздільним факторіальним кодуванням використовувалися методи: теорії ймовірності і математичної статистики, статистичного аналізу, комбінаторики.

Для вирішення задачі розробки методу формування сигнально-кодової конструкції для систем з факторіальним кодуванням з відновленням даних за перестановкою використовувалися методи: теорії решіток, теорії ймовірності і математичної статистики, статистичного аналізу, об’єктно-орієнтованого програмування.

Для вирішення задачі розробки методів декодування факторіальних кодів з відновленням даних за перестановкою для систем передавання мовленнєвої інформації реального часу використовувалися методи: теорії ймовірності і математичної статистики, теорії інформації, цифрової обробки сигналів.

Для вирішення задачі розробки методу побудови телекомунікаційних систем множинного доступу з використанням нероздільного факторіального кодування використовувалися методи: кодування інформації, цифрової обробки сигналів, комп’ютерних мереж, теорії ймовірності і математичної статистики.

Наукова новизна отриманих результатів:

- ***вперше розроблено*** методи підвищення достовірності передавання даних у системах з факторіальним кодування з відновленням даних за перестановкою, які за рахунок доповнення перестановки бітами ознак та комбінування завадостійких кодів дозволяють підвищити здатність нероздільних факторіальних кодів до виявлення помилок;
- ***вперше розроблено*** метод формування сигнально-кодової конструкції для систем з факторіальним кодування з відновленням даних за перестановкою, який за рахунок побудови решіток з високою щільністю розташування сигнальних векторів дозволяє максимізувати

швидкість коду для заданого рівня достовірності передавання даних та відповідної мінімальної відстані між вузлами решітки;

- ***отримав подальший розвиток*** метод факторіального кодування з відновленням даних за перестановкою, який за рахунок відновлення даних з втратами на основі метрики Хеммінга або методу лінійної інтерполяції дозволяє забезпечити захист мовленнєвої інформації від помилок каналу зв'язку та несанкціонованого доступу в телекомунікаційних системах реального часу;
- ***вперше розроблено*** метод побудови систем множинного доступу на основі нероздільного факторіального кодування, який за рахунок використання факторіального коду з заданим числом інверсій та виділення кожному з користувачів підмножини перестановок з певним набором властивостей забезпечує інтегрований захист інформації від несанкціонованого доступу та помилок каналу зв'язку, а також циклову синхронізацію без застосування окремого каналу синхронізації.

Практичне значення отриманих результатів

- розроблено алгоритм факторіального кодування з додатковими перевірними бітами. Реалізація алгоритму дає змогу зменшити ймовірність невиявленої кодом помилки, за незалежних бітових помилок, у порівнянні з факторіальним кодом з відновленням даних за перестановкою з доповненням (ФКВДд) на 2-4 порядки для $M = 8$ у залежності від ймовірності бітової помилки за рахунок незначної втрати в швидкості коду;
- розроблено алгоритм факторіального каскадного кодування. Реалізація алгоритму дає змогу на порядок зменшити ймовірність невиявленої кодом помилки, за незалежних бітових помилок, у порівнянні з ФКВД для $k = 15$ у залежності від ймовірності бітової помилки, при цьому за довжини блоку даних $k \geq 128$ швидкість коду не змінюється;
- побудовано решітки, вузлами якої є перестановки, що дозволяють виявляти помилки кратності $t \leq 5$ і виправляти помилки кратності

$t \leq 2$ з максимальною швидкістю коду для $M = 5,\ k = 3,\ N_{sv} = 10$ та $M = 6, k = 5, N_{sv} = 60$;

- розроблено структурну схему пристрою декодування та алгоритм виправлення помилок з втратами у нероздільних факторіальних кодах в метриці Хеммінга, що забезпечує рівень шуму декодування, який не перевищує значень рівня комфортного шуму за ймовірності бітової помилки $p_0 \leq 10^{-2}$ у каналах з незалежними бітовими помилками. Розроблена схема дозволяє апаратно реалізувати пристрій декодування факторіальних кодів в метриці Хеммінга;
- розроблено структурну схему пристрою декодування та алгоритм виправлення помилок з втратами у нероздільних факторіальних кодах методом лінійної інтерполяції, що забезпечує рівень шуму декодування, який не перевищує значень рівня комфортного шуму за ймовірності бітової помилки $p_0 \leq 10^{-2}$ у каналах з пакетуванням бітових помилок. Розроблена схема дозволяє апаратно реалізувати пристрій декодування факторіальних кодів методом лінійної інтерполяції;
- розроблено структурну схему та алгоритм роботи мультиплексора каналів зв’язку на основі факторіального коду з заданим числом інверсій (ФКЗЧІ), що не потребує організації синхронізуючого каналу та підтримує циклову синхронізацію за робочим сигналом. Розроблена схема дозволяє апаратно реалізувати мультиплексор каналів зв’язку;
- розроблено структурну схему телекомунікаційної системи з множинним доступом з використанням мультиплексора каналів зв'язку на основі факторіального коду з заданим числом інверсій, розроблено алгоритм динамічного розподілу ресурсу пропускної здатності каналу залежно від значення навантаження, що створюється кожним з користувачів системи. Розроблена схема дозволяє апаратно реалізувати телекомунікаційну систему з множинним доступом на основі факторіального кодування.

Результати дисертаційної роботи впроваджені в ТОВ «АНТЕКС УКРАЇНА» (система передавання даних обладнання для паралельного водіння та точного землеробства, акт від 04.05.2020).

Особистий внесок здобувача. Дисертація є самостійно виконаною завершеною роботою здобувача. Наукові результати і практичні розробки, що містяться в дисертаційній роботі, отримані автором самостійно.

У роботах, опублікованих у співавторстві, автором: [3] – виконано аналіз ймовірності невиявленої помилки та швидкості коду для факторіального коду з заданим числом інверсій; [4], [17], [18] – розроблено метод факторіального кодування мовленнєвих сигналів з відновленням даних з втратами в метриці Хеммінга та методом лінійної інтерполяції; [5], [11] – розроблено та описано алгоритм роботи мультиплексора каналів на основі факторіальних кодів; [5], [16] – виконано аналіз причин виникнення невиявлених помилок ФКВД, уточнено значення питомої помилки ФКВД; [9] – запропоновано використовувати класи лишків ФКЗЧІ для формування замкнених груп користувачів; [7], [13], [14] – виконано дослідження та порівняльний аналіз ймовірностей невиявлених помилок факторіальним кодом з виявленням та виправленням помилок (ФКВДвп) для різних сигнально-кодових конструкцій; [12] – виконано експериментальну оцінку ймовірності невиявленої помилки для нероздільних факторіальних кодів; [15] – розроблено та описано структурні схеми пристроїв кодування та декодування, запропоновано принципи формування сеансового ключа для забезпечення захисту від несанкціонованого доступу до інформації. З робіт, опублікованих у співавторстві, для вирішення задач, поставлених у дисертаційному дослідженні, використано результати, отримані здобувачем особисто.

Апробація результатів дисертації. Основні результати дисертаційної роботи доповідалися та обговорювалися на:

- Всеукраїнській науково-практичній конференції «Актуальні задачі та досягнення у галузі кібербезпеки» (м. Кропивницький, 23–25 листопада 2016 року);

- Всеукраїнській науково-практичній Internet-конференції «Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку» (Черкаси, 13-19 березня 2017 р);
- VII Всеукраїнській науково-практичній конференції з міжнародною участю «Надзвичайні ситуації: безпека та захист» (Черкаси, 20-21 жовтня 2017 року);
- IV Міжнародній науково-практичній конференції «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2018) (Черкаси, 17-18 травня 2018 року);
- II Всеукраїнській науково-практичній конференції з міжнародною участю «Наука України – погляд молодих вчених крізь призму сучасності» (Черкаси, 26 вересня 2019 року);
- V Міжнародній науково-практичній конференції «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2020) (Черкаси, 21-23 травня 2020 року).

Публікації. Результати дослідження опубліковані у 18 наукових роботах, в тому числі три наукові статті, що входять до наукометричних баз даних Scopus та / або Web of Science [3] – [5], дві статті у наукових виданнях, що входять до переліку МОН України та інших наукометричних баз даних [1], [2], семи доповідях на науково-практичних конференціях [5] – [12] і шести патентах України на корисні моделі [13] – [18].

Структура та обсяг дисертаційної роботи. Дисертаційна робота складається з вступу, п'яти розділів, висновків, списку використаних джерел і додатків. Повний об'єм дисертації складає 179 сторінок. Робота містить 7 таблиць та 31 рисунок.

РОЗДІЛ 1. АНАЛІЗ СУЧАСНОГО СТАНУ ПРЕДМЕТНОЇ ОБЛАСТІ. ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1. Вступ

У сучасному суспільстві розповсюдження інформації відбувається настільки швидко, що важко уявити його існування без використання телекомунікаційних систем і мереж, що реалізують зберігання, обробку та передавання інформації. Відповідно, однією з головних функцій таких систем є забезпечення цілісності інформації, що знаходиться в системі. У той же час, не менш важливою функцією є захист інформації від впливу завад, що виникають у каналі зв’язку.

Під час вирішення проблеми забезпечення безпеки інформації слід враховувати наступні особливості:

- великі масиви інформації, що зберігаються в базах даних або передаються в мережі;
- можливість доступу сторонніх осіб до масивів даних, що знаходяться в мережі;
- широке використання бездротових (супутникових, радіо-) каналів зв’язку, в яких присутні завади та взаємний вплив абонентів один на одного;
- робота на граничній дальності зв’язку, несанкціоноване прослуховування переданих даних і навмисний вплив на канал зв’язку й інформацію, що транспортується, з метою придушення обміну або нав’язування хибної інформації. Зауважимо, що такі недоліки в значній мірі притаманні каналам, організованим кабельними каналами зв’язку.

Зважаючи на це, основною вимогою, що пред’являються до телекомунікаційних систем, є забезпечення цілісності інформації в умовах сучасного інформаційного середовища.

Перераховані вище особливості призводять до необхідності застосування комплексу різних видів захисту інформації, який повинен забезпечити:

- захист від помилок у каналах зв’язку;

− захист від нав’язування хибної інформації;

− захист від несанкціонованого ознайомлення з інформацією.

Як показано в [1], застосування декількох зазначених видів захисту інформації призводить до підвищення вимог до швидкодії компонентів, що реалізують ці процедури, а також до зростання введеної надлишковості і, як наслідок, до зменшення пропускної здатності каналу зв’язку. Поєднання в єдину процедуру функцій криптографії та завадостійкого кодування є оптимальним вирішенням проблеми підвищення ефективності телекомунікаційних систем і мереж. Такий вид захисту, згідно [1], називається інтегрованим захистом.

Одним з рішень, що забезпечує досягнення поставленої задачі – створення засобів інтегрованого захисту інформації, є факторіальне кодування, яке передбачає використання перестановок в якості носія інформації.

Визначення 1.1. Перестановкою символів скінченної множини A називається бієктивна функція $\pi : A \to A$. Всього існує $M!$ різних перестановок, де M - потужність множини A.

Використання факторіальних кодів (ФК) для вирішення задач завадостійкого кодування, захисту від несанкціонованого доступу та забезпечення контролю цілісності інформації обумовлено їх наступними властивостями:

- кожен символ множини $\{0,1,2,...,M-1\}$ зустрічається в перестановці рівно один раз;
- розташування символів в перестановці дозволяє однозначно встановити інформаційну послідовність, що породила цю перестановку, за умови, що $k \le [\log_2 M!]$, де k – кількість біт у інформаційній послідовності;
- сума символів у перестановці завжди постійна.

Згідно [46], основними властивостями факторіального кодування є:

1. виявлення всіх помилок, що трансформують перестановку в неперестановку, тобто послідовність, у якій деякі символи з множини $\{0,1,2,...,M-1\}$ пропущені або повторюються;
2. приховування закону відображення інформаційного вектора $A(x)$ в перестановку $\pi(x)$: $A(x) \Rightarrow \pi(x)$ виключає можливість несанкціонованого читання інформації, а додавання перестановки до відкритого повідомлення забезпечує імітозахист;
3. сталість суми символів перестановки забезпечує можливість самосинхронізації факторіальних кодів, що виключає необхідність введення в структуру кадру синхропослідовності.

Зауважимо, що якщо правила відображення $A(x) \Rightarrow \pi(x)$ і $\pi(x) \Rightarrow A(x)$ на передавальній і приймальній станціях реалізуються у вигляді таблиць замін, то застосування методів факторіального кодування дозволяє реалізувати єдину процедуру кодування/декодування, а також забезпечує криптографічний захист і спрощує апаратну платформу.

Однак, слід зазначити наступні особливості реалізації методів факторіального кодування:

1. Оскільки потужність множини інформаційних векторів $A(x)$ завжди менше потужності множини перестановок $\pi(x)$, вибір оптимальних значень k та M суттєво впливає всі якісні показники коду;
2. Вплив потоку помилок в каналі зв'язку призводить до деформації перестановок, частина з яких легко виявляється під час перевірки на коректність перестановки. Виправлення помилок відбувається за рахунок повторного запиту прийнятої з помилкою перестановки;
3. Використання таблиць замін для реалізації процесу кодування та декодування вимагає значних апаратних ресурсів при збільшенні потужності множини символів перестановки M.

Враховуючи ці обставини, актуальними напрямками досліджень, що проводяться в цій роботі, є підвищення ефективності існуючих факторіальних кодів [1], [46] – [59], що полягає в наступному:

1. розробка нових методів підвищення достовірності передавання даних у системах з нероздільним факторіальним кодуванням, зокрема ФКВД;
2. розробка нових методів декодування ФКВД для систем передавання мовленнєвої інформації реального часу, що забезпечують інтегрований захист інформації та виправленням помилок з втратами;
3. розробка методу побудови телекомунікаційних систем множинного доступу з використанням нероздільного факторіального кодування.

У першому розділі для постановки задач дисертаційного дослідження необхідно:

- визначити основні переваги факторіального кодування в порівнянні з іншими існуючими методами поєднання функцій завадостійкого кодування та криптографічного захисту інформації;
- виконати аналіз існуючих методів кодування мовленнєвих сигналів в системах загального призначення;
- виконати аналіз методів організації множинного доступу до каналу колективного користування.

1.2. Методи поєднання функцій завадостійкого кодування та криптографічного захисту інформації

Під час передавання даних каналами зв’язку колективного користування необхідно вирішувати ряд задач із забезпечення захисту інформації, включаючи забезпечення конфіденційності, достовірності та цілісності даних, а також виявлення та виправлення помилок каналу зв’язку. Вирішення кожної з цих задач накладає додаткові вимоги до апаратної частини, а також може вимагати введення в блок даних додаткової надлишковості.

Методи поєднання функцій завадостійкого кодування та криптографічного захисту інформації почали розвиватися відносно нещодавно. Зокрема, як показано в роботі [1], до них належать метод криптографії з відкритим ключем на основі кодів, що виправляють помилки, запропонований McEliece R.J. у статті [2].

У роботах Осмоловского С.О. [19] – [21] запропоновано стохастичний метод інтегрального захисту інформації, що включає каскадне виконання операцій завадостійкого кодування та шифрувального стохастичного перетворення.

У роботах Стахова О.П. [22] – [26] для забезпечення інтегрованого захисту запропоновано використання методу “золотої” криптографії.

У роботах Чечельницького В.Я. [27] – [30] запропоновано використовувати параметричну прихованість системи зв'язку з шумоподібними сигналами (ШПС) у поєднанні з завадостійким кодування.

Також, в роботі [1] показано, що цим методам властиві наступні недоліки:

1. для криптосистеми McEliece R.J. властива низька швидкість коду, потреба в дуже великій довжині ключа, а також слабка стійкість до зламу за повторного використання ключа;
2. стохастичні методи захисту Осмоловського С.О. не вирішують проблему поєднання функцій завадостійкого кодування та криптографічного захисту інформації в єдину процедуру, а також передбачають використання гамування, що не завжди є доцільним в реальних системах передавання даних;
3. «золота» криптографія Стахова О.П. вимагає подальших досліджень для визначення параметрів та характеристику коду;
4. методи захисту інформації, запропоновані Чечельницьким В.Я., призначені для систем зв’язку з шумоподібними сигналами і не пристосовані для використання в телекомунікаційних системах загального призначення.

1.3. Захист інформації на основі факторіального кодування

Розглянуті в цій роботі факторіальні коди базуються на роботах [4] – [8], виконаних під керівництвом Борисенка О.А. У цих роботах вперше викладно принципи передавання даних на базі перестановок. У тому числі показано, що побудований на перестановках код є рівноважним зі всіма його властивостями. Використання факторіальних кодів для інтегрованого захисту інформації в системах передавання даних є актуальним напрямком досліджень, результати яких представлені в роботах [1], [46] – [59]. Як уже зазначалося у вступі цього розділу, в основі принципів факторіального кодування лежить побудова перестановок на основі факторіальної системи числення (ФСЧ). Відповідно до [46], перетворення слова джерела в перестановку здійснюється за рахунок відображення точок на числовій вісі, що відповідають інформаційному вектору $A(x)$, який є двійковим представленням слова джерела довжиною *k* біт, у точки числової вісі, що відповідають номерам перестановок $\pi(x)$. Перестановка представляється послідовністю всіх символів $\{0;1;...;M-1\}$.

Відповідно до [1], у залежності від наявності в блоці даних явно вираженої інформаційної частини, факторіальні коди поділяються на дві великі групи – роздільні та нероздільні факторіальні коди.

1.3.1. Роздільні факторіальні коди

Загальною рисою всіх роздільних ФК є використання перестановок символів з множини $\{0;1;...;M-1\}$ в якості перевірної частини, яка формується на основі інформаційного вектору. Таким чином, сформоване кодове слово складається з двох частин: інформаційної та перевірної. Потужність множини M обирається в залежності від необхідного ступеня підвищення достовірності передавання даних і методу кодування.

До роздільних ФК належать:

- повний факторіальний код (ПФК) [46], [47];
- комбінований факторіальний код (КФК) [48];
- факторіальний код з проріджением (ФКП) [49], [50];

- роздільний факторіальний код з декількома контрольними сумами (ФКДКСр) [49], [50];

ПФК передбачає формування перевірної частини кодового слова у вигляді перестановки порядку M, що визначається інформаційною частиною та алгоритмом кодування. Інформаційна частина, в свою чергу, передається у відкритому вигляді і ніяк не захищена від несанкціонованого читання.

Цей метод кодування вразливий до наступних помилок:

1. помилки, що призводять до зміни біт інформаційної частини за незмінної контрольної суми, не будуть виявлятися. Оскільки ПФК не висуває жорстких вимог до порядку перестановки M, то за умови якщо $M! < 2^k$, де k – кількість біт в інформаційній частині, виникне ситуація, що декільком значенням інформаційної частини буде відповідати одна перестановка;
2. помилки, що вражають і інформаційну, і перевірну частину таким чином, що зберігається відповідність інформаційної частини контрольній сумі.

Метод ПФК, відповідно до [46], полягає в наступному:

1) контрольна сума представляється у вигляді однієї перестановки порядку M;
2) формування контрольної суми є ітераційним процесом над синдромами перестановок, представленими в ФСЧ:
   a) інформаційна частина розбивається на k_{symb} блоків (укрупнених символів);
   b) значення поточного укрупненого символу підсумовується з модифікованим синдромом попередньої перестановки, а результат цього перетворення визначає синдром наступної перестановки;
   c) початковий синдром тримається в таємниці та є елементом загального ключа перетворення;

d) процедура і ключ модифікації синдрому тримаються в таємниці та є елементами загального ключа перетворення;

e) перетворення останнього синдрому в перестановку після перебору всіх укрупнених символів інформаційного повідомлення виконується відповідно до ключа, який також є елементом ключа перетворення;

3) для підвищення стійкості до зламу блок даних, що містить інформаційну та перевірну частини, може піддаватися перестановці біт з метою зміни порядку їх слідування в процесі передавання каналом зв'язку приймачу, правило перестановки тримається в таємниці і є частиною ключа перетворення.

Перевагами цього методу є простота реалізації і можливість налаштовувати довжину контрольної суми, виходячи з якості каналу зв’язку та вимог з достовірності передавання даних, що дає змогу ефективно балансувати між достовірністю передавання та швидкістю коду.

КФК передбачає формування контрольної суми CRC-коду, обчисленої за перестановкою порядку *M*, що визначається інформаційною частиною. Іншими словами, КФК є модифікацією ПФК, де замість самої перестановки передається відповідна їй контрольна сума. КФК вимагає більших апаратних затрат на його реалізацію, але забезпечує більшу достовірність передавання даних та швидкість коду.

Запропонований у [48] метод передбачає наступне:

1) образ інформаційної частини блоку даних формується у вигляді однієї перестановки порядку M (або відповідного їй синдрому) на основі прихованої залежності від кожного символу інформаційної частини згідно з механізмами ПФК. Отримана перестановка (синдром) не підлягає передаванню приймачу;
2) представлена в двійковому вигляді перестановка (або її синдром) кодується завадостійким кодом (наприклад, CRC-кодом). Отримана

перевірна частина є частиною кодового слова КФК і вводиться в блок даних;

3) для підвищення стійкості до зламу блок даних може піддаватися перестановці біт з метою зміни порядку їх слідування в процесі передавання каналом зв`язку приймачу, при цьому правило перестановки тримається в таємниці.

ФКП, на відміну від попередніх методів роздільного факторіального кодування, при обчисленні перевірної частини кодового слова використовує лише частину інформаційних символів вектора $A(x)$. Це дозволяє скоротити час формування кодового слова і обсяг необхідних апаратних ресурсів, але робить ФКП менш стійким до впливу завад.

Метод ФКП, відповідно до [49], полягає в наступному:

1) контрольна сума ФКП представлена перестановкою розмірністю M;
2) з інформаційної частини кодового слова відповідно до секретного ключа, який є елементом ключа перетворення, обираються k_{FCD} біт;
3) відповідно до принципів ПФК за обраними k_{FCD} бітами формується контрольна сума – перестановка;
4) для підвищення стійкості до зламу блок даних, що містить інформаційну та перевірну частини, може піддаватися перестановці біт з метою зміни порядку їх слідування в процесі передавання каналом, при цьому правило перестановки тримається в таємниці і є частиною ключа перетворення.

ФКДКСр є вдосконаленням ФКП і передбачає комбінування декількох перевірних частин ФКП. Метод ФКДКСр описано в роботі [49] і полягає в наступному:

1) контрольна сума ФКДКСр представляється у вигляді конкатенації N перестановок потужністю $M(i)$ символів $(1 \le i \le N)$;

2) кожна перестановка формується відповідно до принципів ФКП, причому кожен інформаційний біт бере участь у формуванні однієї перестановки;
3) для підвищення стійкості до зламу блок даних, що містить інформаційну та перевірну частини, може піддаватися перестановці біт з метою зміни порядку їх слідування в процесі передавання каналом зв`язку, при цьому правило перестановки тримається в таємниці і є частиною ключа.

Однак роздільні факторіальні коди мають один суттєвий недолік, який полягає в тому, що в кодовому слові роздільних ФК міститься інформаційна частина, яка не захищена від несанкційованого читання, мають низьку пропускну здатність через виправлення помилок лише за рахунок повторного запиту блоку даних. Крім того, таким кодам властиве явище помилкового перезапиту даних, що обумовлено помилками, що вражають лише перевірну частину кодової послідовності. Нероздільні факторіальні коди позбавлені цих недоліків.

1.3.2. Нероздільні факторіальні коди

Загальною рисою всіх нероздільних ФК є повна заміна інформаційного вектора $A(x)$ на перестановку символів $\pi_n(x)$ з множини $\{0;1;...;M-1\}$, що обирається за умови $M!\geq 2^k$, $k=[\log_2 M!]$ для забезпечення відповідності кожному слову джерела лише одної перестановки.

До нероздільних ФК належать:

- факторіальний код з відновленням даних за перестановкою (ФКВД) [51] – [53];
- факторіальний код з відновленням даних за перестановкою з доповненням (ФКВДд) [52], [54];
- нероздільний факторіальний код з декількома контрольними сумами (ФКДКСн) [1], [49], [50];

- факторіальний код з відновленням даних за перестановкою з заданим числом інверсій (ФКЗЧІ) [55];
- факторіальний код з відновленням даних і виправленням помилок (ФКВДвп) [56] – [59].

Відповідно до [51], факторіальним кодом з відновленням даних за перестановкою називається нероздільний код, який передбачає заміну інформаційної послідовності з k біт на перестановку чисел порядку M, $\left(M! \geq 2^k\right)$ обчислену за всіма k інформаційними бітами.

Описаний у [51] метод передбачає наступне:

1) інформаційна послідовність $A(x)$ з k біт перетворюється в перестановку π з потужністю множини символів $M : M! \geq 2^k$, а множина з $M! - 2^k$ перестановок є забороненою;
2) перетворення інформаційного повідомлення в перестановку є бієктивним відображенням $A(x) \leftrightarrow \pi$ рівнопотужних множин інформаційних векторів $A(x)$ і дозволених перестановок π;
3) правило $A(x) \leftrightarrow \pi$ може триматися в таємниці і складати ключ перетворення;
4) символи перестановки π кодуються двійковим кодом, після чого вона передається каналом зв'язку одержувачу.

Для руйнування статистичного зв'язку між інформаційною послідовністю $A(x)$ і відповідною перестановкою $\pi(x)$ послідовність $A(x)$ може піддатися скремблюванню або гамуванню.

Метод ФКВДд [52], [54] являється модифікацією ФКВД і передбачає використання природної надлишковості факторіальних кодів для введення в інформаційну частину додаткових перевірних біт, що дозволяють підвищити достовірність передавання даних. Згідно [54], ФКВДд полягає у наступному:

1) визначаються значення k і M. Наприклад, для заданого k значення M може вибиратися згідно з умовою:

a) якщо $\alpha = M!/2^k > 2$, перед формуванням перестановки в інформаційну частину вводяться $r_{add} \le \lfloor \log_2 \alpha \rfloor$ додаткових перевірних біт;

b) якщо $\alpha = M!/2^k < 2$, підвищення достовірності передавання може бути досягнуто шляхом зменшення довжини інформаційного вектора на r_{add} біт і введення замість них додаткових перевірних біт;

2) доповнена перевірними бітами інформаційна послідовність перетворюється в перестановку відповідно до принципів ФКВД.

Нероздільним факторіальним кодом з декількома контрольними сумами, згідно [1], називають код, який передбачає заміну інформаційної послідовності на конкатенацію $N \ge 2$ кодових слів ФКВД, обчислених за N різними підблоками, на які розбивається інформаційна послідовність символів.

Описаний в [1] метод ФКДКСн полягає у наступному:

1) кодове слово ФКДКСн представляється у вигляді конкатенації N перестановок потужністю $M(i)$ символів $(1 \le i \le N)$;

2) кожна перестановка формується відповідно до принципів ФКВД за окремими блоками, на які розбивається інформаційна послідовність символів. Кожен інформаційний біт входить тільки в один блок і бере участь у формуванні тільки однієї перестановки.

Відповідно до [55], метод ФКЗЧІ являє собою ФКВД, множина дозволених кодових слів якого складається з перестановок із заданим числом інверсій і передбачає наступну послідовність дій:

1) інформаційна послідовність $A(x)$ з k біт перетворюється в перестановку π порядку M;

2) перетворення інформаційного повідомлення в перестановку є бієктивним відображенням $A(x) \leftrightarrow \pi$ рівнопотужних множин інформаційних векторів $A(x)$ і дозволених перестановок π. Правило відображення $A(x) \leftrightarrow \pi$ може триматися в таємниці і складати ключ перетворення;

3) множина дозволених перестановок π порядку M належить класу $B_M(q,R)=\left\{\pi\left|inv(\pi)\right|_q=R\right\}$ перестановок, число інверсій у яких належить заданому класу лишків $\overline{R_q}$;
4) модуль q класу лишків визначається виходячи з необхідного ступеня підвищення достовірності та допустимої втрати швидкості коду;
5) символи перестановки π кодуються двійковим кодом, після чого вона передається каналом зв'язку одержувачу.

Метод ФКВДвп становить значний інтерес для подальших досліджень, оскільки вперше висвітлює можливість виправлення помилок за допомогою факторіальних кодів, що висвітлено в роботах [56] – [59]. Згідно [58], ФКВДвп називається нероздільний код, який передбачає заміну інформаційної послідовності з k біт на перестановку чисел порядку M $\left(M!\geq 2^k\right)$, обчислену за всіма інформаційними бітами, таким чином, що відстань між кодовими словами є достатньою для виправлення виникаючих у каналі зв'язку помилок. Описаний у [58] метод передбачає наступну послідовність дій:

1) інформаційна послідовність $A(x)$ з k біт перетворюється в перестановку π порядку M:
   a) з усієї множини перестановок потужності $M!$ дозволеною є лише підмножина з 2^k перестановок;
   b) показник α визначає відстань між перестановками і залежить від вимог до достовірності передавання і принципів формування СКК;
2) перетворення $A(x)\leftrightarrow\pi$ є бієктивним відображенням рівнопотужних множин інформаційних векторів $A(x)$ і дозволених перестановок π;
3) правило відображення $A(x)\leftrightarrow\pi$ може триматися в таємниці і складати ключ перетворення;
4) символи перестановки π кодуються двійковим кодом, після чого вона передається каналом зв'язку одержувачу.

Таким чином, розглянуті методи нероздільного факторіального кодування вирішують проблеми криптографічного захисту інформації та завадостійкого кодування.

Однак, актуальною задачею залишається підвищення достовірності передавання даних для існуючих методів нероздільного факторіального кодування, а також розробка нових, що дозволить розширити область використання ФК. Розробка таких методів та оцінка їх властивостей виконана в розділі 2.

Також, ще однією актуальною задачею залишається процес формування СКК-2 для ФКВДвп, вирішення якої безпосередньо впливає на всі параметри коду. Одним з варіантів вирішення цієї задачі є метод формування СКК на основі теорії решіток, розробка та дослідження якого виконано в розділі 3.

1.4. Методи канального кодування мовленнєвих сигналів у системах загального призначення

За умов сучасних темпів розвитку комп'ютерних технологій, актуальною задачею залишається підвищення якості та надійності роботи телекомунікаційних систем, які призначені для передавання мовленнєвих сигналів у режимі реального часу. Такі системі зазвичай використовують відкриті для загального доступу середовища передавання даних, наприклад, лінії повітряного та кабельного зв'язку, радіолінії, лінії оптоволокневого та супутникового зв’язку. Бездротові середовища передавання даних досить чутливі до впливу шумів природного та техногенного походження, тому рівень шуму в них суттєво вищий, ніж у дротових лініях зв’язку, а також слабо захищені від перехоплення даних третіми особами.

Також слід зауважити, що оскільки передача мовленнєвих сигналів відбувається в режимі реального часу, методи кодування даних, які використовуються, повинні передбачати виправлення помилок через вимоги до швидкості передавання даних, що ставляться до систем реального часу.

1.4.1. Метод кодування мовленнєвих сигналів на основі згорткових кодів

Згорткові коди, які вперше запропоновані, незалежно один від одного, Фінком Л.М. [44] у 1955 році, а також Hagelbarger D. W. [31], [32] у 1959 році, отримали широке використання системах супутникового та мобільного зв'язку. Згорткові коди отримали подальший розвиток у роботах [33], [43], [60] і ін.

Характерною рисою таких кодів є те, що операції кодування та декодування відбуваються над неперервним потоком даних, а не над блоками. Згорткові коди описуються трьома цілими числами – (k, n, s), де k – кількість інформаційних символів, що обробляються в кожен момент часу (визначає кількість регістрів зсуву), n – довжина кодової послідовності, s – довільне ціле додатне число, яке називається кроком згорткового коду (залежить від розрядності регістрів зсуву).

Так, наприклад, описаний у роботі [44] метод згорткового кодування передбачає:

1) на початку кожного часового інтервалу k інформаційних символів поступають на входи регістрів зсуву з розрядністю $m_1, m_2,...,m_k$.

2) кодова послідовність утворюється комбінацією інформаційних символів на виході регістрів зсуву та перевірних символів, що визначаються за формулою:

$$b_i = a_{i-s} \oplus a_{i+s+1}. \qquad (1.1)$$

Перевірні символи розташовуються в кодовій послідовності наступним чином:

$$a_1 b_1, a_2 b_2,..., a_k b_k, a_{k+1} b_{k+1},...,$$

де a_i – інформаційний символ, b_i – перевірний символ;

3) після формування перевірних символів у всіх регістрах зсуву виконується зсув вправо на один розряд, після чого послідовність дій повторюється;

4) процес декодування полягає в перевірці кожного з b_i символів на відповідність відношенню (1.1). У випадку невиконання цієї умови для двох значень (b_{k_1}, b_{k_2}), де $k_1 = i - s - 1$, $k_2 = i + s$ і якщо $k_2 - k_1 = 2s + 1$, це вказує на помилку в a_i інформаційному символі і він підлягає виправленню.

Відмітимо відносно малу швидкість коду, яка становить $v = k / n = k / 2k = 0.5$. Оскільки цей метод відноситься до чистого завадостійкого (канального) кодування та передбачає наявність у кодовій послідовності інформаційних символів у відкритому вигляді, він не забезпечує криптозахист даних.

1.4.2. Метод кодування мовленнєвих сигналів на основі каскадних кодів

Для кодування мовленнєвих сигналів часто використовуються каскадні коди або турбо-коди (tubo-codes) [34], [35], які отримали подальший розвиток в роботах [36], [37], [61] – [65] та ін. Каскадні коди використовуються під час передавання мовленнєвих сигналів у мобільному зв'язку, їх використання передбачено стандартом GSM. Каскадне кодування передбачає послідовне використання декількох кодів, у тому числі згорткових.

Так, наприклад, у роботах [34], [35] розглядається наступний алгоритм дій:

1) на початку кожного моменту часу k на вхід кодера потрапляє інформаційний біт. На основі поточного та попередніх значень інформаційного біту d_k формується кодова послідовність з двох бітів (X_k, Y_k), де:

$$X_k = d_k,\ Y_k = \left| \sum_{i=0}^{K-1} g_i a_{k-i} \right|_2, \qquad (1.2)$$

де $a_k = \left| d_k + \sum_{i=1}^{K-1} g_i a_{k-1} \right|_2$, g_i – кодовий генератор псевдовипадкових чисел, K – розрядність кодера;

2) для формування значення Y_k використовуються два блоки рекурсивного згорткового коду C_1 та C_2. Обидва використовують одні і ті ж значення d_k, але в різній послідовності через наявність блоку затримки L_1 та

блоку проріджування на вході блоку C_2. При цьому, якщо виходи (Y_{1k}, Y_{2k}) кодувальних пристроїв C_1 та C_2 використовувались n_1 та n_2 разів відповідно, швидкість коду кожного з блоків буде дорівнювати:

$$R_1 = \frac{n_1 + n_2}{2n_1 + n_2}\ ,\ R_2 = \frac{n_1 + n_2}{2n_2 + n_1}; \tag{1.3}$$

3) для декодування прийнятої кодової послідовності використовуються два блоки декодування DEC_1 та DEC_2, що з'єднанні послідовно і використовують модифікований BCJR алгоритм [38], що працює за принципом найбільшої правдоподібності;

4) декодер DEC_1 використовується для отримання м'якого (зваженого) рішення. Він використовує значення інформаційного біта d_k, отримане з біта X_k кодової послідовності, для обчислення логарифму відношення правдоподібності:

$$\Lambda_1(d_k) = \log\left(\frac{P_r\{d_k = 1 / observation\}}{P_r\{d_k = 0 / observation\}}\right), \tag{1.4}$$

де $P_r\{d_k = i / observation\}, i = 0,1$ – апостеріорна ймовірність інформаційного біта d_k.

Отримане значення $\Lambda_1(d_k)$ використовуються декодером DEC_2 для отримання кінцевого значення прийнятого інформаційного біту d_k ;

5) окрім цього, на вхід кожного з декодерів надходить додаткова інформація у вигляді перевірних біт Y_k. Ця інформація потрапляє на вхід демультиплексора, який направляє її на вхід декодера DEC_1, якщо $Y_k = Y_{1k}$ і на вхід декодера DEC_2, якщо $Y_k = Y_{2k}$. Якщо в момент часу k на вхід демультиплексора нічого не поступає – на відповідні входи декодерів передається нульове значення.

Недоліком розглянутих методів є те, що процес декодування каскадних кодів досить складний та вимагає значних апаратних ресурсів і вносить додаткову затримку. При цьому, якщо використовувати згорткові коди, або

інші поточні коди, в процесі каскадного кодування, то має місце розмноження помилок.

У свою чергу, використання факторіальних кодів створює передумови для розробки нових методів кодування мовленнєвих сигналів, що реалізують єдину процедуру кодування/декодування та забезпечують інтегрований захист інформації. Дослідження таких методів та оцінка їх властивостей виконано в розділі 4.

1.5. Методи організації множинного доступу до каналу колективного користування

Розробка нових і вдосконалення існуючих способів побудови телекомунікаційних систем з множинним доступом до сих пір є актуальним завданням.

1.5.1. Метод організації множинного доступу на основі мультиплексування пакетів

У роботах [66] – [69] розглядається метод підвищення швидкості передавання даних технології LTE (Long Term Evolution) під час обробки коротких голосових повідомлень за рахунок мультиплексування даних від різних користувачів в один LTE пакет. Цей метод передбачає наступні дії:

1. сортування n користувачів (User Entity, UE) на k груп за значенням Modulation and Coding Scheme (MCS). При цьому кожна група використовує RG_j блоків ресурсів (Physical Resource Blocks, PRB) для передавання блоку даних групи, загальна кількість яких $RG^{total} = \sum_{j=1}^{k} RG_j$ має бути менше або дорівнювати кількості PRB для стандартного LTE пакета $RG^{total} = \sum_{i=1}^{n} R_i$, де R_i - кількість PRB для передавання повідомлення від UE_i в стандартному LTE;
2. вибірка і створення списку UEs, привласнення вибірці загального значення MCS. Залежно від якості каналу, можлива ситуація, коли

вибірка може містити *UEs* з різними значеннями MCS. В такому випадку вибирається менше значення MCS, що підвищує достовірність передавання, але вносить додаткову надмірність в блок даних;

3. формування блоку Downlink Control Information (DCI), який містить інформацію про Resource Block Assignment (RBA), MCS і позиції UE_i в блоці. Останній параметр призначений для демультиплексування на приймальній стороні;
4. сформований пакет LTE відправляється всім користувачам всередині мережі у так званому широкомовному режимі. При цьому кожен користувач отримує тільки свою частину даних з пакета.

Недоліком розглянутого методу є те, що структура пакету передбачає лише завадостійке кодування і дані ніяк не захищені від несанкціонованого читання. При цьому, кожен абонент системи має потенційну змогу отримати доступ до даних інших користувачі, оскільки пакети передаються в широкомовному режимі.

1.5.2. Методи організації множинного доступу на основі часового розділення

У роботі [70] запропоновано метод доступу до радіоканалу колективного користування з часовим розподілом, який дозволяє підвищити пропускну здатність каналу за спільного передавання голосових повідомлень і даних у одному кадрі за рахунок спільного використання інтервалів запиту і передавання даних від інших джерел під час пауз у мові.

Цей метод передбачає використання трьох типів часових інтервалів: запит на передавання голосу (voice request interval), запит на передавання даних (data request interval) і дані (information interval). Кожен інтервал запиту містить фіксовану кількість міні-слотів, які містять один пакет запиту. При цьому передбачено використання вільних слотів всередині інтервалу даних в якості додаткових інтервалів запиту, які можуть бути використаними абонентськими

терміналами (АТ) для передавання і голосу, і даних, з пріоритетом на передавання голосу.

Алгоритм заняття часових інтервалів передбачає наступні дії:

1. базова станція (БС) в широкомовному режимі відправляє короткі пакети всім абонентам мережі в кінці кожного міні-слота, який містить інформацію про колізії. Оскільки ці пакети невеликі, а затримки всередині мережі мізерно малі, передбачається, що вони миттєво доступні всім АТ (до початку наступного міні-слота);
2. АТ передає БС пакети запиту і в разі отримання підтвердження очікує до кінця відповідного інтервалу запиту, щоб дізнатися зарезервований їм інтервал (або інтервали) даних. У разі невдачі АТ знову намагається виконати передачу запиту в наступному кадрі;
3. БС підтримує динамічну таблицю активних АТ в мережі для управління ресурсами каналу. Запис в таблиці містить ідентифікатор терміналу, ідентифікатор віртуального каналу, виділені ресурси каналу і параметри якості обслуговування. Після успішного прийому пакета із запитом передавання від АТ, БС відправляє підтвердження і поміщає запит в чергу. БС виділяє ресурси каналу в кінці відповідного інтервалу запиту, якщо вони доступні. Якщо ресурси, необхідні для задоволення запиту, недоступні, запит залишається в черзі. АТ з поставленими в чергу запитами повинні постійно контролювати канал від бази до АТ. По завершенню виклику, передавання повідомлення, або коли активний термінал виходить з мережі (передача обслуговування), БС видаляє запис з таблиці через деякий заданий період часу;
4. АТ, який зарезервував ресурс, виконує передачу даних в межах свого зарезервованого слоту. У разі помилки передавання, АТ повторює спробу в наступних кадрах, поки передача не завершиться успішно. Проте, АТ може припинити передачу запитів, якщо всі його пакети застаріли. В цьому випадку АТ переходить в режим очікування.

Запропонований у [39] метод множинного доступу до радіоканалу з резервуванням пакетів (Packet Reservation Multiple Access, PRMA) надає абонентам спільний доступ до ресурсів каналу використовуючи механізм часового поділу (TDMA). Зокрема, модифікації запропонованого методу описано в роботах [40], [71] – [73].

Основними положеннями PRMA є:

1. часові слоти об'єднуються в кадри, кожен слот всередині кадру позначається як «вільний» і «зарезервований», виходячи з підтверджень від БС в кінці кожного слоту. Кожен термінал містить регістр резервування кадру, де кожен біт, встановлений в 0, означає, що відповідний слот НЕ зарезервований, інакше - 1. Стан регістра оновлюється після кожного підтвердження від БС;

2. щоб почати передачу періодичної інформації, термінал намагається передати перший пакет, використовуючи перший не зарезервований часовий слот. У разі успіху, БС інформує всі інші термінали про резервування слоту і забороняє його використання. Одного разу зарезервувавши слот, термінал вільно використовує його для передавання даних у всіх наступних кадрах. Якщо ж виникає колізія, то з певною ймовірністю, термінал повторить спробу заняття наступного не зарезервованого часового слоту. Так буде тривати до тих пір, поки не надійде підтвердження від БС;

3. термінал оповіщає БС про завершення передавання, залишивши зарезервований слот порожнім в поточному фреймі, після чого БС оповіщає інші термінали;

4. у разі передавання випадкових інформаційних пакетів, термінал виконує схожі дії з п.3, але після успішного відправлення пакета термінала не резервує часовий слот і за наявності декількох пакетів для відправки, він намагається зайняти наступні вільні часові слоти.

Недоліком розглянутих методів є те, що інформація джерела не захищена від несанкціонованого читання, що вимагає попереднього шифрування даних. Також зазначимо, що ці методи використовують службові дані для утримання

синхронізму, що вносить додаткову надлишковість і знижує пропускну здатність каналу.

Використання ФСЧ створює передумови для розробки нового класу систем множинного доступу, здатних до синхронізації по робочому сигналу. Дослідження таких систем та оцінка їх властивостей виконано в розділі 5.

1.6. Цілі та задачі дисертаційного дослідження

Мета дослідженя полягає у розробці методів і засобів забезпечення інтегрованого захисту інформації в телекомунікаційних системах множинного доступу на основі факторіального кодування даних.

У першому розділі дисертації виконано аналіз існуючих методів канального кодування мовленнєвих сигналів у системах загального призначення. Показано, що згорткові коди, які широко використовуються в телекомунікаційних системах, направлені лише на виправлення помилок, мають малу швидкість коду і не забезпечують інтегрованого захисту інформації. У той же час, каскадні коди, що використовуються в мережах мобільного зв'язку, базуються на використанні згорткових кодів і вимагають більших апаратних затрат на їх реалізацію.

Виконано аналіз методів організації множинного доступу до каналу колективного користування. Показано, що ці методи використовують службові дані для утримання синхронізму, що вносить додаткову надлишковість і знижує пропускну здатність каналу, а також вимагають попереднього шифрування даних з метою забезпечення криптографічного захисту інформації. Разом з тим, розглянуто відомі методи забезпечення інтегрованого захисту інформації, що передбачають захист від помилок, що виникають у каналах зв'язку, несанкціонованого доступу до інформації, а також від нав'язування хибних даних. Встановлено, що використання факторіальних кодів для зазначених цілей є ефективним. Крім того, факторіальні коди володіють властивістю самосинхронізації, що підвищуєть їх ефективність.

Аналіз методів факторіального кодування показав, що вони вразливі до помилок парної кратності, таких, що призводять до трансформації однієї перестановки з дозволеної множини в іншу перестановку, що належить цій же множині. З метою підвищення стійкості факторіальних кодів до помилок парної кратності, зокрема $t = 2,4$, сформульовано задачі роботи, які полягають у розробці методів підвищення достовірності передавання даних у системах з нероздільним факторіальним кодуванням.

Ще одним напрямком досліджень, пов'язаним з факторіальним кодуванням, є процес формування сигнально-кодової конструкції для нероздільних методів факторіального кодування, зокрема факторіального кодування з відновленням даних за перестановкою. З метою забезпечення бажаного рівня достовірності передавання даних при максимальній швидкості коду, сформульовано наступну задачу дисертаційної роботи, яка полягає у розробці методу формування сигнально-кодової конструкції для систем *з* факторіальним кодування з відновленням даних за перестановкою.

Аналіз згорткових кодів показав, що сформована кодова послідовність містить інформаційні символи у відкритому вигляді, що накладає обмеження на використання цих кодів під час передавання конфіденційної інформації. Також зауважимо, що швидкість таких кодів не перевищує 1/2, тобто одному інформаційному символу ставиться у відповідність два символи кодової послідовності. У той же час, каскадні коди, за рахунок послідовного використання декількох кодів, дозволяють вирішити проблему захисту інформації від несанкціонованого читання, але вносять додаткову надлишковість, а також збільшують апаратні витрати. Сформульовано наступну задачу дисертаційного дослідження, яка полягає у розробці методів декодування факторіальних кодів з відновленням даних за перестановкою для систем передавання мовленнєвої інформації реального часу.

Аналіз існуючих методів забезпечення множинного доступу до каналу колективного користування з використанням механізму TDMA показав, що більшість з них передбачають наявність службових пакетів для реалізації

механізму резервування / звільнення часових інтервалів під передачу даних, що зменшує загальну пропускну здатність каналу. Також у цих методах не реалізовано єдину процедуру забезпечення інтегрованого захисту інформації. Виходячи з цього, сформульовано ще одну задачу даної роботи, яка полягає в розробці методу побудови телекомунікаційних систем множинного доступу з використанням нероздільного факторіального кодування, який не потребує окремого каналу синхронізації і забезпечує інтегрований захист інформації, що є заключною задачею дисертаційної роботи.

Виходячи з цього, задачами роботи є:

- розробка методів підвищення достовірності передавання даних у системах з нероздільним факторіальним кодуванням;
- розробка методу формування сигнально-кодової конструкції для систем *з* факторіальним кодування з відновленням даних за перестановкою;
- розробка методів декодування факторіальних кодів з відновленням даних за перестановкою для систем передавання мовленнєвої інформації реального часу;
- розробка методу побудови телекомунікаційних систем множинного доступу з використанням нероздільного факторіального кодування.

Поставлені задачі дисертаційного дослідження несуть наукову новизну і практичну цінність, а їх вирішення дозволить досягти поставленої мети.

1.7. Висновки

У першому розділі:

– досліджено сучасний стан предметної області дисертаційної роботи: виконано аналіз методів поєднання функцій криптографії та завадостійкого кодування, зокрема, на основі використання факторіального кодування; виконано аналіз методів кодування мовленнєвих сигналів на основі згорткових та turbo-кодів; виконано

аналіз методів забезпечення множинного доступу до каналу колективного користування, що використовують механізми TDMA;

- сформульовано цілі дисертаційного дослідження;
- визначено коло задач, що підлягають вирішенню для досягнення поставлених цілей.

РОЗДІЛ 2. ПІДВИЩЕННЯ ДОСТОВІРНОСТІ ПЕРЕДАВАННЯ ДАНИХ НЕРОЗДІЛЬНИМИ ФАКТОРІАЛЬНИМИ КОДАМИ

2.1. Вступ

У першому розділі дисертації поставлено підвищення достовірності передавання даних у системах з нероздільним факторіальним кодуванням. Розглянуто методи роздільного та нероздільного факторіального кодування. Методи нероздільного факторіального кодування, на відміну від методів роздільного кодування, забезпечують криптографічний захист інформації і тому саме вони являються предметом дослідження даного розділу дисертації. Однак, як було зазначено в першому розділі, факторіальні коди залишаються вразливими до помилок парної кратності, що виникають у каналах зв'язку та призводять до перетворення переданої перестановки в іншу перестановку з дозволеної множини. Описаний в [55] метод ФКЗЧІ дозволяє виявляти всі двократні помилки, але має низьку швидкість коду.

Метою цього розділу є представлення розроблених у рамках дисертаційного дослідження методів підвищення достовірності передавання даних у системах з нероздільним факторіальним кодуванням, що забезпечується за рахунок виявлення частини помилок парної кратності та обміну швидкості коду на достовірність передавання, а також оцінка їх характеристик у системах передавання даних з вирішувальним зворотнім зв'язком. Оцінці підлягають швидкість коду та ймовірність помилкового декодування, тобто ймовірність не виявленої кодом помилки.

2.2. Метод факторіального кодування з додатковими перевірними бітами

У [54] запропоновано метод підвищення достовірності передавання даних для ФКВД за рахунок використання природньої надлишковості факторіальних

кодів за умови $\frac{M!}{2^k} > 2$ або ж зменшення розміру інформаційної послідовності за $\frac{M!}{2^k} < 2$ і введення в неї додаткових перевірних біт. Однак цей метод не дозволяє виявити всі помилки, що призводять до трансформації початкової перестановки в іншу з дозволеної множини перестановок за рахунок однієї транспозиції.

У рамках дисертаційного дослідження виконано розробку методу підвищення достовірності ФКВД за рахунок додавання додаткових перевірних біт до вже сформованої перестановки, що дозволяє мінімізувати надлишковість і, як наслідок, отримати більшу швидкість коду. Основні результати розробки опубліковано в [75].

2.2.1. Опис методу

Як вже було зазначено, факторіальні коди взагалі, і ФКВД зокрема, забезпечують досить високу достовірність передавання даних за рахунок виявлення більшості помилок, що можуть виникнути в каналі зв’язку. Помилка декодування цих кодів виникає за умови, коли вектор помилки (обумовлений дією завад у каналі зв’язку), трансформує початкову перестановку в іншу перестановку дозволеної множини.

Суть запропонованого методу підвищення достовірності полягає в обчисленні і доповненні перестановки додатковими бітами, що характеризують ознаку парності числа інверсій та/або індексу перестановки.

Відповідно до [74], індексом перестановки $a_1, a_2, a_3, ..., a_n$ будемо називати суму всіх j, таких, що $a_j > a_{j+1}, 1 \le j < n$. Позначати індекс перестановки будемо через $ind(\pi)$.

Визначення 2.1 Згідно [55], кількість інверсій у перестановці – це кількість пар елементів (не обов'язково сусідніх), у яких наступний елемент має менший номер, ніж попередній.

Приклад. Знайти кількість інверсій у перестановці

(2, 3, 1, 6, 4, 5, 7).

Рішення. Перерахуємо всі пари:

(2,3), (2,1), (2,6), (2,4), (2,5), (2, 7),

(3,1), (3,6), (3,4), (3,5), (3,7),

(1,6), (1,4), (1,5), (1,7),

(6,4), (6, 5), (6, 7),

(4, 5), (4, 7) і (5, 7). Інверсії підкреслені – всього їх 4.

Наслідок 2.1. З визначення поняття інверсії випливає, що число інверсій – величина, що чисельно дорівнює сумі всіх елементів синдрому перестановки S_F.

Визначення 2.2. Факторіальним кодом з додатковими перевірними бітами (ФКДБ) називається нероздільний надлишковий код, у якому для підвищення достовірності передавання даних кодове слово ФКВД доповнюється додатковими перевірними бітами.

Нижче наведемо деякі з можливих способів формування перевірних біт.

1. Перестановка доповнюється одним перевірним бітом, який визначає парність числа інверсій перестановки, що розраховується як алгебраїчна сума всіх елементів синдрому за модулем 2:

$$b_0 = \left| \sum_{j=0}^{M-1} S_F(j) \right|_2,$$

де b_0 – перевірний біт, $S_F(j)$ – факторіальний коефіцієнт (елемент синдрому перестановки S_F) на *j*-ій позиції.

2. Перестановка доповнюється одним перевірним бітом, який визначає парність індексу перестановки:

$$b_0 = ind(\pi)$$

3. Перестановка доповнюється двома перевірними бітами, перший з яких визначає парність числа інверсій, другий – парність індексу перестановки:

$$b_0 = \left| \sum_{j=0}^{M-1} S_F(j) \right|_2, \; b_1 = ind(\pi)$$

Оскільки контроль числа інверсій дозволяє виявляти транспозиції символів усередині перестановки, далі будемо розглядати метод ФКДБ з одним додатковим бітом, що визначає парність числа інверсій. Це дозволить виявляти всі помилки кратності $t = 2$ та частину помилок більшої кратності.

У такому випадку метод факторіального кодування з додатковими перевірними бітами полягає в наступному:

1) на основі інформаційного вектору $A(x)$ формується синдром перестановки $S_F(x)$ потужністю M символів;
2) підраховується сума всіх елементів синдрому $S_F(x)$ і формується ознака парності числа інверсій b_0;
3) використовуючи отримане значення синдрому $S_F(x)$, завершується формування кодового слова ФКВД, що являє собою перестановку потужністю M;
4) до кодового слова ФКВД додається біт ознаки парності числа інверсій b_0 і передається приймачу.

Збільшення розміру кодового блоку на 1 біт призводить до незначного зменшення швидкості коду, але забезпечує виявлення всіх помилок ваги $t = 2$ та помилок більшої кратності, що призводять до зміни парності кількості інверсій.

Визначимо конструкцію пристроїв кодування та декодування факторіальних кодів з додатковими перевірними бітами.

2.2.2. Пристрій кодування та декодування факторіальних кодів з додатковими перевірними бітами

Реалізувати метод факторіального кодування з додатковими перевірними бітами можливо за допомогою пристрою, що містить блок кодування та блок декодування. Спрощена структурна схема блоку кодування показана на рис. 2.1.

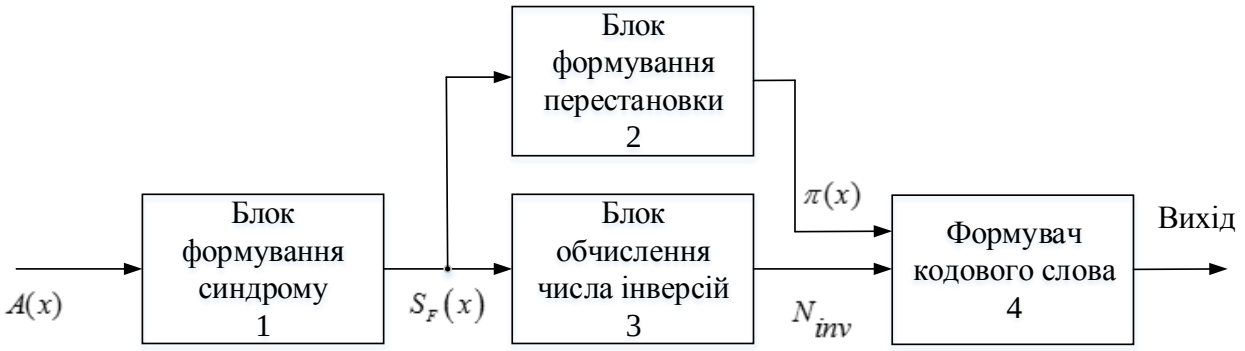


Рис. 2.1. Структурна схема блоку кодування ФКДБ

Блок кодування містить блок формування синдрому (1), який на основі інформаційного вектора $A(x)$ обчислює синдром $S_F(x)$, що поступає на вхід блоку формування перестановки (2), який на основі синдрому формує перестановку $\pi(x)$, і на вхід блоку обчислення числа інверсій (3), де обчислюється число інверсій N_{inv} як сума всіх елементів синдрому $S_F(x)$. Після цього перестановка $\pi(x)$ і число інверсій N_{inv} поступають на вхід блоку формувача кодового слова (4), де до самої перестановки додається біт ознаки парності числа інверсій.

Спрощена структурна схема блоку декодування ФКДБ показана на рис. 2.2.

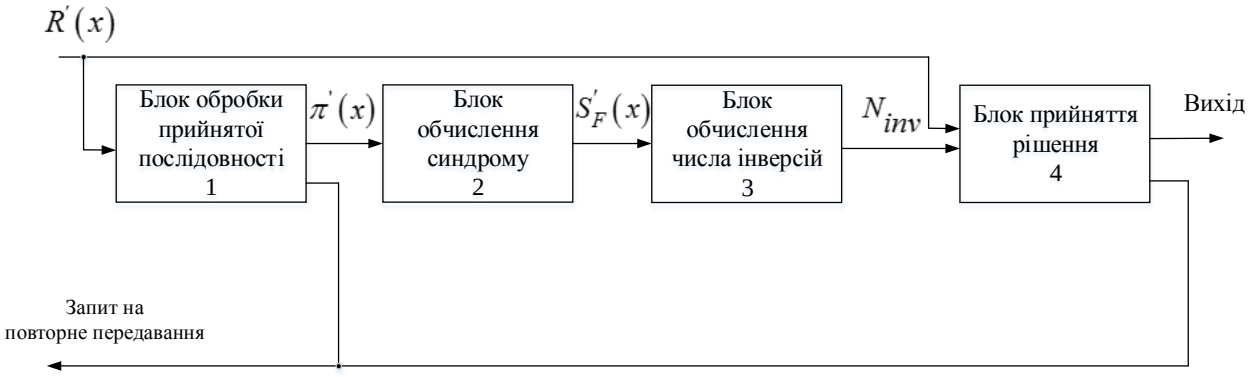


Рис. 2.2. Структурна схема блоку декодування ФКДБ

Блок декодування ФКДБ містить послідовно з'єднані блок обробки прийнятої послідовності $R'(x)$ (1), де відбувається перевірка, чи є прийнятий блок даних перестановкою, яка полягає у встановленні того факту, що кожен із символів множини $\{0;1;...;M-1\}$ зустрічається рівно по одному разу в

прийнятому блоці, блоку обчислення синдрому $S_F^{'}(x)$ (2) за отриманою перестановкою $\pi^{'}(x)$, блоку обчислення числа інверсій (3), де на основі синдрому $S_F^{'}(x)$ обчислюється число інверсій N_{inv} , і блок прийняття рішення (4), де відбувається порівняння обчисленого та прийнятого з каналу біта ознаки парності числа інверсій.

2.2.3. Оцінка показників достовірності передавання та швидкості коду

Перш за все зазначимо, що в блоці декодування виконується перевірка коректності прийнятої з каналу послідовності. У випадку, якщо в прийнятій з каналу послідовності пропущені та повторно застосовані будь-які символи, то такий блок у системі з ВЗЗ підлягає перезапиту.

Імовірність невиявленої декодером ФКДБ помилки є ймовірністю виникнення помилки, яка призводить до трансформації перестановки в іншу перестановку, а також трансформації перевірної частини таким чином, що в результаті перевірні символи підтверджують правильність прийнятої з помилкою перестановки. Так, наприклад, для ФКДБ з контролем парності числа інверсій невиявлена декодером помилка визначається ймовірністю виникнення однієї з двох подій:

1. трансформація перестановки в іншу із збереженням парності числа інверсій у ній, що відповідає парному числу транспозицій символів усередині перестановки, та збереження біта ознаки парності числа інверсій;
2. трансформація перестановки в іншу із зміною парності числа інверсій у ній, що відповідає непарному числу транспозицій символів усередині перестановки, та інверсія біта ознаки парності числа інверсій.

Зазначимо, що згідно [55], множини помилок, що призводять до трансформації однієї перестановки в іншу із збереженням та з інверсією парності числа інверсій чи індексу перестановки, – рівнопотужні, оскільки будь-яка перестановка з дозволеної множини перестановок може бути

трансформована в будь-яку іншу перестановку з цієї ж множини, при цьому половині перестановок властиве парне значення числа інверсій/індексу перестановки, а іншій – непарне. Виходячи з цього, ймовірність невиявленої помилки декодером ФКДБ:

$$\mathrm{P}_{ud}(FCAB, p_0) = \frac{P_{ud}(FCDR, p_0)}{2}, \qquad (2.1)$$

де p_0 - ймовірність бітової помилки, $P_{ud}(FCDR, p_0)$ - ймовірність невиявленої помилки ФКВД [51].

Для визначення ймовірності невиявленої декодером ФКДБ помилки розроблено програмну модель, що імітує середовище передавання даних з біноміальним розподілом помилок у каналі зв'язку. Модель дозволяє налаштувати ймовірність бітової помилки й отримати експериментальну оцінку ймовірності невиявленої помилки.

Для демонстрації результатів роботи програми на рис. 2.3 наведено графік залежності ймовірності не виявленої ФКДБ помилки від ймовірності бітової помилки p_0 для розміру інформаційної частини $k = 15$, порядку перестановки $M = 8$ і одним додатковим бітом b_0. На рис. 2.4 виконано порівняння ймовірності невиявленої помилки для ФКВДд і ФКДБ за ідентичних значень імовірності бітової помилки p_0 з використанням одного перевірного біту для порядку перестановки M. При цьому, оскільки при $k = 15$ показник надлишковості $\alpha < 2$, довжина інформаційної частини для ФКВДд, відповідно до [54], буде зменшена до $k = 14$ для забезпечення можливості введення одного додаткового біту r_{add} і збережені порядку перестановки M.

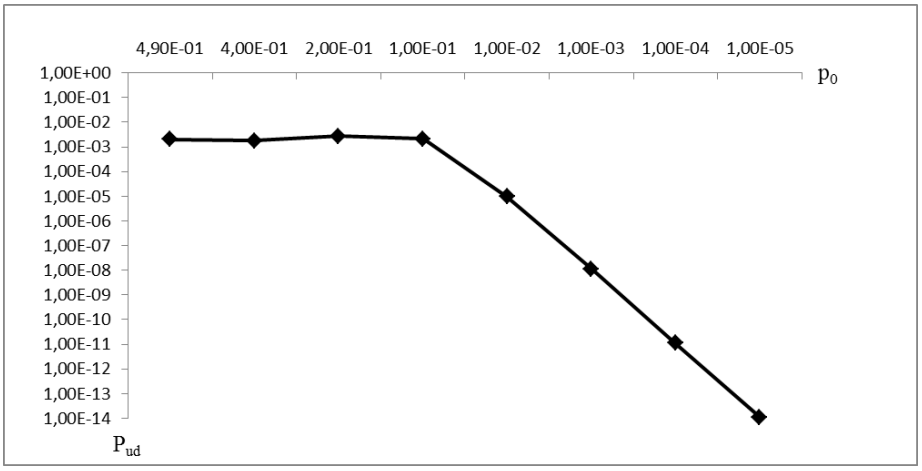


Рис. 2.3 Графік залежності ймовірності невиявленої помилки ФКДБ від ймовірності бітової помилки p_0 при $k=15$, $M=8$, m=1

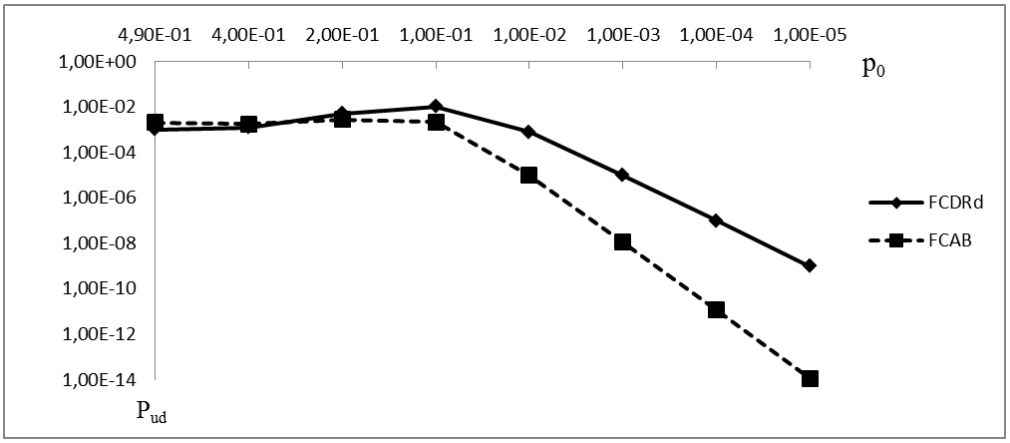


Рис. 2.4 Графік залежності ймовірності невиявленої помилки ФКДБ та ФКВДд від ймовірності бітової помилки p_0

Очевидно, що за рахунок накладання помилки на перевірні біти мають місце помилки декодування, такі що призводять до помилкових перезапитів даних у випадку, коли перестановка прийнята без помилок, але значення перевірних біт спотворено помилкою.

Кодове слово ФКДБ складається з перестановки порядку M, довжина якої, згідно [51], $n_{FCDR}=r_{FCDR}=M\cdot\lceil\log_2 M\rceil$ і блоку додаткових біт довжиною m. Повна довжина блоку $n=n_{FCDR}+b_0+...+b_{m-1}=n_{FCDR}+m$ біт. Швидкість ФКДБ:

$$\nu_{FCDRIC}=k/\left(n_{FCDR}+m\right). \tag{2.2}$$

На рис. 2.5 виконано порівняння швидкості ФКВДд і ФКДБ для різних довжин інформаційного вектора k.

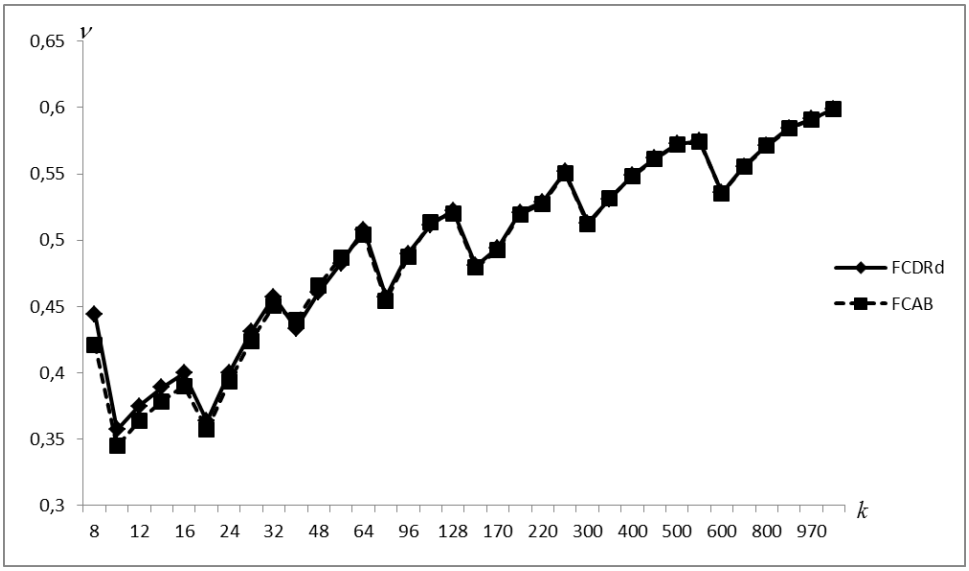


Рис. 2.5 Графік залежності швидкості коду ФКДБ та ФКВДд від довжини блоку даних на вході кодера

Рисунки 2.4 і 2.5 свідчать про те, що ФКДБ має меншу ймовірність невиявленої помилки і, як наслідок, забезпечує вищу достовірність передавання даних, ніж ФКВДд з $r_{add}=1$. При цьому різницею в швидкості коду між ФКДБ та ФКВДд при збільшенні значення k можна знехтувати. У той же час, ФКДБ не потребує зменшення довжини інформаційного вектору, коли показник надлишковості $\alpha<2$.

2.3. Метод факторіального каскадного кодування

Згідно [75], однією з переваг факторіальних кодів, і зокрема ФКВД, є властивість рівноважності. Це обумовлено тим, що будь-яке кодове слово ФКВД представлено у вигляді перестановки символів $\{0;1;...;M-1\}$, внаслідок чого кодове слово (за умови рівномірного двійкового кодування символів перестановки) має сталу кількість бітових одиниць і нулів. За рахунок цього ФКВД вразливі лише до помилок парної кратності.

Запропонований метод полягає в тому, що з метою підвищення достовірності передавання даних виконується попереднє перетворення

інформаційної послідовності за допомогою коду, який дозволяє частково виявляти помилки парної кратності.

Внутрішнім кодом обрано рівноважний код, властивості якого описано в [77], зовнішнім – ФКВД. Такий порядок слідування кодів дозволяє в повній мірі використати переваги ФКВД, а також зменшити ймовірність невиявленої каскадним кодом помилки за рахунок попереднього кодування інформаційного вектору рівноважним кодом. Далі такий код будемо називати факторіальним каскадним кодом.

Визначення 2.3. Факторіальним каскадним кодом (ФКК) називається нероздільний надлишковий код, який з метою підвищення достовірності передавання даних передбачає послідовне використання рівноважного та факторіального кодів.

Основні результати розробки опубліковано в [75], [78].

2.3.1. Опис методу

Рівноважний код передбачає заміну інформаційного блоку $A(x)$, що містить k біт, на послідовність $B(x)$, що містить m біт і має сталу кількість одиниць. Головною перевагою рівноважного коду є те, що він дозволяє виявляти всі помилки, окрім тих, що призводять одночасно до трансформації однакової кількості нулів та одиниць. Після цього інформаційна послідовність $B(x)$ перетворюється в перестановку $\pi(x)$, що містить n біт, за допомогою ФКВД. У результаті отримуємо послідовне перетворення:

$$A(x) \to B(x) \to \pi(x). \qquad (2.3)$$

Для забезпечення можливості зворотного перетворення повинна виконуватися наступна вимога: кожному слову $A(x)$ повинно відповідати лише одне значення $B(x)$, а кожному значенню $B(x)$ – лише одне значення $\pi(x)$. Отже, базова вимога до перетворення має вигляд:

$$M\{\pi(x)\} \geq M\{B(x)\} \geq M\{A(x)\}, \qquad (2.4)$$

де $M\{A(x)\} = 2^k$ – потужність множини інформаційних слів;

$M\{B(x)\}$ – потужність множини рівноважних слів, $M\{B(x)\} \geq C_m^{0.5m}$ або $M\{B(x)\} \geq C_m^{0.5(m\pm1)}$ для парних і непарних m відповідно. Далі покладемо, що m парне. Вираз $C_m^{0.5m}$ показує, скільки існує векторів розмірністю m і вагою Хеммінга $0.5m$. Такий вибір забезпечує найбільшу підмножину рівноважних слів;

$M\{\pi(x)\} = M!$ – потужність множини перестановок.

Спроектуємо пристрій кодування та декодування факторіальних каскадних кодів.

2.3.2. Пристрій кодування та декодування факторіального каскадного коду

Реалізувати запропонований метод можливо за допомогою пристрою, що містить блок кодування та блок декодування. Спрощену структурну схему блоку кодування показано на рис. 2.6.

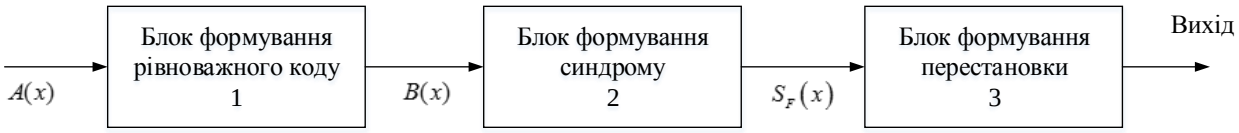


Рис. 2.6. Структурна схема блоку кодування ФКК

Блок кодування містить блок формування рівноважного коду (1), який виконує рівноважне кодування інформаційної послідовності $A(x)$, перетворюючи її на послідовність $B(x)$, блок формування синдрому (2), який на основі послідовності $B_m(x)$ обчислює синдром $S_F(x)$, що поступає на вхід блоку формування перестановки (3), який на основі синдрому формує перестановку $\pi(x)$, що видається в канал.

Спрощену структурну схему блоку декодування ФКК показано на рис. 2.7.

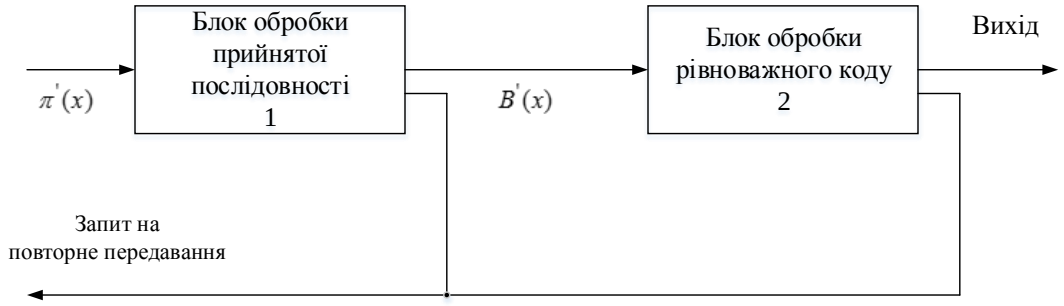


Рис. 2.7. Структурна схема блоку декодування ФКК

Процес декодування відбувається у зворотному до (2.3) порядку: $\pi(x) \to B(x) \to A(x)$. Блок декодування ФКК містить послідовно з'єднані блок обробки прийнятої послідовності $\pi'(x)$ (1), де відбувається перевірка чи є прийнятий блок даних перестановкою, яка полягає у встановленні факту того, що кожне ціле число (діапазону $[0, M-1]$) зустрічається в кодовому слові рівно один раз. Якщо ця умова не виконується, то формується запит на повторення блоку. Інакше з перестановки $\pi'(x)$ вилучається послідовність $B'(x)$, що відповідає отриманому кодовому слову, і передається на вхід обробки рівноважного коду (2), який визначає вагу кодового слова та його належність до дозволеної множини з 2^k кодових слів. Якщо прийнятий вектор належить до дозволеної множини і його вага становить $0.5m$, то декодер рівноважного коду перетворює його в прийнятий вектор $A'(x) = A(x) + \varepsilon(x)$, де $\varepsilon(x)$ – вектор помилки, що накладається на передане слово $A(x)$. Якщо прийнятий вектор належить до забороненої множини кодових слів або його вага не дорівнює $0.5m$, то формується запит на повторення блоку.

2.3.3. Оцінка показників достовірності передавання та швидкості коду

Під час розрахунку параметрів ФКК будемо спиратись на умову (2.4).

З виразу (2.3) слідує, що потужність множини рівноважних слів не менша потужності інформаційних слів, тому очевидно, що розмірність інформаційного вектору $A(x)$ $k \le \log_2 C_m^{0.5m}$. Враховуючи, що $\log_2 C_m^{0.5m}$ може не бути цілим числом,

$$k \le \left\lfloor \log_2 C_m^{0.5m} \right\rfloor \tag{2.5}$$

Примітка. Запис $\lfloor a \rfloor$ означає цілу частину від числа *a*.

Надлишковість рівноважного коду (за потужністю) можна визначити за допомогою співвідношення:

$$\alpha_1 = \frac{2^m}{2^k} = 2^{m-k} \tag{2.6}$$

Очевидно, що $\alpha_1 \to \min$ для $k = \left\lfloor \log_2 C_m^{0.5m} \right\rfloor$. Аналогічно, виходячи з умови $M\{\pi(x)\} \ge M\{B(x)\}$ і враховуючи, що $M\{\pi(x)\} = M!$, $M\{B(x)\} = 2^m$ отримуємо відношення $M! \ge 2^m$. Звідси слідує, що

$$m \le \left\lfloor \log_2 M! \right\rfloor \tag{2.7}$$

Вираз (2.7) слугує для визначення порядку перестановки M, виходячи з якого визначається кількість біт перестановки:

$$n = M \cdot \lceil \log_2 M \rceil \tag{2.8}$$

Примітка. Запис $\lceil a \rceil$ означає функцію округлення в більшу сторону числа a.

Надлишковість факторіального коду по відношенню до рівноважного визначається таким чином:

$$\alpha_2 = \frac{2^n}{2^m} = 2^{n-m} \tag{2.9}$$

Очевидно, що $\alpha_2 \to \min$ для $m = \left\lfloor \log_2 M! \right\rfloor$.

Надлишковість α_3 та швидкість ν ФКК визначаються наступним чином:

$$\alpha_3 = \alpha_1 \alpha_2 = 2^{n-k}, \tag{2.10}$$

$$\nu = \frac{k}{n}. \tag{2.11}$$

Використовуючи вирази (2.5) – (2.11), виконаємо оцінку параметрів ФКК для різних довжин *k* інформаційного слова $A(x)$. Результати оцінки наведено в таблиці 2.1.

Таблиця 2.1 – Оцінка основних параметрів ФКК

k	m	M	n	ν	α_1	α_2	α_3
12	15	8	24	0,54	2^3	2^9	2^{12}
43	45	16	64	0,67	2^2	2^{19}	2^{21}
114	118	32	160	0,71	2^4	2^{42}	2^{46}
210	215	50	300	0,7	2^5	2^{85}	2^{90}

З таблиці слідує, що ФКК має відносно високу швидкість коду та надлишковість, які зростають зі збільшенням розміру інформаційного блоку k і обумовлені властивостями ФКВД. Очевидно, що швидкість та надлишковість ФКК буде співпадати з відповідними параметрами ФКВД, окрім тих випадків, коли після перетворення $A(x) \to B(x)$ виникає необхідність збільшення потужності перестановки M: $2^k \le M! \le 2^m$. Також існуюча надлишковість може бути використана для підвищення достовірності передавання даних, як запропоновано в [54].

У загальному випадку, для рівноважного коду невиявленими будуть помилки, що не призводять до зміни кількості одиничних розрядів, тобто не змінюють вагу кодового слова. Така помилка можлива тоді і тільки тоді, коли рівно $t\left(t \le \lceil m/2 \rceil\right)$ одиничних біт перетворюються в нулі і рівно t нульових біт перетворюються в одиниці. За незалежних помилок та їх біноміального розподілу ймовірність такої події визначається наступним чином:

$$
\begin{aligned}
&P_{ud}\left(\mathrm{EC}, p_0\right) = \\
&= \sum_{t=1}^{\lceil m/2 \rceil} \left(\begin{array}{l} \left(C_{\lceil m/2 \rceil}^{t} \left(p_0\right)^t \left(1-p_0\right)^{\lceil m/2 \rceil - t} \right) \times \\ \times \left(C_{m-\lceil m/2 \rceil}^{t} \left(p_0\right)^t \left(1-p_0\right)^{m-\lceil m/2 \rceil - t} \right) \end{array} \right) = \\
&= \sum_{t=1}^{\lceil m/2 \rceil} \left(\begin{array}{l} C_{\lceil m/2 \rceil}^{t} C_{m-\lceil m/2 \rceil}^{t} \left(p_0\right)^{2t} \left(1-p_0\right)^{\lceil m/2 \rceil - t} \times \\ \times \left(1-p_0\right)^{m-\lceil m/2 \rceil - t} \end{array} \right)
\end{aligned}
\qquad (2.12)
$$

У випадку, коли розмірність вектору рівноважного коду m парна, вираз (2.13) приймає наступний вигляд:

$$P_{ud}\left(\mathrm{EC}, p_0\right) = \sum_{t=1}^{m/2}\left(C_{m/2}^{t}\left(p_0\right)^{t}\left(1-p_0\right)^{m/2-t}\right)^2 \qquad (2.13)$$

Оскільки каскадний код є поєднанням двох кодів, прийнятий з каналу блок даних, перш ніж потрапити на вихід декодера, проходить через зовнішній (у даному випадку ФКВД) та внутрішній (рівноважний код) контур декодування, тому ймовірність невиявленої помилки такого коду може бути визначена, як:

$$P_{ud}\left(FCC, p_0\right) = P_{ud}\left(FCDR, p_0\right)\cdot P_{ud}\left(\mathrm{EC}, p_0^{'}\right), \qquad (2.14)$$

де p_0 – імовірність бітової помилки на вході декодера ФКВД, $p_0^{'}$ – ймовірність бітової помилки на вході декодера рівноважного коду, яка приблизно дорівнює:

$$p_0^{'} \approx \frac{P_{ud}\left(FCDR, p_0\right)}{2\cdot\left(1-P_{dis}\left(FCDR, p_0\right)\right)} \ . \qquad (2.15)$$

Для оцінки ймовірності невиявленної помилки розроблено розрахунково-експериментальну модель, що імітує роботу системи передавання даних, що складається з джерела та приймача інформації, кодера та декодера ФКК і каналу зв’язку. Вхідними параметрами моделі є розмір інформаційного блоку k, що породжується джерелом, імовірність бітової помилки в каналі зв’язку p_0 і загальний обсяг вибірки. Помилки в каналі зв’язку розподіляються за біноміальним законом. Модель дозволяє оцінити ймовірність невиявленої помилки та швидкість коду. Результати моделювання представлено на рисунку 2.8 і 2.9.

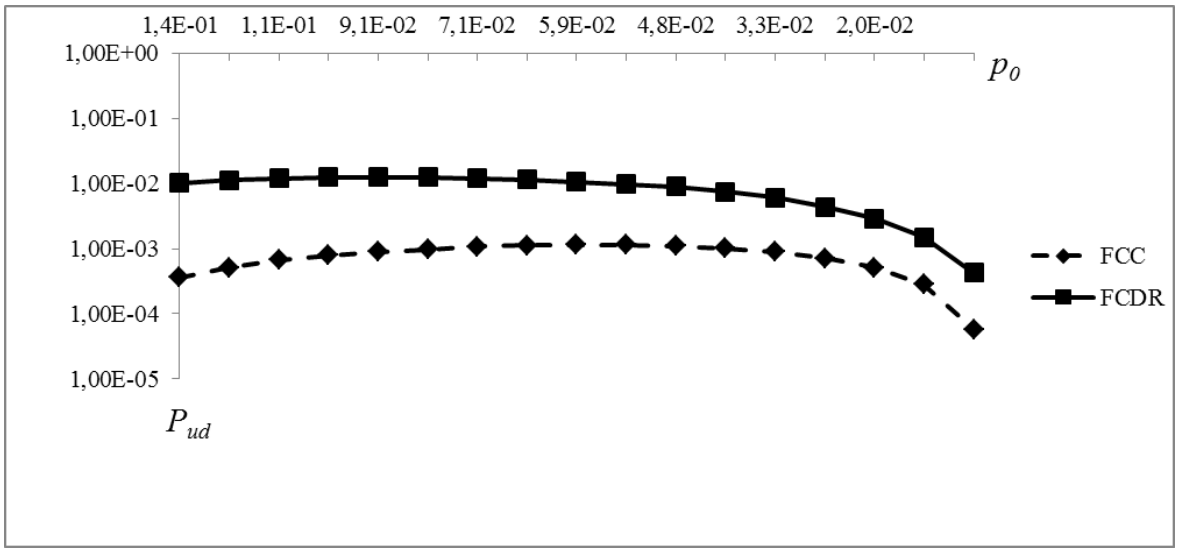


Рис. 2.8. Графік залежності ймовірності невиявленої кодом помилки від імовірності бітової помилки p_0 у каналі зв'язку для ФКК та ФКВД при $k = 15$

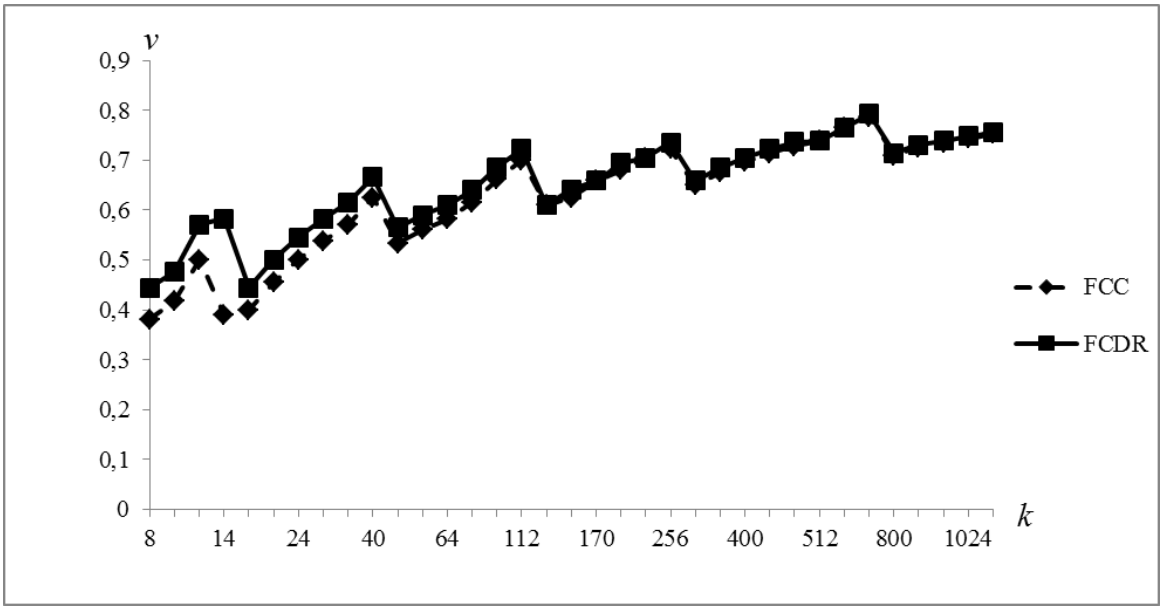


Рис. 2.9. Графік залежності швидкості коду ФКК та ФКВД від довжини блоку даних на вході кодера

Рисунки 2.8 і 2.9 показують, що ймовірність невиявленої ФКК помилки менше, ніж у ФКВД, а втратами в швидкості коду при збільшенні довжини блоку даних k можна знехтувати.

Виконаний аналіз властивостей ФКК показує, що запропонований каскадний код забезпечує більшу достовірність передавання даних, у порівнянні з ФКВД, в обмін на швидкість коду та може бути використаний в системах передавання даних з використанням каналів дротового і радіозв'язку УКХ та мікрохвильового діапазонів.

2.4.Висновки

У другому розділі:

- вперше запропоновано метод факторіального кодування з додатковими бітами, що передбачає розширення кодового слова ФКВД додатковим бітом ознаки парності числа інверсій, що дозволяє підвищити достовірність передавання даних за рахунок виявлення всіх помилок кратності $t = 2$;
- вперше запропоновано метод факторіального каскадного коду, який передбачає заміну інформаційного вектору $A(x)$ на послідовність $B(x)$, що містить сталу кількість одиниць, і подальше кодування методом ФКВД, що дозволяє підвищити достовірність передавання даних за рахунок можливості виявлення всіх помилок, що призводять до зміни ваги Хеммінга послідовності, отриманої на виході декодера ФКВД;
- виконано аналіз основних властивостей розроблених методів, а саме: ймовірності невиявленої помилки та швидкості коду. Показано, що ймовірність невиявленої помилки ФКДБ на 2-4 порядки нижче у порівнянні з ФКВДд при рівній швидкості коду. В той же час, ФКК дозволяє підвищити достовірність передавання за рахунок внесення додаткової надлишковості, при цьому ефективність ФКК значною мірою залежить від вибору внутрішнього коду.

Розроблені методи факторіального кодування інформації дозволяють підвищити достовірність передавання даних існуючих методів нероздільного факторіального кодування за рахунок внесення додаткової надлишковості. Виявлення помилок, внесених каналом зв’язку забезпечується за рахунок контролю характеристик перестановки у випадку зміни порядку символів в ній, а також зміни ваги інформаційного вектору.

Водночас представлені методи факторіального кодування направлені на підвищення ефективності існуючих факторіальних кодів і не вирішують повністю існуючі недоліки. Так, для ФКДБ вразливий до наступних помилок:

- помилок, що модифікують лише значення перевірної частини, в результаті чого відбувається повторний запит правильно прийнятої перестановки;
- помилок, що призводять до трансформації перестановки в перестановку при збереженні ознаки парності числа інверсій/індексу перестановки;
- помилок, що модифікують перевірну частину блоку даних таким чином, що вона вказує на правильність прийнятої перестановки.

У свою чергу, ФКК вимагає суттєвих апаратних затрат, при цьому не дозволяє повністю позбавитись помилок малої кратності, що значною мірою впливає на ефективність факторіального кодування. Тому необхідно виконати розробку методу факторіального кодування, який зможе виявляти всі помилки наперед заданої кратності і виправляти їх, що дозволить ефективно балансувати між достовірністю передавання даних і швидкістю коду. Результати розробки такого методу будуть представлені в наступному розділі дисертації.

РОЗДІЛ 3. МЕТОД ФОРМУВАННЯ СИГНАЛЬНО-КОДОВОЇ КОНСТРУКЦІЇ ДЛЯ ФКВД

3.1. Вступ

У другому розділі дисертації досліджено методи підвищення достовірності передавання даних у системах з нероздільним факторіальним кодуванням, зокрема, ФКВД. Розглянуто методи факторіального кодування з додатковими бітами та факторіального каскадного кодування. Однак, для певних прикладних задач необхідно забезпечити фіксоване, наперед задане, значення достовірності передачі даних. Рішенням цієї задачі є метод ФКВДвп, запропонований в роботах [56], [58], що передбачає використання для перенесення даних перестановок, що знаходяться на відстані Хеммінга $d_{\min} \geq 2t+1$, де t – максимальна кратність помилки, яка задається під час проектування системи. Запропонований метод дозволяє забезпечити виявлення всіх помилок наперед заданої кратності, що суттєво підвищує достовірність передавання. В той же час, швидкість коду залежить від обраної СКК. Тому актуальною задачею залишається розробка методу формування сигнально-кодової конструкції для систем *з* факторіальним кодування з відновленням даних за перестановкою, що забезпечить формування СКК максимальної потужності.

Метою цього розділу є дослідження методу формування сигнально-кодової конструкції, що забезпечує досягнення необхідного значення достовірності передачі при максимальній швидкості коду за рахунок формування підмножини перестановок на основі решіток. Оцінці підлягають швидкість коду та ймовірність невиявленої кодом помилки у системах передавання даних з ВЗЗ. Основні результати розробки опубліковано в [79], [80].

3.2. Сутність методу

Для оцінки сутності методу, спочатку визначимо деякі поняття та терміни.

Визначення 3.1 (з [58]**).** Сигнальним вектором називають перестановку з дозволеної множини перестановок, представлену в двійковому вигляді.

Визначення 3.2 (з [58]**).** Сигнально-кодовою конструкцією називається множина сигнальних векторів, що використовується для кодування інформації.

Визначення 3.3 Решіткою будемо називати сигнально-кодову конструкцію, в якій відстань Хеммінга $d_{\min}$ між будь-якими двома сигнальними векторами $\left(\pi_i, \pi_j\right)$ не менше деякого, наперед заданого значення. При цьому, така СКК повинна бути некритичною (інваріантною) до розміщення перестановки вузлів всередині решітки.

Основними параметрами решітки є:

- мінімальна відстань $d_{\min}$;
- кількість вузлів решітки N_{sv}, відстань між якими не менша за $d_{\min}$;
- потужність множини символів перестановки M.

Зазначимо, що збільшення мінімальної відстані $d_{\min}$ дозволяє підвищити достовірність передавання даних, але призводить до падіння швидкості коду через зменшення кількості вузлів решітки N_{sv}, що відповідають даній вимозі і, як наслідок, зменшення швидкості коду. Суть запропонованого методу формування СКК в метриці Хеммінга полягає в формуванні решітки з максимальною кількістю вузлів за відомих значень $d_{\min}$ та M.

Метод передбачає виконання наступних дій:

- обирається деяке початкове значення M, для якого будується підмножина не менше ніж з двох перестановок з попарними відстанями Хеммінга $d[\pi_i(x), \pi_j(x)] \geq (2t+1)$. У результаті (після M!·M! перевірок) отримаємо не менше двох перестановок $(\pi_i(x), \pi_j(x))$, де номери цих перестановок i і j в упорядкованій

множині всіх $M!$ перестановок з попарними відстанями $d[\pi_i(x), \pi_j(x)] \geq (2t+1)$;

- потужність множини символів M збільшується на одну одиницю, до значення $M+1$. Для цього, по черзі, над усіма вузлами сформованої решітки, що містять M символів, одночасно виконують одну із можливих підстановок і вставляють $(M+1)$-й символ між символами перестановки. Після чого відбирають тільки такі перестановки, у яких між будь-якою парою перестановок $(\pi_i(x), \pi_j(x))$ мінімальна відстань задовольняє умові $d[\pi_i(x), \pi_j(x)] \geq (2t+1)$. За такої процедури потужність кінцевої СКК збільшується в $(M+1)$ разів. Цю операцію повторюють до тих пір, поки число перестановок в підмножині перестановок не досягне або не перевищить наперед заданого значення, що забезпечує необхідну швидкість коду. На цьому процедура формування підмножини перестановок, інваріантних до обертання осей СКК завершена;
- після завершення процедури формування інваріантних перестановок (інваріантної підмножини перестановок) уточнюють число біт k, що переносяться кожною перестановкою цієї підмножини;
- уточнивши склад інваріантної підмножини, формують СКК у вигляді таблиці відображення множини слів джерела $A(x) \to \pi(x)$ для кодера і таблиці відображення множин $\pi(x) \to A(x)$ для декодера.

3.3. Оцінка показників достовірності передавання та швидкості коду

Для оцінки ефективності запропонованого методу формування СКК використаємо методи ФКВД [51] та ФКВДвп для СКК-2 (ФКВДвп-2) [57] - [59], де також дано теоретичну оцінку ймовірності невиявленої помилки.

Для визначення ймовірності невиявленої помилки розроблено програмну модель, що імітує середовище передачі даних з біноміальним розподілом

помилок у каналі зв'язку. Модель дозволяє для заданої якості каналу (ймовірності бітової помилки) оцінити наступні ймовірнісні характеристики:

- ймовірність виявлення помилки;
- ймовірність невиявлення помилки;
- ймовірність прийому кодового слова без помилок.

При цьому сума всіх ймовірностей дорівнює одиниці.

Слід зазначити, що внаслідок того, що кількість перестановок з попарними відстанями Хеммінга $d[\pi_i(x),\pi_j(x)]\geq(2t+1)$ становить $n\geq 2$, то за умови $n>2$ існує деяка множина можливих решіток з N_{sv} вузлами для заданого $d_{\min}$. В таблицях 3.1 та 3.2 приведено декілька різних решіток на 10 та 60 вузлів відповідно.

Таблиця 3.1 Решітки для $M=5$, $N_{sv}=10$, $d_{\min}=5$

Номер вузла	Решітка №1	Решітка №2	Решітка №3	Решітка №4
1	(4, 0, 1, 2, 3)	(4, 0, 1, 2, 3)	(4, 0, 1, 2, 3)	(4, 0, 1, 2, 3)
2	(4, 1, 2, 3, 0)	(4, 2, 3, 1, 0)	(4, 3, 0, 1, 2)	(4, 3, 2, 1, 0)
3	(0, 4, 2, 1, 3)	(1, 4, 2, 0, 3)	(3, 4, 0, 1, 2)	(0, 4, 3, 1, 2)
4	(1, 4, 3, 2, 0)	(3, 4, 1, 2, 0)	(2, 4, 3, 0, 1)	(3, 4, 0, 2, 1)
5	(1, 2, 4, 0, 3)	(0, 1, 4, 3, 2)	(2, 0, 4, 3, 1)	(1, 2, 4, 0, 3)
6	(2, 3, 4, 1, 0)	(2, 3, 4, 0, 1)	(1, 3, 4, 2, 0)	(2, 1, 4, 3, 0)
7	(2, 1, 0, 4, 3)	(0, 3, 1, 4, 2)	(1, 0, 3, 4, 2)	(0, 1, 2, 4, 3)
8	(3, 2, 1, 4, 0)	(2, 0, 3, 4, 1)	(0, 3, 2, 4, 1)	(3, 2, 1, 4, 0)
9	(0, 2, 1, 3, 4)	(1, 3, 0, 2, 4)	(0, 1, 2, 3, 4)	(0, 2, 1, 3, 4)
10	(1, 3, 2, 0, 4)	(3, 0, 2, 1, 4)	(3, 0, 1, 2, 4)	(3, 1, 2, 0, 4)

Таблиця 3.2 Решітки для $M=6$, $N_{sv}=60$, $d_{\min}=5$

Номер вузла	Решітка №1	Решітка №2	Решітка №3	Решітка №4
1	(6, 1, 3, 4, 2, 5)	(6, 1, 3, 4, 2, 5)	(6, 1, 3, 4, 2, 5)	(6, 1, 3, 4, 2, 5)
2	(6, 1, 4, 3, 2, 5)	(6, 1, 4, 3, 2, 5)	(6, 1, 4, 3, 2, 5)	(6, 1, 4, 3, 2, 5)
3	(6, 3, 1, 2, 4, 5)	(6, 3, 1, 2, 4, 5)	(6, 3, 1, 2, 4, 5)	(6, 3, 1, 2, 4, 5)
4	(6, 4, 1, 2, 3, 5)	(6, 4, 1, 2, 3, 5)	(6, 4, 1, 2, 3, 5)	(6, 4, 1, 2, 3, 5)
5	(6, 3, 5, 1, 4, 2)	(6, 3, 5, 1, 4, 2)	(6, 3, 5, 1, 4, 2)	(6, 3, 5, 1, 4, 2)

6	(6, 4, 5, 1, 3, 2)	(6, 4, 5, 1, 3, 2)	(6, 4, 5, 1, 3, 2)	(6, 4, 5, 1, 3, 2)
7	(6, 2, 3, 5, 1, 4)	(6, 2, 3, 5, 1, 4)	(6, 2, 3, 5, 1, 4)	(6, 2, 3, 5, 1, 4)
8	(6, 2, 4, 5, 1, 3)	(6, 2, 4, 5, 1, 3)	(6, 2, 4, 5, 1, 3)	(6, 2, 4, 5, 1, 3)
9	(6, 3, 2, 4, 5, 1)	(6, 3, 2, 4, 5, 1)	(6, 3, 2, 4, 5, 1)	(6, 3, 2, 4, 5, 1)
10	(6, 4, 2, 5, 3, 1)	(6, 4, 2, 5, 3, 1)	(6, 4, 2, 5, 3, 1)	(6, 4, 2, 5, 3, 1)
11	(2, 6, 5, 3, 4, 1)	(2, 6, 5, 3, 4, 1)	(2, 6, 5, 3, 4, 1)	(2, 6, 5, 3, 4, 1)
12	(2, 6, 5, 4, 3, 1)	(2, 6, 5, 4, 3, 1)	(2, 6, 5, 4, 3, 1)	(2, 6, 5, 4, 3, 1)
13	(5, 6, 4, 1, 2, 3)	(5, 6, 4, 1, 2, 3)	(5, 6, 4, 1, 2, 3)	(5, 6, 4, 1, 2, 3)
14	(5, 6, 3, 1, 2, 4)	(5, 6, 3, 1, 2, 4)	(5, 6, 3, 1, 2, 4)	(5, 6, 3, 1, 2, 4)
15	(4, 6, 2, 3, 1, 5)	(4, 6, 2, 3, 1, 5)	(4, 6, 2, 3, 1, 5)	(4, 6, 2, 3, 1, 5)
16	(3, 6, 2, 4, 1, 5)	(3, 6, 2, 4, 1, 5)	(3, 6, 2, 4, 1, 5)	(3, 6, 2, 4, 1, 5)
17	(1, 6, 5, 2, 3, 4)	(1, 6, 5, 2, 3, 4)	(1, 6, 5, 2, 3, 4)	(1, 6, 5, 2, 3, 4)
18	(1, 6, 3, 2, 4, 5)	(1, 6, 3, 2, 4, 5)	(1, 6, 3, 2, 4, 5)	(1, 6, 3, 2, 4, 5)
19	(5, 6, 1, 3, 4, 2)	(5, 6, 1, 3, 4, 2)	(5, 6, 1, 3, 4, 2)	(5, 6, 1, 3, 4, 2)
20	(5, 6, 1, 4, 3, 2)	(5, 6, 1, 4, 3, 2)	(5, 6, 1, 4, 3, 2)	(5, 6, 1, 4, 3, 2)
21	(2, 5, 6, 4, 1, 3)	(2, 5, 6, 4, 1, 3)	(2, 5, 6, 4, 1, 3)	(2, 5, 6, 4, 1, 3)
22	(2, 5, 6, 3, 1, 4)	(2, 5, 6, 3, 1, 4)	(2, 5, 6, 3, 1, 4)	(2, 5, 6, 3, 1, 4)
23	(4, 5, 6, 2, 3, 1)	(4, 5, 6, 2, 3, 1)	(4, 5, 6, 2, 3, 1)	(4, 5, 6, 2, 3, 1)
24	(3, 5, 6, 2, 4, 1)	(3, 5, 6, 2, 4, 1)	(3, 5, 6, 2, 4, 1)	(3, 5, 6, 2, 4, 1)
25	(2, 3, 6, 1, 4, 5)	(2, 3, 6, 1, 4, 5)	(2, 3, 6, 1, 4, 5)	(2, 3, 6, 1, 4, 5)
26	(3, 4, 6, 1, 2, 5)	(3, 4, 6, 1, 2, 5)	(3, 4, 6, 1, 2, 5)	(3, 4, 6, 1, 2, 5)
27	(1, 3, 6, 4, 2, 5)	(1, 3, 6, 4, 2, 5)	(1, 3, 6, 4, 2, 5)	(1, 3, 6, 4, 2, 5)
28	(1, 2, 6, 4, 5, 3)	(1, 2, 6, 4, 5, 3)	(1, 2, 6, 4, 5, 3)	(1, 2, 6, 4, 5, 3)
29	(4, 1, 6, 5, 2, 3)	(4, 1, 6, 5, 2, 3)	(4, 1, 6, 5, 2, 3)	(4, 1, 6, 5, 2, 3)
30	(5, 1, 6, 3, 4, 2)	(5, 1, 6, 3, 4, 2)	(5, 1, 6, 3, 4, 2)	(5, 1, 6, 3, 4, 2)
31	(3, 2, 1, 6, 5, 4)	(3, 2, 1, 6, 5, 4)	(3, 2, 1, 6, 5, 4)	(3, 2, 1, 6, 5, 4)
32	(4, 2, 1, 6, 5, 3)	(4, 2, 1, 6, 5, 3)	(4, 2, 1, 6, 5, 3)	(4, 2, 1, 6, 5, 3)
33	(1, 5, 3, 6, 2, 4)	(1, 5, 3, 6, 2, 4)	(1, 5, 3, 6, 2, 4)	(1, 5, 3, 6, 2, 4)
34	(1, 5, 4, 6, 2, 3)	(1, 5, 4, 6, 2, 3)	(1, 5, 4, 6, 2, 3)	(1, 5, 4, 6, 2, 3)
35	(2, 4, 3, 6, 5, 1)	(2, 4, 3, 6, 5, 1)	(2, 4, 3, 6, 5, 1)	(2, 4, 3, 6, 5, 1)
36	(5, 2, 4, 6, 3, 1)	(5, 2, 4, 6, 3, 1)	(5, 2, 4, 6, 3, 1)	(5, 2, 4, 6, 3, 1)
37	(3, 1, 4, 6, 5, 2)	(3, 1, 4, 6, 5, 2)	(3, 1, 4, 6, 5, 2)	(3, 1, 4, 6, 5, 2)
38	(2, 1, 5, 6, 3, 4)	(2, 1, 5, 6, 3, 4)	(2, 1, 5, 6, 3, 4)	(2, 1, 5, 6, 3, 4)
39	(5, 4, 2, 6, 1, 3)	(5, 4, 2, 6, 1, 3)	(5, 4, 2, 6, 1, 3)	(5, 4, 2, 6, 1, 3)
40	(5, 3, 2, 6, 1, 4)	(5, 3, 2, 6, 1, 4)	(5, 3, 2, 6, 1, 4)	(5, 3, 2, 6, 1, 4)
41	(2, 5, 1, 3, 6, 4)	(2, 5, 1, 3, 6, 4)	(4, 5, 3, 1, 6, 2)	(4, 5, 3, 1, 6, 2)
42	(2, 5, 1, 4, 6, 3)	(2, 5, 1, 4, 6, 3)	(3, 5, 4, 1, 6, 2)	(3, 5, 4, 1, 6, 2)
43	(4, 5, 3, 1, 6, 2)	(4, 5, 3, 1, 6, 2)	(2, 5, 1, 3, 6, 4)	(2, 5, 1, 3, 6, 4)

44	(3, 5, 4, 1, 6, 2)	(3, 5, 4, 1, 6, 2)	(2, 5, 1, 4, 6, 3)	(2, 5, 1, 4, 6, 3)
45	(4, 2, 3, 5, 6, 1)	(5, 4, 2, 3, 6, 1)	(1, 3, 5, 4, 6, 2)	(1, 4, 2, 5, 6, 3)
46	(3, 2, 4, 5, 6, 1)	(5, 2, 3, 4, 6, 1)	(1, 2, 4, 3, 6, 5)	(1, 3, 5, 4, 6, 2)
47	(1, 4, 2, 5, 6, 3)	(1, 3, 5, 4, 6, 2)	(4, 2, 3, 5, 6, 1)	(5, 2, 3, 4, 6, 1)
48	(1, 3, 5, 4, 6, 2)	(1, 2, 4, 3, 6, 5)	(5, 4, 2, 3, 6, 1)	(3, 2, 4, 5, 6, 1)
49	(3, 1, 2, 5, 6, 4)	(3, 1, 2, 5, 6, 4)	(4, 1, 5, 2, 6, 3)	(4, 1, 5, 2, 6, 3)
50	(4, 1, 5, 2, 6, 3)	(4, 1, 5, 2, 6, 3)	(3, 1, 2, 5, 6, 4)	(3, 1, 2, 5, 6, 4)
51	(3, 4, 1, 5, 2, 6)	(3, 4, 1, 5, 2, 6)	(3, 4, 1, 5, 2, 6)	(3, 4, 1, 5, 2, 6)
52	(4, 3, 1, 5, 2, 6)	(4, 3, 1, 5, 2, 6)	(4, 3, 1, 5, 2, 6)	(4, 3, 1, 5, 2, 6)
53	(1, 2, 3, 5, 4, 6)	(1, 2, 3, 5, 4, 6)	(1, 2, 3, 5, 4, 6)	(1, 2, 3, 5, 4, 6)
54	(1, 2, 4, 5, 3, 6)	(1, 2, 4, 5, 3, 6)	(1, 2, 4, 5, 3, 6)	(1, 2, 4, 5, 3, 6)
55	(2, 1, 3, 4, 5, 6)	(2, 1, 3, 4, 5, 6)	(2, 1, 3, 4, 5, 6)	(2, 1, 3, 4, 5, 6)
56	(5, 1, 4, 2, 3, 6)	(5, 1, 4, 2, 3, 6)	(5, 1, 4, 2, 3, 6)	(5, 1, 4, 2, 3, 6)
57	(4, 2, 5, 3, 1, 6)	(4, 2, 5, 3, 1, 6)	(4, 2, 5, 3, 1, 6)	(4, 2, 5, 3, 1, 6)
58	(4, 5, 3, 2, 1, 6)	(4, 5, 3, 2, 1, 6)	(4, 5, 3, 2, 1, 6)	(4, 5, 3, 2, 1, 6)
59	(3, 4, 2, 1, 5, 6)	(3, 4, 2, 1, 5, 6)	(3, 4, 2, 1, 5, 6)	(3, 4, 2, 1, 5, 6)
60	(4, 3, 2, 1, 5, 6)	(4, 3, 2, 1, 5, 6)	(4, 3, 2, 1, 5, 6)	(4, 3, 2, 1, 5, 6)

Визначення 3.4. Питомою помилкою $f_{per}(t)$ називається математичне сподівання кількості невиявлених помилок кратності t для кожного з вузлів решітки.

В таблицях 3.3 та 3.4 наведено значення питомої помилки на вузол решітки $f_{per}(t)$ для приведених решіток. Відзначимо, що для сформованих за допомогою запропонованого методу СКК, значення $f_{per}(t)$ не залежить від того, які саме перестановки входять в решітку.

Таблиця 3.3 Значення $f_{per}(t)$ *для решітки* $k=3,\ M=5,\ N_{sv}=10,\ d_{\min}=5$

Кратність помилки t	$f_{per}(t)$
1	0
2	0
3	0
4	0
5	0

6	4,5
7	0
8	1
9	0
10	1,5
11	0
12	0
13	0
14	0
15	0

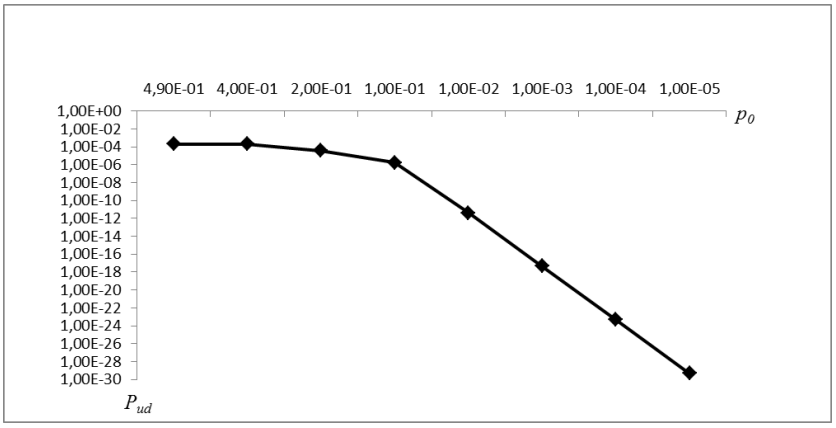
Таблиця 3.4 Значення $f_{per}(t)$ *для решітки* $k=5,\ M=6\ ,\ N_{sv}=60,\ d_{\min}=5$

Кратність помилки t	$f_{per}(t)$
1	0
2	0
3	0
4	0
5	0
6	9,375
7	0
8	12,063
9	0
10	6,063
11	0
12	2,875
13	0
14	0,625
15	0
16	0

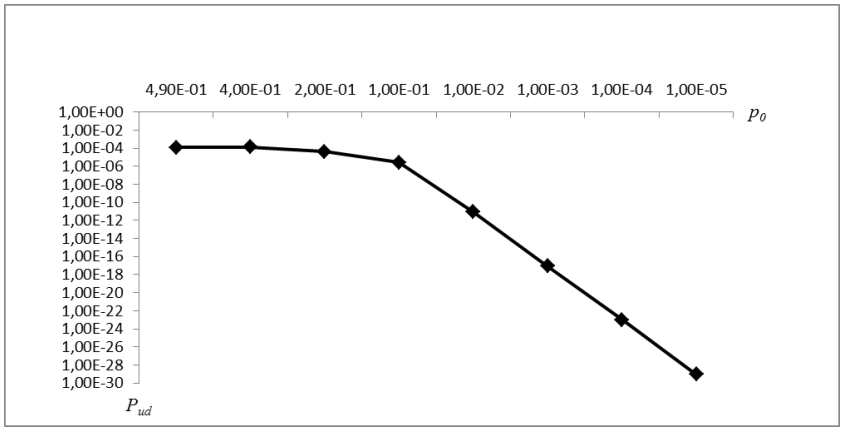
17	0
18	0

Дані, приведені в таблицях 3.3 та 3.4 показують, що сформовані решітки забезпечують виявлення всіх помилок непарної кратності, а також всіх помилок с кратністю $t \le 5$.

Результати роботи програми представлені на рис 3.1, *а* та 3.1, *б*. На рис. 3.1, *а* наведено графік залежності ймовірності невиявленої помилки ФКВД від імовірності бітової помилки p_0 при біноміальному розподілу векторів помилок для $k=3$, $M=5$, $d_{\min}=5$. На рис. 3.1, *б* наведено графік залежності ймовірності невиявленої помилки ФКВД від ймовірності бітової помилки p_0 при біноміальному розподілу векторів помилок для $k=5$, $M=6$, $d_{\min}=5$.



а)



б)

Рис. 3.1. Графіки залежностей оцінок ймовірностей невиявленої помилки ФКВД від ймовірності бітової помилки для $k=3$, $M=5$, $N_{sv}=10$(а); $k=5$, $M=6$, $N_{sv}=60$, (б) для $d_{\min}=5$

Рисунки 3.1, *а* та 3.1, *б* показують, що сформована СКК забезпечує ймовірність невиявленої помилки, що не перевищує значення 10^{-4} при $p_0 = 0,49$, тобто в умовах, близьких до стану обриву каналу зв'язку, за якого на вхід приймача поступає лише шум. При такій ймовірності бітової помилки загальновідомі системи передачі даних не зможуть працювати через втрату циклової синхронізації, в той час, коли ФК мають властивість до самосинхронізації. Це дозволяє використовувати цей метод формування СКК для кодування даних в умовах активної радіоелектронної боротьби (РЕБ), або в умовах, коли достовірність передавання даних важливіше швидкості коду.

Також розглянемо ймовірність невиявленої помилки для ФКВДвп-2. На рис. 3.2 наведено графік залежності ймовірності невиявленої помилки ФКВДвп-2 від ймовірності бітової помилки p_0 для розміру інформаційної частини $k = 5$, $M = 6$, $N_{sv} = 60$, $d_{\min} = 5$. На рис. 3.3 представлено результати порівняння ймовірностей невиявленої помилки ФКВД та ФКВДвп-2 для $k = 5$, $M = 6$, $N_{sv} = 60$, $d_{\min} = 5$.

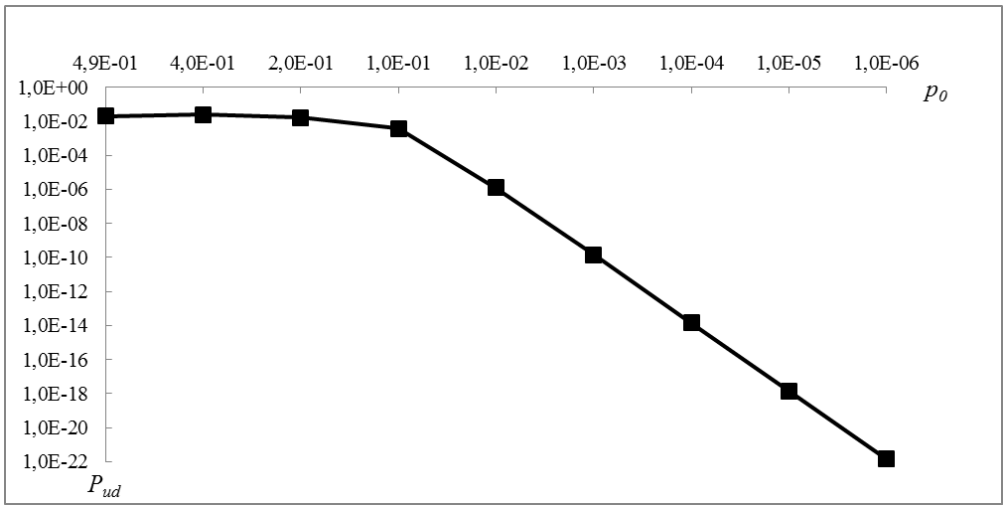


Рис. 3.2. Графіки залежностей оцінок ймовірностей невиявленої помилки ФКВДвп -2 від ймовірності бітової помилки p_0 для $k = 5$, $M = 6$, $N_{sv} = 60$, $d_{\min} = 5$

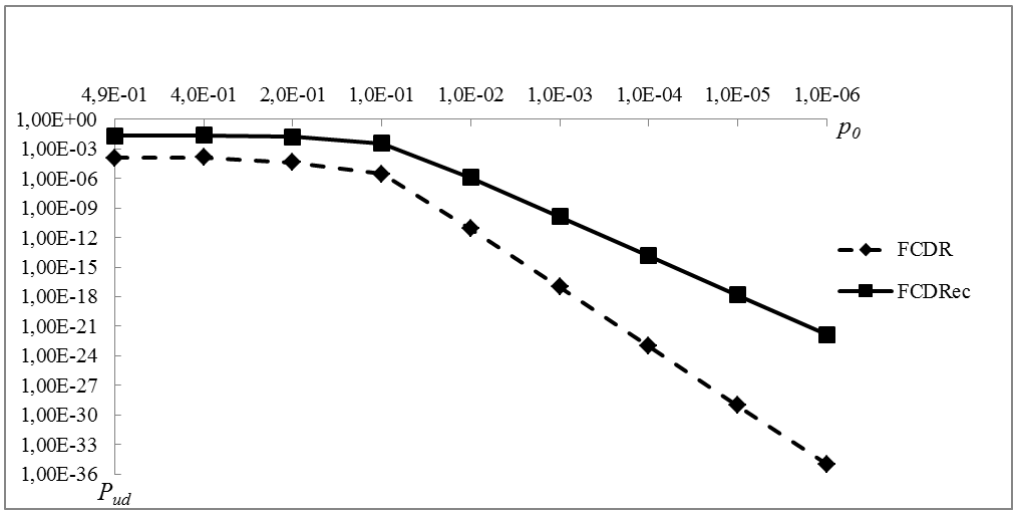


Рис. 3.3. Графіки порівняння ймовірностей невиявленої помилки ФКВД та ФКВДвп-2 для $k=5$, $M=6$, $N_{sv}=60$, $d_{\min}=5$

З рисунків 3.2 та 3.3 слідує, що ФКВДвп-2 має більшу ймовірність невиявленої помилки в порівнянні з ФКВД для заданої СКК, оскільки мінімальна відстань між вузлами решітки $d_{\min}=5$, що забезпечує виправлення помилок кратності $t \le 2$. Всі помилки, кратність яких $t > 2$, призводять до помилки декодування, в той час, коли ФКВД виправляє такі помилки шляхом повторного запиту блоку.

Виконаємо оцінку відносної швидкісті передачі, яка визначається як:

$$v_0 = v_1 v_2, \tag{3.1}$$

де $v_1 = k/n$ – швидкість коду, а v_2 – динамічна складова втрати швидкості в наслідок повторних запитів [44].

Згідно [58], значення v_2 обчислюється як:

$$v_2 = Q + P_{dis} + P_{ud}, \tag{3.2}$$

де Q – ймовірність правильного прийому блоку даних, P_{dis} – ймовірність виправлення помилки, P_{ud} – ймовірність невиявленої помилки.

На рис. 3.4 показано залежність швидкості коду ФКВДвп-2, СКК якого сформовано за допомогою запропонованого методу, для різних довжин інформаційного вектора k. На рис. 3.5 приведено значення відносної пропускної здатності для ФКВДвп-2 при різних значеннях ймовірності бітової помилки p_0 для $k=5$, $M=6$.

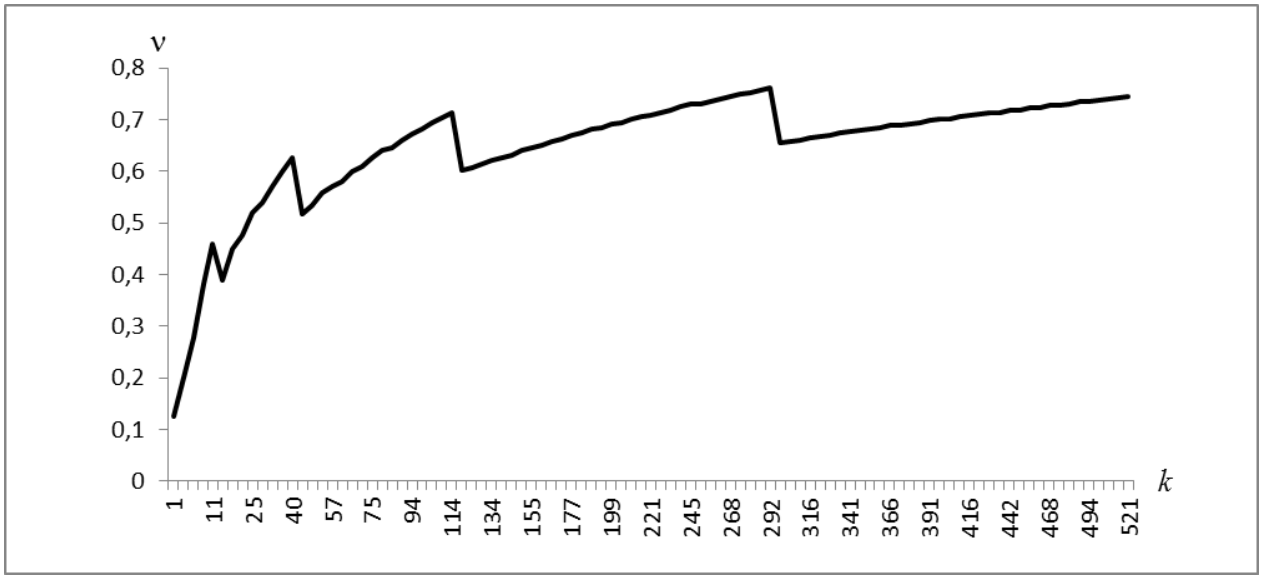


Рис. 3.4 Графік залежності швидкості коду ФКВДвп-2 від довжини блоку даних на вході кодера для $d_{\min} = 5$

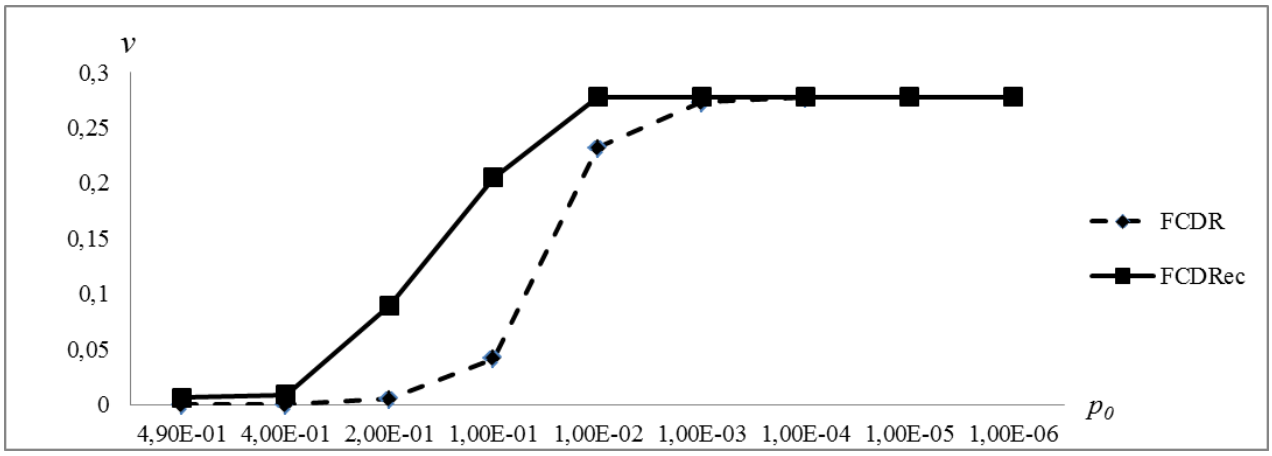


Рис 3.5 Графіки порівняння відносної швидкості передачі ФКВД та ФКВДвп-2 для $k = 5$, $M = 6$

З рисунка 3.4 слідує, що запропонований метод формування СКК забезпечує досить малу швидкість коду, яка зростає при збільшенні кількості вузлів в решітці.

В свою чергу, рисунок 3.5 свідчить про те, що ФКВДвп-2 має значно вищу відносну пропускну здатність для каналів низької якості ($0{,}1 \le p_0 \le 0{,}49$), але при цьому забезпечує гіршу достовірність передавання даних через велику кількість помилок кратності $t > 2$, що призводять до помилкового декодування.

Також, виконаємо порівняння з іншим методом факторіального кодування, описаним в [55], що також передбачає обмін швидкості на

достовірність передавання даних. Оскільки, цей метод передбачає виправлення помилок тільки шляхом повторного запиту блоку даних, будемо використовувати СКК для ФКВД. На рис. 3.6 представлено результати порівняння ймовірностей невиявленої помилки ФКВД та ФКЗЧІ при різних значеннях ймовірності бітової помилки p_0 та рівній швидкості коду для $k=5$, $M=6$.

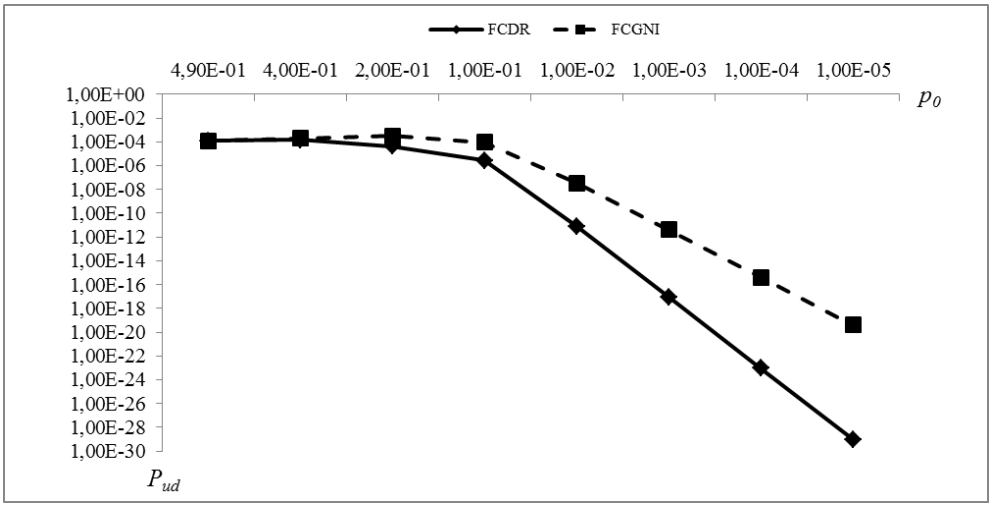


Рис. 3.6. Графіки порівняння ймовірностей невиявленої помилки ФКВД та ФКЗЧІ при різних значеннях ймовірності бітової помилки p_0 для $k=5$, $M=6$

Рисунок 3.6 показує, що запропонований метод формування СКК забезпечує більшу достовірність передавання даних при однаковій швидкості коду за рахунок виявлення всіх помилок кратності $t=2,4$, в той час, коли ФКЗЧІ дозволяє виявляти лише двократні помилки. Але при цьому, для заданого M існує множина класів $B_M(q,R)$ більшої потужності, що забезпечує більшу швидкість коду ФКЗЧІ у порівнянні з решітками. Однак, як вже було відзначено раніше, даний метод формування СКК дозволяє працювати в умовах РЕБ, коли ймовірність бітової помилки $p_0=0,49$, що дозволяє використовувати його в спеціалізованих телекомунікаційних системах.

Оцінка стійкості коду

Описаний метод формування СКК забезпечує можливість побудови решітки з числом вузлів $N_{sv} \geq 2^k$, тому у випадку, коли $N_{sv} > 2^k$, існує деяке

надлишкове число перестановок, які можуть бути використані при побудові інших решіток для цього ж ансамблю слів джерела. Як наслідок, отримуємо можливість створення множини СКК (таблиць перестановок) для одного ансамблю слів джерела $A(x)$. Ця множина може бути істотно розширена шляхом перестановок рядків таблиці. У результаті число різних таблиць буде визначатися виразом:

$$M_c = C_{N_{sv}}^{2^k} \cdot (2^k)!, \qquad (3.3)$$

де $C_{N_{sv}}^{2^k}$ – число можливих комбінацій вузлів, що використовуються для кодування інформації і формують таблицю замін;

$(2^k)!$ – множина перестановок рядків в таблиці замін.

Якщо ці таблиці тримати в секреті (перерахувавши їх у кодовій книзі), то перехоплення повідомлення-перестановки не дозволить без знання рядка книги, який використовується, прочитати повідомлення. Стійкість такого шифру визначається потужністю ключового простору, рівного числу M_c. З огляду на те, що число M_c дуже велике (наприклад, для $N_{sv} = 3360$ і $k = 11$ – $M_c = 3.4 \cdot 10^{6869}$), то ймовірність злому такого ключа методом грубої сили вкрай мала, що дозволяє використовувати його в якості сеансового ключа шифрування.

3.4. Висновки

У третьому розділі:

- вперше запропоновано метод формування сигнально-кодової конструкції на основі теорії решіток, що дозволяє отримати множину перестановок-носіїв інформації, рівновіддалених одна від одної на відстань $d_{\min}$;
- виконано аналіз основних властивостей СКК, сформованої за запропонованим методом, а саме: ймовірності невиявленої помилки, швидкості та стійкість коду. Показано, що запропонований в даному розділі метод забезпечує високу достовірність даних за рахунок виявлення всіх помилок наперед заданої кратності, але має малу

швидкість коду. При цьому, за рахунок надлишкового числа вузлів в решітці, можливе формування множини таблиць замін, що збільшує стійкість коду до злому методом «грубої сили»;

- виконано порівняння ймовірності невиявленої помилки та відносної пропускної здатності для ФКВД та ФКВДвп. Показано, що ФКВДвп має значно вищу відносну пропускну здатність, але програє по достовірності передачі ФКВД.

Розроблений метод формування СКК для факторіальних кодів дозволяє підвищити достовірність передавання даних за рахунок використання для перенесення інформації лише тих перестановок, що знаходяться на відстані Хеммінга не меншій за $d_{\min}$, що визначається на етапі проектування системи. Чим більше значення $d_{\min}$, тим вища достовірність передавання даних і тим менша швидкість коду, оскільки зменшується кількість перестановок-носіїв інформації.

Недоліком даного методу формування СКК є те, що він вимагає значних обчислювальних ресурсів для побудови решіток при збільшені потужності множини символів перестановки M. Також відзначимо, що ФКВДвп передбачає детермінований процес виправлення помилок, що призводить до помилкового виправлення і, як наслідок, невиявленої кодом помилки у випадках, коли кратність помилки перевищує задане значення t. Разом з тим, під час декодування ФКВДвп, у випадку виявлення декількох рівновіддалених сигнальних точок, використовується ВЗЗ з метою виправлення помилки шляхом повторного запиту блоку даних. Ці обставини унеможливлюють використання ФКВДвп при кодуванні мовленнєвих сигналів в режимі реального часу.

Тому необхідно виконати розробку методу факторіального кодування для систем передачі даних реального часу, який зможе виявляти та виправляти помилки із заданою точністю. Результати розробки такого методу будуть представлені в наступному розділі дисертації.

РОЗДІЛ 4. МЕТОДИ ФАКТОРІАЛЬНОГО КОДУВАННЯ МОВЛЕННЄВИХ СИГНАЛІВ

4.1. Вступ

У третьому розділі дисертації досліджено метод формування сигнально-кодової конструкції для систем *з* факторіальним кодування з відновленням даних за перестановкою на основі решіток, що полягає у використанні для перенесення інформації лише тих перестановок, що знаходяться на відстані Хеммінга не меншій за $d_{\min}$, що визначається під час проектування системи. Така СКК дозволяє виявляти всі помилки малої кратності ($t = 2,4$) і виправляти їх за рахунок повторного запиту. Однак такий метод відбору перестановок призводить до зменшення дозволеної підмножини перестановок і, як наслідок, до втрати пропускної здатності. Крім цього, час доставки блоку даних є випадковою величиною, що залежить від інтенсивності завад у каналі зв'язку і від числа повторних запитів блоків даних, прийнятих з помилкою. З урахуванням цих властивостей, існуючі методи факторіального кодування не можуть бути використані в телекомунікаційних системах реального часу, в тому числі для кодування мовленнєвої інформації.

Метою цього розділу є розробка методів декодування факторіальних кодів з відновленням даних за перестановкою для систем передавання мовленнєвої інформації реального часу, які забезпечують інтегрований захист в наступному обсязі:

- виправлення (з втратами) прийнятих з помилкою і виявлених кодом перестановок-носіїв вибірок мовленнєвого сигналу;
- захист від несанкціонованого прослуховування мовленнєвого повідомлення.

Також оцінці підлягають швидкість коду, імовірність помилкового декодування, тобто ймовірність невиявленої кодом помилки, а також стійкість

до злому методом «грубої сили». Основні результати розробки опубліковано в [83], [84].

4.2. Сутність методу

Використання факторіальних кодів для передачі мовленнєвих сигналів в режимі реального часу призводить до необхідності:

- відмови від точного відновлення прийнятих з помилкою вибірок;
- переходу до відновлення з втратами прийнятих з помилкою вибірок;
- врахування психофізичних властивостей сприйняття мови.

Перехід до декодування з втратами неминуче призводить до появи помилки відновлення вибірки декодером, яка проявляється у вигляді шуму, що супроводжує прийнятий сигнал. Помилку декодування *j*-ї вибірки (супроводжуючий *j*-у вибірку мовленнєвого сигналу шум декодування) ε_j визначимо наступним чином:

$$\varepsilon_j^2 = \left(A_j - A_j^{\wedge}\right)^2, \tag{4.1}$$

де: A_j – амплітуда *j*-ї вибірки, сформованої джерелом;

$A_j^{\wedge}$ – амплітуда *j*-ї вибірки, відновленої декодером.

Визначимо потужність шуму декодування W_{noise}:

$$W_{noise} = \sum_{j=1}^{V} \varepsilon_j^2 = \sum_{j=1}^{V} \left(A_j - A_j^{\wedge}\right)^2, \tag{4.2}$$

де V – число вибірок в сеансі зв'язку.

Відзначимо, що потужність сигналу в цьому ж сеансі зв'язку дорівнюватиме:

$$W_{signal} = \sum_{j=1}^{V} A_j^2 . \tag{4.3}$$

Звідси співвідношення сигнал / шум декодера в сеансі зв'язку дорівнюватиме:

$$h_{dec}^2 = \frac{W_{signal}}{W_{noise}} = \frac{\sum_{j=1}^{V} A_j^2}{\sum_{j=1}^{V} \left(A_j - A_j^{\wedge} \right)^2}. \qquad (4.4)$$

Це відношення є випадковою величиною, на яку впливають наступні компоненти:

– помилки відновлення, обумовлені властивостями алгоритму декодування;
– трансформація, під дією завад в каналі, прийнятої перестановки в іншу перестановку з дозволеної множини перестановок [51].

Часто зручно користуватися поняттям захищеності мовленнєвого сигналу, що дорівнює в даному випадку різниці рівнів сигналу і шуму декодування:

$$\Delta P = P_{signal} - P_{noise} = 10\lg W_{signal} - 10\lg W_{noise} = \\ = 10\lg \frac{W_{signal}}{W_{noise}} = 10\lg h_{dec}^2 \left(dB \right) \qquad . \qquad (4.5)$$

Для вирішення поставленого завдання будемо враховувати таку психофізичну особливість сприйняття мови, як придушення сильним сигналом слабкого. Це означає, що якщо мовний сигнал супроводжується шумами і завадами низького рівня, то це не заважає комфортному сприйняттю мови (по суті, в цих умовах шум невідчутний). Більш того, при відсутності шуму, яким зазвичай супроводжується мова, багато людей відчувають дискомфорт. Це обумовлено тим, що механізм сприйняття мови формувався тисячоліттями в умовах природного середовища проживання людини з притаманними цьому середовищі шумами. Тому перебування людини в середовищі з абсолютною відсутністю шуму є протиприродним і може викликати негативну реакцію організму людини. Тому забезпечення комфортного сприйняття мови завжди лежить в полі зору розробників систем мовленнєвого зв'язку.

Відзначимо, що методи виправлення помилок з втратами визначаються об'єктом, над яким виконуються операції реконструкції. В даному контексті можна оперувати або з вибірками мовленнєвого сигналу, або з їх носіями-

перестановками. У будь-якому випадку для відновлення деформованих помилками перестановок (або вибірок) будемо застосовувати не алгебраїчні, а імовірнісні методи корекції помилок - методи, що забезпечують максимум ймовірності ототожнення прийнятої перестановки (або безпосередньо вибірки) з їх істинними значеннями. Однак, для різних об'єктів реконструкції потрібні різні статистичні відомості про властивості потоку помилок в каналі зв'язку і чутливості об'єктів реконструкції до цих чинників. Дана обставина визначає відмінність методів декодування перестановок і вибірок, а також склад виконуваних операцій і результати їх застосування, в тому числі і за величиною шуму декодування. Тому однією з основних задач дослідження є оцінка шуму декодування для різних алгоритмів відновлення інформації. В цілому, підходи до реконструкції перестановок або вибірок зводяться до заміни деформованої помилками перестановки або вибірки найбільш імовірною перестановкою або вибіркою.

4.2.1. Метод відновлення прийнятих з помилкою перестановок в метриці Хеммінга

Суть методу відновлення перестановок в метриці Хеммінга полягає в порівнянні прийнятої з каналу зв'язку перестановки з кожною з перестановок таблиці замін. Для цього визначаються відстані Хеммінга від прийнятої з каналу зв'язку послідовності до всіх сигнальних векторів, використовуваних для перенесення інформації. Результатом виконання цієї операції є створення каталогу відстаней між прийнятої n-бітовою послідовністю $R^{\wedge}(x)$ і кожним з сигнальних векторів $R_i(x)$ таблиці замін, що відповідає виконанню операції:

$$d(i)=R^{\wedge}(x)\oplus R_i(x), \qquad (4.6)$$

де i - порядковий номер сигнального вектора в таблиці замін, $i\in\left[0,2^k-1\right]$.

Якщо врахувати, що $R^{\wedge}(x)=R(x)\oplus\varepsilon(x)$, де $\varepsilon(x)$ – n-бітовий вектор помилок, який вразив перестановку $R(x)$, то в результаті отримаємо $d(i)=R(x)\oplus\varepsilon(x)\oplus R_i(x)$. З цього випливає, що при послідовному переборі

сигнальних векторів $R_i(x)$ неминуче настане момент, коли $R_i(x)=R(x)$ і $d(i)=\varepsilon(x)$. Це означає, що в каталозі відстаней обов'язково присутній *n*-бітовий вектор помилок $\varepsilon(x)$, який вразив перестановку $R(x)$.

Таким чином, каталог відстаней можна трактувати як каталог векторів помилок, що трансформують передану перестановку в прийняту *n*-бітову послідовність. Серед цієї множини (потужністю 2^k векторів) обов'язково присутній реальний вектор помилки. Тепер врахуємо, що при незалежних бітових помилках в блоці з *n* біт ймовірність появи помилки заданої ваги t підпорядковується біноміальному закону:

$$p(t)=C_n^t p_0^t q_0^{n-t},$$

де p_0 – ймовірність бітової помилки, $q_0=1-p_0$.

При цьому, найбільш вірогідними (якщо математичне сподівання дискретної випадкової величини $np_0 \le 1$) є вектора помилок малої кратності. Звідси випливає, що вибір перестановки з таблиці замін за критерієм мінімуму відстані до табличного вектора еквівалентний вибору найбільш ймовірного вектора завади. Виходячи з цього, з каталогу відстаней вибираються перестановки з мінімальною відстанню, які заносяться в каталог кандидатів на заміну прийнятої з помилкою перестановки.

Якщо існує лише один сигнальний вектор з мінімальною відстанню, то прийнята послідовність ототожнюється з відповідною даному сигнальному вектору перестановкою. З неї витягується вибірка, яка видається споживачеві.

Якщо ж існує не менше двох сигнальних векторів $R_i(x)$ з однаковими мінімальними відстанями до прийнятої послідовності $R^{\wedge}(x)$, то з відповідних цим сигнальним векторам перестановок витягуються вибірки-кандидати на заміну прийнятої вибірки. Найкращим кандидатом на заміну прийнятої з помилкою вибірки є найближча (в метриці Евкліда) вибірка зі списку кандидатів по відношенню до попередньої вибірки, виданої споживачеві. Якщо така вибірка одна, то її приймають як декодовану вибірку. Якщо таких вибірок

кілька, то в якості декодованої вибірки випадковим чином вибирають одну з усього списку кандидатів на заміну. Процедура виправлення завершена.

Перераховані операції і порядок їх виконання забезпечують досягнення наступного результату:

- підвищення достовірності відновлення мовленнєвого сигналу, оскільки виправленню (з скінченною точності) підлягають всі без винятку прийняті з помилкою і виявлені кодом перестановки-носії вибірок;
- забезпечується можливість роботи в реальному масштабі часу за рахунок виключення потреби повторного запиту прийнятих з помилкою вибірок.

4.2.2. Метод відновлення прийнятих з помилкою перестановок шляхом лінійної інтерполяції

Алгоритм відновлення вибірок методом інтерполяції передбачає:

- виділення перестановок, в яких декодер не виявив помилок;
- вилучення вибірок з прийнятих перестановок, в яких декодер не виявив помилок;
- виділення пакета перестановок з виявленими помилками;
- відновлення вибірок методом лінійної інтерполяції на основі отриманих значень вибірок відповідних перестановок, що обрамляють пакет перестановок з виявленими помилками, і числа перестановок в цьому пакеті.

Для реалізації описаного алгоритму прийнята з каналу зв'язку n-бітова послідовність піддається перевірці коректності. Якщо за результатами перевірки коректності встановлено, що прийнята послідовність є перестановкою, то проводиться перевірка її належності до дозволеної підмножини перестановок (множини сигнальних векторів). При позитивному результаті цієї перевірки з отриманої перестановки витягується вибірка, яка записується в буферний регістр зсуву пристрою відновлення помилкових вибірок, що містить N комірок для $(k+1)$-бітових слів. При цьому k біт

відводяться для зберігання вибірки або для зберігання поточного номера послідовності в пакеті прийнятих з помилкою перестановок, а $(k+1)$-ий біт є індикатором, що визначає об'єкт зберігання в комірці пам'яті.

Звідси випливає, що перша прийнята з помилкою перестановка має номер 1, а остання прийнята з помилкою перестановка має номер, відповідний довжині пакета прийнятих з помилкою сигнальних векторів.

Це дозволяє виявляти пакети помилкових перестановок і відновлювати вибірки цього пакета відразу по виявленню пакета. Після його виявлення, процес відновлення вибірки зводиться до заміни в регістрі $(k+1)$-бітового слова відновленою вибіркою і, відповідно, біта індикатора.

В результаті, при виведенні вмісту буферного регістра споживачеві інформації виводиться відкоригований потік вибірок, супроводжуваний службовим бітом. Це означає, що на відміну від відновлення вибірок в метриці Хеммінга, метод інтерполяції вносить додаткову затримку, відповідну часу накопичення N вибірок. Величина цієї затримки (і, отже, довжина пакета помилок) визначається декількома факторами. В першу чергу, відзначимо, що величина цієї затримки не повинна перевищувати інтервалу кореляції мови, оскільки виправлення помилок методом інтерполяції некорельованих вибірок позбавлене сенсу. З огляду на те, що інтервал кореляції мови приблизно дорівнює 0,5 секунд [83], то $N \le 4000$. Реально величина N повинна бути істотно менше і визначатися параметрами і властивостями системи циклової синхронізації. Це пояснюється тим, що циклова синхронізація систем з факторіальним кодуванням виконується вибором часового положення вікна з M символів, в якому їх сума $\sigma = 0{,}5 \cdot M \cdot (M-1)$.

Є очевидним, що і в процесі пошуку синхронізму, і в процесі його утримання в межах вікна з M символів можуть перебувати як вірно, так і невірно прийняті перестановки. Тому рішення про встановлення або втрату синхронізму має прийматися на основі мажоритарного рішення - рішення за більшістю голосів. Це число голосів може бути різним для визначення моменту

досягнення синхронізму при пошуку і для визначення моменту втрати синхронізму. Будемо називати ці числа голосів коефіцієнтом накопичення по входу в синхронізм (k_{in}) і коефіцієнтом накопичення по виходу із синхронізму (k_{out}). Звідси випливає, що якщо число помилок в пакеті перевищить значення k_{out} у вікні з N перестановок, то система циклової синхронізації розцінить цю подію як втрату циклової фази, припинить декодування вибірок і перейде в режим пошуку синхронізму. Звідси випливає, що гранична кількість прийнятих з помилкою сигнальних векторів, що виправляються методом інтерполяції, N має задовольняти умові $N \le k_{out}$.

Зауважимо, що аналогічна ситуація має місце і при виправленні помилок в метриці Хеммінга. Якщо довжина пакета перестановок, прийнятих з помилкою, перевищить значення k_{out} системи синхронізації, то відбудеться фіксація події втрати циклової синхронізації і перехід в режим пошуку циклової фази. Таким чином, будь-який з розглянутих методів відновлення мовленнєвого сигналу орієнтований на виправлення помилок в пакеті перестановок, довжина якого не перевищує значення k_{out}. Зазвичай в системах множинного доступу з часовим поділом каналів $k_{out} \le 10$.

4.3.Пристрої кодування та декодування мовленнєвих сигналів в метриці Хеммінга та методом лінійної інтерполяції

Реалізувати алгоритм декодування мовленнєвих сигналів в метриці Хеммінга можливо за допомогою пристрою, спрощена структурна схема якого показана на рис. 4.1.

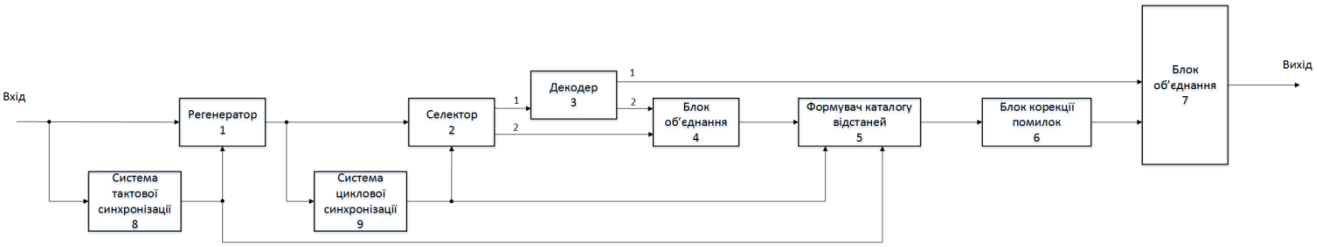


Рис. 4.1. Структурна схема декодера мовленнєвого сигналу з відновленням перестановок в метриці Хеммінга

Перш за все, відзначимо, що послідовності оцифрованих вибірок мовного сигналу, які передаються по каналу зв'язку, на виході каналу спотворені. Фронти одиничних посилок випадковим чином зміщуються щодо свого номінального положення, в результаті тривалість посилок є випадковою величиною. Тому, перш за все, необхідно відновити часові співвідношення між елементами сигналу. Для цього виконують попередню обробку - регенерацію, для чого приймач містить систему тактової синхронізації 8, що забезпечує синхронізацію генератора тактів приймача з тактовим генератором передавача. Синхронними тактами регенератор 1 стробує (опитує) прийнятий двійковий символ в найменш схильній до впливу завад частині - в її середині - і запам'ятовує її до наступного моменту стробування. Цим забезпечується максимальна достовірність прийому послідовності біт і відновлюються часові співвідношення посилок в їх безперервній послідовності. В результаті на виході регенератора формується послідовність біт, в якій можуть міститися помилки, зумовлені дією завад в каналі зв'язку в момент стробування. Ця послідовність подається на вхід системи циклової сигналізації 9 і на вхід селектора 2. Система циклової синхронізації 9 підраховує в плаваючому вікні суму з M символів прийнятої перестановки. Якщо ця сума дорівнює $\sigma = 0{,}5 \cdot M \cdot (M-1)$, то, швидше за все, межі перестановки визначені вірно. Зауважимо, що виносити рішення про встановлення (або втрату) циклової фази по одному спостереженню не можна, оскільки в умовах впливу завад сума символів у вікні є випадковою величиною. Рішення може бути прийняте на підставі обчислення математичного сподівання цього випадкового процесу або в разі, коли на

інтервалі спостереження більшість вибірок підтверджують або спростовують одну з гіпотез (синхронізм встановлений або синхронізм втрачений).

Коли стан синхронізму по циклам досягнуто, приймач переходить до обробки послідовності прийнятих з каналу перестановок, що містять вибірки мовленнєвого сигналу. Прийнята з каналу і записана в селектор 2 вибірка перевіряється на коректність. Якщо прийнята послідовність коректна (тобто є перестановкою), то вона видається декодеру 3 по шині 1 для визначення її приналежності до таблиці замін. При позитивному результаті перевірки - з неї витягується відповідна вибірка, яка по шині 1 декодера 3 через схему об'єднання 7 видається одержувачу.

Якщо прийнята перестановка не міститься в таблиці замін (тобто належить до забороненої підмножини перестановок), то по шині 2 декодера 3 вона видається на перший вхід першого блоку об'єднання 4, на другий вхід якого по шині 2 підключений вихід селектора 2.

У результаті на вхід формувача каталогу відстаней 5 подаються *n*-розрядні слова неперестановок і перестановок з забороненої підмножини.

Пристрій формування каталогу відстаней 5 визначає відстані Хеммінга між прийнятою з каналу зв'язку перестановкою і кожною з перестановок таблиці замін. Каталог відстаней видається в блок корекції помилок 6. У блоці корекції помилок 6 проводиться упорядкування списку перестановок, наприклад, по порядку зменшення відстані, і виділення в окремий список тільки перестановок з мінімальною відстанню Хеммінга. Цей список є списком кандидатів на заміну прийнятої з помилкою перестановки.

Якщо в цьому списку лише одна перестановка-кандидат, то з неї витягується вибірка, яка через блок об'єднання 7 виводиться споживачеві.

Якщо кандидатів на заміну декілька, то з кожного з них витягується вибірка, в результаті утворюється список кандидатів на заміну вибірки. З пам'яті декодера витягується попередня вибірка і обчислюється арифметична різниця амплітуд попередньої вибірки і кожної вибірки-кандидата. Вибірка-кандидат з мінімальним значенням різниці амплітуд приймається в якості

найбільш ймовірного значення переданої вибірки і виводиться споживачеві. На цьому процедура корекції помилок завершена, декодер переходить в режим очікування прийому наступної перестановки з каналу зв'язку, процес декодування повторюється за наведеним алгоритмом.

Реалізувати алгоритм декодування мовленнєвих сигналів методом інтерполяції можливо за допомогою пристрою, спрощена структурна схема якого показана на рис. 4.2.

Рис. 4.2. Структурна схема декодера мовленнєвого сигналу з відновленням перестановок методом інтерполяції

Приймальний пристрій містить систему синхронізації тактів 5, регенератор 1 і систему циклової синхронізації 2, які в цьому пристрої виконані ідентично декодеру в метриці Хеммінга і виконують ті ж самі функції.

Після завершення процедури синхронізації циклів прийняті з каналу *n*-бітові послідовності надходять на вхід декодера 3, в якому виконується перевірка коректності прийнятої перестановки.

Якщо прийнята послідовність є сигнальним вектором, з неї витягується вибірка, що представлена словом з *k* біт, до якого додається *(k+1)*-ий службовий біт - індикатор вмісту даної комірки пам'яті (наприклад, логічний нуль).

Якщо прийнята послідовність не є перестановкою або перестановкою, що не використовується для перенесення мовленнєвої інформації, в комірку пам'яті замість вибірки заноситься слово з *k* біт номера послідовності в пакеті прийнятих з помилкою перестановок, а *(k+1)*-му службовому біту-індикатору присвоюється символ двійковій одиниці.

Таким чином, якщо далі приймається декілька поспіль перестановок, уражених помилкою, то останній номер буде визначати довжину пакета

помилок *N*. Ця послідовність *(k+1)*-бітових слів з виходу декодера передається в блок відновлення вибірок 4 для виправлення пакетів помилок, структурна схема якого наведена на рис. 4.3.

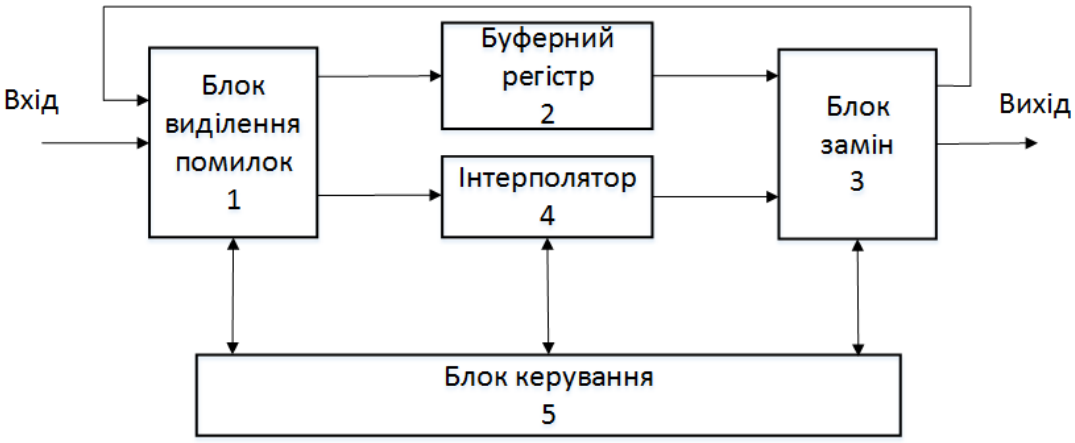


Рис. 4.3. Структурна схема блоку відновлення вибірок декодера мовленнєвого сигналу з відновленням перестановок методом інтерполяції

Слова, що поступають з декодера, потрапляють в блок виділення помилок 1, основним призначенням якого є виділення з вхідного потоку наступних трьох слів: вибірок $A_j^{\wedge}$ і $A_{j-(N+1)}^{\wedge}$, що обрамляють пакет помилок і показник довжини пакета – N.

За цими даними обчислюється крок інтерполяції $\Delta = \dfrac{A_j^{\wedge} - A_{j-(N+1)}^{\wedge}}{N+1}$, який передається в інтерполятор 4. Таким чином, після прийому першої правильно прийнятої вибірки, що закриває пакет прийнятих з помилкою перестановок, в інтерполятор завантажені наступні слова: $A_j^{\wedge}$, $A_{j-(N+1)}^{\wedge}$, N і Δ. Це дозволяє почати процедуру інтерполяції - процедуру обчислення і заміни вибірок, відповідних пошкодженим помилками перестановок. Для цього в блоці замін 3 в інтервалі між двома вибірками (рівним 125 мксек) виконується процедура заміни вмісту комірки на *k*-розрядне слово відновленої вибірки, а також переведення службового символу з двійковій одиниці на нуль.

Управління цим процесом здійснює блок управління 5.

Обчислення відновленої вибірки передбачає виконання операції $A'_{j-(N+1-i)} = A'_{j-(N+1)} + [i\Delta]$

4.4. Оцінка показників достовірності передавання та швидкості коду

Загальновідомо, що основні форманти мовленнєвого сигналу знаходяться в діапазоні $\Delta F = 4000\ Гц$. Тому, відповідно до теореми Котельникова, частота дискретизації мови дорівнює $F_{discr} \geq 8000\ Гц = 8000$ вибірок / сек, а інтервал дискретизації (інтервал між двома суміжними вибірками) $\tau_{discr} \leq 125$ мксек. Динамічний діапазон відтворення мови становить не менше 40 дБ. Звідси випливає, що для отримання необхідного динамічного діапазону число розрядів аналого-цифрового перетворення (АЦП) має лежати в межах $n_{ADC} = (13 \div 15)$.

Зауважимо, що вибір АЦП з числом розрядів $n_{ADC} = (13 \div 15)$ відповідає загальносвітовій практиці. Приймемо $n_{ADC} = 15$ як засіб забезпечення кращої якості відтворення мови. Звідси випливає, що для виконання умови $M! \geq 2^k$ необхідно, щоб $M \geq 8$. Для забезпечення мінімальної надмірності приймемо $M = 8$. При рівномірному кодуванні символів множини потужністю $M = 8$ число біт в кожному з символів дорівнює $l_r = 3$. Тоді перестановка, носій 15-бітової вибірки, буде містити $n = l_r \cdot M = 24$ біта, що відповідає швидкості коду $\nu = k / n = 0{,}625$.

Врахуємо, що в цифрових системах передачі (ЦСП) кожна вибірка кодується 8 бітами, тому швидкість передачі даних в лінії зв'язку (лінійна швидкість) дорівнює $B_{ЦСП} = F_{discr} \cdot n_{ADC} = 8 \cdot 10^3 \cdot 8 = 64$ *Кбіт / сек*. Цифровий канал з такими параметрами називають основним цифровим каналом (ОЦК), який є основою всіх ЦСП. При факторіальному кодуванні, кожна вибірка кодується 24 бітами (що в 3 рази більше, ніж в ОЦК), тому лінійна швидкість при факторіальному кодуванні теж буде в 3 рази вище і дорівнює $B_F = F_{discr} \cdot n = 8 \cdot 10^3 \cdot 24 = 192$ *Кбіт / сек*.

Для оцінки основних характеристик запропонованих методів відновлення вибірок мовленнєвого сигналу, деформованих завадою в каналі зв'язку, розроблена програмна модель тракту «джерело мовленнєвого сигналу - кодер мовленнєвого сигналу - канал зв'язку - декодер мовленнєвого сигналу - приймач мовленнєвого сигналу». Модель забезпечує оцінку основних параметрів мовленнєвого тракту: ймовірність виявлення / невиявлення помилок декодером ФКВД, оцінку величини помилки відновлення вибірок. На підставі цих даних визначаються потужність шуму декодування і захищеність мовленнєвого сигналу. Основним результатом експерименту по визначенню шуму, що супроводжує декодований сигнал, є залежність захищеності від ймовірності бітової помилки в каналі зв'язку. У моделі використаний ФКВД з параметрами $M=8,\ k=15,\ n=24,\ \nu=0.625.$

Розроблена програмна модель дозволяє змінювати характер перешкод в каналі зв'язку. З метою визначення основних параметрів запропонованих методів факторіального кодування мовленнєвих сигналів виконано моделювання процесу передачі голосової інформації для каналів з незалежними бітовими помилками і каналів з пакетуванням бітових помилок. Як джерело використовувалася фіксована вибірка реального мовленнєвого сигналу.

Модель каналу з незалежними бітовими помилками використовує біноміальний закон їх розподілу з одним параметром - ймовірністю бітової помилки p_0.

Для моделювання каналу з пакетуванням помилок використовувалася модель Гільберта-Еліота [41], [42]. Ця модель передбачає два стану каналу - «гарний» і «поганий». Кожен з них має свою ймовірність бітової помилки - p_{0g} і p_{0b} відповідно. Закон зміни станів каналу описується ланцюгом Маркова першого порядку і визначається перехідними ймовірностями P_{gb} і P_{bg}. Моделювання виконано для кабельних каналів, імовірнісні характеристики яких вибиралися на підставі даних з [86], [87]. При цьому для оцінки методів

декодування в умовах пакетування помилок використовувалася безумовна (середня) ймовірність бітової помилки:

$$p_0 = \left(P_{bg} \Big/ \left(P_{bg} + P_{gb} \right) \right) \cdot p_{0g} + \left(P_{gb} \Big/ \left(P_{bg} + P_{gb} \right) \right) \cdot p_{0b}. \qquad (4.8)$$

За результатами випробувань двох моделей декодера факторіального коду, що виправляє помилки, визначена захищеність мовленнєвого сигналу (різниця рівнів вихідного мовленнєвого сигналу і шуму декодування) в залежності від ймовірності бітової помилки в каналі зв'язку і характеру перешкод. Результати випробувань наведені на графіках рис. 4.4 – 4.6.

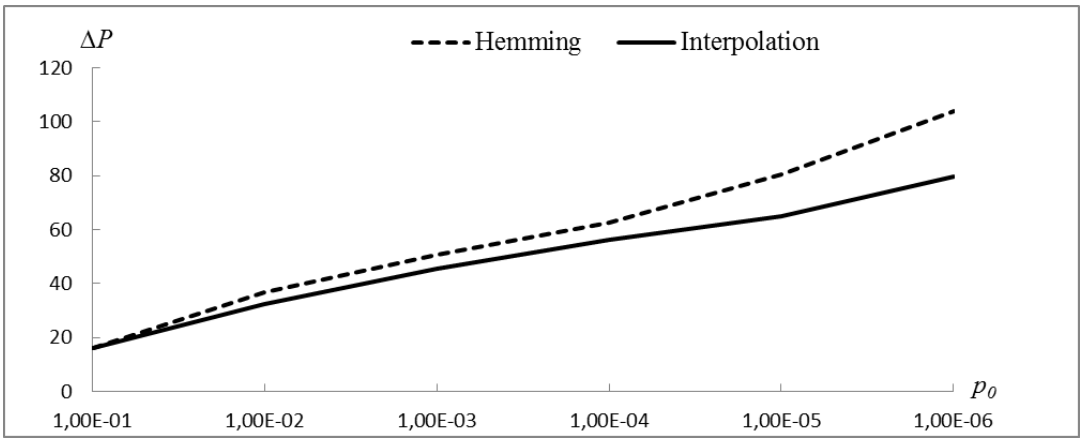


Рис. 4.4. Графік залежності захищеності мовленнєвого сигналу від шуму декодування в каналах з незалежними бітовими помилками при $k = 15,\ M = 8$

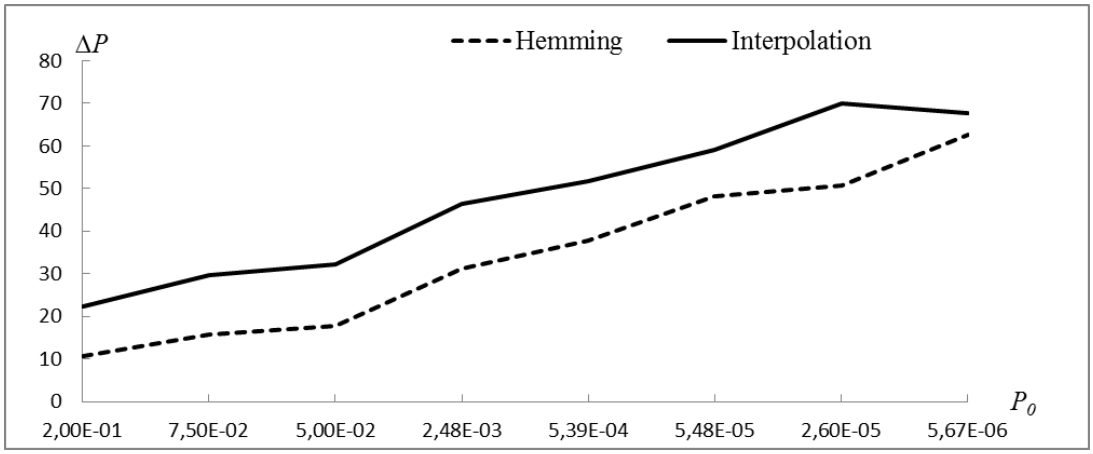


Рис. 4.5. Графік залежності захищеності мовленнєвого сигналу від шуму декодування в каналах з пакетуванням бітових помилок при $k = 15,\ M = 8$

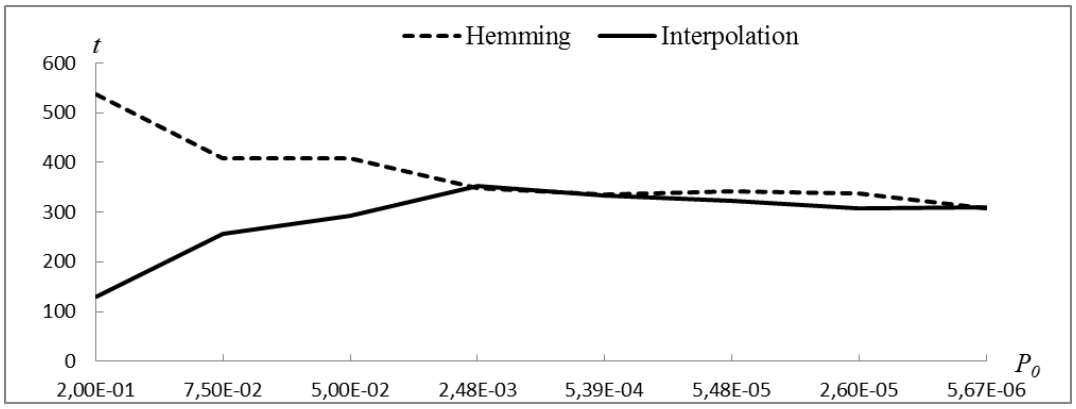


Рис. 4.6. Графік залежності середнього часу відновлення вибірок мовленнєвого сигналу від ймовірності бітової помилки p_0 при $k=15$, $M=8$

З наведених графіків можна зробити висновок, що кожен з методів декодування орієнтований на різні за статистичними властивостями завад типи каналів.

Рис. 4.4. показує, що захищеність декодера в метриці Хеммінга вище, ніж у декодера, що відновлює вибірки методом інтерполяції у всьому діапазоні значень ймовірності бітової помилки в каналі зв'язку з незалежними бітовими помилками. До цієї категорії відносяться канали, організовані в підземних кабельних лініях зв'язку, в тому числі і в волоконно-оптичних. Також сюди можна віднести лінії мікрохвильового радіозв'язку, що функціонують в сприятливих погодних умовах і без застосування до них засобів радіоелектронної протидії.

У свою чергу, рис. 4.5. показує, що декодер, який відновлює вибірки методом інтерполяції, орієнтований на канали з пакетуванням помилок і забезпечує більший рівень захищеності в порівнянні з декодером в метриці Хеммінга. До цієї категорії каналів можна віднести радіоканали короткохвильового діапазону в нормальних умовах і радіоканали будь-якого радіочастотного діапазону в умовах радіоелектронної протидії.

Відзначимо також, що процедура відновлення вибірок методом інтерполяції простіша в реалізації і вимагає менше ресурсів, ніж відновлення в метриці Хеммінга. З рис. 4.6. видно, що швидкодія декодера, що відновлює вибірки методом інтерполяції, значно вище при $10^{-3} < p_0 \le 10^{-1}$. При поліпшенні

якості каналу все частіше починають виникати переважно поодинокі помилки і швидкодія обох алгоритмів стає ідентичною.

Слід зазначити, що при $10^{-4} \le p_0 \le 10^{-2}$ шум, який супроводжує відновлений мовний сигнал при використанні алгоритму відновлення в метриці Хеммінга для каналів з незалежними помилками і методу інтерполяції для каналів з пакетуванням помилок, може бути віднесений до категорії комфортного шуму. При поліпшенні якості каналу зв'язку шумом декодування можна знехтувати.

4.5. Оцінка стійкості коду

Процес біективного відображення множин $\{A(x)\}$ і $\{R(x)\}$ за умови, що закон відображення тримається в секреті, по суті, відповідає процесу шифрування інформації. Стійкість такої криптосистеми визначається потужністю ключового простору. В умовах, коли закон відображення множин заданий таблицею замін, потужність ключового простору визначається потужністю множини таблиць замін. Надлишковість ФКВД (рівна $M!-2^k$ перестановок) забезпечує можливість створення різних, за складом і порядком їх розміщення перестановок, таблиць замін. Розглянуті в даному розділі методи забезпечують потужність ключового простору $\mu_{key} = C_{M!}^{2^k} \cdot \left(2^k!\right)$, де співмножник $C_{M!}^{2^k}$ визначає потужність ключового простору, обумовлену зміною складу перестановок в таблиці замін, а $2^k!$ – потужність ключового простору, обумовлену перестановками рядків таблиці замін. Представлене співвідношення дозволяє визначити стійкість криптосистеми до її злому методом «грубої сили» - послідовним перебором всіх можливих ключів множини μ_{key}. Зокрема, при $M=8$, $k=15$ отримаємо $\mu_{key} = C_{8!}^{2^{15}} \cdot \left(2^{15}!\right) = 7.34 \cdot 10^{142176}$.

Нехай перебір ключів виконує обчислювальний блок, продуктивністю $P_w = 10^{10}$ ключів / сек. В цьому випадку час, витрачений на перебір всіх ключів ключового простору, складе $t = \mu_{key} / P_w = 7.34 \cdot 10^{142166}$ с або $2.33 \cdot 10^{142159}$ років.

Припустимо, що продуктивність обчислювального блоку збільшується на величину 5%, 10% або 15% щорічно. Тоді, відповідно до [88], час, витрачений на перебір всіх ключів ключового простору μ_{key} з урахуванням підвищення продуктивності обчислювального блоку, складе

$$Y(5\%) = \log_{\left(1+\frac{5}{100}\right)}\left[\frac{5 \cdot 7.34 \cdot 10^{142166}}{100 \cdot 10^{10} \cdot 365 \cdot 24 \cdot 60 \cdot 60} + 1\right] = 6.71 \cdot 10^{6} \text{ років},$$

$$Y(10\%) = 3.43 \cdot 10^{6} \text{ років},$$

$$Y(15\%) = 2.34 \cdot 10^{6} \text{ років}.$$

Отримані значення стійкості до атаки методом «грубої сили» криптосистеми мовленнєвого зв'язку на основі факторіального кодування даних дозволяють визначити достатність (або недостатність) вжитих заходів забезпечення безпеки.

Слід зазначити, що властивість ФКВД забезпечувати сталість суми символів перестановки незалежно від типу інформації дає можливість відшукання меж перестановок в їх безперервному потоці. З точки зору забезпечення безпеки мовленнєвого обміну дана властивість створює вразливість криптосистеми мовленнєвого зв'язку за рахунок легкого встановлення циклового синхронізму приймача технічною розвідкою противника. Істотно ускладнити противнику рішення задачі перехоплення повідомлень можна за рахунок виконання перестановки біт в перестановці - носії вибірки мовленнєвого сигналу за таємним (від противника) правилом, що, по суті, означає доповнення системи факторіального кодування другим контуром шифрування.

4.6. Висновки

У четвертому розділі:

- вперше запропоновано методи декодування факторіальних кодів з відновленням даних з втратами, що дозволяє використовувати факторіальні коди для передачі мовленнєвої інформації в телекомунікаційних системах реального часу і забезпечує інтегрований захист;
- виконано аналіз основних показників якості відновленого мовленнєвого сигналу при відновленні вибірок в метриці Хеммінга та методом лінійної інтерполяції, а саме: шуму декодування, швидкості та стійкість коду;
- виконано порівняння шуму декодування запропонованих методів відновлення вибірок мовленнєвого сигналу в каналах з незалежними бітовими помилками та в каналах з пакетуванням помилок.

Розроблені методи декодування факторіальних кодів вирішують актуальна задачу забезпечення інтегрованого захисту мовленнєвих сигналів у реальному часі. Запропоновано два методи декодування факторіальних кодів з відновленням даних з втратами.

Ці методи передбачають перетворення, наприклад, за допомогою таблиці замін, фрагментів мовленнєвої інформації в перестановки, що передаються по каналу зв'язку. Процес декодування полягає в заміні перестановки на відповідний фрагмент мовленнєвого сигналу. У разі пошкодження отриманої перестановки, згідно з першим методом, вона відновлюється шляхом знаходження сигнального вектора, найбільш близького до перестановки в метриці Хеммінга. Якщо існує декілька таких векторів, то вибирається той, який є найбільш близьким в Евклідовій метриці до попереднього закодованого фрагменту. Спосіб відновлення перестановок методом лінійної інтерполяції включає відновлення мовленнєвого сигналу на основі відомих суміжних фрагментів.

Результати експериментів показують, що оптимальною сферою застосування алгоритму відновлення в метриці Хеммінга є канали зв'язку з

незалежними бітовими помилками з ймовірністю появи $p_0 \leq 10^{-2}$. У цих умовах декодер Хеммінга показує кращі результати, а шум декодування не перевищує значень рівня комфортного шуму. Оптимальною областю використання методу відновлення лінійною інтерполяцією є канали зв'язку з множинними бітовими помилками, спричиненими мультиплікативним шумом, зокрема, радіоканали короткометрового радіодіапазону або радіоканали будь-якого радіодіапазону з несприятливою шумовою ситуацією.

При використанні запропонованих алгоритмів в оптимальних умовах при ймовірності бітової помилки $p_0 \leq 10^{-2}$ забезпечується рівень шуму декодування, що не перевищує значень рівня комфортного шуму.

Однак, розглянуті в даній роботі методи факторіального кодування направлені на використання в існуючих телекомунікаційних системах, які передбачають використання додаткової службової інформації для керування системними ресурсами та забезпечення множинного доступу.

Тому актуальною задачею є розробка методу побудови телекомунікаційних систем множинного доступу з використанням нероздільного факторіального кодування, який не вимагає введення в канал додаткової службової інформації, а також забезпечить динамічне розподілення ресурсів мережі. Результати розробки такого методу будуть представлені в наступному розділі дисертації.

РОЗДІЛ 5. СИСТЕМИ МНОЖИННОГО ДОСТУПУ З ДИНАМІЧНИМ РОЗПОДІЛЕННЯМ СИСТЕМНИХ РЕСУРСІВ

5.1. Вступ

У попередніх розділах дисертації досліджено методи факторіального кодування з точки зору забезпечення інтегрованого захисту при використанні в існуючих телекомунікаційних системах. Розглянуто методи підвищення достовірності передачі даних ФК в системах с ВЗЗ, а також розроблено нові методи декодування факторіальних кодів з виявленням та виправленням помилок з втратами для кодування мовлиннєвих сигналів в системах реального часу. Для запропонованих методів виконано оцінку достовірності передачі даних, швидкості та стійкості до злому, а також надано рекомендації щодо їх застосування в залежності від якості каналу зв'язку. Зауважимо, що окрім безпосередніх даних, каналом зв'язку передається службова інформація, що в свою чергу зменшує загальну пропускну здатність системи.

Метою цього розділу є:

- розробка методу мультиплексування каналів, що не потребує окремого каналу синхронізації;
- розробка методу побудови системи з розподіленим доступом на основі факторіальних кодів, що дозволяє сформувати закриті групи користувачів у відкритих мережах колективного користування, а також здійснювати керування системними ресурсами.

Основні результати розробки опубліковано в [89], [90].

5.2. Розробка методу мультиплексування каналів на основі факторіальних кодів

Перш за все, відзначимо, що мультиплексори передачі даних (МПД) знайшли широке застосування при побудові телекомунікаційних систем для

організації групи низько швидкісних каналів в одному каналі колективного користування (в мультиплексному каналі) з більшою пропускною здатністю.

Відзначимо їх основні недоліки:

- для забезпечення синхронної роботи станції передачі і прийому МПД містять канал синхронізації;
- мультиплексор не забезпечує захист інформації.

З цього випливає наступне:

- наявність каналу синхронізації знижує пропускну здатність системи, призводить до втрати часу в сеансі зв'язку на виконання процедур пошуку синхронізму як на початку, так і в ході сеансу зв'язку;
- для забезпечення захисту інформації мультиплексор повинен мати додаткові засоби захисту, що помітно збільшує витрати на побудову і експлуатацію устаткування, знижує його надійність.

Запропонований в даному розділі клас мультиплексорів не потребує окремого каналу синхронізації за рахунок використання властивостей факторіальних кодів. Крім того, використання ФК дозволяє забезпечити інтегрований захист інформації, що підвищує достовірність передавання даних, спрощує елементний базис і, як наслідок, призводить до збільшення довжини ділянки регенерації та підвищує рентабельність такої системи зв’язку. Однак, розгляд всього комплексу пов’язаних з цим питань виходить за рамки даного дисертаційного дослідження.

5.2.1. Сутність методу

Метод мультиплексування каналів на основі факторіальних кодів полягає у використанні в якості носія інформації перестановок, наділених певним набором ознак, індивідуальних для кожного користувача з метою підвищення пропускної здатності системи з одночасним забезпеченням інтегрованого захисту інформації. За цими ознаками виконується поділ користувальницьких каналів між собою, що виключає необхідність організації окремого каналу

синхронізації або введення в блок даних поля адреси та підвищує пропускну здатність системи зв'язку.

Розглянемо організацію обміну і якісні показники каналу даних під час роботи двох користувачів виділеним каналом.

Нехай джерело інформації формує інформаційне слово з k біт. Кожному слову джерела поставлена у відповідність одна перестановка в вигляді послідовності символів – цілих чисел множини $\{0,1,\ldots,M-1\}$, де M – потужність множини символів перестановки. Розташування символів у перестановці визначається інформаційним словом джерела. Після виконання процедури відображення слова джерела у перестановку $A(x) \Rightarrow \pi(x)$, вона передається каналом зв'язку у вигляді двійкової послідовності з n біт. На станції прийому виконується зворотне перетворення – $\pi(x) \Rightarrow A(x)$, після інформація видається користувачеві. Практична реалізація такого відображення здійснюється за допомогою таблиць відображення (замін):

– $A(x) \Rightarrow \pi(x)$ на станції передавання;

– $\pi(x) \Rightarrow A(x)$ на станції прийому.

Кожна з цих таблиць містить 2^k рядків слів джерела і стільки ж відповідних їм перестановок.

До числа властивостей перестановок, наприклад, відносяться: число інверсій, місця розташування інверсій, представлення перестановок у вигляді набору точок відрізка $[0, M!-1]$ числовій осі і т.п. Кожна з цих ознак може бути використаною для організації багатостанційного доступу (БСД). Розглянемо організацію БСД з використанням ознаки належності числа інверсій перестановки певному класу лишків за деяким модулем q для вибору ознаки (ідентифікатора) умовного номеру часового каналу мультиплексора. Нагадаємо, що інверсією перестановки, згідно [55], називається ситуація, за якої наступний символ перестановки a_{i+1} за своїм значенням менше попереднього символу a_i, тобто $a_{i+1} < a_i$, де i – порядковий номер символу в перестановці. Так, половина перестановок (з множини потужністю $M!$) має

парне число інверсій, а інша половина – непарне число інверсій. Ця обставина дозволяє створити систему зв'язку двох користувачів одним каналом, у якій користувачам відводяться перестановки, що відрізняються ознакою парності інверсій. Це означає, що перший користувач буде передавати тільки парні перестановки, а другий – тільки непарні. З цього також випливає, що в розпорядження кожного з користувачів відводиться $0{,}5M!$ перестановок, внаслідок чого множина використовуваних слів алфавіту джерела так само зменшується в два рази. Зменшення потужності множини слів у два рази відповідає зменшенню числа біт в інформаційному слові на одиницю – до значення $k-1$. Якщо потрібно створити систему зв'язку з великим числом користувачів, то потрібно вибрати іншу ознаку закріплення перестановок за каналами. Для цього врахуємо, що число інверсій $\omega = inv(\pi)$ у перестановці на множині символів потужністю M задовольняє умові $0 \le \omega \le 0{,}5 \cdot M \cdot (M-1)$. Звідси випливає, що множина з $M!$ перестановок може бути розділена на класи $B_M(q,R) = \left\{\pi : |inv(\pi)|_q = R\right\}$ з числом інверсій, що належать класу лишків $\overline{R}_q$, де q – модуль класу.

Відповідно до [55], потужності $W_M(q,R)$ класів $B_M(q,R)$ у залежності від значень q і R обчислюються наступним чином:

$$W_M(q,R) = \sum_{j=0}^{\left[(0{,}5 \cdot M \cdot (M-1)-R)/q\right]} N_M(\omega = jq + R), \qquad (5.1)$$

де $N_M(\omega)$ – частотне число (кількість) перестановок порядку М з числом інверсій $\omega = inv(\pi)$ (числа МакМахона). В онлайн-енциклопедії цілочисельних послідовностей (OEIS) [91] представлена послідовність чисел МакМахона $N_M(\omega)$ для $M \in [1,50]$.

У роботі [92] показано, що для $q \le M$ потужності класів $B_M(q,R)$ інваріантні відносно R та є рівними

$$W_M(q,R) = M!/q. \qquad (5.2)$$

Звідси випливає, що існує можливість створення системи зв'язку з множинним доступом для угруповання від 2 до $0{,}5M\left(M-1\right)$ користувачів, а максимальне число біт, що переносяться перестановками заданого класу, визначається як $k_{\max}\left(q,R\right)=\left\lfloor \log_2 W_M\left(q,R\right)\right\rfloor$.

Це означає, що інші $\Delta W_M\left(q,R\right)=W_M\left(q,R\right)-2^{k_{\max}(q,R)}$ перестановок класу $B_M\left(q,R\right)$ не будуть використані для перенесення інформації цього користувача, але можуть бути використані для створення бібліотеки таблиць замін $A(x)\Rightarrow\pi(x)$ і $\pi(x)\Rightarrow \mathrm{A}(x)$. Таблиці кожного користувача повинні відрізнятися складом перестановок і порядком їх розташування в таблиці.

Якщо цю бібліотеку тримати в секреті, то кожна з цих таблиць може служити сеансовим ключем шифрування інформації.

Показники достовірності в результаті використання в якості носія інформації перестановок класу $B_M\left(q,R\right)$ оцінено в роботі [55].

5.2.2. Структурна схема канального мультиплексора на основі факторіальних кодів

Реалізувати передавальну частину канального мультиплексора на основі факторіальних кодів (МПДФК) можна за допомогою пристрою, спрощена структурна схема якого показана на рис. 5.1.

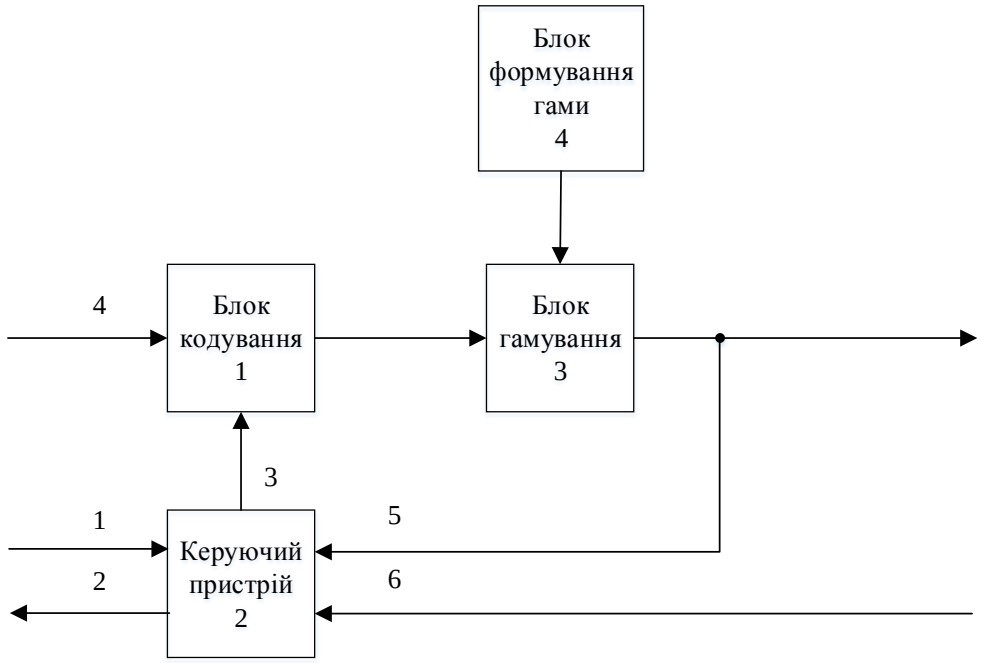


Рис 5.1. Структурна схема передавальної частини канального мультиплексора на основі факторіальних кодів

Принцип роботи передавальної частини системи полягає в наступному. Джерело інформації, у якого з'явилася потреба передати дані іншому абоненту цієї ж мережі, формує сигнал «запит передавання» своєму передавачу ланцюгом 1. Керуючий пристрій передавача 2 підключається до мультиплексного каналу ланцюгом 5 і «слухає» його (визначає стан мультиплексного каналу: вільний або зайнятий). Інтервал часу спостереження за станом каналу перевищує мультиплексний цикл (цикл опитування всіх каналів мультиплексора). Якщо мультиплексний канал «вільний», керуючий пристрій передавача 2 формує сигнал «готовність» джерелу інформації ланцюгом 2, у відповідь на який джерело надсилає передавачу інформаційне слово $A(x)$ ланцюгом 4 на вхід блоку кодування 1. Блок кодування 1 за таблицею замін $A(x) \Rightarrow \pi(x)$ обирає перестановку $\pi(x)$ і видає її в канал зв'язку. У випадку виявлення помилки під час передавання кодового слова зворотним каналом зв'язку ланцюгом 6 на керуючий пристрій передавача 2 надходить сигнал на повторне передавання спотвореного блоку. У цьому разі керуючий пристрій 2 ланцюгом 3 формує команду для блоку кодування 1 (без

формування сигналу «готовності» ланцюгом 2 для джерела інформації), в результаті чого блок кодування 1 повторно видає попередню перестановку.

Перед видачею перестановки в канал зв'язку вона піддається гамуванню в блоці гамування 3, де на сформовану перестановку $\pi(x)$ накладається гама з блоку формування гами 4 за допомогою операції додавання за модулем 2. Сформована бітова послідовність з блоку гамування 3 видається в мультиплексний канал.

Реалізувати приймальну частину канального мультиплексора на основі факторіальних кодів можна за допомогою пристрою, спрощена структурна схема якого показана на рис. 5.2.

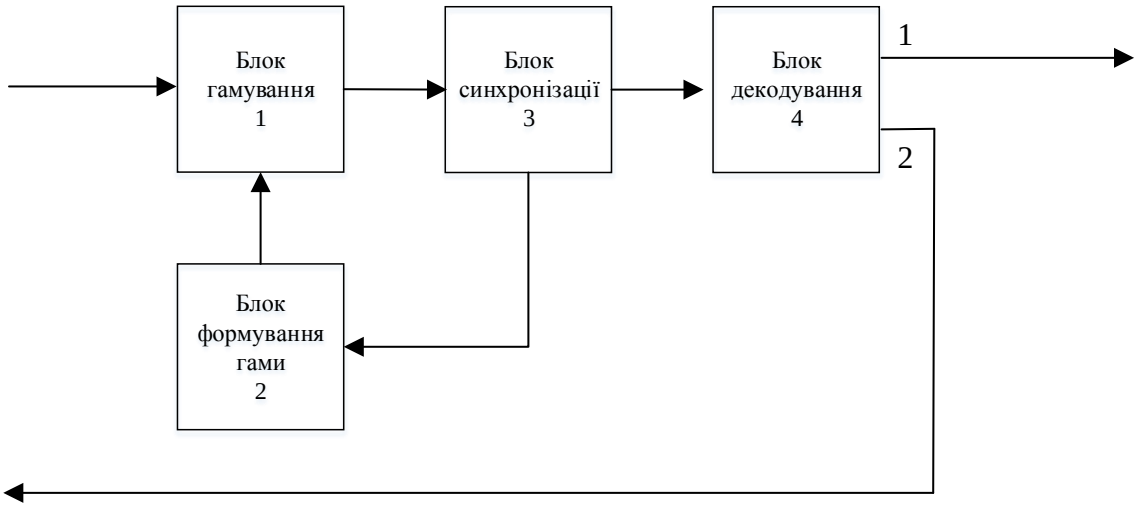


Рис 5.2. Структурна схема приймальної частини канального мультиплексора на основі факторіальних кодів

На вхід приймача з мультиплексного каналу надходить бітова послідовність, яка потрапляє в блок гамування 1, де відбувається зняття гами, сформованої блоком формування гами 2. Отримане на виході блоку гамування 1 кодове слово потрапляє на вхід блоку синхронізації 3. У блоці синхронізації 3 накопичується фрагмент послідовності з $\psi = zn$ біт, де інтервал накопичення z встановлюється під час розгортання системи. У плаваючому вікні з n-бітного фрагмента ψ підраховується сума чисел – символів перестановки. Якщо сума чисел у вікні для більшості з z відрізків довжини n не дорівнює

$S = 0{,}5M\left(M-1\right)$, то синхронізм не знайдений, вікно спостереження зміщується в право на один крок, підрахунок суми повторюється до тих пір, поки сума більшості відрізків не стане рівною $S = 0{,}5M\left(M-1\right)$. Це свідчить про те, стан синхронізму генераторів гами приймальної і передавальної станції знайдено.

Після виконання процедури синхронізації інформація з блоку синхронізації 3 передається до блоку декодування перестановки 4, де відбувається зворотне відображення перестановки в інформаційне слово $\pi\left(x\right) \Rightarrow A\left(x\right)$. У разі виявлення помилки, блок декодування перестановки 4 ланцюгом 2 формує і відправляє через зворотний канал запит на повторення блоку. У іншому випадку декодоване інформаційне слово $A\left(x\right)$ видається отримувачу через ланцюг 1.

5.2.3. Режими роботи мультиплексора на основі факторіальних кодів

Розроблений канальний мультиплексор забезпечує можливість адаптації дисципліни обслуговування до навантаження на мережу. Для цього, за максимального навантаження реалізується режим циклічного доступу, коли мультиплексний канал надається користувачам в порядку черги 1, 2, 3, ..., *N*, де числа 1, 2, 3, ..., *N* – це умовні номери абонентів в циклі мультиплексора.

Розглянемо інші можливі режими роботи.

Покладемо, що в початковому стані, коли навантаження на мережу дорівнює нулю (наприклад, пізно вночі), а канал вільний, джерело інформації, у якого з’явилася потреба передати дані, захоплює канал і починає трансляцію свого повідомлення. Цією дією джерело інформації (яке першим сформувало запит) присвоїло собі номер 1 у циклі мультиплексора. Після завершення передавання першого слова джерела керуючий пристрій контролює канал, визначаючи його стан протягом декількох циклів (число контрольованих циклів задається під час налаштування системи). Часова діаграма роботи мультиплексора на цьому інтервалі показана на рис. 5.3, *а*. Якщо на цьому інтервалі спостереження з’явилося наступне джерело, що має потребу

передавання інформації, керуючий пристрій передавача другого джерела, «слухаючи» канал, визначає, що канал зайнятий і шукає в циклі суміжну вільну часову позицію. Визначивши, що зайнята одна, а всі інші часові позиції циклу вільні, займає другу часову позицію циклу і починає спостереження за вільними часовими каналами. У цей же час керуючий пристрій першого каналу виявляє факт появи другого абонента і продовжує спостереження за станом каналу (зайнятий/вільний) протягом усього сеансу зв'язку. Часова діаграма роботи мультиплексора для цієї ситуації показана на рис. 5.3, *б*. Так буде тривати доти, поки не відбудеться одна з наступних подій:

– з'явиться третій користувач;
– інші часові канали залишаться вільними.

У разі підключення третього користувача він виконує процедури, які виконував другий користувач.

Якщо після закінчення інтервалу спостереження користувачем не виявлено факту заняття наступного часового каналу, то він своїми даними може зайняти всі (крім одного) вільні часові канали. Один часовий канал залишається вільним і служить резервом для підключення ще одного користувача.

У цьому випадку можливі дві ситуації:

– більше ніхто не займе вільний канал до моменту завершення сеансу зв'язку одним з абонентів мережі;
– вільний канал буде зайнятий ще одним користувачем.

Обидві ситуації призводять до необхідності розгляду процедур звільнення каналів, обумовленим завершенням сеансу зв'язку одним з користувачів. Швидкість передавання слів кожного з джерел буде дорівнювати $B_{word} = B_{\pi} \cdot N$ де B_{π} – швидкість передавання перестановок-носіїв інформації в мультиплексному каналі, N – число каналів мультиплексора. Зауважимо, що цей режим циклічного доступу – режим обслуговування максимального навантаження на мережу – дозволяє передавати сукупний об'єм інформації, що чисельно дорівнює пропускній здатності мультиплексного каналу. Часова діаграма роботи мультиплексора для цієї ситуації показана на рис. 5.3, *в*.

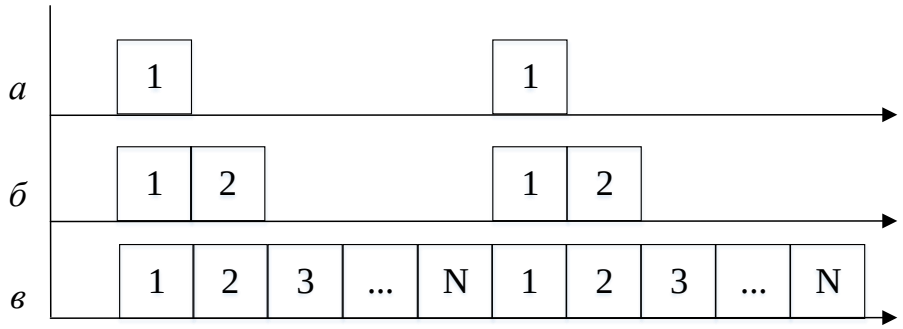


Рис 5.3. Часові діаграми режимів роботи канального мультиплексора на основі факторіальних кодів

Тепер розглянемо режим зниження навантаження на мережу після завершення сеансу зв’язку будь-яким абонентом мережі.

Зауважимо, що оскільки керуючі вузли всіх каналів контролюють завантаження каналу (перевіряють, чи є вільні канали), то факт звільнення будь-якого каналу фіксується кожним з цих пристроїв. Якщо звільняється один часовий канал, то його ніхто не займає, оскільки він є резервом на випадок включення в роботу абонента цієї ж мережі. Коли звільниться ще один канал, його займає пристрій, який захопив попередній часовий канал. Якщо ж цей пристрій не потребує додаткового каналу та не зайняв його на протязі одного мультиплексного циклу, то на початку наступного мультиплексного циклу попередній пристрій може захопити цей канал. Процедура буде повторюватися доти, поки якийсь з пристроїв не захопить канал або всі активні пристрої не скористаються ним. У останньому випадку вільний часовий канал надалі зможе захопити будь-який з пристроїв. В цілому, викладений алгоритм роботи забезпечує максимально можливе використання ресурсів мережі.

5.3. Розробка методу побудови системи з розподіленим доступом на основі факторіальних кодів

5.3.1. Опис методу

Розглянутий принцип організації каналу колективного користування, в якому працює замкнуте угруповання користувачів, полягає у використанні ФК,

а саме перестановки як контейнера для для перенесення інформації. Вибір для організації контейнера ФСЧ обумовлений тим, що вона забезпечує швидке зростання числа контейнерів як функції від потужності множини символів перестановки М.

Цей метод також забезпечує порівняно високу достовірність передавання – він виявляє всі помилки непарної кратності і частину помилок парної кратності, які призводять до трансформації перестановки в неперестановку або в перестановку надлишкової частини множини. Виправлення цих помилок проводиться через повторний запит. Код не виявляє помилки, що трансформують перестановку в іншу перестановку дозволеної частини цієї ж множини. Ще однією перевагою цього коду є те, що цей код має властивість самосинхронізації, тобто код "без коми". Це обумовлено тим, що сума чисел у перестановці не залежить від структури вектора $A(x)$ і дорівнює $S_\pi = \frac{M \cdot (M-1)}{2}$. Це означає, що якщо різні перестановки слідують одна за одною, то межа перестановок (контейнерів) легко визначається за сумою М символів в плаваючому (по часовій осі) вікні. Якщо сума дорівнює S_π, то межі вікна, як правило, збігаються з границями контейнера. Винятком є ті випадки, коли сума рівна S_π виходить при підсумовуванні М символів різних (суміжних) перестановок, особливо в умовах дії завад. Ці ефекти придушуються інтеграцією.

В таких системах для організації всередині каналу колективного користування закритих груп користувачів для передачі даних використовуються множини перестановок з унікальним набором властивостей та параметрів, що забезпечує конфіденційність обміну в межах групи. При цьому система повинна забезпечувати:

- керування темпом вводу/виводу інформації зі сторони каналу передачі даних;
- єдину процедуру моніторингу трафіку і визначення на цій основі порядку захоплення/звільнення часових каналів.

Також слід зазначити, що при розробці системи слід враховувати характер даних, якими будуть обмінюватися користувачі всередині групи, оскільки це може накладати додаткові вимоги до системи. З огляду на це, розглянемо принципи побудови систем з розподіленим доступом, що орієнтовані на передачу голосу та передачу бінарних даних.

5.3.2. Система з розподіленим доступом, орієнтована на передачу голосу

В системі з розподіленим доступом, що орієнтована на передачу голосу, кожний контейнер переносить одну чи декілька вибірок голосового сигналу. Більшість часу кожний окремо взятий абонент перебуває в режимі мовчання и лише прослуховує канал. В цей момент генератори всіх абонентських терміналів взаємно асинхронні. Тому, перш ніж розпочинати передачу корисної інформації, необхідно виконати циклову і тактову синхронізацію.

В стандартних протоколах ТС [93]-[95] для встановлення тактової синхронізації передбачено використання преамбули, що являє собою фіксовану послідовність біт. У запропонованій системі передбачено використання преамбули, довжина якої повинна перевищувати час входження в синхронізм по тактам. У якості такої преамбули пропонується використовувати перестановки із забороненої підмножини. При виборі перестановок слід віддавати перевагу таким, двійковий запис яких налічує максимальну кількість перемикань (перехід із 0 в 1 або із 1 в 0), що мінімізує час встановлення тактової синхронізації. Зауважимо, що преамбулу слід передавати тільки в тому випадку, коли час фази радіомовчання перевищує час утримання тактового синхронізму.

Після встановлення тактової синхронізації відбувається пошук маркера для встановлення циклової синхронізації. Ця операція виконується шляхом підрахунку суми символів перестановки у плаваючому вікні з N біт. Це обумовлюється тим, що незалежно від інформації, що передається перестановкою, сума всіх символів перестановки завжди рівна $\sigma = 0{,}5M\left(M-1\right)$.

Це властивість забезпечує можливість циклової синхронізації на протязі всього сеансу зв’язку.

5.3.3. Система з розподіленим доступом, орієнтована на передачу бінарних даних

Розглянемо задачу забезпечення множинного доступу (для N абонентів) до одного або декількох центрів управління. Такі завдання виникають, наприклад, при забезпеченні безпеки обміну інформацією на ділянці «крайовий банківський термінал - банк».

У великих магазинах, на вокзалах, в готелях або торгово-розважальних центрах практикується установка групи банківських терміналів (БТ), що належать різним банкам.

Зауважимо, що первинна мережа, в якій відбувається перенесення інформації, може мати будь-яку фізичну природу: відкритий радіопростір, пара проводів, мережа електроживлення і т.п. У даній роботі для пояснення принципу побудови такого виду систем МД будемо використовувати моноканал на основі витої пари.

Спрощена структурна схема такої системи наведена на рис. 5.4.

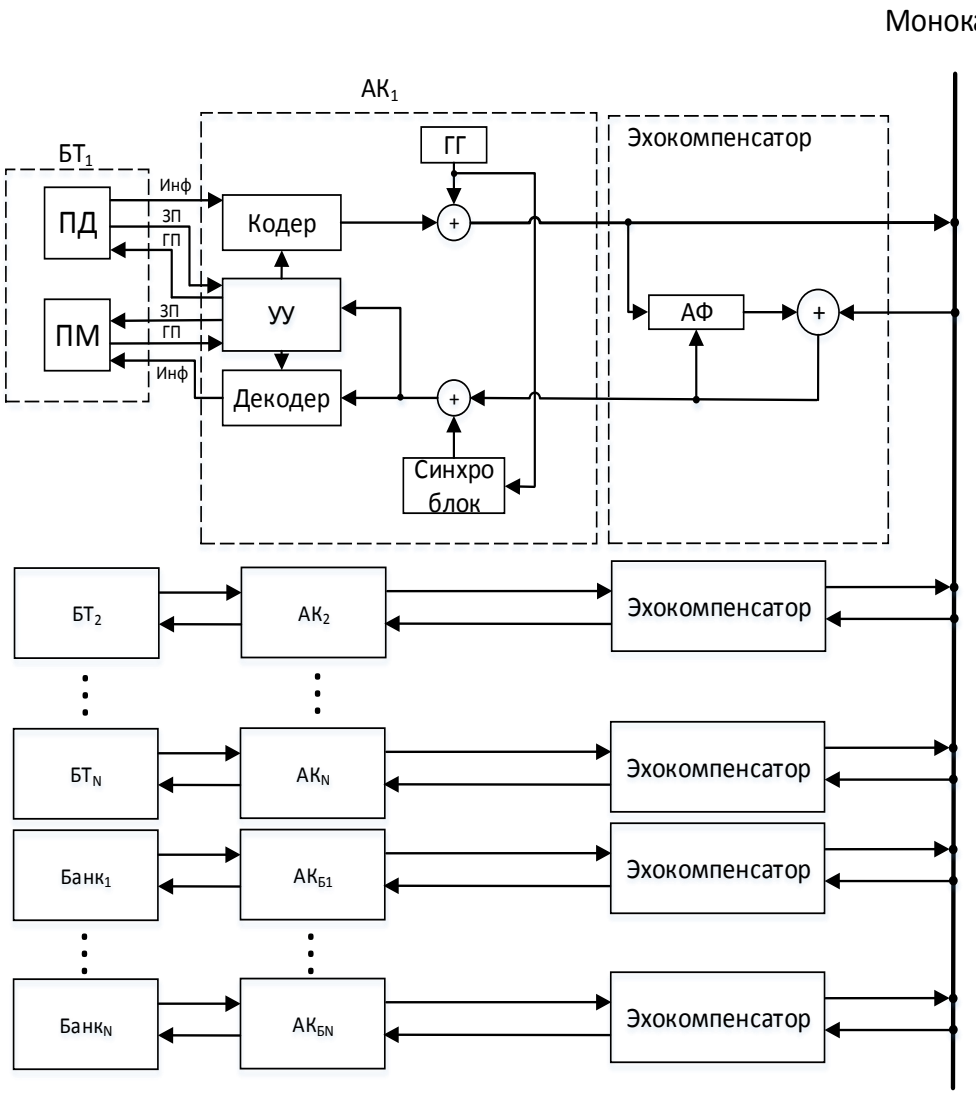


Рис 5.4. Структурна схема СМД

Основними її особливостями є:

– нестаціонарний потік заявок на обслуговування, інтенсивність якого може змінюватися в широких межах;

– необхідність інтегрованого захисту інформації за межами охоронної зони об'єкта;

– кожен БТ (група БТ) може мати власний ключ шифрування (наприклад, таблицю замін);

– всі часові канали, що надаються користувачам, мають рівну пропускну здатність, що забезпечує перенесення перестановок з n біт;

– поділ напрямку передачів двухпровідному каналі колективного користування здійснюється за рахунок ехокомпенсації.

Оскільки інформаційні потоки нестаціонарні, необхідно передбачити можливість перерозподілу ресурсу пропускної здатності каналу колективного користування на користь завантажених терміналів. Перерозподіл ресурсів пропускної здатності зводиться до забезпечення можливості заняття не тільки свого тимчасового каналу, а й вільного від навантаження суміжного каналу. Це забезпечує зменшення довжини черги на обслуговування в завантажених каналах.

В зв'язку з цим, ТС МД передбачає сполучення БТ з моноканалом через абонентский комплект (АК), який включає в себе МПДФК, що дозволяє вирішити наступні задачі:

– забезпечення сполучення з БТ і моноканалом;

– забезпечення перетворення $A(x) \Rightarrow \pi(x)$ змінними ключами (наприклад, таблицями замін);

– забезпечення гамування послідовності перестановок і зняття цієї гами;

– забезпечення безперервного моніторингу трафіку абонентів, для розподілу вільних часових каналів між завантаженими БТ;

– забезпечення зміни темпу видачі інформації БТ від нульового значення (режим накопичення даних) до максимального значення, рівного пропускної здатності моноканала (режим монополізації каналу);

– при максимальному навантаженні на БТ (відповідно, і на систему МД) кожному терміналу повинен бути наданий один часовий канал.

5.3.4. Оцінка основних параметрів тракту

Згідно[45], під основними параметрами тракту мається на увазі:

- ймовірність невиявленої помилки P_{ud};
- еквівалентна ймовірність помилки p_{eq};
- залишкова ймовірність помилкового прийому P_{res};
- відносна швидкість передачі ν_0.

Імовірність невиявленої помилки показує ймовірність того, що інформації вектор даних, який надійшов на вхід приймача, було вражено помилкою, і який

буде оброблений і виданий споживачеві як такий, що не містить помилок. Цей параметр будемо визначати експериментальним шляхом.

Еквівалентна ймовірність помилки, згідно [45], визначена як ймовірність помилки в гіпотетичному симетричному постійному двійковому каналі, при якій ймовірність безпомилкового прийому така ж, як і в системі, що моделюється, і розраховується як $p_{eq} = P_{ud} / \left(k \left(Q + P_{ud} \right) \right)$, де P_{ud} – ймовірність невиявленої помилки, Q – ймовірність правильного прийому блоку даних.

Залишкова ймовірність помилкового прийому показує ймовірність того, що прийнятий вектор даних буде містити хоча б одну помилку – $P_{res} = P_{ud} / \left(Q + P_{ud} \right)$.

Згідно [45], $\nu_0 = \nu_1 \cdot \nu_2$, де $\nu_1 = k / n$ – статична складова втрати швидкості або швидкість коду, ν_2 – динамічна складова втрати швидкості внаслідок повторних запитів, $\nu_2 = \left(1 - p_0 \right)^n$.

Також, за відомими значеннями числа біт k, що переноситься однією перестановкою, потужності множини символів перестановки M, числа біт n при кодуванні перестановки двійковим кодом, і швидкості коду v можна обчислити пропускну здатність кожного з часових каналів системи.

Приймемо, що:

– інформаційна частина блоку даних містить по одному біту від кожного з джерел, тоді $k = N$, а M приймає мінімальне значення, при якому $M! \geq 2^k$;
– кожному джерелу надається часовий канал з наперед заданою швидкістю.

Звідси випливає, що реальний масштаб часу обміну інформацією на ділянці «вхід МПДФК - вихід МПДФК» забезпечується за умови, якщо час $\tau_0 k$ прийому k біт (від $N = k$ джерел) не менше часу $\tau_1 n$ виведення n біт в моноканал на приймальній станції, тобто при виконанні наступної рівності: $\tau_0 k \geq \tau_1 n$. При цьому $\tau_0 = 1 / \left(N B_{N\inf} \right)$, $B_{N\inf}$ – швидкість передачі даних кожним

джерелом інформації, $\tau_1 = 1/B_M$, B_M – швидкість передачі даних в мультиплексному каналі.

Звідси випливає, що швидкість передачі даних, відведена в моноканалі кожному з N джерел,

$$B_{NM} = \frac{1}{k\tau_1} \geq \frac{B_{N\inf}}{\nu_1}, \tag{5.3}$$

а швидкість передачі даних в моноканалі (пропускна здатність моноканала)

$$B_M \geq N \frac{B_{N\inf}}{\nu_1}. \tag{5.4}$$

Виконаємо порівняльну оцінку пропускної здатності моноканала МПДФК$\left(B_M\right)$ і пропускної здатності моноканала МПД типового рішення $\left(B_{M_0}\right)$ [93] при однаковій кількості інформаційних каналів і швидкості передачі в них. Типовий МПД крім N інформаційних каналів містить канал синхронізації з пропускною спроможністю $B_{N\inf}$. Тоді $B_{M_0} \geq \frac{N+1}{N\tau_0} = (N+1)B_{N\inf}$, звідки $\frac{B_{M_0}}{B_M} = \frac{(N+1)\nu_1}{N}$.

Звідси випливає, що МПДФК вимагає більшої пропускної здатності в порівнянні з типовим МПД. При цьому величина зменшення необхідної пропускної здатності через виключення каналу синхронізації порівняно невелика. Цей факт можна розцінювати як обмін пропускної здатності на додаткові корисні властивості у вигляді забезпечення конфіденційності і достовірності.

Для оцінки параметрів тракту розроблена програмна модель «джерело інформації - канал зв'язку - приймач інформації» СПД з МД. Модель передбачає використання ФКВД з наступними параметрами: $M = 8$, $k = 15$, $n = 24$, $\nu_1 = 0.625$. Оцінювати параметри тракту будемо в діапазоні $p_0 \in 0.49 \ldots 10^{-12}$. Це дозволяє отримати оцінку для різних середовищ передачі

інформації (від відкритих радіоканалів до оптоволокна) і для різної якості каналу.

Модель містить:

– генератор слів джерела $A(x)$;

– два генератора джерела помилок (з рівномірним і біноміальним розподілом) $\varepsilon(x)$;

– факторіальний кодер - таблиця замін інформаційного вектора $A(x)$ на перестановку π (представлену в двійковій системі числення вектором $R(x)$);

– імітатор каналу зв'язку, що забезпечує виконання операції $R(x)\oplus\varepsilon(x)$;

– факторіальний декодер - таблиця замін перестановки π (представлену в двійковій системі числення вектором $R(x)$) на інформаційний вектор $A(x)$;

– аналізатор прийнятого блоку даних $R'(x)=R(x)\oplus\varepsilon(x)$.

Аналізатор прийнятого блоку даних, тобто аналізатор вектора $R'(x)$ містить:

– блоку оцінки коректності вектора $R'(x)$;

– блоку оцінки приналежності прийнятої перестановки π' дозволеній множині перестановок;

– блок виявлення помилок декодування.

Алгоритм роботи моделі полягає в наступному. Генератор слів джерела формує фіксований вектор для рівномірного розподілу помилок і випадковий вектор для біноміального розподілу помилок. Кодер витягує з таблиці заміни перестановку π $\left(R(x)\right)$, що відповідає вихідному слову $A(x)$, і виводить його в канал. Канал зв'язку виконує операцію $R(x)\oplus\varepsilon(x)$. В результаті вектор $R'(x)=R(x)+\varepsilon(x)$ надходить на вхід приймача (відповідно, на вхід блоку оцінки коректності). У блоці оцінки правильності перевіряється кількість входжень для кожного із символів перестановки $\{0,1,\ldots,M-1\}$. Якщо отриманий вектор містить кожен із символів перестановки $\{0,1,\ldots,M-1\}$ рівно один раз, то

отриманий блок є перестановкою. В іншому випадку отриманий вектор не є перестановкою і стирається, а сигнал запиту повторної передачі блоку надсилається на станцію передачі через зворотний канал. Якщо отриманий вектор є перестановкою, він передається до блоку оцінки приналежності прийнятої перестановки π' дозволеній множині перестановок. У цьому блоці перевіряється, чи вона входить до таблиці замін. Якщо таку перестановку знайдено в таблиці замін, то з неї витягується відповідний інформаційний вектор $A'(x)=A(x)\oplus\varepsilon(x)$ і видається одержувачу. При цьому, $\varepsilon(x)=0$ і $A'(x)=A(x)$ лише коли $\varepsilon(x)=0$. Ця ситуація відповідає правильному прийому блоку даних.

Якщо $\varepsilon(x)\neq 0$, то одержувач приймає вектор $A'(x)=A(x)\oplus\varepsilon(x)$. Ця ситуація є помилкою декодера. Таким чином, неправильне декодування відбувається лише у випадку, коли початкова перестановка перетворюється в іншу перестановку, що належить до таблиці замін.

Результати експериментальної оцінки основних параметрів тракту наведені на рис. 5.5 – 5.8.

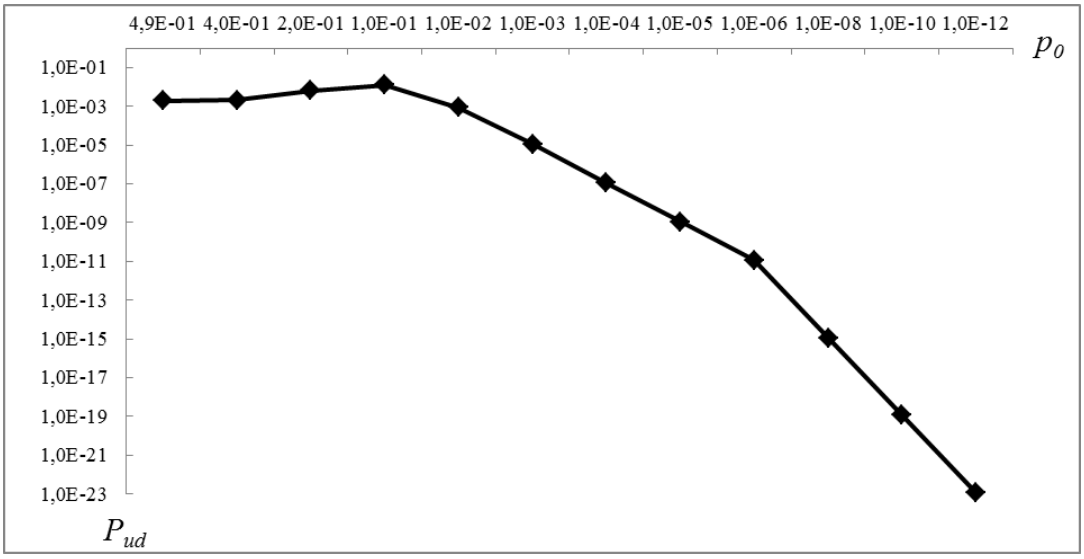


Рис 5.5. Діаграма залежності ймовірності невиявленої помилки від ймовірності бітової помилки

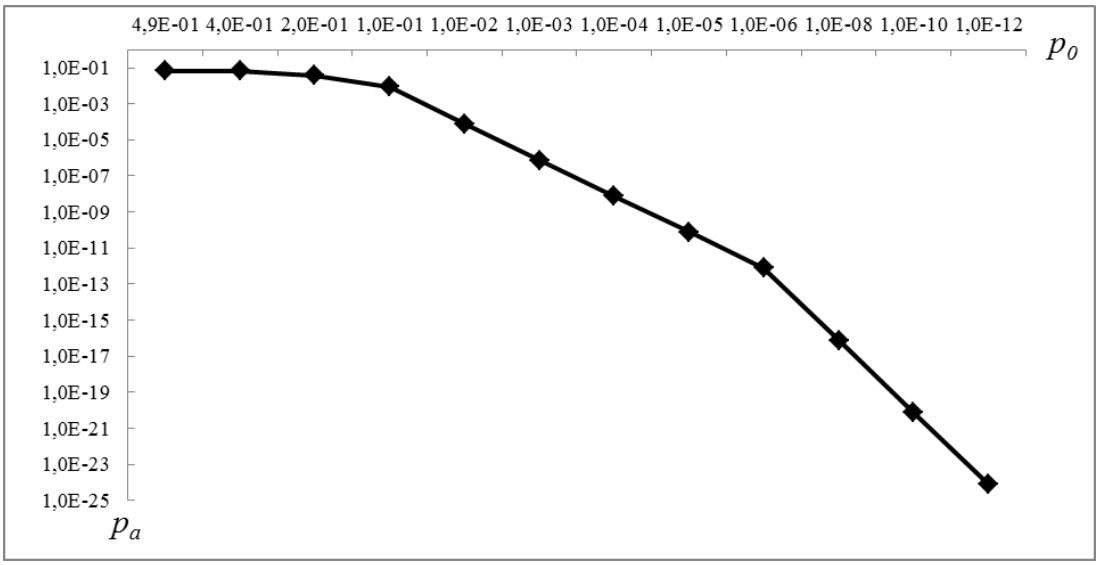


Рис 5.6. Діаграма залежності еквівалентної ймовірності помилки від ймовірності бітової помилки

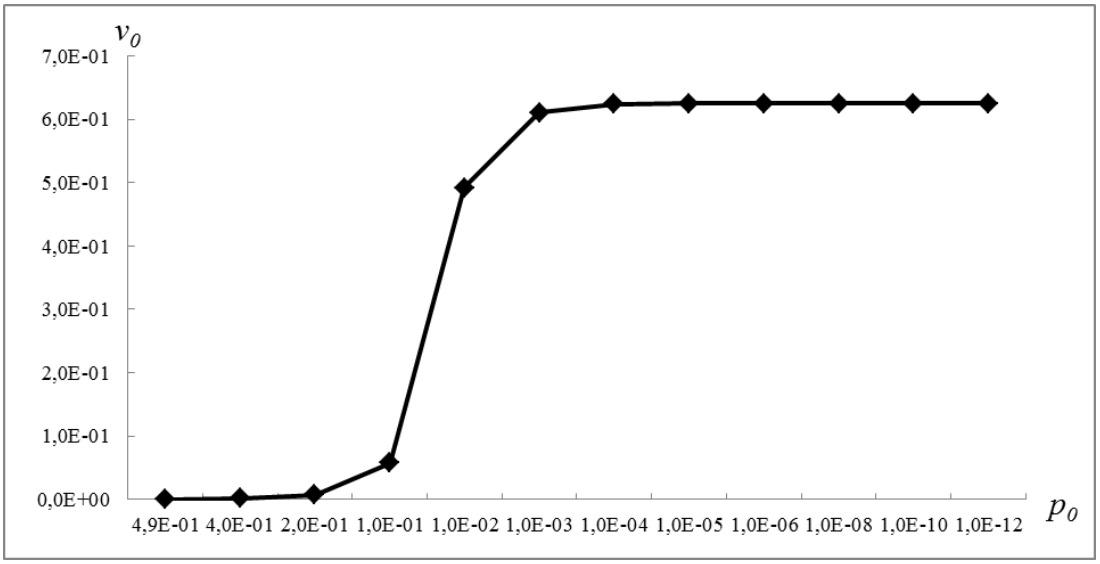


Рис 5.7. Діаграма залежності відносної швидкості передачі від ймовірності бітової помилки

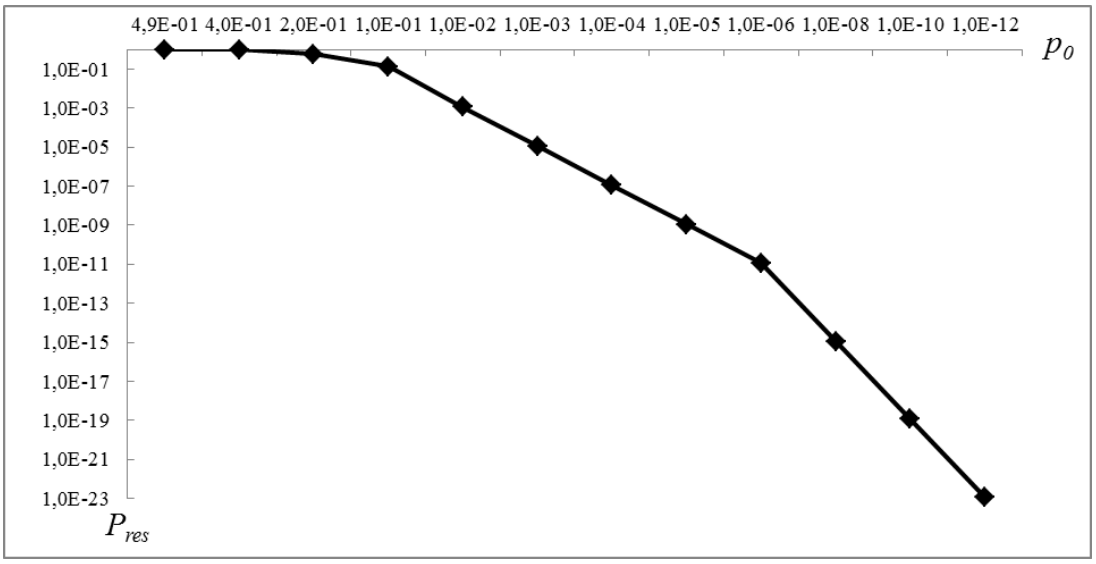


Рис 5.8. Діаграма залежності залишкової ймовірності помилкового прийому від ймовірності бітової помилки

Перш за все, виділимо на осі абсцис графіків наступні 3 ділянки:

- при $p_0 \le 0{,}1$ – ділянка деградації каналу;
- при $0{,}1 < p_0 \le 10^{-3}$ – ділянка стабілізації каналу;
- при $p_0 < 10^{-3}$ – ділянка стабільності.

На ділянці деградації каналу, яка характеризує поведінку системи МД, наприклад, при роботі у відкритому радіоканалі в умовах інтенсивної радіоелектронної протидії, при збільшенні інтенсивності перешкод різко падає відносна швидкість передачі, але настільки ж різко зростає ймовірність виявлення помилки. За своєю ефективністю, роботу системи МД із застосуванням ФК на цій ділянці можна порівняти із застосуванням шумоподібного сигналу як засобу протидії навмисним радіоперешкодам. Однак показники шумоподібного сигналу це постійні величини, що визначаються його базою, в той час як у системі з ФК – основні показники системи адаптивні до якості каналу.

Аналіз ділянки стабілізації показників системи є найважливішим етапом проектування системи, при виборі довжини регенераційної ділянки. Розташування регенераторів по магістралі визначається вибором компромісу між пропускною здатністю та достовірністю передачі.

На ділянці стабільної роботи системи варіація параметрів не призводять до скільки-небудь помітної зміни основних параметрів. Це є ознакою неоптимального вибору порядку перестановки. Очевидно, що для роботи по каналах дуже хорошої якості (лінії мікрохвильового зв'язку, оптоволокно і т.п.) необхідно вибирати значення $M > 8$.

5.3.5. Опис алгоритму керування системними ресурсами

Для вирішення завдання розподілу ресурсу пропускної здатності каналу колективного користування визначимо, яким чином забезпечити управління темпом обміну інформацією в системі. Для цього розглянемо рис. 5.9.

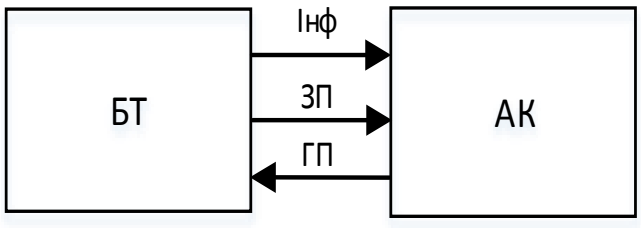


Рис 5.9. Управління темпом передачі даних

Управління темпом видачі інформації між джерелом інформації БТ і АК здійснюється за допомогою двох ланцюгів управління:

- запит передачі (ЗП), який формує джерело;
- готовність до передачі (ГП), який формує АК.

У початковому стані (в стані «спокою»), коли відсутнє навантаження на БТ, ланцюги ЗП і ДП знаходяться в стані «вимкнено».

При появі навантаження на БТ, ланцюг ЗП переводиться у стан «включено», що є ознакою наявності інформації для передачі. Ця інформація накопичується в буферній пам'яті БТ до тих пір, поки ланцюг ГП не перейде в стан «включено».

При включенні ланцюга ГП джерело виводить в АК перше слово, після чого АК вимикає ланцюг ГП, потім перетворює слово джерела в перестановку і видає її в канал.

У свою чергу, джерело, завершивши передачу першого слова АК, вимикає ланцюг ЗП і утримує його у вимкненому стані до тих пір, поки в інформаційному буфері не накопичено наступне слово, яке підлягає передачі.

Включення ланцюга ГП проводиться при виконанні двох умов:

– ланцюг ЗП знаходиться в стані «включено»;

– АК завершив передачу інформації в попередніх мультиплексних циклах.

Аналогічно виконується управління темпом видачі інформації з АК в БТ.

Таким чином, наведений алгоритм обробки запитів на передачу інформації забезпечує регулювання темпу передачі інформації і його адаптацію до величини пропускної здатності каналу колективного користування, відведеної даному джерелу.

Також визначимо поняття інтенсивності навантаження, як кількісної міри обробленого навантаження кожним часовим каналом мультиплексного циклу і мережею в цілому, ресурс (пропускної здатності мережі) і принцип розподілу цього ресурсу.

Під заявкою на обслуговування будемо розуміти перехід ланцюга ЗП зі стану «виключено», в стан «включено», а під значенням інтенсивності потоку заявок (інтенсивності навантаження) λ будемо розуміти число заявок, що надходять в одиницю часу.

Оскільки кожне слово від БТ, що надходить в АК, передається в канал у вигляді перестановки з n біт, максимальне навантаження (слів / с), яке обслуговується одним часовим каналом, визначиться наступним чином:

$$\lambda_{\max} = \frac{B_{NM}}{n}, \tag{5.5}$$

де B_{NM} визначається рівністю (5.3).

Звідси максимальне навантаження (слів / с) на канал загального користування:

$$\Lambda_{\max} = N\lambda_{\max} = N\frac{B_{NM}}{n} = \frac{B_M}{n}. \tag{5.6}$$

При такому значенні максимального навантаження мережа працює в режимі мультиплексора. Якщо фактичне навантаження $\Lambda < \Lambda_{max}$ (тобто один або кілька БТ формують, в середньому, менше, ніж одну заявку в мультиплексному циклі), то з'являється невикористаний ресурс, який може бути переданий завантаженим БТ.

Є очевидним, що оптимальним правилом розподілу ресурсів пропускної здатності моноканала є принцип «кому більше потрібно». Оскільки розглянута система з очікуванням, то кожен з БТ має буфер, що накопичує заявки на обслуговування. Звідси випливає, що ресурс пропускної здатності більше потрібен тим БТ, у яких більше чергу на обслуговування заявок. Процедура розподілу ресурсу пропускної здатності при зміні навантаження на мережу, реалізована за рахунок використання МПДФК в складі АК і передбачає наступні дії:

– якщо j-ий часової канал мультиплексного циклу не займається j-им користувачем деякий, наперед визначений час, то він може бути зайнятий $(j-1)$-им користувачем за умови, що він вимагає додаткової пропускної здатності каналу. Операція по захопленню суміжного каналу може повторюватися до тих пір, поки не будуть зайняті $(N-1)$ часових каналів мультиплексного циклу. Один канал залишається вільним для підключення нових користувачів;

– якщо користувач, часовий канал якого зайнятий, відновив активність (включив ланцюг ЗП), то він у вільному каналі передає свій номер, що ініціює процедуру звільнення відведеного для нього часового каналу.

5.4. Висновки

У пятому розділі:

– вперше розроблено та досліджено метод побудови телекомунікаційних систем множинного доступу з використанням нероздільного факторіального кодування. Цей підхід включає виділення кожному з

користувачів підмножини перестановок з певним набором властивостей. Використання факторіальних кодів забезпечує інтегрований захист інформації, а також синхронізацію кадрів без роздільника блоку даних. Крім того, це підвищує надійність інформації під час її передачі в ТС в умовах обмеження пропускної здатності каналів зв'язку;

– вперше розроблено алгоритми організації закритих груп абонентів у мультиплексному каналі та динамічного розподілу ресурсу пропускної здатності каналу залежно від значення навантаження, що створюється кожним з користувачів системи.

Результати, викладені в цій роботі, можуть сприяти подальшому розвитку телекомунікаційних систем з множинним доступом та впровадженню більш ефективних методів забезпечення інтегрованої інформаційної безпеки.

ВИСНОВКИ

У дисертаційній роботі вирішено актуальну науково-технічну задачу, що полягає в підвищенні достовірності передавання інформації, забезпеченні її конфіденційності та цілісності на основі використання факторіального кодування даних. Ця задача передбачає необхідність удосконалення існуючих методів факторіального кодування, створення методу побудови телекомунікаційних систем на основі нероздільного факторіального кодування з динамічним розподілом ресурсів між користувачами системи в режимі реального часу.

Найбільш значущі результати роботи полягають у наступному:

1. Розроблено методи підвищення достовірності передавання даних у системах з факторіальним кодування з відновленням даних за перестановкою. Показано, що метод факторіального кодування з додатковими перевірними бітами, за рахунок контролю ознаки парності числа інверсій, дає змогу зменшити ймовірність невиявленої кодом помилки, при незалежних бітових помилках, по відношенню до ФКВДд на 2-4 порядки для $M = 8$, в залежності від ймовірності бітової помилки. Разом з тим, метод факторіального каскадного кодування, за рахунок комбінування рівномірного коду та ФКВД, дав змогу на порядок зменшити ймовірність невиявленої кодом помилки, при незалежних бітових помилках, по відношенню до ФКВД для $k = 15$, в залежності від ймовірності бітової помилки, при цьому показано, що за довжини блоку даних $k \geq 128$, швидкість коду не змінюється;
2. Розроблено метод формування сигнально-кодової конструкції для систем *з* факторіальним кодування з відновленням даних за перестановкою, який за рахунок забезпечення мінімальної відстані між сигнальними векторами решітки дозволяє максимізувати швидкість коду для заданого рівня достовірності передавання даних. Побудовано решітки, що дозволяють виявляти помилки кратності $t \leq 5$ і

виправляти помилки кратності $t \leq 2$ з максимальною швидкістю коду для $M = 5, k = 3, N_{sv} = 10$ та $M = 6, k = 5, N_{sv} = 60$;

3. Отримав подальший розвиток метод факторіального кодування з відновленням даних за перестановкою, який за рахунок відновлення даних з втратами дозволяє забезпечити захист мовної інформації від помилок каналу зв’язку та несанкціонованого доступу в телекомунікаційних системах реального часу. Розроблено метод відновлення перестановок в метриці Хеммінга, що орієнтований на канали з незалежними бітовими помилками та метод відновлення перестановок шляхом лінійної інтерполяції – орієнтований на канали зв'язку з множинними бітовими помилками, спричиненими мультиплікативним шумом;
4. Розроблено метод побудови систем множинного доступу на основі нероздільного факторіального кодування, який за рахунок використання факторіального коду з заданим числом інверсій та виділення кожному з користувачів підмножини перестановок з певним набором властивостей забезпечує захист інформації від помилок, що діють у каналі зв'язку, та спроб несанкціонованого доступу до інформації, а також циклову синхронізацію без застосування окремого каналу синхронізації. Також розроблено та описано алгоритм динамічного розподілу ресурсу пропускної здатності каналу, залежно від значення навантаження, що створюється кожним з користувачів системи.

СПИСОК ДЖЕРЕЛ

[1] Е. В. Фауре, «Методологія захисту інформації на основі факторіального кодування даних», дис. д-ра техн. наук, 05.13.21, К., 2017.

[2] R. J. McEliece, «A Public-Key Cryptosystem Based on Algebraic Theory», *The Deep Space Network Progress Report*, pp. 42–44, 1978.

[3] Ф.Дж. Мак-Вильямс и Н.Дж.А. Слоэн, *Теория кодов, исправляющих ошибки*. М.: Связь, 1974.

[4] А. А. Борисенко, *Биномиальные автоматы*. Сумы.:СУМГУ, 2005.

[5] А. А. Борисенко и А. Е. Горячев, «Обнаружение ошибок на основе перестановок», *Известия Юго-Западного государственного университета. Серия Управление, вычислительная техника, информатика. Медицинское приборостроение*, №3, с. 14-19, 2013.

[6] А. А. Борисенко и А. Е. Горячев, «Помехоустойчивая передача экономической информации на основе перестановок», *Актуальні проблеми економіки*, №3, с. 156-163, 2013.

[7] А. А. Борисенко и А. Е. Горячев, «Исправление ошибок в перестановках», *Системи обробки інформації*, №2, с. 171-173, 2013.

[8] А. А. Борисенко, А. Е. Горячев, Б. К. Лопатченко и А. Н. Кобяков, «Перестановки в телекоммуникационных сетях», *Вісник Сумського державного університету*, № 2, с. 15-22, 2013.

[9] И. Д. Горбенко, Ю. В. Стасев, А. В. Ивашкин и А. М. Ткачёв, «Анализ имитостойкости систем спутниковой связи и управления», *Радиотехника. Харьковский государственный технологический университет радиоэлектроники*, № 112, с. 17-21, 1999.

[10] И. Д. Горбенко, Ю. В. Стасев, А. В. Потий и А. М. Ткачев, «Предложения по обеспечению безопасности информации в единой спутниковой системе передачи информации», *Космічна наука і технологія*, том 6, № 5, с. 62-66, 1998.

[11] В. И. Грабчак, «Исследование достоверности передачи данных в АСУВ с использованием каскадных теоретико-кодовых схем», *Системи обробки інформації*, № 9, с.13-16, 2006.

[12] В. І. Грабчак, «Криптоаналіз каскадних теоретико-кодових схем захисту інформації», *Вісник Сумського державного університету. Серія: Технічні науки*, № 3, с. 88-95, 2007.

[13] В. И. Грабчак, И. В. Пасько, Р. В. Королев и И. Е. Кужель, «Алгебраическое кодирование алгебро-геометрическими кодами на пространственных кривих», *Системи обробки інформації*, № 8, с. 134-138, 2007.

[14] А. А. Кузнецов, «Методика оценки эффективности помехоустойчивого кодирования в каналах с группирующимися ошибками», *Электронное моделирование*, № 3, с. 49-60, 2006.

[15] А. А. Кузнецов, «Методика оценки энергетической эффективности двоичных блоковых кодов в каналах с группирующимися ошибками», *Моделювання та інформаційні технології*, № 32, с. 116-124, 2005.

[16] А. А. Кузнецов, «Энергетический выигрыш алгеброгеометрического кодирования», *Радиотехника: Всеукраинский межведомственный научно-технический сборник*, № 134, с. 218-222, 2003.

[17] Е. Л. Онанченко, А. А. Кузнецов, В. Н. Лысенко, В. И. Грабчак и Р. В. Королев, «Исследование методов защиты информации, основанных на использовании алгебраических блоковых кодов», *Системи обробки інформації*, № 7, с. 53-58, 2007.

[18] Е. Л. Онанченко и А. В. Лысенко, «Анализ известных методов декодирования недвоичных блоковых кодов», *Вісник Сумського державного університету. Серія: Технічні науки*, № 3, с. 100-105, 2008.

[19] С. А. Осмоловский, *Стохастические методы передачи данных*. М.: Радио и связь, 1991.

[20] С. А. Осмоловский, *Стохастические методы защиты информации*. М.: Радио и связь, 2003.

[21] С. А. Осмоловский, *Стохастическая информатика: инновации в информационных системах*. М.: Горячая линия-Телеком, 2012.

[22] A. P. Stakhov , V. Massingue and A. Sluchenkova, *Introduction into Fibonacci Coding and Cryptography*. Kharkov: Osnova, 1999.

[23] А. П. Стахов, «Компьютеры Фибоначчи и новая теория кодирования: история, теория, перспективы», *Известия Южного федерального университета. Технические науки*, т. 38, №3, с. 205-213, 2004.

[24] А. П. Стахов, «Золотые матрицы и новый метод криптографии», в *Современные методы кодирования в электронных системах: тезисы докладов третей международной научной конференции, Сумы, 24-25 октября 2006*, Сумы, 2012, с. 41-42.

[25] A. P. Stakhov, «Fibonacci Matrices, a Generalization of the ‘Cassini Formula’, and a New Coding Theory», *Chaos, Solitons & Fractals*, vol. 30, issue 1, pp.56-66, 2006.

[26] A. P. Stakhov, «The ‘Golden’ Matrices and a New Kind of Cryptography», *Chaos, Solitons & Fractals*, vol. 32, issue 3, pp.1138-1146, 2007.

[27] М. И. Мазурков, В. Я. Чечельницький и П. Мурр, «Метод защиты информации на основе совершенных двоичных решеток», *Известия вузов. Радиоэлектроника*, т. 51, №11, с. 53-57, 2008.

[28] М. И. Мазурков, В. Я. Чечельницький, П. Е. Баранов, А. Н. Мелешкевич, С. Н. Кропачев и Н. И. Кушниренко, «Методы повышения защиты информации путем объединения операций уплотнения, шифрования и канального кодирования», *Известия вузов. Радиоэлектроника*, №5, т. 54, с. 3-16, 2011.

[29] В. Я. Чечельницький, «Методологія підвищення ефективності телекомунікаційних систем на основі інтеграції канального кодування та шифрування даних», дис. д-ра техн. наук, 05.12.02, К., 2013.

[30] В. Я. Чечельницкий и Н. И. Кушниренко, «Метод криптографической передачи информации на базе эквивалентного класса совершенных

двоичных решеток», *Інформатика та математичні методи в моделюванні*, т. 4, №3, с. 210-218, 2014.

[31] D. W. Hagelbarger, «Recurrent codes: Easily Mechanized, Burst-Correcting, Binary Codes», *Bell System Tech. J.*, № 38, pp. 969-984, 1959.

[32] D. W. Hagelbarger, *Error Detection Using Recurrent Codes*. AIEE Winter General Meeting, I960.

[33] W. L. Kilmer, «Some results on linear-recurrent binary burst-correcting codes», Montana State College Electronics Research Laboratory, Project Xs 6795-102 Technical Report Xs 3, 1959.

[34] C. Berrou, A. Glavieux and P. Thitimajshima, «Near Shannon limit error-correcting coding and decoding: Turbo-codes», *IEEE International Conference on Communications,* vol. 2, pp. 1064-1070, 1993.

[35] C. Berrou and A. Glavieux, «Near optimum error correcting coding and decoding: Turbo codes», *IEEE Transactions on Communications*, vol. 44, pp. 1261-1271, *1996*.

[36] A. Stefanov and T. M. Duman, «Turbo coded modulation for wireless communications with antenna diversity», *Proceedings of the IEEE Vehicular Technology Conference-Fall*, vol. 3, pp. 1565-1569, 1999.

[37] A. Stefanov and T. M. Duman, «Turbo coded modulation for systems with transmit and receive antenna diversity», *Proceedings of the IEEE Global Telecommunications Conference*, vol. 5, pp. 2336-2340, 1999.

[38] L. Bahl, J. Cocke, F. Jelinek and J. Raviv, «Optimal decoding of linear codes for minimizing symbol error rate», *IEEE Transactions on Information Theory*, vol. 20, issue 2, pp. 284-287, 1974. DOI: 10.1109/TIT.1974.1055186.

[39] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard and B. Ra-mamurthi, «Packet Reservation Multiple Access for Local Wireless Communications», *IEEE Transactions on Communications*, vol. 37, issue 8, pp. 885-890, 1989. DOI: 10.1109/26.31190.

[40] W. C. Wong and D. J. Goodman, «Integrated Data and Speech Transmission Using Packet Reservation Multiple Access» in *IEEE International Conference on Communications ICC*, 1993. Pp. 172-176.

[41] E. N. Gilbert, «Capacity of a Burst-Noise Channel», *Bell System Technical Journal*, vol. 39, pp. 1253–1265, 1960.

[42] E. O. Elliott, «Estimates of Error Rates for Codes on Burst-Noise Channels», *Bell System Technical Journal*, vol. 42, pp. 1977–1997, 1963.

[43] Ф. М. Возенкрафт и В. Рейффен, *Последовательное декодирование*. Москва, 1963.

[44] Л. М. Финк, *Теория передачи дискретных сообщений*. М.: Советское радио, 1970.

[45] Л. М. Фінк, *Теорія передачі дискретних повідомлень*, 2-е видання перероб. і доп. М .: Радянське радіо, 1970.

[46] Э. В. Фауре, В. В. Швыдкий и А. И. Щерба, «Контроль целостности информации на основе факториальной системы счисления», *Journal of Baku engineering university – Mathematics and computer science*, т. 1, № 1, с. 3-13, 2017.

[47] Е. В. Фауре та М. О. Качалова, «Дослідження процедури формування контрольної суми повного факторіального коду на основі ітераційного перетворення», в *Проблеми інформатизаці: Тези доповідей П'ятої Міжнародної науково-технічної конференції, Черкаси, 13-15 листопада 2017 р.*, Черкаси: ЧДТУ, Баку: ВА ЗС АР, Бельсько-Бяла: УтіГН, Полтава: ПНТУ, 2017, с. 17.

[48] Э. В. Фауре, В. В. Швыдкий и А. И. Щерба, «Комбинированное факториальное кодирование и его свойства», *Радіоелектроніка, інформатика, управління*, № 3, с. 80-86, 2016.

[49] Э. В. Фауре, «Факториальное кодирование с несколькими контрольными суммами», *Вісник Житомирського державного технологічного університету. Серія: Технічні науки*, т. 78, № 3, с. 104–113, 2016.

[50] Е. В. Фауре та А. Ю. Бойко, «Дослідження здатності виявлення помилок факторіальним кодом з декількома контрольними сумами», в *Проблеми інформатизаці: Тези доповідей П'ятої Міжнародної науково-технічної конференції, Черкаси, 13-15 листопада 2017 р.*, Черкаси: ЧДТУ, Баку: ВА ЗС АР, Бельсько-Бяла: УтіГН, Полтава: ПНТУ, 2017, с. 16.

[51] Э. В. Фауре, «Факториальное кодирование с восстановлением даннях», *Вісник Черкаського державного технологічного університету*, № 2, с. 33-39, 2016.

[52] Е. В. Фауре, О. О. Харін, В. В. Швидкий та А. І. Щерба, «Спосіб факторіального кодування з відновленням даних», Україна. Пат. 117004, 12.06.2017.

[53] Е. В. Фауре та О. О. Харін, «Дослідження ймовірності виникнення помилки декодування під час використання факторіального коду з відновленням даних», в *Актуальні задачі та досягнення у галузі кібербезпеки: Тези доповідей Всеукраїнської науково-практичної конференції, Кропивницький, 23-25 листопада 2016 р.*, Кропивницький, 2016, с. 178-179.

[54] Э. В. Фауре, «Метод повышения эффективности факториального кодирования с восстановлением даннях», *Вісник Черкаського державного технологічного університету*, № 4, с. 57-61, 2016.

[55] E. V. Faure, A. I. Shcherba and A. A. Kharin, «Factorial Code with a Given Number of Inversions», *Radio Electronics, Computer Science, Control*, vol. 2, p. 143-153, 2018.

[56] Е. В. Фауре, О. О. Харін, В. В. Швидкий та А. І. Щерба, «Спосіб факторіального кодування з виявленням і виправленням помилок», Україна. Пат. 121361, 11.12.2017.

[57] Э. В. Фауре, «Факториальное кодирование с исправлением ошибок. Теоретическое обоснование и примеры реализации», в *Наукоемкие технологии в инфокоммуникациях: обработка информации,*

кибербезопасность, информационная борьба: монографія, В. М. Безрука, В. В. Баранника, Х.: Издательство «Лидер», 2017, с. 291-323.

[58] Э. В. Фауре, «Факториальное кодирование с исправлением ошибок», *Радіоелектроніка, інформатика, управління*, № 3, с. 130-138, 2017.

[59] Е. В. Фауре та О. О. Харін, «Факторіальне кодування з відновленням даних і виправленням помилок», в *Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку: Тези доповідей Всеукраїнської науково-практичної Internet-конференції, Черкаси, 13-19 березня 2017 р.*, Черкаси: ЧДТУ, 2017, с. 74–76.

[60] B. Elspas, «Note of P-nary Adjacent-Error-Correcting codes», *IRE Transactions on Information Theory*, vol. 6, issue 1, pp. 13-15, 1960.

[61] S. Le Goff, A. Glavieux and C. Berrou, «Turbo-Codes and High Spectral Efficiency Modulation», in *Proc. of IEEE ICC*, 1994, pp. 645-649.

[62] S. Le Goff, «Performance of bit-interleaved turbo-coded modulations on Rayleigh fading channels», *Electronics Letters*, vol. 36, №. 8, pp. 731-733, 2000.

[63] D. P. Panayiotis, A. S. Thomas, V. Prabodh and V. Hannu, «Turbo Coded Modulation over GSM Channels», in *International Conference on Third Generation Wireless and Beyond*, 2001.

[64] P. D. Papadimitriou and P. Varshney, «Turbo Coded Modulation for High Throughput TDMA Systems», in *Proc. of IEEE VTC*, 2001, pp. 1390-1394.

[65] A. A. Florea, H. Nguyen, L. Martinod and C. Molko, «Serial Progressive Hierarchical Turbo Codes», in *Global Telecommunications Conference (GLOBECOM), 5-9 December 2011*, Houston, Texas, 2011, pp. 1-5.

[66] D. Jiang, H. Wang, E. Malkamaki and E. Tuomaala, «Principle and performance of semi-persistent scheduling for VoIP in LTE system» in *International Conference on Wireless Communications, Networking and Mobile Computing, 21-25 Sept. 2007*, Shanghai, China, 2007, pp. 2861 - 2864.

[67] D. Astely, E. Dahlman, A. Furuskar, Y. Jading, M. Lindstrom and S. Parkvall, «LTE: the evolution of mobile broadband», *IEEE Communications Magazine*, vol. 47, issue 4, pp. 44-51, 2009.

[68] M. C. Nguyen, H. Nguyen, P. Mege and L. Martinot, «Architecture for Multi-users Multiplexing Radio Voice Transmission for Enhancing Voice Capacity over LTE in PMR Context», in *IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*,Washington, DC, USA, 2-5 Sept. 2014, pp. 1459 - 1463.

[69] M. C. Nguyen, H. Nguyen, E. Georgeaux, P. Mege and L. Martinot, «Scheduling for multi-users multiplexing radio voice transmission for enhancing voice capacity over LTE in PMR context», in *Wireless Communications and Networking Conference*, 2015, pp. 1267-1272. DOI: 10.1109/WCNC.2015.7127651.

[70] P. Koutsakis and M. Paterakis, «Highly Efficient Voice-Data Integration over Medium and High Capacity Wireless TDMA Channels», *Wireless Networks*, vol. 7, pp. 43-54, 2001. DOI: 10.1023/A:1009043914931.

[71] K. C. Chua and W. M. Tan, «Modified Packet Reservation Multiple Access Protocol», *IEEE Electronics Letters*, vol. 29, issue 8, pp. 682-684, 1993.

[72] G. Bianchi, F. Borgonovo, L. Fratta, L. Musumeci and M. Zorzi, «C-PRMA: a centralized packet reservation multiple access for local wireless communications», *IEEE Transactions on Vehicular Technology*, vol. 46, issue 2, pp. 422-436, 1997. DOI: 10.1109/25.580781.

[73]A. Al-Shanyour, A. Mahmoud, T. Sheltami and S.A. AlQahtani, «Packet Reservation Multiple Access (PRMA) with Random Contention», in *IEEE/ACS International Conference on Computer Systems and Applications*, 2007, pp. 158-164. DOI: 10.1109/AICCSA.2007.370878.

[74] Д. Э. Кнут, *Искусство программирования. Том 2: Получисленные алгоритмы*. М.: Вильямс, 2007.

[75] О. О. Харін, «Порівняльна оцінка факторіальних кодів», *Вісник Черкаського державного технологічного університету. Серія: технічні науки*, № 4, с. 88-93, 2017.

[76] Е. В. Фауре, О. О. Харін, В. В. Швидкий та А. О. Лавданський, «Ефективність виявлення помилок факторіальними кодами», в *Інформаційні технології в освіті, науці і техніці (ІТОНТ-2020): Тези доповідей V Міжнародної науково-практичної конференції, Черкаси, 21-23 травня 2020 р.*, Черкаси: ЧДТУ, 2020, с. 94-95.

[77] Л. Ф. Бородин, *Введение в теорию помехоустойчивого кодирования*. М.: Советское радио, 1968.

[78] О. О. Харін, «Оцінка властивостей каскадного коду, що поєднує факторіальний та рівноважний код», *Вісник Черкаського державного технологічного університету*, № 2, с. 86-90, 2017.

[79] О. О.Харін та А. І. Щерба, «Спосіб факторіального кодування в метриці Хеммінга», Україна. Пат. 130458, 10.12.2018.

[80] О. О. Харін, «Формування сигнально-кодової конструкції на основі теорії решіток», в *Наука України – погляд молодих вчених крізь призму сучасності: Тези доповідей ІІ Всеукраїнської науково-практичної конференції з міжнародною участю, Черкаси, 26 вересня 2019 р.*, Черкаси: ЧДТУ, 2019, с.42-44.

[81] А. О. Лавданський, Е. В. Фауре, О. О. Харін та В. В. Швидкий, «Спосіб декодування факторіального коду з відновленням вибірок мовного сигналу реального часу в метриці Хеммінга», Україна. Пат. 137722, 11.11.2019.

[82] А. О. Лавданський, Е. В. Фауре, О. О. Харін та В. В. Швидкий, «Спосіб декодування факторіального коду з відновленням вибірок мовного сигналу реального часу методом лінійної інтерполяції», Україна. Пат. 139760, 27.01.2020.

[83] E. V. Faure, V. V. Shvydkyi, A. O. Lavdanskyi and O. O. Kharin, «Methods of factorial coding of speech signals», *Radio Electronics, Computer Science, Control*, vol. 4, p. 186-198, 2019.

[84] О. О. Харін, Е. В. Фауре та А. О. Лавданський, «Оцінка захищеності мовного сигналу в системах з факторіальним кодуванням», в *Інформаційні технології в освіті, науці і техніці (ІТОНТ-2020): Тези доповідей V Міжнародної науково-практичної конференції, Черкаси, 21-23 травня 2020 р.*, Черкаси: ЧДТУ, 2020, с. 85-87.

[85] М. В. Назаров та Ю. Н. Прохоров, *Методи цифрової обробки і передачі мовних сигналів*. М .: Радио и связь, 1985.

[86] Н. В. Захарченко, С. М. Горохов и А. В. Кочетков, *Информационные параметры позиционных и таймерных кодов. Том 1. Информационные параметры позиционных кодов: учеб. пособ*. Одесса: ОНАС им. А.С. Попова, 2018.

[87] Н. В. Захарченко, Е. Н. Мартынова, Ю. С. Горохов, Д. Н. Бектурсунов та А.С. Криль, «Статистические параметры искажений таймерных сигнальных конструкций в каналах модели Гильберта», *Наукові праці ОНАЗ ім. О.С. Попова*, № 1, с. 42-47, 2015.

[88] А. А. Лавданский, «Оценка времени формирования последовательности чисел с учетом повышения производительности вычислительных блоков», in *International Scientific Conference on The scientific potential of the present: proceedings*, St Andrews, Scotland, United Kingdom, 2016, pp. 123-126.

[89] J. Al-Azzeh, B. Ayyoub, E. Faure, V. Shvydkyi, O. Kharin and A. Lavdanskyi, «Telecommunication Systems with Multiple Access Based on Data Factorial Coding», *International Journal on Communications Antenna and Propagation (IRECAP)*, vol. 10, issue 2, p. 102-113, 2020.

[90] А. А. Харин и А. И. Щерба, «Организация замкнутой группировки абонентов в открытой сети коллективного пользования», в *Інформаційні технології в освіті, науці і техніці (ІТОНТ-2018): Тези*

доповідей IV Міжнародної науково-практичної конференції, Черкаси, 17-18 травня 2018 р., Черкаси: ЧДТУ, 2018, с. 109-111.

[91] Oeis.org (9 July 2018). *A008302 - OEIS*. [online] Available: http://oeis.org/A008302.

[92] Д. Э. Кнут, *Искусство программирования. Том 3: Сортировка и поиск*, 2-е изд. М.: ООО «И.Д. Вильямс», 2007. ISBN: 978-5-8459-0082-1.

[93] G. Miao, J. Zander, K.W. Sung and B. Slimane, *Fundamentals of Mobile Data Networks*. Cambridge University Press, 2016. ISBN 1107143217.

[94] R. Steele and L. Hanzo, *Mobile Radio Communications: Second and Third Generation Cellular and WATM Systems*, 2nd edition. IEEE Press, 1999. ISBN 0-471-97806-x.

[95] E. Dahlman, S. Parkvall, J. Skold and P. Beming, *3G Evolution: HSPA and LTE for Mobile Broadband*, 2nd edition. Academic Press, 2008. ISBN: 9780080923192.

ДОДАТКИ

Додаток А. Лістинги розрахунково-експериментальних моделей

А.1. Бібліотека для роботи в ФСЧ

```
package com.company;

import java.math.BigInteger;



public class Factorial_Lib {
   /*!
    * \brief Факториал большого числа
    * \param[in] val - число, факториал которого нужно расчитать
    * \return Длинное число, факториал

**********************************************************************/
   public static BigInteger BigIntFact(int val) {
      BigInteger result = BigInteger.ONE;

      if(val == 0)
         return  BigInteger.ONE;

      for(int i = 1; i <= val; i++)
      {
         result = result.multiply(BigInteger.valueOf(i));
      }

      return result;
   }

   /*!
    * \brief Преобразование числа в синдром
    * \param[in] val - число, по которому нужно расчитать синдром
    * \param[out] Sf  - указатель на структуру синдрома
    * \param[in] size - мощность перестановки

**********************************************************************/
   public static void NumberToSyndrom(BigInteger val, int[] Sf, int size) {
      BigInteger k;
      BigInteger fact;
      BigInteger value = val;
      int        n     = (size - 1);
```

```
    //Обнуление синдрома
    for(int i = 0; i < size; i++)
        Sf[i] = 0;

    while(value.compareTo(BigInteger.ZERO) != 0)
    {
        //Расчет значения факториала
        fact = BigIntFact(n);
        //Определяем коефициент для текущего значения
        k = value.divide(fact);
        //Запись синдрома
        Sf[n] = k.intValue();
        //Вычетание n! из значения
        value = value.remainder(fact);
        n--;
    }
}

/*!
 * \brief Преобразование синдрома в перестановку
 * \param[in] Sf - указатель на структуру синдрома
 * \param[out] P  - указатель на структуру перестановки
 * \param[in] size - мощность перестановки

**********************************************************************/
public static void SyndromToTransp(int[] Sf, int[] P, int size) {
    int tmp;
    int pos = 0;

    for(int i = 0; i < size; i++)
        P[i] = i;
    //Перебор всех элементов (0..M-1)
    for(int i = size - 1; i >= 0; i--)
    {
        //Синдром для текущего элемента
        int cnt = Sf[i];
        //Поиск позиции элемента в структуре перестановки
        for(int j = size - 1; (j >= 0) && (cnt > 0); j--)
        {
            if(P[j] == i)
            {
                pos = j;
                break;
            }
        }
```

```
        //Выполнение перестановки по синдрому текущего элемента
         //(cnt – количество элементов, меньших за i, что находятся правее)
        while(cnt > 0)
        {
           for(int j = pos - 1; j >= 0; j--)
           {
              if(P[j] < P[pos])
              {
                 tmp    = P[pos];
                 P[pos] = P[j];
                 P[j]   = tmp;
                 pos    = j;
                 break;
              }
           }
           cnt--;
        }
     }
  }

  /*!
   * \brief Преобразование перестановки в синдром
   * \param[out] Sf - указатель на структуру синдрома
   * \param[in] P  - указатель на структуру перестановки
   * \param[in] size - мощность перестановки

**********************************************************************/
   public static void TranspToSyndrom(int[] Sf, int[] P, int size) {
     //Перебор по всем элементам перестановки
     for(int i = 0; i < size; i++)
     {
        int val = P[i];
        int cnt = 0;

        //Подсчет элементов, меньших за val, что находятся правее
        for(int j = i + 1; j < size; j++)
        {
           if(P[j] < val)
              cnt++;
        }
        //Запись коефициента для текущего элемента
        Sf[val] = cnt;
     }
  }
```

```
    /*!
     * \brief Преобразование синдрома в число
     * \param[out] Sf - указатель на структуру синдрома
     * \param[in] size - мощность перестановки
     * \return Полученное из синдрома число

**********************************************************************/
    public static BigInteger SyndromToNumber(int[] Sf, int size) {
       BigInteger fact;
       BigInteger k;
       BigInteger res;

       //Инициализация числа для хранения результата
       res = BigInteger.ZERO;
       for(int i = 1; i < size; i++)
       {
          if(Sf[i] > 0)
          {
             //Инициализация числа для хранения коефициента числа i
             k = BigInteger.valueOf(Sf[i]);
             fact = BigIntFact(i);
             fact = fact.multiply(k);
             //res = res + i! * k
             res  = res.add(fact);
          }
       }

       return res;
    }

    /*!
    * \brief Получение индекса перестановки
    * \param[in] P  - указатель на структуру перестановки
    * \param[in] size - мощность перестановки
    * \return Индекс перестановки

**********************************************************************/
    public static int GetTranspositionIndex(int[] P, int size) {
       int res = 0;
       //Перебор по всем элементам перестановки
       for(int i = 0; i < (size - 1); i++)
       {
          if(P[i] > P[i + 1])
             res += (i + 1);
       }
```

```
        return  res;
    }

    /*!
    * \brief Получение количества инверсий перестановки
    * \param[in] P  - указатель на структуру перестановки
    * \param[in] size - мощность перестановки
    * \return Количество инверсий

**********************************************************************/
    public static int GetTranspositionInversCnt(int[] P, int size) {
        int res   = 0;
        int Sf[]  = new int[size];

        TranspToSyndrom(Sf, P, size);
        //Перебор по всем элементам перестановки
        for(int i = 0; i < size; i++)
            res += Sf[i];
        return  res;
    }
}
```

А.2. Лістинг розрахунково-експериментальної моделі для накопичення статистики невиявлених помилок ФКДБ при рівномірному росподілі

```
package com.company;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.math.BigInteger;
import java.util.Formatter;
import java.util.Scanner;

public class Main {
   public static void main(String[] args) throws IOException {

      long[]    wt;            // Таблица веса ошибки Wt(E)
      int       k;             // Длина информационного блока
      int       M = 0;         // Мощность перестановки
      int       rc;            // Количество бит для кодировки элементов
перестановки
      BigInteger val;
      BigInteger Max_val;
      BigInteger Set_size;
      BigInteger result;
      int       perm_num = 0;
      int       total_cnt;
      Formatter  format = new Formatter();
      Scanner    in     = new Scanner(System.in);

      System.out.println("Enter information block size: ");
      k = in.nextInt();

      //Расчет максимального значения входного вектора - 2^k
      Max_val = BigInteger.valueOf(2);
      Max_val = Max_val.pow(k);

      //Начальное значение - 0!
      result = BigInteger.ONE;
      //Поиск M, M! >= 2^k
      while(Max_val.compareTo(result) == 1)
      {
         M++;
         result = Factorial_Lib.BigIntFact(M);
      }
```

```
        //Общая мощность перестановок - M!
        Set_size = Factorial_Lib.BigIntFact(M);

        //Выделение памяти под структуры
        int[] P   = new int[M];           // Перестановка
        int[] Sf  = new int[M];           // Синдром перестановки
        int[] E   = new int[M];           // Вектор ошибки

        val = BigInteger.ZERO;
        int permutations[][] = new int[Set_size.intValue()][M];
        total_cnt  = 0;

        //Расчет количества разрядов, необходимых для записи элемента
перестановки
        rc  = (int)Math.ceil(Math.log((double)M) / Math.log(2.));
        //Выделение памяти под структуры
        wt  = new long[M * rc + 1];

        do
        {
           //Формирование синдрома перестановки
           Factorial_Lib.NumberToSyndrom(val, Sf, M);
           //Формирование перестановки
           Factorial_Lib.SyndromToTransp(Sf, P, M);

           int invers_cnt = 0;
           for(int i = 0; i < M; i++)
              invers_cnt += Sf[i];
           invers_cnt &= 1;

           //Вставка бита четности
           if(invers_cnt > 0)
              P[M - 1] |= 1 << rc;

           System.arraycopy(P, 0, permutations[total_cnt++], 0, M);
           val = val.add(BigInteger.ONE);
        }
        while (total_cnt < Set_size.intValue());

        int tmp[] = new int[M];
        //Перемешивание множества разрешенных перестановок
        for(int i = 0; i < Max_val.intValue(); i++)
        {
           val = BigInteger.valueOf((int)(Math.random() * Set_size.intValue()));
```

```
    System.arraycopy(permutations[i], 0, tmp, 0, M);
    System.arraycopy(permutations[val.intValue()], 0 ,permutations[i], 0, M);
    System.arraycopy(tmp, 0 ,permutations[val.intValue()], 0, M);
}

//Обнуление таблицы веса ошибки Wt(E)
for (int i = 0; i <= M * rc; i++) {
    wt[i] = 0;
}

do
{
    total_cnt = 0;
    do
    {
        if(perm_num == total_cnt)
        {
            total_cnt++;
            continue;
        }
        int cnt = 0;

        //Наложение вектора ошибки на перестановку
        for(int i = 0; i < M; i++)
        {
            E[i] = permutations[total_cnt][i] ^ permutations[perm_num][i];
            //Подсчет кратности ошибки
            for(int j = 0; j <= rc; j++)
            {
                if((E[i] & (1 << j)) > 0)
                    cnt++;
            }
        }

        wt[cnt]++;
        total_cnt++;
    }
    while(total_cnt < Max_val.intValue());
    perm_num++;
}
while(perm_num < Max_val.intValue());


//Вывод результатов
BufferedWriter file = new BufferedWriter(new FileWriter("ФКДБ.txt", true));
```

```
        format.format("Входные данные:\r\nk = %d\tM = %d\r\n", k, M);
        for(int i = 0; i <= M * rc; i++)
        {
           format.format("%9.6f\r\n", wt[i] * 1. / perm_num);
        }
        file.write(format.toString());
        file.close();
    }
}
```

А.3. Лістинг розрахунково-експериментальної моделі для накопичення статистики невиявлених помилок ФКК при рівномірному росподілі

```
package com.company;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.math.BigInteger;
import java.util.Formatter;
import java.util.Scanner;

public class Main {
   public static void main(String[] args) throws IOException {
      long[]        wt;            // Таблица веса ошибки Wt(E)
      BigInteger[]  table;         // Таблица равновесных кодов
      int           k;             // Длина информационного блока
      int           m;             // Длина блока равновесного кода
      int           M = 0;         // Мощность перестановки
      int           rc;            // Количество бит для кодировки элементов
перестановки
      BigInteger    Ax;
      BigInteger    Am;
      BigInteger    Max_Ax;
      BigInteger    Max_Am;
      BigInteger    Set_size;
      BigInteger    result;
      int           perm_num = 0;
      int           total_cnt;
      Formatter     format = new Formatter();
      Scanner       in     = new Scanner(System.in);

      System.out.println("Enter information block size: ");
      k = in.nextInt();

      //Расчет максимального значения входного вектора - 2^k
      Max_Ax = BigInteger.valueOf(2);
      Max_Ax = Max_Ax.pow(k);

      //Начальное значение - 0!
      result = BigInteger.ONE;
      //Поиск m, C(0.5m,m) >= 2^k
      m = k;
      while(Max_Ax.compareTo(result) == 1)
```

```
        {
            m++;
            result =
Factorial_Lib.BigIntFact(m).divide(Factorial_Lib.BigIntFact(Math.round(m /
2)).multiply(Factorial_Lib.BigIntFact(m - Math.round(m / 2))));
        }

        //Расчет максимального значения равномерного вектора - 2^m
        Max_Am = BigInteger.valueOf(2);
        Max_Am = Max_Am.pow(m);

        //Начальное значение - 0!
        result = BigInteger.ONE;
        //Поиск M, M! >= 2^m
        while(Max_Am.compareTo(result) == 1)
        {
            M++;
            result = Factorial_Lib.BigIntFact(M);
        }

        //Общая мощность перестановок - M!
        Set_size = Factorial_Lib.BigIntFact(M);
        table = new BigInteger[Max_Ax.intValue()];

        int count = 0;
        //Формирование таблицы равновесных кодов
        Am = BigInteger.ZERO;
        while(count < table.length)
        {
            Am = Am.add(BigInteger.ONE);

            int bit_cnt = 0;
            for(int i = 0; i < m; i++)
            {
                if(Am.testBit(i))
                    bit_cnt++;
            }
            if(bit_cnt != Math.round(m / 2))
                continue;
            // Поиск по таблице, если такого вектора еще нет - добавляем
            int cmp = 0;
            for(int i = 0; i < count; i++)
            {
                if(table[i].compareTo(Am) == 0)
                    cmp = 1;
```

```
            }

            if(cmp == 0)
            {
               table[count] = Am;
               count++;
            }
         }

         //Выделение памяти под структуры
         int[] P   = new int[M];            // Перестановка
         int[] Sf  = new int[M];            // Синдром перестановки
         int[] E   = new int[M];            // Вектор ошибки
         int permutations[][] = new int[Set_size.intValue()][M];
         total_cnt  = 0;

         //Расчет количества разрядов, необходимых для записи элемента
перестановки
         rc  = (int)Math.ceil(Math.log((double)M) / Math.log(2.));
         //Выделение памяти под структуры
         wt  = new long[M * rc + 1];

         for(int i = 0; i < table.length; i++) {
            //Формирование синдрома перестановки
            Factorial_Lib.NumberToSyndrom(table[i], Sf, M);
            //Формирование перестановки
            Factorial_Lib.SyndromToTransp(Sf, P, M);

            System.arraycopy(P, 0, permutations[total_cnt++], 0, M);
         }

         //Обнуление таблицы веса ошибки Wt(E)
         for (int i = 0; i <= M * rc; i++) {
            wt[i] = 0;
         }

         do
         {
            total_cnt = 0;
            do
            {
               if(perm_num == total_cnt)
               {
                  total_cnt++;
                  continue;
```

```
            }
            int cnt = 0;

            //Наложение вектора ошибки на перестановку
            for(int i = 0; i < M; i++)
            {
               E[i] = permutations[total_cnt][i] ^ permutations[perm_num][i];
               //Подсчет кратности ошибки
               for(int j = 0; j <= rc; j++)
               {
                  if((E[i] & (1 << j)) > 0)
                     cnt++;
               }
            }

            wt[cnt]++;
            total_cnt++;
         }
         while(total_cnt < Max_Ax.intValue());
         perm_num++;
      }
      while(perm_num < Max_Ax.intValue());


      //Вывод результатов
      BufferedWriter file = new BufferedWriter(new FileWriter("ФКК
(усреднение).txt", true));

      format.format("Входные данные:\r\nk = %d\tm = %d\tM = %d\r\n", k, m, M);
      for(int i = 0; i <= M * rc; i++)
      {
         format.format("%9.6f\r\n", wt[i] * 1. / perm_num);
      }
      file.write(format.toString());
      file.close();
   }
}
```

А.4. Лістинг розрахунково-експериментальної моделі для формування СКК на основі теорії решіток

```
package com.company;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.math.BigInteger;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Formatter;
import java.util.Scanner;

public class Main {
    public static class Pair {
        int upper[];
        int bottom[];
    }

    public static int[] insert(int val, int pos, int[] array, int size) {
        int[] new_array = new int[size + 1];
        System.arraycopy(array, 0, new_array, 0, pos);
        new_array[pos] = val;
        System.arraycopy(array, pos, new_array, pos + 1, size - pos);
        return new_array;
    }

    public static boolean CheckPairDistance(Pair a, Pair b, int dest, int size) {
        int distance = 0;
        int Byte;
        for (int j = 0; j < size; j++) {
            Byte = a.upper[j] ^ b.upper[j];

            for (int pos = 0; pos < 8; pos++) {
                if ((Byte & (1 << pos)) > 0)
                    distance++;
            }
        }

        if (distance <= dest) {
            return false;
        }
        distance = 0;
```

```
    for (int j = 0; j < size; j++) {
        Byte = a.bottom[j] ^ b.bottom[j];

        for (int pos = 0; pos < 8; pos++) {
            if ((Byte & (1 << pos)) > 0)
                distance++;
        }
    }

    if (distance <= dest) {
        return false;
    }

    distance = 0;
    for (int j = 0; j < size; j++) {
        Byte = a.upper[j] ^ b.bottom[j];

        for (int pos = 0; pos < 8; pos++) {
            if ((Byte & (1 << pos)) > 0)
                distance++;
        }
    }

    if (distance <= dest) {
        return false;
    }

    distance = 0;
    for (int j = 0; j < size; j++) {
        Byte = a.bottom[j] ^ b.upper[j];

        for (int pos = 0; pos < 8; pos++) {
            if ((Byte & (1 << pos)) > 0)
                distance++;
        }
    }

    if (distance <= dest) {
        return false;
    }
    return true;
}

public static void main(String[] args) throws IOException {
```

```
        BigInteger val;                         // Значение информационного вектора
        BigInteger Set_size;                    // Размер множества перестановок
        Formatter format = new Formatter();     // Форматированный буфер
        Scanner in = new Scanner(System.in);    // Обработчик перерываний от
I/O устройств
        int total_cnt;                          // Вспомагательный счетчик
        int class_size;                         // Размер класса вычетов
        int M;                                  // Мощность перестановки
        int rc;                                 // Количество бит для кодировки элементов
перестановки
        int t;                                  // Расстояние между элементами решетки

        System.out.println("Введите M: ");
        M = in.nextInt();

        System.out.println("Введите t (расстояние между элементами решетки): ");
        t = in.nextInt();
        //Расчет количества разрядов, необходимых для записи элемента
перестановки
        rc = (int) Math.ceil(Math.log((double) M) / Math.log(2.));

        //Выделение памяти под структуры
        int[] P = new int[M];           // Перестановка
        int[] E = new int[M];           // Вектор ошибки
        int[] Sf = new int[M];          // Синдром перестановки

        //Общая мощность перестановок - M!
        Set_size = Factorial_Lib.BigIntFact(M);

        val = BigInteger.ZERO;
        int permutations[][] = new int[Set_size.intValue()][M];
        total_cnt = 0;
        class_size = 0;

        do {
           total_cnt++;
           //Формирование синдрома перестановки
           Factorial_Lib.NumberToSyndrom(val, Sf, M);
           //Формирование перестановки
           Factorial_Lib.SyndromToTransp(Sf, P, M);
           System.arraycopy(P, 0, permutations[class_size++], 0, M);
           val = val.add(BigInteger.ONE);
        }
        while (total_cnt < Set_size.intValue());
```

```
ArrayList<Pair> generic_pair = new ArrayList<>();

total_cnt = 1;
System.arraycopy(permutations[0], 0, P, 0, M);

//Поиск порождающих пар перестановок
do {
   //Счетчик веса ошибки
   int err_cnt = 0;

   //Определение расстояния Хемминга между текущей перестановкой  и другими перестановками из разрешенного множества
   for (int i = 0; i < M; i++) {
      E[i] = P[i] ^ permutations[total_cnt][i];
      //Подсчет кратности ошибки
      for (int j = 0; j < rc; j++) {
         if ((E[i] & (1 << j)) > 0)
            err_cnt++;
      }
   }
   //Если расстояние больше минимального - сохранить пару
   if (err_cnt > t) {
      Pair tmp = new Pair();
      tmp.upper = new int[M];
      tmp.bottom = new int[M];
      System.arraycopy(P, 0, tmp.upper, 0, M);
      System.arraycopy(permutations[total_cnt], 0, tmp.bottom, 0, M);

      generic_pair.add(tmp);
   }
   //Инкримент счетчика цыкла
   total_cnt++;
}
while (total_cnt < Set_size.intValue());

ArrayList<Pair> Grid = new ArrayList<>(); //Масив для пар,входящих в решетку
//Цыкл по всем генерирующим парам
for (int i = 0; i < generic_pair.size(); i++) {
   Pair fixed_pair         = new Pair();
   Pair current_pair       = generic_pair.get(i);
   Pair transmutation_pair = new Pair();
   //Запись M на 0ю позицию для генерирующей пары
   fixed_pair.upper  = insert(M, 0, current_pair.upper, M);
   fixed_pair.bottom = insert(M, 0, current_pair.bottom, M);
```

```
        //Подготовка пары-претендента на вхождение в решетку. Записываем M
на последнюю позицию для дальнейшей перестановки
        transmutation_pair.upper  = insert(M, M, current_pair.upper, M);
        transmutation_pair.bottom = insert(M, M, current_pair.bottom, M);

        int iterations_cnt       = 1; //Счетчик итераций, он же - позиция M в
перестановке
        int permutations_new[][] = new int[Set_size.intValue()][M + 1];
//Расширеное множество перестановок с M на фиксированной позиции
        do {
           //Формирование нового множества с M на позиции,равной
iterations_cnt
           for (int j = 0; j < Set_size.intValue(); j++) {
              permutations_new[j] = insert(M, iterations_cnt, permutations[j], M);
           }
           int  tmp[] = new int[M + 1]; //Временный массив для формирования
перестановок генерирующей пары

           total_cnt = 0; //Счетчик узлов
           do {
              Pair node  = new Pair();  //Сформированная пара-претендент

              node.upper  = new int[M + 1];
              node.bottom = new int[M + 1];

              System.arraycopy(transmutation_pair.upper, 0, tmp, 0, M + 1);
              for (int j = 0; j < M + 1; j++)
                 node.upper[j] = tmp[permutations_new[total_cnt][j]];

              System.arraycopy(transmutation_pair.bottom, 0, tmp, 0, M + 1);
              for (int j = 0; j < M + 1; j++)
                 node.bottom[j] = tmp[permutations_new[total_cnt][j]];

              if (CheckPairDistance(fixed_pair, node, t, M + 1))
                 Grid.add(node);
              total_cnt++;
           } while (total_cnt < Set_size.intValue());
           iterations_cnt++;
        }
        while (iterations_cnt <= M);
     }
     System.out.println("Завершение моделирования");
  }
}
```

А.5. Лістинг розрахунково-експериментальної моделі для відновлення прийнятих з помилкою перестановок в метриці Хеммінга

```
package com.company;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.math.BigInteger;
import java.util.*;

public class Main {
                                        //0  1  2  3  4  5  6  7
    static public int[][] weight = {{0, 1, 1, 2, 1, 2, 2, 3},  // 0
                                    {1, 0, 2, 1, 2, 1, 3, 2},  // 1
                                    {1, 2, 0, 1, 2, 3, 1, 2},  // 2
                                   {2, 1, 1, 0, 3, 2, 2, 1},  // 3
                                   {1, 2, 2, 3, 0, 1, 1, 2},  // 4
                                   {2, 1, 3, 2, 1, 0, 2, 1},  // 5
                                   {2, 3, 1, 2, 1, 2, 0, 1},  // 6
                                   {3, 2, 2, 1, 2, 1, 1, 0}}; // 7

    public static void main(String[] args) throws IOException {

        long[][] wt;            // Таблица веса ошибки Wt(E)
        long[] dec_err;       // Таблица ошибок декодера
        long err_bit_cnt = 0;
        int M = 0;              // Мощность перестановки
        int rc;                // Количество бит для кодировки элементов перестановки
        BigInteger val;         // Переменная для хранения десятичного представления
перестановки
        BigInteger Max_val;     // Максимальное значение информационного
вектора
        BigInteger Set_size;    // Размер множества перестановок
        int perm_num;           // Текущий номер перестановки
        int prev_perm_num;      // Предидущий номер перестановки
        int total_cnt;          // Счетчик выборок
        long signal_ratio[];    // Уровень сигнала и шума декодирования
        int ch_state = 0;       // Текущее состояние канала
        double[][] prob = new double[2][2];

        Scanner in = new Scanner(System.in);
        Formatter format = new Formatter();
```

```
//Ввод вероятности перехода из хорошего состояния в плохое
System.out.println("Enter Pхп: ");
prob[0][0] = in.nextFloat();
prob[0][1] = 1e-6;
//Ввод вероятности перехода из плохого состояния в хорошее
System.out.println("Enter Pпх: ");
prob[1][0] = in.nextFloat();
prob[1][1] = 0.35;

//Расчет максимального значения входного вектора - 2^15
Max_val = BigInteger.valueOf(2);
Max_val = Max_val.pow(15);

//Начальное значение - 0!
val = BigInteger.ONE;
//Поиск M, M! >= 2^k
while (Max_val.compareTo(val) == 1) {
   M++;
   val = Factorial_Lib.BigIntFact(M);
}

//Общая мощность перестановок - M!
Set_size = Factorial_Lib.BigIntFact(M);

//Выделение памяти под структуры
int[] P = new int[M];              // Перестановка
int[] Sf = new int[M];             // Синдром перестановки
int[] E = new int[M];              // Вектор ошибки

val = BigInteger.ZERO;
int permutations[][] = new int[Set_size.intValue()][M];
total_cnt = 0;

do {
   //Формирование синдрома перестановки
   Factorial_Lib.NumberToSyndrom(val, Sf, M);
   //Формирование перестановки
   Factorial_Lib.SyndromToTransp(Sf, P, M);
   System.arraycopy(P, 0, permutations[total_cnt++], 0, M);
   val = val.add(BigInteger.ONE);
}
while (total_cnt < Set_size.intValue());

Scanner file_scanner = new Scanner(new File("test.txt"));
```

```
        ArrayList<Integer> selections = new ArrayList<>();
        //Чтение из файла массива выборок
        while (file_scanner.hasNextInt()) {
            selections.add((file_scanner.nextInt() >> 1) + 16384);
        }

        //Формирование разрешенного подмножества перестановок
        file_scanner = new Scanner(new File("Таблица замен.txt"));
        //Чтение из файла массива выборок
        total_cnt = 0;
        while (file_scanner.hasNextInt()) {
            System.arraycopy(permutations[file_scanner.nextInt()], 0,
permutations[total_cnt], 0, M);
            total_cnt++;
        }

        //Расчет количества разрядов, необходимых для записи элемента
перестановки
        rc = (int) Math.ceil(Math.log((double) M) / Math.log(2.));

        //Выделение памяти под структуры
        wt = new long[M * rc + 1][5];
        dec_err = new long[16];
        signal_ratio = new long[2];
        //Обнуление таблицы веса ошибки Wt(E)
        for (int i = 0; i <= M * rc; i++) {
            wt[i][0] = 0;
            wt[i][1] = 0;
            wt[i][2] = 0;
            wt[i][3] = 0;
            wt[i][4] = 0;
        }

        for (int i = 0; i <= 15; i++) {
            dec_err[i] = 0;
        }

        signal_ratio[0] = 0;
        signal_ratio[1] = 0;
        total_cnt = 0;

        prev_perm_num = -1;
        //Основной цикл программы
        long startTime = System.nanoTime();
        do {
```

```
//Генерирование входного вектора
perm_num = selections.get(total_cnt);

//Инкримент счетчика посылок
total_cnt++;
double new_state = Math.random();
if( new_state <= prob[ch_state][0]) {
   ch_state ^= 1;
}

int cnt = 0;
int tmp1;
//Генерирование вектора ошибки
for (int i = 0; i < M; i++) {
   tmp1 = 0;
   for (int j = 0; j < rc; j++) {
      if (Math.random() <= prob[ch_state][1]) {
         tmp1 |= 1 << j;
         cnt++;
      }
   }
   E[i] = tmp1;
}
wt[cnt][0]++;
err_bit_cnt += cnt;
//Наложение вектора ошибки на перестановку
for (int i = 0; i < M; i++) {
   P[i] = permutations[perm_num][i] ^ E[i];
}

//Проверка является ли блок данных перестановкой
int compare_res = 0;
for (int j = 0; j < M; j++) {
   if (P[j] < M) {
      for (int k = j + 1; k < M; k++) {
         if ((P[j] ^ P[k]) == 0) {
            compare_res = 1;
            break;
         }
      }
   } else
      compare_res = 1;

   if (compare_res > 0)
      break;
```

```
        }

        int min_dist = 9999;
        int num;
        ArrayList<Integer> closiest = new ArrayList<>();
        //Поиск ближайшей перестановки из разрешенного множества,
соответствующей принятому блоку
        for (int i = 0; i < Max_val.intValue(); i++) {
            int dist = 0;
            for (int pos = 0; pos < M; pos++)
                dist += weight[P[pos]][permutations[i][pos]];
            //Если расстояние Хемминга меньше или равно минимальному, то
выбираем перестановку, которая соответствует
            //ближайшей выборке к предидущей в метрике Эвклида
            if (dist <= min_dist) {
                if(dist < min_dist) {
                    closiest.clear();
                    min_dist = dist;
                }
                closiest.add(i);
            }
            // Если найдено полное совпадение - выход
            if (min_dist == 0)
                break;
        }
        num = closiest.get(0);
        if(closiest.size() > 1)
        {
            if (prev_perm_num > 0)
            {

                for(int i = 1; i < closiest.size(); i++) {
                    int el = closiest.get(i);
                    if (Math.abs(prev_perm_num - el) < Math.abs(prev_perm_num -
num)) {
                        num = el;
                    }
                    else if (Math.abs(prev_perm_num - el) == Math.abs(prev_perm_num -
num)) {
                        num = prev_perm_num;
                        break;
                    }
                }
            }
```

```
            }
            // Обнаруженные ошибки
            if (num == perm_num)
               //Корректно исправленные ошибки
               wt[cnt][1]++;
            else {
               //Ошибки исправления
               if (min_dist > 0) {
                  //Если в результате воздействия ошибки получена не перестановка
                  if (compare_res > 0)
                     wt[cnt][2]++;
                  else {
                     //Если в результате воздействия ошибки произошла
трансформация в перестановку из запрещенного множества
                     wt[cnt][3]++;
                  }
               } else {
                  //Если в результате воздействия ошибки произошла трансформация
в перестановку из разрешенного множества
                  wt[cnt][4]++;
               }
            }
            // Ошибки декодера
            if (num != perm_num) {
               int dist = num ^ perm_num;
               cnt = 0;
               while (dist > 0) {
                  cnt += dist & 0x01;
                  dist >>= 1;
               }
               dec_err[cnt]++;
            }
            //Накапливаем уровень сигнала и шум декодирования
            signal_ratio[0] += Math.pow(perm_num, 2);
            signal_ratio[1] += Math.pow(perm_num - num, 2);

            //Перезаписываем номер предидущей перестановки
            prev_perm_num = num;
            System.out.format("\rThe simulation process is running....%6.2f %%",
total_cnt * 100. / selections.size());
         }
         while (total_cnt < selections.size());
         long endTime   = System.nanoTime();
         long totalTime = endTime - startTime;
         System.out.println();
```

```
        //Вывод результатов
        BufferedWriter file = new BufferedWriter(new FileWriter("ФКМХ.txt", true));
        format.format("Входные данные:\r\nk = %d\tM = %d\tV = %d\tPхп =
%6.5f\tPпх = %6.5f\tp0х = %e\t p0п = %e\tp0 = %e\r\n", 15, M, total_cnt,
prob[0][0], prob[1][0], prob[0][1], prob[1][1], (1. * err_bit_cnt / (total_cnt * rc *
M)));
        for (int i = 0; i <= (M * rc); i++) {
            format.format("%d\t%d\t%d\t%d\t%d\t%d\r\n", i, wt[i][0], wt[i][1], wt[i][2],
wt[i][3], wt[i][4]);
        }
        format.format("Ошибки декодера\r\n");
        for (int i = 0; i <= 15; i++) {
            format.format("%d\t%d\r\n", i, dec_err[i]);
        }
        format.format("Защищенность:\t%6.3f\r\n", 10 * Math.log10(signal_ratio[0] *
1. / total_cnt) - 10 * Math.log10(signal_ratio[1] * 1. / total_cnt));
        format.format("Среднее время декодирования:\t%e\r\n", 1. *
Math.round(totalTime * 1e-9) / total_cnt);
        file.write(format.toString());
        file.close();
    }
}
```

А.6. Лістинг розрахунково-експериментальної моделі для відновлення прийнятих з помилкою перестановок методом лінійної інтерполяції

```
package com.company;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.math.BigInteger;
import java.util.*;

public class Main {
                                        //0  1  2  3  4  5  6  7
   static public int[][] weight = {{0, 1, 1, 2, 1, 2, 2, 3},   // 0
                                    {1, 0, 2, 1, 2, 1, 3, 2},  // 1
                                    {1, 2, 0, 1, 2, 3, 1, 2},  // 2
                                   {2, 1, 1, 0, 3, 2, 2, 1},   // 3
                                   {1, 2, 2, 3, 0, 1, 1, 2},   // 4
                                   {2, 1, 3, 2, 1, 0, 2, 1},   // 5
                                   {2, 3, 1, 2, 1, 2, 0, 1},   // 6
                                   {3, 2, 2, 1, 2, 1, 1, 0}}; // 7

   private static void fix_errors(ArrayList<Integer> queue, int first_perm, int
last_perm, int packet_len)
   {
      float delta = (first_perm - last_perm) / (packet_len + 1.f);
      boolean fract = (delta - (int)delta) != 0;

      for(int i = 1; i <= packet_len; i++) {
         queue.remove(packet_len - i);
         queue.add(packet_len - i, last_perm + (int)((fract && ((i & 1) == 1)) ? delta +
1 : delta));
      }
   }
   public static void main(String[] args) throws IOException {

      long[][] wt;           // Таблица веса ошибки Wt(E)
      long[] dec_err;        // Таблица ошибок декодера
      long err_bit_cnt = 0;
      int M = 0;             // Мощность перестановки
      int rc;               // Количество бит для кодировки элементов перестановки
      int p0;               // Вероятность битовой ошибки в канале
```

```
        BigInteger val;         // Переменная для хранения десятичного представления
перестановки
        BigInteger Max_val;     // Максимальное значение информационного
вектора
        BigInteger Set_size;    // Размер множества перестановок
        int perm_num;           // Текущий номер перестановки
        int total_cnt;          // Счетчик выборок
        long signal_ratio[];    // Уровень сигнала и шума декодирования
        int ch_state = 0;       // Текущее состояние канала
        double[][] prob = new double[2][2];


        ArrayList<Integer> queue = new ArrayList<>();
        Scanner in = new Scanner(System.in);
        Formatter format = new Formatter();

        //Ввод вероятности перехода из хорошего состояния в плохое
        System.out.println("Enter Pхп: ");
        prob[0][0] = in.nextFloat();
        prob[0][1] = 1e-6;
        //Ввод вероятности перехода из плохого состояния в хорошее
        System.out.println("Enter Pпх: ");
        prob[1][0] = in.nextFloat();
        prob[1][1] = 0.35;

        //Расчет максимального значения входного вектора - 2^15
        Max_val = BigInteger.valueOf(2);
        Max_val = Max_val.pow(15);

        //Начальное значение - 0!
        val = BigInteger.ONE;
        //Поиск M, M! >= 2^k
        while (Max_val.compareTo(val) == 1) {
            M++;
            val = Factorial_Lib.BigIntFact(M);
        }

        //Общая мощность перестановок - M!
        Set_size = Factorial_Lib.BigIntFact(M);

        //Выделение памяти под структуры
        int[] P = new int[M];               // Перестановка
        int[] Sf = new int[M];              // Синдром перестановки
        int[] E = new int[M];               // Вектор ошибки
```

```
        val = BigInteger.ZERO;
        int permutations[][] = new int[Set_size.intValue()][M];
        total_cnt = 0;

        do {
            //Формирование синдрома перестановки
            Factorial_Lib.NumberToSyndrom(val, Sf, M);
            //Формирование перестановки
            Factorial_Lib.SyndromToTransp(Sf, P, M);
            System.arraycopy(P, 0, permutations[total_cnt++], 0, M);
            val = val.add(BigInteger.ONE);
        }
        while (total_cnt < Set_size.intValue());

        Scanner file_scanner = new Scanner(new File("test.txt"));
        ArrayList<Integer> selections = new ArrayList<>();
        //Чтение из файла массива выборок
        while (file_scanner.hasNextInt()) {
            selections.add((file_scanner.nextInt() >> 1) + 16384);
        }

        //Формирование разрешенного подмножества перестановок
        file_scanner = new Scanner(new File("Таблица замен.txt"));
        //Чтение из файла массива выборок
        total_cnt = 0;
        while (file_scanner.hasNextInt()) {
            System.arraycopy(permutations[file_scanner.nextInt()], 0,
permutations[total_cnt], 0, M);
            total_cnt++;
        }

        //Расчет количества разрядов, необходимых для записи элемента
перестановки
        rc = (int) Math.ceil(Math.log((double) M) / Math.log(2.));

        //Выделение памяти под структуры
        wt = new long[M * rc + 1][5];
        dec_err = new long[16];
        signal_ratio = new long[2];
        //Обнуление таблицы веса ошибки Wt(E)
        for (int i = 0; i <= M * rc; i++) {
            wt[i][0] = 0;
            wt[i][1] = 0;
            wt[i][2] = 0;
            wt[i][3] = 0;
```

```
    wt[i][4] = 0;
}

for (int i = 0; i <= 15; i++) {
    dec_err[i] = 0;
}

signal_ratio[0] = 0;
signal_ratio[1] = 0;
total_cnt = 0;

int prev_decoded_perm = 0;
int current_decoded_perm = 0;
//Основной цикл программы
int error_packet_len = 0;
long startTime = System.nanoTime();
do {
    //Генерирование входного вектора
    perm_num = selections.get(total_cnt);

    int cnt = 0;
    for(int i = 0; i < M; i++){
        for(int j = 0; j < rc; j++){
            if(((E[i] >> j) & 1) == 1)
                cnt++;
        }
    }

    wt[cnt][0]++;
    err_bit_cnt += cnt;

    //Наложение вектора ошибки на перестановку
    for (int i = 0; i < M; i++) {
        P[i] = permutations[perm_num][i] ^ E[i];
    }

    //Проверка является ли блок данных перестановкой
    int compare_res = 0;
    for (int j = 0; j < M; j++) {
        if (P[j] < M) {
            for (int k = j + 1; k < M; k++) {
                if ((P[j] ^ P[k]) == 0) {
                    compare_res = 1;
                    break;
                }
```

```
                }
            } else
                compare_res = 1;

            if (compare_res > 0)
                break;
        }

        int num = 0;
        if(compare_res > 0)
        {
            // Ошибка трансформации перестановки в не перестановку
            wt[cnt][2]++;
            queue.add(0, ++error_packet_len + (1 << 15));
        }
        else {
            compare_res = 1;
            for (int i = 0; i < Max_val.intValue(); i++) {
                int dist = 0;
                for (int pos = 0; pos < M; pos++)
                    dist += weight[P[pos]][permutations[i][pos]];
                // Если есть совпадение
                if (dist == 0) {
                    // Если есть не исправленные ошибки - фиксируем текущую
перестановку как конец пакета ошибок
                    if(error_packet_len > 0)
                        current_decoded_perm = i;
                    else
                    // Если ошибок до этого не было, просто запоминаем как
предидущую верно принятую перестановку
                        prev_decoded_perm = i;
                    num = i;
                    //Если перестановка совпадает с исходной
                    if(num == perm_num)
                        wt[cnt][1]++;
                    else
                        // Иначе - ошибка трансформации перестановки в перестановку
                        wt[cnt][4]++;
                    compare_res = 0;
                    break;
                }
            }

            // Если совпадений в таблице замен не найдено - ошибка
трансформации в перестановку из запрещенного множества
```

```
            if(compare_res > 0) {
                wt[cnt][3]++;
                queue.add(0, ++error_packet_len + (1 << 15));
            }
        }

        // Если на текущей итерации декодер принял последовательность из таблицы замен
        if(compare_res == 0)
        {
            // Проверяем были ли до этого приняты блоки с ошибками, если да - исправляем
            if(error_packet_len > 0) {
                fix_errors(queue, prev_decoded_perm, current_decoded_perm, error_packet_len);
                for(int i = queue.size(); i > 0; i--) {
                    int fixed_perm = queue.get(queue.size() - i);
                    int real_perm = selections.get(total_cnt - i);

                    // Ошибки декодера
                    if (fixed_perm != real_perm) {
                        int dist = fixed_perm ^ real_perm;
                        cnt = 0;
                        while (dist > 0) {
                            cnt += dist & 0x01;
                            dist >>= 1;
                        }
                        dec_err[cnt]++;
                    }
                    //Накапливаем уровень сигнала и шум декодирования
                    signal_ratio[0] += Math.pow(real_perm, 2);
                    signal_ratio[1] += Math.pow(real_perm - fixed_perm, 2);
                }
                prev_decoded_perm = current_decoded_perm;
                error_packet_len = 0;
                queue.clear();
            }
            else
            {
                // Ошибки декодера
                if (num != perm_num) {
                    int dist = num ^ perm_num;
                    cnt = 0;
                    while (dist > 0) {
                        cnt += dist & 0x01;
```

```
                    dist >>= 1;
                }
                dec_err[cnt]++;
            }
            //Накапливаем уровень сигнала и шум декодирования
            signal_ratio[0] += Math.pow(perm_num, 2);
            signal_ratio[1] += Math.pow(perm_num - num, 2);
        }
    }

    double new_state = Math.random();
    if(new_state <= prob[ch_state][0]) {
        ch_state ^= 1;
    }

    //Генерирование вектора ошибки
    int tmp1;
    for (int i = 0; i < M; i++) {
        tmp1 = 0;
        for (int j = 0; j < rc; j++) {
            if (Math.random() <= prob[ch_state][1]) {
                tmp1 |= 1 << j;
                cnt++;
            }
        }
        E[i] = tmp1;
    }

    //Инкримент счетчика посылок
    total_cnt++;

    System.out.format("\rThe simulation process is running....%6.2f %%",
total_cnt * 100. / selections.size());
}
while (total_cnt < selections.size());
long endTime   = System.nanoTime();
long totalTime = endTime - startTime;

System.out.println();
//Вывод результатов
BufferedWriter file = new BufferedWriter(new FileWriter("ФКЛП.txt", true));

format.format("Входные данные:\r\nk = %d\tM = %d\tV = %d\tPхп =
%6.5f\tPпх = %6.5f\tp0x = %e\t p0п = %e\tp0 = %e\r\n", 15, M, total_cnt,
```

```
prob[0][0], prob[1][0], prob[0][1], prob[1][1], (1. * err_bit_cnt / (total_cnt * rc *
M)));
        for (int i = 0; i <= (M * rc); i++) {
            format.format("%d\t%d\t%d\t%d\t%d\t%d\r\n", i, wt[i][0], wt[i][1], wt[i][2],
wt[i][3], wt[i][4]);
        }

        format.format("Ошибки декодера\r\n");
        for (int i = 0; i <= 15; i++) {
            format.format("%d\t%d\r\n", i, dec_err[i]);
        }
        format.format("Защищенность:\t%6.3f\r\n", 10 * Math.log10(signal_ratio[0] *
1. / total_cnt) - 10 * Math.log10(signal_ratio[1] * 1. / total_cnt));
        format.format("Среднее время декодирания:\t%e\r\n", 1. *
Math.round(totalTime * 1e-9) / total_cnt);
        file.write(format.toString());
        file.close();
    }
}
```

Додаток Б. Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації

Наукові праці, в яких опубліковані основні наукові результати дисертації

[1] О. О. Харін, «Оцінка властивостей каскадного коду, що поєднує факторіальний та рівноважний код», *Вісник Черкаського державного технологічного університету*, № 2, с. 86-90, 2017.

[2] О. О. Харін, «Порівняльна оцінка факторіальних кодів», *Вісник Черкаського державного технологічного університету. Серія: технічні науки*, № 4, с. 88-93, 2017.

[3] E. V. Faure, A. I. Shcherba and A. A. Kharin, «Factorial Code with a Given Number of Inversions», *Radio Electronics, Computer Science, Control*, vol. 2, p. 143-153, 2018.

[4] E. V. Faure, V. V. Shvydkyi, A. O. Lavdanskyi and O. O. Kharin, «Methods of factorial coding of speech signals», *Radio Electronics, Computer Science, Control*, vol. 4, p. 186-198, 2019.

[5] J. Al-Azzeh, B. Ayyoub, E. Faure, V. Shvydkyi, O. Kharin and A. Lavdanskyi, «Telecommunication Systems with Multiple Access Based on Data Factorial Coding», *International Journal on Communications Antenna and Propagation (IRECAP)*, vol. 10, issue 2, p. 102-113, 2020.

Наукові праці, які засвідчують апробацію матеріалів дисертації

[1] Е. В. Фауре та О. О. Харін, «Дослідження ймовірності виникнення помилки декодування під час використання факторіального коду з відновленням даних», в *Актуальні задачі та досягнення у галузі кібербезпеки: Тези доповідей Всеукраїнської науково-практичної конференції, Кропивницький, 23-25 листопада 2016 р.*, Кропивницький, 2016, с. 178-179.

[2] Е. В. Фауре та О. О. Харін, «Факторіальне кодування з відновленням даних і виправленням помилок», в *Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку: Тези доповідей Всеукраїнської науково-*

практичної Internet-конференції, Черкаси, 13-19 березня 2017 р., Черкаси: ЧДТУ, 2017, с. 74–76.

[3] О. О. Харін, «Підвищення достовірності передачі даних за допомогою факторіального кодування з виявленням транспозицій», в *Надзвичайні ситуації: безпека та захист: Тези доповідей VII Всеукраїнської науково-практичної конференції з міжнародною участю, Черкаси, 20-21 жовтня 2017 р.*, Черкаси: ЧІПБ ім. Героїв Чорнобиля НУЦЗ України, 2017, с. 162-163.

[4] А. А. Харин и А. И. Щерба, «Организация замкнутой группировки абонентов в открытой сети коллективного пользования», в *Інформаційні технології в освіті, науці і техніці (ІТОНТ-2018): Тези доповідей IV Міжнародної науково-практичної конференції, Черкаси, 17-18 травня 2018 р.*, Черкаси: ЧДТУ, 2018, с. 109-111.

[5] О. О. Харін, «Формування сигнально-кодової конструкції на основі теорії решіток», в *Наука України – погляд молодих вчених крізь призму сучасності: Тези доповідей II Всеукраїнської науково-практичної конференції з міжнародною участю, Черкаси, 26 вересня 2019 р.*, Черкаси: ЧДТУ, 2019, с.42-44.

[6] О. О. Харін, Е. В. Фауре та А. О. Лавданський, «Оцінка захищеності мовного сигналу в системах з факторіальним кодуванням», в *Інформаційні технології в освіті, науці і техніці (ІТОНТ-2020): Тези доповідей V Міжнародної науково-практичної конференції, Черкаси, 21-23 травня 2020 р.*, Черкаси: ЧДТУ, 2020, с. 85-87.

[7] Е. В. Фауре, О. О. Харін, В. В. Швидкий та А. О. Лавданський, «Ефективність виявлення помилок факторіальними кодами», в *Інформаційні технології в освіті, науці і техніці (ІТОНТ-2020): Тези доповідей V Міжнародної науково-практичної конференції, Черкаси, 21-23 травня 2020 р.*, Черкаси: ЧДТУ, 2020, с. 94-95.

Наукові праці, які додатково відображають наукові результати дисертації

[1] Е. В. Фауре, О. О. Харін, В. В. Швидкий та А. І. Щерба, «Спосіб факторіального кодування з виявленням і виправленням помилок», Україна. Пат. 121361, 11.12.2017.

[2] Е. В. Фауре та О. О. Харін, «Пристрій кодування та декодування факторіальних кодів з виявленням і виправленням помилок», Україна. Пат. 123640, 12.03.2018.

[3] О. О.Харін та А. І. Щерба, «Спосіб факторіального кодування в метриці Хеммінга», Україна. Пат. 130458, 10.12.2018.

[4] Е. В. Фауре, О. О. Харін, В. В. Швидкий та А. І. Щерба, «Спосіб факторіального кодування з відновленням даних», Україна. Пат. 117004, 12.06.2017.

[5] А. О. Лавданський, Е. В. Фауре, О. О. Харін та В. В. Швидкий, «Спосіб декодування факторіального коду з відновленням вибірок мовного сигналу реального часу в метриці Хеммінга», Україна. Пат. 137722, 11.11.2019.

[6] А. О. Лавданський, Е. В. Фауре, О. О. Харін та В. В. Швидкий, «Спосіб декодування факторіального коду з відновленням вибірок мовного сигналу реального часу методом лінійної інтерполяції», Україна. Пат. 139760, 27.01.2020.

Апробацію результатів дисертації проведено на:

- Всеукраїнській науково-практичній конференції «Актуальні задачі та досягнення у галузі кібербезпеки» (Кропивницький, 23-25 листопада 2016) – заочна участь;
- Всеукраїнській науково-практичній Internet-конференції «Автоматизація та комп’ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку» (Черкаси, 13-19 березня 2017) – заочна участь;

- VII Всеукраїнській науково-практичній конференції з міжнародною участю «Надзвичайні ситуації: безпека та захист» (Черкаси, 20-21 жовтня 2017) – очна участь;
- IV Міжнародній науково-практичній конференції «Інформаційні технології в освіті, науці і техніці (ІТОНТ-2018)» (Черкаси, 17-18 травня 2018) – очна участь;
- II Всеукраїнській науково-практичній конференції з міжнародною участю «Наука України – погляд молодих вчених крізь призму сучасності» (Черкаси, 26 вересня 2019) – заочна участь;
- V Міжнародній науково-практичній конференції «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2020) (Черкаси, 21-23 травня 2020 року) – очна участь.

Додаток В. Акт впровадження результатів дисертаційної роботи

Затверджую

Директор ТОВ «АНТЕКС УКРАЇНА»

С.М. Капрізов

«01» липня 2020 р.

Акт

впровадження результатів дисертаційної роботи

Харіна Олександра Олександровича

Під час виконання ТОВ «АНТЕКС УКРАЇНА» дослідно-конструкторських робіт з проектування систем передавання даних обладнання для паралельного водіння та точного землеробства, а також під час розробки відповідного програмного забезпечення використано наукові результати, одержані в дисертаційній роботі на здобуття ступеня доктора філософії Харіна Олександра Олександровича, а саме метод формування сигнально-кодових конструкцій для нероздільних факторіальних кодів.

За попередньо проведеними оцінками, впровадження вказаного методу дозволить підвищити ефективність і надійність системи паралельного водіння та точного землеробства за рахунок виявлення та виправлення помилок наперед заданої кратності, що досягається за рахунок визначення на етапі проектування основних параметрів сигнально-кодової конструкції – мінімальної відстані між сигнальними векторами та потужності множини символів перестановки. Це дозволить забезпечити необхідний рівень достовірності передавання даних за зміни якості каналу зв'язку в польових умовах.

Директор
ТОВ «АНТЕКС УКРАЇНА» С.М. Капрізов