МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Кваліфікаційна наукова
праця на правах рукопису

Ступка Богдан Анатолійович

ДИСЕРТАЦІЯ

МЕТОДИ ДОСТОВІРНОГО ПЕРЕДАВАННЯ ІНФОРМАЦІЇ В СИСТЕМАХ З НЕРОЗДІЛЬНИМ ФАКТОРІАЛЬНИМ КОДУВАННЯМ ДАНИХ ЗА ВИСОКОЇ ЙМОВІРНОСТІ БІТОВОЇ ПОМИЛКИ

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

Б.А. СТУПКА

Науковий керівник:
Фауре Еміль Віталійович
доктор технічних наук, професор

Черкаси – 2024

АНОТАЦІЯ

Ступка Б.А. Методи достовірного передавання інформації в системах з нероздільним факторіальним кодуванням даних за високої ймовірності бітової помилки. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття ступеня доктора філософії за спеціальністю 123 – Комп'ютерна інженерія. – Черкаський державний технологічний університет, Черкаси, 2024.

Дисертаційна робота спрямована на вирішення актуальної науково-технічної задачі, що полягає в забезпеченні достовірності передавання інформації на основі використання нероздільного факторіального кодування даних. Ця задача передбачає необхідність створення методів встановлення циклової синхронізації та методу достовірного передавання перестановки в системах з нероздільним факторіальним кодуванням даних з каналами зв'язку з високою ймовірністю бітової помилки.

Проведений у роботі аналіз методів нероздільного факторіального кодування даних свідчить, що вони одночасно вирішують проблеми криптографічного захисту інформації та завадостійкого кодування.

Виконано аналіз методів циклової синхронізації. Показано, що існують методи, які використовують підхід грубої сили та передбачають буферизацію даних двох довжин кадру й декодування за кожним можливим зміщенням синхрокомбінації. Інші методи для підвищення ефективності циклової синхронізації використовують операцію XOR для синхрокомбінацій з потоком даних. Існують також методи циклової синхронізації, які не використовують символи преамбули та передбачають адаптацію формату кадру. Разом з тим, розглянуті методи циклової синхронізації мають обмеження, що призводять до значної обчислювальної складності алгоритмів, що їх реалізують, і неможливості застосування в комунікаційних системах із блоковими кодами.

Розглянуто існуючі застосування, що реалізують нероздільне факторіальне кодування, зокрема трьохетапний криптографічний протокол на основі перестановок, який дає змогу захищеним шляхом передавати повідомлення між

двома сторонами без необхідності передавання або оголошення відкритого чи закритого ключа. Разом з тим, особливістю таких протоколів є те, що вони потребують більш високих показників достовірності, оскільки для передавання одного повідомлення дані передають тричі, що збільшує ймовірність їх ураження завадою, що особливо відчутно в умовах високого їх рівня.

Виконаний аналіз дав змогу чітко сформулювати задачі роботи. Вони полягають у розробці методу циклової синхронізації для комунікаційних систем передавання інформації з нероздільним факторіальним кодуванням, у тому числі, за ймовірності бітової помилки, близької до 0.5, а також у розробці методу достовірного передавання інформації в системах зв'язку з нероздільним факторіальним кодуванням даних за такої ж імовірності бітової помилки.

З метою верифікації та дослідження ефективності розроблених методів циклової синхронізації, достовірного передавання інформації, а також для формулювання рекомендацій щодо їх застосування, заключна задача дисертаційного дослідження полягає у виконанні порівняльної експериментальної оцінки ефективності розроблених методів.

У дисертаційній роботі представлено вперше розроблений метод циклової синхронізації, який за рахунок використання як синхрокомбінації перестановки чисел, її поділу на префіксну та суфіксну частини, а також за рахунок мажоритарної обробки прийнятих фрагментів, дозволяє забезпечити циклову синхронізацію приймальної та передавальної станцій комунікаційних систем передавання інформації з нероздільним факторіальним кодуванням. Розроблено структурну схему пристрою циклової синхронізації. Побудовано програмну модель передавання даних, в якій реалізовано розроблений алгоритм встановлення циклового синхронізму Розроблений метод дає змогу для довжини кодового слова в 8 елементів (24 біти) в каналах зв'язку з імовірністю бітової помилки $p_0 = 0,495$ отримати значення відносної частоти встановлення правильного синхронізму $W_{true} = 0,993$ та відносної частоти хибної синхронізації $W_{false} = 0,006$ для максимального коефіцієнту накопичення $l_{\max} = 90603$ фрагментів (2174472 біт). У

той же час, розроблений метод дає змогу забезпечити встановлення синхронізму, в середньому, після отримання 4, 7, 15, 54, 14526 фрагментів (96, 168, 360, 1296, 348624 біт) за ймовірностей бітової помилки . $p_0 = 0{,}1; 0{,}2; 0{,}3; 0{,}4; 0{,}495$ відповідно.

Набув подальшого розвитку метод циклової синхронізації нероздільного факторіального коду, який за рахунок використання як синхрокомбінації перестановки, яка має максимальне значення мінімальної відстані Хеммінга від її двійкового представлення до всіх її циклічних зсувів, а також за рахунок кореляційної та мажоритарної обробки фрагментів даних, що передаються каналом зв'язку, де довжина фрагмента дорівнює довжині синхрокомбінації, дозволяє забезпечити циклову синхронізацію за ймовірності бітової помилки, близької до 0.5. Розроблено структурну схему пристрою циклової синхронізації. Побудовано програмну модель передавання даних, в якій реалізовано розроблений алгоритм встановлення циклового синхронізму. Застосування розробленого методу циклової синхронізації дозволило забезпечити ймовірність правильної синхронізації $P_{true} \geq 0{,}9997$ та ймовірність хибної синхронізації $P_{false} \leq 3 \cdot 10^{-4}$ у модельній системі передавання даних з незалежними бітовими помилками з імовірністю їх появи $p_0 = 0.495$ для синхрокомбінації-перестановки з 8 елементів (24 бітів) за максимального коефіцієнту накопичення $l_{\max} = 30603$ (734472 біт) та за середньої кількості прийнятих фрагментів, що необхідна для встановлення синхронізму, 15787 фрагментів (378888 біт).

Розроблено та досліджено алгоритм перемішування прийнятих з каналу зв’язку фрагментів, що дозволяє додатково зменшити необхідний коефіцієнт накопичення для встановлення циклового синхронізму. Так, різниця між середнім значенням кількості накопичених фрагментів для систем синхронізації з перемішуванням і без перемішування накопичених фрагментів з параметрами, визначеними для ймовірності бітової помилки $p_{0\_\max} = 0.495$ та довжини синхрокомбінації-перестановки 8 елементів (24 біта), дорівнює 4607 фрагментам (110 568 бітам) за $p_0 = 0.495$, 9.1 фрагментам за $p_0 = 0.4$ і 2.6 фрагментам за

$p_0 = 0.35$, що зменшує необхідний час встановлення синхронізму. Розроблений метод може бути ефективним для реалізації не тільки в системах з нероздільним факторіальним кодуванням, а й у класичних системах передавання даних, де використовується стандартний роздільник між кадрами.

Вперше розроблено метод достовірного передавання перестановок, який за рахунок подання кожного елементу перестановки, що передається, у вигляді циклічного двійкового зсуву перестановки-переносника, що має максимальне значення мінімальної відстані Хеммінга від її двійкового представлення до всіх її циклічних зсувів, дозволяє забезпечити інформаційний обмін трьохетапним криптографічним протоколом на основі перестановок за ймовірності бітової помилки, близької до 0,5. Для підвищення достовірності передавання перестановок метод також використовує мажоритарну та кореляційну обробку фрагментів, отриманих з каналу зв'язку. Розроблено математичну модель системи передавання даних. Розроблено та реалізовано алгоритм, що застосовує запропонований метод та дає змогу за ймовірності бітової помилки $p_0 = 0.495$ досягти ймовірності приймання перестановок без помилок $P_{W\_true\_final} \geq 0.999$ і ймовірність невиявленої помилки $P_{W\_false\_final} \leq 3.6 \cdot 10^{-4}$. Результати побудованої імітаційної програмної моделі системи передавання даних підтверджують ефективність розробленого методу в порівнянні з традиційним методом DSSS: для досягнення заданої ймовірності приймання перестановок без помилок і ймовірності невиявленої помилки, що вказані вище, метод DSSS потребує приймання $l = 36413$ фрагментів (20099976 біт), в той час, як розроблений метод, потребує $l = 29123$ фрагментів (16075896 біт).

У заключній частині дисертаційної роботи досліджено ефективність методів циклової синхронізації систем передавання інформації з нероздільним факторіальним кодуванням: методу, який базувався на використанні як синхрокомбінації перестановки чисел з її поділом на префіксну та суфіксну частини, та методу, що використовує як синхрокомбінацію перестановку, що має максимальне значення мінімальної відстані Хеммінга від її двійкового

представлення до всіх її циклічних зсувів. Розроблено та описано структурні схеми імітаційних моделей системи передавання даних для кожного з методів. Описано середовище розробки та параметри апаратної частини, на якій виконувалося моделювання. Побудовано програмні імітаційні моделі систем передавання даних, у яких реалізовано алгоритми встановлення циклового синхронізму для кожного з наведених методів. Обґрунтовано основні модулі, які використовувалися як для реалізації моделей, так і для інтерпретації результатів. Розроблено рекомендації щодо застосування розроблених методів циклової синхронізації в каналах зв'язку з високою ймовірністю бітової помилки з використанням нероздільного факторіального кодування. Зокрема, продемонстровано, що за параметрів циклової синхронізації, визначених для $p_{0\_\max} = 0,495$, імовірність правильної синхронізації є вищою для методу на основі поділу синхрокомбінації на префіксну й суфіксну частини за $p_0 \le 0.495$ і вищою для методу на основі кореляційної обробки за $p_0 > 0.495$. Разом з тим, варто враховувати, що реалізація методу на основі поділу синхрокомбінації на префіксну й суфіксну частини має вищі значення ймовірності хибного фазування за $p_0 > 0.3$ у порівняння з методом на основі кореляційної обробки. Для прикладу, швидкість встановлення циклового синхронізму за $p_0 = 0.4$ та $p_0 = 0.495$ вища для методу на основі поділу синхрокомбінації на префіксну й суфіксну частини, в середньому, на 73.29% та 11,83 % відповідно, а за $p_0 = 0.496$ - вища для методу на основі кореляційної обробки, в середньому, на 56,94%; у той же час, відносна частота хибної синхронізації за $p_0 = 0.4$, $p_0 = 0.495$ та $p_0 = 0.496$ є меншою для методу на основі кореляційної обробки (0 для 10000 експериментів) порівняно з методом на основі поділу синхрокомбінації на префіксну й суфіксну частини (0.003, 0.006 та 0.008 відповідно, для 10000 експериментів).

Ключові слова: перестановка, синхрокомбінація, циклова синхронізація, короткопакетний зв'язок, криптографічний протокол, факторіальне кодування, шум, завада, імовірність синхронізації, оцінка ймовірності, кореляція, комунікації.

SUMMARY

Stupka B.A. Methods of reliable information transmission in systems with non-separable factorial data coding at high bit error probability. – Qualifying scientific work on the rights of the manuscript.

Dissertation for the degree of Doctor of Philosophy in specialty 123 – Computer Engineering. – Cherkasy State Technological University, Cherkasy, 2024.

The dissertation is aimed at solving an urgent scientific and technical problem, which is to increase the reliability of information transmission through the use of factorial data coding. This task implies the need to create methods for establishing cyclic synchronization and a method for reliable permutation transmission using non-separable factorial coding in communication channels with a high bit error probability.

The analysis of non-separable factorial data coding methods in this paper shows that they simultaneously solve the problems of cryptographic information protection and noise-resistant coding.

The methods of cyclic synchronization are analyzed. It is shown that there are methods that use a brute force approach and involve buffering data of two frame lengths and decoding for each possible synchronization offset. Other methods use an XOR operation to synchronize the data stream to increase the efficiency of the cyclic synchronization. There are also cyclic synchronization methods that do not use preamble characters and provide for adaptation of the frame format. At the same time, the considered methods of cyclic synchronization have limitations that lead to significant computational complexity of the algorithms that implement them and the impossibility of using them in communication systems with block codes.

Existing applications that implement non-separable factorial coding are considered, in particular a three-pass permutation-based cryptographic protocol that allows secure transmission of messages between two parties without the need to transmit or declare a public or private key. At the same time, the peculiarity of such protocols is that they require higher reliability indicators, since data is transmitted three times to transmit one message, which increases the probability of its being affected by interference, which is especially noticeable in conditions of high security.

The analysis made it possible to clearly formulate the tasks of the work. They consist in developing a method of cyclic synchronization for communication systems of information transmission with non-separable factorial coding, including with a bit error probability close to 0.5, as well as in developing a method of reliable information transmission in communication systems with non-separable factorial data coding with the same bit error probability.

In order to verify and study the effectiveness of the developed methods of cyclic synchronization, reliable information transmission, as well as to formulate recommendations for their application, the final task of the dissertation research is to perform a comparative experimental evaluation of the effectiveness of the developed methods.

The thesis presents the first developed method of cyclic synchronization, which, by using a permutation of numbers as a syncword, its division into prefix and suffix parts, as well as by majority processing of received fragments, allows for cyclic synchronization of the receiving and transmitting stations of communication systems for transmitting information with non-separable factorial coding. A structural diagram of the cyclic synchronization device is developed. A software model of data transmission is built, in which the developed algorithm for establishing cyclic synchronism is implemented The developed method allows for a codeword length of 8 elements (24 bits) in communication channels with a bit error probability $p_0 = 0,495$, get value of relative frequency setting the correct synchronization $W_{true} = 0,993$ and the relative frequency of false synchronization $W_{false} = 0,006$, for the parameters $p_{0\_\max} = 0.495$, for maximum accumulation rate $l_{\max} = 90603$ (2174472 bits). At the same time, the developed method makes it possible to ensure synchronization, on average, after obtaining 4, 7, 15, 54, 14526 fragments (96, 168, 360, 1296, 348624 bits) with bit error probabilities of $p_0 = 0,1; 0,2; 0,3; 0,4; 0,495$, accordingly.

The method of cyclic synchronization of an indivisible factorial code has been further developed, which, by using as a syncword a permutation that has the maximum value of the minimum Hamming distance from its binary representation to all its cyclic

shifts, and also due to correlation and majoritarian processing of data fragments transmitted by the communication channel, where the length of the fragment is equal to the length of the syncword, allows to provide cyclic synchronization with a bit error probability close to 0.5. A block diagram of the cyclic synchronization device was developed. A software model of data transmission was built, which implements the developed algorithm for establishing cyclic synchronization. The application of the developed method of cyclic synchronization made it possible to ensure the probability of correct synchronization $P_{true} \geq 0,9997$ and the possibility of false synchronization $P_{false} \leq 3 \cdot 10^{-4}$ in the model system transmission of data with independent bit errors with the probability of their occurrence $p_0 = 0.495$ for a syncword-permutation of 8 elements (24 bits) with the maximum accumulation factor $l_{\max} = 30603$ (734472 bits) and with the average number of received fragments required to establish synchronization, 15787 fragments (378888 bits).

An algorithm for mixing fragments received from the communication channel has been developed and studied, which allows to further reduce the required accumulation factor for establishing cyclic synchronism. Thus, the difference between the average value of the number of accumulated fragments for synchronization systems with and without mixing of accumulated fragments with parameters defined for the bit error probability $p_{0\_\max} = 0.495$ and the length of the syncword-permutation of 8 elements (24 bits), is equal to 4607 fragments (110 568 bits) for $p_0 = 0.495$, 9.1 fragments for $p_0 = 0.4$ i 2.6 fragments for $p_0 = 0.35$, which reduces the required time for establishing synchronization. The developed method can be effective for implementation not only in systems with non-separable factorial coding, but also in classical data transmission systems that use a standard frame separator.

For the first time, a method of reliable transmission of permutations has been developed, which, by representing each element of the transmitted permutation as a cyclic binary shift of the carrier permutation, which has the maximum value of the minimum Hamming distance from its binary representation to all its cyclic shifts, allows for

information exchange using a three-pass cryptographic protocol based on permutations with a bit error probability close to 0.5. To increase the reliability of permutation transmission, the method also uses majority and correlation processing of fragments received from the communication channel. Developed mathematical model of the data transmission system. An algorithm is developed and implemented that applies the proposed method and allows, given the bit error probability $p_0 = 0.495$ achieve the probability of accepting permutations without errors $P_{W\_true\_final} \geq 0.999$ and the probability of an undetected error $P_{W\_false\_final} \leq 3.6 \cdot 10^{-4}$. The results of the constructed simulation software model of the data transmission system confirm the effectiveness of the developed method in comparison with the traditional DSSS method: to achieve the given probability of accepting permutations without errors and the probability of an undetected error, as indicated above, the DSSS method requires accepting $l = 36413$ fragments (20099976 bits), while the developed method requires $l = 29123$ fragments (16075896 bits).

The final part of the dissertation investigates the effectiveness of methods of cyclic synchronization of information transmission systems with non-separable factorial coding: a method based on the use of a permutation of numbers with its division into prefix and suffix parts as a syncword, and a method that uses as a syncword a permutation that has the maximum value of the minimum Hamming distance from its binary representation to all its cyclic shifts. The structural diagrams of the simulation models of the data transmission system for each of the methods are developed and described. The development environment and parameters of the hardware on which the simulation was performed are described. Software simulation models of data transmission systems are built, which implement algorithms for establishing cyclic synchronism for each of the above methods. The main modules used to implement the models and to interpret the results are substantiated. Recommendations for the application of the developed methods of cyclic synchronization in communication channels with a high bit error probability using non-separable factorial coding are developed. In particular, the probability of correct synchronization is higher for the method based on the division of the syncword

into prefix and suffix parts by $p_0 \le 0.495$ and higher for the method based on correlation processing for $p_0 > 0.495$. At the same time, it should be borne in mind that the implementation of the method based on the division of the syncword into prefix and suffix parts has a higher probability of false phasing $p_0 > 0.3$ compared to the method based on correlation processing. For example, the speed of establishing cyclic synchronism for $p_0 = 0.4$ and $p_0 = 0.495$ is higher for the method based on the division of the syncword into prefix and suffix parts, on average, by 73.29% and 11.83%, respectively, and for $p_0 = 0.496$ - is higher for the method based on correlation processing, on average, by 56.94%; at the same time, the relative frequency of false synchronization for $p_0 = 0.4$, $p_0 = 0.495$ and $p_0 = 0.496$ is smaller for the method based on correlation processing (0 for 10000 experiments) compared to the method based on dividing the syncword into prefix and suffix parts (0.003, 0.006 and 0.008, respectively, for 10000 experiments).

Keywords: permutation, syncword, cyclic (frame) synchronization, short-packet communications, cryptographic protocol, factorial coding, noise, interference, synchronization probability, probability estimation, correlation, communications.

Список публікацій здобувача

[1] Е. В. Фауре, В. В. Швидкий, А. І. Щерба, О. О. Харін, і Б. А. Ступка, «Метод циклової синхронізації на основі перестановок», *Вісник черкаського державного технологічного університету*, вип. 4, с. 67–76, 2020, doi: 10.24025/2306-4412.4.2020.222439.

[2] Е. В. Фауре і Б. А. Ступка, «Імітаційне моделювання процесу встановлення циклового синхронізму в системах зв'язку з нероздільним факторіальним кодуванням», *Вісник Черкаського державного технологічного університету*, вип. 4, с. 16–24, 2021, doi: 10.24025/2306-4412.4.2021.252807.

[3] J. Al-Azzeh, E. Faure, A. Shcherba, і B. Stupka, «Permutation-based frame synchronization method for data transmission systems with short packets», *Egypt. Inform. J.*, том 23, №3, с. 529–545, 2022, doi: 10.1016/j.eij.2022.05.005. **(Scopus, Q1)**

[4] E. Faure, A. Shcherba, M. Makhynko, B. Stupka, J. Nikodem, і R. Shevchuk, «Permutation-Based Block Code for Short Packet Communication Systems», *Sensors*, том 22, №14, с. 5391, 2022, doi: 10.3390/s22145391. **(Scopus, Q1)**

[5] Е. В. Фауре і Б. А. Ступка, «Залежність ефективності кадрової синхронізації нероздільних факторіальних кодів від параметрів синхронізації», *Електронне моделювання*, том 44, № 6, с. 21–35, 2022, doi: 10.15407/emodel.44.06.021.

[6] E. Faure, A. Shcherba, B. Stupka, I. Voronenko, і A. Baikenov, «A Method for Reliable Permutation Transmission in Short-Packet Communication Systems», в Information Technology for Education, Science, and Technics, том 178, E. Faure, O. Danchenko, M. Bondarenko, Y. Tryus, C. Bazilo, і G. Zaspa, Ред., в Lecture Notes on Data Engineering and Communications Technologies, vol. 178., Cham: Springer Nature Switzerland, 2023, с. 177–195. doi: 10.1007/978-3-031-35467-0_12. **(Scopus)**

[7] E. Faure, A. Shcherba, і B. Stupka, «Permutation-Based Frame Synchronisation Method for Short Packet Communication Systems», в *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Cracow, 22-25 sempember 2021*, Cracow, Poland: IEEE, 2021, с. 1073–1077. doi: 10.1109/IDAACS53288.2021.9660996. **(Scopus).**

[8] Е.В. Фауре, А.І. Щерба, Б.А. Ступка, А.С. Байкенов, «Метод достовірного передавання перестановок у системах зв'язку з короткими пакетами», в *Інформаційні технології в освіті, науці і техніці (ІТОНТ-2022): Тези доповідей VI Міжнародної науково-практичної конференції, Черкаси, 23-25 червня 2020р.,* Черкаси: ЧДТУ, 2020, с. 70-71.

[9] Е. В. Фауре, В. В. Швидкий, О.О. Харін, А.О. Лавданський, Б.А. Ступка, «Спосіб циклової синхронізації», Україна. Пат. 148842, 22.09.2021.

[10] Е. В. Фауре, В. В. Швидкий, О.О. Харін, А.О. Лавданський, Б.А. Ступка, «Система циклової синхронізації», Україна. Пат. 148847, 22.09.2021.

[11] Е. В. Фауре, А. І. Щерба, А.О. Лавданський, Б.А. Ступка, «Спосіб циклової синхронізації», Україна. Пат. 150959, 18.05.2022.

[12] Е. В. Фауре, А. І. Щерба, А.О. Лавданський, Б.А. Ступка, «Система циклової синхронізації», Україна. Пат. 150883, 04.05.2022.

[13] Е. В. Фауре, А. І. Щерба, М.В. Махинько, Б.А. Ступка, «Спосіб прогнозування потужності нероздільного факторіального коду», Україна. Пат. 152846, 19.04.2023.

[14] Е. В. Фауре, А. І. Щерба, М.В. Махинько, Б.А. Ступка, «Спосіб побудови нероздільного факторіального коду», Україна. Пат. 152845, 19.04.2023.

[15] Е. В. Фауре, А. І. Щерба, А.О. Лавданський, К.В. Базіло, Б.А. Ступка, «Спосіб циклової синхронізації», Україна. Пат. 153803, 30.08.2023.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 18

ВСТУП 19

1. АНАЛІЗ СУЧАСНОГО СТАНУ ПРЕДМЕТНОЇ ОБЛАСТІ. ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ 29

1.1. Вступ 29

1.2. Методи поєднання завадостійкого кодування та криптографічного захисту 31

1.3. Захист інформації на основі факторіального кодування 33

1.3.1. Роздільні факторіальні коди 34

1.3.2. Нероздільні факторіальні коди 37

1.3.3. Трьохетапний криптографічний протокол на основі перестановок 41

1.4. Методи циклової синхронізації 45

1.5. Підходи до забезпечення достовірного передавання інформації в умовах великої ймовірності бітової помилки. 46

1.6. Цілі та задачі дисертаційного дослідження. 48

1.7. Висновки 50

2. МЕТОД ЦИКЛОВОЇ СИНХРОНІЗАЦІЇ СИСТЕМ ПЕРЕДАВАННЯ ДАНИХ З НЕРОЗДІЛЬНИМ ФАКТОРІАЛЬНИМ КОДУВАННЯМ НА ОСНОВІ ПОДІЛУ КОДОВОГО СЛОВА НА ПРЕФІКСНУ Й СУФІКСНУ ЧАСТИНИ 52

2.1. Вступ 52

2.2. Опис методу 52

2.3. Імовірнісні показники встановлення циклового синхронізму 58

2.4. Оцінка ефективності методу 61

2.5. Система циклової синхронізації 67

2.6. Висновки 69

3. МЕТОД ЦИКЛОВОЇ СИНХРОНІЗАЦІЇ СИСТЕМ ПЕРЕДАВАННЯ ДАНИХ З НЕРОЗДІЛЬНИМ ФАКТОРІАЛЬНИМ КОДУВАННЯМ НА ОСНОВІ КОРЕЛЯЦІЙНОЇ ОБРОБКИ 70

3.1. Вступ 70

3.2. Опис методу 71

3.2.1. Вибір синхрокомбінації 74

3.2.2. Розпізнавання синхрокомбінації 77

3.2.3. Кореляційна обробка 77

3.2.4. Імовірність бітової помилки в уточненій послідовності R 77

3.3. Імовірнісні показники встановлення циклового синхронізму 79

3.3.1. Імовірність правильної синхронізації 79

3.3.2. Імовірність хибної синхронізації 81

3.3.3. Зменшення ймовірності хибної синхронізації 84

3.3.4. Оцінки інтервальних імовірностей синхронізації 89

3.3.5. Вибір значень K та l 90

3.4. Оцінка ефективності методу 93

3.4.1. Результати та їх обговорення 94

3.4.2. Вплив процедури перемішування на отримані результати. 100

3.5. Система циклової синхронізації 103

3.6. Висновки 105

4. МЕТОД ДОСТОВІРНОГО ПЕРЕДАВАННЯ ПЕРЕСТАНОВОК У СИСТЕМАХ ЗВ’ЯЗКУ З НЕРОЗДІЛЬНИМ ФАКТОРІАЛЬНИМ КОДУВАННЯМ 107

4.1. Вступ..... 107

4.2. Опис методу 108

4.3. Імовірнісні показники розпізнавання слова 111

4.3.1. Імовірність правильного та хибного розпізнавання слова 111

4.3.2. Оцінки сумарних інтервальних імовірностей правильного й хибного розпізнавання слова 117

4.4. Оцінка ефективності методу..... 120

4.4.1. Експериментальна перевірка одержаних результатів 121

4.4.2. Обговорення отриманих результатів 125

4.5. Висновки 128

5. ЕКСПЕРИМЕНТАЛЬНА ПОРІВНЯЛЬНА ОЦІНКА ЕФЕКТИВНОСТІ РОЗРОБЛЕНИХ МЕТОДІВ ЦИКЛОВОЇ СИНХРОНІЗАЦІЇ В ЗАЛЕЖНОСТІ ВІД ЯКОСТІ КАНАЛУ ЗВ’ЯЗКУ ТА ПАРАМЕТРІВ СИНХРОНІЗАЦІЇ 130

5.1. Вступ 130

5.2. Експериментальне порівняння ефективності розроблених методів циклової синхронізації в залежності від якості каналу зв’язку 131

5.2.1. Опис основних відмінностей методів..... 132

5.2.2. Побудова та опис імітаційних моделей 134

5.2.3. Опис використаного програмного та апаратного забезпечення 137

5.2.4. Отримані результати..... 138

5.3. Оцінка ефективності циклової синхронізації нероздільних факторіальних кодів у залежності від параметрів синхронізації 141

5.3.1. Вибір параметрів синхронізації 142

5.3.2. Експериментальні результати та їх обговорення 145

5.4. Висновки 151

ВИСНОВКИ 155

СПИСОК ДЖЕРЕЛ 159

ДОДАТКИ 173

Додаток А. Лістинги розрахунково-експериментальних моделей 174

А.1. Лістинг імітаційно-програмної моделі методу циклової синхронізації на основі поділу синхрокомбінації на префіксну та суфіксну частини 174

А.2. Лістинг імітаційно-програмної моделі методу циклової синхронізації на основі кореляційної обробки 183

А.3. Лістинг імітаційно-програмної моделі методу достовірного передавання перестановки 192

Додаток Б. Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації 202

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

DSSS – direct sequence spread spectrum;

NOMA – non orthogonal multiple access;

АК – аварія каналу;

АКФ – автокореляційна функція;

КФК – комбінований факторіальний код;

КФР – кумулятивна функція розподілу;

ПЗ – пошук синхронізму завершений;

ПЗП – постійний запам’ятовувальний пристрій;

ПФК – повний факторіальний код;

СПД – система передавання даних;

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 проріджением;

ФСЧ – факторіальна система числення.

ВСТУП

Актуальність теми дослідження. У нинішній час інформаційно-комунікаційні технології та системи стали невід'ємною частиною повсякденного буття. Розвиток цифрових технологій і систем є безпосереднім каталізатором для змін і розвитку інших сучасних технологій і систем. У результаті, темпи зростання об'ємів інформації, яку потрібно зберігати, обробляти, передавати та отримувати в електронному вигляді, в тому числі конфіденційну, постійно прогресують. Усе це є наслідком критичної важливості цих технологій і систем майже в усіх сферах діяльності сучасного суспільства. Як результат, це породжує необхідність досліджень і розробок нових, покращених методів обробки, захисту та передавання інформації в комп'ютерних системах і мережах.

Процедура циклової синхронізації [1], [2], [3] є обов'язковим компонентом усіх мережевих протоколів [4], [5], [6], [7], [8], [9]. Виявлення меж кадрів на приймачі може бути складним через низьке співвідношення сигнал/шум, яке може бути спричинено радіоелектронною боротьбою чи затуханням сигналу.

Під час встановлення з'єднання до процедури перенесення призначених для користувача даних процедура синхронізації реалізує пошук меж синхрокомбінації (роздільника). Одним з основних параметрів, за яким оцінюється якість системи синхронізації, є час входження в синхронізм. Розвиток комунікаційних систем приводить до необхідності постійного зменшення часу входження в синхронізм як засобу зменшення витрат часу на процедуру встановлення з'єднання в загальному часі сеансу зв'язку. Ця обставина стимулює постійні пошуки нових технічних рішень, спрямованих на досягнення цієї мети.

Крім того, деякі додатки [10], [11], [12], [13], [14] потребують більш високих показників достовірності. До таких додатків можуть бути віднесені протоколи передавання даних в умовах високого рівня завад (low SNR – Signal-to-noise ratio) [15], [16], а також трьохетапні криптографічні протоколи [17], [18], [19], [20], зокрема, трьохетапний криптографічний протокол на основі перестановок [21]. У

трьохетапних протоколах передавання одного повідомлення дані передають тричі, що збільшує ймовірність їх ураження завадою.

Варто зазначити, що і використання ефективних методів циклової синхронізації є одним із варіантів забезпечення підвищення достовірності передавання даних в каналах зв'язку з високою ймовірністю бітової помилки. Ефективна система циклової синхронізації, яка може виявляти межі кадру з високою частотою бітових помилок, забезпечує зв'язок за високих рівнів природного або штучного шуму. Крім того, в таких системах зв'язку об'єднання захисту від помилок каналу та захисту від несанкціонованого доступу в єдину структуру даних може реалізувати конфіденційний обмін інформацією.

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 [22], F.J. MacWilliams та N.J.A. Sloane [23], О.А. Борисенко [24], [25], [26], [27], І.Д. Горбенко [28], [29], В.І. Грабчак [30], [31], [32], О.О. Кузнецов [33], [34], [35], Е.Л. Онанченко [36], [37], О.П. Стахов [38], [39], [40], [41], В.Я. Чечельницький [42], [43], [44], [45]. Питанням пошуку циклового синхронізму в комунікаційних системах присвячено праці J. Hamkins [46], J. Massey [2], R. Scholtz [3], P. Kartaschoff [7], Hansheng Wang [8] та інших.

У цій дисертаційній роботі загальновідомі теоретичні основи та методи інтегрованого захисту інформації на основі факторіального кодування даних [47], [48] використовуються для створення нових методів підвищення достовірності передавання інформації в системах передавання даних з короткими пакетами. Актуальність цього дослідження, зокрема, зумовлена необхідністю інформаційної взаємодії між об'єктами машинної комунікаційної мережі з динамічно змінюваною структурою та унікальною системою команд або сповіщень для кожного об'єкта мережі.

Серед існуючих на сьогодні підходів щодо вирішення завдання поєднання завадостійкого кодування та криптографічного захисту використання перестановок як формату запису числа в факторіальній системі числення (ФСЧ) є відносно новим

і найменш поширеним, але перспективним підходом [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62], що дозволяє досягти розумний компроміс між достовірністю передавання даних, криптографічною стійкістю, відносною швидкістю передавання й швидкістю коду.

Водночас, питання використання факторіальних кодів у спеціалізованих комп'ютерних і комунікаційних системах, у яких є необхідність забезпечення заданого рівня достовірності передавання даних в умовах високої інтенсивності завад у каналі зв'язку, є недостатньо дослідженим.

Існуючі ж методи достовірного передавання даних якщо і дозволяють достовірно передавати дані за високої інтенсивності бітової помилки в каналі зв'язку, то з певними апаратними обмеженнями, що залежать від потужності приймача (передавача) та призводять до збільшення енерговитрат на процедуру передавання інформації. Разом з тим, використання нероздільного факторіального кодування, який, у свою чергу, є надлишковим і стійким до завад [63], може дозволити підвищити показник достовірності передавання даних у каналах зв'язку з високою ймовірністю бітової помилки. Такий результат може бути досягнутий за рахунок розроблення методів циклової синхронізації та методу достовірного передавання перестановки-носія.

У працях [63], [64] досліджено властивості нероздільного факторіального коду виявляти та виправляти помилки каналу зв'язку. Доведено ефективність коду, в тому числі й за рахунок його властивостей синхронізації [65], [66]. Разом з тим, розглянуті можливості нероздільного факторіального кодування не дозволяють їм бути використаними в системах передавання даних з високим рівнем завад і ймовірністю бітової помилки близькою до 0.5.

Завадостійкість нероздільного факторіального кодування інформації може бути підвищена шляхом введення додаткової надлишковості. Використання з цією метою існуючих завадостійких кодів [67], [68], [69], [70] може дати позитивний ефект, проте їх застосування обмежене для ймовірності бітової помилки близької до 0.5. У такій ситуації варто врахувати, що сам нероздільний факторіальний код є надлишковим і стійким до завад [63].

У цьому дисертаційному дослідженні вирішується важлива науково-технічна задача, що полягає в забезпеченні достовірності передавання інформації на основі використання нероздільного факторіального кодування даних. Ця задача передбачає необхідність створення методів встановлення циклової синхронізації та методу достовірного передавання перестановки з використанням нероздільного факторіального кодування в каналах зв'язку з високою ймовірністю бітової помилки.

Виходячи з цього, тема дисертаційної роботи «Методи достовірного передавання інформації в системах з нероздільним факторіальним кодуванням даних за високої ймовірності бітової помилки» 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 «Інформаційно-комунікаційні та радіоелектронні системи та технології, засоби радіоелектронної боротьби для забезпечення національної безпеки і оборони. Інформаційна безпека та кібербезпека» і виконувалися відповідно до програм і планів науково-дослідних робіт Черкаського державного технологічного університету, в тому числі в рамках держбюджетної науково-технічної (експериментальної) розробки молодих вчених «Розробка методів, протоколів і засобів захищеного інформаційного обміну з використанням трьохетапного криптографічного протоколу на основі перестановок в умовах зашумленості каналів зв'язку» (номер державної реєстрації 0123U100270), в якій автор був виконавцем.

Мета роботи полягає в забезпеченні достовірності передавання інформації в системах з нероздільним факторіальним кодуванням даних за високої ймовірності бітової помилки шляхом розробки методів циклової синхронізації та методу достовірного передавання кодових слів нероздільного факторіального коду.

Для досягнення поставленої мети вирішуються наступні завдання:

• розробити метод циклової синхронізації для комунікаційних систем передавання інформації з нероздільним факторіальним кодуванням;

•розробити метод циклової синхронізації для комунікаційних систем передавання інформації з нероздільним факторіальним кодуванням за ймовірності бітової помилки, близької до 0.5;

• розробити метод достовірного передавання інформації в системах зв'язку з нероздільним факторіальним кодуванням даних за ймовірності бітової помилки, близької до 0.5;

•провести порівняльні експериментальні оцінки розроблених методів циклової синхронізації, достовірного передавання інформації, сформувати рекомендації щодо їх застосування.

Об'єктом дослідження є процеси достовірного передавання інформації у комп'ютерних і комунікаційних системах і мережах.

Предметом дослідження є методи та засоби забезпечення циклової синхронізації та достовірного передавання інформації в системах з нероздільним факторіальним кодуванням даних.

Методи досліджень. Для вирішення завдання розробки методів циклової синхронізації, методу достовірного передавання інформації в системах зв'язку з нероздільним факторіальним кодуванням даних використано методи: теорії систем передавання даних, теорії завадостійкого кодування, теорії ймовірностей і математичної статистики, статистичного аналізу, комбінаторики, функційного та об'єктно-орієнтованого програмування. Для порівняльної експериментальної оцінки розроблених методів використано методи: імітаційного моделювання, теорії ймовірностей і математичної статистики, статистичного аналізу, комбінаторики, функційного та об'єктно-орієнтованого програмування.

Наукова новизна отриманих результатів:

- ***вперше розроблено*** метод циклової синхронізації нероздільних факторіальних кодів, який за рахунок використання як синхрокомбінації перестановки чисел, її поділу на префіксну та суфіксну частини, а також за рахунок мажоритарної обробки прийнятих фрагментів, де довжина фрагмента дорівнює

довжині синхрокомбінації, дозволяє забезпечити циклову синхронізацію приймальної та передавальної станцій системи інформаційного обміну;

- ***набув подальшого розвитку*** метод циклової синхронізації нероздільних факторіальних кодів, який за рахунок використання як синхрокомбінації перестановки чисел, яка має максимальне значення мінімальної відстані Хеммінга від її двійкового представлення до всіх її циклічних зсувів, а також за рахунок кореляційної та мажоритарної обробки прийнятих фрагментів, де довжина фрагмента дорівнює довжині синхрокомбінації, дозволяє підвищити ймовірність правильної синхронізації та зменшити ймовірність хибної синхронізації за ймовірності бітової помилки, близької до 0.5;

- ***вперше розроблено*** метод достовірного передавання перестановок, який за рахунок подання кожного елементу перестановки у вигляді циклічного двійкового зсуву перестановки-переносника, що має максимальне значення мінімальної відстані Хеммінга від її двійкового представлення до всіх її циклічних зсувів, а також за рахунок кореляційної та мажоритарної обробки прийнятих фрагментів, де довжина фрагмента дорівнює довжині елементу перестановки, дозволяє забезпечити достовірний інформаційний обмін за ймовірності бітової помилки, близької до 0.5.

Практичне значення отриманих результатів

- розроблено алгоритм циклової синхронізації для систем передавання даних з нероздільним факторіальним кодуванням. Його застосування в модельному прикладі системи передавання даних для довжини кодового слова в 8 елементів (24 біти) з незалежними бітовими помилками з імовірністю їх появи $p_0 = 0,495$, дозволило отримати значення відносної частоти встановлення правильного синхронізму $W_{true} = 0,993$ та відносної частоти хибної синхронізації $W_{false} = 0,006$ для максимального коефіцієнту накопичення $l_{\max} = 90603$ (2174472 біт). У той же час, розроблений алгоритм дає змогу забезпечити встановлення синхронізму, в середньому, після отримання 4, 7, 15, 54, 14526 фрагментів (96, 168, 360, 1296, 348624 біт) за ймовірностей бітової помилки $p_0 = 0,1; 0,2; 0,3; 0,4; 0,495$, відповідно;

- розроблено алгоритм циклової синхронізації для систем передавання даних з нероздільним факторіальним кодуванням ймовірності бітової помилки, близької до 0.5. Його застосування в модельному прикладі системи передавання даних з незалежними бітовими помилками з імовірністю їх появи $p_0 = 0.495$ дозволило забезпечити встановлення правильного синхронізму для синхрокомбінації-перестановки з 8 елементів (24 бітів) за максимального коефіцієнту накопичення $l_{\max} = 30603$ (734472 біт); середня кількість прийнятих фрагментів, що необхідна для встановлення синхронізації, склала 15787 фрагментів (378888 біт) для 10000 експериментів;

- розроблено алгоритм реалізації циклової синхронізації для систем передавання даних з нероздільним факторіальним кодуванням на основі кореляційної обробки, який за рахунок процедури перемішування отриманих з каналу зв'язку фрагментів дозволив зменшити кількість необхідних для встановлення циклового синхронізму даних і, як наслідок, зменшити час на пошук меж кодових слів. Наприклад, різниця між середнім значенням кількості накопичених фрагментів для систем синхронізації з перемішуванням і без перемішування накопичених фрагментів з параметрами, визначеними для ймовірності бітової помилки $p_{0\_\max} = 0.495$ та довжини синхрокомбінації-перестановки 8 елементів (24 біта), дорівнює 4607 фрагментам (110 568 бітам) за $p_0 = 0.495$, 9.1 фрагментам за $p_0 = 0.4$ і 2.6 фрагментам за $p_0 = 0.35$, що зменшує необхідний час встановлення синхронізму. Розроблений метод може бути ефективним для реалізації не тільки в системах з нероздільним факторіальним кодуванням, а й у класичних системах передавання даних, де використовується стандартний роздільник між кадрами;

- розроблено алгоритм реалізації достовірного передавання перестановок каналами зв'язку з імовірністю бітової помилки, близькою до 0.5. Реалізація алгоритму дає змогу за ймовірності бітової помилки $p_0 = 0.495$ досягти ймовірності приймання перестановок без помилок $P_{W\_true\_final} \geq 0.999$ і ймовірність

невиявленої помилки $P_{W\_false\_final} \le 3.6 \cdot 10^{-4}$. Для досягнення заданої ймовірності приймання перестановок без помилок і ймовірності невиявленої помилки, що вказані вище, метод DSSS потребує приймання $l = 36413$ фрагментів (20099976 біт), в той час, як розроблений метод, потребує $l = 29123$ фрагментів (16075896 біт);

- розроблено рекомендації щодо застосування розроблених методів циклової синхронізації в каналах зв'язку з високою ймовірністю бітової помилки з використанням нероздільного факторіального кодування. Зокрема, продемонстровано, що за параметрів циклової синхронізації, визначених для $p_{0\_\max} = 0,495$, імовірність правильної синхронізації є вищою для методу на основі поділу синхрокомбінації на префіксну й суфіксну частини за $p_0 \le 0.495$ і вищою для методу на основі кореляційної обробки за $p_0 > 0.495$. Разом з тим, варто враховувати, що реалізація методу на основі поділу синхрокомбінації на префіксну й суфіксну частини має вищі значення ймовірності хибного фазування за $p_0 > 0.3$ у порівняння з методом на основі кореляційної обробки. Для прикладу, швидкість встановлення циклового синхронізму за $p_0 = 0.4$ та $p_0 = 0.495$ вища для методу на основі поділу синхрокомбінації на префіксну й суфіксну частини, в середньому, на 73.29% та 11,83 % відповідно, а за $p_0 = 0.496$ - вища для методу на основі кореляційної обробки, в середньому, на 56,94%; у той же час, відносна частота хибної синхронізації за $p_0 = 0.4$, $p_0 = 0.495$ та $p_0 = 0.496$ є меншою для методу на основі кореляційної обробки (0 для 10000 експериментів) порівняно з методом на основі поділу синхрокомбінації на префіксну й суфіксну частини (0.003, 0.006 та 0.008 відповідно, для 10000 експериментів).

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: [1], [9], [10] – розроблено та досліджено метод циклової синхронізації на основі поділу синхрокомбінації на

префіксну й суфіксну частини для систем з нероздільним факторіальним кодуванням в умовах впливу в каналі зв'язку завад високої інтенсивності; [2] – досліджено ефективність методів циклової синхронізації систем передавання інформації з нероздільним факторіальним кодуванням; [3] – реалізовано та досліджено алгоритм перемішування отриманих з каналу зв'язку фрагментів, для алгоритмів циклової синхронізації в системах передачі даних із короткими пакетами, зокрема тих, які використовують нероздільне факторіальне кодування; [4] – досліджено завадостійкість нероздільного факторіального коду; [5] – реалізовано алгоритм циклової синхронізації нероздільних факторіальних кодів, застосовано операції перемішування отриманих з каналу зв'язку фрагментів для підвищення ефективності циклової синхронізації; [6], [8] – розроблено метод достовірного передавання перестановки в каналах зв'язку з імовірністю бітової помилки, близькою до 0.5; [7], [11], [12] – розроблено та досліджено метод циклової синхронізації на основі кореляційної обробки для систем з нероздільним факторіальним кодуванням в умовах впливу в каналі зв'язку завад високої інтенсивності; [13], [14] – досліджено потужність нероздільного факторіального коду; [15] – розроблено алгоритм циклової синхронізації з використанням процедури перемішування для систем передавання інформації з нероздільним факторіальним кодуванням. 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:

- 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS 2021) (Poland, 22-25 September 2021).
- VI Міжнародній науково-практичній конференції «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2022), (Черкаси, 23-25 червня 2022р.).

Публікації. Результати дослідження опубліковано в 15 наукових працях, у

тому числі в трьох статтях у періодичних наукових виданнях, що входять до наукометричної бази даних Scopus [3], [4], [6], трьох статтях у наукових виданнях, включених до Переліку наукових фахових видань України та інших наукометричних баз даних [1], [2], [5], двох доповідях на науково-практичних конференціях [7], [8] і семи патентах України на корисні моделі [9], [10], [11], [12], [13], [14], [15].

Структура та обсяг дисертаційної роботи. Дисертаційна робота складається з вступу, п’яти розділів, висновків, списку використаних джерел і додатків. Повний об’єм дисертації складає 204 сторінки. Робота містить 12 таблиць і 46 рисунків.

1. АНАЛІЗ СУЧАСНОГО СТАНУ ПРЕДМЕТНОЇ ОБЛАСТІ. ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1. Вступ

У останні десятиліття розвиток інформаційних технологій і систем відбувається надшвидкими темпами: розвиток штучного інтелекту, біомедицини, різногалузевих онлайн платформ з різним функціоналом тощо. Як наслідок, обсяги інформації, в тому числі конфіденційної, яку потрібно передавати, постійно зростають. Згідно зі звітом дослідницького департаменту Statista [71], протягом наступних років, до 2025 року, прогнозується, що глобальне створення даних зросте до більш ніж 180 зеттабайт. Зберігання, обробку та передавання інформації, забезпечують комп'ютерні та комунікаційні системи та мережі. Забезпечення захисту від несанкціонованого доступу та модифікації інформації, що зберігається, обробляється та передається в системі, є її необхідною властивістю. Окрім того, забезпечення захисту інформації від впливу завад у каналі зв'язку є не менш важливою функцією комунікаційної системи.

Для вирішення завдань забезпечення безпеки інформації і її достовірного передавання варто враховувати наступні особливості:

- наявність великих масивів інформації, так звані BigData, що зберігаються в базах даних або переміщаються мережею;
- доступність сторонніх осіб до даних, які знаходяться в мережі;
- широке використання бездротових каналів зв'язку, для яких характерним є присутність завад і взаємний вплив сигналів користувачів один на одного;
- робота на завеликих відстанях між абонентами інформаційного обміну, несанкціоноване зчитування переданих даних та навмисний штучний вплив на канали зв'язку для придушення обміну або ж нав'язування недостовірних даних (радіоелектронна боротьба та радіоелектронна розвідка).

Враховуючи вище сказане, забезпечення достовірного передавання інформації є однією з основних вимог, що пред'являються до комунікаційних

систем.

Одним з можливих шляхів забезпечення одночасного захисту інформації від несанкціонованого доступу або модифікації, а також забезпечення її достовірного передавання є використання факторіального кодування даних [49], [51], [52], [55], [57], [58], [59], [62], [63], [64], [72], [73], [74], [75], [76], [77].

Методологія факторіального кодування передбачає створення та дослідження методів об'єднання в єдиній процедурі (далі – інтегрований захист) таких видів захисту:

- захист від помилок, викликаних шумами в каналі зв'язку;
- захист від несанкціонованої модифікації даних;
- захист від несанкціонованого доступу до даних.

Як показано в [47], [48], окреме застосування декількох зазначених видів захисту інформації призводить до підвищення вимог до швидкодії комп’ютерних компонентів та до зростання введеної надлишковості і, в результаті, до зменшення пропускної здатності каналу зв’язку. Об’єднання завадостійкого кодування та функцій криптографії в єдину процедуру (інтегрований захист згідно з [47]) є одним з варіантів вирішенням проблеми підвищення ефективності комунікаційних систем і мереж.

Як результат, використання факторіального кодування, яке передбачає використання перестановок як носія інформації, повністю перекриває спектр основних вимог, щодо забезпечення достовірного передавання даних в комунікаційних системах.

З метою постановки задач дисертаційного дослідження, у першому розділі необхідно:

- визначити основні переваги факторіального кодування в порівнянні з іншими існуючими методами поєднання функцій завадостійкого кодування та криптографічного захисту інформації;
- виконати порівняльний аналіз нероздільного та роздільного факторіального кодування;
- виконати аналіз основних існуючих методів циклової синхронізації для

комунікаційних систем;

– виконати аналіз існуючих методів достовірного передавання даних для комунікаційних систем з нероздільним факторіальним кодуванням.

1.2. Методи поєднання завадостійкого кодування та криптографічного захисту

Під час передавання даних незахищеними каналами зв’язку (каналами колективного користування) виникає необхідність у вирішенні ряду таких задач із забезпечення захисту інформації, як забезпечення конфіденційності, достовірності, цілісності даних, виявлення та виправлення помилок каналу зв’язку. Окреме вирішення цих задач пов'язане із застосуванням різних математичних методів та алгоритмів, а також послідовною обробкою інформації, що призводить до збільшення навантаження на перетворювачі інформації та підвищення вимог до їх швидкодії, збільшення введеної надмірності та, як наслідок, зменшення відносної швидкості передачі.. Як наслідок, розгляд технічних та організаційних заходів щодо запобігання шкоди процесам управління, є виправданими для захисту інформації під час її зберігання або передавання.

Серед робіт, які поклали початок розвитку щодо розробок та досліджень методів кодування інформації, в тому числі і завадостійкого, варто виділити роботу К. Шеннона [78]. Орім того, робота К. Шеннона [79] належить до основ сучасної криптографії.

У стандартних протоколах HDLC, X.25/2 (LAP-B, LAP-M), SLIP, PPP для завадостійкого кодування застосовують циклічні коди з виявленням помилок. У радіоканалах успішно використовують коди з виправленням помилок Ріда-Соломона [67], [69], [80] та Боуза-Чоудхурі-Хоквінгема. У супутникових системах зв’язку знаходять застосування для згорткових кодів.

Для забезпечення імітозахисту і підтвердження достовірності інформації сучасні методи криптографії потребують введення додаткової надлишковості. Зокрема, до повідомлення додається імітовставка, вироблена на основі секретного

ключа, для захисту від нав’язування хибних даних. Електронний цифровий підпис використовують як для підтвердження достовірності інформації та її авторства, так і для захисту від нав’язування хибної інформації. Найвідоміші сучасні алгоритми: RSA [81]; Ель-Гамаля [82]; алгоритми на еліптичних кривих [83], [84].

Способи, що забезпечують виявлення факту зміни переданої інформації після навмисних дій зловмисника, або ж внаслідок впливу завад у каналі зв’язку [85], [86], [87], під час збільшення довжини блоку та кількості контрольних ознак або ж за необхідності виконання обчислень у класах лишків за модулем, відмінним від степеня числа два, мають невисоку швидкість роботи алгоритму.

Праці [47], [48] демонструють особливості існуючих методів поєднання функцій завадостійкого кодування та криптографічного захисту інформації, а також властиві їм недоліки:

- запропонований McEliece R.J. [22] метод криптографії з відкритим ключем на основі кодів, що виправляють помилки, володіє низькою швидкість коду, має потребу в дуже великій довжині ключа, а також має слабку стійкість до зламу за повторного використання ключа;
- стохастичний метод інтегрального захисту інформації [48], що включає каскадне виконання операцій завадостійкого кодування та шифрувального стохастичного перетворення, не вирішує проблему поєднання функцій завадостійкого кодування та криптографічного захисту інформації в єдину процедуру, а також передбачає використання гамування, що не завжди є доцільним в реальних системах передавання даних;
- метод “золотої” криптографії, запропонований у роботах Стахова О.П. [38], [39], [40], [41], для забезпечення інтегрованого захисту вимагає подальших досліджень для визначення параметрів та характеристику коду;
- методи захисту інформації, представлені в працях Чечельницького В.Я. [43], [44], [45], пропонують використовувати параметричну прихованість системи зв'язку з шумоподібними сигналами (ШПС) у поєднанні з завадостійким кодуванням. Разом з цим, ці методи призначено для систем зв’язку з ШПС і не адаптовані для використання в комунікаційних системах загального призначення.

1.3. Захист інформації на основі факторіального кодування

Згідно з [48], перестановкою символів скінченної множини A називається бієктивна функція $\pi : A \to A$. Загалом існує $M!$ різних перестановок, де M – потужність множини A.

Застосування факторіального кодування для вирішення задач завадостійкого кодування, захисту від несанкціонованого доступу та нав'язування хибних даних обумовлено їх наступними властивостями [47], [48]:

- 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 $k \le \lfloor \log_2 M! \rfloor$, де k – кількість біт у інформаційній послідовності;
- сума символів у перестановці завжди постійна.

Відповідно до дослідження, що проведене в роботі [49], та зроблених висновків у роботі [48] основними властивостями факторіального кодування є:

1. виявлення всіх помилок, що трансформують перестановку в неперестановку;

2. приховування закону відображення інформаційного повідомлення $A(x)$ в перестановку $\pi(x) : A(x) \Rightarrow \pi(x)$ забезпечує захист від несанкціонованого доступу, а додавання перестановки до повідомлення забезпечує його імітозахист;

3. незмінність суми елементів перестановки забезпечує можливість самосинхронізації факторіальних кодів.

Варто зазначити наступні особливості реалізації методів факторіального кодування, які описані в [48]:

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 зі збільшенням потужності множини символів перестановки M.

Факторіальні коди базуються на працях Борисенка О.А. [24], [25], [26], [27]. У них уперше викладно принципи передавання інформації перестановками. Праці [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61], [62] містять результати використання факторіальних кодів з метою забезпечення інтегрованого захисту інформації.

Як зазначено в [47], [48], перетворення слова джерела в перестановку здійснюється за рахунок відображення точок на числовій осі, що відповідають інформаційному вектору $A(x)$, який є двійковим представленням слова джерела довжиною k біт, у точки числової осі, що відповідають номерам перестановок $\pi(x)$. Перестановку представляють послідовністю всіх символів $\{0,1,2,...,M-1\}$.

Згідно з [47], у залежності від наявності в кодовому слові окремих інформаційної та перевірної частин, факторіальне кодування ділять на роздільне та нероздільне.

1.3.1. Роздільне факторіальне кодування

Спільною характеристикою всіх роздільних факторіальних кодів, згідно з [48], є використання перестановок символів з множини $\{0,1,2,...,M-1\}$ як перевірну частину, сформовану на основі інформаційної частини. У залежності від необхідного ступеня забезпечення підвищення достовірності передавання даних і методу кодування обирають потужність множини M.

Роздільні факторіальні коди:

- повний факторіальний код (ПФК) [49], [50];
- комбінований факторіальний код (КФК) [51];

– факторіальний код з проріджениям (ФКП) [52], [53];
– роздільний факторіальний код з декількома контрольними сумами (ФКДКСр) [52], [53];

ПФК забезпечує формування перевірної частини кодового слова у вигляді перестановки чисел довжини M. Перевірна частина визначається інформаційною частиною та алгоритмом кодування.

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

1) контрольною сумою є перестановка довжини 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.

Загалом, метод ПФК, дає змогу ефективно знаходити компроміс між достовірністю передавання та швидкістю коду за рахунок простоти в реалізації та налаштування довжини контрольної суми.

КФК передбачає формування контрольної суми CRC-коду, обчисленої за перестановкою довжини M, сформованою за інформаційною частиною. Таким чином, КФК є модифікацією ПФК. КФК хоч і забезпечує більшу достовірність передавання даних та швидкість коду, але вимагає більших апаратних затрат на його реалізацію [48].

Запропонований у [51] метод передбачає наступне:

1) образ інформаційної частини блоку даних формується у вигляді однієї перестановки довжини M 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, правило перестановки тримається в таємниці.

Відповідно до [48], ФКП, на відміну від ПФК і КФК, для обчислення перевірної частини кодового слова використовує лише частину інформаційних символів $A(x)$, що дозволяє скоротити час формування кодового слова та обсяг необхідних для цього апаратних ресурсів. Разом з тим, це робить ФКП менш стійким до впливу завад.

Метод ФКП, відповідно до [52], полягає в наступному:

1) контрольна сума ФКП є перестановкою довжиною M;

2) відповідно до секретного ключа, який є елементом ключа перетворення, з інформаційної частини кодового слова обирають k біт;

3) за обраними k бітами формують контрольну суму – перестановку відповідно до принципів ПФК;

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. Правило перестановки тримають у таємниці, що є частиною ключа перетворення.

ФКДКСр передбачає комбінування декількох перевірних частин ФКП.

Метод ФКДКСр описано в роботі [52]:

1) контрольна сума ФКДКСр є конкатенацією N перестановок з довжинами $M(i)$ елементів $(1 \le i \le N)$;

2) кожну перестановку формують 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. Правило перестановки тримають у таємниці, що є частиною ключа перетворення.

Таким чином, особливістю роздільних факторіальних кодів є те, що в кодовому слові роздільних факторіальних кодів міститься інформаційна частина, яка не захищена від несанкціонованого читання. Крім того, ці коди забезпечують низьку пропускну здатність через виправлення помилок лише за рахунок повторного запиту блоку даних.

1.3.2. Нероздільне факторіальне кодування

Спільною характеристикою всіх нероздільних факторіальних кодів є повна заміна інформаційного вектора $A(x)$ на перестановку елементів $\pi_n(x)$ з множини $\{0,1,...,M-1\}$, що обирається за умови $M! \ge 2^k$, $k = \lfloor \log_2 M! \rfloor$ для забезпечення відповідності кожному слову джерела лише однієї перестановки [48].

До нероздільних факторіальних кодів відносять [48]:

- факторіальний код з відновленням даних за перестановкою (ФКВД) [54], [55], [56];

- факторіальний код з відновленням даних за перестановкою з доповненням (ФКВДд) [55], [57];

- нероздільний факторіальний код з декількома контрольними сумами (ФКДКСн) [47], [52], [53];

- факторіальний код з відновленням даних за перестановкою з заданим числом інверсій (ФКЗЧІ) [58];

- факторіальний код з відновленням даних і виправленням помилок (ФКВДвп) [59], [60], [61], [62].

Відповідно до [54], факторіальним кодом з відновленням даних за перестановкою називається нероздільний код, який передбачає заміну інформаційної послідовності з k біт на перестановку чисел довжини M, $(M! \geq 2^k)$ обчислену за всіма k інформаційними бітами.

Описаний у [54] метод ФКВД передбачає наступне:

1) інформаційна послідовність $A(x)$ з k біт перетворюється в перестановку π з потужністю множини елементів $M : M! \geq 2^k$, 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 бути ключем 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 [55], [57] є модифікацією ФКВД і передбачає введення в інформаційну частину додаткових перевірних біт. Це дозволяє підвищити достовірність передавання даних.

Згідно з [57], ФКВДд полягає в 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 [47] 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 у [47] метод ФКДКСн полягає у наступному:

1) кодове слово ФКДКСн є конкатенацією N перестановок довжини $M(i)$ елементів $(1 \le i \le N)$;

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 [58], 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 довжини 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 довжини M належить класу $B_M(q,R)=\{\pi\,|\,inv(\pi)|_q=R\}$ 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.

Праці [59], [60], [61], [62] показують, що метод ФКВДвп забезпечує можливість виправлення помилок та має значний інтерес для подальших досліджень.

Згідно з [60], ФКВДвп називається нероздільний код, який передбачає заміну інформаційної послідовності з k біт на перестановку чисел довжини $M(M!\geq 2^k)$, обчислену за всіма інформаційними бітами таким чином, що відстань між кодовими словами є достатньою для виправлення виникаючих у каналі зв'язку помилок.

Метод, описаний у [60], передбачає наступну послідовність дій:

1) інформаційна послідовність $A(x)$ з k біт перетворюють у перестановку π довжини 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 сигнально-кодової конструкції;

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.

У результаті аналізу методів нероздільного факторіального кодування визначено, що вони одночасно забезпечують криптографічний захист інформації та реалізують завадостійке кодування. Водночас, актуальною задачею залишається забезпечення достовірності передавання інформації для існуючих методів нероздільного факторіального кодування в умовах завад високої інтенсивності в каналі зв'язку.

1.3.3. Трьохетапний криптографічний протокол на основі перестановок

Трьохетапний криптографічний протокол дає змогу захищеним шляхом передавати повідомлення між двома сторонами без необхідності передання або оголошення відкритого чи закритого ключа.

Уперше трьохетапний протокол був запропонований Аді Шаміром у 1980-ті роки, але не був опублікований. Базова концепція протоколу полягає в тому, що кожна зі сторін передавання має власний ключ для шифрування і ключ для розшифрування. Кожна сторона використовує свої ключі незалежно, спочатку для шифрування повідомлення, а потім для його розшифрування.

Протокол використовує функцію шифрування E і функцію розшифрування D. Функції шифрування та розшифрування можуть як збігатися, так і відрізнятися. Функція шифрування використовує ключ шифрування e, щоб перетворити відкрите повідомлення m в шифртекст $E(e,m)$. Для кожного ключа шифрування e є відповідний ключ розшифрування d, який дає змогу відновити вихідний текст за допомогою функції розшифрування $D\left(d,E\left(e,m\right)\right)$.

Для того, щоб функції шифрування E і розшифрування D підходили для трьохетапного протоколу, для будь-якого повідомлення m, будь-якого ключа шифрування e з відповідним йому ключем розшифрування d і будь-якого незалежного ключа шифрування k має виконуватися $D\left(d,E\left(k,E\left(e,m\right)\right)\right)=E\left(k,m\right)$. Іншими словами, має розшифровуватися перше шифрування з ключем e, навіть якщо повідомлення зашифровано другим ключем k. Таку властивість має

комутативне шифрування, де $E\left(a,E\left(b,m\right)\right)=E\left(b,E\left(a,m\right)\right)$ для будь-яких ключів a та b для всіх повідомлень m. Для комутативного шифрування виконується $D\left(d,E\left(k,E\left(e,m\right)\right)\right)=D\left(d,E\left(e,E\left(k,m\right)\right)\right)=E\left(k,m\right)$.

Для прикладу, наведемо процес обміну повідомленням двох учасників мережі. Припустимо, Аліса хоче надіслати Бобу повідомлення. Тоді трьохетапний протокол працює наступним чином [17]:

1) Аліса вибирає закритий ключ шифрування s і відповідний ключ розшифрування t. Аліса шифрує вихідне повідомлення m за допомогою ключа s і відправляє шифртекст $E\left(s,m\right)$ Бобу;

2) Боб вибирає закритий ключ шифрування r і відповідний ключ розшифрування q, а потім повторно шифрує перше повідомлення $E\left(s,m\right)$ за допомогою ключа r і надсилає двічі зашифроване повідомлення $E\left(r,E\left(s,m\right)\right)$ назад до Аліси;

3) Аліса розшифровує друге повідомлення за допомогою ключа t. Через комутативність, описану вище, $D\left(t,E\left(r,E\left(s,m\right)\right)\right)=E\left(r,m\right)$, тобто формується повідомлення, зашифроване тільки закритим ключем Боба. Аліса пересилає цей шифртекст Бобу;

4) Боб розшифровує третє повідомлення за допомогою ключа q і отримує $D\left(q,E\left(r,m\right)\right)=m$ вихідне повідомлення.

Варто зауважити, що всі операції з використанням закритих ключів Аліси s і t здійснюються Алісою, а всі операції з використанням закритих ключів Боба r і q здійснюються Бобом, тобто одній стороні обміну не потрібно знати ключі іншої.

Трьохетапний протокол Шаміра [88], розроблений у 1980-х роках, використовує піднесення до степеня за модулем великого простого числа як функцію і шифрування, і розшифрування, тобто $E\left(e,m\right)=m^e \bmod p$ і $D\left(d,m\right)=m^d \bmod p$, де p – велике просте число [89]. Для будь-якого шифрування

показник ступеня e знаходиться у відрізку $[1,\ldots,p-1]$ і для нього справедливо $\gcd(e,p-1)=1$. Відповідний показник для розшифрування d вибирається так, щоб $de \bmod (p-1)=1$. З малої теореми Ферма випливає, що $D(d,E(e,m))=m^{de} \bmod p=m$.

Протокол Шаміра має комутативність, оскільки $E(a,E(b,m))=m^{ab} \bmod p=m^{ba} \bmod p=E(b,E(a,m))$.

Існує безліч реалізацій протоколу Шаміра з різними методами шифрування.

Зокрема, цей протокол може бути використаний для безпечного обміну зображеннями. Так, в роботі [90] автори пропонують схему безпечного обміну зображеннями без обміну ключами (no-key-exchange secure image sharing scheme). Ця схема використовує багатопараметричне дробове перетворення Фур'є (Multiple-Parameter Fractional Fourier Transform) як криптографічний алгоритм для трьохетапного протоколу.

В роботі [18] запропоновано поліпшення реалізації протоколу Шаміра у квантовій криптографії - квантовий трьохетапний протокол, що використовує властивість квантової суперпозиції.

В роботі [19] представлено трьохетапний протокол Шаміра з модифікацією алгоритмом Ель-Гамаля використаний для обміну ключами AES в ad hoc 802.11 мережі.

Автори роботи [91] пропонують метод надійного шифрування на основі комутативних перетворень. Цей метод включає як три базові компоненти такі криптографічні протоколи: протокол узгодження ключів Діффі-Хеллмана, алгоритм комутативного шифрування Поліґа-Геллмана (Pohlig-Hellman) і трьохетапний протокол Шаміра. Для виконання комутативного шифрування використовується шифр піднесення до степеня.

Разом з тим, як показано в [20], шифр піднесення до степеня безпечний настільки, наскільки важка проблема дискретного логарифма. Відповідно,

протоколи на основі шифру піднесення до степеня не захищені від атак із використанням гіпотетичних квантових комп'ютерів.

В роботі [20] запропоновано постквантовий трьохетапний криптографічний протокол. Цей протокол заснований на операціях піднесення до степеня і розкладання на множники. Він передбачає більший порівняно з алгоритмом Шаміра обсяг передавання даних каналом зв'язку (10 посилок проти 3), а також більшу кількість операцій, які виконує кожна зі сторін.

У роботі [21] запропоновано принципово новий принцип реалізації трьохетапного криптографічного протоколу. Його основною особливістю є те, що в ньому використовуються не просто операції множення вектора інформації, перестановки, перестановки ключів та їх інверсії, а застосовуються нелінійні перетворення, що базуються на ідентичній циклічній структурі спряжених перестановок. Це дозволяє забезпечити безпечний обмін інформацією через незахищений канал без необхідності спільного використання ключа.

Варто зазначити, що один з учасників інформаційного обміну, розшифрувавши перестановку π, може легко обчислити приватний ключ шифрування іншого учасника σ_A. Для цього він повинен обчислити перестановку π^{-1} а потім виконайте процедуру $\sigma_A = Y_1 \cdot \pi^{-1}$.

Таким чином, секретні ключі для запропонованого трьохетапного криптографічного алгоритму необхідно змінювати після кожної передачі повідомлення.

Крім того, учасників обміну може бути значно більше двох. Тоді доцільно значення Y_i від i-го користувача поміщати у відкритий довідник – публічний файл або директорію, – а не передавати кожен раз між користувачами. У цьому випадку два користувача i та j, які встановлюють захищений зв’язок, формують спільний ключ шифрування шляхом обчислень $K_{ij} = \sigma_i \cdot Y_j \cdot \omega_i$ і $K_{ji} = \sigma_j \cdot Y_i \cdot \omega_j$.

Розглянуті трьохетапні протоколи передавання даних доводять, що вони потребують забезпечення більш високих показників достовірності, оскільки для передавання одного повідомлення дані передають тричі, що збільшує ймовірність їх

ураження завадою, що особливо відчутно, наприклад, в умовах високої інтенсивності шуму.

Таким чином, актуальною задачею дослідження є розробка методу достовірного передавання інформації в системах зв’язку з нероздільним факторіальним кодуванням даних за ймовірності бітової помилки, близької до 0.5. Результати розробки та дослідження цього методу наведено в 4 розділі.

1.4. Методи циклової синхронізації

Під час отримання потоку даних циклова синхронізація ідентифікує вхідні циклові сигнали вирівнювання (синхронізаційні послідовності або синхрокомбінації). Звичайні формати кадрів гарантують, що синхрокомбінація надсилається в кожному кадрі та використовується для оцінки каналу та синхронізації кадрів.

Запропоновано кілька методів синхронізації кадрів кодових слів без використання окремих синхрокомбінацій. Наприклад, у дослідженні [46], запропоновано метод циклової синхронізації, який є різновидом підходу грубої сили. Цей метод буферизує дві довжини кадру символів і намагається декодувати з кожним можливим зміщенням, доки не буде визначено зсув, для якого декодування є успішним. Однак цей метод є більш складним з точки зору обчислень і не працює так добре, як синхронізатори циклів, які використовують синхрокомбінації.

Інше дослідження запропонувало використовувати операцію XOR для синхрокомбінацій і потоком даних для підвищення ефективності циклової синхронізації. Ця методика зменшує обсяг службової інформації, що передається через канал зв'язку, для встановлення циклової синхронізації повідомлень, тим самим реалізуючи більш ефективний розподіл ресурсів каналу зв'язку. Однак застосування такого підходу має значне обмеження для застосування в комунікаційних системах із блоковими та факторіальними кодами [21], [92], [93] порівняно з системами передавання даних із згортковими кодами [94], [95],

оскільки накладення синхрокомбінації на інформаційний блок може призвести до деформації блоку даних і безповоротної втрати інформації.

Особливої уваги заслуговують методи циклової синхронізації, які не покладаються на символи преамбули, які створюють накладні витрати з точки зору споживання енергії та використання каналу. Основна концепція таких методів [96], [97] полягає в тому, щоб опустити преамбулу та адаптувати формат кадру, однак адаптований формат кадру забезпечує синхрокомбінація (початок розділювача кадру (SFD)) у структурі кадру.

Комунікаційні системи з нероздільним факторіальним кодуванням вже використовують нестандартну та надлишкову структуру, яка не забезпечує окремого поля SFD. Крім того, структура кодового слова нероздільного факторіального коду дозволяє йому функціонувати як транспортний механізм у зв'язку з короткими пакетами [10], [11], [98], [99], [100], [101], [102], [103], [104], що є особливістю бездротових мереж 5G, мереж даних датчиків, машинної взаємодії. У таких системах не можна ігнорувати накладні витрати на синхрокомбінації [105], [106], [107].

Таким чином, підсумовуючи вище сказане, актуальною задачею залишається розробка методу циклової синхронізації для комунікаційних систем передавання інформації з нероздільним факторіальним кодуванням та методу циклової синхронізації для комунікаційних систем передавання інформації з нероздільним факторіальним кодуванням за ймовірності бітової помилки, близької до 0.5. Результати розробки таких методів та їх порівняльну оцінку ефективності наведено в розділах 2, 3, 5.

1.5. Підходи до забезпечення достовірного передавання інформації в умовах великої ймовірності бітової помилки

Для забезпечення достовірного передавання даних у комунікаційних системах та мережах використовуються різні підходи для вирішення цієї задачі. Одним із варіантів є використання методу прямої послідовності для розширення

спектру (DSSS), який часто використовується у Wi-Fi та інших бездротових технологіях.

Під DSSS розуміють метод передавання даних, який використовується в бездротовому зв’язку. Цей метод передбачає поширення сигналу на більшу смугу пропускання, ніж потрібна для передавання повідомлення. Вихідний сигнал даних поєднується з бітовою послідовністю вищої швидкості, також відомою як чіп-код, тим самим збільшуючи пропускну здатність сигналу.

Принцип роботи DSSS полягає в наступному: DSSS-передавач приймає вихідний сигнал даних і множить його на чіп-код, що призводить до ширшого частотного спектру сигналу. Цей чіп-код, або псевдовипадковий код, є послідовністю з 0 та 1 з набагато вищою частотою, ніж вихідний сигнал. Приймач, використовуючи той самий чіп-код, може потім відновити сигнал прямого спектру та отримати вихідні дані.

Це означає, що навіть якщо частина сигналу зазнає впливу під час передавання, приймач все одно може відновити вихідні дані, без повторного передавання сигналу, тим самим зменшуючи ймовірність втрати або пошкодження даних. Саме цей процес розширення та відновлення забезпечує DSSS стійкість до завад і здатність підтримувати цілісність сигналу.

Порівнюючи з методом FHSS (Frequency Hopping Spectrum Spreading - технологія передавання сигналу з швидкою псевдовипадковою перебудовою робочої частоти), який використовує обмежену кількість частот, але менше каналів у заданому діапазоні частот, DSSS використовує більшу пропускну здатність [108], [109].

DSSS забезпечує кілька переваг:

- DSSS покращує цілісність сигналу, поширюючи сигнал ширшим спектром, зменшуючи вплив завад і шуму;
- оскільки сигнал поширюється в ширшому діапазоні частот, неавторизованим користувачам важче перехопити або підслухати;

- кілька систем DSSS можуть співіснувати в одному діапазоні частот, не заважаючи одна одній завдяки унікальним чіп-кодам;
- незважаючи на збільшену пропускну здатність, DSSS може забезпечити високу швидкість передавання даних.

DSSS широко використовується в різноманітних програмах завдяки своїй стійкості до завад і цілісності сигналу. Ось деякі з основних застосувань DSSS:

- DSSS використовується в мережах Wi-Fi, щоб дозволити кільком пристроям спілкуватися одночасно без взаємних завад. Це особливо поширено у стандарті Wi-Fi 802.11b;
- GPS використовує DSSS для забезпечення цілісності сигналу та точності даних про місцезнаходження;
- DSSS використовується в стільникових мережах 3G, 4G і 5G для покращення якості сигналу та зменшення помилок.

Загалом DSSS пропонує такі переваги, як стійкість до завад, покращений захист і збільшена пропускна здатність мережі.

1.6. Цілі та задачі дисертаційного дослідження

Мета дослідження полягає в забезпеченні достовірного передавання інформації в системах з нероздільним факторіальним кодуванням даних за високої ймовірності бітової помилки.

Аналіз розглянутих методів нероздільного факторіального кодування, показує, що вони вирішують проблеми криптографічного захисту інформації та завадостійкого кодування. Однак, актуальною задачею залишається підвищення достовірності передавання інформації для існуючих методів нероздільного факторіального кодування, а також розробка нових методів, що дозволить розширити область використання факторіальних кодів.

Аналіз методів циклової синхронізації показав, що існують методи, які є різновидом підходу грубої сили, що буферизують дві довжини кадру символів і намагаються декодувати з кожним можливим зміщенням, доки не буде визначено

зсув, для якого декодування є успішним. Однак ці метод є більш складним з точки зору обчислень. Інші методи циклової синхронізації використовують операцію XOR для синхрокомбінацій з потоком даних для підвищення ефективності циклової синхронізації, однак застосування такого підходу має значне обмеження для застосування в комунікаційних системах із блоковими та факторіальними кодами, оскільки накладення синхрокомбінації на інформаційний блок може призвести до деформації блоку даних і безповоротної втрати інформації. Розглянуто методи циклової синхронізації, які не використовують символи преамбули. Їх основна концепція, полягає в тому, щоб адаптувати формат кадру, однак адаптований формат кадру забезпечує синхрокомбінація (початок розділювача кадру (SFD)) у структурі кадру.

Розглянуто існуючі застосування, що реалізують нероздільне факторіальне кодування, зокрема трьохетапний криптографічний протокол на основі перестановок, який дає змогу захищеним шляхом передавати повідомлення між двома сторонами без необхідності передавання або оголошення відкритого чи закритого ключа. Разом з тим, особливістю таких протоколів є те, що вони потребують більш високих показників достовірності, оскільки для передавання одного повідомлення дані передають тричі, що збільшує ймовірність їх ураження завадою, що особливо відчутно в умовах високого їх рівня.

Виконано аналіз методу DSSS, який забезпечує стійкість до завад і здатність підтримувати цілісність сигналу під час процесу передавання даних.

Виконаний аналіз дав змогу чітко сформулювати задачі роботи. Вони полягають у розробці методу циклової синхронізації для комунікаційних систем передавання інформації з нероздільним факторіальним кодуванням, у тому числі, за ймовірності бітової помилки, близької до 0.5, а також у розробці методу достовірного передавання інформації в системах зв’язку з нероздільним факторіальним кодуванням даних за такої ж імовірності бітової помилки.

Для верифікації та дослідження ефективності розроблених методів циклової синхронізації, достовірного передавання інформації, а також для формулювання

рекомендацій щодо їх застосування необхідно виконати порівняльну експериментальну оцінку, що є заключною задачею дисертаційної роботи.

Виходячи з цього, задачами роботи є:

- розробка методу циклової синхронізації для комунікаційних систем передавання інформації з нероздільним факторіальним кодуванням;
- розробка методу циклової синхронізації для комунікаційних систем передавання інформації з нероздільним факторіальним кодуванням за ймовірності бітової помилки, близької до 0.5;
- розробка методу достовірного передавання інформації в системах зв'язку з нероздільним факторіальним кодуванням даних за ймовірності бітової помилки, близької до 0.5;
- проведення порівняльної експериментальної оцінки розроблених методів циклової синхронізації, достовірного передавання інформації, формування рекомендації щодо їх застосування.

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

1.7. Висновки

У першому розділі:

- 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 підвищує їх ефективність;
- досліджено сучасний стан предметної області дисертаційної роботи: виконано аналіз основних варіантів підвищення достовірності передавання інформації; проаналізовано роботи присвячені методам інтегрованого захисту інформації; розглянуто та проаналізовано роботи, присвячені основам

факторіального кодування даних, наведено порівняльний аналіз їх типів; виконано аналіз існуючих методів циклової синхронізації; виконано аналіз методів достовірного передавання інформації в комунікаційних системах, зокрема, з використанням факторіального кодування;

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

2. МЕТОД ЦИКЛОВОЇ СИНХРОНІЗАЦІЇ СИСТЕМ ПЕРЕДАВАННЯ ДАНИХ З НЕРОЗДІЛЬНИМ ФАКТОРІАЛЬНИМ КОДУВАННЯМ НА ОСНОВІ ПОДІЛУ КОДОВОГО СЛОВА НА ПРЕФІКСНУ Й СУФІКСНУ ЧАСТИНИ

2.1. Вступ

У першому розділі дисертації поставлено завдання забезпечення встановлення циклового синхронізму в системах з нероздільним факторіальним кодуванням. Розглянуто основні роботи щодо процедур циклової синхронізації в різних мережевих протоколах. Описано основну роль та ключові аспекти використання процедур циклової синхронізації в комунікаційних системах і мережах, у тому числі з короткими пакетами. Розглянуто особливості систем з факторіальним кодуванням та їхні відмінності.

Метою цього розділу є представлення розробленого та дослідженого у рамках дисертаційного дослідження методу циклової синхронізації приймальної та передавальної станцій систем з нероздільним факторіальним кодуванням в умовах впливу в каналі зв’язку завад високої інтенсивності, що забезпечується за рахунок використання як синхрокомбінації перестановки чисел, її поділу на префіксну та суфіксну частини, а також використання мажоритарної обробки прийнятих фрагментів. Крім того, в другому розділі представлено оцінку теоретично визначених і досліджених імовірнісних показників розробленого методу та перевірено їх відповідності експериментально отриманим імовірнісним характеристикам. Зокрема, визначено оцінку ймовірності правильного встановлення синхронізму та ймовірності хибного встановлення синхронізму.

2.2. Опис методу

Основною ідеєю для запропонованого методу встановлення циклової синхронізації є наступне:

– синхрокомбінацією є перестановка π;

– конструкція синхрокомбінації поділяється на префіксну частину, що визначає межі її символів, і суфіксну частину, яка визначає межі синхрокомбінації-перестановки;

– забезпечення достовірності передавання синхрокомбінації забезпечується її багаторазовим повторенням і мажоритарною обробкою прийнятих фрагментів.

Метод встановлення циклової синхронізації передбачає такі кроки:

1) синхрокомбінацією є перестановка π довжини M символів. Символи перестановки π кодуються рівномірним двійковим кодом таким чином, щоб перший символ записувався послідовністю, що складається тільки з нулів, а другий – тільки з одиниць. Крім того, третій символ синхрокомбінації в двійковому записі повинен починатися нулем, а останній символ повинен закінчуватися двійковою одиницею. Решта символів повинна містити максимальну кількість переходів з одиниці в нуль і навпаки (для мінімізації витрат часу на синхронізацію біт) і не повинна містити комбінації $10\ldots01\ldots10$;
$l_r \quad l_r$

2) під час старту процедури пошуку синхронізму в накопичувач приймача записуються три послідовних фрагмента отриманої з каналу послідовності біт. Довжина кожного з фрагментів дорівнює довжині синхрокомбінації $n = M \cdot l_r = M\lceil \log_2 M \rceil$;

3) за прийнятими фрагментами формується уточнена послідовність R. Кожен біт цієї послідовності обчислюється за мажоритарним принципом на основі відповідних біт прийнятих фрагментів. Таким чином, якщо i-ті біти фрагментів містять більше «одиниць», i-тому біту уточненої послідовності присвоюється значення «одиниці», в іншому випадку – «нуля»;

4) в уточненій послідовності R з урахуванням її циклічного зсуву перевіряється наявність комбінації $10\ldots01\ldots10$. Якщо її знайдено, причому вона
$l_r \quad l_r$

одна, уточнену послідовність R циклічно зсувають таким чином, щоб вона починалась з префіксу $0\ldots01\ldots1$;
$\quad\ l_r \quad\ l_r$

5) уточнена послідовність R порівнюється з еталоном π. Якщо вона збігається з еталонною синхрокомбінацією з точністю до одного з її символів, то процедура підстроювання циклової фази припиняється, циклічним зсувом компенсується фазова неузгодженість і формується сигнал «Пошук циклової фази завершений». Цей сигнал відправляється на станцію передавання даних і є командою на перехід до процедури передавання даних користувача;

6) якщо комбінацію $10\ldots01\ldots10$ в уточненій послідовності R не знайдено,
$\quad\ l_r \quad\ l_r$

знайдено таких комбінацій дві та більше, а також якщо суфікс уточненої послідовності R відрізняється від суфіксу еталонної синхрокомбінації π більше, ніж на один символ, то додатково приймаються ще два фрагменти, тим самим збільшуючи їх число до п’яти. Знову повторюються всі операції виявлення синхрокомбінації, починаючи з п. 3;

7) число накопичених фрагментів може послідовно збільшуватися до деякого, наперед заданого порогу. Після досягнення цього порогу число накопичених фрагментів не змінюється. Процес пошуку синхронізму триває, поки або не буде знайдений синхронізм, або не закінчиться ліміт часу на виконання пошуку синхронізму. У останньому випадку процедура пошуку синхронізму завершується, а на вихід системи видається сигнал «Аварія каналу».

Пояснювати метод циклової синхронізації будемо на прикладі використання як синхрокомбінації перестановки π довжини $M=8$ (послідовності десяткових символів множини $\{0,1,2,3,4,5,6,7\}$).

Кожен символ цієї множини кодується рівномірним двійковим кодом, наприклад, як показано в таблиці 2.1.

Таблиця 2.1 – **Схема кодування символів перестановки**

Десятковий запис	0	1	2	3	4	5	6	7
Двійковий запис	000	001	010	011	100	101	110	111

Сконструюємо синхрокомбінацію з префіксом, що виявляє межі символів. Таким префіксом може бути комбінація, складена з першого та останнього символів множини $\{0,1,2,3,4,5,6,7\}$ – символів 0 і 7. Останній символ синхрокомбінації повинен закінчуватися одиницею, а перший символ суфікса повинен починатися нулем. Решта символів обираються таким чином, щоб число знакозмін (змін двійкової одиниці на двійковий нуль або навпаки – нуля на одиницю) було максимальним.

Діаграму розподілу кількості перестановок з повної множини перестановок довжини $M=8$, символи яких закодовано відповідно до схеми таблиці 2.1, у залежності від кількості знакозмін наведено на рисунку 2.1.

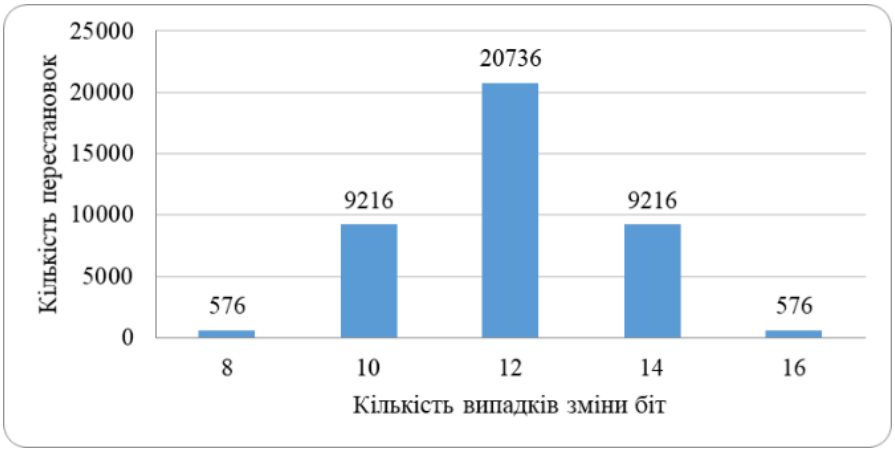


Рисунок 2.1 – Діаграма розподілу кількості перестановок у залежності від кількості знакозмін

З рисунку 2.1 видно, що найбільша можлива кількість переходів з 0 в 1 та з 1 в 0 дорівнює 16. Виконаємо пошук перестановок, де перші символи 0 і 7, останній символ закінчується двійковою одиницею, перший символ суфікса починається

двійковим нулем, а сама перестановка містить 16 знакозмін. Знайдені експериментальним шляхом такі перестановки наведено в таблиці 2.2.

Одна з наведених у таблиці 2.2 послідовностей записується в постійний запам'ятовувальний пристрій (ПЗП), який стане джерелом синхрокомбінації передавальної станції.

Приклад такої синхрокомбінації:

- десятковий запис $(0,7,1,2,4,6,5,3)$;
- двійковий запис $(000,111,001,010,100,110,101,011)$.

Розглянемо процедури, реалізовані на приймальній станції.

Таблиця 2.2 – **Перелік перестановок довжини $M=8$ для використання як синхрокомбінації**

№	Десятковий запис	Двійковий запис
1	$(0,7,1,2,4,6,5,3)$	$(000,111,001,010,100,110,101,011)$
2	$(0,7,1,2,6,4,5,3)$	$(000,111,001,010,110,100,101,011)$
3	$(0,7,1,3,2,4,6,5)$	$(000,111,001,011,010,100,110,101)$
4	$(0,7,1,3,2,6,4,5)$	$(000,111,001,011,010,110,100,101)$
5	$(0,7,2,4,6,5,1,3)$	$(000,111,010,100,110,101,001,011)$
6	$(0,7,2,4,6,5,3,1)$	$(000,111,010,100,110,101,011,001)$
7	$(0,7,2,6,4,5,1,3)$	$(000,111,010,110,100,101,001,011)$
8	$(0,7,2,6,4,5,3,1)$	$(000,111,010,110,100,101,011,001)$
9	$(0,7,3,1,2,4,6,5)$	$(000,111,011,001,010,100,110,101)$
10	$(0,7,3,1,2,6,4,5)$	$(000,111,011,001,010,110,100,101)$
11	$(0,7,3,2,4,6,5,1)$	$(000,111,011,010,100,110,101,001)$
12	$(0,7,3,2,6,4,5,1)$	$(000,111,011,010,110,100,101,001)$

Перш за все, зазначимо, що в приймач надходить синхрокомбінація, уражена дією завади в каналі зв'язку і зсунута за фазою щодо положення ковзного вікна (виконаного у вигляді регістра зсуву). При цьому ні інтенсивність завади, ні зсув ковзного вікна (величина фазової неузгодженості циклів передавача та приймача) апріорно невідомі. Тому першим завданням системи циклової синхронізації є підвищення достовірності прийнятих даних, а другий – визначення та компенсація фазової неузгодженості.

Підвищення достовірності прийнятих даних досягається багаторазовим повторенням синхрокомбінації та накопиченням результату. Так, коефіцієнтом накопичення, який дорівнює числу накопичуваних фрагментів отриманої з каналу послідовності біт, довжина кожного з яких дорівнює довжині синхрокомбінації, обирається з ряду непарних чисел 3, 5, 7…. Очевидно, що чим вищий коефіцієнт накопичення, тим більша ймовірність правильного рішення (тобто менша ймовірність помилкової синхронізації) та більший час входження в синхронізм. У запропонованому методі циклової синхронізації виконується адаптивний підбір коефіцієнта накопичення (чим гірший канал зв'язку, тим більшим встановлюється коефіцієнт накопичення). Це досягається тим, що прийнята з каналу зв'язку послідовність біт записується в буферний накопичувач загальною ємністю, наприклад, 11 синхрокомбінацій. Під час старту процедури пошуку синхронізму встановлюється мінімальне значення коефіцієнта накопичення $l = 3$, після чого в буферний накопичувач записується три фрагменти прийнятої послідовності, довжина кожного з яких дорівнює довжині синхрокомбінації. Циклічний зсув прийнятої послідовності випадковий. За цими трьома фрагментами мажоритарно обчислюється уточнена послідовність R, у якій частину помилок (якщо вони є) виправлено – зазначеною процедурою виправляються всі помилки кратністю до $\frac{l-1}{2}$ включно, які виникають у відповідних бітах прийнятих фрагментів. Відповідно, максимальна загальна кількість бітових помилок, яку здатна виправити мажоритарна обробка, становить $\frac{l-1}{2} n = \frac{l-1}{2} M \lceil \log_2 M \rceil$ з прийнятих ln біт.

Наприклад, для $M = 8$ і $l = 3$ це значення дорівнює 24 бітові помилки з прийнятих 72 біт. Для $M = 8$ і $l = 5$ – збільшується до 48 з 120 біт.

У накопичувачі виконується побітовий циклічний зсув уточненої послідовності R, за яким після виконання кожного з зсувів перевіряється, чи виявлено префікс синхрокомбінації. Якщо префікс виявлено, то перевіряється і суфікс. Якщо обидві перевірки завершено позитивно, то процедуру адаптації коефіцієнта накопичення до якості каналу зв'язку завершено, а синхронізм встановлено. У цьому випадку сигнал «Пошук синхронізму завершений» (ПЗ) відправляється зворотним каналом на передавальну станцію, а канал даних переходить зі стану встановлення з'єднання в стан перенесення призначених для користувача даних.

Якщо не виявлено префікс синхрокомбінації або не підтверджено її цілісність, тобто для визначеного значення коефіцієнта накопичення синхрокомбінацію не визначено, то в накопичувач додатково записуються два фрагменти отриманої з каналу послідовності біт. Після цього обчислюється уточнена після мажоритарної обробки п'яти фрагментів послідовність, за якою виконують виявлення синхрокомбінації. Якщо і ця спроба виявити префікс і суфікс синхрокомбінації невдала, то коефіцієнт накопичення збільшується ще на два. Збільшення коефіцієнта накопичення триває до тих пір, поки не буде досягнуто його граничне значення, рівне, наприклад, 11. Далі пошук синхронного стану триває або до завершення процедури пошуку, або до вичерпання ліміту часу на пошук синхронізму. Якщо після закінчення ліміту часу, відведеного на пошук синхронізму, він не знайдений, то система циклової синхронізації формує сигнал «Аварія каналу». Процедура пошуку припиняється.

2.3. Імовірнісні показники встановлення циклового синхронізму

Теоретично визначимо одну з основних показників системи циклової синхронізації – ймовірність встановлення циклового синхронізму за прийнятими

фрагментами, $l = 3,5,\ldots l$, у залежності від імовірності бітової помилки p_0 у каналі зв'язку.

Імовірність бітової помилки в уточненій послідовності R після мажоритарної оброки l прийнятих фрагментів

$$p_0^* = \sum_{i=\frac{(l+1)}{2}}^{l} C_l^i p_0^i \left(1 - p_0\right)^{l-i}. \tag{2.1}$$

Помилка в символі перестановки виникає тоді, коли хоча б один біт у його двійковому представленні в уточненій послідовності R визначено невірно. Імовірність такої події

$$p_{symb} = \sum_{i=1}^{l_r} C_{l_r}^i \left(p_0^*\right)^i \left(1 - p_0^*\right)^{l_r - i} = 1 - \left(1 - p_0^*\right)^{l_r}. \tag{2.2}$$

Система циклової синхронізації прийме вірне рішення про встановлення циклового синхронізму, якщо:

1) перші два символи (двійкове представлення яких містить усі нулі та всі одиниці) уточненої послідовності R буде визначено без помилки;

2) третій символ уточненої послідовності R буде починатися з двійкового нуля;

3) останній символ уточненої послідовності R буде закінчуватися двійковою одиницею;

4) не більше одного з групи від третього до останнього символів уточненої послідовності R визначено невірно.

Імовірність першої події

$$P_1 = \left(1 - p_{symb}\right)^2. \tag{2.3}$$

Імовірність того, що одночасно відбудуться друга, третя та четверта події

$$P_{2-4} = \left(1 - p_{symb}\right)^{M-2} +$$

$$+2\left(\left(1 - p_0^*\right)\left(1 - \left(1 - p_0^*\right)^{l_r - 1}\right)\right)\left(1 - p_{symb}\right)^{M-3} + \left(M - 4\right) p_{symb} \left(1 - p_{symb}\right)^{M-3}. \tag{2.4}$$

Тоді ймовірність встановлення правильного циклового синхронізму за l прийнятими фрагментами

$$P_{true} = P_1 P_{2-4}. \tag{2.5}$$

Підставляючи вирази (3) і (4) для P_1 і P_{2-4} до (5), отримаємо

$$P_{true} = \left(1 - p_{symb}\right)^{M-1} \left(\begin{array}{c} 1 + 2\left(\left(1 - p_0^*\right)\left(1 - \left(1 - p_0^*\right)^{l_r - 1}\right)\right) + \\ + \left(M - 5\right) p_{symb} \end{array} \right). \tag{2.6}$$

Імовірність встановлення хибного циклового синхронізму за l прийнятими фрагментами визначається ймовірністю виникнення події, за якої уточнена послідовність R буде циклічно зсунутою синхрокомбінацію з точністю до символу суфікса за умови, що перший і останній його біти змінені не будуть. Таким чином, ця ймовірність залежить від розподілу відстаней Хеммінга від синхрокомбінації до її циклічних зсувів. Такий розподіл для перестановок, наведених у таблиці 2.2, представлений у таблиці 2.3.

Таблиця 2.3 – **Розподіл відстаней Хеммінга від синхрокомбінації до її циклічних зсувів**

№ перестановки в таблиці 2.2	Відстань Хеммінга t						
	6	8	10	12	14	16	18
1	0	1	4	12	0	6	0
2	2	2	2	7	2	8	0
3, 5	0	2	4	6	8	3	0
4, 7	2	0	4	11	0	2	4
6, 9	2	2	2	5	6	6	0
8, 10	0	2	4	7	6	4	0
11	0	4	2	7	4	6	0
12	2	2	2	6	4	7	0

Імовірність того, що уточнена послідовність R буде циклічно зсунутою синхрокомбінацію, дорівнює

$$P_{shift} = \sum_{t=1}^{n} f(t)\left(p_0^*\right)^t \left(1-p_0^*\right)^{n-t}, \qquad (2.7)$$

де $f(t)$ – кількість помилок кратності t в уточненій послідовності R, які призводять до циклічного зсуву синхрокомбінації. Розподіл значень $f(t)$ якраз і наведено в таблиці 2.3. Значення $f(t) \equiv 0$ для всіх t, відсутніх у цій таблиці.

Розроблений метод передбачає можливість невірного розпізнавання одного символу суфіксу синхрокомбінації. Тому кратність помилки в уточненій послідовності R, яка призводить до циклічного зсуву синхрокомбінації, може відрізнятися на відстань Хеммінга від оригінального символу синхрокомбінації до пошкодженого символу уточненої послідовності R. Оскільки вага кожного символу помилки, яка переводить синхрокомбінацію до її циклічного зсуву, невідома, її ймовірність можна оцінити зверху виразом $\left(1-p_0^*\right)^{l_r}$. Кількість можливих помилок ваги t збільшується в кількість разів, що відповідає кількості всіх варіантів бітових помилок для кожного з символів суфікса – добутку суми комбінацій помилок символу на кількість символів суфікса. Таким чином, імовірність встановлення хибного циклового синхронізму за l прийнятими фрагментами можна оцінити зверху наступним чином:

$$P_{false} \le \sum_{t=1}^{n} f(t) \left(\begin{array}{l} (M-4)\sum\limits_{j=0}^{l_r} C_{l_r}^{j} \times \\ \times \left(p_0^*\right)^{|t-l_r|} \left(1-p_0^*\right)^{n-|t-l_r|} + \\ +2\sum\limits_{j=0}^{l_r-1} C_{l_r-1}^{j} \times \\ \times \left(p_0^*\right)^{|t-l_r+1|} \left(1-p_0^*\right)^{n-|t-l_r+1|} \end{array} \right). \qquad (2.8)$$

2.4. Оцінка ефективності методу

Для перевірки ефективності розробленого методу встановлення циклового синхронізму побудовано програмну модель передавання даних. Алгоритм роботи системи циклової синхронізації моделі наведено на рисунку 2.2.

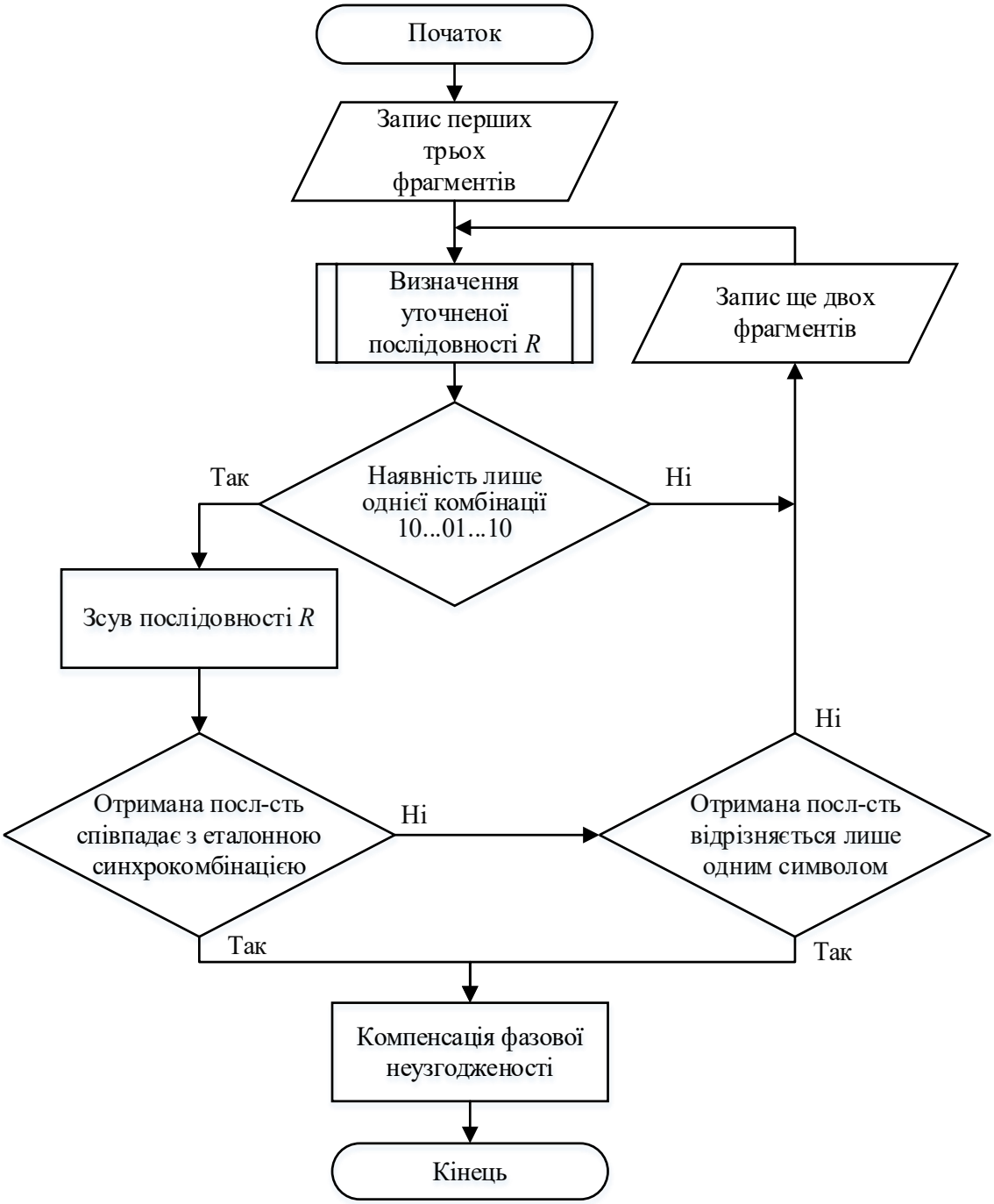


Рисунок 2.2 – Алгоритм роботи системи циклової синхронізації

Графік експериментально визначеної на 10000 випробуваннях залежності відносної частоти встановлення циклового синхронізму від фіксованого значення коефіцієнта накопичення l для ймовірності бітової помилки $p_0 \in \{0.1, 0.2, 0.3, 0.4\}$ наведено на рисунку 2.3. Як синхрокомбінацію використано перестановку 3 з

таблиці 2.2.

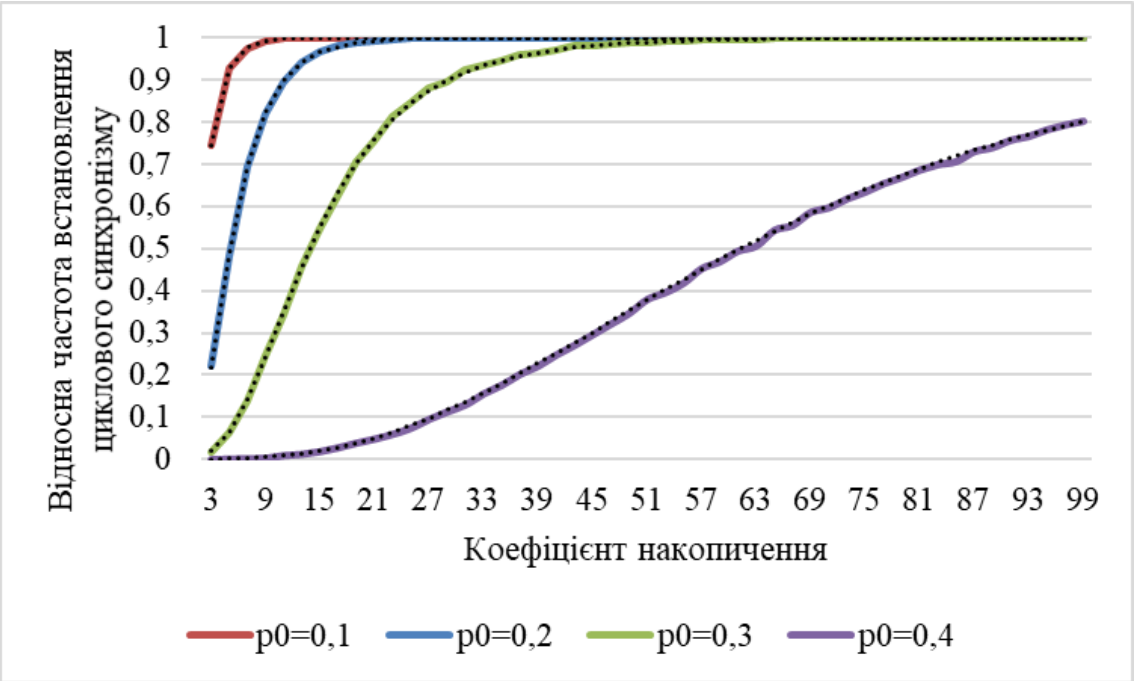


Рисунок 2.3 – Графік залежності відносної частоти встановлення циклового синхронізму від фіксованого значення коефіцієнта накопичення

На рисунку 2.3 пунктиром показано також відповідні графіки теоретичних залежностей P_{true} від l відповідно до виразу (2.6).

Наведені на рисунку 2.3 теоретичні та експериментальні залежності узгоджуються між собою за критерієм Пірсона з близькими до одиниці досягнутими рівнями значущості (p-value), інтерпретованими згідно з викладеною в [110] методологією. Це свідчить про коректність побудови програмної моделі.

Графік експериментально визначеної на 10000 випробуваннях залежності відносної частоти встановлення циклового синхронізму від коефіцієнта накопичення l для розробленого алгоритму та ймовірності бітової помилки $p_0 \in \{0.1, 0.2, 0.3, 0.4\}$ наведено на рисунку 2.4.

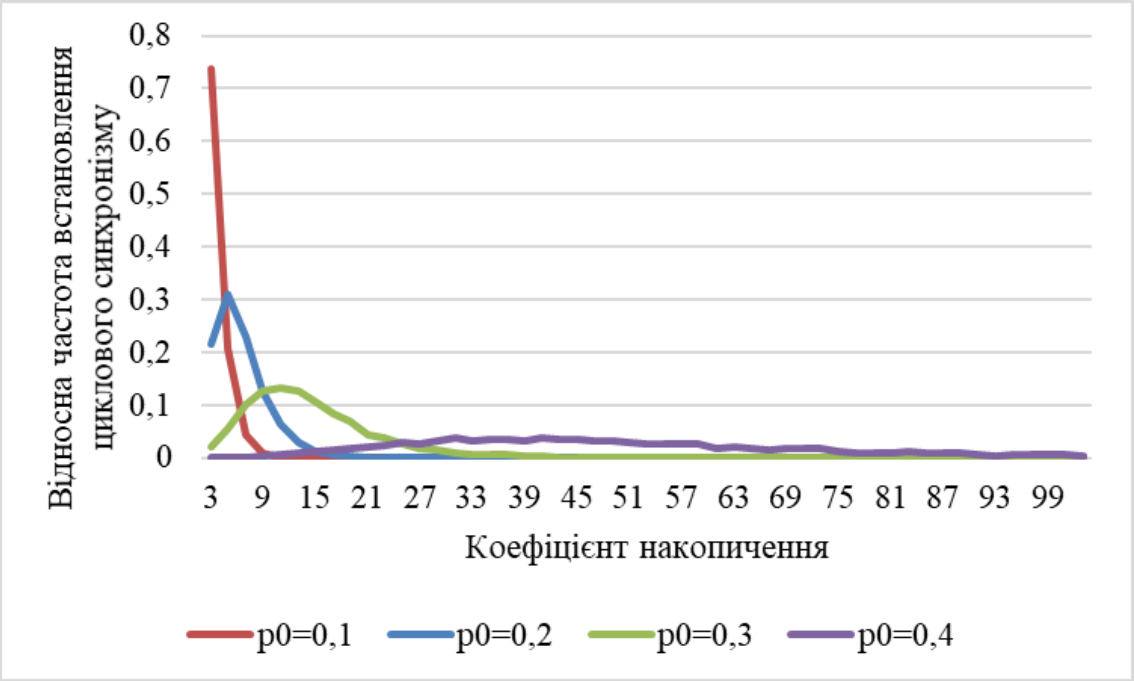


Рисунок 2.4 – Графік залежності відносної частоти встановлення циклового синхронізму від коефіцієнта накопичення

На рисунку 2.5 представлено графік експериментально визначеної інтегральної функції часу входження в синхронізм (імовірності встановлення циклового синхронізму за l прийнятими фрагментами), побудованої на основі наведеної на рисунку 2.4 залежності відносної частоти встановлення циклового синхронізму від коефіцієнта накопичення l.

Пояснимо відмінності залежностей, продемонстрованих на рисунках 2.3 і 2.5. На рисунку 2.3 показано відносну частоту встановлення циклового синхронізму за умови, якщо значення коефіцієнта накопичення є фіксованим і не змінюється в процесі роботи системи циклової синхронізації. Рисунок 2.4 демонструє відносну частоту встановлення циклового синхронізму за умови, що коефіцієнт накопичення поступово збільшує своє значення в залежності від результату пошуку на попередньому етапі. Як можна бачити з графіків 2.3 і 2.5, ці відносні частоти не є однаковими, а значення P_{true}, обчислене за (2.6) для заданих коефіцієнта накопичення встановлення циклового синхронізму l_{def} і ймовірності бітової помилки p_0 менше за суму представлених на рисунку 2.4 відносних частот встановлення циклового

синхронізму для $l \le l_{def}$. Ця ситуація обумовлена тим, що множина подій, за яких відбувається циклова синхронізація за фіксованого значення коефіцієнта накопичення l_{def}, не складається з множин подій, за яких відбувається циклова синхронізація під час поступового збільшення коефіцієнта накопичення $l \le l_{def}$. Так, наприклад, пошкодження біту в уточненій послідовності R, обчисленій за l фрагментами, за умови, що в обчисленій за $(l-2)$ фрагментами уточненій послідовності бітові помилки відсутні, породжує ситуацію, коли таку подію враховано в відносній частоті встановлення циклового синхронізму для коефіцієнта накопичення $(l-2)$, проте не враховано для коефіцієнта накопичення l.

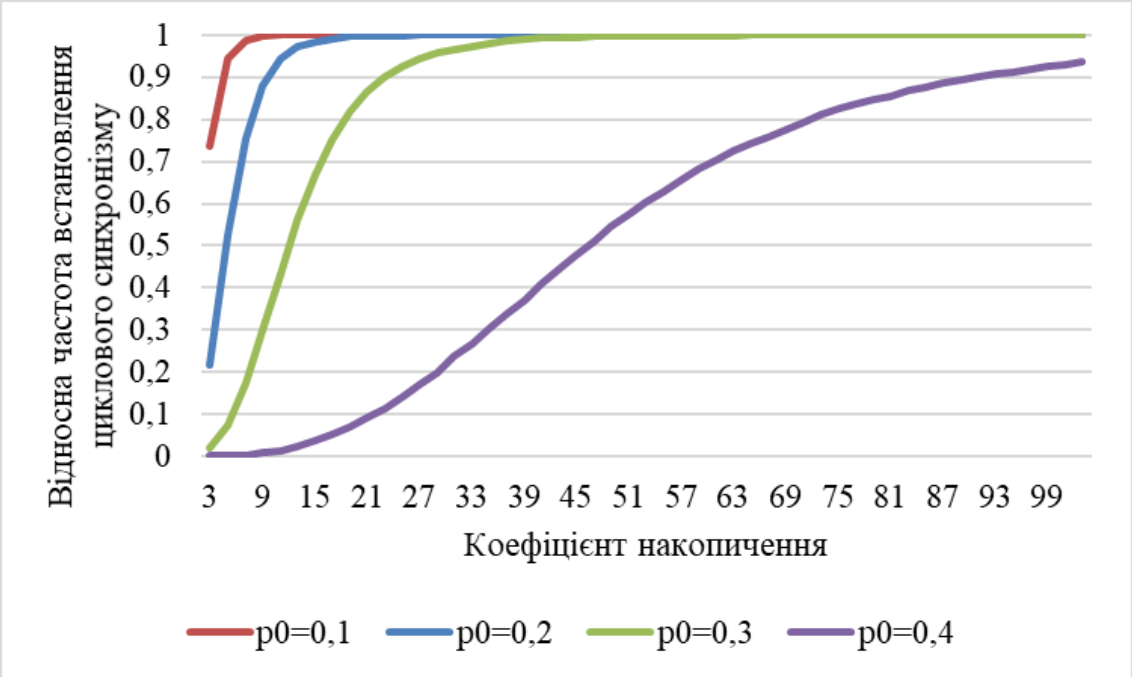


Рисунок 2.5 – Графік залежності ймовірності встановлення циклового синхронізму від коефіцієнта накопичення

Графіки залежностей оцінки зверху ймовірності встановлення хибного циклового синхронізму від кількості прийнятих фрагментів для синхрокомбінації 1 з таблиці 2.2 для ймовірності бітової помилки $p_0 \in \{0.1, 0.2, 0.3, 0.4\}$ наведено на рисунку 2.6.

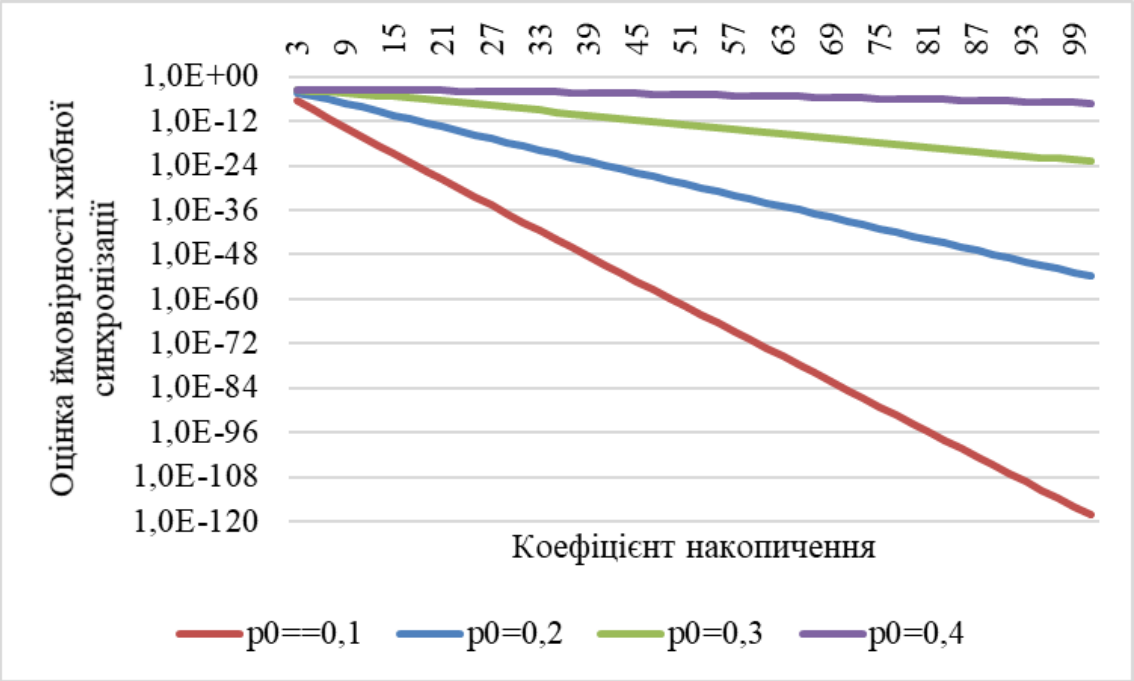


Рисунок 2.6 – Графіки залежностей оцінки зверху ймовірності встановлення хибного циклового синхронізму від коефіцієнта накопичення

Під час експериментального дослідження розробленого методу для перестановки 3 з таблиці 2.2 та значень імовірності бітової помилки $p_0 \in \{0.1, 0.2, 0.3, 0.4\}$ випадків встановлення хибного циклового синхронізму не виявлено для 10000 випробувань.

Зауважимо, що за показником імовірності встановлення хибного циклового синхронізму, зважаючи на показники з таблиці 2.3, найбільш ефективним є використання перестановки 1 з таблиці 2.2. Як підтвердження на рисунку 2.7 наведемо графіки залежностей оцінки зверху ймовірності встановлення хибного циклового синхронізму від кількості прийнятих фрагментів для синхрокомбінацій 1, 2, 3, 11 з таблиці 2 для ймовірності бітової помилки $p_0 = 0.4$.

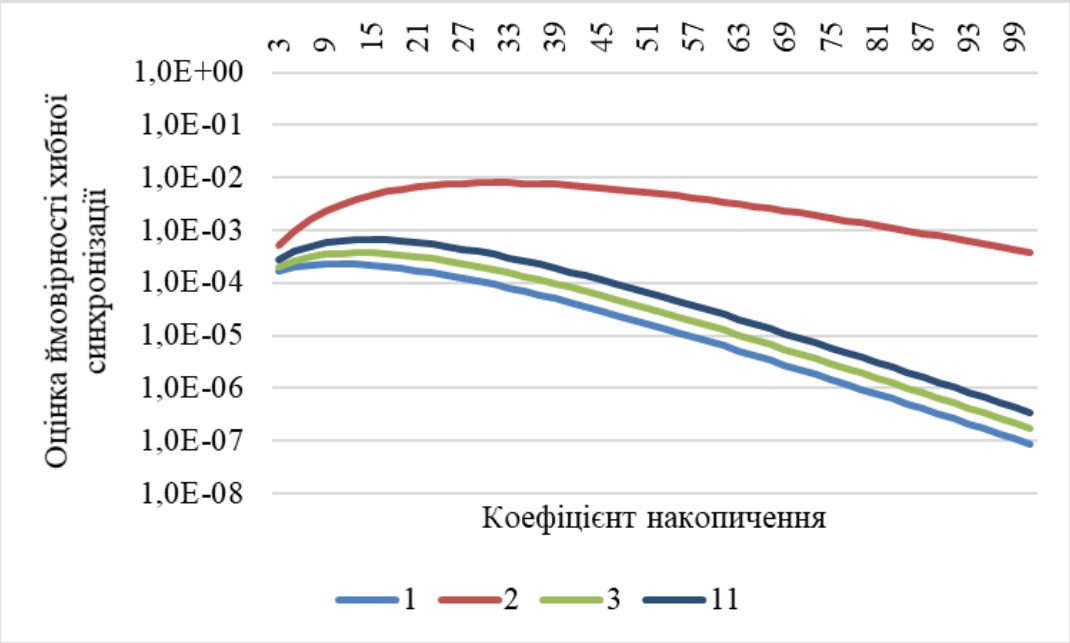
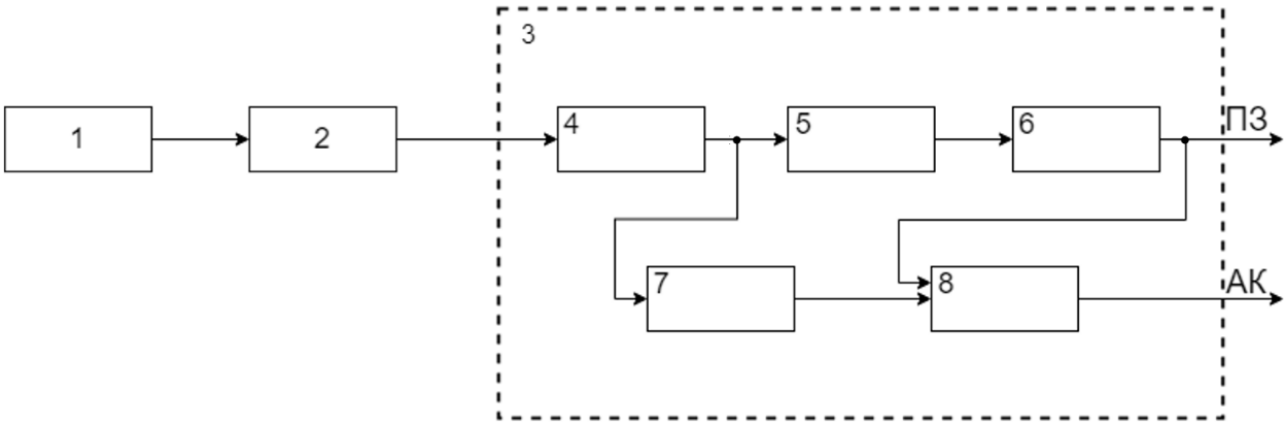


Рисунок 2.7 – Графіки залежностей оцінки зверху ймовірності встановлення хибного циклового синхронізму від коефіцієнта накопичення для ймовірності помилки $p_0 = 0.4$

2.5. Система циклової синхронізації

Схему системи циклової синхронізації наведено на рисунку 2.8.



1 – джерело синхрокомбінації; 2 – канал зв'язку; 3 – приймач синхрокомбінації; 4 – накопичувач синхрокомбінації; 5 – пристрій пошуку префіксу; 6 – пристрій пошуку суфіксу; 7 – лічильник числа циклів; 8 – вирішальний пристрій; ПЗ – сигнал «пошук синхронізму завершений»; АК – сигнал «Аварія каналу»

Рисунок 2.8 – Структурна схема системи циклової синхронізації

Пояснювати принцип побудови системи циклової синхронізації будемо на прикладі використання як синхрокомбінації перестановки π довжини $M = 8$ (послідовності десяткових символів множини $\{0,1,2,3,4,5,6,7\}$).

Кожен символ цієї множини представляється в двійковій системі числення, наприклад, як показано в таблиці 2.1.

Правила та вимоги щодо формування синхрокомбінації описано в пункті 2.2 цього розділу.

Сформовану послідовність записують у постійний запам'ятовувальний пристрій (ПЗП) 1, який стане джерелом синхрокомбінації передавальної станції.

Короткий опис процедур, які реалізовані на приймальній станції (більш детально процедуру синхронізації описано в пункті 2.2 цього розділу):

1) отримання приймачем 3 синхрокомбінації, яка уражена дією завади в каналі зв'язку 2 і зсунута за фазою щодо положення ковзного вікна;

2) запис отриманих послідовностей біт з каналу зв'язку 2 в буферний накопичувач 4. Визначення уточненої послідовності R за допомогою мажоритарної обробки накопичених даних;

3) пошук дешифратором префікса 5 шляхом перевірки в накопичувачі 4 відповідності кожного побітового циклічного зсуву уточненої послідовності R префіксу синхрокомбінації;

4) пошук дешифратором суфікса 6, за аналогічним алгоритмом, суфікса синхрокомбінації;

5) видача дешифратором суфікса 6 сигналу «Пошук синхронізму завершений» (ПЗ) за вдалого завершення процедур пошуку префікса та суфікса;

6) контроль використання ліміту часу на пошук синхронізму за допомогою лічильник числа циклів 7;

7) формування сигналу «Аварія каналу» (АК) вирішальним пристроєм 8 у випадку закінчення ліміту часу, відведеного на пошук синхронізму.

2.6. Висновки

У другому розділі розроблено метод циклової синхронізації для систем зв’язку з нероздільним факторіальним кодуванням, який за рахунок використання як синхрокомбінації перестановки заданої довжини, її поділу на префіксну та суфіксну частини, а також мажоритарної обробки прийнятих фрагментів дозволяє досягти циклової синхронізації в умовах впливу в каналі зв’язку завад високої інтенсивності. Метод може бути використаний для зменшення часу входження в синхронізм і підвищення стійкості комунікаційної системи в умовах впливу завад у каналі зв’язку.

Розроблено та застосовано алгоритм, що реалізує запропонований метод.

Технічним результатом, на який спрямовано застосування запропонованого методу в комунікаційній системі, є зменшення втрати часу сеансу зв’язку на виконання функції встановлення з’єднання і, отже, збільшення часу в сеансі на передавання даних користувача і, як наслідок, на збільшення пропускної здатності каналу зв’язку.

Разом з тим, розроблений метод може бути ефективним для реалізації не тільки в системах з короткими пакетами та нероздільним факторіальним кодуванням. Це питання є предметом подальших досліджень.

Основні результати дослідження та розробки методу представлено в [111].

3. МЕТОД ЦИКЛОВОЇ СИНХРОНІЗАЦІЇ СИСТЕМ ПЕРЕДАВАННЯ ДАНИХ З НЕРОЗДІЛЬНИМ ФАКТОРІАЛЬНИМ КОДУВАННЯМ НА ОСНОВІ КОРЕЛЯЦІЙНОЇ ОБРОБКИ

3.1. Вступ

У другому розділі представлено та досліджено метод циклової синхронізації приймальної та передавальної станцій комунікаційних систем передавання інформації з нероздільним факторіальним кодуванням в умовах впливу в каналі зв’язку завад високої інтенсивності, що забезпечується за рахунок використання як синхрокомбінації перестановки чисел, її поділу на префіксну та суфіксну частини, а також використання мажоритарної обробки прийнятих фрагментів. Однак метод має декілька недоліків:

1. інтегральна функція часу входження в синхронізм не є оптимальною, оскільки метод не враховує всі надлишкові особливості структури кодового слова нероздільного факторіального коду;

2. імовірність хибної синхронізації є високою. Ця особливість стає більш помітною, коли ймовірність бітової помилки зростає, а значення накопичення l зменшується.

Тому актуальною задачею є підвищення швидкості та достовірності встановлення циклового синхронізму в системах з нероздільним факторіальним кодуванням, у тому числі, в умовах впливу в каналі зв’язку завад високої інтенсивності. У цьому розділі вирішення поставленої задачі досягнуто за рахунок використання як синхрокомбінації перестановки, в якої мінімальна відстань Хеммінга до всіх її циклічних зсувів є максимальною, а також за рахунок використання кореляційної та мажоритарної обробки фрагментів даних, де довжина фрагмента дорівнює довжині синхрокомбінації. Отримано теоретичну та практичну оцінки кількості накопичених фрагментів, необхідних для встановлення циклової синхронізації. Крім того, оцінено ефективність застосування процедури перемішування накопичених фрагментів.

3.2. Опис методу

Для підвищення ймовірності циклової синхронізації та зменшення ймовірності хибної синхронізації в умовах шуму високої інтенсивності для систем з короткими пакетами даних, які використовують нероздільне факторіальне кодування, необхідно вирішити наступні завдання:

- розробити метод і відповідний алгоритм циклової синхронізації для систем з короткими пакетами даних з використанням нероздільного факторіального кодування;
- оцінити ймовірність встановлення циклової синхронізації в умовах шуму різної інтенсивності;
- побудувати модель, яка може реалізувати алгоритм, розроблений для встановлення циклової синхронізації;
- виконати порівняльний аналіз імовірності знаходження меж циклів за різних умов інтенсивності шуму.

У цьому дослідженні будемо оцінювати ефективність методу циклової синхронізації на основі перестановки в модельній системі передавання даних з такими обмеженнями.

- канал зв'язку – двійковий симетричний з незалежними бітовими помилками;
- імовірність бітової помилки в каналі зв'язку $p_0 \le 0.495$;
- імовірність встановлення правильної синхронізації $P_{true} \ge 0.9997$ для будь-якого заданого значення ймовірності бітової помилки $p_0 \le 0.495$;
- імовірність встановлення хибної синхронізації $P_{false} \le 3 \cdot 10^{-4}$ для будь-якого заданого значення ймовірності бітової помилки $p_0 \le 0.495$.

Для методу запропоновано використовувати як синхрокомбінацію перестановку, в якої мінімальна відстань Хеммінга до всіх її циклічних зсувів є максимальною. Крім того, для розпізнавання синхрокомбінації введено кореляційну обробку фрагментів даних, отриманих з каналу зв'язку.

Метод циклової синхронізації на основі кореляційної обробки реалізується за допомогою наступних етапів:

1. Передавач послідовно видає синхрокомбінацію в канал зв'язку. Синхрокомбінація — це перестановка π довжиною M, де мінімальна відстань Хеммінга d до всіх її циклічних зсувів є максимальною. Наприклад, для $M=8$, такою синхрокомбінацією є перестановка $\pi=(000,001,111,011,010,101,100,110)$ з точністю до її циклічного зсуву на кількість біт, кратну довжині символу в його двійковому представленні $l_r=3$, інверсії бітів та їх зворотного порядку.

2. Приймач накопичує K блоків, отриманих з каналу зв'язку, що складаються з l фрагментів по n біт. Значення K і l змінюються відповідно процедури, яку буде описано в цьому розділі.

3. Для кожного блоку уточнені послідовності R_k, де $k\in[1,K]$, розраховуються самостійно. Кожен біт цієї послідовності обчислюється за мажоритарним принципом на основі відповідних бітів отриманих фрагментів. Таким чином, якщо «одиниці» переважають у i-тих бітах фрагмента, то i-тому біту уточненої послідовності присвоюється значення «один». Якщо ж вони містять більше «нулів», то навпаки, їм присвоюється значення «нуль».

4. Для кожної уточненої послідовності R_k. обчислюються відстані Хеммінга для всіх циклічних зсувів синхрокомбінації . Якщо для якогось із зсувів відстань менша або дорівнює $d_{lim}=\lfloor(d-1)/2\rfloor$, уточнена послідовність ототожнюється з циклічним зсувом, якому відповідає ця відстань.

5. Синхронізація встановлюється, якщо кожна з послідовностей R_k, де $k\in[1,K]$, ідентифікується одним і тим же циклічним зсувом синхрокомбінації; інакше операції з виявлення синхрокомбінації повторюються, починаючи з другого пункту в поточному списку.

6. Кількість накопичених фрагментів може бути збільшена послідовно до попередньо визначеного порогу $l_{\max}(1)$. Якщо синхронізація не була встановлена

після досягнення цього порогу, процедура пошуку припиняється і система виводить повідомлення про збій каналу.

Нагадаємо, що кодові слова нероздільного факторіального коду є перестановками [63]. Кожен символ перестановки π кодується двійковим кодом фіксованої довжини. Довжина кодового слова дорівнює $l_r = \lceil \log_2 M \rceil$, де M – довжина перестановки.

Метод, який запропонований у цьому розділі та метод, що описаний у попередньому розділі, використовують перестановку π довжиною M як синхрокомбінацію. Крім того, ці методи використовують мажоритарну обробку отриманих даних. Проте, на відміну від методу з попереднього розділу, запропонований метод використовує кореляційну обробку прийнятих з каналу зв'язку фрагментів.

В подальшому, для більш лаконічного викладення матеріалу, будемо називати методом на основі поділу кодового слова – метод циклової синхронізації на основі поділу кодового слова на префіксну і суфіксну частини, описаний в попередньому розділі, а методом на основі кореляційної обробки – метод циклової синхронізації, який запропонований у цьому розділі.

Принцип роботи методу циклової синхронізації продемонстровано на прикладі використання, як синхрокомбінації перестановки π довжиною $M = 8$ (послідовності десяткових символів множини $\{0,1,2,3,4,5,6,7\}$).

Кожен символ множини кодується рівномірним двійковим кодом фіксованої довжини. Довжина кодового слова становить $l_r = \lceil \log_2 M \rceil = 3$ $(n = M \cdot l_r = 24)$, як показано в таблиці 3.1.

Таблиця 3.1 **– Схема кодування символів перестановки**

Десятковий запис	0	1	2	3	4	5	6	7
Двійковий запис	000	001	010	011	100	101	110	111

3.2.1. Вибір синхрокомбінації

Звернемо увагу на те, що перестановка π, яка використовується як синхрокомбінація, має задовольняти наступну умову: мінімальна відстань Хеммінга до всіх її циклічних зсувів є максимальною.

Позначимо через $\pi_i(j)$ перестановку π_i, яка циклічно зсунута вліво на j біт, та нехай d_{ij} буде відстанню Хеммінга від π_i до її циклічного зсуву $\pi_i(j)$, де $0 \le i \le M!-1$, $1 \le j \le n-1$. Крім того, нехай $d_i = \min_j(d_{ij})$, а $d = \max_i(d_i) = \max_i\left(\min_j(d_{ij})\right)$. Тоді синхрокомбінацією може бути будь-яка перестановка π_i в якій $\forall d_{ij} \ge d$.

Зауваження 3.1. *Завдяки n-періодичності відстаней d_{ij} відносно циклічних зсувів випливає, що для $\forall j,k$, $j \ne k$, $1 \le j,k \le n-1$, $d(\pi_i(j),\pi_i(k)) \ge d$ є вірною для відстані Хеммінга $d(\pi_i(j),\pi_i(k))$ між циклічними зсувами $\pi_i(j)$ та $\pi_i(k)$, якщо $d_{ij} \ge d$ для $\forall j \in [1,n-1]$.*

Теорема 3.1. *Якщо $n > 2$, то не існує перестановки π_i, для якої $2d_i > n+1$, тобто $d \le (n+1)/2$.*

Доведення. Значення d_i еквівалентне кодовій відстані коду, в якому комбінації утворюються всіма n циклічними зсувами бінарно представленої перестановки π_i. Визначимо межу Плоткіна відносно кодової відстані d_i такого коду. Для цього застосуємо співвідношення, отримані в ході доведення теореми 3.1 (межі Плоткіна) [68]. Для будь-якого (n,M,d)-коду, де n – довжина кодової комбінації, M – потужність коду, d – його кодова відстань, нерівність $M(M-1)d \le nM^2/2$ виконується якщо M є парним числом. Для кодових комбінацій у вигляді циклічних зсувів у двійковому представленні перестановки π_i справедливо $M = n$. Тоді $n(n-1)d_i \le n^3/2$ або $2d_i \le n+1+1/(n-1)$ за $n > 1$. Оскільки $1/(n-1) < 1$ за $n > 2$, то $2d_i \le n+1$.

За непарного значення M має місце нерівність $M(M-1)d \le n(M^2-1)/2$. Якщо $M=n$, тоді $n(n-1)d_i \le n(n^2-1)/2$ або $2d_i \le n+1$ за $n>1$.

Таким чином, якщо $n>2$, то $d \le (n+1)/2$. *Теорему доведено.*

Зауваження 3.2. *За* $M=8$ *та* $n=24$*, нерівність* $d \le (n+1)/2$ *набуває вигляду* $d \le 12$.

Розподіл значень d_i для кожної перестановки довжиною $M=8$ зображено на рисунку 3.1.

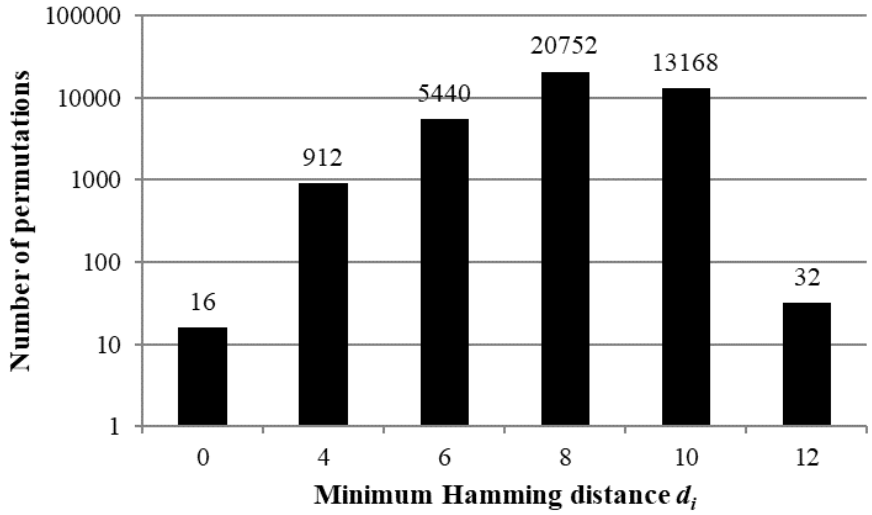


Рисунок 3.1 – Зв'язок між розподілом числа перестановок та відстанню d_i

З даних про розподіл, наведених на рисунку 3.1 слідує, що $d=\max_i(d_i)=12$ для $M=8$. Кількість перестановок, для яких $d=12$, дорівнює 32.

Зауваження 3.3. *Число* d_i*, що відповідає перестановці* π_i*, є інваріантом щодо посимвольних циклічних зсувів символу (кодового слова), інверсії бітів та їхнього зворотного порядку.*

На основі зауваження 3.3, знайдені 32 перестановки з $d=12$ можуть бути відповідним чином отримані з однієї перестановки, наприклад, яку наведено в таблиці 3.2.

Таблиця 3.2 – **Перестановка π_i, для якої $\forall d_{ij} \geq d$**

Десятковий запис	**Двійковий запис**
$(0,1,7,3,2,5,4,6)$	$(000,001,111,011,010,101,100,110)$

На наступному кроці побудуємо нормалізовану автокореляційну функцію (АКФ) для перестановки наведеної в таблиці 3.2. Обчислимо коефіцієнти автокореляції наступним чином:

$$\rho_j = \frac{1}{n}\left(n - 2d_{ij}\right).$$

На рисунку 3.2 відображено нормалізований графік АКФ для перестановки, поданої в таблиці 3.2.

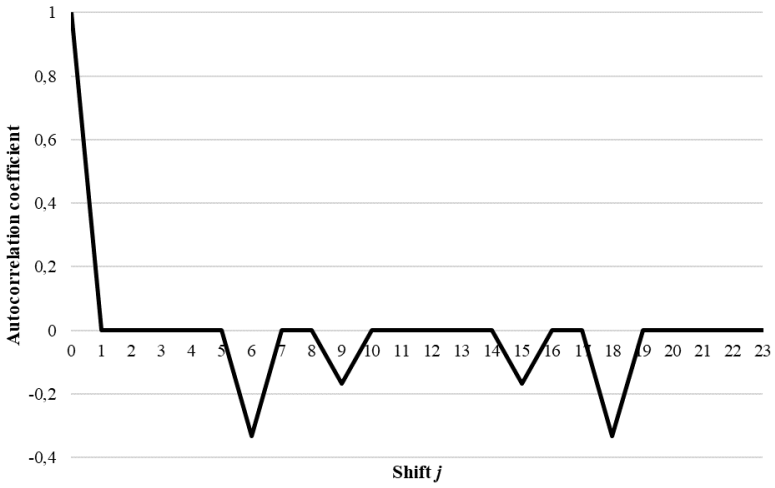


Рисунок 3.2 – Нормалізований графік АКФ для перестановки з таблиці 3.2

Кількість переходів від 0 до 1 та від 1 до 0, у двійковому записі наведеної в таблиці 3.2 перестановки, дорівнює 12. Це вказує на те, що показники для встановлення та підтримки тактового синхронізму трохи нижчі, в порівнянні з перестановками з таблиці 2.2. Окрім того, це означає, що для накопичення певної, заздалегідь визначеної, кількості значущих моментів модуляції (відновлення) для перестановки з таблиці 3.2 потрібно на 33% більше часу порівняно з перестановками, що наведені в таблиці 2.2.

3.2.2. Розпізнавання синхрокомбінації

Для підвищення надійності отриманих даних, будемо використовувати мажоритарну обробку, яку також реалізовано в попередньому розділі. Звернемо увагу на те, що цей метод обробки передбачає повторення та накопичення синхрокомбінацій. Нехай l буде коефіцієнтом накопичення, що визначає кількість накопичених бітових фрагментів, де довжина фрагмента дорівнює довжині синхрокомбінації. Крім того, коефіцієнт накопичення є непарним числом. Мажоритарна обробка визначає уточнену послідовність R. Відповідно до цієї процедури кожна помилка з кратністю до $(l-1)/2$ включно, що виникає у відповідних бітах отриманих фрагментів, коригується. Таким чином, максимальна загальна кількість бітових помилок, які може виправити мажоритарна обробка, становить $((l-1)/2)\cdot n = ((l-1)/2)\cdot M\lceil \log_2 M \rceil$ серед отриманих ln біт.

3.2.3. Кореляційна обробка

Після мажоритарної обробки, приймач визначає відстані Хеммінга від уточненої послідовності R до кожного циклічного зсуву синхрокомбінації. Якщо визначена відстань Хеммінга менша або дорівнює d_{lim} для певного циклічного зсуву синхрокомбінації, приймач ідентифікує уточнену послідовність R з цим зсувом.

Так як для вибраної в якості синхрокомбінації перестановки з таблиці 3.2 відстань Хеммінга між синхрокомбінацією та її циклічними зсувами не менше 12, то кожна помилка в уточненій послідовності R з кратністю до $d_{lim}=5$ не призводить до помилки.

3.2.4. Імовірність бітової помилки в уточненій послідовності R

Припустимо, що канал зв'язку двійковий симетричний з незалежними бітовими помилками. Тоді результат мажоритарної обробки дозволяє зменшити ймовірність бітової помилки до наступного значення:

$$p_0^* = \sum_{i=(l+1)/2}^{l} C_l^i p_0^i (1-p_0)^{l-i}. \qquad (3.1)$$

У випадках, коли $l \geq 1027$ (це значення l необхідне за високої ймовірності бітової помилки), стандартними засобами виконати обчислення за допомогою формули (3.1) неможливо. Тому визначимо апроксимацію за ймовірності бітової помилки p_0^*.

Теорема 3.2. *Імовірність бітової помилки після мажоритарної обробки* l *фрагментів можна приблизно оцінити наступним чином:*

$$p_0^* \cong \Phi(x) + \frac{1-2p_0}{6\sqrt{lp_0(1-p_0)}} \cdot (x^2-1) \cdot \varphi(x), \qquad (3.2)$$

де $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt$ *– інтегральна функція Лапласа,*

$\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ *– диференціальна функція Лапласа,*

$x = -\sqrt{l}\,\frac{0.5-p_0}{\sqrt{p_0(1-p_0)}}$ *– ймовірність бітової помилки каналу* $p_0 \in (0;0.5)$.

У такому разі точність апроксимації задається наступним чином:

$$|\varepsilon| \leq \frac{1}{12\pi p_0(1-p_0)} \cdot \frac{l}{(l-2)(l-3)} + \frac{(1-2p_0)^2}{9\pi p_0(1-p_0)} \cdot \frac{1}{l} + \\ + \frac{1-2p_0}{4\sqrt{2\pi p_0(1-p_0)}} \cdot \frac{l}{(l-2)(l-3)\sqrt{l-4}} + \frac{(1-2p_0)^2}{9\pi p_0(1-p_0)} \cdot \frac{l}{(l-1)(l-2)(l-3)}. \qquad (3.3)$$

Доведення. Розподіл $F_l(x)$ можна апроксимувати наступним чином (див. формули (36) і (40) з [112]):

$$F_l(x) \cong \Phi(x) - \frac{\theta_3}{3!\sqrt{l}} \cdot H_2(x) \cdot \varphi(x),$$

де θ_3 – третій момент розподілу Чебишева-Ерміта;

$H_2(x) = x^2 - 1$ – поліном Чебишева-Ерміта, $H_k(x) = (-1)^k \varphi^{(k)}(x) / \varphi(x)$.

Для розподілу Бернуллі з нульовим середнім та одиничною дисперсією $\theta_3 = \alpha_3$, де $\alpha_3 = -(1-2p_0)\Big/\sqrt{p_0(1-p_0)}$ – третій момент (асиметрія) розподілу. Звідси випливає формула (3.2).

Отримуємо точність апроксимації (3.3), підставляючи вираз для абсолютного четвертого моменту функції розподілу Бернуллі $\beta_4 + 3 = 1\Big/\big(p_0(1-p_0)\big)$, а також вирази для третього, четвертого та п'ятого абсолютних моментів стандартного нормального розподілу, поділених на $\sqrt{2\pi}$, $B_3 = 2/\pi$, $B_4 = 3\Big/\sqrt{2\pi}$, $B_5 = 8/\pi$ у формулу (40) з роботи [112]. *Теорему доведено.*

Використовуючи формулу (3.3), можемо визначити та представити оцінку точності апроксимації (3.2) для $l \geq 1027$ за різних значень p_0 (таблиця 3.3).

Таблиця 3.3 – **Оцінка точності апроксимації (3.2) для** $l \geq 1027$

p_0	0,4	0,43	0,45	0,47	0,475	0,48	0,485	0,49	0,495
$\lvert\varepsilon\rvert \leq$	$1.15\cdot10^{-4}$	$1.1\cdot10^{-4}$	$1.07\cdot10^{-4}$	$1.05\cdot10^{-4}$		$1.04\cdot10^{-4}$			

3.3. Імовірнісні показники встановлення циклового синхронізму

3.3.1. Імовірність правильної синхронізації

Імовірність правильної синхронізації – це ймовірність помилок в $\nu \leq d_{lim}$ бітах у R:

$$P_{true}\left(n, d_{lim}, p_0, l\right) = \sum_{v=0}^{d_{lim}} C_n^v \left(p_0^*\right)^v \left(1-p_0^*\right)^{n-v}. \qquad (3.4)$$

Для синхрокомбінації з таблиці 3.2, $P_{true}\left(24,5,p_0,l\right) = \sum_{v=0}^{5} C_{24}^v \left(p_0^*\right)^v \left(1-p_0^*\right)^{24-v}$.

Графіки $P_{true}\left(24,5,p_0,l\right)$ для $p_0 \in \{0.1, 0.2, 0.3, 0.4, 0.43, 0.45, 0.47, 0.475, 0.48, 0.485, 0.49, 0.495\}$ в залежності від $l = 1,3,5,...,1001$ зображені на рисунку 3.3.

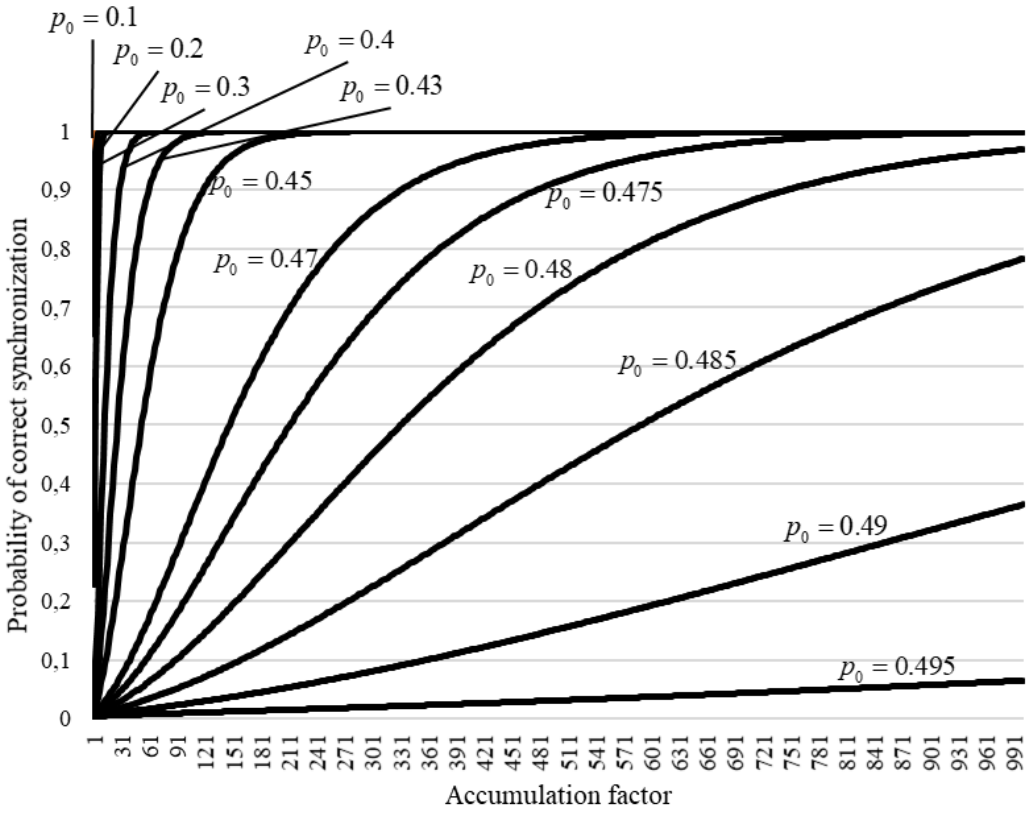


Рисунок 3.3 – Залежності ймовірності правильної синхронізації від коефіцієнта накопичення за різних імовірностей бітових помилок

Зауважимо, що ймовірність бітової помилки апріорно не відома, таким чином, процедура циклової синхронізації повинна бути адаптивною. На основі залежностей, показаних на рисунку 3.3, коефіцієнт накопичення може змінюватися в широких межах для досягнення заданої ймовірності правильної синхронізації. Іншими словами, щоб досягти $P_{true} \geq 0.9997$ за $p_0 = 0.1$ необхідний коефіцієнт накопичення $l = 3$, за $p_0 = 0.3$ необхідний $l = 19$, а за $p_0 = 0.45$ необхідний $l = 305$. Метод циклової синхронізації на основі кореляційної обробки забезпечує накопичення n-бітних фрагментів з каналу зв'язку та відповідну мажоритарну та кореляційну обробку. Максимальне значення коефіцієнта накопичення l визначається максимальним значенням імовірності бітової помилки, на яку розрахована система синхронізації, а також заданою при проектуванні системи мінімальною ймовірністю встановлення синхронізму. Якщо немає синхронізації на максимумі l, система синхронізації генерує сигнал про збій каналу. Як зазначено

нижче, для $n=24$, $d_{lim}=5$, та за заданих обмежень $P_{true}\geq 0.9997$ і $p_0\leq 0.495$, максимальне значення коефіцієнта l обмежено значенням 30603.

Крім того, під час розробки процедури встановлення циклової синхронізації, необхідно враховувати випадки хибної синхронізації, ймовірність якої зростає зі збільшенням ймовірності p_0 та зі зменшенням коефіцієнта накопичення l. За цих умов, процедура циклової синхронізації повинна гарантувати, що ймовірність встановлення хибного синхронізму не може перевищувати заданий поріг для будь-якої довільно високої ймовірності бітової помилки.

3.3.2. Імовірність хибної синхронізації

Визначивши ймовірність правильної синхронізації, визначимо ймовірність хибної синхронізації.

Оцінку зверху ймовірності хибної синхронізації можна розрахувати приблизно наступним чином:

$$P_{false}\left(n,d_{lim},p_0,l\right)\leq 1-P_{true}\left(n,d_{lim},p_0,l\right). \tag{3.5}$$

Ця оцінка відображає можливість перетворення переданої синхрокомбінації в будь-яку з $2^n-\sum_{v=0}^{d_{lim}}C_n^v$ послідовностей. Разом з тим, хибна синхронізація відбувається, якщо завада в каналі зв'язку перетворює передану синхрокомбінацію у будь-яку з послідовностей, розташованих на відстані Хеммінга, що не перевищує d_{lim}, до всіх можливих циклічних зсувів синхрокомбінації (у геометричній інтерпретації, у сферах n-вимірного простору з центрами в точках, що відповідають циклічним зсувам синхрокомбінації та радіусом d_{lim}). Кількість таких можливих послідовностей дорівнює $(n-1)\sum_{v=0}^{d_{lim}}C_n^v$. Під час обчислення відношення $(n-1)\sum_{v=0}^{d_{lim}}C_n^v \Big/ \left(2^n-\sum_{v=0}^{d_{lim}}C_n^v\right)$, наприклад, для $M=8$, $n=24$, та $d_{lim}=5$, отримаємо значення 0.076. Врахуємо, що точки всередині сфер радіуса d_{lim} не є

рівноймовірними за $0 < p_0 < 0.5$. Таким чином, оцінка за формулою (3.5) є неточною. Далі розрахуємо точну ймовірність хибної синхронізації.

Теорема 3.3. *Імовірність хибної синхронізації*

$$P_{false}\left(n, d_{lim}, p_0, l\right) = \sum_{j=1}^{n-1}\left(\sum_{v=d_{ij}-d_{lim}}^{d_{ij}} C_{d_{ij}}^{v} \left(\sum_{w=0}^{v-d_{ij}+d_{lim}} C_{n-d_{ij}}^{w} \left(p_0^*\right)^{v+w} \left(1-p_0^*\right)^{n-(w+v)} \right) \right). \qquad (3.6)$$

*Доведення***.** Імовірність помилки в уточненій послідовності R, що призведе до неправильного прийняття рішення та встановлення хибної синхронізації – ймовірність появи в уточненій послідовності R будь-якого вектора помилки, який перетворює синхрокомбінацію π_i у будь-який її циклічний зсув $\left(1 \le j \le n-1\right)$ з точністю до d_{lim} бітів.

Як зазначено раніше, d_{ij} – відстань Хеммінга від синхрокомбінації π_i до її циклічного зсуву на j біт $\pi_i\left(j\right)$, $1 \le j \le n-1$. У такому разі помилка перетворює перестановку π_i в її циклічний зсув $\pi_i\left(j\right)$, якщо вона містить v помилок в d_{ij} бітах, в яких ці послідовності відрізняються, причому $d_{ij} - d_{\lim} \le v \le d_{i,j}$. Крім того, в інших$\left(n - d_{ij}\right)$ бітах, можлива поява w більшої кількості бітових помилок, причому $0 \le w \le v - \left(d_{ij} - d_{\lim}\right)$. Останнє обмеження пов'язане з тим, що для хибної синхронізації перестановка, змінена помилкою, не повинна відрізнятися більше ніж на d_{lim} біт від циклічного зсуву перестановки π_i.

Імовірність описаної вище події виражається наступним чином:

$$P_{false}\left(n, d_{lim}, p_0, l\right) = \sum_{j=1}^{n-1}\left(\sum_{v=d_{ij}-d_{lim}}^{d_{ij}} \left(\begin{array}{l} C_{d_{ij}}^{v} \left(p_0^*\right)^{v} \left(1-p_0^*\right)^{d_{ij}-v} \times \\ \times \sum\limits_{w=0}^{v-d_{ij}+d_{lim}} C_{n-d_{ij}}^{w} \left(p_0^*\right)^{w} \left(1-p_0^*\right)^{n-d_{ij}-w} \end{array} \right) \right). \qquad (3.7)$$

Групуючи фактори в формулі (3.7), отримуємо вираз для обчислення ймовірності хибної синхронізації (3.6). *Теорему доведено.*

Зауваження 3.4. *Для синхрокомбінації із таблиці 3.2 з* $d_{lim} = 5$*, значення* $d_{ij} = 12$ *зустрічаються в 19 випадках, значення* $d_{ij} = 14$ *зустрічаються у двох*

випадках та значення $d_{ij}=16$ *зустрічаються також у двох випадках (див. нормалізований графік АКФ на рисунку 3.2). Таким чином, після групування доданків формули (3.6) для синхрокомбінації з таблиці 3.2 виконується:*

$$P_{false}(24,5,p_0,l)=19\sum_{v=7}^{12}C_{12}^{v}\left(\sum_{w=0}^{v-7}C_{12}^{w}\left(p_0^*\right)^{v+w}\left(1-p_0^*\right)^{24-v-w}\right)+$$
$$+2\sum_{v=9}^{14}C_{14}^{v}\left(\sum_{w=0}^{v-9}C_{10}^{w}\left(p_0^*\right)^{v+w}\left(1-p_0^*\right)^{24-v-w}\right)+2\sum_{v=11}^{16}C_{16}^{v}\left(\sum_{w=0}^{v-11}C_{8}^{w}\left(p_0^*\right)^{v+w}\left(1-p_0^*\right)^{24-v-w}\right). \quad (3.8)$$

Зауваження 3.5. *Під час виконання числових розрахунків для ймовірності хибної синхронізації* $P_{false}(n,d_{lim})$*, в модельному прикладі дослідження була використана формула (3.8).*

Графіки $P_{false}(24,5,p_0,l)$ для $p_0\in\{0.1,0.2,0.3,0.4,0.43,0.45,0.47,0.475,0.48,0.485,0.49,0.495\}$ в залежності від $l=1,3,5,...,1001$ зображено на рисунку 3.4.

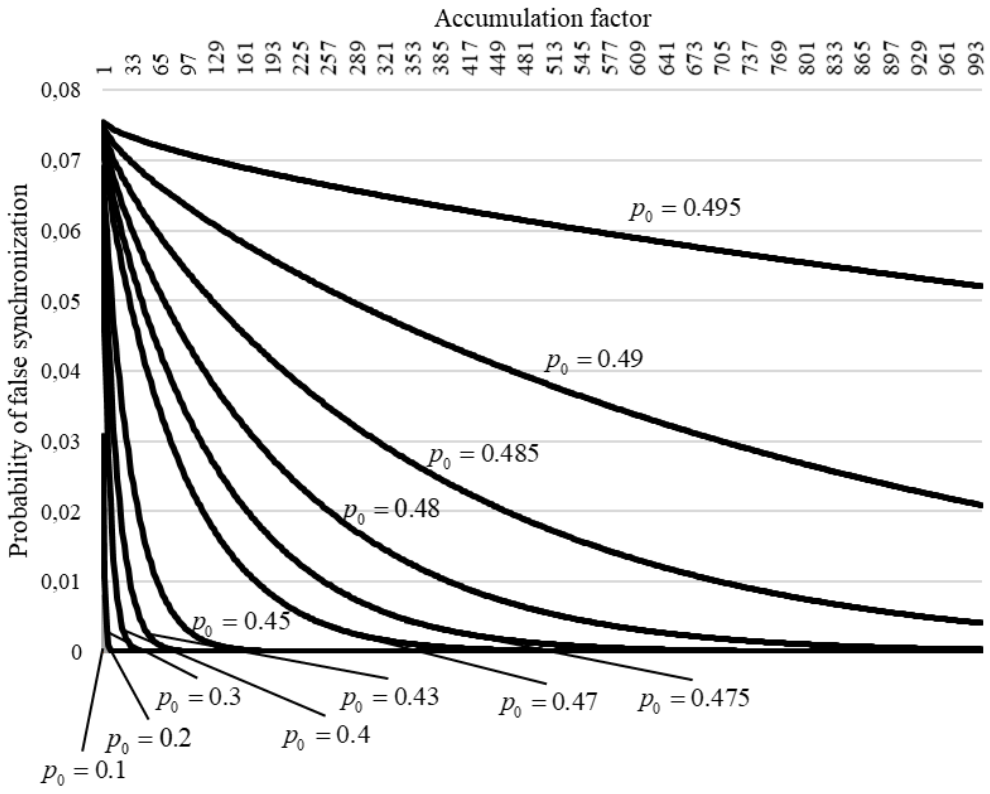


Рисунок 3.4 – Залежності ймовірності хибної синхронізації за різних імовірностей бітових помилок від коефіцієнта накопичення

З аналізу графіків, наведених на рисунку 3.4, випливає наступне, що під час збільшення коефіцієнта накопичення l, формування уточненої послідовності R та обчислення відстаней від неї до всіх циклічних зсувів синхрокомбінації, може виникнути ситуація, коли за невідомої ймовірності бітової помилки p_0 ймовірність хибної синхронізації перевищить своє максимально допустиме значення. Наприклад, якщо аналіз уточненої послідовності починається з коефіцієнта накопичення $l=3$ за $p_0=0.4$, ймовірність хибної синхронізації для $l=3$ становить $P_{false}=0.046$. Для зменшення цієї ймовірності використовується наступний підхід.

3.3.3. Зменшення ймовірності хибної синхронізації

Приймач групує отримані з каналу зв'язку фрагменти в K блоки по l фрагментів. Потім уточнені послідовності R_k, де $k\in[1,K]$, обчислюються незалежно для кожного блоку. Синхронізм встановлюється, якщо кожна з послідовностей R_k відповідає одному і тому ж циклічному зсуву синхрокомбінації.

Тоді ймовірність правильної синхронізації (див. формулу (3.4)):

$$P_{true}\left(n,d_{lim},p_0,l,K\right)=P_{true}^{K}\left(n,d_{lim},p_0,l\right)=\left(\sum_{v=0}^{d_{lim}}C_n^v\left(p_0^*\right)^v\left(1-p_0^*\right)^{n-v}\right)^K. \qquad (3.9)$$

З формули (3.6) ймовірність хибної синхронізації змінюється на:

$$P_{false}\left(n,d_{lim},p_0,l,K\right)=\sum_{j=1}^{n-1}\left(\sum_{v=d_{ij}-d_{lim}}^{d_{ij}}C_{d_{ij}}^v\sum_{w=0}^{v-d_{ij}+d_{lim}}C_{n-d_{ij}}^w\left(p_0^*\right)^{v+w}\left(1-p_0^*\right)^{n-v-w}\right)^K. \qquad (3.10)$$

Формулу (3.10) можна пояснити наступним чином. Імовірність перетворення перестановки π_i в її циклічний зсув $\pi_i(j)$ дорівнює (див. формулу (3.6))

$P_{false\,j}\left(n,d_{lim},p_0,l\right)=\sum_{v=d_{ij}-d_{lim}}^{d_{ij}}C_{d_{ij}}^v\left(\sum_{w=0}^{v-d_{ij}+d_{lim}}C_{n-d_{ij}}^w\left(p_0^*\right)^{v+w}\left(1-p_0^*\right)^{n-(w+v)}\right)$. Імовірність того, що завада в каналі зв'язку перетворить перестановку π_i в її циклічний зсув $\pi_i(j)$ у K блоках, дорівнює $P_{false\,j}^{K}\left(n,d_{lim},p_0,l\right)$. Таким чином, ймовірність хибної синхронізації для K блоків задана як:

$$P_{false}\left(n,d_{lim},p_0,l,K\right)=P^K_{false1}\left(n,d_{lim},p_0,l\right)+P^K_{false2}\left(n,d_{lim},p_0,l\right)+\ldots+$$

$$+P^K_{false(n-1)}\left(n,d_{lim},p_0,l\right)=\sum_{j=1}^{n-1}P^K_{false\,j}\left(n,d_{lim},p_0,l\right)$$

Графіки $P_{true}\left(24{,}5,p_0,l,K\right)$ та $P_{false}\left(24{,}5,p_0,l,K\right)$ за $p_0=0.495$ і $K\in\{1,2,3,4\}$ в залежності від коефіцієнта накопичення зображені на рисунках 3.5 – 3.6.

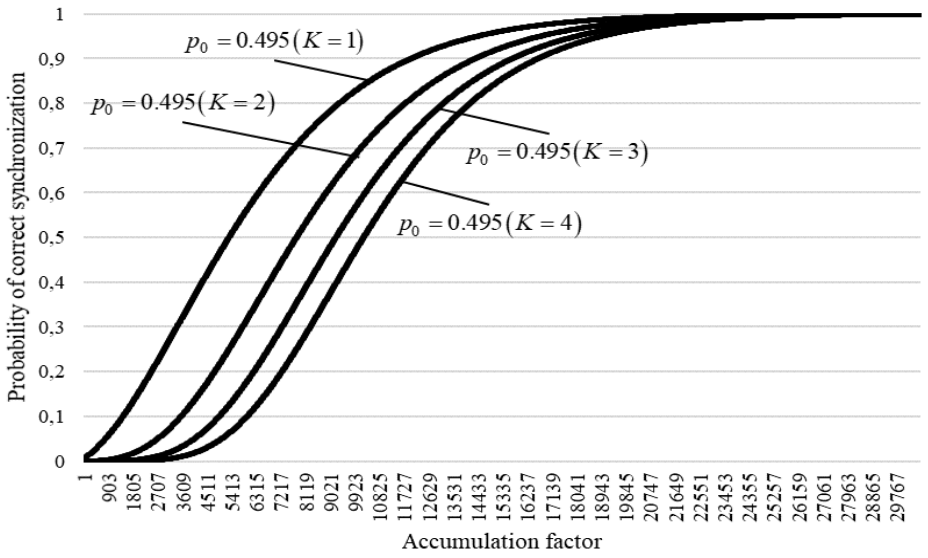


Рисунок 3.5 – Залежності ймовірності правильної синхронізації від коефіцієнта накопичення за $p_0=0.495$ та різних значень $K\in\{1,2,3,4\}$

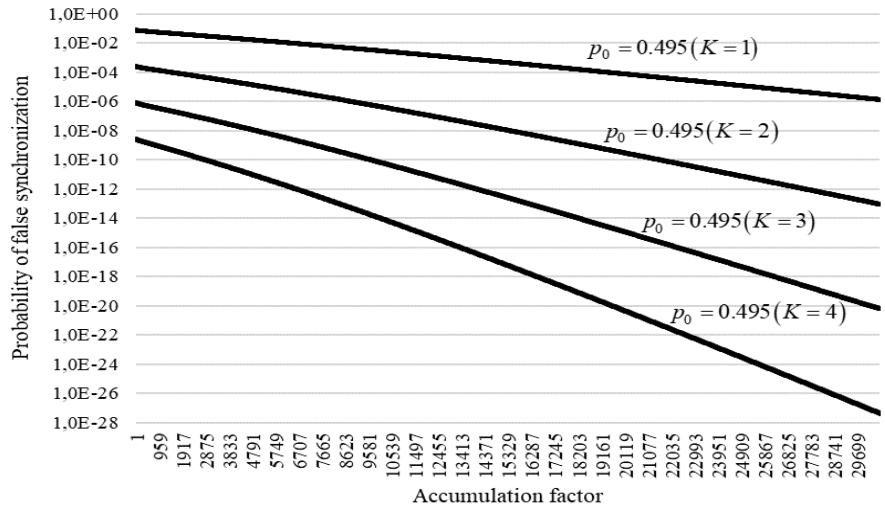


Рисунок 3.6 – Залежності ймовірності хибної синхронізації від коефіцієнта накопичення за $p_0=0.495$ та різних значень $K\in\{1,2,3,4\}$

Графіки залежностей $P_{true}(24,5,p_0,l,K)$ та $P_{false}(24,5,p_0,l,K)$ за $p_0 = 0.495$ і $K \in \{1,2,3,4\}$ в залежності від кількості накопичених фрагментів, $L = l \cdot K = 1,3,5,...,30603$ зображені на рисунках 3.7 та 3.8.

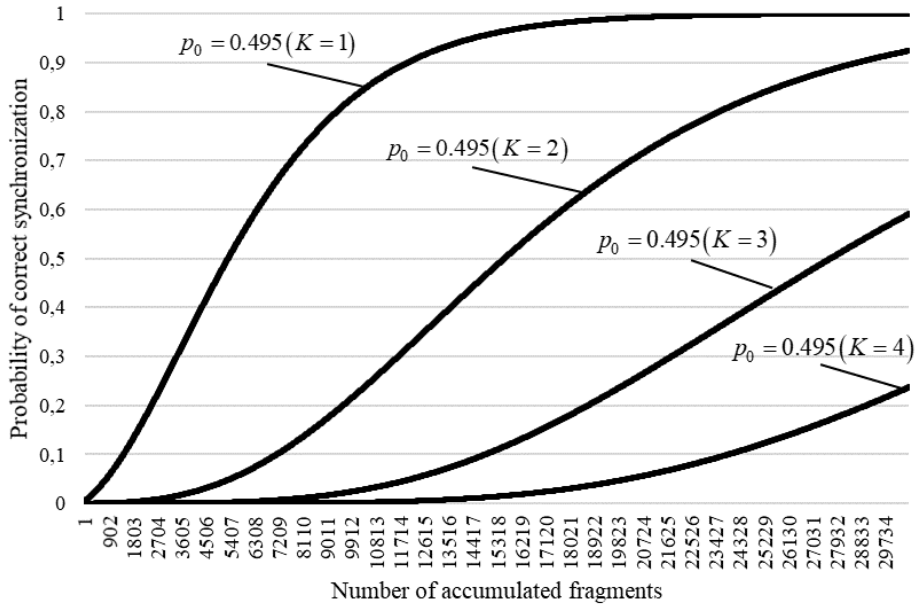


Рисунок 3.7 – Залежності ймовірності правильної синхронізації від кількості накопичених фрагментів за $p_0 = 0.495$ та різних значень $K \in \{1,2,3,4\}$

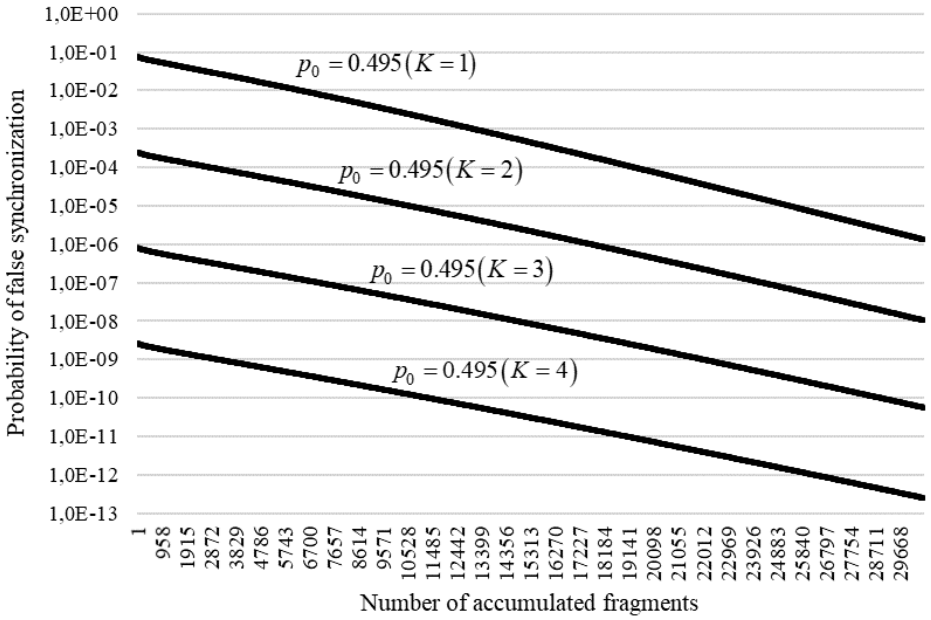


Рисунок 3.8 – Залежності ймовірності хибної синхронізації від кількості накопичених фрагментів за $p_0 = 0.495$ та різних значень $K \in \{1,2,3,4\}$

На рисунках 3.6 та 3.8 показано, що значення $P_{false}\left(24,5,p_0,l,4\right)\le 2.68\cdot 10^{-9}$, $P_{false}\left(24,5,p_0,l,3\right)\le 8.15\cdot 10^{-7}$, $P_{false}\left(24,5,p_0,l,2\right)\le 2.48\cdot 10^{-4}$, а $P_{false}\left(24,5,p_0,l,1\right)\le 7.55\cdot 10^{-2}$ за $p_0=0.495$.

Далі виконаємо оцінку зверху для ймовірності $P_{false}\left(n,d_{lim},p_0,l,K\right)$ обчисленої за допомогою формули (3.10).

Теорема 3.4. *Оцінку ймовірності хибної синхронізації* $P_{false}\left(n,d_{lim},p_0,l,K\right)$ *можна виразити наступним чином:*

$$P_{false}\left(n,d_{lim},p_0,l,K\right)\le \max_s\left\{p\left(n,d_{lim},K,s\right)\right\}, \tag{3.11}$$

де

$$p\left(n,d_{lim},K,s\right)=\sum_{j=1}^{n-1}\left(\sum_{v=d_{ij}-d_{lim}}^{d_{ij}} C_{d_{ij}}^{v}\sum_{w=0}^{v-d_{ij}+d_{lim}} C_{n-d_{ij}}^{w}F\left(n,v+w,s\right)\right)^{K}. \tag{3.12}$$

За даних умов,

$$F\left(n,v+w,s\right)=\begin{cases}\left(s/Nn\right)^{v+\omega}\left(1-s/Nn\right)^{n-(v+\omega)}, N\left(v+w\right)\le s;\\ \left(\left(s+1\right)/Nn\right)^{v+\omega}\left(1-\left(s+1\right)/Nn\right)^{n-(v+\omega)}, N\left(v+w\right)\ge s+1;\end{cases}$$

N *– кількість інтервалів, на які розділені інтервали* $\left[z/n;\left(z+1\right)/n\right]$*;*

z *– натуральне число,* $d-d_{lim}\le z\le\lceil n/2\rceil-1$*;*

s *– натуральне число,* $N\left(d-d_{lim}\right)\le s\le\lceil Nn/2\rceil-1$.

*Доведення***.** Розглянемо поведінку функції . Графіки $f\left(n,v+w,p_0^*\right)$ за $n=24$ та $v+w=\left\{7,9,12,15,17\right\}$ в залежності від p_0^* зображені на рисунку 3.9, де діапазон значень $p_0^*\in\left(0,1\right)$ використовується для покращення візуалізації залежності.

На наступному кроці необхідно визначити точку p_0^* в якій функція $f\left(n,v+w,p_0^*\right)$ отримує максимальне значення. Нехай $g\left(n,v+w,p_0^*\right)=\ln f\left(n,v+w,p_0^*\right)=\left(v+w\right)\ln p_0^*+\left(n-v-w\right)\ln\left(1-p_0^*\right)$. Потім,

$g'\left(n,v+w,p_0^*\right)=\left(v+w-np_0^*\right)\Big/ p_0^*\left(1-p_0^*\right)$, а $g'\left(n,v+w,p_0^*\right)=0$ за $p_0^*=\left(v+w\right)\Big/n$.

Таким чином, отримуємо $\max\left(f\left(n,v+w,p_0^*\right)\right)=f\left(n,v+w,\left(v+w\right)\Big/n\right)$.

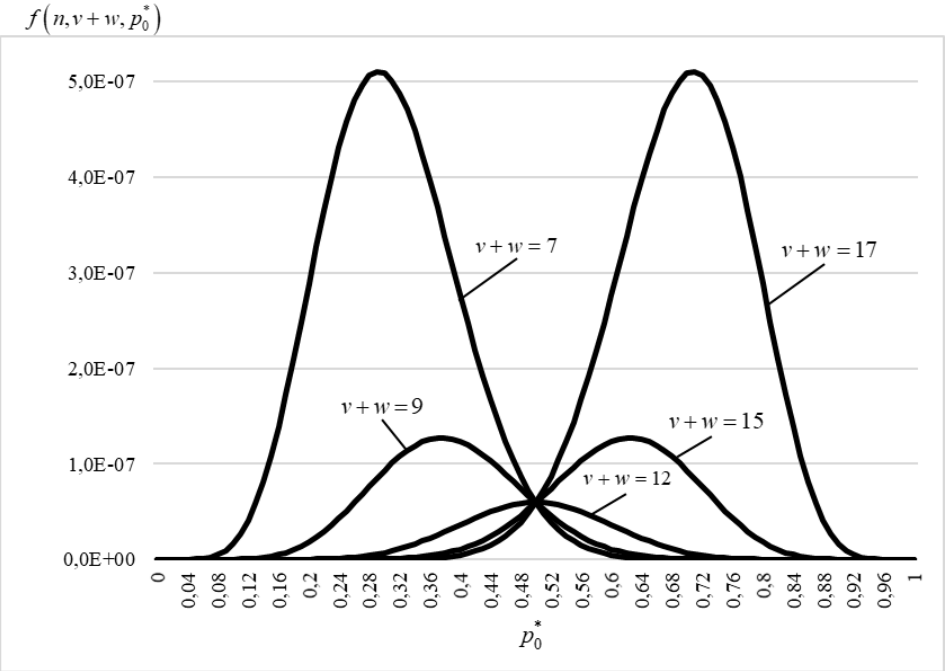


Рисунок 3.9 – Залежності $f\left(n,v+w,p_0^*\right)$ від p_0^* за $n=24$ та $v+w=\{7,9,12,15,17\}$

Для модельного прикладу, максимум функцій $f\left(n,v+w,p_0^*\right)$, $d-d_{lim}\le v+w<n/2$ знаходиться в точці $p_0^*\in\left[7/24;1/2\right)$. За $v+w\ge n/2$ функції монотонно зростають на інтервалі $\left(7/24;1/2\right)$.

Кожна функція $f\left(n,v+w,p_0^*\right)$ є монотонною на інтервалах $\left(z/n;\left(z+1\right)/n\right)$, де z – натуральне число, $d-d_{lim}\le z\le\left\lceil n/2\right\rceil-1$. Кожен інтервал $\left[z/n;\left(z+1\right)/n\right]$ ділиться на N інтервалів $\left[z/n+i/Nn;z/n+\left(i+1\right)/Nn\right]0\le i\le N-1$. Для кожного натурального числа s, $N\left(d-d_{lim}\right)\le s\le\left\lceil Nn/2\right\rceil-1$, та значення параметра $p_0^*\in\left[s/Nn;\left(s+1\right)/Nn\right]$, застосовуються такі обмеження:

$$f\left(n,\nu+w,p_0^*\right)=\left(p_0^*\right)^{\nu+w}\left(1-p_0^*\right)^{n-\nu-w}\le F\left(n,\nu+w,s\right),\qquad(3.13)$$

де $F\left(n,\nu+w,s\right)=\begin{cases}\left(s/Nn\right)^{\nu+\omega}\left(1-s/Nn\right)^{n-(\nu+\omega)},N\left(\nu+w\right)\le s;\\ \left(\left(s+1\right)/Nn\right)^{\nu+\omega}\left(1-\left(s+1\right)/Nn\right)^{n-(\nu+\omega)},N\left(\nu+w\right)\ge s+1.\end{cases}$

Оцінки (3.12) для $p\left(n,d_{lim},K,s\right)$ та (3.11) для $P_{false}\left(n,d_{lim},p_0,l,K\right)$ формуються послідовно на основі співвідношень (3.10) та (3.13). *Теорему доведено.*

Для модельного прикладу розділимо відрізок $p_0^*\in\left[7/24,1/2\right]$ на $\lceil Nn/2\rceil-N\left(d-d_{lim}\right)=\lceil 2\cdot 24/2\rceil-2\left(12-5\right)=10$ відрізків з кроком $1/48$, тоді $s=14,15,\ldots,23$. Із формул (3.11) та (3.12) випливають наведені в таблиці 3.4 оцінки зверху ймовірності $P_{false}\left(24,5,p_0,l,K\right)$ для $K\in\left\{1,2,3,4\right\}$.

Таблиця 3.4 – **Оцінки зверху ймовірності хибної синхронізації для** $n=24$**,** $d_{lim}=5$

K	**1**	**2**	**3**	**4**
$P_{false}\left(24,5,p_0,l,K\right)\le$	$8.03\cdot10^{-2}$	$2.80\cdot10^{-4}$	$9.78\cdot10^{-7}$	$3.42\cdot10^{-9}$

Наведені оцінки вказують на те, що навіть за $p_0^*\to 1/2$ можна обрати початкове значення K, за якого ймовірність хибної синхронізації задовольнятиме заданим вимогам.

Зауваження 3.6. *Формули (3.9) і (3.10), а також формули (3.4) і (3.6) визначають імовірності правильної та хибної синхронізації «в точці», тобто для окремого експерименту із заданими значеннями коефіцієнта накопичення l та числа блоків K, ігноруючи процедуру послідовного збільшення l та варіації K.*

3.3.4. Оцінки інтервальних імовірностей синхронізації

Результат синхронізації для поточного значення l залежить від результатів синхронізації для попередніх значень l. Це пов'язано з тим, що статистика в накопичених фрагментах в межах одного блоку може дещо відрізнятися..

Імовірність P_{true_final} правильної синхронізації після досягнення коефіцієнта накопичення, рівного l , оцінюється знизу ймовірністю правильної синхронізації для фіксованого значення l наступним чином:

$$P_{true\_final}\left(n,d_{lim},p_0,l,K\right)\geq P_{true}\left(n,d_{lim},p_0,l,K\right). \tag{3.14}$$

Імовірність P_{false_final} хибної синхронізації синхронізації після досягнення коефіцієнта накопичення, рівного l за $K=1$, оцінюється з наведеного вище наступним чином:

$$P_{false\_final}\left(n,d_{lim},p_0,l,K\right)=\sum_{i=1}^{K}P_{false\_sum}\left(n,d_{lim},p_0,l_{\min}\left(i\right),l_{\max}\left(i\right),i\right), \tag{3.15}$$

де

$$P_{false\_sum}\left(n,d_{lim},p_0,l_{\min}\left(i\right),l_{\max}\left(i\right),i\right)\leq\sum_{j=l_{\min}(i)}^{l_{\max}(i)}P_{false}\left(n,d_{lim},p_0,j,i\right) \tag{3.16}$$

$P_{false\_sum}\left(n,d_{lim},p_0,l_{\min}\left(i\right),l_{\max}\left(i\right),i\right)$ визначає ймовірність хибної синхронізації для i блоків; $l_{\min}\left(i\right)\leq l\leq l_{\max}\left(i\right)$; $l_{\min}\left(K\right)=1$; $l_{\max}\left(1\right)=l$.

Зауваження 3.7. *$P_{true\_final}\left(n,d_{lim},p_0,l,K\right)$ та $P_{false\_final}\left(n,d_{lim},p_0,l,K\right)$ фактично є оцінками кумулятивної функції розподілу (КФР)[113] числа $L_{fr}=Kl$ до встановлення правильного або хибного синхронізму відповідно.*

3.3.5. Вибір значень K та l

Значення K та межі інтервалів $\left[l_{\min}\left(i\right),l_{\max}\left(i\right)\right]$ вибираються для задовільнення нерівності $P_{false\_final}\left(n,d_{lim},p_0,l,K\right)\leq P_{false\_\max}$, де $P_{false\_\max}$ – граничне значення ймовірності хибної синхронізації, а $p_{0\_\max}$ – граничне значення ймовірності бітової помилки.

На основі формули (3.15) обмеження для окремих доданків $P_{false\_final}\left(n,d_{lim},p_0,l,K\right)$ визначаються наступним чином:

$$P_{false\_sum}\left(n,d_{lim},p_0,l_{\min}\left(i\right),l_{\max}\left(i\right),i\right)\leq\gamma_i\cdot P_{false\_\max}, \tag{3.17}$$

причому , $\gamma_i \geq 0$, $\sum_{i=1}^{K} \gamma_i = 1$.

Звернемо увагу на те, що визначення меж інтервалів $\left[l_{\min}(i), l_{\max}(i)\right]$ починається з $i = 1$.

Тоді $l_{\max}(1) = \min(l) : P_{true}(n, d_{lim}, p_0, l, 1) \geq P_{true\_\min}$. Таким чином, за аналогією з формулою (14), виконується наступна нерівність: $P_{true\_final}(n, d_{lim}, p_0, l_{\max}(1), K) \geq P_{true\_min}$. Для модельного прикладу з $n = 24$, за $P_{true\_\min} = 0.9997$ та $p_{0\_\max} = 0.495$, отримуємо $l_{\max}(1) = 30603$.

Значення $l_{\min}(i) = \min(l) : P_{false\_sum}(n, d_{lim}, p_0, l, l_{\max}(i), i) \leq \gamma_i \cdot P_{false\_\max}$.

За допомогою значень $l_{\min}(i)$ визначаються верхні межі $l_{\max}(i+1)$:

$$l_{\max}(i+1) = \left[l_{\min}(i) \cdot \left(i/(i+1)\right)\right] - prtsgn,$$

де

$$prtsgn = \begin{cases} 0 \; if \; \left[l_{\min}(i) \cdot \left(i/(i+1)\right)\right] - \textit{непарний}; \\ 1 \; \textit{інакше.} \end{cases}.$$

Метод, який використовується для вибору значень K, $l_{\min}(i)$, $l_{\max}(i)$, $i \in [1; K]$, застосовується до будь-якого заданого обмеження.

Для модельного прикладу $P_{false\_\max} = 3 \cdot 10^{-4}$. Відповідно до таблиці 3.4 та формули (3.16) отримуємо $P_{false\_sum}(24, 5, 0.495, 1, 30603, 4) \leq 1.05 \cdot 10^{-4}$. Таким чином, $K = 4$, а $P_{false\_sum}(24, 5, 0.495, 1, l_{\max}(4), 4) \ll \sum_{i=1}^{3} P_{false\_sum}(n, d_{lim}, p_0, l_{\min}(i), l_{\max}(i), i)$.

Припустимо що $\gamma_1 = 1/8$, $\gamma_2 = 3/4$, $\gamma_3 = 1/8$, тоді межі інтервалів $\left[l_{\min}(i), l_{\max}(i)\right]$ приймають значення, що наведені в таблиці 3.5.

Кількість накопичених фрагментів становить $L_{fr} = Kl$, а кількість накопичених бітів дорівнює $L = Kln$.

Таблиця 3.5 – **Межі інтервалів** $\left[l_{\min}(i), l_{\max}(i)\right]$

i	**1**	**2**	**3**	**4**
$l_{\min}(i)$	30549	10457	2779	1
$l_{\max}(i)$	30603	15273	6971	2083

На рисунку 3.10 наведено графіки залежностей оцінок, розрахованих за формулами (3.14) і (3.15), імовірностей правильної та хибної синхронізації від кількості накопичених фрагментів за заданих обмежень для $K = \{1,2,3,4\}$ та визначених значень точок переходу між ними.

Графіки з рисунку 3.10 визначають адаптивний процес синхронізації за невідомого рівня ймовірності бітової помилки $p_0 \le 0.495$ для $P_{true} \ge 0.9997$ та $P_{false} \le 3 \cdot 10^{-4}$.

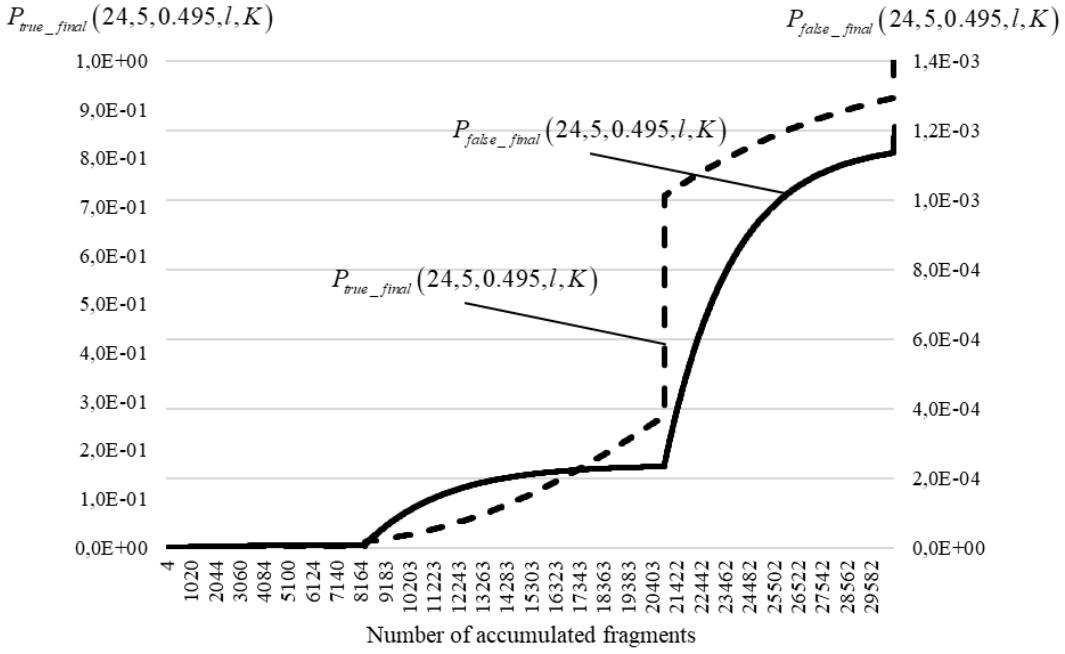


Рисунок 3.10 – Залежності оцінок імовірностей правильної та хибної синхронізації від кількості накопичених фрагментів для процесу адаптивної синхронізації

3.4. Оцінка ефективності методу

Для оцінки методу циклової синхронізації на основі кореляційної обробки розроблена програмна модель Python для передавання інформації з факторіальним кодуванням даних. Параметри моделювання наведені в таблиці 3.6.

Таблиця 3.6 – **Параметри моделювання**

Канал зв'язку	Двійковий симетричний
Бітові помилки	Незалежні
Ймовірність бітової помилки	$0.4 \le p_0 < 0.498$
Кількість експериментів	10^3, 10^4
Синхрокомбінація	$\pi = (0,1,7,3,2,5,4,6) = (000,001,111,011,010,101,100,110)$
K	4
$l_{\min}(i)$ **значення**	30549, 10457, 2779, 1
$l_{\max}(i)$ **значення**	30603, 15273, 6971, 2083

Алгоритм моделі методу циклової синхронізації наведено на рисунку 3.11.

В якості синхрокомбінації використано перестановку, наведену в таблиці 3.2. Межі інтервалів , $l_{\max}(i)$ визначені вище для $P_{true} \ge 0.9997$, $P_{false} \le 3 \cdot 10^{-4}$ за будь-якого заданого значення ймовірності бітової помилки $p_0 \le 0.495$. Крім того, проведено 10 000 експериментів для точного визначення статистичних показників циклової синхронізації.

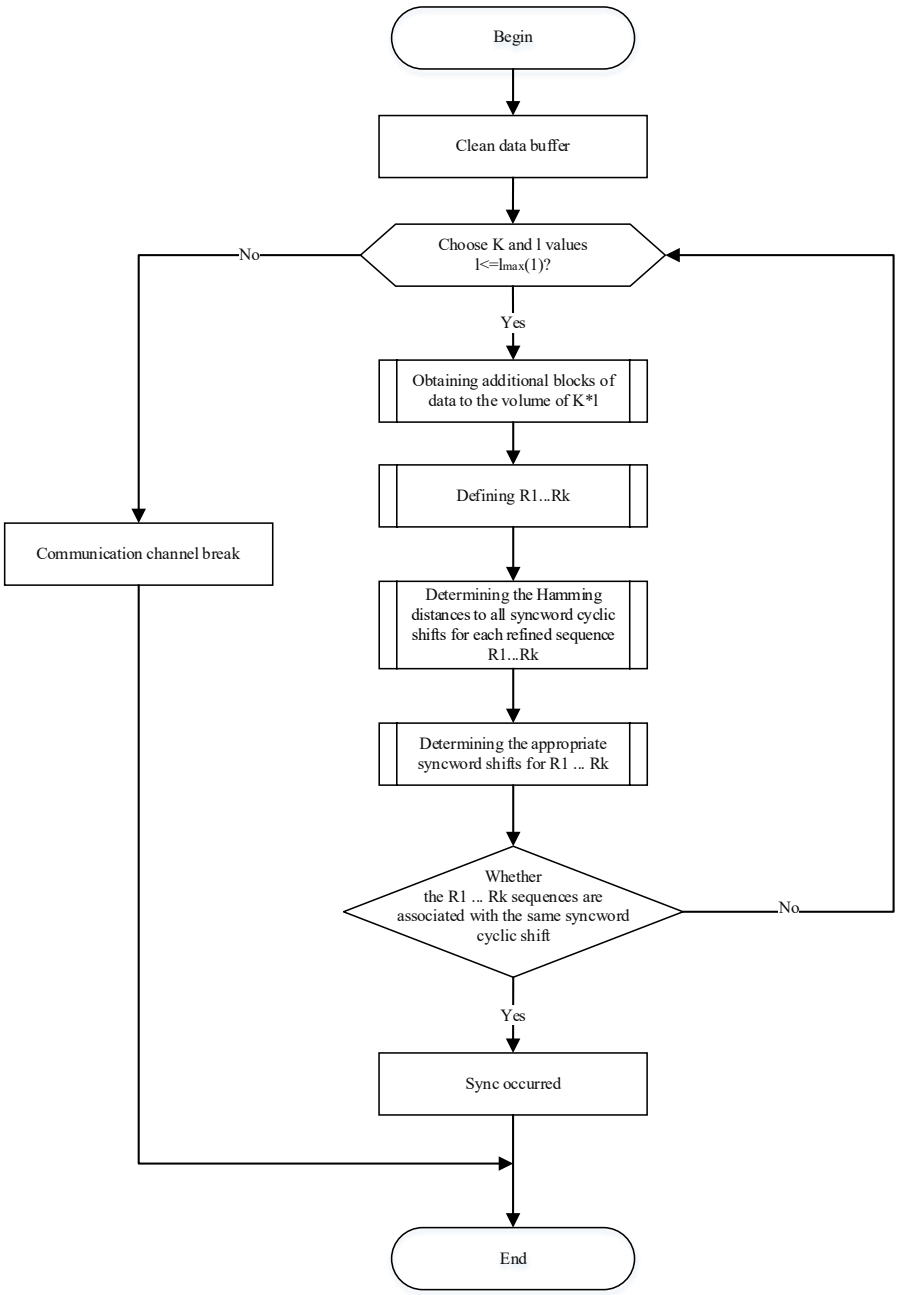


Рисунок 3.11 – Алгоритм методу циклової синхронізації на основі кореляційної обробки

3.4.1. Результати та їх обговорення

На графіках з рисунку 3.12 наведено залежності відносної частоти $W_{true}\left(24,5,p_0,l,K\right)$ встановлення правильного циклового синхронізму від фіксованого значення коефіцієнта накопичення l за ймовірності бітової помилки

$p_0 = 0.4$ та кількості блоків $K \in \{1;2;3;4\}$. Кожне значення $W_{true}(24,5,p_0,l,K)$ перевірено експериментально через 1000 експериментів.

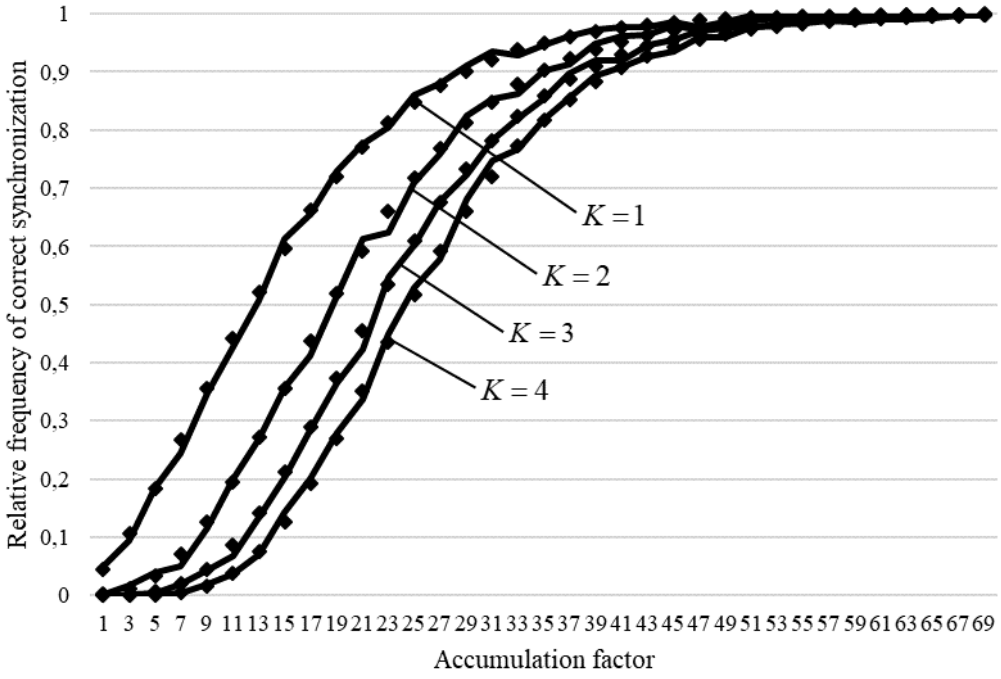


Рисунок 3.12 – Залежності $W_{true}(24,5,p_0,l,K)$ від фіксованого значення l

Маркери, що показані на рисунку 3.12, також ілюструють відповідні графіки теоретичних залежностей P_{true} від l згідно з формулою (3.9).

Зазначимо, що теоретична та експериментальна залежності, що показані на рисунку 3.12, мають однаковий розподіл. Крім того, отримані p-значення по критерію Пірсона, інтерпретованого згідно з методологією, описаною в [114], близькі до одиниці. Такі ж висновки були зроблені щодо ймовірності хибної синхронізації. Це свідчить про коректність побудованої моделі.

На рисунку 3.13 наведено графік експериментально отриманої КФР $W_{true\_final}(24,5,p_0,l,K) = \sum_{K^* l^* \le Kl} W_{true}(24,5,p_0,l^*,K^*)$ за кількістю оброблених фрагментів $L_{fr} = Kl$ для $p_0 = 0.495$ через 10 000 експериментів.

На рисунку 3.13 пунктирна лінія показує оцінку знизу КФР , розраховану за формулою (3.14), а штрихпунктирна лінія представляє експериментально

визначену КФР для методу на основі поділу кодового слова, який описано в другому розділі, за $p_0 = 0.495$.

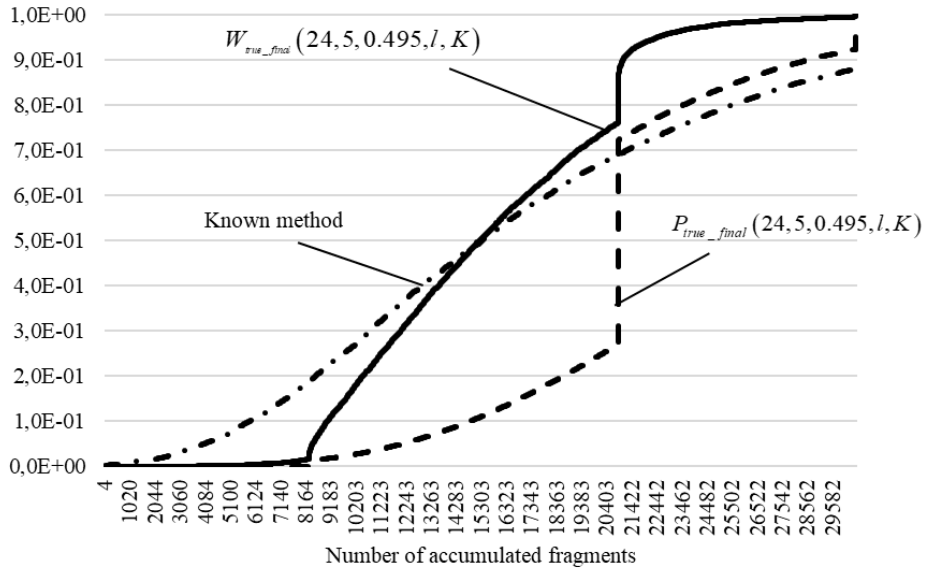
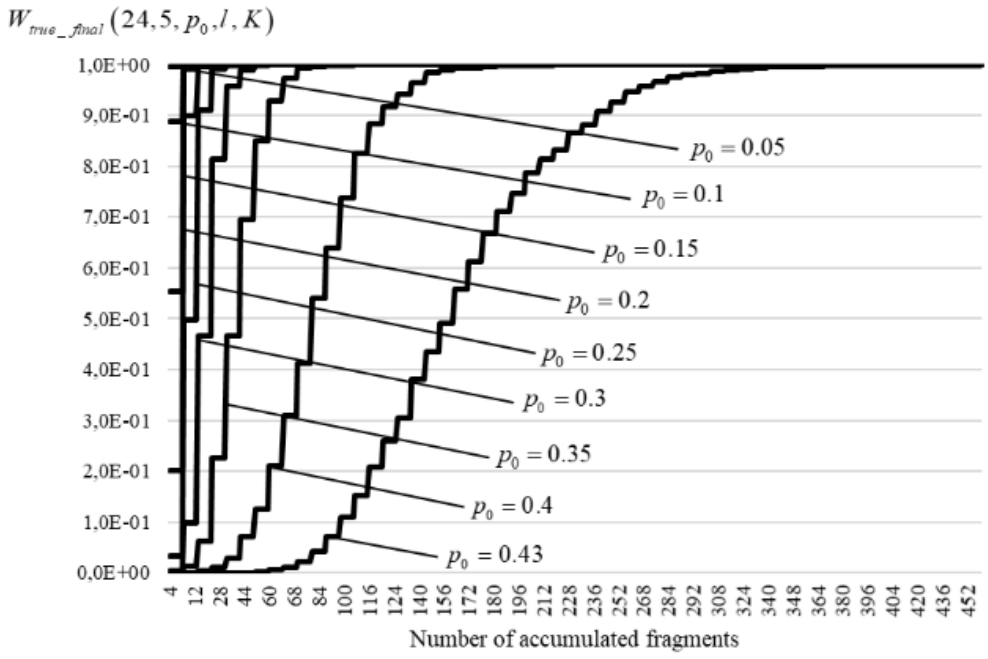


Рисунок 3.13 – КФР за ймовірності бітової помилки $p_0 = 0.495$

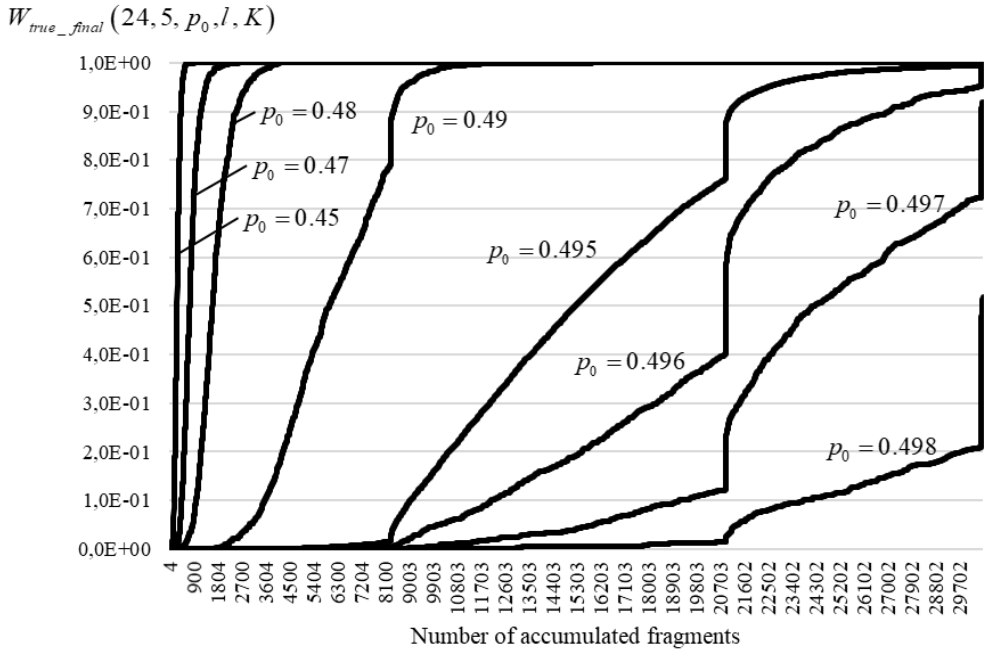
Зауваження 3.8. *Під час експериментального аналізу розробленого методу в 10 000 експериментах хибної синхронізації не виявлено, за ймовірності бітової помилки* $p_0 = 0.495$.

Графіки на рисунку 3.13 підтверджують відносно приблизну оцінку, отриману за формулою (3.14). Результати також свідчать про те, що метод на основі кореляційної обробки дозволяє досягти заданих імовірнісних показників циклової синхронізації швидше, ніж метод на основі поділу кодового слова. Проте визначені діапазони значень кількості накопичених фрагментів ($L_{fr} = 1\ldots14984$ на наведеному графіку), де відносна частота правильної синхронізації для методу на основі поділу кодового слова є вищою. У цьому випадку метод на основі поділу кодового слова не відповідає вимогам щодо ймовірності встановлення хибного синхронізму. Зауважимо, що існує можливість комбінованого застосування як методу на основі поділу кодового слова, так і методу на основі кореляційної обробки, але це виходить за рамки поточного дослідження.

Необхідно розглянути поведінку КФР $W_{true\_final}(24,5,p_0,l,K)$ від $L_{fr}=Kl$ за ймовірності бітової помилки $p_0 \le 0.498$ (рисунок 3.14). Тут значення K та l були визначені для $p_0 = 0.495$, як показано в таблиці 3.5. Для кожного значення ймовірності бітової помилки проведено 1000 експериментів.



(**a**)



(**b**)

Рисунок 3.14 – КФР за: (a) $p_0 \le 0.43$**; (b)** $0.45 \le p_0 \le 0.498$

Графіки на рисунку 3.14 визначають імовірнісні показники встановлення правильного синхронізму за допомогою методу циклової синхронізації за різних значеннях імовірності бітової помилки p_0.

На рисунку 3.14(b) показано, що коли ймовірність бітової помилки перевищує граничне значення (для даного модельного зразку це $p_0 = 0.495$), можуть бути порушені вимоги щодо ймовірності правильної синхронізації. Таким чином, $W_{true\_final}(24,5,0.496,30603,1) = 0.997$, $W_{true\_final}(24,5,0.497,30603,1) = 0.920$, а $W_{true\_final}(24,5,0.498,30603,1) = 0.519$. За такої умови відносна частота встановлення хибного циклового синхронізму становить $W_{false\_final}(24,5,0.496,30603,1) = 0.003$, $W_{false\_final}(24,5,0.497,30603,1) = 0.001$, а $W_{false\_final}(24,5,0.498,30603,1) = 0.033$. Крім того, відносна частота збою каналу зв'язку (ситуація, в якій $L_{fr} = Kl = 30603$ синхронізм не знайдений) становить 0.003 за $p_0 = 0.496$, 0.080 за $p_0 = 0.497$ та 0.465 за $p_0 = 0.498$.

Важливо відзначити, що для максимальної ефективності розробленого методу, ймовірність бітової помилки має бути максимально точно спрогнозована. Проте процедура вибору моментів для зміни значень K та l може бути адаптивною, залежати від імовірності бітової помилки, і може визначатися, наприклад, відповідно до середньої кількості накопичених фрагментів для реалізації циклової синхронізації.

На цьому етапі можна визначити та продемонструвати (рисунок 3.15) залежність середнього значення кількості накопичених фрагментів до встановлення циклового синхронізму $\overline{L_{fr}}$ від імовірності бітової помилки p_0.

Рисунок 3.15 ілюструє експоненціальний характер збільшення середнього значення кількості накопичених фрагментів до встановлення циклового синхронізму (залежно від імовірності бітової помилки). Ця залежність може бути використана для адаптивної зміни моментів змін значень K і l в процесі роботи методу циклової синхронізації.

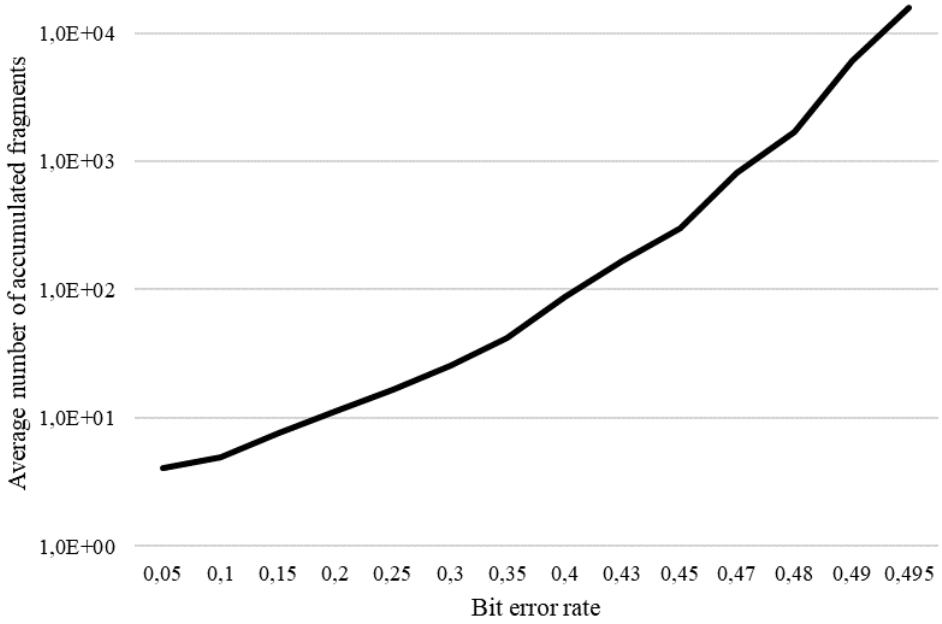


Рисунок 3.15 – Середня кількість накопичених фрагментів для встановлення циклового синхронізму в залежності від імовірності бітової помилки

Згідно з методом циклової синхронізації на основі кореляційної обробки, який описано на початку підрозділу 3.2, уточнені послідовності R_k, $k \in [1, K]$, обчислюються незалежно. Кожен біт цих послідовностей обчислюється за мажоритарним принципом на основі відповідних бітів повторно отриманих фрагментів.

У побудованій моделі, для якої результати показані на рисунках 3.13 – 3.15, фрагменти, які визначають послідовності R_k залишаються незмінними, коли значення K і l змінюються. Зі збільшенням l, нові фрагменти лише доповнюють набори фрагментів, які визначають послідовності R_k (для константи K, кожне наступне значення l призводить до того, що кожен набір фрагментів доповнюється лише двома новими фрагментами). Таким чином, кількість доповнених фрагментів відносно невелика, наприклад 8 фрагментів для $l = 50$ та $K = 4$, що становить лише 4%. Таким чином, результат мажоритарної обробки бітів фрагментів має високу

кореляцію з результатом, отриманим на попередньому кроці для меншого значення l. Для зменшення цієї кореляції пропонується наступний підхід.

3.4.2. Вплив процедури перемішування на отримані результати.

Система синхронізації, отримуючи необхідну на певному кроці кількість фрагментів, накопичує їх у одному буфері та випадковим чином порівну розподіляє між K блоками. У разі, якщо синхронізацію на цьому кроці не встановлено, приймач додатково приймає нові фрагменти, записуючи їх у загальний буфер. Потім система синхронізації знову розподіляє фрагменти випадковим чином між K блоками. Такий підхід щодо обробки фрагментів будемо називати обробкою з перемішуванням, оскілки зазначений випадковий розподіл фрагментів між блоками можна досягнути перемішуванням фрагментів у загальному масиві в буфері.

На рисунку 3.16 показаний графік експериментально отриманої КФР $W_{true\_final}(24,5,p_0,l,K)$ для запропонованого підходу обробки з перемішуванням у порівнянні з підходом без перемішування. Тут усі параметри моделі такі ж, як і в моделі попереднього прикладу, результати якої показано на рисунку 3.13.

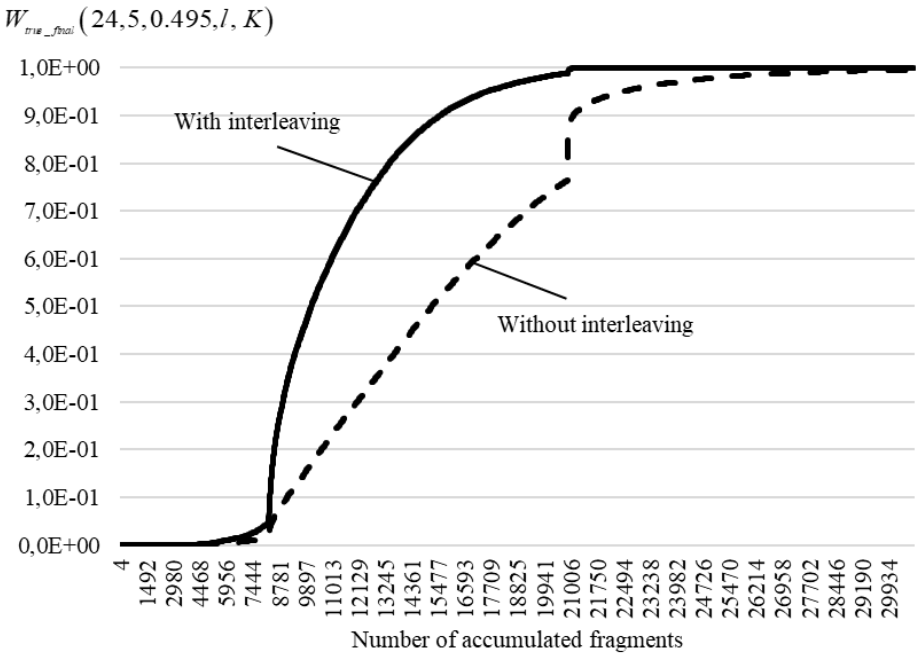


Рисунок 3.16 – КФР за $p_0 = 0.495$ (з перемішуванням)

Наведені на рисунку 3.16 графіки свідчать про ефективність процедури перемішування фрагментів перед обчисленням значень R_k. Її використання дозволяє зменшити кількість прийнятих з каналу зв'язку фрагментів, необхідних для встановлення синхронізму. Як наслідок, зменшується час встановлення сенсу зв'язку.

На рисунку 3.17 наведено залежності середнього значення кількості накопичених фрагментів до встановлення циклового синхронізму $\overline{L_{fr}}$ від імовірності бітової помилки p_0, для методу на основі кореляційної обробки з перемішуванням та без перемішування.

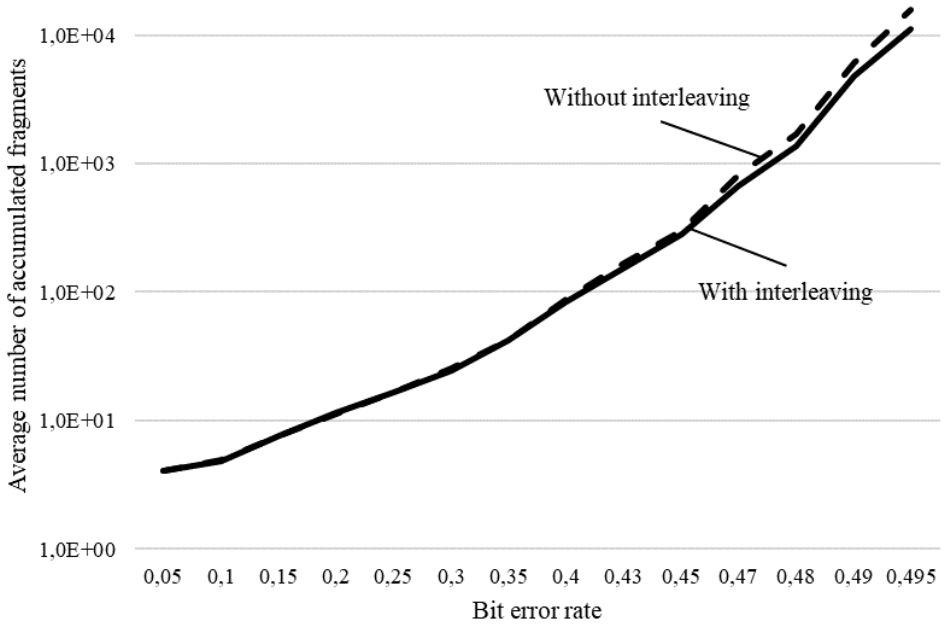


Рисунок 3.17 – Середня кількість накопичених фрагментів для встановлення циклового синхронізму в залежності від імовірності бітової помилки (з перемішуванням)

Графік на рисунку 3.17 демонструє позитивний ефект перемішування. Цей ефект стає особливо помітним зі збільшенням ймовірності бітової помилки.

Наприклад, за $p_0 = 0.495$ середня кількість накопичених фрагментів, отриманих за допомогою підходу перемішування, зменшилася на 29,4%, тобто 11 060 фрагментів порівняно з 15 667 фрагментами.

Однак реалізація цього ефекту вимагає достатніх ресурсів буферної пам'яті для реалізації запропонованого методу циклової синхронізації з перемішуванням накопичених фрагментів. Розмір буфера, який використовується для зберігання отриманих фрагментів, залежить від кількості фрагментів. Для реалізованої моделі на мові програмування Python на рисунку 3.18 показано розміри буферів, необхідні для збільшення кількості накопичених фрагментів.

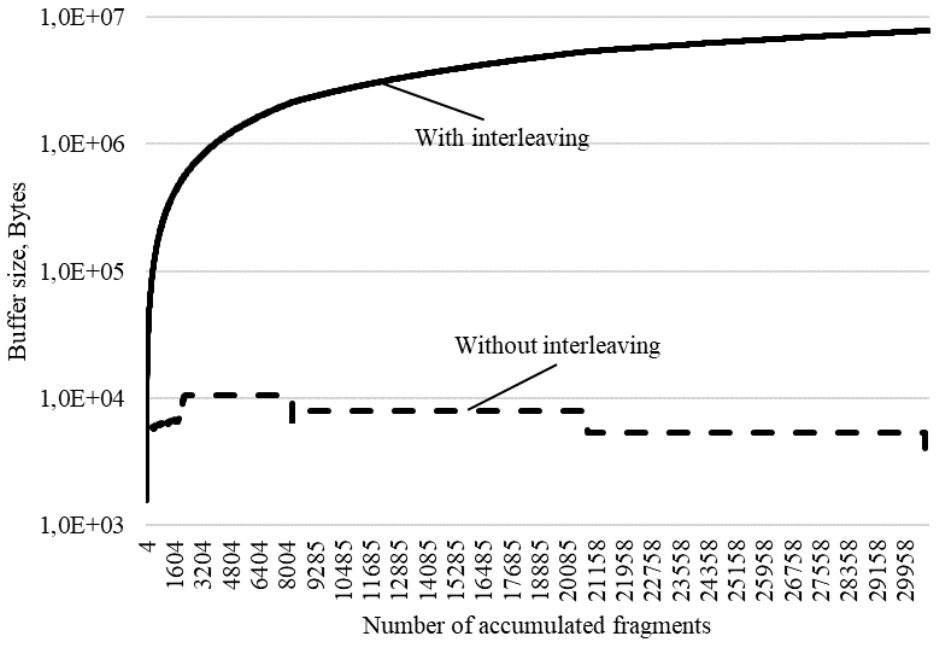


Рисунок 3.18 – Залежність розміру буферу пам’яті для зберігання фрагментів від їхньої кількості

Таким чином, процедура перемішування накопичених фрагментів дозволяє виконати обмін пам’яті на кількість необхідних для встановлення синхронізму фрагментів і, як наслідок, на час встановлення сенсу зв’язку.

3.5. Система циклової синхронізації

Структурну схему системи циклової синхронізації наведено на рисунку 3.19.

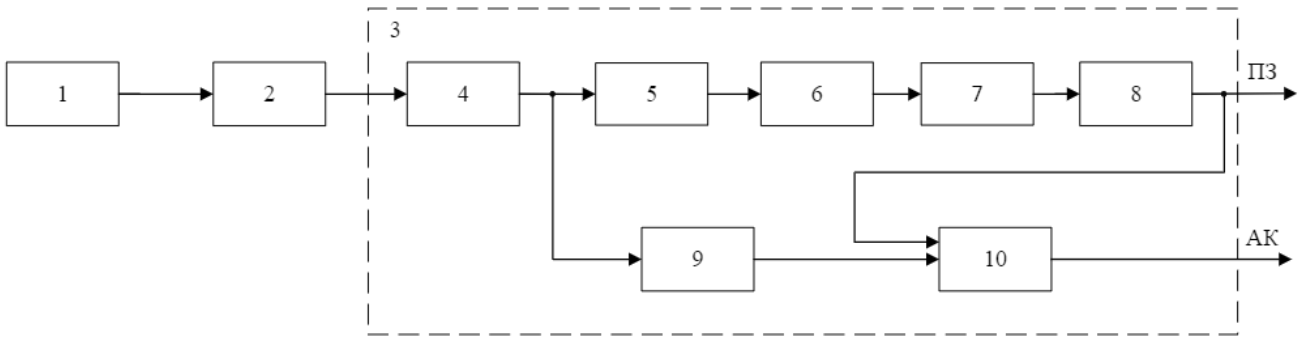


Рисунок 3.19 – Схема системи циклової синхронізації на основі розробленого методу з використанням процедури перемішування

Джерело 1 формує та циклічно передає в канал зв’язку 2 синхрокомбінацію.

Приймач синхрокомбінації 3 записує в накопичувач 4 прийняті з каналу зв'язку 2 $K \cdot l$ фрагментів по n біт.

Для формування уточнених послідовностей R_k, $k \in [1, K]$, фрагменти з накопичувача 4 надходять у блок перемішування 5, де їх випадковим чином перемішують, порівну ділять на K блоків і передають у блок 6 мажоритарної обробки.

У блоці 6 мажоритарної обробки для кожного блоку фрагментів незалежно обчислюють уточнену послідовність R_k, $k \in [1, K]$. Кожен біт цієї послідовності обчислюють за мажоритарним принципом на основі відповідних біт прийнятих фрагментів. Таким чином, якщо i-ті біти фрагментів містять більше «одиниць», i-му біту уточненої послідовності присвоюють значення «одиниці», в іншому разі – «нуля».

Далі в блоці 7 для кожної уточненої послідовності R_k обчислюють відстані Хеммінга до всіх циклічних зсувів синхрокомбінації. Якщо для якогось із зсувів ця

відстань не перевищує значення $d_{lim} = \lfloor (d-1)/2 \rfloor$, приймають рішення про відповідність уточненої послідовності цьому зсуву.

Значення зсувів синхрокомбінації, що відповідають уточненим послідовностям R_k, надходять у блок 8, де виконують перевірку, чи всі зсуви однакові. Якщо це так, система циклової синхронізації приймає рішення про встановлення синхронізму. У такому разі, блок 8 перевірки на відповідність всіх R_k одному і тому ж зсуву синхрокомбінації видає сигнал «Пошук синхронізму завершений» (ПЗ). Його відправляють зворотним каналом на передавальну станцію, а канал даних переходить зі стану встановлення з'єднання в стан перенесення призначених для користувача даних.

Якщо значення зсувів синхрокомбінації, що відповідають уточненим послідовностям R_k, не однакові, пару значень K і l змінюють, а з каналу зв'язку в накопичувач 4 додатково приймають фрагменти до досягнення їхньої кількості значення $K \cdot l$. Далі масив з $K \cdot l$ фрагментів знову ділять випадковим чином на K блоків і повторюють усі операції виявлення синхрокомбінації.

Число накопичених фрагментів послідовно збільшують до деякого, заздалегідь заданого, порогу $l_{\max}(1)$. Пошук синхронного стану триває або до завершення процедури пошуку, або до вичерпання ліміту часу на пошук синхронізму. Для цього система циклової синхронізації містить лічильник числа циклів 9, вхід якого підключено до виходу накопичувача 4, а вихід – до першого входу блоку 10 контролю часу встановлення синхронізму. Другий вхід блоку 10 контролю часу встановлення синхронізму підключено до виходу блоку 8 перевірки на відповідність одному і тому ж зсуву синхрокомбінації для всіх R_k, який виводить сигнал ПЗ. Якщо після закінчення ліміту часу, відведеного на пошук синхронізму, його не знайдено, то блок контролю часу встановлення синхронізму 10 формує сигнал «Аварія каналу». Процедуру пошуку припиняють.

Технічним результатом застосування запропонованої системи в комунікаційній системі є зменшення часу сеансу зв'язку на виконання функції встановлення синхронізму за циклами.

3.6. Висновки

У третьому розділі розроблено метод циклової синхронізації для систем зв'язку з нероздільним факторіальним кодуванням, який за рахунок використання як синхрокомбінації перестановки, яка володіє максимальним значенням мінімальної відстані Хеммінга від її двійкового представлення до всіх її циклічних зсувів, мажоритарної обробки прийнятих фрагментів для виправлення, в прийнятій послідовності, помилок та кореляційної обробки отриманої послідовності на основі якої формується рішення про встановлення синхронізму, дозволяє досягти зменшення часу входження в синхронізм і підвищення стійкості комунікаційної системи в умовах впливу завад високої інтенсивності.

Розроблено та застосовано алгоритм, що реалізує розроблений метод на основі кореляційної обробки .

Технічним результатом застосування запропонованого методу в комунікаційній системі є зменшення часу сеансу зв'язку на виконання функції встановлення циклового синхронізму.

Результати дослідження свідчать про те, що за допомогою кореляційної та мажоритарної обробки розроблений метод ефективно реалізує циклову синхронізацію в системах передавання даних за несприятливих шумових умов. Процедуру вибору параметрів системи синхронізації реалізовано в модельній системі передавання даних з імовірністю бітової помилки $p_0 \le 0.495$, імовірністю правильної синхронізації $P_{true} \ge 0.9997$, та ймовірністю хибної синхронізації $P_{false} \le 3 \cdot 10^{-4}$.

Виявлено, що метод на основі кореляційної обробки зменшує обсяг отриманих даних, що, у свою чергу, скорочує час, необхідний для встановлення з'єднання. Це призводить до збільшення часу передавання даних користувача.

Крім того, виявлено, що перемішування накопичених фрагментів реалізує подальше зменшення необхідної кількості отриманих фрагментів ціною

збільшення використання пам’яті на приймачі. Для наведеного прикладу з імовірністю… $p_0 = 0.495$, середню кількість накопичених фрагментів зменшено з 15 667 до 11 060 фрагментів.

Розроблений метод циклової синхронізації на основі кореляційної обробки може бути ефективно реалізований у системах, які використовують нероздільне факторіальне кодування, але не обмежується цими системами. Потенційним застосуванням методу циклової синхронізації на основі перестановки є неортогональний множинний доступ (NOMA) [115], [116], [117], який є перспективним методом радіодоступу в бездротових комунікаціях нового покоління, зокрема в контексті Інтернет речей.

Основні результати дослідження та розробки методу опубліковано в [65], [66].

4. МЕТОД ДОСТОВІРНОГО ПЕРЕДАВАННЯ ПЕРЕСТАНОВОК У СИСТЕМАХ ЗВ’ЯЗКУ З НЕРОЗДІЛЬНИМ ФАКТОРІАЛЬНИМ КОДУВАННЯМ

4.1. Вступ

У третьому розділі досліджено метод циклової синхронізації для систем зв'язку з нероздільним факторіальним кодуванням, у якому синхрокомбінацією є перестановка, в якій мінімальна відстань Хеммінга до всіх її циклічних зсувів є максимальною. Результати дослідженого методу свідчать про те, що за допомогою кореляційної та мажоритарної обробки запропонований метод циклової синхронізації ефективно реалізує циклову синхронізацію в системах передавання даних з короткими пакетами за несприятливих шумових умов.

Очевидно, що одним із головних призначень будь-якої комунікаційної системи є безпосередньо передача корисної інформації (даних користувача) із забезпеченням її цілісності та захисту. Ця робота розглядає системи зв’язку з нероздільним факторіальним кодуванням та їх використання в середовищах з завадами високої інтенсивності. Тому однією з задач дисертаційного дослідження є розробка та вивчення методу достовірного передавання кодових слів нероздільного факторіального коду – перестановок – за високої ймовірності виникнення помилок у каналі.

У четвертому розділі представлено результати розробки та дослідження достовірного методу передавання перестановок каналами зв'язку з імовірністю бітової помилки, близькою до 0,5. Такий технічний результат забезпечено завдяки використанню циклічних зсувів перестановки-носія для представлення (кодування) кожного елемента перестановки, що передається. Перестановка-носій має максимальне значення мінімальної відстані Хеммінга від її двійкового представлення до всіх її циклічних зсувів. Мажоритарна та кореляційна обробка фрагментів, отриманих з каналу зв'язку, дозволяє реалізувати достовірне передавання перестановок в умовах високоінтенсивних канальних завад.

4.2. Опис методу

Метод достовірного передавання перестановок у системах зв’язку з нероздільним факторіальним кодуванням передбачає наступні етапи:

1) передавач послідовно видає в канал зв'язку перестановку W довжиною N, іменовану словом. Кожен елемент перестановки, іменований літерою L_j, $1 \le j \le N$, є циклічним бітовим зсувом перестановки π довжиною M, що володіє максимальним значенням мінімальної відстані Хеммінга від її n-бітного двійкового представлення до всіх її циклічних зсувів (наприклад, для $M = 8$. перестановка $\pi = (000,001,111,011,010,101,100,110)$). Очевидно, що кількість циклічних зсувів перестановки π має бути не меншим за довжину перестановки W: $n \ge N$. Процедурі передавання даних передує процедура встановлення синхронізації за літерами, наприклад, так, як запропоновано в третьому розділі;

2) для кожної літери приймач накопичує прийняті з каналу зв'язку l фрагментів по n біт;

3) для кожної літери незалежно обчислюється уточнена послідовностей R_j, $j \in [1, N]$. Кожен біт цієї послідовності обчислюється за мажоритарним принципом на основі відповідних біт прийнятих фрагментів. Таким чином, якщо i-ті біти фрагментів містять більше "одиниць", i-му біту уточненої послідовності присвоюється значення "одиниці", в іншому разі - "нуля";

4) для кожної уточненої послідовності R_j обчислюються відстані Хеммінга до використовуваних джерелом літер. Якщо до якоїсь із літер ця відстань не перевищує значення $d_{lim} = \lfloor (d-1)/2 \rfloor$, j-му символу слова W встановлюється у відповідність ця літера;

5) якщо всі послідовності R_j, $j \in [1, N]$, відповідають різним використовуваним джерелом літерам, тобто прийняте слово є перестановкою цих літер, і ця перестановка використовується джерелом, слово видають споживачеві. У іншому випадку повторюються всі операції розпізнавання слова, починаючи з п. 2 цього списку;

6) число накопичених фрагментів може послідовно збільшуватися до деякого, заздалегідь заданого порога $l_{\max}$. Якщо після досягнення цього порога слово не розпізнано, процедура приймання завершується, а на вихід системи видається сигнал "Аварія каналу".

Кодові слова нероздільного факторіального коду належать підмножині безлічі перестановок $\{\pi\}$ довжини M, символи яких кодуються рівномірним двійковим кодом з довжиною кодової комбінації $l_r = \lceil \log_2 M \rceil$ [63].

Пояснювати принцип побудови системи надійного передавання будемо на прикладі використання перестановки π довжини $M = 8$ (послідовності десяткових символів множини $\{0,1,2,3,4,5,6,7\}$).

Кожен символ цієї множини кодується рівномірним двійковим кодом з $l_r = \lceil \log_2 M \rceil = 3$ $(n = M \cdot l_r = 24)$, наприклад, як показано в таблиці 4.1.

Таблиця 4.1 – **Схема кодування символів перестановки**

Десятковий запис	0	1	2	3	4	5	6	7
Двійковий запис	000	001	010	011	100	101	110	111

Позначимо через $\pi_i(j)$ циклічний зсув перестановки π_i вліво на j біт, а через d_{ij} – відстань Хеммінга від перестановки π_i до її циклічного зсуву $\pi_i(j)$, при цьому $0 \le i \le M!-1$, $1 \le j \le n-1$. Нехай $d_i = \min_j(d_{ij})$, а $d = \max_i(d_i) = \max_i\left(\min_j(d_{ij})\right)$.

Визначення 4.1. *Літерою L_j будемо називати циклічний зсув $\pi_i(j)$ такої перестановки π_i, в якої відстані Хеммінга до всіх її циклічних зсувів не менші за d: $\forall d_{ij} \ge d$.*

Зауваження 4.1. *У цьому розділі в модельному прикладі для формування літер L_j будемо використовувати перестановку $\pi_i = \{0,1,7,3,2,5,4,6\} = \{000,001,111,011,010,101,100,110\}$, отриману в третьому розділі.*

Визначення 4.2. *Словом W будемо називати перестановку літер L_j.*

Зауваження 4.2. *Для формування слова W можуть бути використані всі або частина літер L_j. Кількість літер у слові визначає його довжину, яку будемо позначати через N, $N \le n$. Відстань Хеммінга між L_0 і L_j будемо позначати через d_j.*

Зауваження 4.3. *У цьому розділі для модельного прикладу з $M = 8$ слово W будемо формувати з 23 літер L_j, $1 \le j \le 23$.*

Для підвищення достовірності прийнятих даних використовуватимемо їх мажоритарну і кореляційну обробку, як це реалізовано в методі циклової синхронізації, представленому в третьому розділі.

Мажоритарна обробка передбачає багаторазове повторення слова та накопичення результату його прийому. Коефіцієнт накопичення $l = 3,5,7\ldots$ дорівнює числу повторених слів. Приймач накопичує фрагменти даних. Довжина кожного фрагмента дорівнює довжині слова. За отриманими l фрагментами формується послідовність R, що складається з N послідовностей R_j, $j \in [1, N]$. Довжина кожної послідовності R_j дорівнює довжині літери. Кожен біт послідовності R обчислюється за відповідними бітами прийнятих фрагментів. Якщо i-ті біти фрагментів містять більше двійкових «одиниць», i-ий біт послідовності R набуває значення «одиниці», інакше – «нуля».

Кореляційна обробка передбачає обчислення відстаней Хеммінга від кожної послідовності R_j, $j \in [1, N]$, до всіх використовуваних передатчиком літер. Якщо до якоїсь із літер ця відстань не перевищує значення $d_{lim} = \lfloor (d-1)/2 \rfloor$, ця літера встановлюється у відповідність послідовності R_j.

Для модельного прикладу відстані Хеммінга між перестановкою $\pi_i = \{0,1,7,3,2,5,4,6\}$ та її бітовими циклічними зсувами не менше $d = 12$. Звідси $d_{lim} = 5$.

Якщо кожній із послідовностей R_j, $j \in [1, N]$, відповідають різні літери джерела, прийняте слово є перестановкою цих літер, і це перестановка використовується джерелом, слово видається споживачеві.

4.3. Імовірнісні показники розпізнавання слова

Імовірність бітової помилки в уточненій послідовності R після мажоритарної обробки l фрагментів, прийнятих з каналу зв'язку з ймовірністю бітової помилки p_0, дорівнює

$$p_0^* = \sum_{i=(l+1)/2}^{l} C_l^i p_0^i (1-p_0)^{l-i}. \qquad (4.1)$$

У разі, коли $l \geq 1027$, для знаходження p_0^* доцільно скористатися апроксимаційною формулою (3.2).

4.3.1. Імовірність правильного та хибного розпізнавання слова

Імовірність правильного розпізнавання літери визначається ймовірністю появи в уточненій послідовності до d_{lim} (включно) помилок:

$$P_{L\_true} = \sum_{\nu=0}^{d_{lim}} C_n^{\nu} \left(p_0^*\right)^{\nu} \left(1-p_0^*\right)^{n-\nu}. \qquad (4.2)$$

Правильне розпізнавання слова довжини N відбувається тоді і тільки тоді, коли всі N літер розпізнано правильно. Імовірність цієї події

$$P_{W\_true} = \left(P_{L\_true}\right)^N. \qquad (4.3)$$

Графіки $P_{W\_true}(l)$ для аналізованого модельного прикладу з $\pi_i = \{0,1,7,3,2,5,4,6\}$ та $N = 23$ за різних значень p_0 представлено на рисунку 4.1.

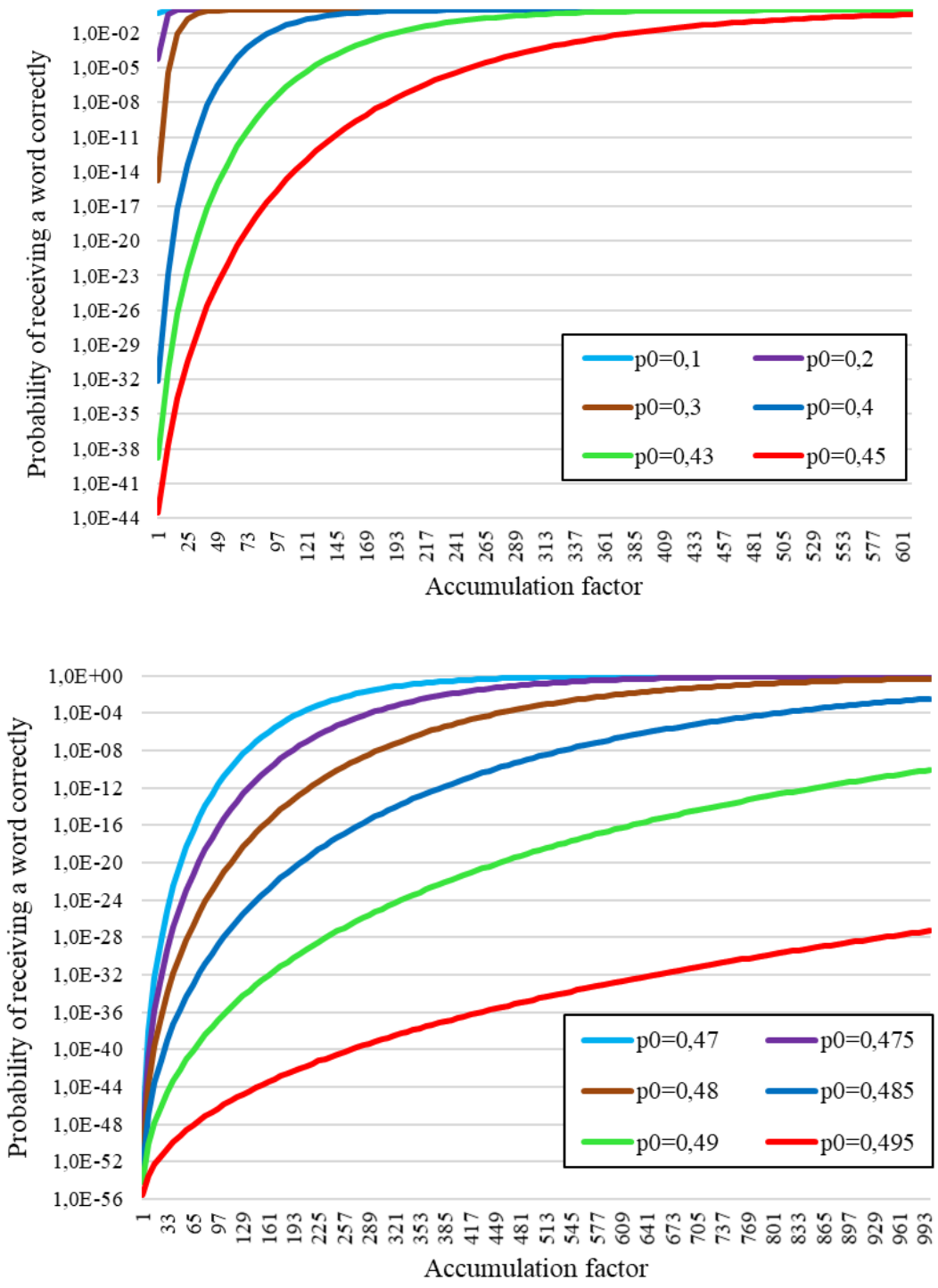


Рисунок 4.1 – Графіки залежностей імовірності правильного розпізнавання слова від коефіцієнта накопичення для різних імовірностей бітової помилки

Виходячи з характеру наведених на рисунку 4.1 залежностей, для досягнення заданої ймовірності правильної синхронізації коефіцієнт накопичення може змінюватись у широких межах. Так, для досягнення $P_{W\_true} \geq 0.999$ за $p_0 = 0.1$

необхідний коефіцієнт накопичення $l = 3$, за $p_0 = 0.3$ – $l = 21$, за $p_0 = 0.45$ – $l = 363$, а за $p_0 = 0.495$ – $l = 36413$. Тому пропонований у цьому розділі підхід до обробки даних передбачає поступове накопичення прийнятих з каналу зв'язку фрагментів довжини n, їх мажоритарну обробку з подальшою обробкою літер і слів. Максимальне значення коефіцієнта накопичення l визначається максимальним значенням ймовірності бітової помилки, на яку розрахована система передавання даних, і також заданою мінімальною ймовірністю безпомилкового прийому слова.

Під час створення системи передавання необхідно враховувати можливість появи невиявленої помилки у прийнятому слові. Визначимо цю можливість.

Теорема 4.1. *Імовірність помилкового розпізнавання літери*

$$P_{L\_false} = \sum_{j=1}^{N-1}\left(LW_j \cdot \sum_{v=d_j-d_{lim}}^{d_j} C_{d_j}^{v}\left(\sum_{w=0}^{v-d_j+d_{lim}} C_{n-d_j}^{w}\left(p_0^*\right)^{v+w}\left(1-p_0^*\right)^{n-(w+v)} \right)\right), \qquad (4.4)$$

де LW_j *– індикаторний коефіцієнт, що вказує на приналежність літери* L_j *підмножині літер, з якого формуються слова. Якщо* L_j *використовується для формування слова,* $LW_j = 1$*, в іншому випадку* $LW_j = 0$.

Доведення.

Для доведення цієї теореми скористаємося підходом, використаному під час доведення теореми 3.3.

Імовірність появи в уточненій послідовності R помилки, що призводить до хибного розпізнавання літери, дорівнює ймовірності виникнення в уточненій послідовності R будь-якого з векторів помилки, що переводить літеру в будь-яку іншу з решти, які використовуються для формування слова, $N-1$ літер з точністю до d_{lim} біт.

Нагадаємо, що d_j – відстань Хеммінга від літери L_0 до літери L_j (циклічного зсуву L_0 на j біт). Тоді помилка переводить літеру L_0 в її циклічний зсув L_j, якщо вона містить v помилок в d_j бітах, в яких ці послідовності відрізняються, причому

$d_j - d_{lim} \le \nu \le d_j$. Крім того, в інших $\left(n - d_j\right)$ бітах можлива поява ще w бітових помилок, причому $0 \le w \le \nu - \left(d_j - d_{lim}\right)$.

Імовірність зазначеної події:

$$P_{L\_false} = \sum_{j=1}^{N-1} \left(LW_j \cdot \sum_{\nu=d_j-d_{lim}}^{d_j} \left(\begin{array}{l} C_{d_j}^{\nu} \left(p_0^*\right)^{\nu} \left(1-p_0^*\right)^{d_j-\nu} \times \\ \times \sum\limits_{w=0}^{\nu-d_j+d_{lim}} C_{n-d_j}^{w} \left(p_0^*\right)^{w} \left(1-p_0^*\right)^{n-d_j-w} \end{array} \right) \right). \qquad (4.5)$$

Групуючи множники в (4.5), отримаємо формулу знаходження ймовірності хибного розпізнавання літери (4.4). *Теорему доведено.*

Зауваження 4.4. *Для перестановки* $\pi_i = \{000,001,111,011,010,101,100,110\}$ *з* $d_{lim} = 5$, *як показано в третьому розділі, значення* $d_{ij} = 12$ *зустрічаються в 19 випадках, значення* $d_{ij} = 14$ *зустрічаються у двох випадках і значення* $d_{ij} = 16$ *зустрічаються також у двох випадках. Тоді вираз (4.4) для зазначеної перестановки в разі використання для формування слова всіх її 24 циклічних зсувів набуде такого вигляду:*

$$\begin{aligned} P_{L\_false} = {} & 19\sum_{\nu=7}^{12} C_{12}^{\nu} \left(\sum_{w=0}^{\nu-7} C_{12}^{w} \left(p_0^*\right)^{\nu+w} \left(1-p_0^*\right)^{24-\nu-w} \right) + \\ & +2\sum_{\nu=9}^{14} C_{14}^{\nu} \left(\sum_{w=0}^{\nu-9} C_{10}^{w} \left(p_0^*\right)^{\nu+w} \left(1-p_0^*\right)^{24-\nu-w} \right) + 2\sum_{\nu=11}^{16} C_{16}^{\nu} \left(\sum_{w=0}^{\nu-11} C_{8}^{w} \left(p_0^*\right)^{\nu+w} \left(1-p_0^*\right)^{24-\nu-w} \right). \end{aligned} \qquad (4.6)$$

У разі використання для формування слова 23 циклічних зсувів перестановки $\pi_i = \{000,001,111,011,010,101,100,110\}$ імовірність хибного розпізнавання літери можна оцінити зверху таким виразом:

$$\begin{aligned} P_{L\_false} \le {} & 19\sum_{\nu=7}^{12} C_{12}^{\nu} \left(\sum_{w=0}^{\nu-7} C_{12}^{w} \left(p_0^*\right)^{\nu+w} \left(1-p_0^*\right)^{24-\nu-w} \right) + \\ & +2\sum_{\nu=9}^{14} C_{14}^{\nu} \left(\sum_{w=0}^{\nu-9} C_{10}^{w} \left(p_0^*\right)^{\nu+w} \left(1-p_0^*\right)^{24-\nu-w} \right) + \sum_{\nu=11}^{16} C_{16}^{\nu} \left(\sum_{w=0}^{\nu-11} C_{8}^{w} \left(p_0^*\right)^{\nu+w} \left(1-p_0^*\right)^{24-\nu-w} \right). \end{aligned} \qquad (4.7)$$

У разі використання для формування слова довільного числа N, $2 \le N \le n$, циклічних зсувів перестановки $\pi_i = \{000,001,111,011,010,101,100,110\}$ ймовірність хибного розпізнавання літери можна оцінити зверху таким виразом:

$$P_{L\_false} \le (N-1)\sum_{v=7}^{12} C_{12}^{v}\left(\sum_{w=0}^{v-7} C_{12}^{w}\left(p_0^*\right)^{v+w}\left(1-p_0^*\right)^{24-v-w}\right). \qquad (4.8)$$

Зауваження 4.5. *Під час виконання чисельних розрахунків для ймовірності хибного розпізнавання літери* P_{L_false} *у модельному прикладі цього розділу використано формулу (4.7).*

Теорема 4.2. *Імовірність хибного розпізнавання слова*

$$P_{W\_false} = \sum_{j=2}^{N} C_N^j \left(\frac{P_{L\_false}}{N-1}\right)^j \cdot !j \cdot P_{L\_true}^{N-j}, \qquad (4.9)$$

де $!j = j!\sum_{i=0}^{j}\frac{(-1)^k}{k!}$ *– субфакторіал числа* j.

Доведення.

Імовірність появи в слові помилки, що призводить до хибного його розпізнавання, дорівнює ймовірності перестановки двох або більше літер у слові. Решта літер при цьому розпізнаються без помилок. Тоді ймовірність хибного розпізнавання слова

$$P_{W\_false} = \sum_{j=2}^{N} C_N^j P_{der\_j} P_{L\_true}^{N-j}, \qquad (4.10)$$

де P_{der_j} – ймовірність появи помилки, що призводить до перестановки без нерухомих точок j літер у слові (ймовірність безладу з j літер).

Ця ймовірність визначається таким чином. Імовірність трансформації літери в певну іншу літеру підмножини літер слова дорівнює $\frac{P_{L\_false}}{N-1}$. Кількість безладів довжини дорівнює субфакторіалу $!j$. Тоді ймовірність безладу з j літер дорівнює $\left(\frac{P_{L\_false}}{N-1}\right)^j \cdot !j$. Підставляючи цей вираз у (4.10), отримаємо формулу знаходження ймовірності хибного розпізнавання слова (4.9). *Теорему доведено.*

Підставляючи для модельного прикладу оцінку зверху P_{L_false} з (4.7) в (4.9), отримаємо оцінку зверху для P_{W_false}. Графіки оцінок $P_{W\_false}(l)$ для розглянутого

модельного прикладу з $\pi_i = \{0,1,7,3,2,5,4,6\}$ і $N = 23$ за різних значень p_0 представлено на рисунку 4.2.

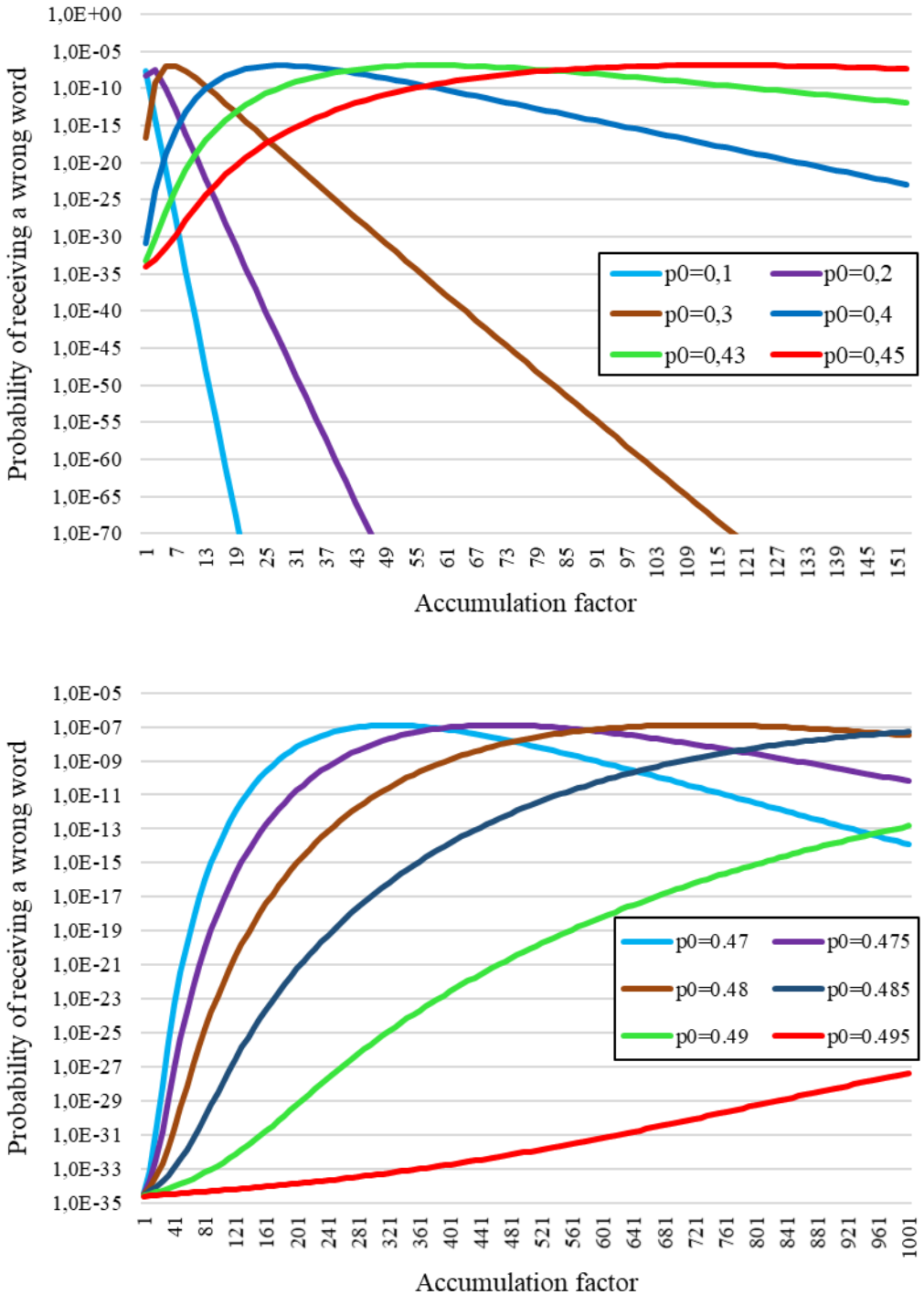


Рисунок 4.2 – Графіки залежностей імовірності хибного розпізнавання слова від коефіцієнта накопичення для різних імовірностей бітової помилки

Варто зазначити, що всі наведені на рисунку 4.2 графіки мають точки максимуму. Ці точки відповідають різним значенням коефіцієнта накопичення l. Максимальні значення $P_{W\_false}(l)$ для $p_0 \le 0.495$ наведено в таблиці 4.2.

Зауваження 4.6. *Для розглянутого модельного прикладу* $P_{W\_false}(l) \le 1.26580 \cdot 10^{-7}$ *для* $\forall l$ *при* $\forall p_0 \le 0.495$.

Таблиця 4.2 – **Максимальні значення** $P_{W\_false}(l)$

p_0	0.1	0.2	0.3	0.4	0.43
$\max\left(P_{W\_false}(l)\right)$	$2.53101 \cdot 10^{-8}$	$3.44773 \cdot 10^{-8}$	$1.03869 \cdot 10^{-7}$	$1.24529 \cdot 10^{-7}$	$1.26329 \cdot 10^{-7}$
p_0	0.45	0.47	0.475	0.48	0.495
$\max\left(P_{W\_false}(l)\right)$	$1.26568 \cdot 10^{-7}$	$1.26569 \cdot 10^{-7}$	$1.26577 \cdot 10^{-7}$	$1.26578 \cdot 10^{-7}$	$1.26580 \cdot 10^{-7}$

Зауваження 4.7. *Вирази (4.3) та (4.9) визначають імовірності правильного та хибного розпізнавання слова для окремого експерименту з фіксованим значенням коефіцієнта накопичення* l *та не враховують процедуру послідовного збільшення* l .

4.3.2. Оцінки сумарних інтервальних імовірностей правильного й хибного розпізнавання слова

Оцінимо сумарні інтервальні ймовірності правильного $\left(P_{W\_true\_final}(l)\right)$ та хибного $\left(P_{W\_false\_final}(l)\right)$ розпізнавання слів за фрагментами.

Нехай подія $A(i)$={приймач не розпізнав слово для всіх $j < i$ накопичених фрагментів}. Іншими словами, подія $A(i)$ – це накопичення приймачем i фрагментів. Позначимо також через $B(i)$ подію правильного розпізнавання слова за i фрагментами, а через $C(i)$ – подію хибного розпізнавання слова за i фрагментами. Крім того, для підходу з послідовним збільшенням коефіцієнта накопичення $D(l) = \bigcup_{i \le l} B(i)$ позначимо через $D(l)$ подію правильного

розпізнавання слова за $\forall i \le l$ фрагментами, а через $E(l)$ – подію хибного розпізнавання слова за $\forall i \le l$ фрагментами.

Подія $D(l)$ є об'єднанням усіх подій правильного розпізнавання слова за $i \le l$ фрагментами:

$$D(l) = \bigcup_{i \le l} B(i). \tag{4.11}$$

Імовірність події $D(l)$:

$$P(D(l)) = P\left(\bigcup_{i \le l} B(i)\right). \tag{4.12}$$

Зауважимо, що $P(D(l)) = P\left(\bigcup_{i \le l} B(i)\right) \ge P(B(i))$ для $\forall i \le l$. Оскільки згідно з рисунком 4.1 $P(B(i))$ монотонно зростає за i, для більш точної оцінки знизу $P(D(l))$ доцільно вибрати максимальне значення $P(B(i))$, $i \le l$. Цим значенням є $P(B(l))$. З урахуванням того, що $P(D(l)) = P_{W\_true\_final}(l)$ і $P(B(i)) = P_{W\_true}(i)$, має місце оцінка

$$P_{W\_true\_final}(l) \ge P_{W\_true}(l). \tag{4.13}$$

Подія $E(l)$ є диз'юнктивним об'єднанням усіх подій хибного розпізнавання слова за $i \le l$ фрагментам, помножених на події накопичення приймачем i фрагментів:

$$E(l) = \coprod_{i \le l} C(i) \cdot A(i). \tag{4.14}$$

Імовірність події $E(l)$:

$$P(E(l)) = P\left(\coprod_{i \le l} C(i) \cdot A(i)\right). \tag{4.15}$$

Оскільки події $C(i) \cdot A(i)$ у виразі (4.14) несумісні,

$$P(E(l)) = P\left(\coprod_{i \le l} C(i) \cdot A(i)\right) = \sum_{i \le l} P(C(i) \cdot A(i)). \tag{4.16}$$

Оскільки $P\left(C\left(i\right)\cdot A\left(i\right)\right)\le P\left(C\left(i\right)\right)$ для $\forall i\le l$, а $P\left(E\left(l\right)\right)=P_{W\_false\_final}\left(l\right)$ і $P\left(C\left(i\right)\right)=P_{W\_false}\left(i\right)$, справедлива оцінка

$$P_{W\_false\_final}\left(l\right)\le\sum_{i\le l}P_{W\_false}\left(i\right). \qquad (4.17)$$

Для розглянутого модельного прикладу наведемо на рисунку 4.3 графіки залежностей оцінок (4.13) і (4.17) імовірностей правильного та хибного розпізнавання слова від кількості накопичених фрагментів l.

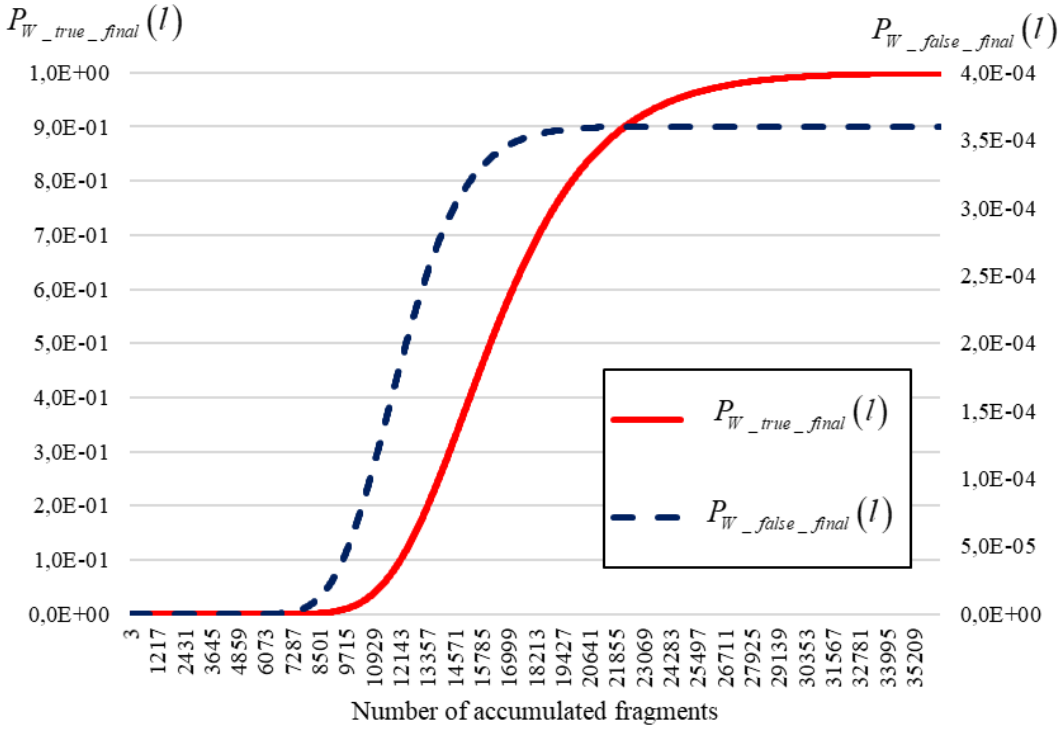


Рисунок 4.3 – Графіки залежностей оцінок імовірностей правильного і хибного розпізнавання слова від кількості накопичених фрагментів

На підставі описаного підходу до розпізнавання слів сформовано запропонований метод надійного передавання слів-перестановок.

4.4. Оцінка ефективності методу

Для перевірки ефективності роботи розробленого методу надійного передавання перестановок побудовано програмну модель. Алгоритм роботи приймача даних моделі наведено рисунку 4.4.

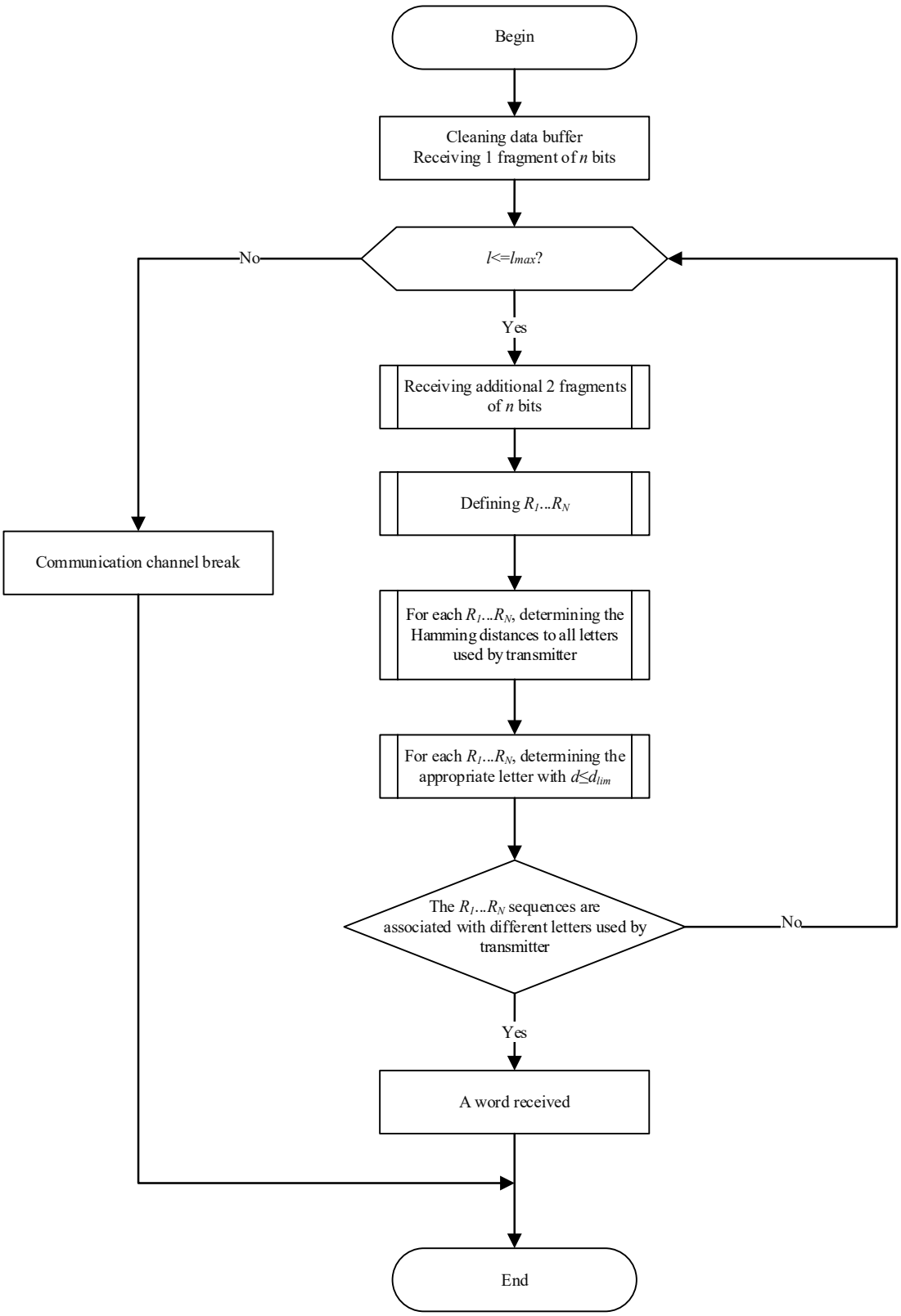


Рисунок 4.4. Алгоритм роботи приймача

Відповідно до зауваження 4.3, у модельному прикладі розділу слово W формуватимемо з $N=23$ літер – ненульових зсувів L_j, $1\le j\le 23$ перестановки довжиною $M=8$. Значення $L_0=(0,1,7,3,2,5,4,6)=(000,001,111,011,010,101,100,110)$.

Канал зв'язку у модельному прикладі – двійковий симетричний. Бітові помилки – незалежні.

4.4.1. Експериментальна перевірка одержаних результатів

Графіки експериментально визначених за 1000 випробуваннями залежностей відносних частот правильного прийому слова та літери фіксованого значення коефіцієнта накопичення l для ймовірності бітової помилки $p_0=0.4$ наведено на рисунку 4.5.

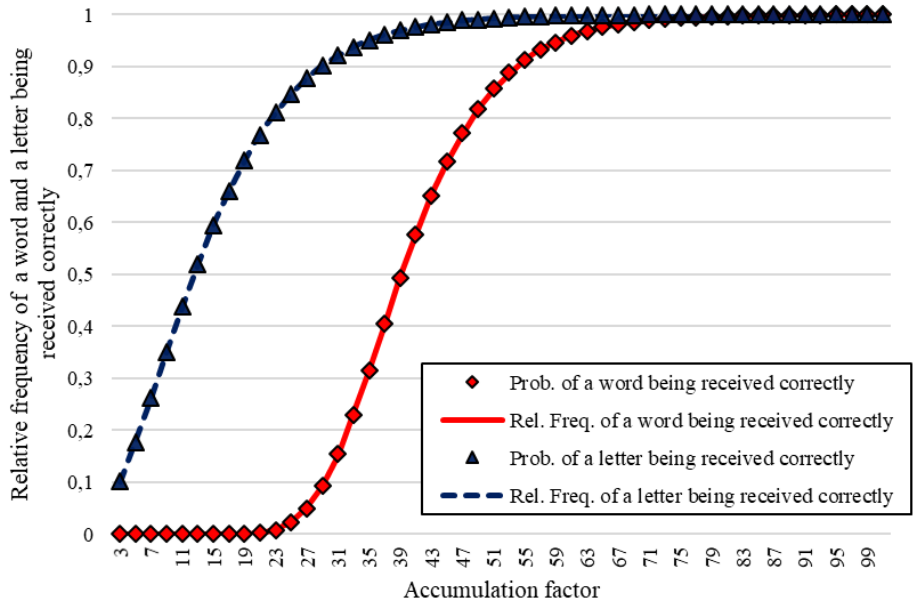


Рисунок 4.5 – Графіки залежностей відносних частот правильного прийому слова та літери від фіксованого значення коефіцієнта накопичення

Крім того, на рисунку 4.5 маркерами позначено відповідні графіки теоретичних залежностей P_{L_true} і P_{W_true} від l відповідно до виразів (4.2) і (4.3). Наведені на

рисунку 4.5 теоретичні та експериментальні залежності узгоджуються між собою за критерієм Пірсона з близькими до одиниці досягнутими рівнями значимості (*p*-value [118]). Подібна відповідність теоретичних та експериментальних залежностей спостерігається і для інших значень p_0. Усе це свідчить про коректність побудованої моделі передавання даних.

На рисунку 4.6 представлено графік експериментально визначеної відносної частоти правильного прийому слова залежно кількості накопичених фрагментів l для ймовірності бітової помилки $p_0 = 0.495$. Крім того, для $p_0 = 0.495$ представлено графік оцінки ймовірності правильного прийому слова $P_{W\_true\_final}(l)$, обчисленої за (4.13).

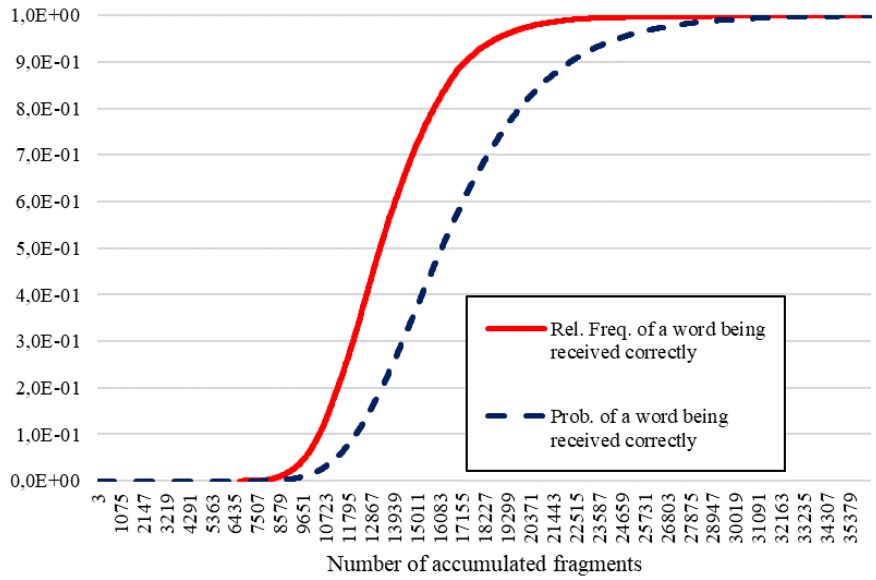


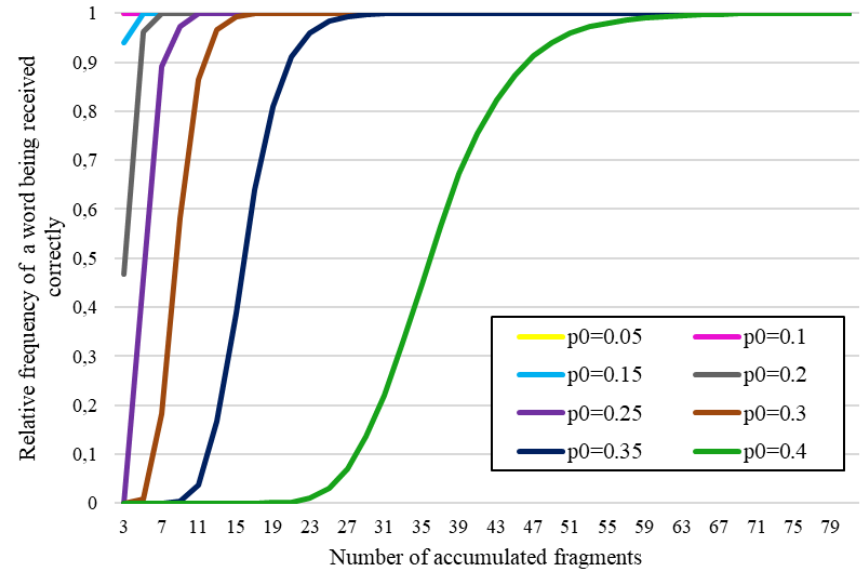
Рисунок 4.6 – Графіки залежностей відносної частоти правильного прийому слова від кількості накопичених фрагментів

Аналіз продемонстрованих на рисунку 4.6 результатів підтверджує справедливість оцінки (4.13). Разом з тим варто відзначити, що ця оцінка є досить грубою.

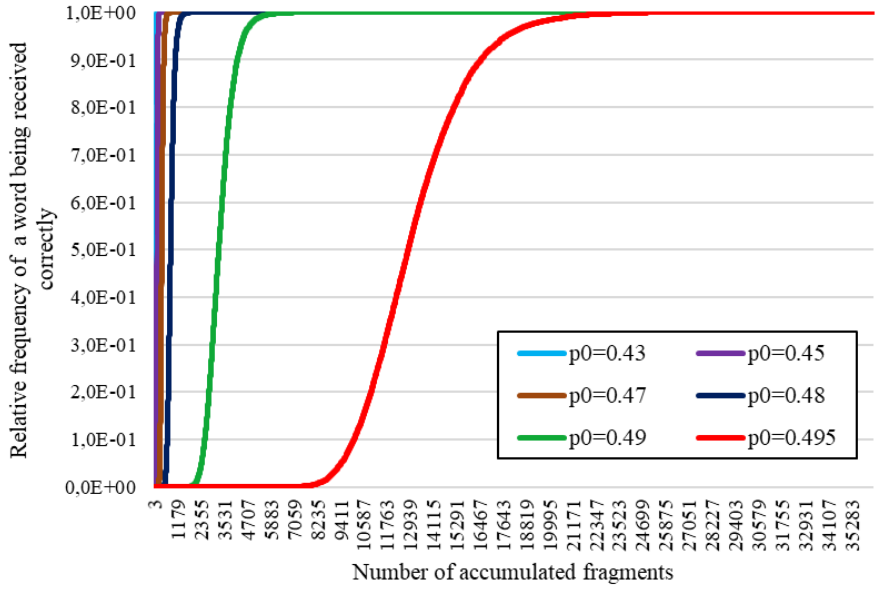
Зауваження 4.9. *Під час експериментального дослідження розробленого методу* $p_0 \le 0.495$ *випадків хибного розпізнавання слова для 10000 випробувань не*

сталося.

Наведені на рисунку 4.7 графіки визначають експериментально визначені ймовірнісні показники правильного прийому слова за ймовірності бітової помилки $p_0 \le 0.495$.



а)



б)

Рисунок 4.7 – Графіки залежностей відносної частоти правильного прийому слова від кількості накопичених фрагментів для $p_0 \le 0.495$

Розглянемо ситуацію, коли ймовірність бітової помилки перевищує $p_0 = 0.495$, а система передавання розрахована на $p_0 \le 0.495$ і видає сигнал відмови каналу зв'язку за неможливості розпізнати передане слово для максимального коефіцієнта накопичення $l_{\max}$. У модельному прикладі розділу $l_{\max} = 36413$ обчислене для забезпечення $P_{W\_true} \ge 0.999$ за $p_0 = 0.495$.

Подані на рисунку 4.8 залежності визначають експериментально визначені ймовірнісні показники правильного прийому слова за ймовірності бітової помилки $p_0 \ge 0.495$.

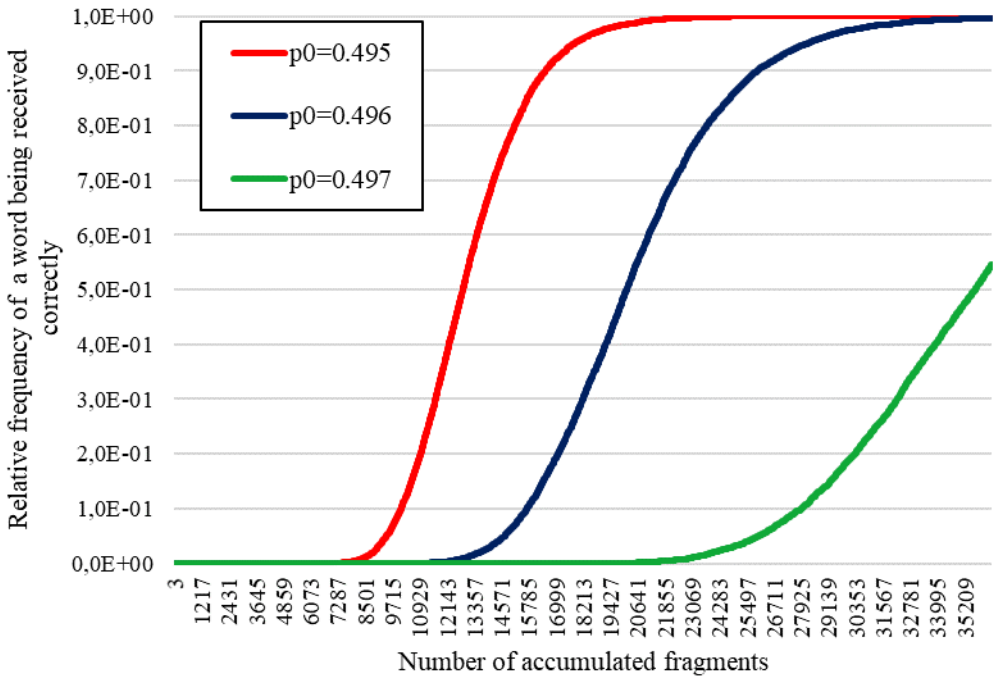


Рисунок 4.8 – Графіки залежностей відносної частоти правильного прийому слова від кількості накопичених фрагментів для $p_0 \ge 0.495$

Рисунок 4.8 свідчить, що з перевищенням імовірності бітової помилки граничного значення (для розглянутого модельного прикладу $p_0 = 0.495$) вимоги щодо ймовірності правильного прийому слова порушуються. Наприклад, відносна частота правильного прийому слова для коефіцієнта накопичення $l_{\max} = 36413$ дорівнює 0.9966 для $p_0 = 0.496$ і 0.5454 для $p_0 = 0.497$. Для $p_0 = 0.498$ та 10000

експериментів випадків правильного прийому переданого слова не спостерігалося.

Зазначимо також, що для $p_0 = 0.496$, $p_0 = 0.497$ і $p_0 = 0.498$ випадків хибного розпізнавання слова для 10000 випробувань не відбулося.

Визначимо і наведемо на рисунку 4.9 середню кількість накопичених фрагментів, необхідної для правильного прийому слова, залежно від імовірності бітової помилки p_0.

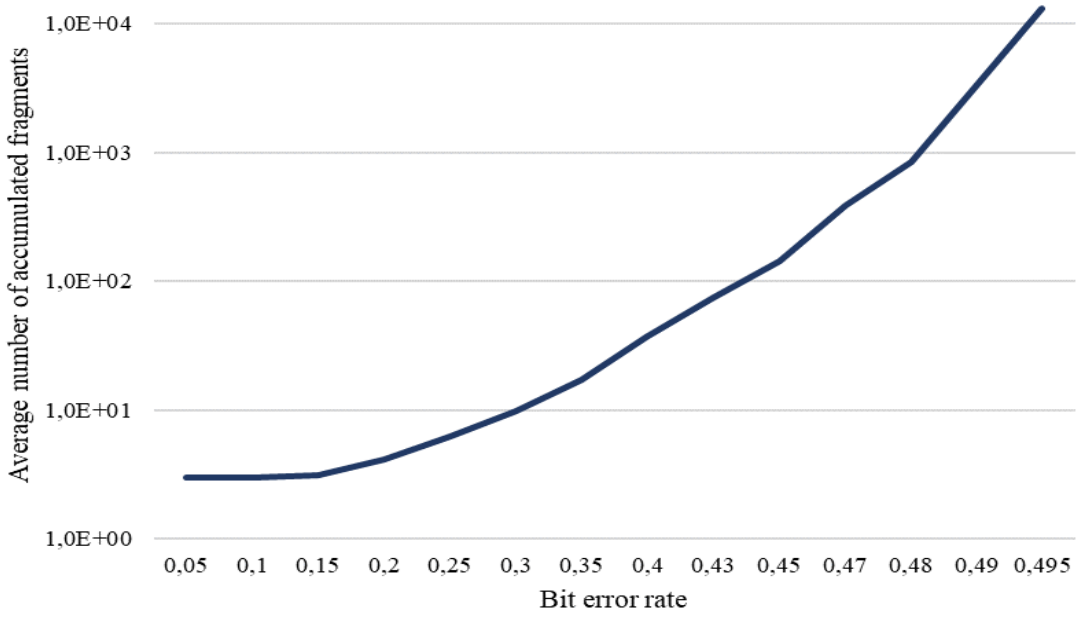


Рисунок 4.9 – Графік середнього значення кількості накопичених фрагментів, необхідної для правильного прийому слова, від імовірності бітової помилки

Графік рисунку 4.9 свідчить про експоненціальний характер збільшення середнього значення кількості накопичених фрагментів.

4.4.2. Обговорення отриманих результатів

Ефективність методу надійного передавання перестановок підтверджено розглянутим модельним прикладом реалізації методу для $M = 8$, $n = 24$, $N = 23$, $P_{W\_true\_final} \geq 0.999$ за максимально допустимої ймовірності бітової помилки $p_0 = 0.495$, $l_{\max} = 36413$.

Разом з тим, під час проектування системи можуть бути задані інші значення довжини N перестановки, що передається, а також мінімальної ймовірності її правильного прийому P_{W_true}.

У цьому випадку як базова перестановка для формування N літер може бути обрана перестановка π довжиною M, де $\lceil \log_2 M \rceil \cdot M \geq N$. При цьому перестановка π повинна відповідати вимогам визначення 4.1.

Імовірність P_{W_true} за допомогою формули (4.3) визначить граничне значення коефіцієнта накопичення $l_{\max}$.

Оскільки $P_{L\_true} < 1$, з формули (4.3) випливає, що P_{W_true} зростає зі зменшенням N. Тому вимоги щодо забезпечення заданої ймовірності правильного прийому перестановки P_{W_true} можуть бути досягнуті для будь-якого $M : \lceil \log_2 M \rceil \cdot M \geq N$.

Вибір оптимальних значень $\{N, M\}$ для досягнення необхідних імовірнісних показників системи передавання даних виходить за рамки розгляду цього дослідження.

Крім того, метод надійного передавання перестановок у частині представлення кожної літери слова деякою бітовою послідовністю схожий на принцип побудови шумоподібного сигналу під час використання методу прямої послідовності для розширення спектру (DSSS).

Порівняємо ймовірнісні характеристики цих двох методів.

Для розглянутого модельного прикладу кожне слово W є перестановкою довжини $N = 23$ і кодується послідовністю з $23 \cdot 24 = 552$ біт. При цьому для забезпечення $P_{W\_true\_final} \geq 0.999$ $\left(P_{W\_false\_final} \leq 3.6 \cdot 10^{-4}\right)$ за $p_0 = 0.495$ необхідний коефіцієнт накопичення $l_{\max} = 36413$. Такий коефіцієнт накопичення потребує передавання $552 \cdot 36413 = 20099976$ біт.

У випадку використання DSSS кожен біт перестановки передається у каналі як послідовності з B чіпів. Нехай символи перестановки W кодуються рівномірним

кодом. Для $N = 23$ довжина кодової комбінації кожного символу перестановки складе $\lceil \log_2 23 \rceil = 5$ біт. Тоді довжина перестановки становитиме 115 біт. Для забезпечення однакової канальної швидкості передачі кожен біт перестановки W для методу DSSS представляється $B = 20099976/115 = 174782$ чіпами.

Визначимо завадостійкість методу DSSS із заданими параметрами.

Нехай під час передавання використовується двопозиційна фазова маніпуляція. Імовірність бітової помилки в каналі з адитивним білим гаусівським шумом обчислюється за формулою:

$$p_0 = Q\left(\sqrt{2E_b / N_0}\right), \tag{4.18}$$

де $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-\frac{1}{2}t^2} dt$ – комплементарна гаусова функція похибки; E_b – енергія на біт; $\frac{1}{2} N_0$ – спектральна щільність потужності шуму. Імовірність бітової помилки в слові W дорівнює $p_0(W) = Q\left(\sqrt{2BE_b / N_0}\right)$.

За ймовірності бітової помилки $p_0 = 0.495$ відповідно до (4.18) значення $\sqrt{2E_b / N_0} = 12.53 \cdot 10^{-3}$. Тоді $p_0(W) = Q\left(\sqrt{2BE_b / N_0}\right) = 80.34 \cdot 10^{-9}$.

Слово W буде прийнято вірно тоді і тільки тоді, коли всі його біти будуть вірно прийняті. Імовірність цієї події $P_{W\_true\_DSSS} = \left(1 - p_0(W)\right)^{115} = 1 - 9 \cdot 10^{-6}$.

Зауважимо, що ця ймовірність перевищує вимогу $P_{W\_true\_final} \geq 0.999$, якій визначено значення $l_{\max} = 36413$.

Водночас оцінка $P_{W\_true\_final}$ є досить грубою.

Визначивши за експериментально побудованою кривою відносної частоти правильного розпізнавання слова рисунка 4.6 мінімальне значення l, за якого ця відносна частота досягає значення $1 - 9 \cdot 10^{-6}$, отримаємо $l = 29123$. За такого коефіцієнта накопичення відносна частота правильного розпізнавання слова досягає значення $1 - 4.6 \cdot 10^{-14}$.

Коефіцієнт накопичення $l_{\max} = 29123$ потребує передавання $552 \cdot 29123 = 16075896$ біт. Тоді для забезпечення однакової канальної швидкості передавання, кількість чіпів на кожний біт перестановки W для методу DSSS має дорівнювати $B = 16075896/115 = 139791$. В цьому випадку $p_0(W) = Q\left(\sqrt{2BE_b/N_0}\right) = 1.39 \cdot 10^{-6}$, а $P_{W\_true\_DSSS} = \left(1 - p_0(W)\right)^{115} = 0.9998$.

Отримані результати свідчать про вищу завадостійкість запропонованого методу порівняно з методом DSSS для розглянутого модельного прикладу. Разом з тим, для остаточних висновків порівняння завадостійкості цих методів потрібні додаткові дослідження.

4.5. Висновки

У четвертому розділі розроблено метод достовірного передавання перестановок, який за рахунок подання кожного елементу (літери) переданої перестановки (слова) у вигляді циклічного двійкового зсуву перестановки-носія, що має максимальне значення мінімальної відстані Хеммінга від її двійкового подання до всіх її циклічних зсувів, а також за рахунок використання мажоритарної та кореляційної обробки прийнятих з каналу зв'язку фрагментів, довжина яких дорівнює довжині літери, дозволяє реалізувати надійне передавання слова в умовах впливу в каналі зв'язку завад високої інтенсивності.

Розроблено математичну модель процесу передавання перестановки-носія. Наведено аналітичні вирази для обчислення ймовірності правильного та хибного прийому перестановки. Розроблено та реалізовано алгоритм, який застосовує запропонований метод.

Розроблений метод може бути використаний для підвищення стійкості комунікаційної системи до впливу завад.

Виконано побудову імітаційної програмної моделі системи передавання даних. Аналіз отриманих результатів її роботи свідчать про коректність наведених теоретичних оцінок. Крім того, порівняння експериментально отриманої ймовірності приймання перестановки без помилок для розробленого методу з

відповідною ймовірністю для методу DSSS за однакової швидкості передавання та ймовірності бітової помилки в каналі зв'язку $p_0 = 0.495$ підтверджує ефективність розробленого методу.

Основні результати дослідження та розробки методу представлено в [119], [120].

5. ЕКСПЕРИМЕНТАЛЬНА ПОРІВНЯЛЬНА ОЦІНКА ЕФЕКТИВНОСТІ РОЗРОБЛЕНИХ МЕТОДІВ ЦИКЛОВОЇ СИНХРОНІЗАЦІЇ В ЗАЛЕЖНОСТІ ВІД ЯКОСТІ КАНАЛУ ЗВ'ЯЗКУ ТА ПАРАМЕТРІВ СИНХРОНІЗАЦІЇ

5.1. Вступ

У другому та третьому розділах розроблено, описано та досліджено методи циклової синхронізації систем передавання інформації з нероздільним факторіальним кодуванням: методу на основі використання як синхрокомбінації перестановки чисел з її поділом на префіксну та суфіксну частини, та методу, що використовує як синхрокомбінацію перестановку, що має максимальне значення мінімальної відстані Хеммінга від її двійкового представлення до всіх її циклічних зсувів. У кожного з запропонованих методів є свої переваги та недоліки. Тому в першій частині цього розділу виконано експериментальне порівняння ефективності цих методів в залежності від якості каналу зв'язку та надано рекомендації щодо застосування для кожного з них.

Для цього розроблено та описано структурні схеми імітаційних моделей системи передавання даних для кожного з запропонованих методів циклової синхронізації. Описано середовище розробки та параметри апаратної частини, на якій виконувалося моделювання. Побудовано програмні імітаційні моделі систем передавання даних, у яких реалізовано алгоритми встановлення циклового синхронізму для кожного з наведених методів. Обґрунтовано основні модулі, які використовувалися як для реалізації моделей, так і для інтерпретації результатів.

У другій частині цього розділу отримано оцінку ефективності циклової синхронізації нероздільних факторіальних кодів у залежності від параметрів синхронізації. Для цього виконано реалізацію принципів встановлення циклової синхронізації нероздільних факторіальних кодів, а також застосовано операцію перемішування отриманих з каналу зв'язку фрагментів для підвищення ефективності знаходження меж перестановок. Застосовано алгоритм встановлення циклового синхронізму з параметрами, визначеними за верхньої межі ймовірності

бітової помилки в каналі зв'язку $p_{0\_\max} = 0,495$, для середовищ з імовірністю бітової помилки $p_0 \le 0.495$. Визначено параметри алгоритму встановлення циклового синхронізму за верхньої межі ймовірності бітової помилки в каналі зв'язку $p_{0\_\max} = 0,4$. Оцінено ефективність використання операції перемішування отриманих з каналу зв'язку фрагментів. Порівняно ефективність реалізації алгоритмів встановлення циклового синхронізму з параметрами, визначеними за верхньої межі ймовірності бітової помилки в каналі зв'язку $p_{0\_\max} = 0,495$ та $p_{0\_\max} = 0,4$, для середовищ з імовірністю бітової помилки $p_0 \le 0,4$. Надано рекомендації до вибору параметрів алгоритму синхронізації, які можуть бути використані для підвищення ефективності алгоритмів встановлення циклового синхронізму під час проектування комунікаційних систем з нероздільним факторіальним кодуванням даних в умовах дії в каналі зв'язку природних або штучно створених інтенсивних завад.

5.2. Експериментальне порівняння ефективності розроблених методів циклової синхронізації в залежності від якості каналу зв'язку

Виконаємо оцінювання показників ефективності методів циклової синхронізації, описаних у розділах 2 і 3, для комунікаційних систем передавання інформації з нероздільним факторіальним кодуванням на основі імітаційного моделювання та дослідження області застосування для кожного з зазначених методів.

Для цього необхідно виконати наступні задачі:

– побудувати імітаційну модель системи передавання інформації з нероздільним факторіальним кодуванням;

– у блоках циклової синхронізації приймачів реалізувати алгоритми, побудовані за описаними методами;

– виконати дослідження часу входження в цикловий синхронізм для різних параметрів каналу зв'язку (значень бітової помилки в каналі зв'язку);

– виконати порівняльний аналіз отриманих показників часу входження в цикловий синхронізм;

– сформулювати рекомендації щодо використання методів циклової синхронізації для комунікаційних систем передавання інформації з нероздільним факторіальним кодуванням.

5.2.1. Опис основних відмінностей методів

У розділах 2 та 3 детально досліджено два методи входження в цикловий синхронізм для комунікаційних систем передавання інформації з нероздільним факторіальним кодуванням в умовах впливу в каналі зв'язку завад високої інтенсивності.

Для більш лаконічного викладення матеріалу в подальшому будемо називати першим методом метод на основі використання як синхрокомбінації перестановки чисел з її поділом на префіксну та суфіксну частини, описаний в другому розділі, а другим методом – метод, який використовує як синхрокомбінацію перестановку, що має максимальне значення мінімальної відстані Хеммінга від її двійкового представлення до всіх її циклічних зсувів, описаний у третьому розділі.

Коротко нагадаємо основні особливості реалізації для кожного з методів.

Перший метод передбачає виконання таких кроків:

1) формування перестановки π довжини M, відповідно до правил та вимог, які описані в другому розділі;

2) накопичення фрагментів отриманої з каналу бітової послідовності, починаючи з трьох;

3) формування уточненої послідовність R, за мажоритарним принципом;

4) перевірка наявності шаблонної комбінації в уточненій послідовності R;

5) порівняння уточненої послідовності R з синхрокомбінацією π та завершення процедури циклового фазування (формування сигналу «Пошук синхронізму завершений» (ПЗ)) у випадку їх збігів з точністю до одного з символів;

6) перехід до п. 3 за невиконання умов п. 5 та додаткове накопичення ще двох фрагментів.

Після досягнення деякого заданого порогу кількість накопичених фрагментів не змінюється. Процес пошуку синхронізму триває, поки або не буде знайдений синхронізм, або не закінчиться ліміт часу на виконання пошуку синхронізму.

Другий метод передбачає виконання таких кроків:

7) формування перестановки π довжини M, відповідно до правил та вимог, які описані в третьому розділі;

8) накопичення прийнятих з каналу зв'язку K блоків, що складаються з l фрагментів по n біт. Зміна значень K і l відповідно до описаної в третьому розділі методики;

9) обчислення для кожного блоку незалежно уточненої послідовності R_k, $k \in [1, K]$, за мажоритарним принципом;

10) обчислення для кожної уточненої послідовності R_k відстані Хеммінга до всіх її циклічних зсувів синхрокомбінації π та пошук їх відповідності відповідним циклічним зсувам за принципом кореляційної обробки;

11) прийняття рішення про встановлення синхронізму за відповідності одному і тому ж зсуву синхрокомбінації всіх послідовностей R_k, $k \in [1, K]$;

12) повторення всіх операцій виявлення синхрокомбінації, починаючи з п. 2 за невиконання умов попереднього пункту.

Число накопичених фрагментів може послідовно збільшуватися до деякого, заздалегідь заданого порогу.

Як зазначено в третьому розділі, перший метод має наступні недоліки:

1) інтегральна функція часу входження в синхронізм не є оптимальною;

2) зі збільшенням імовірності бітової помилки та за невеликих значень коефіцієнта накопичення імовірність встановлення хибного синхронізму відносно велика.

За представленими в третьому розділі теоретичними оцінками другий метод має кращі кількісні показники процедури встановлення циклового синхронізму

комунікаційних систем передавання інформації з ФКВД в умовах впливу в каналі зв'язку завад високої інтенсивності. Зокрема, підвищується ймовірність встановлення циклового синхронізму для заданого коефіцієнта накопичення, а також зменшується ймовірність встановлення хибного синхронізму.

Наведений на рисунку 3.13 графік свідчить про більш швидке досягнення відносною частотою встановлення правильного синхронізму заданого порогового значення для другого методу порівняно з першим методом. Разом з тим, існують діапазони значень кількості накопичених фрагментів, де відносна частота встановлення правильного синхронізму для першого методу вища.

5.2.2. Побудова та опис імітаційних моделей

Виконаємо побудову імітаційних моделей систем передавання даних (СПД) з ФКВД.

Розглянемо структурну схему імітаційної моделі СПД із застосуванням першого методу циклової синхронізації (рисунок 5.1).

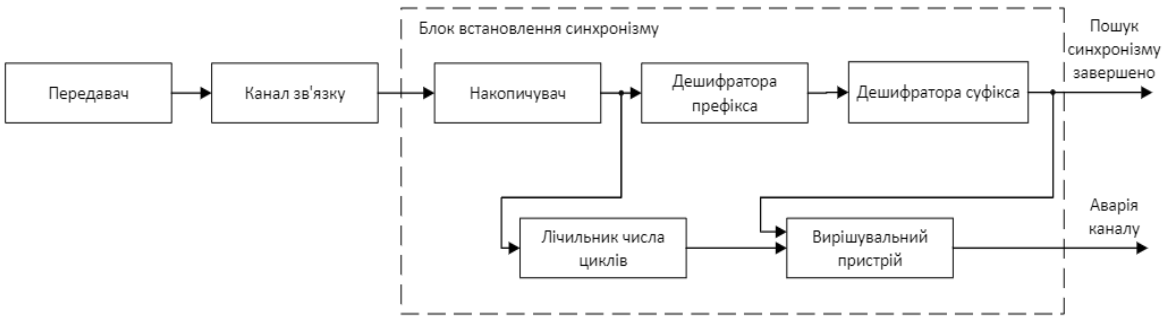


Рисунок 5.1 – Структурна схема імітаційної моделі СПД із застосуванням першого методу циклової синхронізації

Для цього методу синхрокомбінація містить префікс, що виявляє межі символів. Прикладом такої синхрокомбінації для $M = 8$ може бути перестановка

$$\pi = (000,111,001,010,100,110,101,011).$$

Обрана послідовність записується в постійний запам’ятовувальний пристрій (ПЗП) передавача, який є джерелом синхрокомбінації передавальної станції.

У приймач надходить синхрокомбінація, уражена дією завад у каналі зв'язку і зсунута за фазою щодо положення ковзного вікна (виконаного у вигляді регістра зсуву). При цьому ні інтенсивність завади, ні зсув ковзного вікна (величина фазової неузгодженості циклів передавача та приймача) апріорно невідомі.

Під час старту процедури пошуку синхронізму встановлюється мінімальне значення коефіцієнта накопичення $l = 3$. У буферний накопичувач записуються три фрагменти прийнятої з каналу зв'язку послідовності. Довжина кожного фрагмента дорівнює довжині синхрокомбінації. За цими трьома фрагментами мажоритарно обчислюється уточнена послідовність R.

У накопичувачі виконується побітовий циклічний зсув уточненої послідовності R. Дешифратор префікса після виконання кожного з зсувів перевіряє, чи виявлено префікс синхрокомбінації. Якщо префікс виявлено, то дешифратором суфікса перевіряється суфікс. Якщо обидві перевірки завершено позитивно, синхронізм встановлено. У цьому випадку, дешифратор суфікса формує сигнал «Пошук синхронізму завершений». Він надсилається зворотним каналом на передавальну станцію.

Якщо ж дешифратором префікса не виявлено префікса синхрокомбінації або дешифратором суфікса не підтверджено цілісність синхрокомбінації, то в накопичувач додатково записуються ще два фрагменти отриманої з каналу послідовності біт. Після цього знову повторюються описані вище дії: обчислюється уточнена послідовність, за якою дешифратор префікса і дешифратор суфікса виконують виявлення синхрокомбінації. Такий пошук синхронного стану триває або до успішного завершення процедури пошуку, або до вичерпання ліміту часу на пошук синхронізму. Для цього система циклової синхронізації містить лічильник числа циклів. Якщо після закінчення ліміту часу, відведеного на пошук синхронізму, він не знайдений, то вирішувальний пристрій формує сигнал «Аварія каналу». Процедура циклової синхронізації припиняється.

Розглянемо структурну схему імітаційної моделі СПД із застосуванням другого методу циклової синхронізації (рисунок 5.2).

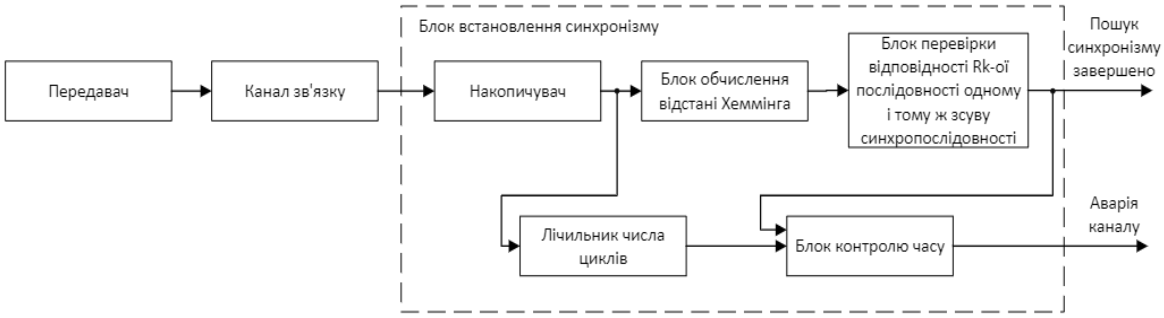


Рисунок 5.2 – Структурна схема імітаційної моделі СПД із застосуванням другого методу циклової синхронізації

Синхрокомбінацією є перестановка π довжини M, що має максимальне значення мінімальної відстані Хеммінга від її двійкового представлення до всіх її циклічних зсувів. Для $M=8$ такою синхрокомбінації може бути перестановка $\pi=(000,001,111,011,010,101,100,110)$. Ця синхрокомбінація записується в ПЗП передавача, який стане джерелом синхрокомбінації передавальної станції.

У накопичувачі приймача накопичуються прийняті з каналу зв'язку K блоків, що містять l фрагментів по n біт. Значення K і l змінюються відповідно до методики, яка забезпечує дотримання заданої ймовірності хибного фазування. У накопичувачі також для кожного блоку незалежно обчислюється уточнена послідовностей R_k, $k\in[1,K]$.

У блоці обчислення відстані Хеммінга для кожної уточненої послідовності R_k обчислюються відстані Хеммінга до всіх циклічних зсувів синхрокомбінації. Якщо для якогось із зсувів ця відстань не перевищує значення $d_{lim}=\lfloor(d-1)/2\rfloor$, приймається рішення про відповідність прийнятої послідовності цьому зсуву.

У блоці перевірки відповідності R_k-ої послідовності одному і тому ж зсуву синхропослідовності виконується перевірка, чи всі послідовності R_k, $k\in[1,K]$, відповідають одному і тому ж зсуву синхропослідовності. Якщо це так, то системою циклової синхронізації приймається рішення про встановлення синхронізму. У цьому випадку, блок перевірки на відповідність одному і тому ж зсуву синхрокомбінації для всіх R_k формує сигнал «Пошук синхронізму завершений».

Він надсилається зворотним каналом на передавальну станцію. У іншому випадку повторюються всі операції виявлення синхрокомбінації.

Як і у випадку з першим методом, пошук синхронного стану триває або до успішного завершення процедури пошуку, або до вичерпання ліміту часу на пошук синхронізму.

5.2.3. Опис використаного програмного та апаратного забезпечення

Імітаційні моделі розроблено мовою програмування Python [121] з використанням інтегрованого середовища розробки PyCharm community edition 2020.3 [122].

PyCharm [122] – це інтегроване середовище розробки мови програмування Python [121]. Забезпечує інструменти аналізу коду, графічний налагоджувач, запуск модульного тестування та підтримку веб-дозволів для Django. PyCharm [122] створюється JetBrains на основі IntelliJ IDEA. PyCharm [122] є кросплатформеним середовищем розробки, сумісним з Windows, macOS, Linux. PyCharm Community Edition (безкоштовна версія) ліцензується за ліцензією Apache, а PyCharm Professional Edition (платна версія) є власним програмним забезпеченням.

Під час розробки моделі використано наступні модулі Python:

- collections [123], [124], [125], [126] – надає спеціалізовані типи даних на основі словників, кортежів, наборів, списків. З цього модуля в роботі використано collections.Counter – своєрідний словник, який дозволяє підрахувати кількість незмінних об'єктів (в більшості випадків рядків). Модуль застосовано для полегшення роботи з результатами експериментів (процес, вихід);
- multiprocessing [124], [127], [128] – дозволяє створювати процеси так само, як і під час створення потоків за допомогою модуля потокової обробки. Цей модуль дозволяє обійти GIL (Global Interpreter Lock), яке дозволяє лише одному потоку керувати інтерпретатором Python, і скористатися можливістю використання декількох (всіх) процесорів на комп’ютері. Модуль використано для пришвидшення

виконання експериментів за рахунок розпаралелювання виконання експериментів на всіх можливих потоках використовуваного центрального процесора;

- random [121], [123] – для створення блоку даних із заданою ймовірністю бітової помилки p_0;
- os [121], [123] – для отримання від системи кількості потоків використовуваного центрального процесора. Це значення використовується для модульної багатопроцесорної обробки;

Розробка програмної моделі відбувалася на персональному комп'ютері форм-фактора ноутбук з наступними параметрами:

- ОС- Windows 10;
- ЦП - Intel Core i5-8250U;
- ОЗУ 8gb (2x4gb dual channel 2133Mhz);
- Відеокарта - nVIDIA GeForce MX150 2gb;
- Сховище - INTEL SSDSC2KW256G8L (256 Гб, SATA-III).

5.2.4. Отримані результати

Імітаційне моделювання мало на меті визначення наступних параметрів:

– швидкість входження в синхронізм в залежності від імовірності бітової помилки p_0. Цей показник отримано шляхом визначення середнього часу встановлення синхронізму для різних значень p_0;

– відносну частоту хибного фазування в залежності від імовірності бітової помилки p_0. Цей показник отримано шляхом визначення кількості хибних фазувань, поділеної на загальну кількість виконаних експериментів для конкретного значення p_0.

Через значне збільшення часу виконання експерименту для великих значень імовірності бітової помилки $\left(p_0 \geq 0.48\right)$ кількість експериментів не була однаковою для всіх значень p_0:

- для кожного значення $p_0 < 0.48$ проводилося 10000 експериментів;

- для кожного значення $p_0 \geq 0.48$ проводилося 1000 експериментів.

Усі експерименти проводилися в програмному середовищі PyCharm Community Edition [122] на персональному комп'ютері форм-фактора десктоп з наступними параметрами:

- ОС- Windows 10;
- ЦП - Intel Core i5-10400F;
- ОЗУ 32Gb (2x16Gb dual channel 3200Mhz);
- Відеокарта - GeForce GTX 1650 4Gb;
- Сховище - SSD M.2 2280 1TB Samsung.

Результати виконання експериментів та обробки їх результатів представлено в графічному вигляді.

Так, на рисунку 5.3 зображено графіки залежності середнього часу встановлення синхронізму від імовірності бітової помилки для кожного з розглянутих методів.

Рисунок 5.3 – Графіки залежностей середнього часу встановлення синхронізму від імовірності бітової помилки

На рисунку 5.4 зображено графіки залежності відносної частоти хибного фазування від імовірності бітової помилки. Такі залежності дозволяють виконати оцінку надійності (достовірності) встановлення синхронізму для кожного з методів.

Аналіз результатів експериментів свідчить про те, що:

– у каналах з рівнем бітової помилки $p_0 \le 0.495$ швидкість встановлення циклового синхронізму є вищою для реалізації першого методу, для каналів з $p_0 > 0.495$ – для реалізації другого методу;

– реалізація першого методу має вищі значення ймовірності хибного фазування.

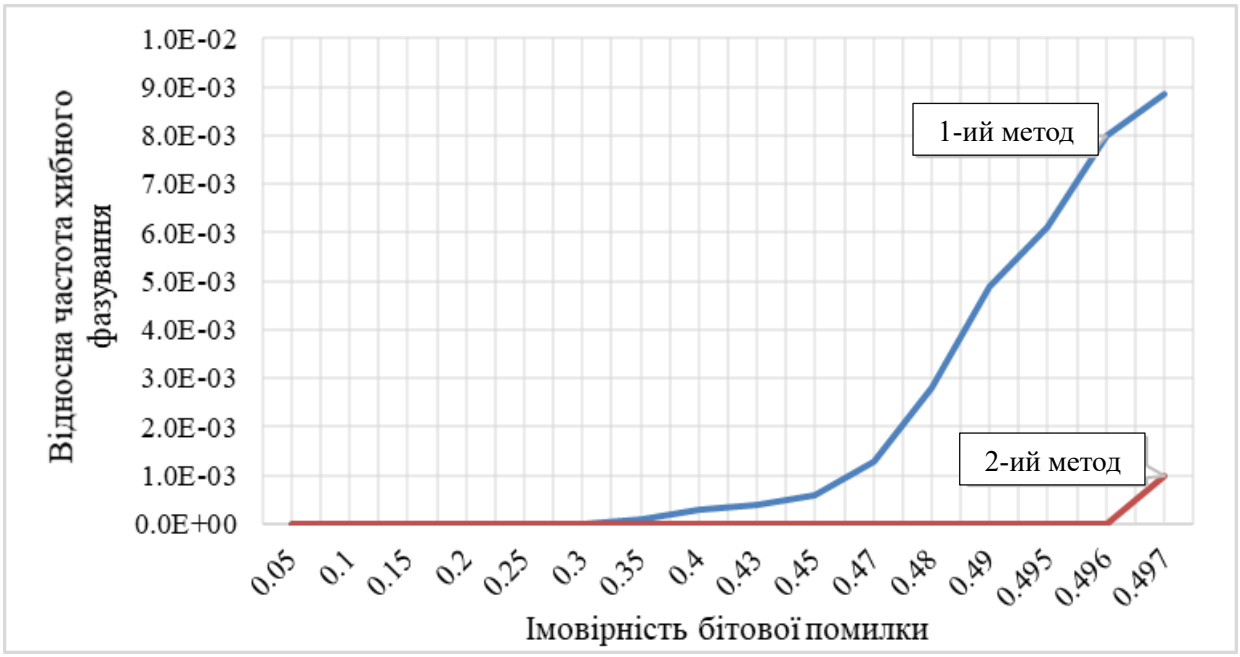


Рисунок 5.4 - Графіки залежностей відносної частоти хибного фазування від імовірності бітової помилки

Зазначені результати пояснюються наступним:

– параметри в реалізації другого методу циклової синхронізації не є оптимальним для всіх значень бітової помилки p_0, а визначені для $p_0 = 0.495$. Для максимальної ефективності другого методу ймовірність бітової помилки в каналі має бути спрогнозована максимально точно. З іншого боку, процедура вибору

моментів зміни значень параметрів K і l може бути адаптивною та залежати від імовірності виникнення бітової помилки в каналі зв'язку;

– параметри другого методу циклової синхронізації формуються виходячи з вимог до забезпечення дотримання заданої ймовірності хибного фазування, дозволяючи визначити її на етапі проектування системи циклової синхронізації.

5.3. Оцінка ефективності циклової синхронізації нероздільних факторіальних кодів у залежності від параметрів синхронізації

Описаний у третьому розділі другий метод дозволяє реалізувати циклову синхронізацію для систем передавання даних з короткими пакетами [129], [130], [131], [132], [133], зокрема, побудованих на основі нероздільного факторіального кодування даних. Проведене дослідження підтвердило ефективність застосування розробленого методу для каналів з високою інтенсивністю завад.

Разом з тим, параметри алгоритму циклової синхронізації обмежено одним випадком для $p_0 = 0,495$.

Метою другої частини цього розділу є дослідження ефективності другого методу циклової синхронізації нероздільного факторіального коду шляхом визначення та аналізу показників синхронізації з параметрами, обчисленими для різних граничних імовірностей бітової помилки p_0.

Ефективність методу циклової синхронізації будемо оцінювати на двох модельних системах передавання даних із наступними обмеженнями:

– довжина синхрокомбінації-перестановки $M = 8$, довжина синхрокомбінації в двійковому вигляді за рівномірного кодування символів перестановки – $n = 24$;

– комунікаційний канал є двійковим симетричним [134] з незалежними бітовими помилками;

– імовірність бітової помилки в каналі зв'язку p_0 обмежено значеннями 0,495 і 0,4;

– імовірність встановлення правильного синхронізму P_{true} не повинна бути менша за значення 0,9997 $\left(P_{true} \geq 0,9997\right)$ для будь-якого заданого значення ймовірності бітової помилки;

– імовірність встановлення хибного синхронізму P_{false} не повинна перевищувати значення $3 \cdot 10^{-4}$ $\left(P_{false} \leq 3 \cdot 10^{-4}\right)$ для будь-якого заданого значення ймовірності бітової помилки.

5.3.1. Вибір параметрів синхронізації

Значення K та l для $p_{0\_\max} = 0,495$ визначено в розділі 3 (таблиця 3.5).

Для визначення значень K та l для $p_{0\_\max} = 0,4$ скористаємося методикою, яку наведено в третьому розділі.

Нагадаємо, що відповідно до цієї методики початкове значення кількості блоків K та межі інтервалів $\left[l_{\min}(i); l_{\max}(i)\right]$ вибираються таким чином, щоб ймовірність хибної синхронізації

$$P_{false\_final}\left(n; d_{lim}; p_0; l; K\right) = \sum_{i=1}^{K} P_{false\_sum}\left(n; d_{lim}; p_0; l_{\min}(i); l_{\max}(i); i\right) \leq P_{false\_\max}, \quad (5.1)$$

де n – довжина синхрослова;

d_{lim} – максимальна кратність бітових помилок, що не призводять до хибної ідентифікації синхрослова;

$P_{false\_\max}$ – задане граничне значення ймовірності хибної кадрової синхронізації;

P_{false_sum} – імовірність помилкового фазування при послідовному збільшенні значень коефіцієнта накопичення з $l_{\min}(i)$ до $l_{\max}(i)$ для i блоків:

$$P_{false\_sum}\left(n; d_{lim}; p_0; l_{\min}(i); l_{\max}(i); i\right) \leq \sum_{j=l_{\min}(i)}^{l_{\max}(i)} P_{false}\left(n; d_{lim}; p_0; j; i\right) \quad , \quad (5.2)$$

$P_{false}\left(n;d_{lim};p_0;j;i\right)$ – імовірність хибної синхронізації у випадку прийому з каналу зв'язку i блоків, що містять j (j – непарне) фрагментів по n біт.

Обмеження для окремих доданків $P_{false\_final}\left(n;d_{lim};p_0;l;K\right)$ можна задати в такому вигляді:

$$P_{false\_sum}\left(n,d_{lim},p_0,l_{\min}\left(i\right),l_{\max}\left(i\right),i\right)\le\gamma_i\cdot P_{false\_\max},$$

причому $\gamma_i\ge 0$, $\sum_{i=1}^{K}\gamma_i=1$.

Визначення меж відрізків $\left[l_{\min}\left(i\right);l_{\max}\left(i\right)\right]$ починається з $i=1$.

Максимальне значення $l_{\max}\left(1\right)$ вибирається як мінімальне значення l, за якого ймовірність правильної синхронізації для $K=1$ не менша за задане $P_{true\_\min}$, тобто $l_{\max}\left(1\right)=\min\left(l\right):P_{true}\left(n;d_{lim};p_0;l;1\right)\ge P_{true\_\min}$.

Для модельного прикладу з $n=24$, $d_{lim}=5$ для $P_{true\_\min}=0{,}9997$ і $p_{0\_\max}=0{,}4$ значення $l_{\max}\left(1\right)=75$.

Нижні межі відрізків $\left[l_{\min}\left(i\right);l_{\max}\left(i\right)\right]$ значень коефіцієнта накопичення l, відповідних кількості блоків i, вибираються як мінімальні значення $l_{\min}\left(i\right)$, для яких виконується умова (5.2), тобто $l_{\min}\left(i\right)=\min\left(l\right):P_{false\_sum}\left(n;d_{lim};p_0;l;l_{\max}\left(i\right);i\right)\le\gamma_i\cdot P_{false\_\max}$.

Верхні межі відрізків $l_{\max}\left(i+1\right)$ обчислюються за значеннями $l_{\min}\left(i\right)$ наступним чином: $l_{\max}\left(i+1\right)=\left[l_{\min}\left(i\right)\cdot\left(i/\left(i+1\right)\right)\right]$, якщо $\left[l_{\min}\left(i\right)\cdot\left(i/\left(i+1\right)\right)\right]$ – непарне, та $l_{\max}\left(i+1\right)=\left[l_{\min}\left(i\right)\cdot\left(i/\left(i+1\right)\right)\right]-1$, якщо $\left[l_{\min}\left(i\right)\cdot\left(i/\left(i+1\right)\right)\right]$ – парне.

Для аналізованого в розділі модельного прикладу $P_{false\_\max}=3\cdot 10^{-4}$.

Визначати значення K будемо відповідно до виразу (5.1) як мінімальне значення, за якого сума $\sum_{j=1}^{\lfloor 75/K\rfloor}P_{false}\left(24;5;0{,}4;j;K\right)\le P_{false\_\max}$, j – непарне. Такі суми

для $K=\{1;2;3\}$ рівні: $\sum_{j=1}^{75} P_{false}(24;5;0,4;j;1)=2,56\cdot 10^{-1}$, $\sum_{j=1}^{37} P_{false}(24;5;0,4;j;2)=3,96\cdot 10^{-4}$ і $\sum_{j=1}^{25} P_{false}(24;5;0,4;j;3)=7,99\cdot 10^{-7}$. Звідси слідує, що $K=3$, а $P_{false\_sum}(24;5;0,4;1;l_{\max}(3);3)\ll \sum_{i=1}^{2} P_{false\_sum}(n;d_{lim};p_0;l_{\min}(i);l_{\max}(i);i)$.

Приймемо $\gamma_1=1/4$, $\gamma_2=3/4$. Тоді межі відрізків $[l_{\min}(i);l_{\max}(i)]$ набувають значень, наведених у таблиці 5.1.

Таблиця 5.1 – **Межі відрізків** $[l_{\min}(i);l_{\max}(i)]$

i	1	2	3
$l_{\min}(i)$	59	5	1
$l_{\max}(i)$	75	29	3

Кількість накопичених фрагментів $L_{fr}=Kl$, а кількість накопичених біт – $L=Kln$.

Наведемо на рисунку 5.5 графіки залежностей оцінок ймовірностей правильної та хибної синхронізації від кількості накопичених фрагментів за заданих обмежень для $K=\{1;2;3\}$ та визначених значень точок переходу між ними.

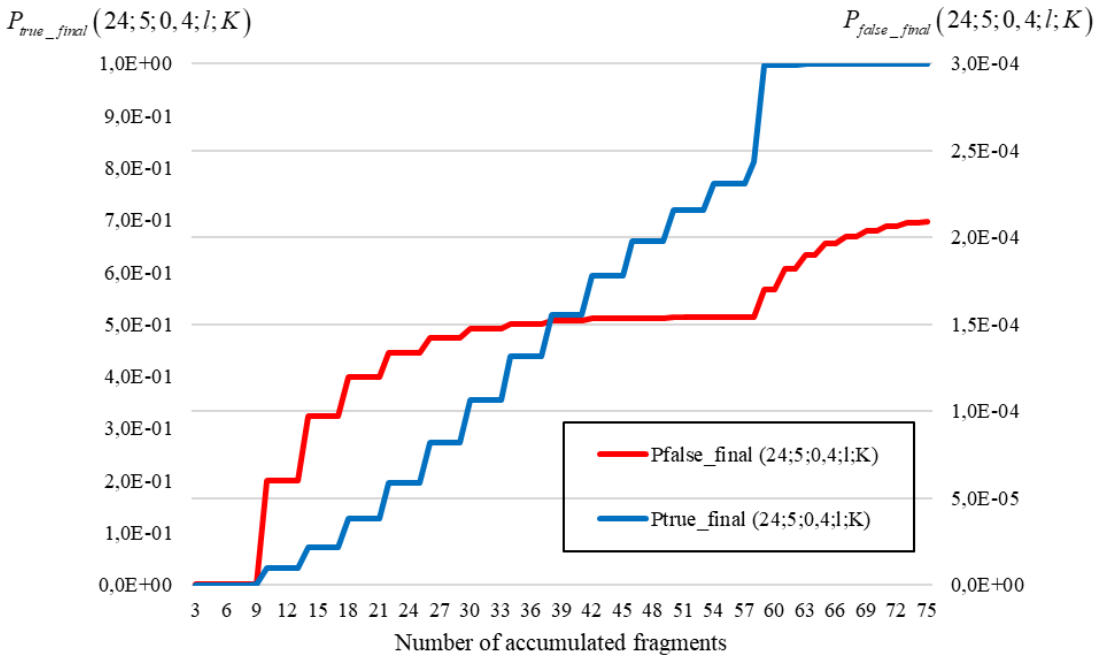


Рисунок 5.5 – Графіки залежностей оцінок імовірностей правильної та хибної синхронізації від кількості накопичених фрагментів для адаптивного процесу синхронізації

5.3.2. Експериментальні результати та їх обговорення

На рисунку 5.6 представлено графік експериментально визначеної інтегральної функції часу входження в синхронізм (відносної частоти встановлення правильного циклового синхронізму) $W_{true\_final}(24;5;p_0;l;K)$ від кількості накопичених фрагментів $L_{fr} = Kl$ (з їх перемішуванням) для ймовірності бітової помилки $p_0 = 0,4$. Крім того, на рисунку 5.6 наведено також графік теоретичної оцінки знизу ймовірності правильної синхронізації P_{true_final} для параметрів системи синхронізації з $p_{0\_\max} = 0,4$.

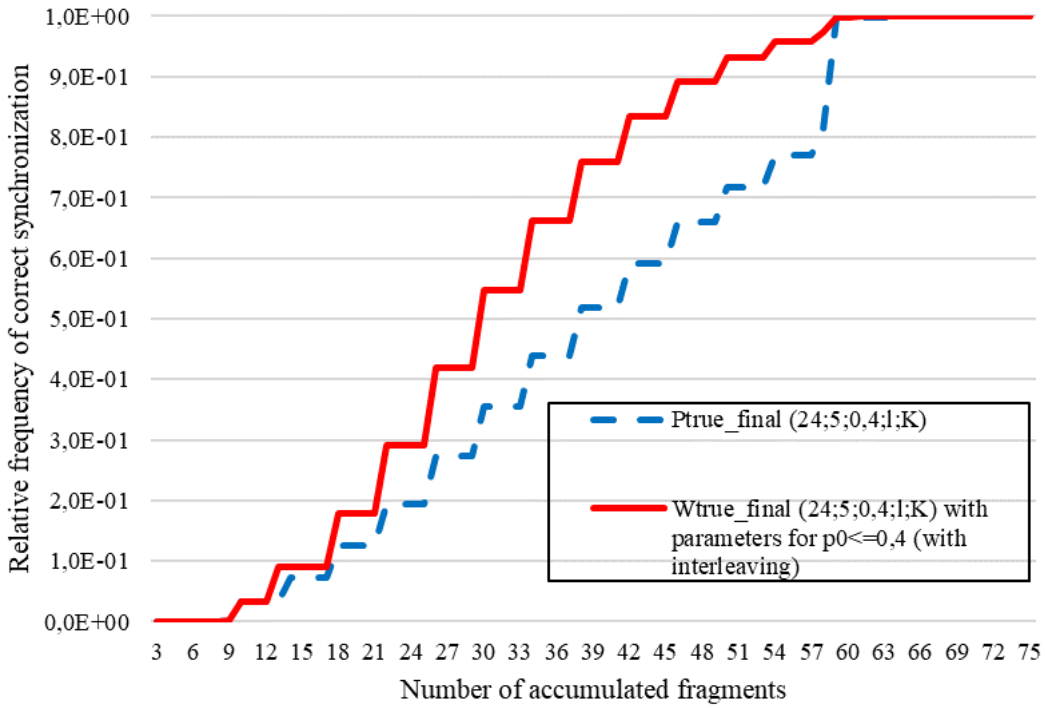


Рисунок 5.6 – Графіки залежностей відносної частоти встановлення циклового синхронізму від кількості накопичених фрагментів (з їх перемішуванням)

Зауваження 5.1. *Під час експериментального дослідження розробленого методу за граничної ймовірності бітової помилки* $p_0 = 0,4$ *встановлення хибного циклового синхронізму для 10000 випробувань траплялося в середньому 1 раз. Для 100000 таких же випробувань кількість випадків встановлення хибного циклового синхронізму складала 16-18 разів. Така ситуація повністю відповідає визначеній умові* $P_{false} \le 3 \cdot 10^{-4}$.

Аналіз продемонстрованих на рисунку 5.6 результатів підтверджує відносно грубу оцінку P_{true_final} із третього розділу. Водночас оцінка є справедливою.

Наведемо на рисунку 5.7 та порівняємо ефективність операції перемішування фрагментів перед кожним наступним етапом формування блоків.

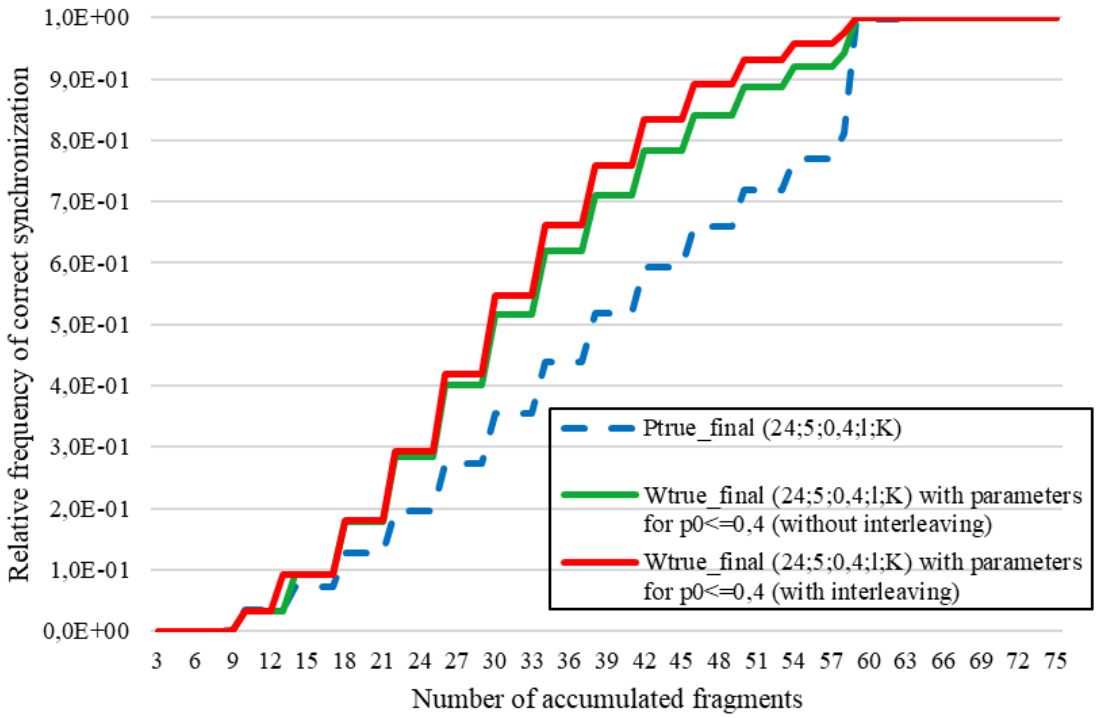


Рисунок 5.7 – Графіки залежностей відносної частоти встановлення циклового синхронізму від кількості накопичених фрагментів з перемішуванням та без перемішування (для $p_0 \le 0,4$)

Аналіз графіків рисунку 5.7 свідчить, що застосування операції перемішування фрагментів, як і для випадку з $p_{0\_max} = 0,495$, дає позитивний ефект. Встановлення меж перестановок із застосуванням перемішування буде швидшим, оскільки вимагає меншої кількості накопичених фрагментів L_{fr} (середня кількість фрагментів 31,57 з перемішуванням фрагментів проти 33,01 без перемішування).

Визначимо та представимо на рисунку 5.8 залежність середнього значення кількості накопичених фрагментів до встановлення циклового синхронізму $\overline{L_{fr}}$ від імовірності бітової помилки $p_0 \le 0,4$.

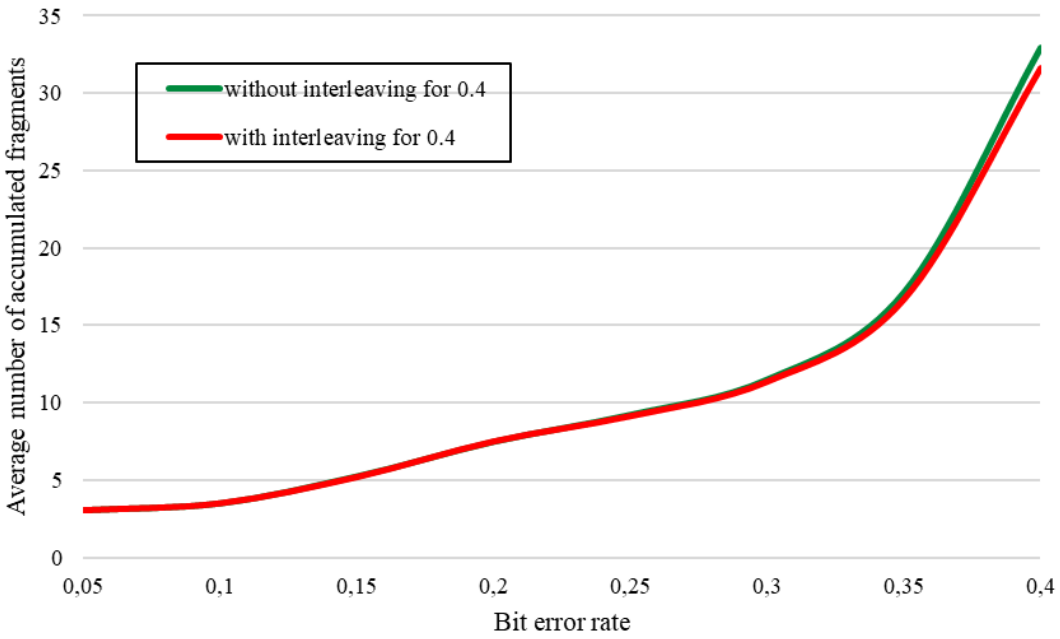


Рисунок 5.8 - Графіки середньої кількості накопичених фрагментів до встановлення циклового синхронізму $\overline{L_{fr}}$ від імовірності бітової помилки $p_0 \le 0.4$

З графіків на рисунку 5.8 видно, що різниця між середньою кількістю накопичених фрагментів $\overline{L_{fr}}$ для алгоритму з перемішування та алгоритму без перемішування збільшується зі збільшенням імовірності бітової помилки і для $p_0 = 0,4$ досягає свого максимального значення в 1,44 фрагменти.

Виконаємо порівняння показників систем циклової синхронізації з моментами зміни значень K і l, визначеними для $p_{0\_\max} = 0,4$ і для $p_{0\_\max} = 0,495$.

Рисунок 5.9 у порівнянні з рисунком 5.7 доповнено графіком відносної частоти встановлення циклового синхронізму від кількості накопичених фрагментів L_{fr} для системи синхронізації з перемішуванням фрагментів, параметри якої визначено для $p_{0\_\max} = 0,495$.

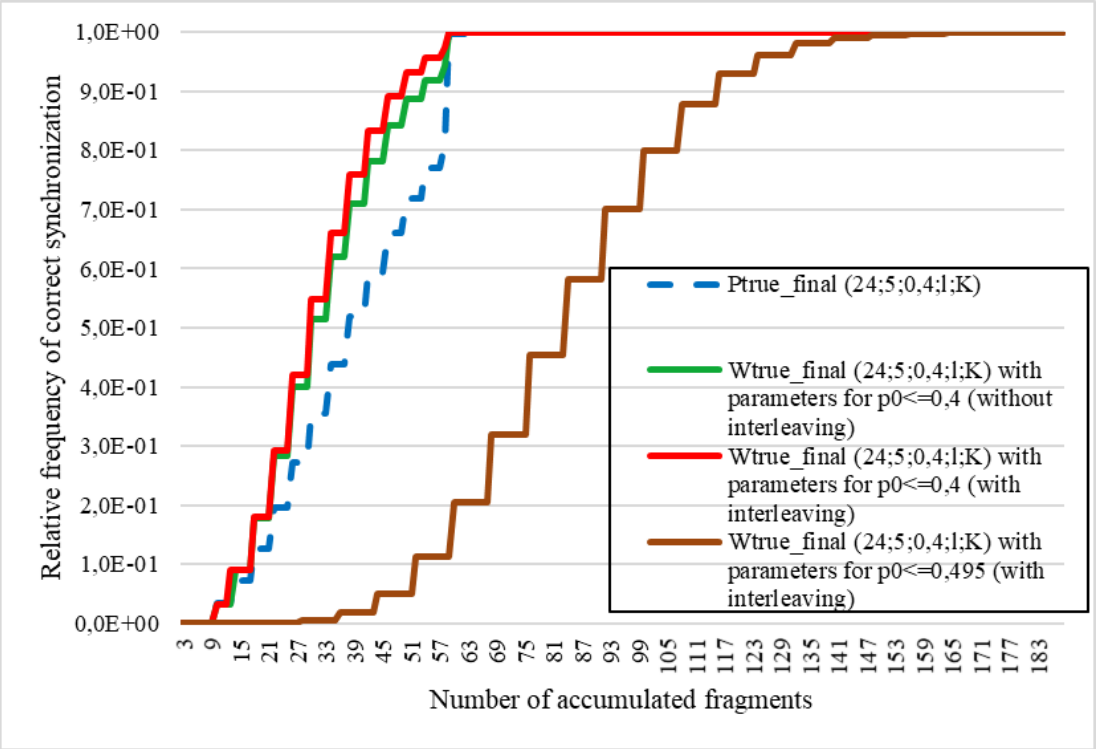


Рисунок 5.9 – Графіки відносної частоти встановлення циклового синхронізму від кількості накопичених фрагментів

Наведені графіки свідчать про те, що значення K та l, а також порядок їх зміни відіграють важливу роль в швидкодії алгоритму синхронізації.

Визначимо різницю між мінімальною кількістю накопичених фрагментів L_{fr} для системи синхронізації з перемішуванням фрагментів, параметри якої визначено для $p_{0\_\max} = 0,495$, та системи синхронізації з перемішуванням фрагментів, параметри якої визначено для $p_{0\_\max} = 0,4$, за якої досягається значення відносної частоти $W_{true\_final}(24,5,0.4,l,K) = 0,9997$. У першому випадку $L_{fr} = 172$, у другому – $L_{fr} = 69$. Таким чином, різниця в необхідній кількості накопичених фрагментів для досягнення відносної частоти встановлення правильного синхронізму на заданому рівні $0,9997$ дорівнює 103 фрагменти, а їх відношення – 2,49. Час встановлення синхронізму, очевидно, буде значно меншим для системи синхронізації з перемішуванням фрагментів, параметри якої визначено для $p_{0\_\max} = 0,4$.

Рисунок 5.10 ілюструє залежність середнього значення кількості

накопичених фрагментів до встановлення циклового синхронізму $\overline{L_{fr}}$ від імовірності бітової помилки $p_0 \le 0,4$ для систем синхронізації з параметрами K та l, визначеними для $p_{0\_max} = 0,495$ і $p_{0\_max} = 0,4$, з перемішуванням і без перемішування накопичених фрагментів.

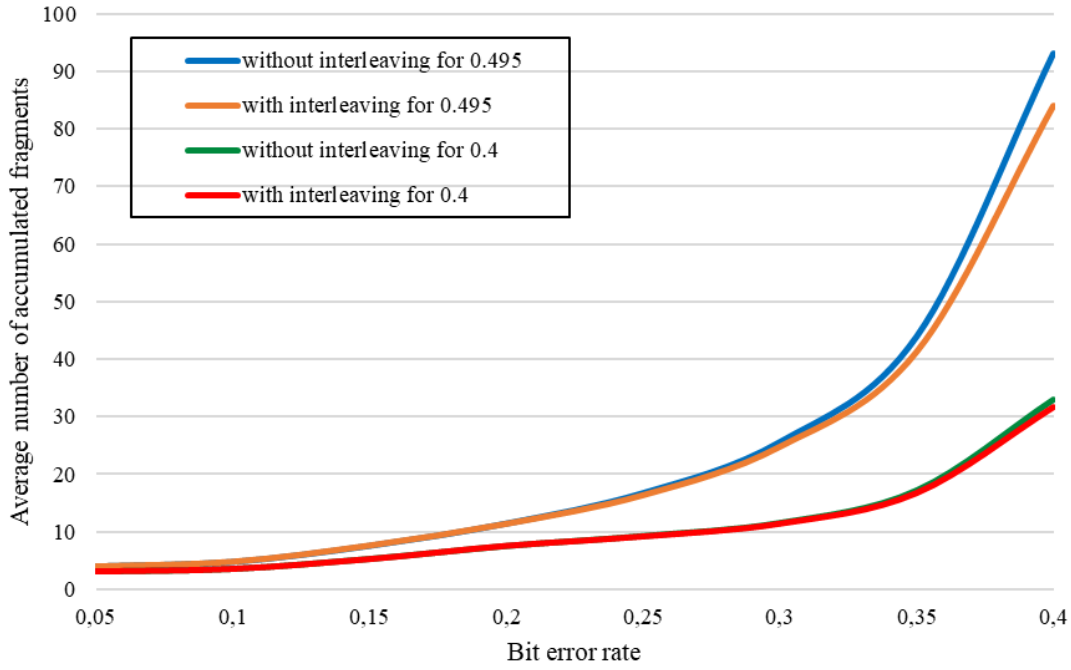


Рисунок 5.10 – Графіки середньої кількості накопичених фрагментів для $p_0 \le 0.4$

З графіків на рисунку 5.10 видно, що різниця між середнім значенням кількості накопичених фрагментів для систем синхронізації з перемішуванням і без перемішування накопичених фрагментів збільшується зі збільшенням імовірності бітової помилки. Так, ця різниця для систем синхронізації з параметрами, визначеними для $p_{0\_max} = 0,495$ дорівнює 4607 фрагментам за $p_0 = 0.495$, 9.1 фрагментам за $p_0 = 0.4$ і 2.6 фрагментам за $p_0 = 0.35$. Для систем синхронізації з параметрами, визначеними для $p_{0\_max} = 0,4$, відповідні значення дорівнюють 1,4 і 0,4. Ці показники також свідчать про те, що перемішування має більш значний позитивний ефект для системи синхронізації з параметрами для $p_{0\_max} = 0,495$.

Вихідні дані для побудови графіків рисунку 5.10, визначають те, що за ймовірності бітової помилки $p_0 = 0,4$ для встановлення циклового синхронізму системою синхронізації з перемішуванням накопичених фрагментів і параметрами, визначеними для $p_{0\_\max} = 0,4$, потрібно в середньому на 52,5 фрагменти менше, ніж для системи з параметрами, визначеними для $p_{0\_\max} = 0,495$.

Порівнюючи отримані результати різниці середньої кількості накопичених фрагментів для систем з $p_{0\_\max} = 0,4$ і $p_{0\_\max} = 0,495$ з використанням перемішування і без нього, можна зазначити, що використання точного значення ймовірності бітової помилки $\left(p_{0\_\max} = 0,4\right)$ для визначення послідовності значень K і l дає в цьому прикладі більший ефект, ніж використання перемішування фрагментів для системи синхронізації з параметрами, визначеними для неточного значення ймовірності бітової помилки $\left(p_{0\_\max} = 0,495\right)$. Разом з тим, це не означає справедливість такого твердження для інших значень p_0 і $p_{0\_\max}$, проте максимально точне знання ймовірності бітової помилки в каналі зв'язку дозволить максимально ефективно визначити процедуру зміни значень K і l.

У випадку, коли ймовірність бітової помилки в каналі зв'язку не є постійною, доцільно орієнтуватися на її максимальне значення. Використання ж адаптивної процедури зміни параметрів системи синхронізації в залежності від завадової ситуації в каналі зв'язку може дозволити підвищити ефективність системи кадрової синхронізації, проте в цьому випадку потрібно враховувати ризик збільшення частоти хибної синхронізації внаслідок збільшення ймовірності бітової помилки та інерційності процесу зміни параметрів.

5.4. Висновки

У п'ятому розділі дисертаційного дослідження:

– представлено побудовані імітаційні моделі систем передавання інформації з нероздільним ФКВД із застосуванням методів циклової синхронізації ФКВД, описаних у другому та третьому розділах;

– реалізовано алгоритми роботи імітаційних моделей, побудовані за описаними методами;

– виконано оцінювання показників ефективності цих методів циклової синхронізації (часу входження в цикловий синхронізм для різних параметрів каналу зв'язку);

– виконано порівняльний аналіз отриманих показників часу входження в цикловий синхронізм, продемонстровано особливості та переваги кожного з методів, сформульовано рекомендації щодо їх застосування. Зокрема, визначено, що за параметрів циклової синхронізації, визначених для $p_{0\_\max} = 0,495$, швидкість встановлення циклового синхронізму для $p_0 = 0.495$ вища для методу на основі поділу синхрокомбінації на префіксну й суфіксну частини, в середньому, на 11,83 %, але відносна частота хибного фазування, у цьому разі, дорівнює 0.006103662 та є вищою порівняно з методом на основі кореляційної обробки для якого це значення дорівнює 0, за 10000 експериментів. А для $p_0 = 0.496$ швидкість встановлення циклового синхронізму вища для методу на основі кореляційної обробки на 56,94%. Разом з тим, за параметрів циклової синхронізації, визначених для $p_0 = 0.495$, швидкість встановлення циклового синхронізму для $p_0 \le 0.495$ є вищою для методу на основі поділу синхрокомбінації на префіксну й суфіксну частини, а для $p_0 > 0.495$ – для методу на основі кореляційної обробки, що підтверджують графіки зображені на рисунку 5.3. Разом з тим, згідно з наведеними графіками на рисунку 5.4, реалізація першого методу має вищі значення ймовірності хибного фазування за $p_0 > 0.3$, що може не задовольняти необхідним умовам;

– виконано дослідження ефективності методу циклової синхронізації нероздільного факторіального коду на основі кореляційної обробки шляхом визначення та аналізу показників синхронізації з параметрами, обчисленими для граничних імовірностей бітової помилки $p_{0\_\max} = 0,4$ і $p_{0\_\max} = 0,495$, та наступними умовами: імовірність правильної синхронізації $P_{true} \ge 0,9997$, імовірність хибної синхронізації $P_{false} \le 3 \cdot 10^{-4}$. Підтверджено, що:

- різниця між середнім значенням кількості накопичених фрагментів для систем синхронізації з перемішуванням і без перемішування накопичених фрагментів з параметрами, визначеними для $p_{0\_\max} = 0,495$, дорівнює 4607 фрагментам за $p_0 = 0.495$, 9.1 фрагментам за $p_0 = 0.4$ і 2.6 фрагментам за $p_0 = 0.35$. Для систем синхронізації з параметрами, визначеними для $p_{0\_\max} = 0,4$, відповідні значення дорівнюють 1,4 і 0,4. Отримані показники свідчать про те, що використання процедури перемішування отриманих з каналу зв'язку фрагментів має позитивний ефект, значущість якого зростає зі збільшенням імовірності бітової помилки та зі збільшенням граничних імовірностей бітової помилки $p_{0\_\max}$, на основі яких і визначаються параметри систем синхронізації;
- різниця між мінімальною кількістю накопичених фрагментів L_{fr} для системи синхронізації з перемішуванням фрагментів, параметри якої визначено для $p_{0\_\max} = 0,495$, та системи синхронізації з перемішуванням фрагментів, параметри якої визначено для $p_{0\_\max} = 0,4$, за якої досягається значення відносної частоти $W_{true\_final}(24,5,0.4,l,K) = 0,9997$, дорівнює 103 фрагменти, а їх відношення – 2,49. Як результат, необхідний час для встановлення синхронізму значно менший для системи синхронізації з перемішуванням фрагментів, параметри якої визначено для $p_{0\_\max} = 0,4$.

Таким чином, для підвищення ефективності синхронізації доцільно додатково використовувати процедуру перемішування отриманих фрагментів та максимально точно визначати ймовірність бітової помилки в каналі зв'язку. У випадку, якщо ймовірність бітової помилки носить змінний характер, доцільно орієнтуватися на максимально можливе значення.

У п'ятому розділі детально описано середовище розробки та основні модулі, які використовувалися для реалізації алгоритмів встановлення циклової синхронізації для ФКВД.

Застосування адаптивної процедури зміни параметрів системи синхронізації в залежності від завадової ситуації в каналі зв’язку може бути напрямом подальших досліджень. Крім того, можна зазначити, що потенційно існує можливість комбінованого використання зазначених методів, однак цей напрямок виходить за рамки цього дослідження.

Основні результати порівняльного аналізу ефективності методів циклової синхронізації, викладені в цьому розділі, представлено в працях [135], [136].

ВИСНОВКИ

У дисертаційній роботі вирішено актуальну науково-технічну задачу, що полягає в забезпеченні достовірності передавання інформації на основі використання факторіального кодування даних. Ця задача передбачає необхідність створення методів встановлення циклової синхронізації та методу достовірного передавання перестановки з використанням нероздільного факторіального кодування в каналах зв'язку з високою ймовірністю бітової помилки.

Найбільш значущі результати роботи полягають у наступному:

1. Вперше розроблено метод циклової синхронізації нероздільних факторіальних кодів, який за рахунок використання як синхрокомбінації перестановки чисел, її поділу на префіксну та суфіксну частини, а також за рахунок мажоритарної обробки прийнятих фрагментів, дозволяє забезпечити циклову синхронізацію приймальної та передавальної станцій комунікаційних систем передавання інформації з нероздільним факторіальним кодуванням. Розроблено структурну схему пристрою циклової синхронізації. Побудовано програмну модель передавання даних, в якій реалізовано розроблений алгоритм встановлення циклового синхронізму Метод дає змогу для перестановки з 8 елементів (24 біт) у каналах зв'язку з імовірністю бітової помилки $p_0 = 0,495$ отримати значення відносної частоти встановлення правильного синхронізму $W_{true} = 0,993$ та відносної частоти хибної синхронізації $W_{false} = 0,006$ для максимального коефіцієнту накопичення $l_{\max} = 90603$ (2174472 біт). Крім того, метод дає змогу забезпечити встановлення синхронізму, в середньому, після отримання 4, 7, 15, 54, 14526 фрагментів (96, 168, 360, 1296, 348624 біт) за ймовірностей бітової помилки . $p_0 = 0,1; 0,2; 0,3; 0,4; 0,495$, відповідно.

2. Набув подальшого розвитку метод циклової синхронізації нероздільних факторіальних кодів, який за рахунок використання як синхрокомбінації перестановки, яка має максимальне значення мінімальної відстані Хеммінга від її двійкового представлення до всіх її циклічних зсувів, а також за рахунок кореляційної та мажоритарної обробки фрагментів даних, що передаються

каналом зв'язку, де довжина фрагмента дорівнює довжині синхрокомбінації, дозволяє забезпечити циклову синхронізацію в системах передавання даних з нероздільним факторіальним кодуванням за ймовірності бітової помилки, близької до 0,5. Розроблено структурну схему пристрою циклової синхронізації. Побудовано програмну модель передавання даних, в якій реалізовано розроблений алгоритм встановлення циклового синхронізму. Розроблений метод забезпечує ймовірність правильної синхронізації $P_{true} \geq 0{,}9997$ та ймовірність хибної синхронізації $P_{false} \leq 3 \cdot 10^{-4}$ у модельній системі передавання даних з незалежними бітовими помилками з імовірністю їх появи $p_0 = 0.495$ для синхрокомбінації-перестановки з 8 елементів (24 бітів) за максимального коефіцієнту накопичення $l_{\max} = 30603$ (734472 біт) та за середньої кількості прийнятих фрагментів, що необхідна для встановлення синхронізму, 15787 фрагментів (378888 біт). Розроблено та досліджено алгоритм перемішування накопичених фрагментів, що дозволяє додатково зменшити їх необхідну кількість: різниця між середнім значенням кількості накопичених фрагментів для систем синхронізації з перемішуванням і без перемішування накопичених фрагментів з параметрами, визначеними для $p_{0\_\max} = 0.495$, та довжини синхрокомбінації-перестановки 8 елементів (24 біта) дорівнює 4607 фрагментам (110568 бітам) за $p_0 = 0.495$, 9.1 фрагментам за $p_0 = 0.4$ і 2.6 фрагментам за $p_0 = 0.35$, що зменшує необхідний час встановлення синхронізму. Розроблений метод може бути ефективним для реалізації не тільки в системах з нероздільним факторіальним кодуванням, а й у класичних системах передавання даних, де використовується стандартний роздільник між кадрами.

3. Розроблено метод достовірного передавання перестановок, який за рахунок подання кожного елементу перестановки, що передається, у вигляді циклічного двійкового зсуву перестановки-переносника, що має максимальне значення мінімальної відстані Хеммінга від її двійкового представлення до всіх її циклічних зсувів, дозволяє забезпечити інформаційний обмін за ймовірності бітової помилки, близької до 0,5. Для підвищення достовірності передавання

перестановок метод також використовує мажоритарну та кореляційну обробку фрагментів, отриманих з каналу зв'язку. Розроблено математичну модель системи передавання даних. Розроблений метод дає змогу за ймовірності бітової помилки $p_0 = 0.495$ досягти ймовірності приймання перестановок без помилок $P_{W\_true\_final} \geq 0.999$ і ймовірність невиявленої помилки $P_{W\_false\_final} \leq 3.6 \cdot 10^{-4}$. Окрім того, результати побудованої імітаційної програмної моделі системи передавання даних підтверджують ефективність розробленого методу в порівнянні з традиційним методом DSSS: для досягнення заданої ймовірності приймання перестановок без помилок і ймовірності невиявленої помилки, що вказані вище, метод DSSS потребує приймання $l = 36413$ фрагментів (20099976 біт), в той час, як розроблений метод, потребує $l = 29123$ фрагментів (16075896 біт).

4. Досліджено ефективність методів циклової синхронізації систем передавання інформації з нероздільним факторіальним кодуванням: методу на основі використання як синхрокомбінації перестановки чисел з її поділом на префіксну та суфіксну частини, та методу, що використовує як синхрокомбінацію перестановку, що має максимальне значення мінімальної відстані Хеммінга від її двійкового представлення до всіх її циклічних зсувів. Розроблено рекомендації щодо застосування розроблених методів циклової синхронізації в каналах зв’язку з високою ймовірністю бітової помилки з використанням нероздільного факторіального кодування. Зокрема, продемонстровано, що за параметрів циклової синхронізації, визначених для $p_{0\_\max} = 0,495$, імовірність правильної синхронізації є вищою для методу на основі поділу синхрокомбінації на префіксну й суфіксну частини за $p_0 \leq 0.495$ і вищою для методу на основі кореляційної обробки за $p_0 > 0.495$. Разом з тим, варто враховувати, що реалізація методу на основі поділу синхрокомбінації на префіксну й суфіксну частини має вищі значення ймовірності хибного фазування за $p_0 > 0.3$ у порівняння з методом на основі кореляційної обробки. Для прикладу, швидкість встановлення циклового синхронізму за $p_0 = 0.4$ та $p_0 = 0.495$ вища для методу на основі поділу

синхрокомбінації на префіксну й суфіксну частини, в середньому, на 73.29% та 11,83 % відповідно, а за $p_0 = 0.496$ - вища для методу на основі кореляційної обробки, в середньому, на 56,94%; у той же час, відносна частота хибної синхронізації за $p_0 = 0.4$, $p_0 = 0.495$ та $p_0 = 0.496$ є меншою для методу на основі кореляційної обробки (0 для 10000 експериментів) порівняно з методом на основі поділу синхрокомбінації на префіксну й суфіксну частини (0.003, 0.006 та 0.008 відповідно, для 10000 експериментів). Доведено важливість правильно обраних параметрів для методу циклової синхронізації на основі кореляційної обробки в залежності від імовірності бітової помилки: різниця між мінімальною кількістю накопичених фрагментів $L_{fr.}$ для системи синхронізації з перемішуванням фрагментів, параметри якої визначено для $p_{0\_\max} = 0,495$, та системи синхронізації з перемішуванням фрагментів, параметри якої визначено для $p_{0\_\max} = 0,4$, за якої досягається значення відносної частоти $W_{true\_final}(24,5,0.4,l,K) = 0,9997$, дорівнює 103 фрагменти, а їх відношення – 2,49. Як результат, необхідний час для встановлення синхронізму значно менший для системи синхронізації з перемішуванням фрагментів, параметри якої визначено для $p_{0\_\max} = 0,4$.

СПИСОК ДЖЕРЕЛ

[1] F. Ling, *Synchronization in digital communication systems*. Cambridge: Cambridge university press, 2017.

[2] J. Massey, «Optimum Frame Synchronization», *IEEE Trans. Commun.*, том 20, №2, с. 115–119, 1972, doi: 10.1109/TCOM.1972.1091127.

[3] R. Scholtz, «Frame Synchronization Techniques», *IEEE Trans. Commun.*, том 28, № 8, с. 1204–1213, 1980, doi: 10.1109/TCOM.1980.1094813.

[4] D. Hercog, *Communication protocols: principles, methods and specifications*. Cham: Springer Nature Switzerland, 2020.

[5] H. König, *Protocol engineering*. Heidelberg ; New York: Springer, 2012.

[6] M. Popovic, *Communication protocol engineering*, 2nd edition. CRC Press, 2021.

[7] P. Kartaschoff, «Synchronization in digital communications networks», *Proc. IEEE*, том 79, №7, с. 1019–1028, 1991, doi: 10.1109/5.84979.

[8] Hansheng Wang, Xiaoyi Qin, Lieguang Zeng, і Fuqin Xiong, «Coding, decoding, and recovery of clock synchronization in digital multiplexing system», *IEEE Trans. Commun.*, том 51, №5, с. 825–831, 2003, doi: 10.1109/TCOMM.2003.811432.

[9] Y. Khlaponin, E.K. Khalifa, D. Khlaponin, A. Selyukov, A. Tolbatov, V. Tolbatov and R. Odarchenko, «Method of Improving the Stability of Network Synchronization in Multiservice Macro Networks», *CEUR Workshop, Proceedings*, том 2654, с. 786–797, 2020.

[10] A.-S. Bana, K. F. Trillingsgaard, P. Popovski, і E. De Carvalho, «Short Packet Structure for Ultra-Reliable Machine-Type Communication: Tradeoff between Detection and Decoding», в *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB: IEEE, 2018, с. 6608–6612. doi: 10.1109/ICASSP.2018.8461650.

[11] B. Lee, S. Park, D. J. Love, H. Ji, і B. Shim, «Packet Structure and Receiver Design for Low Latency Wireless Communications With Ultra-Short Packets», *IEEE Trans. Commun.*, том 66, №2, с. 796–807, 2018, doi: 10.1109/TCOMM.2017.2755012.

[12] H. Lee і Y.-C. Ko, «Physical Layer Enhancements for Ultra-Reliable Low-Latency Communications in 5G New Radio Systems», *IEEE Comm. Stand. Mag.*, том 5, №4, с. 112–122, 2021, doi: 10.1109/MCOMSTD.0001.2100002.

[13] J. Park *et al.*, «Extreme ultra-reliable and low-latency communication», *Nat Electron*, том 5, №3, с. 133–141, 2022, doi: 10.1038/s41928-022-00728-8.

[14] Y. Li, D. V. Huynh, T. Do-Duy, E. Garcia-Palacios, і T. Q. Duong, «Unmanned aerial vehicle-aided edge networks with ultra-reliable low-latency communications: A digital twin approach», *IET Signal Processing*, том 16, №8, с. 897–908, 2022, doi: 10.1049/sil2.12128.

[15] A. Traßl *et al.*, «Outage prediction for ultra-reliable low-latency communications in fast fading channels», *J Wireless Com Network*, том 2021, №1, с. 92, 2021, doi: 10.1186/s13638-021-01964-w.

[16] K. Wang, C. Pan, H. Ren, W. Xu, L. Zhang, і A. Nallanathan, «Packet Error Probability and Effective Throughput for Ultra-Reliable and Low-Latency UAV Communications», *IEEE Trans. Commun.*, том 69, №1, с. 73–84, 2021, doi: 10.1109/TCOMM.2020.3025578.

[17] B. Schneier, *Applied cryptography: protocols, algorithms, and source code in C*, 2. ed., [Nachdr.]. New York, NY: Wiley, 1996.

[18] D. M. Nguyen і S. Kim, «A quantum three pass protocol with phase estimation for many bits transfer», в *2019 International Conference on Advanced Technologies for Communications (ATC)*, Hanoi, Vietnam: IEEE, 2019, с. 129–132. doi: 10.1109/ATC.2019.8924514.

[19] A. Badawi, M. Zarlis, і S. Suherman, «Impact three pass protocol modifications to key transmission performance», *J. Phys.: Conf. Ser.*, том 1235, №1, с. 012050, 2019, doi: 10.1088/1742-6596/1235/1/012050.

[20] A. Moldovyan, D. Moldovyan, and N. Moldovyan, «Post-quantum commutative encryption algorithm», *Comput. Sci. J. Mold.*, том 81, №3, с. 299–317, 2019.

[21] S. Anatoly, F. Emil, і L. Olha, «Three-Pass Cryptographic Protocol Based on Permutations», в *2020 IEEE 2nd International Conference on Advanced Trends in*

Information Theory (ATIT), Kyiv, Ukraine: IEEE, 2020, с. 281–284. doi: 10.1109/ATIT50783.2020.9349343.

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

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

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

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

[26] 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.

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

[28] 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.

[29] 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.

[30] В. И. Грабчак, «Исследование достоверности передачи данных в АСУВ с использованием каскадных теоретико-кодовых схем», № 9, с. 13–16, 2006.

[31] 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.

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

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

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

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

[36] Е. Л. Онанченко и А. В. Лысенко, «Анализ известных методов декодирования недвоичных блоковых кодов», *Вісник Сумського державного університету*, №3, с. 100–105, 2008.

[37] 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.

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

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

[40] A. P. Stakhov, «Fibonacci Matrices, a Generalization of the ‘Cassini Formula’, and a New Coding Theory», *Chaos, Solitons & Fractals*, том 30, №1, с. 56–66, 2006.

[41] A. P. Stakhov, «The ‘Golden’ Matrices and a New Kind of Cryptography», *Chaos, Solitons & Fractals*, том 32, №3, с. 1138–1146, 2007.

[42] В. Я. Чечельницкий и Н. И. Кушниренко, «Метод криптографической передачи информации на базе эквивалентного класса совершенных двоичных решеток», *Інформатика та математичні методи в моделюванні*, том 4, №3, с. 210–218, 2014.

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

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

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

[46] J. Hamkins, «Frame synchronization without attached sync markers», в *2011 Aerospace Conference*, Big Sky, USA: IEEE, 2011, с. 1–7. doi: 10.1109/AERO.2011.5747327.

[47] Е. В. Фауре, «Методологія захисту інформації на основі факторіального кодування даних», дис. д-ра техн. наук, Київ, 2017. [Online]. Доступний у: http://er.nau.edu.ua/handle/NAU/35990

[48] Харін О.О., «Методи та засоби інтегрованого захисту інформації в телекомунікаційних системах множинного доступу на основі факторіального кодування даних», ЧДТУ, Черкаси, 2020. [Online]. Доступний у: https://er.chdtu.edu.ua/handle/ChSTU/1147

[49] Фауре, Еміль Віталійович, Швидкий, Валерій Васильович, і Щерба, Анатолій Іванович, «Контроль целостности информации на основе факториальной системы счисления», *Journal of Baku engineering university – Mathematics and computer science*, том 1, №1, с. 3–13, 2017, [Online]. Доступний у: http://er.chdtu.edu.ua/handle/ChSTU/706

[50] Е. В. Фауре та М. О. Качалова, «Дослідження процедури формування контрольної суми повного факторіального коду на основі ітераційного перетворення», представлена на Проблеми інформатизаці: Тези доповідей П’ятої Міжнародної науково-технічної конференції, Черкаси, 2017, с. 17.

[51] E. V. Faure, V. V. Shvydkyi, і V. O. Shcherba, «Combined factorial coding and its properties», *Radio Electronics, Computer Science, Control*, том 0, №3, 2016, doi: 10.15588/1607-3274-2016-3-10.

[52] Э. В. Фауре, «Факториальное кодирование с несколькими контрольными суммами», *Вісник Житомирського державного технологічного університету*, том 78, №3, с. 104–113, 2016, [Online]. Доступний у: http://vtn.ztu.edu.ua/article/view/86481/82932.

[53] Е. В. Фауре та А. Ю. Бойко, «Дослідження здатності виявлення помилок факторіальним кодом з декількома контрольними сумами», представлена на Проблеми інформатизаці: Тези доповідей П’ятої Міжнародної науково-технічної конференції, Черкаси, 2017, с. 16.

[54] Э. В. Фауре, «Факториальное кодирование с восстановлением данных», *Вісник ЧДТУ*, №2, с. 33–39, 2016.

[55] Е. В. Фауре, О. О. Харін, В. В. Швидкий, і А. І. Щерба, «Спосіб факторіального кодування з відновленням даних», 117004, 12, 2017

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

[57] Фауре Э.В., «Метод повышения эффективности факториального кодирования с восстановлением данных», *Вісник Черкаського державного технологічного університету*, №4, с. 57–61, 2016, [Online]. Доступний у: http://visnyk.chdtu.edu.ua/images/tech/4_2016/11.pdf

[58] E. V. Faure, A. I. Shcherba, і A. A. Kharin, «Factorial code with a given number of inversions», *Radio Electronics, Computer Science, Control*, том 0, №2, с. 143–153, 2018, doi: 10.15588/1607-3274-2018-2-16.

[59] Е. В. Фауре, О. О. Харін, В. В. Швидкий та А. І. Щерба, «Спосіб факторіального кодування з виявленням і виправленням помилок», 121361, 11, 2017

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

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

[62] Э. В. Фауре, «Факториальное кодирование с исправлением ошибок. Теоретическое обоснование и примеры реализации», в *Наукоемкие технологии в инфокоммуникациях: обработка информации, кибербезопасность, информационная борьба: монографія*, Харків: Лідер, 2017, с. 291–323.

[63] E.V. Faure, «Factorial Coding with Data Recovery», *Visnyk Cherkaskogo Derzhavnogo Tehnologichnogo Universitetu*, № 2, с. 33–39, 2017.

[64] E. V. Faure, «Factorial coding with error correction», *Radio Electronics, Computer Science, Control*, том 0, №3, с. 130–138, 2017, doi: 10.15588/1607-3274-2017-3-15.

[65] E. Faure, A. Shcherba, і B. Stupka, «Permutation-Based Frame Synchronisation Method for Short Packet Communication Systems», в *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Cracow, Poland: IEEE, 2021, с. 1073–1077. doi: 10.1109/IDAACS53288.2021.9660996.

[66] J. Al-Azzeh, E. Faure, A. Shcherba, і B. Stupka, «Permutation-based frame synchronization method for data transmission systems with short packets», *Egyptian Informatics Journal*, том 23, №3, с. 529–545, 2022, doi: 10.1016/j.eij.2022.05.005.

[67] W. W. Peterson і E. J. Weldon, *Error-correcting codes*, 2d ed. Cambridge: MIT Press, 1972.

[68] F. J. MacWilliams і N. J. A. Sloane, *The theory of error correcting codes*. в North-Holland mathematical library ; v. 16. Amsterdam ; New York : New York: North-

Holland Pub. Co. ; sole distributors for the U.S.A. and Canada, Elsevier/North-Holland, 1977.

[69] R. H. Morelos-Zaragoza, *The art of error correcting coding*, 2nd ed. Chichester ; Hoboken, NJ: John Wiley, 2006.

[70] W. C. Huffman і V. Pless, *Fundamentals of error-correcting codes*, This digitally printed version. Cambridge New York Melbourne $nCambridge University Press, 2010.

[71] «Data growth worldwide 2010-2025», Statista. Дата звернення: 01, 2024. [Online]. Доступний у: https://www.statista.com/statistics/871513/worldwide-data-created/

[72] Е. В. Фауре, В. В. Швидкий, і А. І. Щерба, «Method of forming reproducible and unpredictable sequence of permutations», *Bezpeka ìnf.*, том 20, №3, с. 253–258, 2014, doi: 10.18372/2225-5036.20.7552.

[73] Рудницький В.М., Фауре Е.В., Швидкий В.В., Щерба А.І., «Спосіб контролю цілісності інформації», 107655, 24, 2016

[74] Е. В. Фауре та О. О. Харін, «Пристрій кодування та декодування факторіальних кодів з виявленням і виправленням помилок», 123640, 12, 2018

[75] Э. В. Фауре, В. В. Швыдкий, і В. А. Щерба, «Method of message authentication code formation based on permutations», *Zahist ìnf.*, том 16, №4, с. 340, 2014, doi: 10.18372/2410-7840.16.7620.

[76] Фауре Е.В., Швидкий В.В., Щерба А.І., «Спосіб формування імітовставки», 106669, 10, 2016

[77] Рудницький В.М., Фауре Е.В., Швидкий В.В., Щерба А.І., «Спосіб комбінованого кодування інформації», 107657, 24, 2016

[78] C. E. Shannon, «A Mathematical Theory of Communication», *Bell System Technical Journal*, том 27, №3, с. 379–423, 1948.

[79] C. E. Shannon, «Communication theory of secrecy systems», *Bell Systems Technical Journal*, том 28, с. 656–715, 1948.

[80] E. R. Berlekamp, *Algebraic coding theory*, Revised edition. New Jersey: World Scientific, 2015.

[81] «RSA Cryptography Standard». 27, 2012.

[82] T. Elgamal, «A public key cryptosystem and a signature scheme based on discrete logarithms», *IEEE Transactions on Information Theory*, том 31, №4, с. 469–472, 1985.

[83] «Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевірка». Укрвїна, 07, 2003.

[84] «Digital Signature Standard (DSS)». USA, 2013.

[85] В. С. Василенко, А. В. Чунарьова, М. Ю. Василенко і А. В. Чунарьов, «Спосіб забезпечення цілісності інформації на базі лишково-хеммінгового коду», 75938, 25, 2012

[86] В. С. Василенко, А. В. Чунарьова, М. Ю. Василенко і А. В. Чунарьов, «Спосіб забезпечення цілісності інформації на базі коду умовних лишків», 75935, 25, 2012

[87] М. Ю. Василенко, В. С. Василенко і А. В. Чунарьов, «Спосіб забезпечення цілісності інформації на базі завадостійкого коду умовних лишків», 67988, 12, 2012

[88] Oktaviana B., Siahaan A. P. U., «Three-Pass Protocol Implementation on Caesar Cipher in Classic Cryptography», *IOSR Journal of Computer Engineering (IOSR-JCE)*, том 18, №4, 2016.

[89] A. M. 3. P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, 5th printing. CRC Press, 1996.

[90] J. Lang, «A no-key-exchange secure image sharing scheme based on Shamir's three-pass cryptography protocol and the multiple-parameter fractional Fourier transform», *Opt. Express*, том 20, №3, с. 2386, 2012, doi: 10.1364/OE.20.002386.

[91] N. H. Nguyen, N. A. Moldovyan, A. V. Shcherbacov, H. M. Nguyen, і D. T. Nguyen, «No-Key Protocol for Deniable Encryption», в *Information Systems Design and Intelligent Applications*, том 672, V. Bhateja, B. L. Nguyen, N. G. Nguyen, S. C. Satapathy, і D.-N. Le, Ред., в Advances in Intelligent Systems and Computing, vol. 672. , Singapore: Springer Singapore, 2018, с. 96–104. doi: 10.1007/978-981-10-7512-4_10.

[92] J. S. Al-Azzeh, B. Ayyoub, E. Faure, V. Shvydkyi, O. Kharin, і A. Lavdanskyi, «Telecommunication Systems with Multiple Access Based on Data Factorial Coding», *Int. J. Commun. Antenna Propag.*, том 10, вип. 2, с. 102–113.

[93] E. Faure, A. Shcherba, Y. Vasiliu and A. Fesenko, «Cryptographic Key Exchange Method for Data Factorial Coding», *CEUR Workshop, Proceedings*, том 2654, с. 643–664, 2020.

[94] S. Lin і D. J. Costello, *Error control coding: fundamentals and applications*, 2nd ed. Upper Saddle River, N.J: Pearson-Prentice Hall, 2004.

[95] J. J. Kong і K. K. Parhi, «Interleaved Convolutional Code and Its Viterbi Decoder Architecture», *EURASIP J. Adv. Signal Process.*, том 2003, №13, с. 417892, 2003, doi: 10.1155/S1110865703309126.

[96] B. Bloessl і F. Dressler, «Poster: mSync -- Frames without Preambles», в *Proceedings of the 2015 Workshop on Software Radio Implementation Forum*, Paris France: ACM, 2015, с. 11–11. doi: 10.1145/2801676.2801678.

[97] B. Bloessl і F. Dressler, «mSync: Physical Layer Frame Synchronization without Preamble Symbols», *IEEE Trans. on Mobile Comput.*, том 17, №10, с. 2321–2333, 2018, doi: 10.1109/TMC.2018.2808968.

[98] G. Durisi, T. Koch, і P. Popovski, «Toward Massive, Ultrareliable, and Low-Latency Wireless Communication With Short Packets», *Proc. IEEE*, том 104, №9, с. 1711–1726, 2016, doi: 10.1109/JPROC.2016.2537298.

[99] A. Zaman, Z. Hassan, R. Odarchenko, S. Hassan, S. Ahmed, M. Bilal, I. Ahmad, A. Faheem and V. Tiurin, «Wireless Underground Sensor Networks: Packet Size Optimization Survey - researchr publication related», *CEUR Workshop, Proceedings*, том 2616, с. 353–365, 2021. Дата звернення: 29, 2024. [Online]. Доступний у: https://researchr.org/publication/ZamanHOHABATF20/related

[100] S. S. Ullah, S. C. Liew, G. Liva, і T. Wang, «Implementation of Short-Packet Physical-Layer Network Coding», *IEEE Trans. on Mobile Comput.*, том 22, №1, с. 284–298, 2023, doi: 10.1109/TMC.2021.3071329.

[101] S. Salamat Ullah, S. C. Liew, G. Liva, i T. Wang, «Short-Packet Physical-Layer Network Coding», *IEEE Trans. Commun.*, том 68, №2, с. 737–751, 2020, doi: 10.1109/TCOMM.2019.2956920.

[102] J. Wu, W. Kim, i B. Shim, «Pilot-Less One-Shot Sparse Coding for Short Packet-Based Machine-Type Communications», *IEEE Trans. Veh. Technol.*, том 69, №8, с. 9117–9120, 2020, doi: 10.1109/TVT.2020.2995840.

[103] J. Ostman, G. Durisi, E. G. Strom, M. C. Coskun, i G. Liva, «Short Packets Over Block-Memoryless Fading Channels: Pilot-Assisted or Noncoherent Transmission?», *IEEE Trans. Commun.*, том 67, №2, с. 1521–1536, 2019, doi: 10.1109/TCOMM.2018.2874993.

[104] X. Liu i X. Zhang, «Rate and Energy Efficiency Improvements for 5G-Based IoT With Simultaneous Transfer», *IEEE Internet Things J.*, том 6, №4, с. 5971–5980, 2019, doi: 10.1109/JIOT.2018.2863267.

[105] A. T. P. Nguyen, R. Le Bidan, i F. Guilloud, «Trade-Off Between Frame Synchronization and Channel Decoding for Short Packets», *IEEE Commun. Lett.*, том 23, №6, с. 979–982, 2019, doi: 10.1109/LCOMM.2019.2913363.

[106] A. T. P. Nguyen, R. Le Bidan, i F. Guilloud, «Superimposed Frame Synchronization Optimization for Finite Blocklength Regime», в *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*, Marrakech, Morocco: IEEE, 2019, с. 1–6. doi: 10.1109/WCNCW.2019.8902552.

[107] A. T. P. Nguyen, F. Guilloud, i R. Le Bidan, «On the optimization of resources for short frame synchronization», *Ann. Telecommun.*, том 75, №11–12, с. 635–640, 2020, doi: 10.1007/s12243-020-00787-y.

[108] I. Dubrawsky, «Wireless Networks», в *Eleventh Hour Security+*, Elsevier, 2010, с. 77–88. doi: 10.1016/B978-1-59749-427-4.00006-X.

[109] P. Zhang, «Industrial control networks», в *Advanced Industrial Control Technology*, Elsevier, 2010, с. 363–427. doi: 10.1016/B978-1-4377-7807-6.10010-5.

[110] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, i E. Barker, «A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications», *NIST Special Publication 800-22, Gaithersburg, MD, US,* вип. 800, с. 163, 2001.

[111] Е. В. Фауре, В. В. Швидкий, А. І. Щерба, О. О. Харін, і Б. А. Ступка, «Метод циклової синхронізації на основі перестановок», *Вісник ЧДТУ*, вип. 4, с. 67–76, 2020, doi: 10.24025/2306-4412.4.2020.222439.

[112] V. V. Senatov, «On the real accuracy of approximation in the central limit theorem. II», *Sib. Adv. Math.*, том 27, №2, с. 133–152, 2017, doi: 10.3103/S1055134417020043.

[113] M. P. Deisenroth, A. A. Faisal, і C. S. Ong, *Mathematics for machine learning*. Cambridge ; New York, NY: Cambridge University Press, 2020.

[114] L. E. Bassham *et al.*, «A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications», *NIST*, 2010, Дата звернення: 26, 2024. [Online]. Доступний у: https://www.nist.gov/publications/statistical-test-suite-random-and-pseudorandom-number-generators-cryptographic

[115] R. Abozariba, M. K. Naeem, M. Patwary, M. Seyedebrahimi, P. Bull, і A. Aneiba, «NOMA-Based Resource Allocation and Mobility Enhancement Framework for IoT in Next Generation Cellular Networks», *IEEE Access*, том 7, с. 29158–29172, 2019, doi: 10.1109/ACCESS.2019.2896008.

[116] X. Liu і X. Zhang, «NOMA-Based Resource Allocation for Cluster-Based Cognitive Industrial Internet of Things», *IEEE Trans. Ind. Inf.*, том 16, №8, с. 5379–5388, 2020, doi: 10.1109/TII.2019.2947435.

[117] X. Liu, X. Zhai, W. Lu, і C. Wu, «QoS-Guarantee Resource Allocation for Multibeam Satellite Industrial Internet of Things With NOMA», *IEEE Transactions on Industrial Informatics*, том PP, №17, с. 2052–2061, 2019, doi: 10.1109/TII.2019.2951728.

[118] L. E. Bassham III *et al.*, «Special Publication (NIST SP) - 800-22 Rev 1a. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications», National Institute of Standards & Technology, Gaithersburg, MD, United States, 2010.

[119] E. Faure, A. Shcherba, B. Stupka, I. Voronenko, і A. Baikenov, «A Method for Reliable Permutation Transmission in Short-Packet Communication Systems», в *Information Technology for Education, Science, and Technics*, том 178, E. Faure,

O. Danchenko, M. Bondarenko, Y. Tryus, C. Bazilo, і G. Zaspa, Ред., в Lecture Notes on Data Engineering and Communications Technologies, vol. 178. , Cham: Springer Nature Switzerland, 2023, с. 177–195. doi: 10.1007/978-3-031-35467-0_12.

[120] «Метод достовірного передавання перестановок у системах зв'язку з короткими пакетами», представлена на VI Міжнародної науково-практичної конференції «Інформаційні технології в освіті, науці і техніці», Черкаси, 2022, с. 70–71.

[121] «Python 3.9.6 documentation». [Online]. Доступний у: https://docs.python.org/3

[122] *PyCharm: the Python IDE for Professional Developers by JetBrains*. JetBrains. [Online]. Доступний у: https://www.jetbrains.com/ru-ru/pycharm.

[123] Ch. Severance, *Python for Everybody: Exploring Data Using Python 3*. University of Michigan, 2016.

[124] K. Lioy, *Python Advanced Programming: The guide to learn python programming. Reference with exercises and samples about dynamical programming, multithreading, multiprocessing, debugging, testing and more*. Independently published, 2019.

[125] W. McKinney, *Python for Data Analysis*, 2nd ed. O'Reilly Media, 2017.

[126] J. VanderPlas, *Python Data Science Handbook*, 1st ed. O'Reilly Media, 2016.

[127] Rick van Hattem, *Mastering Python: Master the art of writing beautiful and powerful Python by using all of the features that Python 3.5*. Packt Publishing, 2016.

[128] G. Zaccone, *Python Parallel Programming Cookbook: Over 70 recipes to solve challenges in multithreading and distributed system with Python 3*, 2nd ed. Packt Publishing, 2019.

[129] D. Durisi, G. Liva, і Y. Polyanskiy, «Short-Packet Transmission», в *Information Theoretic Perspectives on 5G Systems and Beyond*, 1ий вид., I. Marić, S. Shamai (Shitz), і O. Simeone, Ред., Cambridge University Press, 2022.

[130] A. T. P. Nguyen, R. Le Bidan, і F. Guilloud, «Trade-Off Between Frame Synchronization and Channel Decoding for Short Packets», *IEEE Commun. Lett.*, том 23, №6, с. 979–982, 2019, doi: 10.1109/LCOMM.2019.2913363.

[131] C. Feng, H.-M. Wang, і H. V. Poor, «Reliable and Secure Short-Packet Communications», *IEEE Trans. Wireless Commun.*, том 21, №3, с. 1913–1926, 2022, doi: 10.1109/TWC.2021.3108042.

[132] A.-S. Bana, K. F. Trillingsgaard, P. Popovski, і E. de Carvalho, «Short Packet Structure for Ultra-Reliable Machine-Type Communication: Tradeoff between Detection and Decoding», в *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB: IEEE, 2018, с. 6608–6612. doi: 10.1109/ICASSP.2018.8461650.

[133] G. Durisi, T. Koch, і P. Popovski, «Toward Massive, Ultrareliable, and Low-Latency Wireless Communication With Short Packets», *Proc. IEEE*, том 104, №9, с. 1711–1726, 2016, doi: 10.1109/JPROC.2016.2537298.

[134] O. C. Ibe, *Fundamentals of applied probability and random processes*, 2nd edition. Amsterdam ; Boston: Elsevier/AP, 2014.

[135] Е. В. Фауре і Б. А. Ступка, «Залежність ефективності кадрової синхронізації нероздільних факторіальних кодів від параметрів синхронізації», Електронне моделювання, том 44, №6, с. 21–35, 2022, doi: 10.15407/emodel.44.06.021.

[136] Е. В. Фауре і Б. А. Ступка, «Імітаційне моделювання процесу встановлення циклового синхронізму в системах зв’язку з нероздільним факторіальним кодуванням», *Вісник Черкаського державного технологічного університету*, вип. 4, с. 16–24, 2021, doi: 10.24025/2306-4412.4.2021.252807.

ДОДАТКИ

Додаток А. Лістинги розрахунково-експериментальних моделей

А.1. Лістинг імітаційно-програмної моделі методу циклової синхронізації на основі поділу синхрокомбінації на префіксну та суфіксну частини

```
from random import random
from collections import Counter
from multiprocessing import Process
import os
import time
globals()


def tobin(x, s):
    bin_list = [(x >> k) & 1 for k in range(0, s)]
    last_el = bin_list.pop(2)
    first_el = bin_list.pop(0)
    bin_list.append(first_el)
    bin_list.insert(0, last_el)
    return bin_list


def bin24_to_dec8(bin_list):
    bin24_to_dec8_table = {'000': 0,
                           '001': 1,
                           '010': 2,
                           '011': 3,
                           '100': 4,
                           '101': 5,
                           '110': 6,
                           '111': 7
                           }
    list8_dec = []
    list8_bin3 = []
```

```
    for i in range(8):
        list8_bin3.append(str(bin_list[3 * i]))
        list8_bin3[i] += str(bin_list[3 * i + 1])
        list8_bin3[i] += str(bin_list[3 * i + 2])
        list8_dec.append(bin24_to_dec8_table[list8_bin3[i]])
    # print("dec final3 : ", list8_bin3)
    # print("dec final: ", list8_dec)
    return list8_dec


def compare_r_list(r_list):
    sequence = [0, 7, 1, 3, 2, 4, 6, 5]
    list_r_dec = bin24_to_dec8(r_list)
    counter_repetitions = 0
    n = 0
    m = 0
    for i in range(len(list_r_dec)):
        for j in range(len(list_r_dec)):
            if i == j:
                continue
            if list_r_dec[i] == list_r_dec[j]:
                counter_repetitions += 1
                n = i
                m = j
    counter_repetitions /= 2
    counter_equal_dec = 0
    if int(counter_repetitions) == 1:
        for i in range(8):
            if (i == n) or (i == m):
                continue
            if sequence[i] == list_r_dec[i]:
                counter_equal_dec += 1

    return counter_equal_dec
```

```
def shift_left(input_list):
    new_list_shifts = []
    new_list_shifts.extend(input_list)
    first_el = new_list_shifts.pop(0)
    new_list_shifts.append(first_el)
    return new_list_shifts


def shift_left_for_list(input_list):
    shifts_list = []
    new_list_shifts = input_list
    for i in range(24):

        first_el = new_list_shifts.pop(0)
        new_list_shifts.append(first_el)
        shifts_list.append(new_list_shifts)
        new_list_shifts = []
        new_list_shifts.extend(shifts_list[i])

    return shifts_list


def find_need_value(first_list, second_list):
    # print("Temporary list = ", first_list)
    # print("Next_list = ", second_list)
    xor_list = []
    sum_mod_2 = 0
    for i in range(24):
        xor_list.append(first_list[i] ^ second_list[i])
        sum_mod_2 += xor_list[i]
    return sum_mod_2
```

```
def find_R(received_data):
    sum_bit_list = []
    list_R = []

    for i in range(24):
        sum_bit_list.append(0)
    # print("sum_bit_list ", sum_bit_list)
    for i in range(len(received_data)):
        for j in range(24):
            sum_bit_list[j] += received_data[i][j]
    # print("sum_bit_list ", sum_bit_list)
    for i in range(24):
        if sum_bit_list[i] >= int((len(received_data)+1)/2):
            list_R.append(1)
        elif sum_bit_list[i] < int((len(received_data)+1)/2):
            list_R.append(0)

    return list_R


def create_error_received_block():

    bin_list_24 = [0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1]
    output_list = []
    count_error = 0
    for i in bin_list_24:
        probability_value = random()
        # print("Probability[", i, "]: ", probability_value)
        if probability_value <= 0.495:
            # print("Error is true")
            count_error += 1
            if i == 0:
                output_list.append(1)
```

```
            else:
                output_list.append(0)
        else:
            output_list.append(i)
    # print("Output list: ", output_list)
    # print("error count = ", count_error)
    return output_list


def main_func():
    bin24_to_dec8_table = {'000': 0,
                           '001': 1,
                           '010': 2,
                           '011': 3,
                           '100': 4,
                           '101': 5,
                           '110': 6,
                           '111': 7
                           }
    cells = 8  # number of elements
    list_min_value = []
    list_of_null = []
    sequence = [0, 7, 1, 3, 2, 4, 6, 5]
    p0 = [0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.495]  # add 0.495
    count_error_bit = [12, 60, 120, 240, 360, 480]
    size_of_block = 24

    num_of_experiment = 13
    counter_bit = 24
    # bin_list_24 = [0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1]
    bin_list_24 = [0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1]
    compare_sequence = [1, 0, 0, 0, 1, 1, 1, 0]

    # print("List of decimal", sequence)
```

```
# start experiments

list_counter_received_blocks = []
list_counter_received_blocks_with_error = []
false_phasing_error_count = 0

for i in range(num_of_experiment):
    print("Number of experiment = ", i)

    # create input dat=a with error (50 blocks)
    received_data = []
    counter_received_blocks = 0
    received_data.append(create_error_received_block())
    counter_received_blocks += 1

    while True:
        received_data.append(create_error_received_block())
        counter_received_blocks += 1
        received_data.append(create_error_received_block())
        counter_received_blocks += 1
        if counter_received_blocks == 90605:
            print("l > 90603")
            break
        # for block in range(len(received_data)):
        # print("block[{0}:]".format(block), received_data[block])
        new_R_list = find_R(received_data)
        # print("R_list:  ", new_R_list)
        # compare_r_list(new_R_list)
        new_list_for_shifts = []
        new_list_for_shifts.extend(new_R_list)
        list_of_shifts = shift_left_for_list(new_list_for_shifts)
        # print("list_of_shifts", list_of_shifts)
        # for shifts in range(len(list_of_shifts)):
```

```
# print("shift[{0}:]".format(23-shifts), list_of_shifts[23-shifts])
count_equal_block = 0
number_shifts = 0
for compare_list in list_of_shifts:
    count_equal_bit = 0
    # print("compare_list", compare_list)
    # print("compare_sequence", compare_sequence)
    for k in range(8):
        if compare_list[k] == compare_sequence[k]:
            count_equal_bit += 1
    #         print("{0} == {1}".format(compare_list[k], compare_sequence[k]))
    # print("count_equal_bit = {0}".format(count_equal_bit))
    if count_equal_bit == 8:
        count_equal_block += 1
        list_which_equal = compare_list
        number_shifts_saved = number_shifts
        # print("number_shifts: ", number_shifts)
        # print("count_equal_block = ", count_equal_block)
        # print("compare_list: ", compare_list)

    number_shifts += 1

# print("Count equal = ", count_equal_block)
counter_repetitions = 0

if count_equal_block == 1:

    final_compare_list = shift_left(list_which_equal)
    # print("final list equal: ", list_which_equal)
    # print("Final compare_list", final_compare_list)
    counter_repetitions = compare_r_list(final_compare_list)
    # print("dec final: ", bin24_to_dec8(final_compare_list))
    # print("default sequence:", sequence)
    # print("bin list", bin_list_24)
```

```
            count_equal_bit = 0
            for k in range(size_of_block):
                if bin_list_24[k] == final_compare_list[k]:
                    count_equal_bit += 1
            # print("count_equal_bit = ", count_equal_bit)

        if (count_equal_bit == 24) or (counter_repetitions == 6):
            # print("count_equal_bit == {0}, counter_repetitions = {1}\n "
            #       "and  received  sequence is equal need sequence,"
            #       " go to next experiment".format(count_equal_bit, counter_repetitions, ))
            # print("final_compare_list: ", final_compare_list)
            # print("new_R_list: ", new_R_list)
            # print("dec final: ", bin24_to_dec8(final_compare_list))
            # print("dec new_R_list: ", bin24_to_dec8(new_R_list))
            # print("default sequence:", sequence)
            # print("number_shifts_saved: ", number_shifts_saved)
            if number_shifts_saved == 22:
                # print("phasing successfully done")
                list_counter_received_blocks.append(counter_received_blocks)
            else:
                # print("phasing error")
                list_counter_received_blocks_with_error.append(counter_received_blocks)
            break
        # else:
            # print(" add 2 blocks")

        # received_data.append(create_error_received_block())
        # counter_received_blocks += 1
        # received_data.append(create_error_received_block())
        # counter_received_blocks += 1
        # print(received_data)

print("Count blocks = ", list_counter_received_blocks)
print("len count blocks = ", len(list_counter_received_blocks))
```

```
    print("Count blocks with errors = ", list_counter_received_blocks_with_error)
    print("len count blocks with errors = ", len(list_counter_received_blocks_with_error))
    # print("max = ", max(list_counter_received_blocks))
    # print("min = ", min(list_counter_received_blocks))
    # the end of experiments

    letter_counts = Counter(list_counter_received_blocks)
    print("letter_counts = ", letter_counts, "TYPE OF letter_counts: ", type(letter_counts))
    # df = pandas.DataFrame.from_dict(letter_counts, orient='index')
    # print("df = ", df, "TYPE OF df: ", type(df))
    # df.plot(kind='bar')

    # plt.show()


if __name__ == '__main__':

    start_time_ALL = time.time()
    processes = []
    for i in range(os.cpu_count()):
        print('registering process %d' % i)
        processes.append(Process(target=main_func, args=()))

    for process in processes:
        process.start()

    for process in processes:
        process.join()

    finish_time = time.time() - start_time_ALL
    print("finish_time:", finish_time)
    print("avg_time:", finish_time)
    print("End program")
```

А.2. Лістинг імітаційно-програмної моделі методу циклової синхронізації на основі кореляційної обробки

```
from random import random
from collections import Counter
from multiprocessing import Process
import os
import time


def shift_left(input_list):
    first_el = input_list.pop(0)
    input_list.append(first_el)
    return input_list


def find_need_value(first_list, second_list):
    xor_list = []
    sum_mod_2 = 0

    for i in range(24):
        xor_list.append(first_list[i] ^ second_list[i])
        if xor_list[i]:
            sum_mod_2 += 1
    return sum_mod_2


def find_need_value_list(r_lists, shifts_list):

    i = 0
    length_Ham = []
    index_list_shifts_for_compare = []

    for r_list in r_lists:
```

```
        length_Ham.append([])

        for j in range(24):
            length_Ham[i].append(find_need_value(r_list, shifts_list[j]))

        if min(length_Ham[i]) <= 5:
            index_shift = length_Ham[i].index(min(length_Ham[i]))
            index_list_shifts_for_compare.append(index_shift)
        else:
            return False
        i += 1
    return index_list_shifts_for_compare


def find_R(received_data):
    list_R = []

    for i in range(24):
        counter_true = 0
        for block in received_data:
            if block[i]:
                counter_true += 1

        if counter_true >= int((len(received_data) + 1) / 2):
            list_R.append(True)
        elif counter_true < int((len(received_data) + 1) / 2):
            list_R.append(False)

    return list_R


def create_error_received_block(p0, bin_list_24):
    output_list = []
    count_error = 0
```

```
    for i in bin_list_24:
        probability_value = random()

        if probability_value <= p0:
            count_error += 1

            if i == 0:
                output_list.append(True)
            else:
                output_list.append(False)

        else:
            output_list.append(i)

    # Amount_of_create_error.append(count_error)
    return output_list


def compare_index_shifts_list(index_shifts):
    len_list_index = len(index_shifts)
    count_equal_list = 0

    if len_list_index == 1:
        return True
    elif len_list_index > 1:

        for r in range(1, len_list_index):
            if index_shifts[r] == index_shifts[0]:
                count_equal_list += 1

    if count_equal_list == (len_list_index - 1):
        return True
    else:
```

```
        return False


def unpacking_received_data(received_data):
    result_data = []

    while len(received_data) != 0:
        result_data.extend(received_data.pop(0))
    return result_data


def func_for_p0(composition, count_iter, p0):

    #print("composition, count_iter, p0", composition, count_iter, p0)

    bin_list_24 = [False, False, False, False, True, True, False, False, True, True, False, True, False, True,
False,
                   True,
                   True, False, True, True, True, True, False, False]
    # Amount_of_create_error = []
    size_of_block = 24
    num_experiments = 126
    dict_table = composition
    frequency_print = 5

    # creating loop shifts
    first_list_bin = bin_list_24
    temporary_list = bin_list_24
    length_Ham = []
    list_of_shifts = [first_list_bin]

    for j in range(size_of_block - 1):
        next_list = []
        next_list.extend(temporary_list)
```

```
    next_list = shift_left(next_list)
    list_of_shifts.append(next_list)
    length_Ham.append(find_need_value(first_list_bin, next_list))
    temporary_list = next_list

k_l_list_result = []
k_l_list_result_err = []
sum_time = 0

count_err_phase = 0
count_good_phase = 0

for experement in range(num_experiments):
    last_time = time.time()

    # if (experement%frequency_print) == 0:
    #     print("!!!-----------------------------------------------")
    p0_iter = count_iter
    k = 4
    l = 0
    len_previous = 0
    len_max = len(dict_table[k])
    old_received_data = []

    for iter in range(p0_iter):
        received_data = []
        for_r_list = []

        if iter == len_max:
            k -= 1
            len_previous = len_max
            len_max += len(dict_table[k])

        l = dict_table[k][iter - len_previous]
```

```
    for block in range(k):
        received_data.append([])
        for subblock in range(l):
            if len(old_received_data) == 0:
                received_data[block].append(create_error_received_block(p0, bin_list_24))
            else:
                received_data[block].append(old_received_data.pop(0))

        for_r_list.append(find_R(received_data[block]))
    # ----------------------------------------------------

    index_lists_for_compare = find_need_value_list(for_r_list, list_of_shifts)

    if index_lists_for_compare:
        if compare_index_shifts_list(index_lists_for_compare):
            # print("d<=5, lists_for_compare EQUAL, go to next experiment")
            if index_lists_for_compare[0] == 0:
                k_l_list_result.append("K={0}, l={1}".format(k, l))
                # print("Good, index = ", index_lists_for_compare[0])
                # print("For:  K={0}, l={1}".format(k, l))
                count_good_phase += 1
            else:
                # print("Error phasing, index = ", index_lists_for_compare[0])
                # print("For:  K={0}, l={1}".format(k, l))
                k_l_list_result_err.append("K={0}, l={1}".format(k, l))
                count_err_phase += 1
            # break
    if iter == (p0_iter - 1):
        k_l_list_result.append("Error")
        break
    old_received_data = unpacking_received_data(received_data)

time_for_exp = time.time() - last_time
sum_time += time_for_exp
```

```
    print("Amount of good phasing = ", count_good_phase)
    print("Amount of error phasing = ", count_err_phase)
    print("Time for p0 one experiment avg : ", sum_time / num_experiments)
    letter_counts = Counter(k_1_list_result)
    print("letter_counts for p0 = ", p0, " = ", letter_counts, "TYPE OF letter_counts: ",
          type(letter_counts))

    letter_counts_err = Counter(k_1_list_result_err)
    print("letter_counts_err for p0 = ", p0, " = ", letter_counts_err, "TYPE OF letter_counts_err: ",
          type(letter_counts_err))

    return True


if __name__ == '__main__':
    print("start program")
    dict_table = {
        # 0.05: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
        # 0.1: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
        # 0.15: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
        # 0.2: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
        # 0.25: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
        # 0.3: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
        # 0.35: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
        # 0.4: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
        # 0.43: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
        # 0.45: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
        # 0.47: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
        # 0.48: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
        # 0.49: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]}
        0.495: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
        # 0.496: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
        # 0.497: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
```

```
    # 0.498: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]},
    # 0.499: {4: [1, 2083], 3: [2779, 6971], 2: [10457, 15273], 1: [30549, 30603]}
  }
  # p0_const = [0.2, 0.3, 0.4, 0.43, 0.45, 0.47, 0.475, 0.48, 0.485, 0.49, 0.495]
  p0_iter_list = []

  last_time_ALL = time.time()
  global_size_max_buffer = []

  for p0 in dict_table:
    p0_iter = 0
    for k in dict_table[p0]:
      if len(dict_table[p0][k]) > 1:
        dict_table[p0][k] = [l for l in range(dict_table[p0][k][0], dict_table[p0][k][1] + 1) if l % 2 ==
1]
        p0_iter += len(dict_table[p0][k])
      else:
        p0_iter += 1
    p0_iter_list.append(p0_iter)
    # func_for_p0(dict_table[p0], p0_iter, p0)
    processes = []
    # print("dict_table[0.495] = , p0_iter_list[0] = ", dict_table[0.495], p0_iter_list[0])
    for i in range(os.cpu_count()):  # os.cpu_count()
      print('registering process %d' % i)
      processes.append(Process(target=func_for_p0, args=(dict_table[p0], p0_iter, p0, )))

    for process in processes:
      process.start()

    for process in processes:
      process.join()

    time_for_ALL = time.time() - last_time_ALL
    print("MAX SIZE", global_size_max_buffer)
```

```
print("Time for ALL iteration: ", time_for_ALL)
print("End program")
```

А.3. Лістинг імітаційно-програмної моделі методу достовірного передавання перестановки

```
from random import random
# from random import shuffle
from collections import Counter
import time


def shift_left_for_list(input_list):
    shifts_list = []
    new_list_shifts = input_list
    for i_bit in range(24):

        first_el = new_list_shifts.pop(0)
        new_list_shifts.append(first_el)
        shifts_list.append(new_list_shifts)
        new_list_shifts = []
        new_list_shifts.extend(shifts_list[i_bit])

    return shifts_list


def find_r_new(received_data, R0_list, R1_list):
    count_block = len(received_data)

    for block in range(count_block):
        for let in range(23):
            for index_i in range(24):
                if received_data[block][let][index_i]:
                    R1_list[let][index_i] += 1
                else:
                    R0_list[let][index_i] += 1
```

```
    # print("sum_bit_list ", sum_bit_list)
    list_R = []
    for let in range(23):
        list_letter_R = []
        for index_i in range(24):
            if R1_list[let][index_i] > R0_list[let][index_i]:
                list_letter_R.append(True)
            elif R1_list[let][index_i] < R0_list[let][index_i]:
                list_letter_R.append(False)
            else:
                print("ERROR!!! R1=R0")
                return EOFError
        list_R.append(list_letter_R)
    return list_R


def find_need_value(first_list, second_list):
    xor_list = []
    sum_mod_2 = 0
    for i_bit in range(24):
        xor_list.append(first_list[i_bit] ^ second_list[i_bit])
        sum_mod_2 += xor_list[i_bit]
    return sum_mod_2


def find_need_value_list(r_word, alphabet_sub):

    i_let = 0
    length_Ham = []
    index_list_shifts_for_compare = []

    for r_letter in r_word:
        length_Ham.append([])
```

```
        for j in range(23):
            length_Ham[i_let].append(find_need_value(r_letter, alphabet_sub[j]))

        if min(length_Ham[i_let]) <= 5:
            index_shift = length_Ham[i_let].index(min(length_Ham[i_let]))
            index_list_shifts_for_compare.append(index_shift)
        else:
            return False
        i_let += 1
    return index_list_shifts_for_compare


def create_error_received_block(p0_err, bin_list_24):
    output_list = []
    count_error = 0

    for bit in bin_list_24:
        probability_value = random()

        if probability_value <= p0_err:
            count_error += 1

            if not bit:
                output_list.append(True)
            else:
                output_list.append(False)

        else:
            output_list.append(bit)

    # Amount_of_create_error.append(count_error)
    return output_list
```

```
def function_p0(p0_sub, num_of_experiment_sub, alphabet_sub, create_word_sub):

    num_of_experiment_with_error_associate = 0
    num_of_experiment_not_unique = 0
    num_of_experiment_good = 0
    num_of_experiment_error = 0
    num_of_experiment_many_blocks = 0
    count_received_list = []
    if num_of_experiment_sub == 1000:
        frequency_print = 100
    elif num_of_experiment_sub == 10000:
        frequency_print = 100
    elif num_of_experiment_sub == 100:
        frequency_print = 10
    else:
        frequency_print = 100

    for exp in range(num_of_experiment_sub):
        if (exp % frequency_print) == 0:
            print(f"Start {exp}-th experiment!!!")
            print(f"num_of_experiment_good  for {exp} experiments with p0={p0_sub} "
                  f"{num_of_experiment_good}")
            print(f"num_of_experiment_error  for {exp} experiments with p0={p0_sub}: "
                  f"{num_of_experiment_error}")
            print(f"num_of_experiment_with_error_associate  for {exp} "
                  f"experiments with p0={p0_sub}: {num_of_experiment_with_error_associate}")
            print(f"num_of_experiment_not_unique  for {exp} "
                  f"experiments with p0={p0_sub}: {num_of_experiment_not_unique}")
            print(f"num_of_experiment_many_blocks  for {exp} "
                  f"experiments with p0={p0_sub}: {num_of_experiment_many_blocks}")
            List_with_count_received_block = Counter(count_received_list)
            print(
                f"List_with_count_received_block for {exp} experiments with p0={p0_sub}:"
                f"\n {List_with_count_received_block}")
```

```
# receiving data
received_word_all = []
# adding bir error
received_word = []
for i_let in range(size_of_alphabet):
    received_word.append(create_error_received_block(p0_sub, create_word_sub[i_let]))
received_word_all.append(received_word)
# print(f"Received word:  {received_word}")
CHECK_DONE = False
count_received = 1

R0_list = []
R1_list = []
for let in range(23):
    R0_list.append([])
    R1_list.append([])
for ind in range(23):
    for jek in range(24):
        R0_list[ind].append(0)
        R1_list[ind].append(0)

while not CHECK_DONE:
    # if count_received == 36413:
    #     num_of_experiment_many_blocks += 1
    #     break
    # adding new data
    for block in range(2):
        received_word = []
        for j in range(size_of_alphabet):
            received_word.append(create_error_received_block(p0_sub, create_word[j]))
        received_word_all.append(received_word)
        # print(f"Received word:  {received_word}")
    count_received += 2
```

```
        # print(f'Before R0_list: {R0_list}')
        # print(f'Before R1_list: {R1_list}')
        R_result = find_r_new(received_word_all, R0_list, R1_list)
        # print(f'After R0_list: {R0_list}')
        # print(f'After R1_list: {R1_list}')
        # print(f'Received word R:{R_result}')
        received_word_all.clear()
        index_choose_received = find_need_value_list(R_result, alphabet_sub)
        if index_choose_received:
            if len(set(index_choose_received)) != 23:
                num_of_experiment_not_unique += 1
                # print("letter(s) not unique!!!")
                continue
            # print(f' index origin : {index_random_choose}\n index choose : {index_choose_received}')
            count_of_equal_letters = 0
            for i_let in range(size_of_alphabet):
                if index_choose_received[i_let] == index_random_choose[i_let]:
                    count_of_equal_letters += 1
            if count_of_equal_letters == size_of_alphabet:
                num_of_experiment_good += 1
                # print("GOOD PHASING!!!")
                CHECK_DONE = True
                count_received_list.append(count_received)
            else:
                num_of_experiment_error += 1
                print("ERROR PHASING!")
                print(f' index origin : {index_random_choose}\n index choose : {index_choose_received}')
                CHECK_DONE = True
        else:
            num_of_experiment_with_error_associate += 1
            # print("ERROR WITH ASSOCIATE PROCEDURE!!!")
            # print("NEED MORE DATA!!!")
    # print(f"Count of received block data = {count_received}")
```

```
        # print(f"End {exp}-th experiment!!!")

    print(f"num_of_experiment_good  for {num_of_experiment_sub} experiments with p0={p0_sub} "
          f"{num_of_experiment_good}")
    print(f"num_of_experiment_error  for {num_of_experiment_sub} experiments with p0={p0_sub}: "
          f"{num_of_experiment_error}")
    print(f"num_of_experiment_with_error_associate  for {num_of_experiment_sub} "
          f"experiments with p0={p0_sub}: {num_of_experiment_with_error_associate}")
    print(f"num_of_experiment_not_unique  for {num_of_experiment_sub} "
          f"experiments with p0={p0_sub}: {num_of_experiment_not_unique}")
    print(f"num_of_experiment_many_blocks  for {num_of_experiment_sub} "
          f"experiments with p0={p0_sub}: {num_of_experiment_many_blocks}")

    global_error_count.append(num_of_experiment_error)
    List_with_count_received_block = Counter(count_received_list)
    print(
        f"List_with_count_received_block for {num_of_experiment_sub} experiments with
p0={p0_sub}:"
        f"\n {List_with_count_received_block}")
    return List_with_count_received_block


if __name__ == "__main__":
    # letter_dec = [0, 1, 7, 3, 2, 6, 4, 5]
    # letter = [0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1]
    # letter_dec = [0, 7, 1, 3, 2, 6, 4, 5]
    # letter = [0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1]
    # letter_dec = [0, 7, 1, 3, 2, 5, 4, 6]
    # letter = [0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0]
    letter = [False, False, False, False, False, True, True, True, True, False, True, True, False, True, False,
              True, False, True, True, False, False, True, True, False]

    p0_const = [
        # 0.05, 0.1, 0.15,
```

```
    # 0.2,
    # 0.25, 0.3, 0.35,
    # 0.4,
    # 0.41, 0.42, 0.43, 0.44,
    # 0.45,
    # 0.46, 0.47,
    # 0.48,
    # 0.49,
    # 0.495,
    # 0.496,
    # 0.497,
    0.498
]
num_of_experiment = 10000
global_bad_count = []
global_good_count = []
global_error_count = []
global_bad_let_count = []
global_good_let_count = []
global_error_let_count = []
global_p0_result = []
global_many_blocks = []
# create alphabet
size_of_alphabet = 23
size_of_letter = 24
temporary_list = []
temporary_list.extend(letter)

alphabet = []
alphabet.extend(shift_left_for_list(temporary_list)[0:size_of_alphabet])

# print(f"Create alphabet: ")
# for let in alphabet:
#     print(let)
```

```
    # print(f"Create alphabet len = {len(alphabet)}")

    # create data to send
    # index_random_choose = []
    # index_random_choose.extend(range(23))
    # shuffle(index_random_choose)
    index_random_choose = [15, 11, 6, 3, 2, 7, 21, 22, 0, 1, 5, 8, 20, 17, 9, 10, 14, 19, 12, 18, 13, 16, 4]

    create_word = []
    temporary_alphabet = []
    temporary_alphabet.extend(alphabet)

    for index in index_random_choose:
        create_word.append(temporary_alphabet[index])

    # print(f"Create word: ")
    # for let in create_word:
    #     print(let)
    # print(f"Create word len = {len(create_word)}")
    # print(f"Create word: ", create_word)

    result_list = []
    for p0 in p0_const:
        # if p0 <= 0.4:
        #     num_of_experiment = 1000
        # else:
        #     num_of_experiment = 1000
        my_time = time.time()
        result_list.append(function_p0(p0, num_of_experiment, alphabet, create_word))
        print(f"Result time: {time.time()-my_time}")

    # for i in range(len(p0_const)):
    #             print(f"num_of_experiment_error      for      {num_of_experiment}      experiments      with
p0={p0_const[i]}:"
```

```
    #         f"{global_error_count[i]}")
    #             print(f"List_with_count_received_block for {num_of_experiment} experiments with
p0={p0_const[i]}:"
    #         f"\n {result_list[i]}")

    print("The end!!!")
```

Додаток Б. Список публікацій здобувача за темою дисертації та відомості про апробацію результатів дисертації

Наукові праці, в яких опубліковані основні наукові результати дисертації

[1] Е. В. Фауре, В. В. Швидкий, А. І. Щерба, О. О. Харін, і Б. А. Ступка, «Метод циклової синхронізації на основі перестановок», *Вісник черкаського державного технологічного університету*, вип. 4, с. 67–76, 2020, doi: 10.24025/2306-4412.4.2020.222439.

[2] Е. В. Фауре і Б. А. Ступка, «Імітаційне моделювання процесу встановлення циклового синхронізму в системах зв'язку з нероздільним факторіальним кодуванням», *Вісник Черкаського державного технологічного університету*, вип. 4, с. 16–24, 2021, doi: 10.24025/2306-4412.4.2021.252807.

[3] J. Al-Azzeh, E. Faure, A. Shcherba, і B. Stupka, «Permutation-based frame synchronization method for data transmission systems with short packets», *Egypt. Inform. J.*, том 23, №3, с. 529–545, 2022, doi: 10.1016/j.eij.2022.05.005. **(Scopus, Q1)**

[4] E. Faure, A. Shcherba, M. Makhynko, B. Stupka, J. Nikodem, і R. Shevchuk, «Permutation-Based Block Code for Short Packet Communication Systems», *Sensors*, 22, №14, с. 5391, 2022, doi: 10.3390/s22145391. **(Scopus, Q1)**

[5] Е. В. Фауре і Б. А. Ступка, «Залежність ефективності кадрової синхронізації нероздільних факторіальних кодів від параметрів синхронізації», *Електронне моделювання*, том 44, № 6, с. 21–35, 2022, doi: 10.15407/emodel.44.06.021.

[6] E. Faure, A. Shcherba, B. Stupka, I. Voronenko, і A. Baikenov, «A Method for Reliable Permutation Transmission in Short-Packet Communication Systems», в Information Technology for Education, Science, and Technics, том 178, E. Faure, O. Danchenko, M. Bondarenko, Y. Tryus, C. Bazilo, і G. Zaspa, Ред., в Lecture Notes on Data Engineering and Communications Technologies, vol. 178., Cham: Springer Nature Switzerland, 2023, с. 177–195. doi: 10.1007/978-3-031-35467-0_12. **(Scopus)**

Наукові праці, які засвідчують апробацію матеріалів дисертації

[1] E. Faure, A. Shcherba, і B. Stupka, «Permutation-Based Frame Synchronisation Method for Short Packet Communication Systems», в *2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Cracow, 22-25 sempember 2021*, Cracow, Poland: IEEE, 2021, с. 1073–1077. doi: 10.1109/IDAACS53288.2021.9660996. **(Scopus).**

[2] Е.В. Фауре, А.І. Щерба, Б.А. Ступка, А.С. Байкенов, «Метод достовірного передавання перестановок у системах зв'язку з короткими пакетами», в *Інформаційні технології в освіті, науці і техніці (ІТОНТ-2022): Тези доповідей VI Міжнародної науково-практичної конференції, Черкаси, 23-25 червня 2020р.,* Черкаси: ЧДТУ, 2020, с. 70-71.

Наукові праці, які додатково відображають наукові результати дисертації

[1] Е. В. Фауре, В. В. Швидкий, О.О. Харін, А.О. Лавданський, Б.А. Ступка, «Спосіб циклової синхронізації», Україна. Пат. 148842, 22.09.2021.

[2] Е. В. Фауре, В. В. Швидкий, О.О. Харін, А.О. Лавданський, Б.А. Ступка, «Система циклової синхронізації», Україна. Пат. 148847, 22.09.2021.

[3] Е. В. Фауре, А. І. Щерба, А.О. Лавданський, Б.А. Ступка, «Спосіб циклової синхронізації», Україна. Пат. 150959, 18.05.2022.

[4] Е. В. Фауре, А. І. Щерба, А.О. Лавданський, Б.А. Ступка, «Система циклової синхронізації», Україна. Пат. 150883, 04.05.2022.

[5] Е. В. Фауре, А. І. Щерба, М.В. Махинько, Б.А. Ступка, «Спосіб прогнозування потужності нероздільного факторіального коду», Україна. Пат. 152846, 19.04.2023.

[6] Е. В. Фауре, А. І. Щерба, М.В. Махинько, Б.А. Ступка, «Спосіб побудови нероздільного факторіального коду», Україна. Пат. 152845, 19.04.2023.

[7] Е. В. Фауре, А. І. Щерба, А.О. Лавданський, К.В. Базіло, Б.А. Ступка, «Спосіб циклової синхронізації», Україна. Пат. 153803, 30.08.2023.

Апробацію результатів дисертації проведено на:

- 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), (Cracow, Poland, September 22-25) – дистанційна участь;
- VI Міжнародній науково-практичній конференції «Інформаційні технології в освіті, науці і техніці» (ІТОНТ-2022), (Черкаси, 23-25 червня 2022 року) – очна участь.