

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка автоматизованого модуля
для декодування блокових кодів турбо-добутків»

Виконав: студент 2-ого курсу групи 1АКІТР-23м
спеціальності 174 – Автоматизація,
комп'ютерно-інтегровані технології
та робототехніка

Мельник Є. С.

Керівник: к.т.н., доцент, доцент каф. АІТ

Іванов Ю. Ю.

« 11 » 12 2024 р.

Опонент: к.т.н., доцент, професор каф. КСУ
Бішков М. М.

« 11 » 12 2024 р.

Допущено до захисту

зав. кафедри АІТ

д.т.н., проф. Бісікало О. В.

« 16 » 12 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет

Факультет інтелектуальних інформаційних технологій та автоматизації

Кафедра автоматизації та інтелектуальних інформаційних технологій

Рівень вищої освіти II-ий (магістерський)

Галузь знань – 17 – Електроніка, автоматизація та електронні комунікації

Спеціальність – 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка

Освітньо-професійна програма – Інтелектуальні комп'ютерні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ

д.т.н., проф. Бісікало О. В.

«19» 09 2024 р.

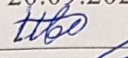
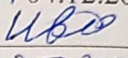
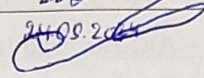
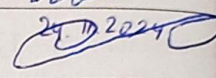
ЗАВДАННЯ

НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Мельнику Євгенію Сергійовичу

1. Тема роботи: Розробка автоматизованого модуля для декодування блокових кодів турбо-добутків.
Керівник роботи: к.т.н., доцент каф. АІТ Іванов Ю. Ю.
Затверджені наказом ВНТУ від «17» 09 2024 року № 310.
2. Строк подання роботи студентом: до 19.12.2024 р.
3. Вихідні дані до роботи: код турбо-добуток; спискове декодування; пороговий параметр $Q = [0, 1]$; кількості ненадійних бінарних символів t та ітерацій декодування $I \leq 10$; відношення сигнал/шум до 5 дБ; набори бінарних та мультимедійних даних.
4. Зміст текстової частини: вступ; теоретична основа та аналіз предметної області роботи; розробка модифікованого математичного апарату для декодування блокових кодів турбо-кодів-добутків; розробка програмного забезпечення та експериментальне дослідження; економічний розділ; висновки; список літератури.
5. Перелік ілюстративного матеріалу: структура системи передавання даних; структура кодера; структура декодера; схема експерименту; схема модифікованого методу генерації векторів помилок; схема головної програми для декодера.

6. Консультанти розділів магістерської кваліфікаційної роботи

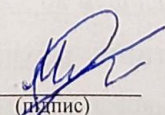
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1–3	Іванов Ю. Ю. к.т.н., доцент каф. АІТ	20.09.2024 	, 04.12.2024 
4	Козловський В. О., к.е.н., проф. каф. ЕПВМ	24.09.2024 	24.12.2024 

7. Дата видачі завдання: «20» _____ 09 _____ 2024 р.

КАЛЕНДАРНИЙ ПЛАН

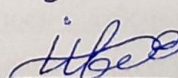
№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Теоретична основа та аналіз предметної області роботи	до 11.10.24	Виконано
2	Розробка модифікованого математичного апарату для декодування блокових кодів турбо-добутків	до 26.10.24	Виконано
3	Розробка програмного забезпечення та експериментальні дослідження	до 10.11.24	Виконано
4	Економічний розділ	до 24.11.24	Виконано
5	Оформлення пояснювальної записки та ілюстративного матеріалу	до 30.12.24	Виконано
6	Попередній захист роботи	до 05.12.24	Виконано
7	Остаточний захист роботи	до 19.12.24	Виконано

Студент


(підпис)

Мельник Є. С.
(прізвище та ініціали)

Керівник роботи


(підпис)

Іванов Ю. Ю.
(прізвище та ініціали)

АНОТАЦІЯ

УДК 519.725 + 004.942

Мельник Є. С. Розробка автоматизованого модуля для декодування блокових кодів турбо-добутків. Магістерська кваліфікаційна робота зі спеціальності 174 – Автоматизація, комп'ютерно-інтегровані технології та робототехніка, освітньо-професійна програма – Інтелектуальні комп'ютерні системи управління. Вінниця: ВНТУ, 2024. 110 с.

На укр. мові. Бібліогр.: 31 назва; рис.: 27; табл.: 9.

У даній роботі розроблено математичне, алгоритмічне та програмне забезпечення для кодування / декодування кодів турбо-добутків, представлено математичний апарат модифікованого декодера. Протестовано систему передавання даних та оцінено ефективність роботи кодека.

Ключові слова: дані, завадостійкість, код турбо-добуток, кодек, ітеративне декодування, декодер.

ABSTRACT

Melnyk E.S. Development of an Automated Module for Decoding Block Turbo-Product Codes. Master's thesis in specialty 174 – Automation, computer-integrated technologies and robotics, educational and professional program – Intelligent computer control systems. Vinnytsia: VNTU, 2024. 110 p.

In Ukrainian language. Bibliography: 31 titles; fig.: 27; tabl.: 9.

In this work mathematical, algorithmic tools and software for encoding / decoding turbo-product codes was developed, and the mathematical apparatus of the modified decoder was presented. The data transmission system was tested and the effectiveness of the codec was evaluated.

Keywords: data, error correction, turbo-product code, codec, iterative decoding, decoder.

ЗМІСТ

ВСТУП	4
1 ТЕОРЕТИЧНА ОСНОВА ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	
РОБОТИ	7
1.1 Основні поняття	7
1.2 Постановка задачі.....	13
1.3 Методи завадостійкого кодування	18
1.4 Принципи декодування кодів турбо-добутків	26
1.5 Висновки	30
2 РОЗРОБКА МОДИФІКОВАНОГО МАТЕМАТИЧНОГО АПАРАТУ...	
ДЛЯ ДЕКОДУВАННЯ БЛОКОВИХ КОДІВ ТУРБО-ДОБУТКІВ	32
2.1 Дослідження об'єкта автоматизації.....	32
2.2 Алгебра логарифмічного відношення функцій правдоподібності	35
2.3 Структура турбо-кодека.....	39
2.4 Математичний апарат турбо-декодера.....	43
2.5 Запропонована модифікація для декодера.....	52
2.6 Висновки	55
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА	
ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ	56
3.1 Обґрунтування вибору середовища розробки	56
3.2 Опис програми.....	58
3.3 Експериментальні дослідження.....	63
3.4 Висновки	73
4 ЕКОНОМІЧНИЙ РОЗДІЛ	74

4.1 Технологічний аудит розробленої модифікації.....	74
4.2 Розрахунок витрат на розробку та проведення досліджень.....	78
4.3 Прогнозування комерційного ефекту від можливої комерціалізації.....	
розробки.....	83
4.4 Висновки.....	89
ВИСНОВКИ.....	90
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	91
ДОДАТКИ.....	96
Додаток А (обов'язковий). Технічне завдання.....	97
Додаток Б (обов'язковий). Ілюстративна частина	100
Додаток В (обов'язковий). Лістинг програмного забезпечення	104
Додаток Г (обов'язковий). Довідка про впровадження	107
Додаток Д (обов'язковий). Протокол перевірки МКР.....	109

ВСТУП

Актуальність теми. У час сучасних технологій відбувається інтенсивний розвиток інформаційно-комунікаційних систем різного функціонального призначення (супутниковий і мобільний зв'язок, цифрове телебачення, програмовані радіосистеми). Такі системи представляють собою територіально розподілену сукупність обчислювальних секторів із високошвидкісними каналами передавання, апаратними та програмними засобами, які виконують автоматизоване збирання, збереження, оброблення та передавання даних [1].

На дані під час передавання діють завади різної фізичної природи, створюючи помилки. Для боротьби з ними застосовується завадостійке кодування даних. Зі збільшенням трафіку та швидкостей передавання інформації зростає інтерес до ефективних технік завадостійкого кодування, із яких виділяються турбо-подібні коди, рекомендовані до використання у сучасних стандартах зв'язку, оскільки дозволяють працювати на показниках, близьких до теоретичної межі [2].

У наукових працях [3–7] показано, що клас блокових кодів турбо-добутків ефективно працює з невеликими кодовими блоками на дуже високих кодових швидкостях, переважаючи низку інших завадостійких кодів за певних умов. Такі коди мають меншу за потужні згорткові турбо-коди складність декодування, а також у них практично відсутній “поріг насичення”. Сучасні декодери кодів турбо-добутків представляють ресурсомісткі компоненти апаратури та містять складні математичні методи, дослідження яких актуальні.

Результати теоретичної та експериментальної роботи в даному напрямку відображено в наукових роботах закордонних та вітчизняних авторів,

наприклад, F.L. Morgos [8], K. Suman [9], B.O. Зубенко [10], B.O. Перов [11] та інших. Вчені акцентують увагу на методі ітеративного спискового декодування, який є ефективним та зручним для практичної реалізації, але при виборі тестових символів та гіпотетичних кодових слів виникає екстремальна дискретна оптимізаційна задача [9, 12].

Отже, актуальною є науково-практична задача розробки оптимізованих обчислювальних процедур для спеціалізованих методів декодування у пошуці компромісу за певними критеріями якості в рамках системи зв'язку з метою підвищення обчислювальної ефективності декодування та економії ресурсів.

Метою роботи є зменшення обчислювальної складності декодування блокового коду турбо-добутку за рахунок спрощення процедури вибору ненадійних символів у зашумленій послідовності даних без значного зниження ефективності роботи системи передавання даних.

Для досягнення мети необхідно розв'язати такі *задачі*:

- проаналізувати роботу завадостійких кодів;
- розробити модифікацію математичного апарату для декодування турбо-добутків та виконати його теоретичне дослідження;
- розробити програмні засоби та експериментально дослідити ефективність роботи розробленого кодека.

Об'єктом дослідження є процес декодування даних у системах передавання даних.

Предметом дослідження є моделі, методи та інструментальні засоби декодування кодів турбо-добутків у системах передавання даних.

Методи дослідження. У роботі використано елементи теорії ймовірності, теорію завадостійкого кодування, основи передавання та оброблення даних,

аналітичні методи дослідження математичних моделей для аналізу декодера коду турбо-добутку; імітаційне моделювання – для перевірки достовірності отриманих теоретичних результатів.

Наукова новизна роботи полягає у тому, що запропоновано модифікацію методу спискового декодування кодів турбо-добутків, особливістю якої є використання евристичної процедури вибору ненадійних зашумлених символів, що дозволяє виконати порівняльний аналіз складності декодування без значного зниження ефективності роботи системи передавання даних.

У роботі отримані результати, які мають *практичну цінність*, а саме: наведено експериментальні дослідження передавання даних із використанням кодека блокових кодів турбо-добутків.

Апробація результатів та публікації. За результатами даної роботи опубліковано тези доповіді з міжнародної науково-практичної конференції (м. Полтава, 2024) [12]. Результати досліджень можна застосувати у ході розробки програмного модуля для обробки та передавання даних, що підтверджено довідкою про можливість впровадження розробки.

1 ТЕОРЕТИЧНА ОСНОВА ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ РОБОТИ

1.1 Основні поняття

Інформація – об'єкт вивчення та дослідження розділу кібернетики під назвою теорія інформації, який займається математичним описом та оцінкою методів передавання, збереження, пошуку і класифікації інформації, а також вивчає і досліджує загальні сторони інформаційних процесів різної природи – технічної (сигнали в електронних схемах), біологічної (біопотоки), економічної, соціальної тощо. Теорія інформації вивчає кількісну сторону інформації без семантичного змісту та цінності для отримувача.

Аналізом інформаційних потоків займається інформатика – наука, що вивчає та досліджує інформаційні процеси та системи в соціальній сфері, їх роль, методи побудови, механізм взаємодії з практичною діяльністю людини та посилення цієї взаємодії за допомогою засобів обчислювальної техніки.

Оброблення інформації – вся сукупність операцій (збирання, введення, записування, перетворення, зчитування, зберігання, знищення, реєстрація), що здійснюються за допомогою технічних і програмних засобів, включаючи обмін каналами передавання даних [1, 13].

За визначенням академіка В.М. Глушкова, процеси, де основною продукцією, яка переробляється, є інформація, називаються інформаційними технологіями. Причому, інформація розглядається поданою в знаковій формі: у вигляді текстів повідомлень, відомостей, вихідних даних, таблиць, графіків.

Загалом теорія інформаційної безпеки передбачає три види перетворення, які доповнюють одне одного, а їх комплексне застосування допомагає ефективно використовувати канали зв'язку для надійного захисту інформації: криптографічне шифрування та дешифрування, завадостійке кодування та декодування і ущільнення даних (ефективне кодування). Спільним для цих видів перетворень є те, що дані певним чином змінюють форму подання, зберігаючи інформацію [1].

Надійність систем передавання даних у міру їх удосконалення постійно зростає, але все одно у роботі можливі помилки, як систематичні, так і випадкові. Помилки апаратури ведуть до спотворення або втрати даних, що передаються або зберігаються. Сигнал у каналі зв'язку може бути спотворений завадами, поверхня магнітного носія може бути пошкоджена, в роз'ємі може бути втрачений контакт. Історія завадостійкого кодування почалася в 1948 році з публікації знаменитої статті К.Е. Шеннона, в якій він сформулював ряд важливих теорем, довівши існування коригуючих кодів, але не вказавши методу їх знаходження. Ця подія стала стимулом до розробки теорії завадостійкого кодування, яка для кожного конкретного каналу зв'язку дозволяє вибрати ефективний метод виявлення і виправлення помилок.

У сучасному світі, в якому велика кількість даних зберігається та передається через різноманітні системи й мережі, захист інформації стає критично важливим завданням для багатьох організацій та індивідуальних користувачів. Ця теорія є невід'ємною частиною інформаційних технологій і сприяє надійності та безпеці використання даних у різних сферах. Передавання, отримання, обробка та зберігання різних типів інформації є обов'язковими для

функціонування будь-якої системи, і в цьому процесі завжди відбувається обмін даними між її складовими частинами.

Призначення завадостійкого кодування – захист інформації від завад та помилок під час передавання та зберігання інформації. Без використання завадостійкого кодування було б неможливо передавати великі обсяги інформації (файли), тому що в будь-якій системі передавання та зберігання інформації неминуче виникають помилки. Залежно від того, який підхід використовується в системі – виявлення або виправлення помилок, розрізняють такі варіанти [1]:

1) запит повторного передавання (automatic repeat request, ARQ): за допомогою завадостійкого коду виконується лише виявлення помилок, за їх наявності виконується запит на повторне передавання пакета даних;

2) пряме виправлення помилок (forward error correction, FEC): проводиться декодування завадостійкого коду, тобто виправлення помилок за його допомогою.

3) можливий також гібридний варіант, наприклад, отримали пакет інформації, спробували виправити помилки, а якщо не змогли це зробити, тоді надсилається запит на повторне передавання даних.

У даний час широко поширені цифрові системи зв'язку, у яких здійснюється передавання бінарних даних. Найбільш поширені два підходи до мінімізації ймовірності помилки [1]:

1) збільшення енергії сигналу перед передаванням. Це досить важко реалізувати, наприклад, якщо на розроблювану апаратуру систем зв'язку

накладені різні обмеження: економічні, ергономічні і технічні (габарити передавача, його енергоспоживання тощо).

2) застосування завадостійкого кодування. Використовуються певні правила співставлення кожному конкретному повідомленню строго певної комбінації символів. Широке застосування завадостійкого кодування в сучасних цифрових системах зв'язку обумовлено тим, що воно є ефективним засобом приведення параметрів системи до бажаного компромісу між достовірністю, потужністю та швидкістю передавання даних.

Основним принципом завадостійкого кодування є використання надлишкових кодів, причому якщо для кодування повідомлення використовується простий код, то в нього спеціально вводять надлишковість. Це пояснюється тим, що в простих кодах всі кодові комбінації є дозволеними, тому помилка в будь-якому з розрядів призведе до появи іншої дозволеної комбінації, і виявити помилку буде неможливо. Такий вид кодування дозволяє зменшити ймовірність бітової помилки за рахунок внесення надлишковості в передані повідомлення, тобто застосовується код, в якому використовуються не всі можливі комбінації, а лише деякі з них. Ця особливість на приймальній стороні використовується для боротьби з бінарними помилками. Відповідно такі коди називають надлишковими або коригуючими [13].

Процес введення надлишковості (рисунок 1.1), який полягає у перетворенні інформаційних символів у кодове слово, називається кодуванням. Відповідно, зворотний процес, що відновлює інформацію з кодового слова, є декодуванням. Ефективність завадостійких кодів у виявленні та виправленні

помилки залежить від правил їх побудови, параметрів та методів декодування.

Надлишковість коригуючого коду може бути абсолютною і відносною.



Рисунок 1.1 – Приклад надлишковості
для систематичного коду

Абсолютна надлишковість визначається як співвідношення між обсягом доданої надлишкової інформації та початковим обсягом даних. Чим вища надлишковість, тим більше інформації додається до даних для забезпечення їхньої надійності. Під абсолютною надлишковістю $r_{абс}$ розуміють кількість додаткових символів [13]:

$$r_{абс} = n - k, \quad (1.1)$$

де n – кількість кодових символів на виході;

k – кількість інформаційних символів на вході.

Відносною надлишковістю $r_{відн}$ коригуючого коду називають величину:

$$r_{відн} = \frac{r_{абс}}{n} = \frac{n - k}{n} = 1 - \frac{k}{n} = 1 - R. \quad (1.2)$$

Кодова відстань d або відстань Хеммінга характеризує ступінь відмінності будь-яких двох кодових комбінацій. Вона виражається кількістю розрядів, в яких комбінації відрізняються одна від одної [2]:

$$d(X, Y) = W(X \oplus Y) = \sum_{i=1}^m |x_i - y_i|, \quad (1.3)$$

де W – вагова функція;

$X = (x_1, \dots, x_m)$, $Y = (y_1, \dots, y_m)$ – кодові слова (вектори).

Особливу важливість для характеристики коригувальних властивостей коду має мінімальна кодова відстань $d_{\min}$ – найменша відстань Хеммінга, яка визначається при попарному порівнянні всіх кодових комбінацій:

$$d_{\min} = \min_{i=1, \dots, q} (\{d_i\}), \quad (1.4)$$

де $q = 0,5(n^2 + n)$ – кількість порівнянь;

d_i – відстань Хеммінга між i -ою парою кодових слів.

У надлишкових кодах для передавання повідомлень використовується лише частина кодових комбінацій (дозволені комбінації). Приймання забороненої кодової комбінації означає помилку. Якщо прийнята кодова комбінація в результаті дії завад перейшла з однієї дозволеної комбінації в іншу, то визначити помилку неможливо, навіть при використанні завадостійкого кодування. Якщо ж передана дозволена кодова комбінація, в

результаті дії завад, переходить у заборонену комбінацію, то в цьому випадку існує можливість виявити помилку і виправити її [13].

Кратність гарантовано виявлених і виправлених помилок залежить від мінімальної кодової відстані $d_{\min}$. Під кратністю розуміється кількість пошкоджених завадами символів кодової комбінації. Відстані d між різними комбінаціями деякого конкретного коду можуть істотно відрізнятися. Так, зокрема, в безнадлишковому коді — ця відстань для різних комбінацій може змінюватися від одиниці до величини n .

Мінімальна кодова відстань є характеристикою завадостійких кодів, яка вказує на гарантовану кількість помилок e , які виправляє даний код, тобто [4]:

$$e \leq \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor, \quad (1.5)$$

де $\lfloor x \rfloor$ — округлення до найближчого цілого від x в меншу сторону.

1.2 Постановка задачі

Важливою складовою завадостійкого кодування є розуміння теореми, яка фактично визначає достовірність передавання даних. Теорема Шеннона-Хартлі: верхня теоретична межа пропускну здатності C каналу зв'язку з адитивним білим гаусівським шумом (АБГШ) за умови, що [1]

$$R \leq C, \quad (1.6)$$

визначається у вигляді:

$$C = f(W, S, N) = W \cdot \log_2 \left(1 + \frac{S}{N} \right). \quad (1.7)$$

де W – ширина полоси пропускання (Гц);

S – потужність прийнятого сигналу (дБ);

N – середня потужність шуму (дБ).

Значення W , S та N дозволяють визначити межі швидкості передавання даних, але не ймовірність появи помилки. Формулу (1.7) можна представити із застосуванням нормованого відношення сигнал / шум $\frac{E_b}{N_o}$ замість S/N .

Потужність шуму визначимо у такому вигляді:

$$N = N_o \cdot W, \quad (1.8)$$

де N_o – спектральна щільність потужності шуму.

Застосувавши рівняння (1.8) та (1.7), отримаємо, що:

$$\frac{C}{W} = \log_2 \left(1 + \frac{S}{N_o \cdot W} \right). \quad (1.9)$$

де $\frac{C}{W}$ – нормована пропускна здатність каналу зв'язку.

Якщо прийняти, що $R = C$ (гранична умова теореми Шеннона), то отримаємо:

$$\frac{S \cdot T_b}{N_0} = \frac{S}{R \cdot N_0} = \frac{S}{C \cdot N_0} = \frac{E_b}{N_0}, \quad (1.10)$$

де E_b – енергія некодованого біта;

T_b – час передавання символу.

Виконавши ряд перетворень в (1.10), маємо:

$$\frac{C}{W} = \log_2 \left(1 + \frac{E_b \cdot C}{N_0 \cdot W} \right) = \log_2(1 + z), \quad (1.11)$$

де $z = \frac{E_b}{N_0} \cdot \frac{C}{W}$.

Тоді з виразу (1.11) можна визначити, що [4]:

$$\frac{E_b}{N_0} = \left(\frac{W}{C} \right) \cdot \left(2^{\frac{C}{W}} - 1 \right) = \frac{2^{\frac{C}{W}} - 1}{\frac{C}{W}} = \frac{2^\eta - 1}{\eta}, \quad (1.12)$$

де $\frac{W}{C}$ – нормована полоса пропускання каналу зв'язку, а $\eta = \frac{C}{W}$.

При розробці системи зв'язку головною метою є мінімізація показника

$\frac{E_b}{N_0}$ для заданого значення BER , що представляє собою складну оптимізаційну

задачу, цільову функцію якої можна представити у такому вигляді [1, 12]:

$$p \approx BER = f\left(\frac{E_b}{N_0}\right), \quad \frac{E_b}{N_0} \rightarrow \min. \quad (1.13)$$

Щоб знайти граничне значення $\frac{E_b}{N_0}$, при якому для будь-якої швидкості передавання R не існуватиме можливості здійснити безпомилкове передавання даних, застосуємо рівність (1.11) та умову $\lim_{x \rightarrow 0} (1+x)^{1/x} = e$. Отримаємо:

$$\lim_{z \rightarrow 0} \left(\frac{C}{W} \right) = \lim_{z \rightarrow 0} \left(z \cdot \log_2 (1+z)^{1/z} \right), \quad (1.14)$$

$$\frac{C}{W} = z \cdot \log_2 e \rightarrow \frac{E_b}{N_0} \approx 0,693. \quad (1.15)$$

Перетворивши значення $\frac{E_b}{N_0}$ в децибели, маємо границю Шеннона (позиція 1 на рисунку 1.2):

$$\frac{E_b}{N_0}_{dB} = 10 \cdot \lg \frac{E_b}{N_0} \approx -1,6 \text{ (дБ)}. \quad (1.16)$$

Робоча точка 2 на рисунку 1.2 вказує на відсутність кодування даних. Позиції 3–7 формують з точкою 2 траєкторії ефективності системи $T_{234} < T_{256} < T_{276}$.

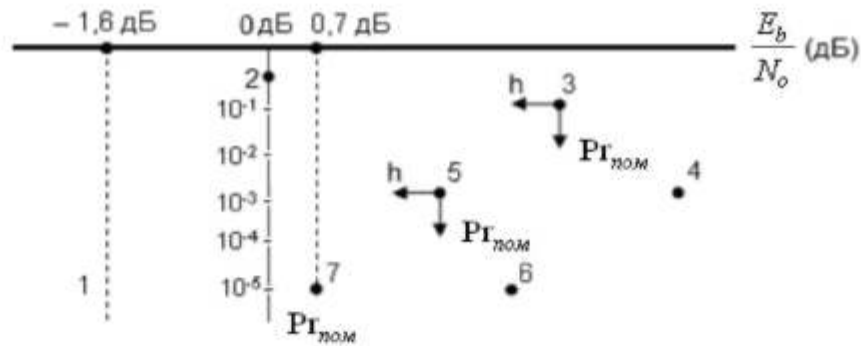


Рисунок 1.2 – Траєкторії ефективності системи передавання даних

З теорем К.Е. Шеннона [1, 2, 4] можна зробити важливий практичний висновок про те, що застосування завадостійкого кодування економічно доцільне, оскільки дозволяє отримати найбільш загальний показник для якісної оцінки методів кодування – енергетичний резерв, який показує зниження енергії, необхідної для передавання даних, при певній середній імовірності помилки на біт у разі використання заданих методів кодування, в порівнянні з випадком, коли кодування немає взагалі (рисунок 1.3).

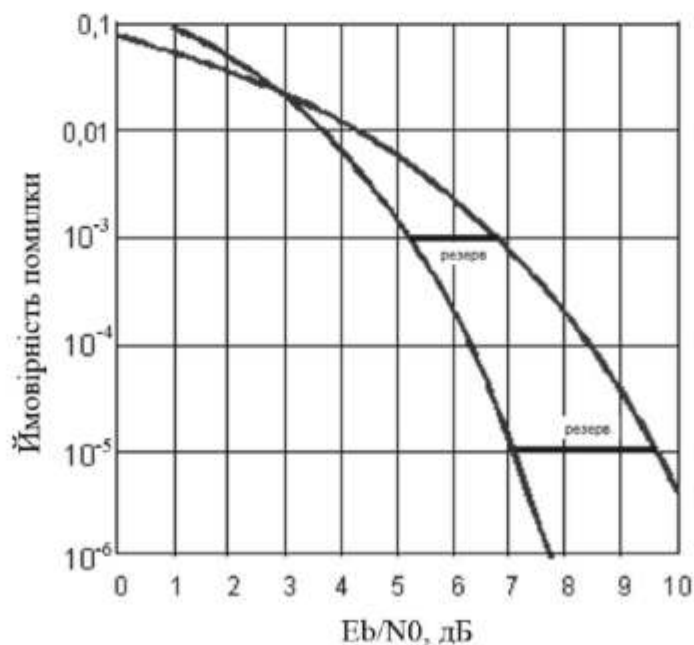


Рисунок 1.3 – Графічне означення резерву

Економічна цінність енергетичного виграшу надзвичайно актуальна і вимагає впровадження теоретичних досягнень з області завадостійкого кодування при створенні систем передавання цифрових даних різного функціонального призначення.

До задач, для яких знайдені задовільні розв'язки засобами завадостійкого кодування, слід віднести [4]:

- кодування з метою забезпечення необхідного для практичного використання значення BER у каналах передавання даних різної природи (канальний і фізичний рівні);
- кодування в системах зі зворотними зв'язками;
- кодування для захисту від несанкціонованого доступу;
- кодування для ущільнення інформації з втратами.

Методи канального кодування вирішують питання захисту від завад і стають актуальними у зв'язку зі стрімким розвитком цифрових систем передавання даних. Але найбільш проблемними залишаються радіоканали зв'язку, які особливо важливі для сучасних технологій.

1.3 Методи завадостійкого кодування

Блоковий код працює за принципом кодування фіксованих блоків інформаційних символів. Він об'єднує k інформаційних символів у кодове слово з n символів, де швидкість R визначається як відношення:

$$R = \frac{k}{n}. \quad (1.17)$$

Це означає, що чим більша кількість символів у кодовому слові порівняно з інформаційними символами, тим нижча швидкість коду і більше надлишкової інформації для підвищення надійності передавання.

Деревовидний код, або код з розгалуженою структурою, є більш складним, оскільки він обробляє нескінченну послідовність інформаційних символів, які надходять зі швидкістю k_0 символів за один інтервал часу. Він перетворює цю послідовність в неперервну послідовність кодових символів зі швидкістю n_0 символів за інтервал часу. Це забезпечує більш гнучке кодування, що підходить для потоків даних, і дозволяє адаптувати швидкість вихідної послідовності кодових символів, роблячи деревовидні коди ефективними для передавання великих масивів даних в реальному часі.

У залежності від методів надання надлишковості неперервні та блокові коди діляться на нероздільні та роздільні. Нероздільні коди не мають розділення кодової комбінації на інформаційні і перевірочні символи. У роздільних відокремлена роль окремих символів (інформаційні та перевірочні, тобто ті, які використовують для знаходження і виправлення помилок). Роздільні блокові коди – це (n, k) -коди, де n – довжина кодових комбінацій, а k – кількість інформаційних символів (довжина такого коду $l = n + k$). Роздільні блокові коди можна поділити на систематичні і несистематичні [4].

Систематичні коди складають більшість роздільних кодів. У них перевірочні символи знаходяться у ході лінійних операцій над інформаційними символами. Для двійкових кодів перевірочний символ вибирається так, щоб

його сума за модулем два з даними інформаційними символами була нульовою. Тоді декодування можна звести до перевірки на парність деяких груп символів. В результаті перевірок отримується інформація про наявність помилок та позицію таких символів.

Для опису лінійних кодів розроблений апарат лінійної алгебри. Формування нелінійних кодів відбувається на основі нелінійних процедур, що дозволяє отримати коди з рядом унікальних властивостей [2].

За допомогою експериментів з каналами зв'язку було виявлено, що помилки символів групуються в кадри (пошкоджена послідовність символів) різної довжини. Для знаходження місць кодової комбінації, де наявні помилки, застосовується n -розрядний вектор (синдром коду), що являє собою n -розрядну комбінацію символів, в якій одиниці показують положення "перекручених" символів.

Зі зростанням довжини кодової комбінації збільшується складність реалізації процедур кодування та декодування, що обумовлює труднощі практичної реалізації кодеків. У прикладній теорії кодування поряд з оцінками коригуючої здатності кодів прийнято оцінювати складність реалізації процедур кодування / декодування, які можуть бути реалізовані програмними або апаратними засобами. При цьому аргументом функції складності повинна виступати довжина кодової комбінації n .

Складність кодування блокових кодів зазвичай оцінюють величиною, пропорційною кількості елементів породжуючої матриці коду [1]

$$O(n) = n^2 \cdot (1 - R). \quad (1.18)$$

Алгоритми декодування виявляються складнішими. Серед них найбільш складним прийнято вважати алгоритм повного перебору, відповідно до якого декодер отримує на вході певну комбінацію A , порівнює всі дозволені комбінації із нею, а потім приймає рішення про передавання тієї із комбінацій, яка виявляється на мінімальній відстані від комбінації A (декодування за мінімальною відстанню). Складність такого алгоритму прийнято вважати пропорційною кількості всіх можливих комбінацій коду, тобто [1]

$$O(n) = m^n, \quad (1.19)$$

де m – кількість символів у алфавіті коду.

Складність переборного декодування зростає експоненціально з ростом довжини коду n , виникає проблема "прокляття розмірності": будь-яка спроба реалізувати декодування дуже довгих кодів виявляється безуспішною через катастрофічне зростання складності програмної або апаратної реалізації алгоритму декодування.

Існують різні види завадостійких кодів, але сучасна теорія цифрового зв'язку за принципами організації (побудови) простору кодових комбінацій розрізняє три основні напрямки: блокові, неперервні (деревовидні) і турбо-подібні коди, які відрізняються структурою, принципами побудови та коригувальною здатністю (рисунок 1.4) [1].

Для опису методів кодування / декодування як блокових, так і згорткових кодів застосовують адекватний математичний апарат. Для лінійних кодів використовується апарат лінійної алгебри, а формування нелінійних кодів

виконується із застосуванням нелінійних процедур, які в певних випадках дають можливість отримати коди зі спеціальними властивостями.

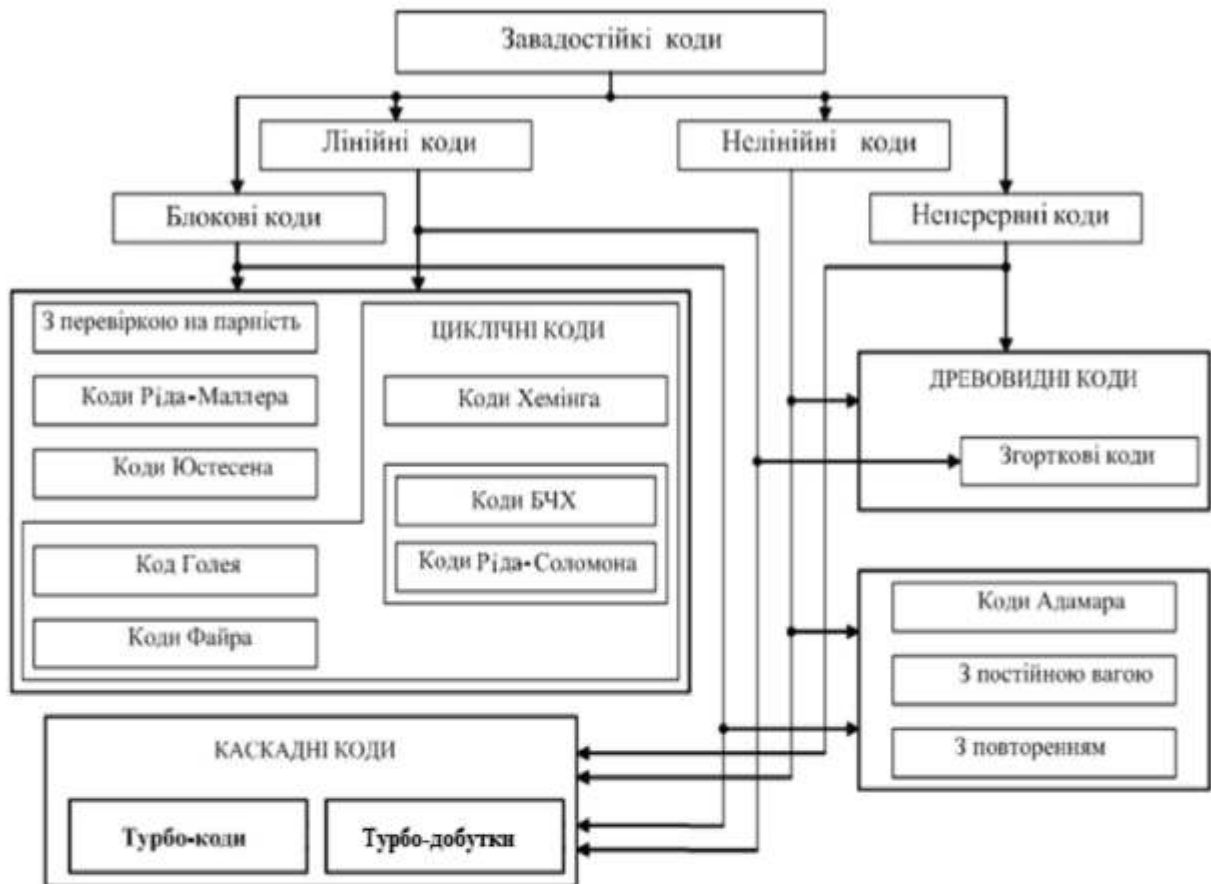


Рисунок 1.4 – Огляд методів завадостійкого кодування

У теорії завадостійкого кодування важливою є проблема складності реалізації методів кодування / декодування й, особливо, саме методів декодування. Тому певні класи блокових кодів (Хеммінга, Боуза-Чоудхурі-Хоквінгема, Ріда-Соломона, Файра) були розроблені разом з методами декодування, пов'язаними зі структурними властивостями цих кодів. І, навпаки, розробка нових методів декодування згорткових кодів (алгоритми Вітербі, Бала-Кока-Джелінека-Равіва, послідовне та порогове декодування) ініціювала пошуки відповідних згорткових кодів.

Формування блокових кодів передбачає розбиття переданих цифрових послідовностей на окремі блоки, які подаються на вхід кодера. Кожному такому блоку на виході кодера відповідає блок кодових символів, робота кодера визначається алгоритмом кодування. Блокові коди історично були запропоновані й вивчені на початку розвитку теорії кодування.

Блоковий код представляє собою код без пам'яті. Кодер блокового коду відображає блок довжиною із k інформаційних символів, який подається на вхід, у кодову послідовність з n вихідних символів. Термін “без пам'яті” вказує на те, що кожен блок з n символів залежить тільки від відповідного блоку з k символів і не залежить від інших блоків.

Багато кодів мають імена вчених, які їх запропонували й досліджували. Таким прикладом є неперервний код Фінка-Хагельбаргера, який розробили радянський учений Фінк і німецький фахівець Хагельбаргер. Цей код довго був прикладом неперервного коду із простим алгоритмом кодування / декодування, але, після відкриття згорткових кодів Елайесом, поступився їм місце [1].

Формування неперервних кодів здійснюється неперервно в часі, без поділу на блоки, що й визначає назву цього класу кодів. У даному класі кодів слід виділити згорткові коди, які за характеристиками перевершують блокові коди. Вони мають високу коригуючу здатність як для випадкових, так і для пакетних помилок. Кодер згорткового коду є пристроєм з пам'яттю. На його вхід (аналогічно до блокового) надходять послідовності з k вхідних інформаційних символів, а на виході з'являються послідовності з n кодових символів. Але при цьому кожна послідовність з n кодових символів залежить від поточної вхідної послідовності і від v попередніх вхідних символів. Отже,

кодер повинен містити пристрій пам'яті на m вхідних символів, а параметр $K = m + 1$ називають довжиною кодового обмеження.

Основні переваги коригуючих кодів (як блокових, так і згорткових) спонукали пошуки нових підходів до реалізації шляхів підвищення завадостійкості й ефективності систем цифрового зв'язку.

Результатом цих пошуків стала каскадна конкатенація блокових або згорткових кодів. Такий код представляє собою високоефективний каскадний блоковий систематичний код з високою коригуючою здатністю та спеціальними властивостями компонентних кодів.

На практиці для турбо-подібних кодів існують певні реалізаційні особливості, про які слід згадати. Типова крива частоти виникнення помилок має три основні області: спочатку відбувається поступове зменшення BER (bit error rate) – область неузгодженості, потім різке зменшення – “скеляста область”, а потім зустрічається “плато насичення”, в якій нахил кривої різко зменшується (рисунк 1.5). При цьому, для досягнення малої величини BER (10^{-9} – 10^{-11}) необхідні значні витрати енергетичних ресурсів [2].

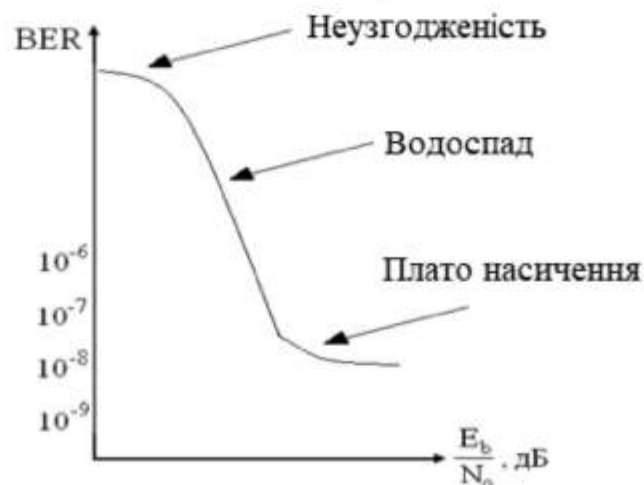


Рисунок 1.5 – Узагальнена характеристика BER для турбо-подібних кодів

Турбо-подібні коди поділяються на згорткові турбо-коди і блокові турбо-добутки (композиційний код). При високих імовірностях виникнення помилки $BER = 10^{-5} - 10^{-7}$ (мале відношення сигнал/шум, від 0 до 3 дБ) краще показує себе перший код. Його недоліками є відносно висока складність практичної реалізації, а також наявність явної області насичення ймовірності помилки.

Турбо-добутки мають значно більшу кодову відстань, а область насичення ймовірності помилки для них буде набагато нижчою, ніж для згорткового турбо-коду. Саме тому даний код є оптимальним вибором при низьких $BER = 10^{-8} - 10^{-12}$ (велике відношення сигнал/шум), оскільки він може забезпечувати дуже малі ймовірності помилки декодування при високій швидкості передавання даних для малих та середніх пакетів [3].

Турбо-подібні коди застосовуються в цифровому зв'язку (цифрове телебачення, супутниковий та мобільний зв'язок, програмовані радіосистеми тощо), ефективно вирішуючи задачу досягнення максимальної ефективності передавання даних у зашумленому каналі при обмеженій смузі частот. Турбо-коди з імовірнісними алгоритмами декодування дозволяють здійснити високоефективне передавання даних у різних стандартах зв'язку: телебаченні (DVB-RCS, DVB-RCT, DVB-SSP), бездротових локальних мережах (WiMAX), телеметрії, мобільних (3G, 4G LTE, Inmarsat) і космічних (CCSDS) комунікаціях. Наприклад, дослідження NASA JPL дозволили отримати гігабайти даних, включаючи високоякісні знімки та відеофайли поверхні Марсу за допомогою марсохода Curiosity та згорткового турбо-кода [6].

Важливою особливістю турбо-кодів є те, що вони дозволяють при розумних обчислювальних витратах наблизитися до теоретичної межі

завадостійкості Шеннона.

1.4 Принципи декодування кодів турбо-добутків

Блокові турбо-коди з імовірнісними алгоритмами декодування дозволяють здійснити високоефективне передавання даних у різних стандартах зв'язку: телебаченні, бездротових локальних мережах (WiMAX), телеметрії, SDR системах, мобільних комунікаціях, дискових масивах даних. Загалом майже всі потужні компанії, які є світовими експертами в області комунікацій, використовують саме турбо-добутки, згорткові турбо-коди та код з малою щільністю перевірок на парність (МЦПП), що показано у таблиці 1.1 [3].

Таблиця 1.1 – Використання кодів у сучасних стандартах зв'язку

Організація	Вид коду	Стандарт зв'язку
Xilinx	турбо-добутки, турбо-коди, код з МЦПП	W-CDMA, DVB-S2, IEEE 802.16
ViaSat	турбо-добутки, код з МЦПП	DVB-S2
ComBlock	турбо-добутки	IEEE 802.16
Comtech ANA	турбо-добутки, турбо-коди, код з МЦПП	W-CDMA, CDMA2000, DVB-S2, IEEE 802.16

Це компанії, які спеціалізуються на розробці різноманітної електронної апаратури та програмного забезпечення для застосування в сфері комунікацій та передавання даних.

Наприклад, він знайшов широке застосування в мобільних системах зв'язку третього покоління (рисунок 1.6). Концепція систем 3G полягає у створенні умов для надання послуг мультимедіа, включаючи високошвидкісне передавання відео, звуку, факсимільних повідомлень і даних будь-якому абоненту за допомогою мобільного термінала (низькошвидкісний та інтерактивний обмін даними). Вартість послуг повинна бути мінімальною, мати прийнятну якість і рівень безпеки [1].

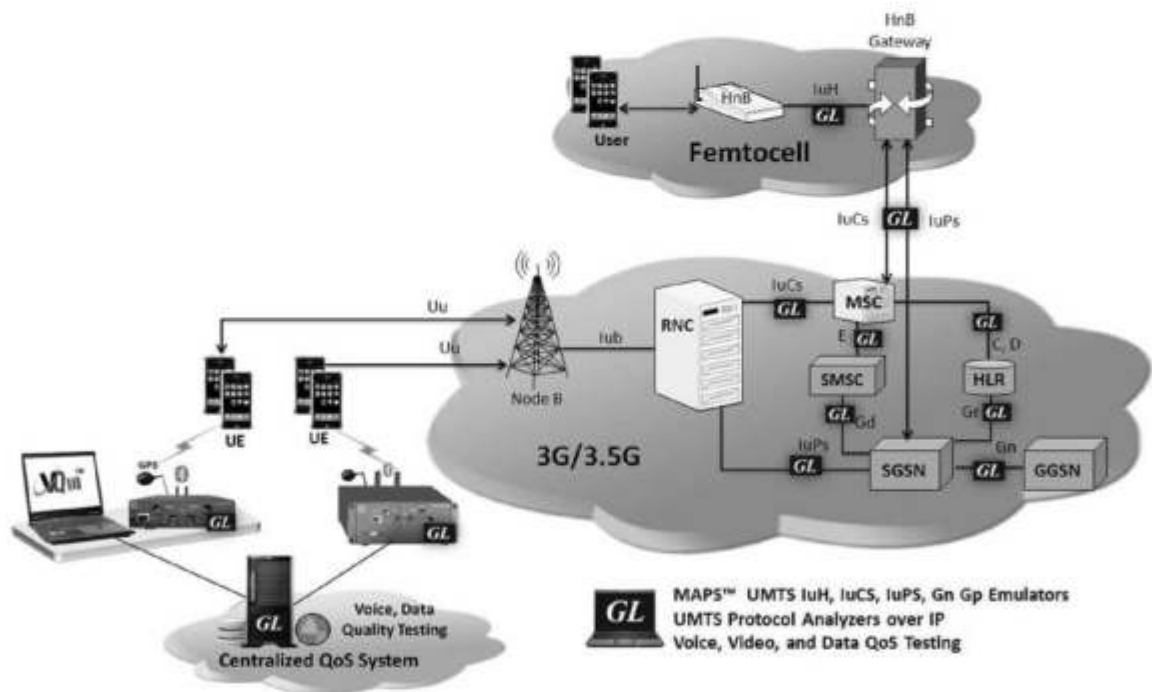


Рисунок 1.6 – Принцип роботи стандарту 3G

Архітектура таких систем включає два основні елементи: мережну інфраструктуру (Access Network), магістральні базові мережі (Core Network).

Така структура забезпечує можливість нарощування інфраструктури шляхом послідовної модифікації складових елементів. Мережі 3G працюють на частотах дециметрового діапазону, зазвичай, в діапазоні близько 2 Гц, передаючи дані зі швидкістю до 3,6 Мбіт/с. 3G включає в себе 5 стандартів сімейства IMT-2000 (UMTS/WCDMA, CDMA2000/IMT-MC, TD-CDMA/TD-SCDMA, DECT і UWC-136), із яких найбільшого поширення в світі набули UMTS (W-CDMA) і CDMA2000 (IMT-MC), в основі яких лежить одна і та ж технологія – CDMA (Code Division Multiple Access – множинний доступ з кодовим розподілом каналів).

Необхідно забезпечити визначені значення швидкості передавання для різних ступенів мобільності абонента у залежності від величини зони покриття:

- до 2048 Кбіт/с при низькій мобільності абонента і локальній зоні покриття;
- до 144 Кбіт/с при високій мобільності абонента (швидкість до 120 км/год) і широкій зоні покриття;
- до 64 (144) Кбіт/с при глобальному покритті (супутниковий зв'язок).

Кодова структура турбо-добутку вимагає використання двох ідентичних компонентних блокових кодів для кодування рядків та стовпців відповідно. Два блокових коди з'єднують послідовно, а оскільки коди незалежні і працюють по рядках і стовпцях, що забезпечує випадковість коду, то процедура переміщення бітів не потрібна.

При реалізації операції кодування вихідні дані записуються в двовимірний масив по рядках, після чого кодуються по рядках за допомогою першого коду, а потім дані і перевіірочні біти (або тільки дані) першого коду

кодуються за стовпцями другим кодом. Для передавання в канал дані з масиву зчитуються по рядках.

Найбільш складним компонентом є декодер, тому потрібно розробляти нові або модифікувати існуючі методи декодування, досягаючи компромісу між обчислювальною складністю та ефективністю декодування даних.

У ході турбо-декодування на вхід елементарних декодерів надходять “м’які” оцінки, результат декодування на виході елементарних декодерів – також “м’яке” рішення. Тому такі схеми отримали назву декодерів (рисунок 1.7) з м’яким входом і м’яким виходом (soft-input soft-output, SISO). Такі декодери ефективні, але мають високу складність реалізації. Також можна використовувати методи декодування з “жорсткими” рішеннями (hard-input hard-output), але з втратою у ефективності декодування [7].

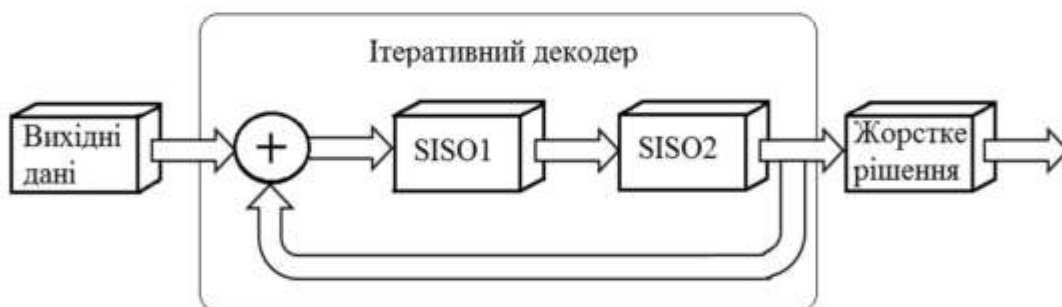


Рисунок 1.7 – Структура декодера турбо-добутку

Надійність м’яких рішень a_i визначається їх абсолютним значенням, причому чим воно далі від нуля, тим вищою буде надійність рішення:

$$LLR = |a_i|, i = \overline{1, n}. \quad (1.20)$$

Відповідно знак дійсного числа a_i , що представляє м'яке рішення, показує, яке жорстке рішення отримане:

$$D_i = \text{sign}(a_i), i = \overline{1, n}. \quad (1.21)$$

Зупинка ітеративного процесу декодування відбувається або після виконання заданої кількості ітерацій, або після того, як значення результату декодування за певним критерієм зупинки досягне встановленого порогу. Для моделювання використовують ідеальний критерій "магії джина", коли досліднику відомі місця розміщення помилок [6].

Таким чином, розглянуті світові тенденції показують перевагу паралельного каскадного коду перед відомими блоковими та згортковими коригуючими кодами і, тому, застосування такої конструкції є необхідною умовою збереження цілісності інформації у комп'ютерних системах різного функціонального призначення, особливо для яких існує вплив потужних завад.

1.5 Висновки

У даному розділі проаналізовано методи завадостійкого кодування, які використовуються в сучасних системах передавання даних. Найбільшого поширення отримали блокові та згорткові коди, а також каскадні конструкції, які мають ряд важливих переваг. Розглянуто їх основні поняття та ключові особливості кодів, виконано постановку задачі оптимізації.

Визначено, що при великому відношенні сигнал/шум застосовують блокові коди турбо-добутки, які на високій швидкості передавання даних для малих пакетів дозволяють отримати малу частоту виникнення помилок декодування. Подальше дослідження таких кодів буде проведено у наступних розділах для розроблених математичної та імітаційної моделей у програмній реалізації.

2 РОЗРОБКА МОДИФІКОВАНОГО МАТЕМАТИЧНОГО АПАРАТУ ДЛЯ ДЕКОДУВАННЯ БЛОКОВИХ КОДІВ ТУРБО-ДОБУТКІВ

2.1 Дослідження об'єкта автоматизації

Передпроектне дослідження об'єктів автоматизації представляє собою один із найважливіших видів робіт, які виконують на передпроектній стадії. Під об'єктом автоматизації розуміється фрагмент реального світу, дані про який є основою для прийняття проектних рішень. Це може бути організація в цілому, її структурні підрозділи та реалізовані в них процеси.

Метою передпроектного дослідження є отримання повного уявлення про об'єкт автоматизації, включаючи з'ясування цілей його функціонування та виявлення значень основних параметрів діяльності. На основі результатів обстеження здійснюється обґрунтування потреб та техніко-економічної доцільності створення автоматизованої системи, попередня оцінка можливості її створення та формування попередніх вимог до неї [14].

У цій роботі розглядається автоматизована інформаційна система, призначена для обробки інформації. Такі системи можуть використовуватися в різних сферах, таких як управління, проектування та виробництво. Проектування системи охоплює розробку структури, алгоритмів обробки інформації та загального алгоритму її функціонування. Технічні засоби, що застосовуються для автоматизації управлінських процесів, здатні збирати, обробляти та трансформувати інформацію про стан процесу, передавати дані через канали зв'язку, формувати управлінські команди та забезпечувати взаємодію з оператором.

Тоді потрібно виконати дослідження процесу обробки даних, організації роботи з ними та розглянути об'єкт автоматизації, такий як кодек завадостійкого коду. Для цього необхідно описати дані, які надходять на вхід цього об'єкта, а також отриманий результат. Для передавання інформації від джерела до отримувача використовується структурна схема, яка наведена на рисунку 2.1 [1].

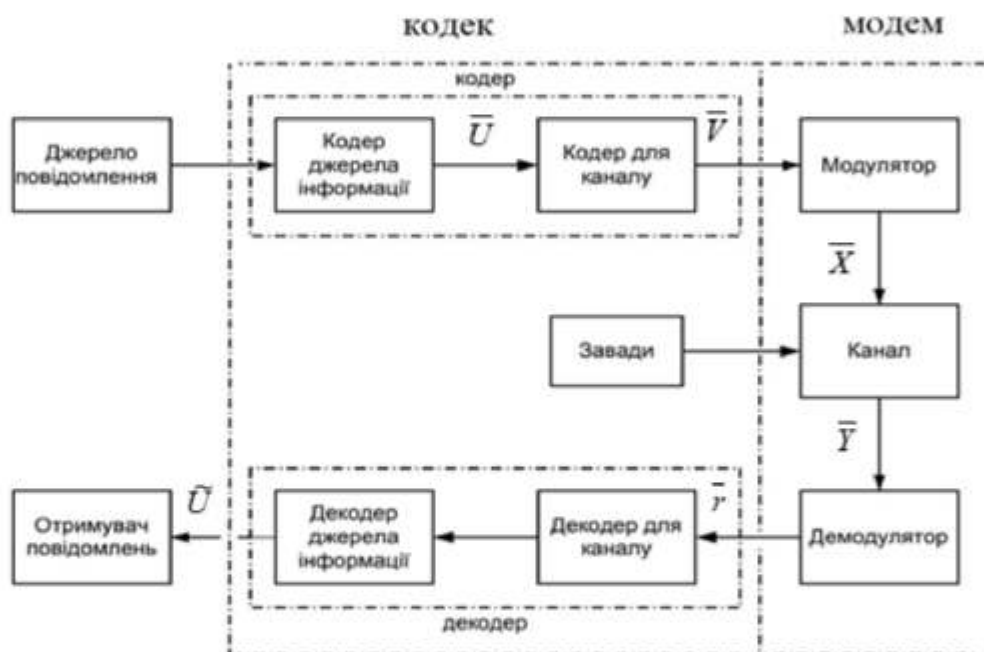


Рисунок 2.1 – Узагальнена структура для роботи з інформацією:

$\bar{U}$ – початкові дані, які надаються кодеру; $\bar{V}$ – дані після кодування;

$\bar{X}$ – дані після модуляції, які надходять до каналу; $\bar{Y}$ – дані на виході каналу;

$\bar{r}$ – дані після демодуляції, які надходять до декодера;

$\tilde{U}$ – отримана інформація після проходження вузлів системи

Основна задача системи передавання даних – обмін даними із заданими вірогідністю та швидкістю передавання, причому ці вимоги перебувають у

протиріччі, тобто підвищення швидкості передавання призводить до зниження завадостійкості та вірогідності передавання даних.

Кодер системи отримує інформаційні символи від джерела і додає до них надлишкові символи таким чином, щоб могла бути виправлена значна частина помилок, що виникають під час передавання даних каналом з шумом. Закодовані символи надходять на модулятор, який перетворює їх в аналоговий сигнал.

Демодулятор навпаки перетворює прийнятий сигнал в послідовність чисел, які представляють оцінку переданих даних (метрики). Вони надходять у декодер, який виправляє помилки, приймає рішення про символ, використовуючи внесену кодером надлишковість, та передає його до приймача.

Процес двійкової фазової модуляції (binary phase shift keying, BPSK) описується формулою (2.1), а демодуляції – рівнянням (2.2), де E_c – енергія кодованого символу [2]:

$$\overline{X_i} = (-1)^{\overline{V_i}+1} \cdot \sqrt{E_c}, \quad (2.1)$$

$$\overline{r_i} = \begin{cases} -\sqrt{E_c} \Rightarrow \overline{V_i} = 0; \\ \sqrt{E_c} \Rightarrow \overline{V_i} = 1. \end{cases} \quad (2.2)$$

Дійсні величини шуму ξ з каналу зв'язку додаються до символів з модулятора, формуючи зашумлений сигнал, у формі:

$$\overline{Y_{\xi i}} = \overline{X_i} + \xi. \quad (2.3)$$

Якщо на значення величини впливає багато випадкових факторів, то вона буде розподілена нормально. Саме тому для моделювання застосовують канал зв'язку з адитивним гаусовим білим шумом.

2.2 Алгебра логарифмічного відношення функцій правдоподібності

Алгебра логарифмічного відношення функцій правдоподібності є важливою частиною процесу декодування даних у системах цифрового зв'язку. Вона допомагає в оцінці ймовірностей для прийняття рішень у ході процесу декодування даних, особливо для турбо-подібних кодів з їх ітеративними обмінними алгоритмами. Усі операції виконуються над кінцевим полем $GF(2)$ з елементами $\{-1, +1\}$. Нехай $+1$ – це логічна одиниця (“одиничний” елемент), а -1 – логічний нуль (“нуль”-елемент). Операція додавання за модулем два (XOR) для роботи з цими значеннями подана в таблиці 2.1 [3, 5].

Таблиця 2.1 – Застосування операції XOR в полі $GF(2)$

$D_1 \oplus D_2$	$D_2 = -1$	$D_2 = +1$
$D_1 = -1$	-1	+1
$D_1 = +1$	+1	-1

Ймовірність суми за модулем два для двох бінарних інформаційних символів або статистично незалежних інформаційних бітів D_k за теоремою повної ймовірності може бути знайдена за формулою:

$$\begin{aligned}
 p(D_1 \oplus D_2 = +1) &= p(D_1 = +1) \cdot p(D_2 = -1) + p(D_1 = -1) \cdot p(D_2 = +1) = \\
 &= p(D_1 = +1) \cdot (1 - p(D_2 = +1)) + (1 - p(D_1 = +1)) \cdot p(D_2 = +1) = \\
 &= p(D_1 = +1) - 2 \cdot p(D_1 = +1) \cdot p(D_2 = +1) + p(D_2 = +1).
 \end{aligned} \quad (2.4)$$

Використаємо логарифмічне відношення функцій правдоподібності та врахуємо умову нормування ймовірностей, тоді отримаємо вираз:

$$LLR(D_k) = \ln \left(\frac{p(D_k = +1)}{1 - p(D_k = +1)} \right) = \left(\frac{1 - p(D_k = -1)}{p(D_k = -1)} \right). \quad (2.5)$$

Графічно LLR можна представити у формі кривих (рисунок 2.1).

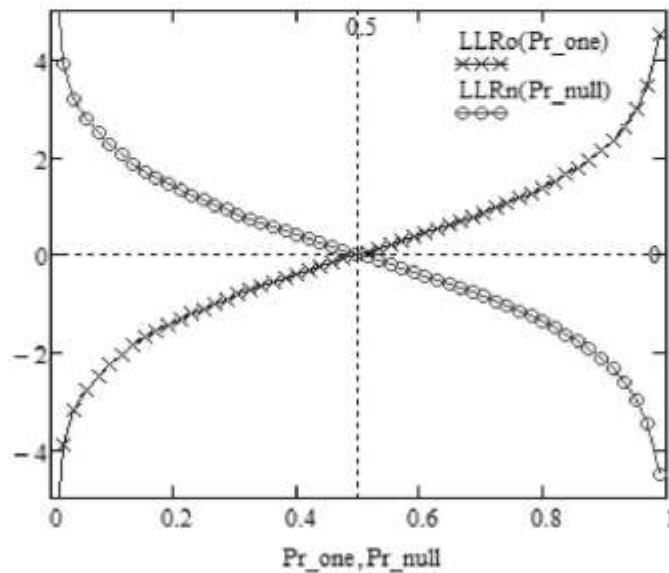


Рисунок 2.2 – Графічна інтерпретація логарифмічного відношення функцій правдоподібності

Тоді розв'язком рівняння (2.5) відносно $p(D_k = +1)$ буде вираз:

$$p(D_k = +1) = \frac{\exp(LLR(D_k))}{1 + \exp(LLR(D_k))}. \quad (2.6)$$

Підставивши рівняння (2.6) до (2.4), можна отримати формулу:

$$p(D_1 \oplus D_2 = +1) = \frac{\exp(LLR(D_1)) + \exp(LLR(D_2))}{(1 + \exp(LLR(D_1))) \cdot (1 + \exp(LLR(D_2)))}. \quad (2.7)$$

Якщо використати означення повної імовірності, то для випадку, коли $D_1 \oplus D_2 = -1$, можна отримати наступний вираз:

$$\begin{aligned} p(D_1 \oplus D_2 = -1) &= 1 - p(D_1 \oplus D_2 = +1) = \\ &= \frac{1 + \exp(LLR(D_1) + LLR(D_2))}{(1 + \exp(LLR(D_1))) \cdot (1 + \exp(LLR(D_2)))}. \end{aligned} \quad (2.8)$$

Застосувавши рівняння (2.5), замінивши $D_k = D_1 \oplus D_2$, а також виконавши підстановки (2.7) та (2.8), можна отримати наступний вираз:

$$\begin{aligned} LLR(D_1 \oplus D_2) &= \ln \left(\frac{p(D_1 \oplus D_2 = +1)}{p(D_1 \oplus D_2 = -1)} \right) = \\ &= \ln \left(\frac{\exp(LLR(D_1)) + \exp(LLR(D_2))}{1 + \exp(LLR(D_1) + LLR(D_2))} \right). \end{aligned} \quad (2.9)$$

Також можна використати операцію box-plus [5]:

$$LLR(D_1)[+]LLR(D_2) = LLR(D_1 \oplus D_2). \quad (2.10)$$

Дослідивши формулу (2.10) з використанням рівняння (2.9), можна виділити три властивості:

$$LLR(D_k)[+](+\infty) = -LLR(D_k), \quad (2.11)$$

$$LLR(D_k)[+](-\infty) = LLR(D_k), \quad (2.12)$$

$$LLR(D_k)[+]0 = 0. \quad (2.13)$$

Критерій прийняття рішення за максимумом апостеріорної імовірності можна представити у вигляді критерію мінімальної імовірності помилки. Для цього використовується умовна ймовірність отримання символу D_k після прийняття зашумленої послідовності x_ξ , що можна записати у такій формулі:

$$p(D_k = +1 | x_\xi) \underset{H_2}{\overset{H_1}{>}} p(D_k = -1 | x_\xi). \quad (2.14)$$

Вибирається одна з гіпотез $H_1(D_k = +1)$, якщо апостеріорна ймовірність

$$p(D_k = +1 | x_\xi) > p(D_k = -1 | x_\xi), \quad (2.15)$$

інакше приймаємо гіпотезу $H_2(D_k = -1)$.

За допомогою логарифмування обох частин формули (2.14) м'який вихід декодера буде визначатися наступним чином [15]:

$$LLR_{MAP}(D_k) = \ln \frac{p(D_k = +1 | x_{\xi})}{p(D_k = -1 | x_{\xi})} \underset{H_2}{\overset{H_1}{>}} 0. \quad (2.16)$$

У загальному випадку на практиці для турбо-добутків застосовується наближене обчислення $LLR_{MAP}(D_k)$, де ймовірність замінюється спеціальними кореляційними величинами.

2.3 Структура турбо-кодека

Блокові турбо-коди можуть формуватися як з послідовним, так і паралельним каскадуванням декількох блокових кодів. Вважається, що блокові турбо-коди з послідовним каскадуванням ефективніші, ніж з паралельним. На рисунку 2.3 зображено структуру такого каскадного кодера, яка складається з κ_1 рядків та κ_2 стовпців, причому в κ_1 рядках знаходяться кодові слова, утворені κ_2 бітами даних та $n_2 - \kappa_2$ перевірочними бітами [7, 11, 12, 16, 17].

Фактично кожний блок бінарних даних розміром κ_1 на κ_2 кодується двома кодами – горизонтальним та вертикальним. Кожний з κ_1 рядків представляє собою кодове слово. Аналогічно у κ_2 стовпцях є кодові слова, які утворені κ_1 бітом даних та $n_1 - \kappa_1$ перевірочними бітами. Відповідно кожен стовець теж представляє собою кодове слово.

Турбо-добутки допускають ітеративне декодування, коли на кожній ітерації за допомогою простих процедур декодування аналізуються дані, які належать компонентним кодам. Ітеративна процедура декодування такого коду

є двоетапною – горизонтальне та вертикальне декодування.



Рисунок 2.3 – Структура двовимірного блокового коду турбо-добутку:

D – інформаційні біти; p_h , p_v – перевірочні біти горизонтального та вертикального кодів; p_p – блок перевірок від перевірочних бітів

Ітеративний декодер двовимірного турбо-добутку представляє собою послідовне з'єднання двох таких елементарних декодерів, які виконують декодування даних у протилежному до кодування порядку (рисунок 2.4). Фактично інформація обертається петлею між компонентними декодерами рядків та стовпців такого коду. Подібні декодери ефективні, але мають високу складність реалізації [7].

Для декодування турбо-добутків можна застосовувати методи з “жорсткими” рішеннями [2, 4]:

- а) Редді-Робінсона;
- б) метод інверсії бінарного символу.

Більш ефективно застосовувати методи декодування, які базуються на SISO декодерах. Виділяють дві групи: перша – мінімізує ймовірність помилки

бінарного символу як максимум апостеріорної імовірності, а друга – мінімізує ймовірність помилки бінарної інформаційної послідовності. Загалом можна застосовувати такі методи декодування [2, 4]:

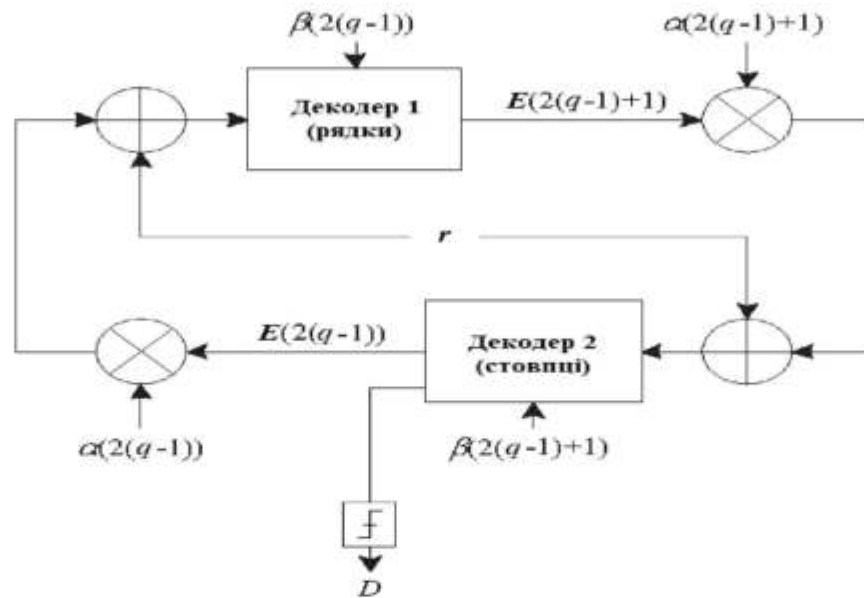


Рисунок 2.4 – Структура декодера турбо-добутку:

α, β – масштабуючі множники; q – номер ітерації декодування;

D – інформаційні біти; E – інформація з декодера (зовнішня інформація);

$\bar{r}$ – дані після проходження каналу, які надходять до декодера

а) спискове декодування:

- розширене спискове декодування з процедурою Чейза другого типу;
- метод Фанга-Баттайла з упорядкуванням статистик.

б) Хартмана-Назарова зі швидким перетворенням Адамара.

в) декодування на основі методів оптимізації багатопараметричних функцій.

г) декодування на графах Таннера:

- “сума-добутків”;
- “мінімум суми”.

д) на основі принципів Вольфа для блокових кодів на ґратці коду:

- MAP (за максимумом апостеріорної імовірності);
- log-MAP;
- модифікації типу PL-log-MAP;
- max-log-MAP;
- SOVA (метод декодування Вітербі з м’яким виходом);
- Bi-SOVA (двонаправлений SOVA).

Загальний принцип декодування зображено на рисунку 2.5.

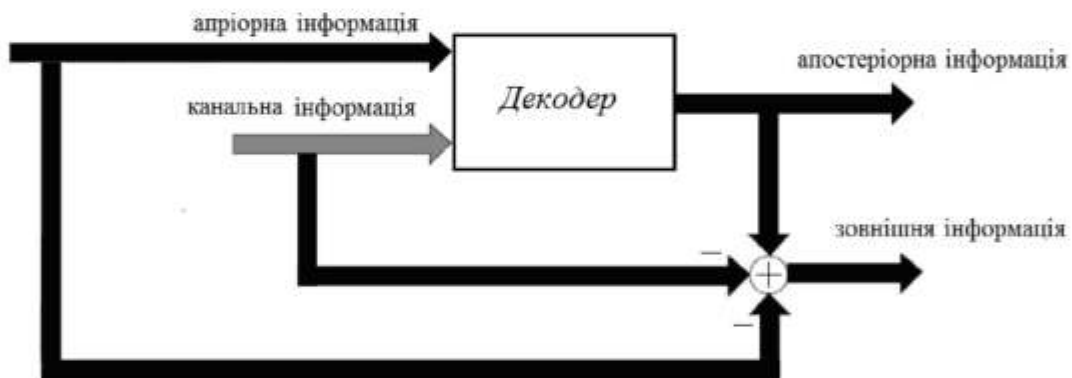


Рисунок 2.5 – Принцип декодування

Слід зазначити, що кількість ітерацій декодування кодів турбо-добутків спадає зі зростанням відношення сигнал / шум (E_b/N_0). Можуть виникати ситуації, коли помилки у даних залишаються невиявленими. Для зменшення кількості таких ситуацій необхідно застосовувати додаткові заходи, наприклад, використання каскадування з циклічним надлишковим кодом.

2.4 Математичний апарат турбо-декодера

Одним із ефективних і найбільш вживаних на практиці методів ітеративного декодування блокових турбо-кодів є спискове декодування. Основна ідея полягає в тому, що замість одного кодового слова алгоритм повертає список із можливих варіантів, один з яких є вірним. Розглянемо обчислення $LLR_{MAP}(D_k)$, в якому проводиться перебір гіпотетичних кодових слів та їх метрик. При цьому можна забезпечити майже оптимальне декодування.

Метод використовує список найімовірніших послідовностей помилок, який формується на основі оцінок надійності прийнятих символів. Для кожної комбінації помилок, що додається до кодового слова, використовується декодер з жорстким рішенням. Для декодованих послідовностей розраховується м'яка метрика. Рішенням є найбільш ймовірне кодове слово, тобто послідовність символів з найкращою (мінімальною) метрикою.

У роботі [7] розроблено математичний апарат та декодер турбо-добутків, який дозволяє оцінити надійність кожного бінарного символу після декодування Чейза [18], а також здійснити обмінний ітеративний процес.

Алгоритм методу включає наступні кроки (рисунки 2.6) [7]:

Крок 1. Нехай $r = (r_1, \dots, r_n)$ буде кодовим словом після передавання каналом зв'язку. Необхідно вибрати t позицій P_k у бінарній послідовності, які містять найменш надійні символи (тобто ті t позицій j , для яких значення r_j є найменшими за абсолютним значенням).

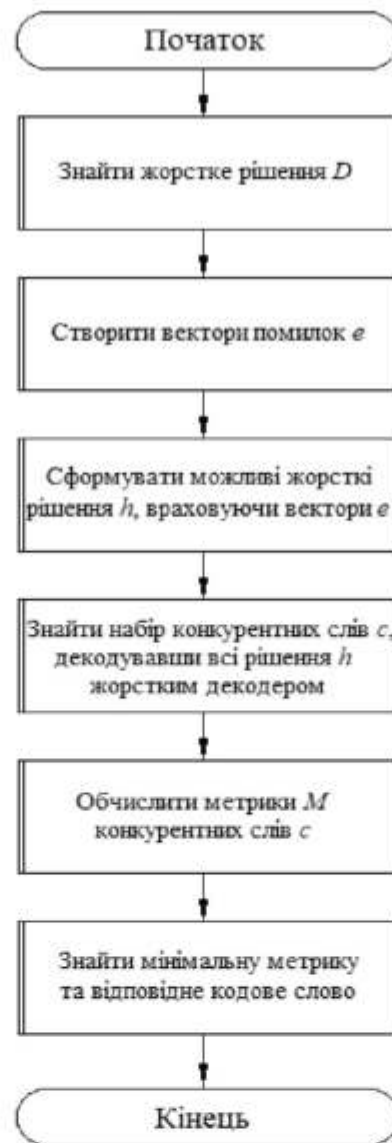


Рисунок 2.6 – Схема програми для перебору слів

Крок 2. Створити вектор можливих рішень $h_0 = (h_{01}, \dots, h_{0n})$ таких, що $h_{0j} = 1$, якщо $r_i > 0$, інакше – $h_{0j} = 0$. Створити вектори $h_1, \dots, h_{2^l-1}$ такі, що:

$$h_{ij} = h_{0j}, \quad \text{якщо } j \notin \{P_k\}, \quad (2.17)$$

$$h_{ip_k} = h_{0p_k} \oplus \text{Num}(i, k) = h_{0p_k} \oplus e, \quad (2.18)$$

де $e = \text{Num}(i, k)$ показує k -ий біт бінарного представлення числа i , представляючи вектор помилок.

Крок 3. Декодувати слова $h_0, \dots, h_{2^i-1}$ за допомогою жорсткого декодера.

Таким чином, можна одержати конкурентні кодові слова $c_0, \dots, c_{2^i-1}$.

Крок 4. Обчислити метрики конкурентних слів M_i за допомогою лінійної метрики:

$$M_i = \sum_{j=1}^n r_j (1 - 2c_{i,j}), \quad (2.19)$$

або квадрату евклідової відстані:

$$M_i = \sum_{j=1}^n (r_j + (1 - 2c_{i,j}))^2. \quad (2.20)$$

Крок 5. Визначити найбільш імовірне кодове слово c_{pp} за індексом pp :

$$M_{pp} = \min\{M_i\}. \quad (2.21)$$

Кодове слово c_{pp} є найбільш імовірним кодовим словом.

Розширене спискове декодування включає такі кроки [7, 11, 17-21]:

Крок 1. Для кожного біта j в послідовності розрахувати надійність символу (м'які рішення) за формулою

$$LLR(D_j) = \frac{1}{4} \cdot (\min\{M_i, c_{i,j} \neq c_{pp,j}\} - M_{pp}) \cdot (2c_{pp,j} - 1). \quad (2.22)$$

Якщо немає конкурентних слів, для яких j -й біт не відрізняється від $c_{pp,j}$, то надійність LLR_j має фіксоване значення:

$$LLR(D_j) = \beta \cdot (2c_{pp,j} - 1). \quad (2.23)$$

де $\beta = [0.2; 0.4; 0.6; 0.8; 1.0; 1.0; 1.0; 1.0; \dots]$, тобто приймає фіксоване значення залежно від номера ітерації декодування q .

Крок 2. Обчислити значення зовнішньої інформації E з декодера для кожного біта j за формулою:

$$E(D_j) = LLR(D_j) - r_j. \quad (2.24)$$

Крок 3. Оновити інформацію, яка входить в декодер, на ітерації q за виразом:

$$w_j^q = r_j + \alpha \cdot E(D_j^q). \quad (2.25)$$

Крок 4. Виконати ітеративний процес декодування, який схематично зображено на рисунку 2.4, на якому декодери рядків та стовпчиків обмінюються зовнішньою інформацією про біти, причому $\alpha = [0.0; 0.2; 0.3; 0.5; 0.7; 0.9; 1.0; 1.0; \dots]$ – множник, який зменшує вплив зовнішньої інформації з декодера.

Крок 5. Використати критерій зупинки процесу декодування (час, стагнація обчислень, гранична кількість ітерацій) та знайти отриману інформаційну послідовність бінарних символів ("жорсткі" рішення).

Нехай маємо послідовність символів після кодування за допомогою коду Хеммінга (спочатку по стовпцях, а потім по рядках), *BPSK*-модуляції та проходження через канал з шумом, яка записана у матрицю розміром N на N . Декодувати код турбо-добуток, використавши метод спискового декодування.

Розглянемо перший рядок: $r = (0,5; 0,7; -0,9; 0,2; -0,3; 0,1; 0,6)$. Нехай кількість інвертованих позицій t дорівнює 3 ($t = 3$), кількість ітерацій $q = 1$, множники $\alpha_1 = 0, \beta_1 = 0,2$. Вибираємо найменш надійні за модулем символи $P_1 = 6, P_2 = 4, P_3 = 5$. Тепер потрібно знайти жорстке рішення $D = (1; 1; 0; 1; 0; 1; 1)$. Формуємо вектори помилок e , які складаються з трьох символів. Формуємо вектор "жорстких" можливих рішень (таблиці 2.2 та 2.3), враховуючи вектор помилок e . Підкреслені біти відповідають інверсіям, які виникли після застосування перебору слів.

Таблиця 2.2 – Генерація векторів помилок

i	e	i	e
0	0; 0; 0	4	1; 0; 0
1	0; 0; 1	5	1; 0; 1
2	0; 1; 0	6	1; 1; 0
3	0; 1; 1	7	1; 1; 1

Таблиця 2.3 – Результат після обробки векторів помилок

i	h_i
0	(1; 1; 0; 1; 0; 1; 1)
1	(1; 1; 0; 1; 0; <u>0</u> ; 1)
2	(1; 1; 0; 1; <u>1</u> ; 1; 1)
3	(1; 1; 0; 1; <u>1</u> ; <u>0</u> ; 1)
4	(1; 1; 0; <u>0</u> ; 0; 1; 1)
5	(1; 1; 0; <u>0</u> ; 0; <u>0</u> ; 1)
6	(1; 1; 0; <u>0</u> ; <u>1</u> ; 1; 1)
7	(1; 1; 0; <u>0</u> ; <u>1</u> ; <u>0</u> ; 1)

Декодуємо код Хеммінга "жорстким" декодером (таблиця 2.4). Біти із символом зірки відповідають позиціям, скоригованим декодером у слові h_i .

Таблиця 2.4 – Декодування коду Хеммінга

i	c_i
0	(1; 1; 0; 1; 0; 1; 0*)
1	(1; 1; 0; 0*; 0; 0; 1)
2	(1; 1; 1*; 1; 1; 1; 1)
3	(0*; 1; 0; 1; 1; 0; 1)
4	(1; 1; 0; 0; 0; 0*; 1)
5	(1; 1; 0; 0; 0; 0; 1)
6	(1; 0*; 0; 0; 1; 1; 1)
7	(1; 1; 0; 0; 0*; 0; 1)

Обчислюємо метрики конкурентних слів, використовуючи формулу (2.19).

$$0,5 \cdot (-1) + 0,7 \cdot (-1) + (-0,9) \cdot 1 + 0,2 \cdot (-1) + \\ + (-0,3) \cdot 1 + 0,1 \cdot (-1) + 0,6 \cdot 1 = -2,1,$$

$$0,5 \cdot (-1) + 0,7 \cdot (-1) + (-0,9) \cdot 1 + 0,2 \cdot 1 + \\ + (-0,3) \cdot 1 + 0,1 \cdot 1 + 0,6 \cdot (-1) = -2,7,$$

$$0,5 \cdot (-1) + 0,7 \cdot (-1) + (-0,9) \cdot (-1) + 0,2 \cdot (-1) + \\ + (-0,3) \cdot (-1) + 0,1 \cdot (-1) + 0,6 \cdot (-1) = -0,9,$$

$$0,5 \cdot 1 + 0,7 \cdot (-1) + (-0,9) \cdot 1 + 0,2 \cdot (-1) + \\ + (-0,3) \cdot (-1) + 0,1 \cdot 1 + 0,6 \cdot (-1) = -1,5,$$

$$0,5 \cdot (-1) + 0,7 \cdot (-1) + (-0,9) \cdot 1 + 0,2 \cdot 1 + \\ + (-0,3) \cdot 1 + 0,1 \cdot 1 + 0,6 \cdot (-1) = -2,7,$$

$$0,5 \cdot (-1) + 0,7 \cdot (-1) + (-0,9) \cdot 1 + 0,2 \cdot 1 + \\ + (-0,3) \cdot 1 + 0,1 \cdot 1 + 0,6 \cdot (-1) = -2,7,$$

$$0,5 \cdot (-1) + 0,7 \cdot 1 + (-0,9) \cdot 1 + 0,2 \cdot 1 + \\ + (-0,3) \cdot (-1) + 0,1 \cdot (-1) + 0,6 \cdot (-1) = -0,9,$$

$$0,5 \cdot (-1) + 0,7 \cdot (-1) + (-0,9) \cdot 1 + 0,2 \cdot 1 + \\ + (-0,3) \cdot 1 + 0,1 \cdot 1 + 0,6 \cdot (-1) = -2,7.$$

Тоді маємо, що індекс:

$$pp = 1,$$

$$M_{pp} = -2,7,$$

а відповідне декодоване слово

$$D = (1; 1; 0; 0; 0; 0; 1),$$

причому його можна зустріти декілька разів у списку конкурентних слів.

Розраховуємо м'які рішення з виходу декодера для кожного біта j послідовності D (таблиця 2.5), використовуючи рівняння Піндайя за формулою (2.22).

Таблиця 2.5 – Результуючі м'які рішення та їх жорсткий еквівалент

j	$\lambda(D_j)$	D_j
1	$1 \cdot (-1,5 - (-2,7)) / 4 = 0,30$	1
2	$1 \cdot (-0,9 - (-2,7)) / 4 = 0,45$	1
3	$(-1) \cdot (-0,9 - (-2,7)) / 4 = -0,45$	0
4	$(-1) \cdot (-2,1 - (-2,7)) / 4 = -0,15$	0
5	$(-1) \cdot (-1,5 - (-2,7)) / 4 = -0,30$	0
6	$(-1) \cdot (-2,1 - (-2,7)) / 4 = -0,15$	0
7	$1 \cdot (-2,1 - (-2,7)) / 4 = 0,15$	1

Обчислюємо значення зовнішньої інформації з декодера для кожного біта j (таблиця 2.6). Збільшуємо лічильник ітерацій $q = 2$ та оновлюємо показники $\alpha_2 = 0,2, \beta_2 = 0,4$.

Таблиця 2.6 – Зовнішня інформація з декодера

j	$E(D_j)$
1	$0,30 - 0,5 = -0,20$
2	$0,45 - 0,7 = -0,25$
3	$-0,45 - (-0,9) = 0,45$
4	$-0,15 - 0,2 = -0,35$
5	$-0,30 - (-0,3) = 0,00$
6	$-0,15 - 0,1 = -0,25$
7	$0,15 - 0,6 = -0,45$

Переходимо до наступної ітерації. Для цього обчислимо оновлені значення отриманої послідовності, яка входить в декодер (таблиця 2.7).

Повторити дану процедуру для всіх рядків. Виконати декодування стовпців, використавши значення w , отримане після декодування рядків. Потім знову перейти до рядків. Отже, інформація обертається петлею між двома компонентними декодерами, що зображено на рисунку 2.4. Виконати низку таких ітерацій та розрахувати остаточні "жорсткі" рішення для символів D_j .

Зупинити процес можна за одним із критеріїв [22-24]:

1. Досягнення максимальної кількості ітерацій, що можна застосувати тоді, коли декодування займає забагато часу.

2. Зупинка на основі стабілізації вихідної інформації, тобто розраховані значення не змінюються впродовж певного часового проміжку, або відстань між ними дуже мала (менша порогового значення). Такий підхід може бути

ефективним, коли відомо, що сигнал є досить стійким до помилок у каналі зв'язку.

Таблиця 2.7 – Оновлення стартової послідовності

j	w
1	$0,5 + 0,2 \cdot (-0,2) = 0,46$
2	$0,7 + 0,2 \cdot (-0,25) = 0,65$
3	$-0,9 + 0,2 \cdot 0,45 = -0,81$
4	$0,2 + 0,2 \cdot (-0,35) = 0,13$
5	$-0,3 + 0,2 \cdot 0 = -0,3$
6	$0,1 + 0,2 \cdot (-0,25) = 0,05$
7	$0,6 + 0,2 \cdot (-0,45) = 0,51$

3. Зупинка на основі порогового значення помилок, коли канал протестований, тобто відомий середній рівень помилок у ньому при передаванні даних. Тоді, коли кількість помилок декодування стає меншою за певне порогове значення, процес можна зупинити.

2.5 Запропонована модифікація для декодера

Для формування набору найбільш ймовірно прийнятих кодових слів використовується той факт, що помилки найімовірніше знаходяться на найменш надійних прийнятих позиціях. Тому різниця у алгоритмах перебору

полягає в кількості позицій, що вибираються для аналізу, від яких залежить кількість генерованих кодових послідовностей.

Повний перебір говорить про те, що потрібно взяти всі символи і дослідити всі можливі комбінації слів, що на практиці не має змісту, оскільки наявні зайві обчислення та великі часові затрати. Другий підхід – це грубе наближення, наприклад, взяти половину всіх прийнятих символів, частина з яких буде з певною ймовірністю достовірно прийнятими. Ще один підхід полягає в тому, щоб на основі ключових параметрів коду шукати певні залежності та задавати евристичні оцінки кількості вибраних ненадійних символів. Тобто виникає складна оптимізаційна задача пошуку такої кількості ненадійних символів, яка після перебору та аналізу кодових слів дозволить отримати найменшу кількість помилок при передаванні даних для заданого відношення сигнал / шум:

$$BER = f(Eb/N0, t) \rightarrow \min . \quad (2.26)$$

У нашому випадку оптимізаційна задача розв'язується наступним чином. Спочатку алгоритм модифікується так, щоб адаптувати його до якості каналу зв'язку. У компонентному методі декодування Чейза [18, 25, 26] застосовано нормалізацію отриманих зашумлених значень бінарних символів у межах діапазону [0, 1]:

$$X'_{j\text{ норм}} = \frac{|r_j| - |\min|}{|\max| - |\min|}, \quad (2.27)$$

де X' – початкове зашумлене значення символу;

$\min$ – мінімальне значення в наборі даних;

$\max$ – максимальне значення в наборі даних.

Також введено евристичний позитивний параметр, який визначає порогове значення $Q \in [0,1]$ (порог для обробки даних). Для аналізу вибираються ті ненадійні біти, які мають зашумлене значення менше за даний параметр:

$$X'_{j_{\text{норм}}} < Q. \quad (2.28)$$

Слід зазначити, що їх кількість сильно впливає на перебір комбінацій, подальше спискове декодування та відповідно складність декодера. Якщо $Q = 0$, то декодування не відбувається, оскільки дана послідовність символів вважається надійною. У випадку $Q = 1$ всі символи, крім того, який має максимальне нормалізоване значення, ненадійні, а процес декодування буде максимально складним, що не є виправданим.

Для пошуку значення параметру Q необхідно виконати низку дослідів для вибраного значення параметра та усереднити результат, повторити такі дії зі зміною Q до вибору оптимального значення:

$$BER = f(Eb/N_0, Q) \rightarrow \min. \quad (2.29)$$

Таким чином, необхідно балансувати між значеннями t та n , враховуючи ту особливість кодів турбо-добутків, що для малої кількості інформаційних символів n , результати декодування кращі.

2.6 Висновки

Таким чином, розглянуто систему передавання цифрових даних, проаналізовано особливості її складових частин. Також проаналізовано математичний апарат розробленого кодека для роботи з кодами турбо-добутками. Визначено, що спискове декодування з м'якими рішеннями є досить ефективним. У наступному розділі необхідно виконати імітаційне моделювання та експериментально перевірити розробку.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

3.1 Обґрунтування вибору середовища розробки

Середовище MatLab містить високорівневу інтерпретовану мову програмування та набір прикладних програм з інтегрованим середовищем, призначений для виконання інженерних та математичних розрахунків, роботи з матричними даними та візуалізації. Його використання дозволяє зменшити час, необхідний для розв'язання різноманітних наукових задач, а також спростити розробку нових алгоритмів у таких сферах, як моделювання об'єктів і систем управління, проєктування комунікаційних систем, обробка сигналів і зображень, обчислювальна біологія, математична фізика тощо [27, 28].

Система MatLab складається з:

- а) програмного інтерфейсу користувача;
- б) мови M-language;
- в) середовища обробки даних;
- г) керованої графіки;
- д) бібліотеки математичних функцій.

Основні особливості системи MatLab (рисунок 3.1):

1. Можливість інтеграції з іншими мовами програмування, такими як C, C++, Java та інші.
2. Матрична мова програмування, тобто всі дані представлені у вигляді матриць та векторів, що дозволяє легко та швидко виконувати операції з ними.

Також реалізована підтримка багатопотокових обчислень, що дозволяє ефективно використовувати багатоядерні процесори та обчислювальні кластери.

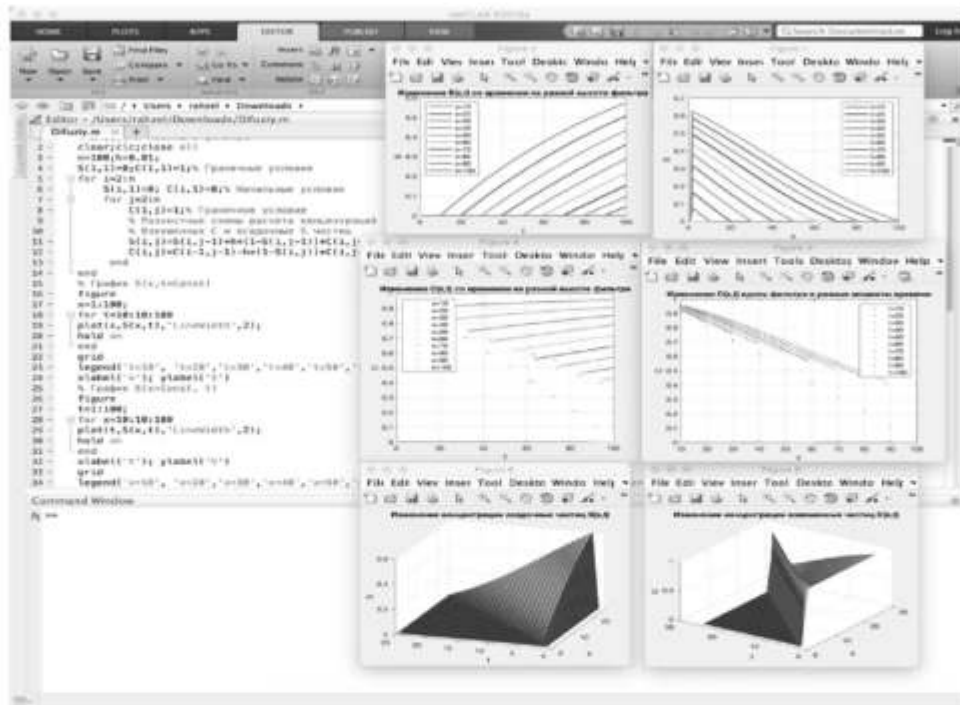


Рисунок 3.1 – Вигляд системи MatLab

3. Інтерактивне середовище розробки, що дозволяє швидко виконувати та тестувати код програми.

4. Простота, тобто простий та зрозумілий синтаксис, що дозволяє швидко вивчати мову програмування, виконувати розробку та тестування різноманітних алгоритмів.

5. Підтримка графіки, що дозволяє візуалізувати результати обчислень та досліджень.

6. Велика кількість функцій та бібліотек для чисельних обчислень, що дозволяє легко вирішувати складні задачі та робить MatLab популярним в

наукових дослідженнях та інженерних розробках, в яких зазвичай потрібно працювати з великими об'ємами даних різного типу, а також спеціалізованими алгоритмами.

Отже, MatLab може бути хорошим вибором, зокрема, завдяки своїм простоті та зручності використання, наявності широкого спектру бібліотек та фреймворків, великій та активній спільноті користувачів, високій продуктивності та можливості комбінування з іншими мовами програмування. MatLab є зручним засобом для розробки великомасштабних проєктів різного рівня складності в багатьох галузях науки та техніки.

3.2 Опис програми

Засоби перетворення даних незалежно від середовища передавання (оптичне волокно, дротові лінії, радіоканали) розробляються на основі мікропроцесорних структур. Відповідно головними їх функціями на передавальному боці є зчитування даних із носія, кодування, перетворення на послідовний код та формування модульованих сигналів. На приймальному боці сигнали демодулюються, дані конвертуються у паралельний формат, декодуються та зберігаються на носії [29].

Обмін даними в системах зв'язку може відбуватися в трьох режимах:

- а) програмному, який вимагає мінімум апаратури;
- б) програмно-апаратному, що передбачає використання режиму переривань;
- в) апаратному, що забезпечує прямий доступ до пам'яті.

Порівняння цих режимів показує, що програмний режим є найбільш простим у реалізації та налагодженні, але має найнижчу швидкість. Апаратний режим складніший, проте швидший. На практиці паралельні каскадні коди зазвичай реалізуються за допомогою програмно-апаратного (змішаного) режиму на цифрових сигнальних процесорах.

Програмно-апаратний режим пов'язаний з тим, що сам обмін інформацією здійснюється в програмному режимі, а сигнали управління цим процесом формуються апаратно. При виникненні сигналу INT, що формується зовнішнім пристроєм, центральний процесор закінчує виконання основної програми, всі необхідні дані фіксує в допоміжному запам'ятовувальному пристрої (стеку), і у відповідності з підпрограмою обробки переривання здійснює обмін даними в програмному режимі.

У випадку застосування переривань процесор звільняється від опитування пристрою зв'язку з об'єктом і може займатися іншими задачами. Якщо необхідність в обміні виникає рідко, то вигравш у швидкодії системи в цілому може бути значним. Використання режиму переривань дозволяє вивільнити процесор для виконання інших процедур (обробка даних, допоміжних математичних розрахунків, адаптації пристрою до параметрів об'єкта за певним алгоритмом тощо), але налагодження апаратної та програмної частини викликає певні складності. Програмні налагоджувачі в більшості не реагують на апаратні переривання, тобто програмні модулі доводиться налагоджувати окремо.

Висока швидкість обміну даними досягається лише в межах передавання блоку інформації, розмір якого визначається розміром буферного ОЗП. При

необхідності ж передавання більшого обсягу інформації середня швидкість передавання буде визначатись швидкодією комп'ютера в цілому і виявиться значно нижчою.

Спеціалізована цифрова обробка сигналів дозволяє суттєво збільшити ефективність побудови і експлуатації сучасних засобів передавання інформації. Часткова заміна обладнання і повна або часткова заміна програмного забезпечення може дати значний позитивний результат [29].

Це пов'язано з тим, що перехід до мікропроцесорів та програмованих логічних інтегральних схем принципово змінив підхід до розробки та експлуатації спеціалізованого обладнання цифрового оброблення сигналів. Суттєво зросла роль алгоритмічного програмного забезпечення, яке тепер практично повністю визначає ефективність засобів передавання і оброблення даних. Наявність потужних програмних пакетів дозволяє розраховувати характеристики розробленого пристрою або системи з використанням імітаційного моделювання ще до початку етапу проєктування.

Архітектура апаратної частини засобів передавання і оброблення інформації не лише наближається до архітектури універсальної мікропроцесорної системи, але й може бути орієнтована на використання базового обладнання мікропроцесорних систем, що випускається серійно. Функціональні можливості програмних пакетів моделювання дозволяють вибрати оптимальний режим роботи обладнання і транслювати налагоджене програмне забезпечення в мову програмування високого рівня (зазвичай C/C++), яке потім за допомогою спеціалізованих трансляторів може бути перетворене на асемблер практично будь-якого процесора. Все це дозволяє

отримати гнучку адаптивну апаратуру, яка під час експлуатації може переналаштовуватись під вирішення різних конкретних задач, однією з яких є обробка даних в процесі декодування.

У програмі реалізовано модуль керування та низку функціональних модулів. Модуль керування відповідає за базові функції для маніпуляції з даними та їх візуалізацією. Центральний модуль керування є ключовою компонентою системи. Він відповідає за координацію та керування роботою інших складових, виконуючи такі важливі функції, як прийняття рішень, розподіл ресурсів, збирання та аналіз даних, керування взаємодією з користувачем та іншими компонентами, ведення журналу подій.

Відповідно функціональні модулі реалізують обробники даних та візуалізатори. Модуль вводу вхідних даних відповідає за збирання та підготовку інформації, яка буде використовуватись в програмі. Його основна функція – отримання даних від користувача або з зовнішніх джерел і їх передавання до центрального модуля керування, який переправляє дані до модуля опрацювання вхідних даних для подальшої обробки. Відповідно далі можна переглянути вибрані зображення в інтерфейсі користувача.

Під час роботи з програмним забезпеченням користувач повинен ввести та вибрати необхідні для роботи вхідні дані в зазначених для цього місцях інтерфейсу програми. Після цього натиснути на кнопку-перемикач $BER = f(E_b / N_0)$ або $BER = f(q)$ у правій нижній панелі результатів для початку роботи програмних модулів. Програма починає видавати детальну статистику роботи у робочому вікні. Після закінчення процесу передавання даних можна переглянути графічні відображення залежно від вибору користувача. Виконавши всі необхідні операції, користувач може зберегти

необхідні результати, роздрукувати їх або закрити вікно програми, закінчивши роботу з нею. Концептуально інтелектуальний модуль представляє набір підпрограм, модулі яких представлено на рисунку 3.2 [30].



Рисунок 3.2 – Структура програмного забезпечення

Основні функції, які розроблені у програмі:

а) `turbo_encode(cod, ...)`, `turbo_decode(dec, ...)` – кодування коду турбо-добутку та спискове декодування, де `cod`, `dec` – компонентний метод кодування / декодування;

б) `BPSK(...)`, `de_BPSK(...)` – проведення модуляції / демодуляції;

в) `AWGN_channel(...)` – моделювання каналу зв'язку з нормально розподіленим адитивним шумом;

г) `result(...)`, `save(...)`, `print(...)` – обчислення результатів, їх збереження та друк.

3.3 Експериментальні дослідження

Проведемо дослідження ефективності системи цифрового зв'язку з кодом турбо-добутком та визначимо експериментальну функціональну залежність частоти виникнення бітових помилок (BER) від нормованого співвідношення сигнал/шум E_b/N_0 (дБ). Вхідними даними є кількість ненадійних символів, кількість ітерацій декодування, послідовності інформаційних бітів і мультимедійні дані. Результати роботи програми представлені у вигляді експериментальних графічних залежностей $BER = f(E_b/N_0)$ та декодованого зображення. Модель системи ілюструється на рисунку 3.3 [30].

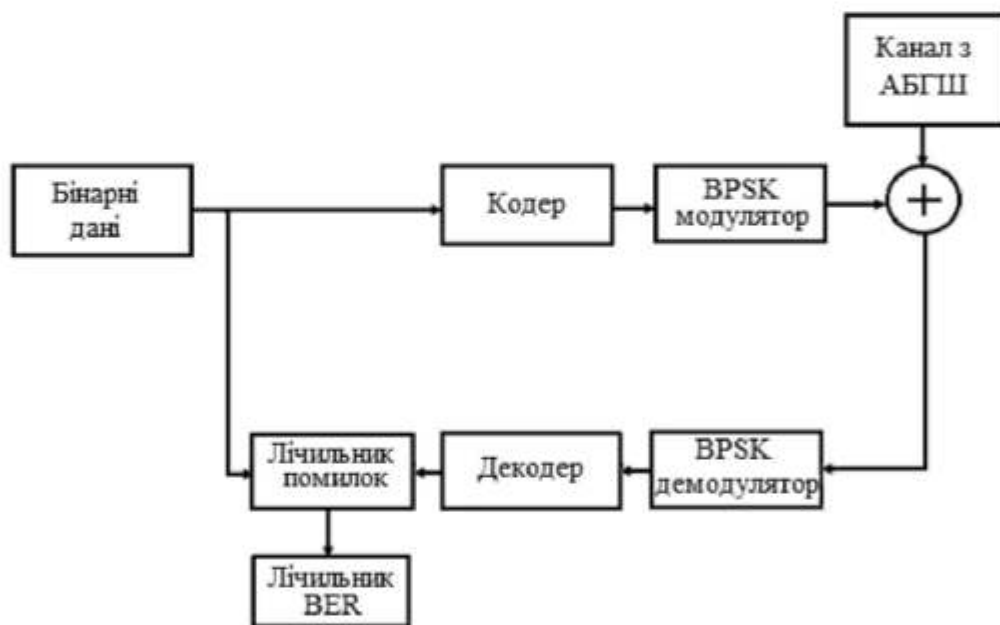


Рисунок 3.3 – Модель системи передавання даних

Відомо, що ефективність завадостійких кодів покращується завдяки збільшенню довжини кодової послідовності символів n , яка в свою чергу, залежить від кількості інформаційних символів k . Але для великих значень k

буде зростати складність декодування. Це означає, що бажано побудувати довгий код із набагато коротших кодових компонентів, які можна легко декодувати.

Для коду турбо-добутку з довжиною k отримати такий код можна шляхом об'єднання кодів Хеммінга або Боуза-Чоудхурі-Хоквінгема (БЧХ) [1]. Важливість цих кодів забезпечується не тільки гнучкістю вибору їх параметрів, але і тим, що при довжинах інформаційної послідовності в декілька сотень бітів вони є оптимальними серед відомих кодів з тими ж параметрами. Для будь-якого додатного $h \geq 3$ і $h < 2^{h-1}$ існує бінарний БЧХ-код із наступними параметрами:

$$d_{\min} \geq 2i + 1, \quad (3.1)$$

$$n = 2^h - 1, \quad (3.2)$$

$$l = n - k \leq hi, \quad (3.3)$$

де $d_{\min}$ – мінімальна кодова відстань;

n – довжина блоку;

l – кількість перевірочних бітів.

Знайдемо аналітичний вираз для оцінки складності роботи турбо-спискового декодера з урахуванням розміру матриці турбо-коду-добутку $N = s \cdot n$, кількості бінарних символів у стрічці (або стовпці) та ненадійних символів t у ній.

Для цього визначимо кількість елементарних математичних операцій (ЕМО), які необхідно виконати для декодування інформаційної послідовності:

додавання, віднімання, множення, ділення $1/a$, визначення абсолютного значення, мінімальне значення та порівняння, виключне АБО, додавання двійкових чисел. Виконання жорсткого декодування компонентного коду позначимо *Hard*, а його складність – O .

Проаналізуємо декодування одного рядка (стовпця) коду та визначимо загальну кількість ЕМО у виразах для обчислення векторів помилок e , кодових слів з помилками h , кодових слів після жорсткого декодування s , лінійних метрик M , апостеріорних м'яких рішень $LLR(D)$, зовнішньої інформації E , оновлених значень w та жорсткого рішення D . Після цього обчислимо загальну суму всіх операцій.

Для кодів турбо-добутків межі обчислювальної складності декодера $[O_{low}, O_{high}]$, які показують нижню та верхню кількості виконаних декодером ЕМО над s стрічками, можна записати у таких аналітичних виразах [26]:

$$O_{low_s}(n, s, t, O_{hard}) = s \cdot (18n - t - 1 + 2^t \cdot (2t + 3n + O_{hard})), \quad (3.4)$$

$$O_{high_s}(n, s, t, O_{hard}) = s \cdot (17n - t - 1 + 2^t \cdot (2t + 4n + O_{hard})), \quad (3.5)$$

де n – кількість символів у коді;

t – кількість ненадійних символів;

O_{hard} – кількість ЕМО для декодування складового компонентного коду.

Для декодування стовпців у даних формулах s та n міняються місцями. Загальна кількість елементарних математичних операцій, яку виконав декодер за N циклів (1 цикл = 2 ітерації) буде складати:

$$O(n, s, t, O_{hard}) = [N \cdot (O_{low_n} + O_{low_s}), N \cdot (O_{high_n} + O_{high_s})]. \quad (3.6)$$

Для випадку квадратної кодової таблиці вираз (3.6) буде таким:

$$O(n, s, t, O_{hard}) = [2N \cdot O_{low_s}, 2N \cdot O_{high_s}]. \quad (3.7)$$

Наведемо графічні залежності складності декодування турбо-коду-добутку з компонентами $(n, k, d_{min}) = (7, 4, 3)$, $(15, 11, 3)$ та $(63, 45, 7)$, кількість ненадійних бітів $t = [1, \dots, 63]$, складність жорсткого декодування константна. Результати розрахунків показано на рисунку 3.4 – 3.7, де на графіках наведено криві кількості ЕМО для одного циклу ($N = 1$), отримані за формулою (3.6) для заданих кодів, а на рисунку 3.7 представлено криві найбільшої можливої кількості виконаних операцій. Можна помітити, що зі збільшенням показника t складність декодера починає різко зростати.

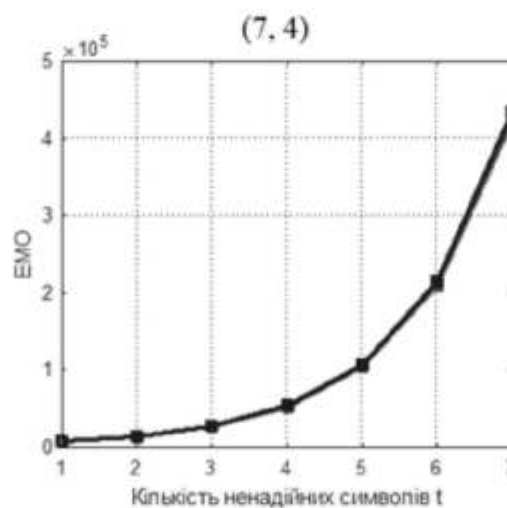


Рисунок 3.4 – Кількість ЕМО залежно від значення t для коду (7,4)

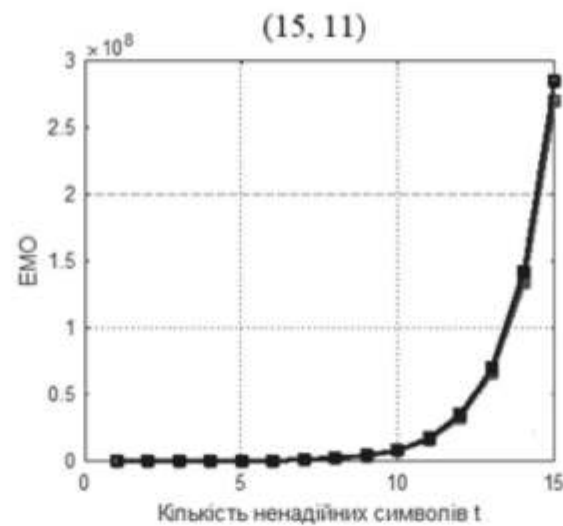


Рисунок 3.5 – Кількість ЕМО залежно від значення t для коду (15,11)

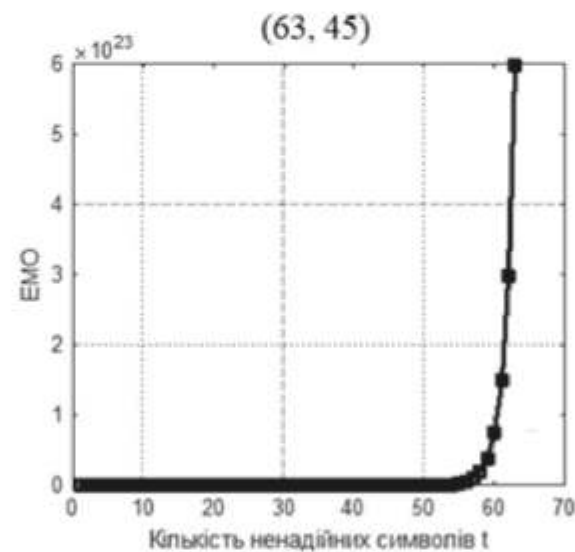


Рисунок 3.6 – Кількість ЕМО залежно від значення t для коду (63, 45)

Можна помітити, що зі збільшенням показника t складність декодера починає різко зростати. Використання більшого значення t вигідно, але вимагає використання потужніших обчислювальних ресурсів.

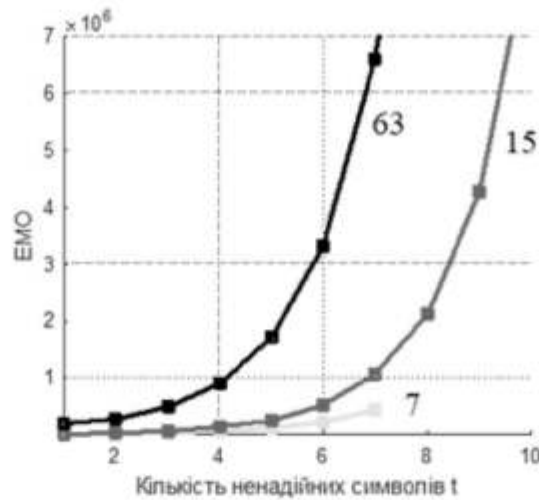


Рисунок 3.7 – Загальна кількість ЕМО
залежно від значення t

Для пошуку експериментальних функціональних залежностей $BER = f(E_b/N_0)$, які характеризують ефективність методу декодування та роботи кодека в цілому, застосуємо імітаційне моделювання, суть якого полягає в багаторазовому випробуванні системи зв'язку зі статистичним обробленням одержаних результатів [29].

Пакети даних з виходу генератора псевдовипадкової послідовності, код турбо-добуток, компонентні БЧХ-кодери з параметрами коду (n, k, d_{min}) : $(32, 26, 4)$, $(65, 57, 4)$, $(128, 120, 4)$, $(256, 247, 4)$ зі швидкостями $R = (R_{ком})^2 \approx (0,66; 0,7689; 0,8789; 0,9309)$, метод спискового декодування, експериментальний пороговий параметр $Q \approx \{0,01; 0,18; 0,24; 0,32\}$ для заданих параметрів кодів, 10 ітерацій декодування, $BPSK$ -модем, канал з адитивним білим гаусовим шумом (АБГШ). Моделювання проводилося від 0 до 5 дБ. Критерієм закінчення ітеративного процесу для кожного значення E_b/N_0 є час (не більше 40 хв). Результати експерименту показані на рисунку 3.8.

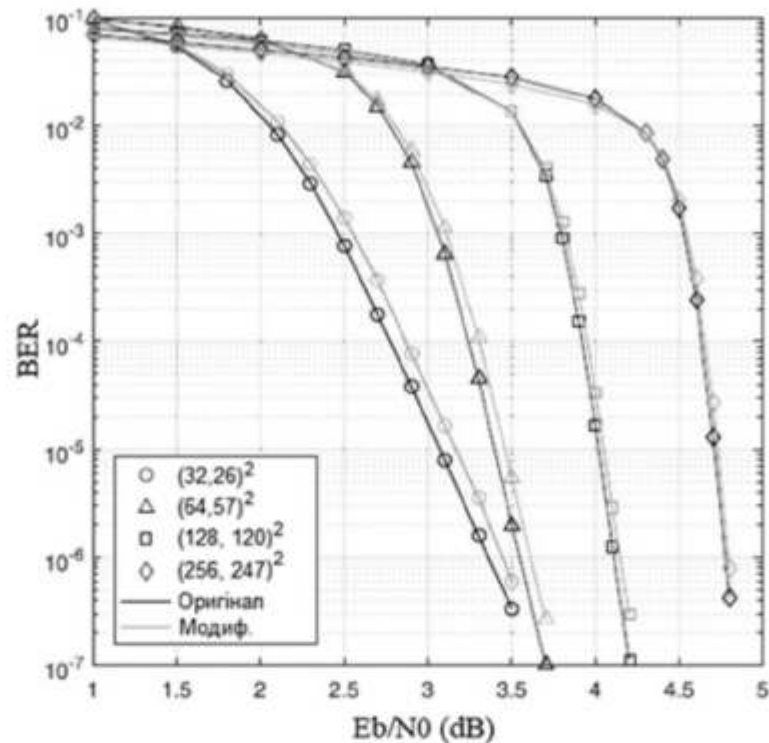


Рисунок 3.8 – Результати порівняння оригінального та модифікованого алгоритмів декодування

Отримані експериментальні залежності $BER = f(E_b/N_0)$ для системи з ітеративним декодером показують тенденцію до зменшення BER зі зростанням показника E_b/N_0 . Складність обчислень може бути меншою або дорівнювати кількості ЕМО оригінального алгоритму декодування за рахунок евристичного вибору ненадійних символів. Модифікований метод має незначно меншу ефективність за показником BER , наприклад, для $BER = 10^{-5}$:

- а) для 1-го коду – +0,094 дБ;
- б) для 2-го коду – +0,047 дБ;
- в) для 3-го коду – +0,014 дБ;
- г) для 4-го коду – +0,001 дБ.

Далі протестуємо алгоритм на мультимедійних даних. Необхідно використати кольорове зображення (256 на 256), код турбо-добуток із

параметрами $(63, 45, 7)^2$ та швидкістю $R \approx 0,51$, компонентний БЧХ-код, показник $E_b/N_0 = 4,0$ дБ, списковий декодер, пороговий параметр $Q \approx 0,15$. Результати роботи програми показані на рисунках 3.9 – 3.12 на 0 (без кодування), 2, 5 та 10 ітераціях відповідно.



Рисунок 3.9 – Зображення без кодування, BPSK-модем



Рисунок 3.10 – Зображення після 2 ітерацій декодування



Рисунок 3.11 – Зображення після 5 ітерацій декодування



Рисунок 3.12 – Зображення після 10 ітерацій декодування

Чисельні показники наведені в таблиці 3.1, де BEN / SEN – кількість бітових та символічних помилок, а BER / SER – коефіцієнт бітових та символічних помилок.

Таблиця 3.1 – Чисельні показники експерименту

E_b/N_0 , дБ	Кількість ітерацій, q	BEN	SEN	BER	SER
4,0 дБ	0	434213	123856	$1,380 \cdot 10^{-1}$	$6,299 \cdot 10^{-1}$
	2	18892	13613	$6,006 \cdot 10^{-3}$	$6,924 \cdot 10^{-2}$
	5	60	52	$1,907 \cdot 10^{-5}$	$2,645 \cdot 10^{-4}$
	10	1	1	$3.179 \cdot 10^{-7}$	$5.086 \cdot 10^{-6}$

На рисунках 3.13 та 3.14 показано криві частоти виникнення помилок залежно від кількості ітерацій декодування коду q . Можна побачити, що якість переданого зображення покращується, а кількість помилок зменшується, причому на 10 ітерації декодування проведене практично ідеальне передавання даних без помилок. Отже, розробка працює коректно.

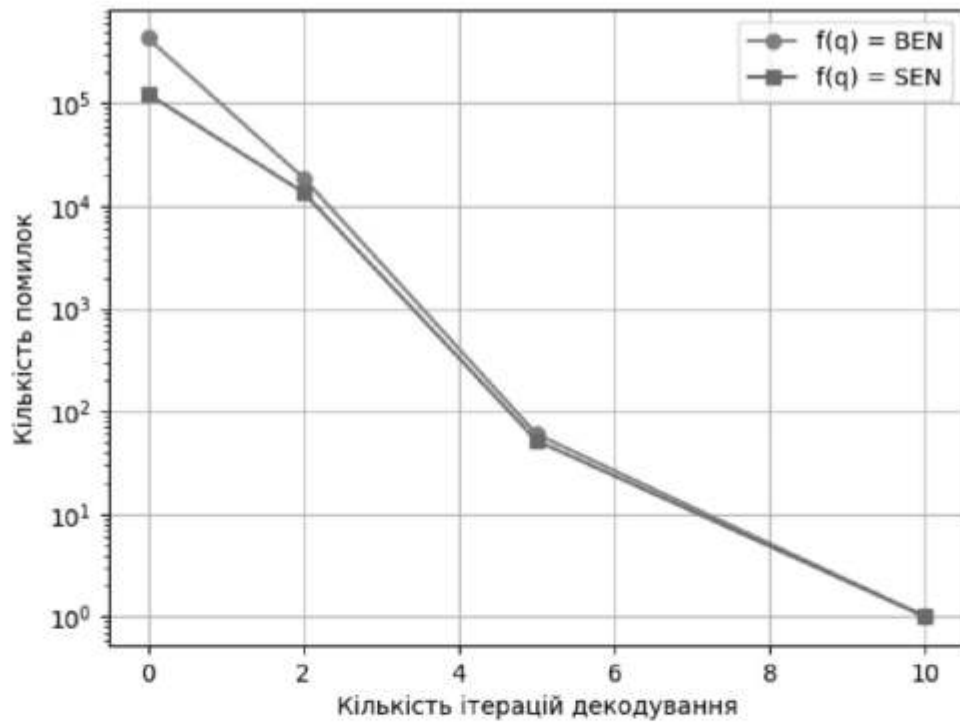


Рисунок 3.13 – Експериментальні залежності *BEN* та *SEN*

для експерименту з мультимедійними даними

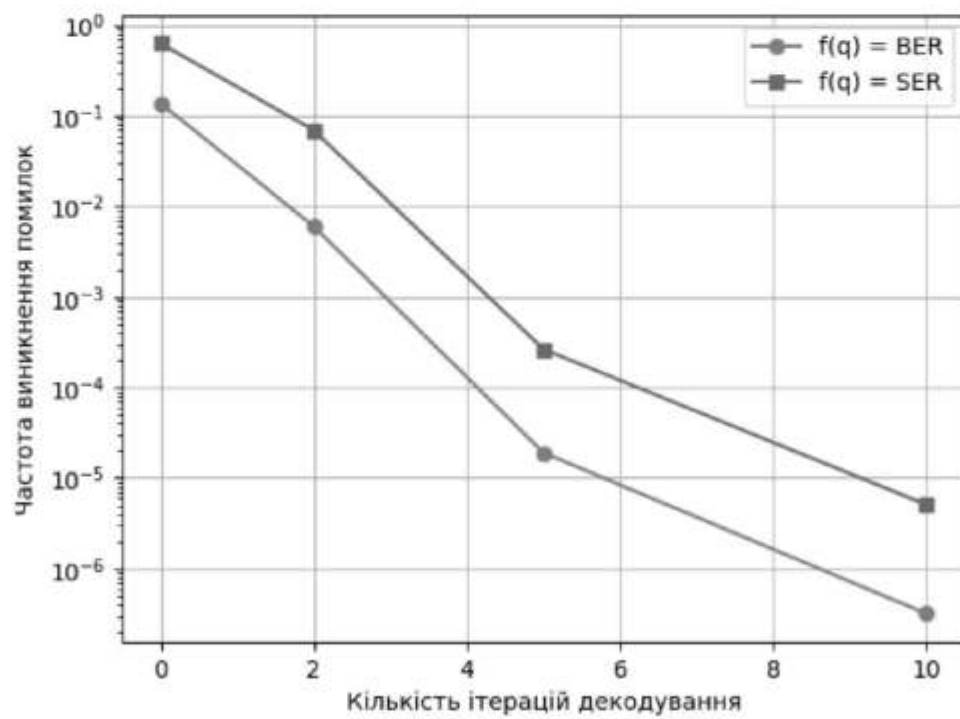


Рисунок 3.14 – Експериментальні залежності *BER* та *SER*

для експерименту з мультимедійними даними

3.4 Висновки

У розділі наведено результати експериментальних досліджень ефективності системи цифрового зв'язку з розробленим кодеком у відповідному програмному додатку, визначено експериментальну функціональну залежність частоти виникнення бітових помилок від нормованого відношення сигнал/шум (для $BER=10^{-5}$ розробка незначно поступається за енергетикою, $+0,001-0,094$ дБ). Результати мультимедійних експериментів підтверджують коректність роботи розробки

4 ЕКОНОМІЧНИЙ РОЗДІЛ

4.1 Технологічний аудит розробленої модифікації

Як було зазначено раніше, збільшення трафіку та швидкостей передавання інформації підсилило інтерес до розроблення та впровадження нових, більш ефективних методів завадостійкого кодування, одними із яких, як було доведено у нашій роботі, є так звані турбо-подібні коди. Вони дозволяють ефективно працювати з невеликими кодовими блоками на дуже високих кодових швидкостях, переважаючи низку інших завадостійких кодів.

Але при використанні таких кодів виникає проблема пошуку компромісу між вибором тих чи інших тестових символів, які застосовуються при кодуванні, та якістю (ефективністю) самого декодування, оскільки воно вимагає при цьому витрат певних ресурсів, величину яких потрібно мінімізовувати.

У результаті виконаної роботи нами було розроблено новий модифікований метод (автоматизований модуль) спискового декодування кодів турбо-добутків, який дозволяє виконувати порівняльний аналіз складності декодування без значного зниження ефективності роботи системи передавання.

Для встановлення комерційного потенціалу розробленого нами автоматизованого модуля для декодування блокових кодів турбо-добутків (у подальшому – нашої розробки) було запрошено 3-х відомих експертів – фахівців у цій галузі технічних знань: кандидатів технічних наук В.В. Гармаша, С.Г. Кривогубченка та В.М. Папінова, які здійснили оцінювання комерційного потенціалу розробки за критеріями, наведеними в таблиці 4.1 [31].

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу
будь-якої розробки і їх бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
Ринкові перспективи					
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Запрошені експерти оцінили комерційний потенціал розробленого нами автоматизованого модуля. Результати наведені у таблиці 4.2. Встановлення комерційного потенціалу розробленого автоматизованого модуля будемо здійснювати на основі рекомендацій, наведених в таблиці 4.3 [31].

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Ініціали, прізвища експертів		
	В.В. Гармаш	С.Г. Кривогубченко	В.М. Папінов
	Бали, що їх виставили експерти:		
1	4	4	3
2	4	3	4
3	4	3	4
4	4	4	4
5	4	4	3
6	4	4	4
7	3	4	4
8	4	3	3
9	4	4	3
10	4	4	4
11	4	4	4
12	3	3	3
Сума балів	СБ ₁ = 46	СБ ₂ = 44	СБ ₃ = 43
Середньо-арифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{3} = \frac{46 + 44 + 43}{3} = \frac{133}{3} = 44,33$		

Таблиця 4.3 – Рівні комерційного потенціалу будь-якої наукової розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Оскільки середньоарифметична сума балів, що їх виставили експерти, складає 44,33 балів, то це свідчить, що розроблений нами автоматизований модуль має рівень комерційного потенціалу, який вважається "високим". Це пояснюється тим, що він використовує евристичну процедуру вибору ненадійних зашумлених символів, що дозволяє виконати порівняльний аналіз складності декодування.

4.2 Розрахунок витрат на розробку та проведення досліджень

При розробленні автоматизованого модуля були зроблені певні витрати:

1. Основна заробітна плата Z_o розробників, яка визначається за формулою (таблиця 4.4):

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де M – місячний посадовий оклад розробника, грн; прийmemo, що $M = (8000 \dots 24000)$ грн./місяць; T_p – кількість робочих днів у місяці; прийmemo $T_p = 22$ дні; t – кількість днів роботи розробників.

2. Додаткова заробітна плата Z_d розробників розраховується як $(10 \dots 12)\%$ від величини їх основної заробітної плати, тобто:

$$Z_d = \alpha \cdot Z_o = (0,1 \dots 0,12) \cdot Z_o. \quad (4.2)$$

Прийmemo, що $\alpha = 0,1$. Тоді для нашого випадку отримаємо:

$$Z_d = 0,1 \times 36569 = 3656,9 \approx 3657 \text{ грн.}$$

Таблиця 4.4 – Основна заробітна плата розробників

Найменування посади виконавця	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Кількість днів (годин) роботи	Витрати на оплату праці, грн.
1. Науковий керівник магістерської роботи	24000	1090,91	20 годин	$(1090,91 / 6) \times 20 = \approx 3636$ (при 6-годинному робочому дні)
2. Здобувач-магістрант (виконавець)	2000 (беремо 8000)	363,64	90 днів	≈ 32728
3. Консультант з економічної частини	18000	818,18	1,5 години	≈ 205 (при 6-годинному робочому дні)
Загалом				$3_0 = 36569$ грн.

3. Нарахування на заробітну плату $НЗП_{\text{зп}}$ розробників (дослідників) розраховуються за формулою:

$$НЗП_{\text{зп}} = (3_0 + 3_{\text{д}}) \cdot \frac{\beta}{100}, \quad (4.3)$$

де β – ставка обов'язкового єдиного внеску на державне соціальне страхування, %. Нехай $\beta = 22\%$. Тоді:

$$НЗН_{\text{зп}} = (36569 + 3657) \times 0,22 = 8849,72 \approx 8850 \text{ грн.}$$

4. Амортизація основних засобів A , які використовувались під час виконання магістерської кваліфікаційної роботи (таблиця 4.5):

$$A = \frac{Ц \cdot H_a}{100} \cdot \frac{T}{12}, \quad (4.4)$$

де $Ц$ – загальна балансова вартість основних засобів, грн.; H_a – річна норма амортизаційних відрахувань. Спрощено можна прийняти, що $H_a = (5...25)\%$; T – термін використання основних засобів, місяці.

Таблиця 4.5 – Розрахунок амортизаційних відрахувань

Найменування обладнання, приміщень тощо	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн.
1. Комп'ютерна техніка, обладнання тощо	60000	25	3,5 (при 60% використанні)	≈ 2625
2. Приміщення університету	22000	4	3,5 при 30% використанні	77
Всього				$A = 2702$ грн

5. Витрати на матеріали M розраховуються за формулою:

$$M = \sum_1^n H_i \cdot Ц_i \cdot K_i - \sum_1^n B_i \cdot Ц_b, \quad (4.5)$$

де H_i – витрати матеріалу i -го найменування, кг; $Ц_i$ – вартість матеріалу i -го найменування; K_i – коефіцієнт транспортних витрат, $K_i = (1,1...1,15)$; B_i – маса відходів матеріалу i -го найменування; $Ц_b$ – ціна відходів матеріалу i -го найменування; n – кількість видів матеріалів.

6. Витрати на комплектуючі K розраховуються за формулою:

$$K = \sum_1^n H_i \cdot C_i \cdot K_i, \quad (4.6)$$

де H_i – кількість комплектуючих i -го виду, шт.; C_i – ціна комплектуючих i -го виду; K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$; n – кількість видів комплектуючих.

Під час виконання магістерської роботи загальні витрати на матеріали та комплектуючі склали приблизно 1600 грн.

7. Витрати на силову електроенергію V_e розраховуються за формулою:

$$V_e = \frac{V \cdot P \cdot \Phi \cdot K_n}{K_d}, \quad (4.7)$$

де V – вартість 1 кВт-год електроенергії, в 2024 році $V \approx 4,8$ грн./кВт; P – установлена потужність обладнання, кВт; $P = 1,2$ кВт; Φ – фактична кількість годин роботи обладнання; прийmemo, що $\Phi = 250$ годин; K_n – коефіцієнт використання потужності; $K_n < 1 = 0,79$; K_d – коефіцієнт корисної дії, $K_d = 0,65$.

Тоді витрати на силову електроенергію будуть складати:

$$V_e = \frac{V \cdot P \cdot \Phi \cdot K_n}{K_d} = \frac{4,8 \cdot 1,2 \cdot 250 \cdot 0,79}{0,65} = 1750,15 \approx 1751 \text{ грн.}$$

8. Інші витрати $V_{\text{інш}}$ можна прийняти як $(50 \dots 300)\%$ від основної заробітної плати розробників, тобто:

$$V_{\text{інш}} = (0,5 \dots 3) \times 3_0. \quad (4.8)$$

Для нашого випадку отримаємо:

$$B_{\text{інш}} = 1,25 \times 36569 = 45711,25 \approx 45712 \text{ грн.}$$

9. Сума всіх попередніх статей витрат складає витрати на виконання нашої магістерської роботи безпосередньо розробником-магістрантом:

$$B = 36569 + 3657 + 8850 + 2702 + 1600 + 1751 + 45712 = 100841 \text{ грн.}$$

10. Загальні витрати на розробку та можливе впровадження розробленого нами автоматизованого модуля $B_{\text{заг}}$ розраховуються за формулою:

$$B_{\text{заг}} = \frac{B}{\beta}, \quad (4.9)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання цієї роботи.

Оскільки наша розробка на цей момент часу перебуває у процесі доопрацювання документації та програмного коду, а в подальшому можна розширити її функціональні можливості, доповнивши програмне забезпечення новими модулями та встановивши специфічні набори налаштувань з урахуванням особливих ситуацій, то можна прийняти, що $\beta \approx 0,6$ [31]. Тоді:

$$B_{\text{заг}} = \frac{100841}{0,6} = 168068,33 \text{ грн. або приблизно 168 тис. грн.}$$

Тобто прогнозовані загальні витрати на розробку та можливе впровадження розробленого нами автоматизованого модуля становлять приблизно 168 тис. грн.

4.3 Прогнозування комерційного ефекту від можливої комерціалізації розробки

Проведене нами дослідження ринку показало, що розроблений програмний автоматизований модуль відкриває широкий спектр можливостей для застосування нашої розробки при розв'язанні низки специфічних задач. При цьому сьогодні на ринку існує недостатня кількість подібних аналогів-програм. Частина із цих програм-аналогів написана відомими компаніями “ViaSat”, “Comtech”, “Xilinx” і перебуває у закритому доступі з можливістю використання мінімальних демоверсій.

Аналіз місткості ринку цього продукту також показує, що на сьогодні в Україні кількість реальних користувачів подібних автоматизованих модулів для декодування блокових кодів турбо-добутків може становити приблизно 50 юридичних осіб щороку. Це, насамперед, різні промислові підприємства, навчальні заклади, охоронні структури тощо. Тому можна очікувати зростання попиту на нашу розробку принаймні впродовж 3-х років після її впровадження.

Тобто, якщо розробка буде впроваджена з 1 січня 2026 року (оскільки потребує доопрацювання), то її результати будуть виявлятися у 2026-му, 2027-му та 2028-му роках. Прогноз зростання попиту складатиме по роках:

- а) 2026 р. – +20 шт. до базового року;
- б) 2027 р. – +25 шт. до базового року;
- в) 2028 р. – +40 шт. до базового року.

Економічний ефект від впровадження та можливої комерціалізації автоматизованого модуля пояснюється його кращими функціональними можливостями. Тому нашу розробку можна буде реалізовувати на ринку дещо дорожче, ніж аналоги. Так, якщо подібні за функціями розробки у 2024 році коштували на ринку приблизно 180 тис. грн, то наш модуль можна буде реалізовувати приблизно за 200 тис. грн. або на 20 тис. грн. дорожче.

Тоді можливе збільшення чистого прибутку $\Delta\Pi_1$, що його може отримати потенційний інвестор від комерціалізації, тобто виведення розробки на ринок:

$$\Delta\Pi_1 = \sum_1^n (\Delta C_0 \cdot N + C_0 \cdot \Delta N)_1 \cdot \lambda \cdot \rho \cdot \left(1 - \frac{v}{100}\right), \quad (4.10)$$

де ΔC_0 – покращення основного якісного показника від впровадження результатів розробки (для нашого випадку це є збільшення ціни реалізації нашої розробки $\Delta C_0 = 200 - 180 = + 20$ тис. грн.); N – основний кількісний показник, який визначає обсяг діяльності у році до впровадження результатів розробки; $N = 50$ шт.; ΔN – покращення основного кількісного показника від впровадження результатів розробки; C_0 – основний якісний показник (ціна), після її виведення на ринок, грн.; $C_0 = 200$ тис. грн; n – кількість років, впродовж яких очікується отримання позитивних результатів від впровадження розробки; для нашого випадку $n = 3$; λ – коефіцієнт, який враховує сплату податку на додану вартість; $\lambda = 0,8333$; ρ – коефіцієнт, який враховує рентабельність продукту (рекомендується приймати $\rho = (0,2...0,5)$; візьмемо $\rho = 0,45$); v – ставка податку на прибуток (у 2024-2028 роках $v = 18\%$).

Тоді можливе зростання чистого прибутку $\Delta\Pi_1$ для потенційного інвестора від можливого впровадження нашої розробки складе:

$$\Delta\Pi_1 = [20 \cdot 50 + 200 \cdot 20] \cdot 0,8333 \cdot 0,45 \cdot \left(1 - \frac{18}{100}\right) \approx 1538 \text{ тис. грн.}$$

$$\Delta\Pi_2 = [20 \cdot 50 + 200 \cdot 25] \cdot 0,8333 \cdot 0,45 \cdot \left(1 - \frac{18}{100}\right) \approx 1845 \text{ тис. грн.}$$

$$\Delta\Pi_3 = [20 \cdot 50 + 200 \cdot 40] \cdot 0,8333 \cdot 0,45 \cdot \left(1 - \frac{18}{100}\right) \approx 2768 \text{ тис. грн.}$$

Приведена вартість зростання для потенційного інвестора всіх чистих прибутків від можливого впровадження нашої розробки становитиме:

$$\text{ПП} = \sum_1^t \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (4.11)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, впродовж яких виявляються результати виконаної та впровадженої роботи, грн.; t – період часу, впродовж якого виявляються результати впровадженої роботи, роки ($t = 3$ роки); τ – ставка дисконтування ($\tau = 0,10$); t – період часу від моменту початку розроблення автоматизованого модуля до моменту отримання можливих чистих прибутків.

Тоді приведена вартість зростання всіх можливих чистих прибутків ПП, що їх може отримати потенційний інвестор від комерціалізації нашої розробки, складе:

$$\text{ПП} = \frac{1538}{(1+0,1)^3} + \frac{1845}{(1+0,1)^4} + \frac{2768}{(1+0,1)^5} \approx 1156 + 1260 + 1719 = 4135 \text{ тис. грн.}$$

Теперішня вартість інвестицій PV, що повинні бути вкладені в реалізацію нашої розробки:

$$\text{PV} = (1,0 \dots 5) \times B_{\text{заг.}} \quad (4.12)$$

Для нашого випадку:

$$\text{PV} = (1,0 \dots 5) \times 168 = 2,5 \times 168 = 420 \text{ тис. грн.}$$

Розраховуємо абсолютний ефект від можливих вкладених інвестицій:

$$E_{абс} = ПП - PV. \quad (4.13)$$

Абсолютний ефект від можливого впровадження нашої розробки (при прогнозованому ринку збуту) за три роки складе:

$$E_{абс} = 4135 - 420 = 3715 \text{ тис. грн.}$$

Оскільки $E_{абс} > 0$, то комерціалізація розробки може бути доцільною.

Далі розрахуємо внутрішню дохідність E_v вкладених інвестицій:

$$E_v = T_x \sqrt[3]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.14)$$

де T_x – життєвий цикл розробки, роки.

Для нашого випадку отримаємо:

$$E_v = \sqrt[3]{1 + \frac{3715}{420}} - 1 = \sqrt[3]{1 + 8,8452} - 1 = \sqrt[3]{9,8452} - 1 = 1,5799 - 1 = 0,5799 \approx 58\%.$$

Далі визначимо ту мінімальну дохідність, нижче за яку потенційному інвестору не вигідно буде займатися комерціалізацією нашої розробки. Мінімальна дохідність $\tau_{\min}$ визначається за формулою:

$$\tau_{\min} = d + f, \quad (4.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = (0,10...0,12)$; f – показник, що характеризує ризикованість вкладень; $f = (0,05...0,30)$.

Для нашого випадку отримаємо:

$$\tau_{\min} = 0,12 + 0,30 = 0,42 \text{ або } \tau_{\min} = 42\%.$$

Оскільки величина $E_v = 58\% > \tau_{\min} = 42\%$, то потенційний інвестор може бути зацікавлений у фінансуванні та комерціалізації розробленого нами автоматизованого модуля.

Далі розраховуємо термін окупності коштів, вкладених у можливу комерціалізацію розробки, за формулою:

$$T_{\text{ок}} = \frac{1}{E_v}. \quad (4.16)$$

Для нашого випадку термін окупності $T_{\text{ок}}$ коштів становитиме:

$$T_{\text{ок}} = \frac{1}{0,58} = 1,72 \text{ років} < 3 \text{ років.}$$

що свідчить про потенційну доцільність комерціалізації розробки.

Проведемо моделювання залежності величини внутрішньої дохідності вкладених потенційних інвестицій від рівня інфляції у країні. Як відомо, в наступні роки рівень інфляції може зрости. Якщо він буде 30%, то отримаємо:

$$\text{ПП} = \frac{1538}{(1+0,3)^3} + \frac{1845}{(1+0,3)^4} + \frac{2768}{(1+0,3)^5} \approx 700 + 646 + 745 = 2091 \text{ тис. грн.}$$

Тоді абсолютний економічний ефект від можливого впровадження нашої розробки за три роки складе:

$$E_{\text{абс}} = 2091 - 420 = 1671 \text{ тис. грн.}$$

Внутрішня дохідність E_v вкладених інвестицій становитиме:

$$E_v = \sqrt[3]{1 + \frac{1671}{420}} - 1 = \sqrt[3]{1 + 3,9786} - 1 = \sqrt[3]{4,9786} - 1 = 1,3785 - 1 = 0,3785 \approx 37,8\%.$$

Зроблені розрахунки у вигляді графіків наведено на рисунку 4.1.

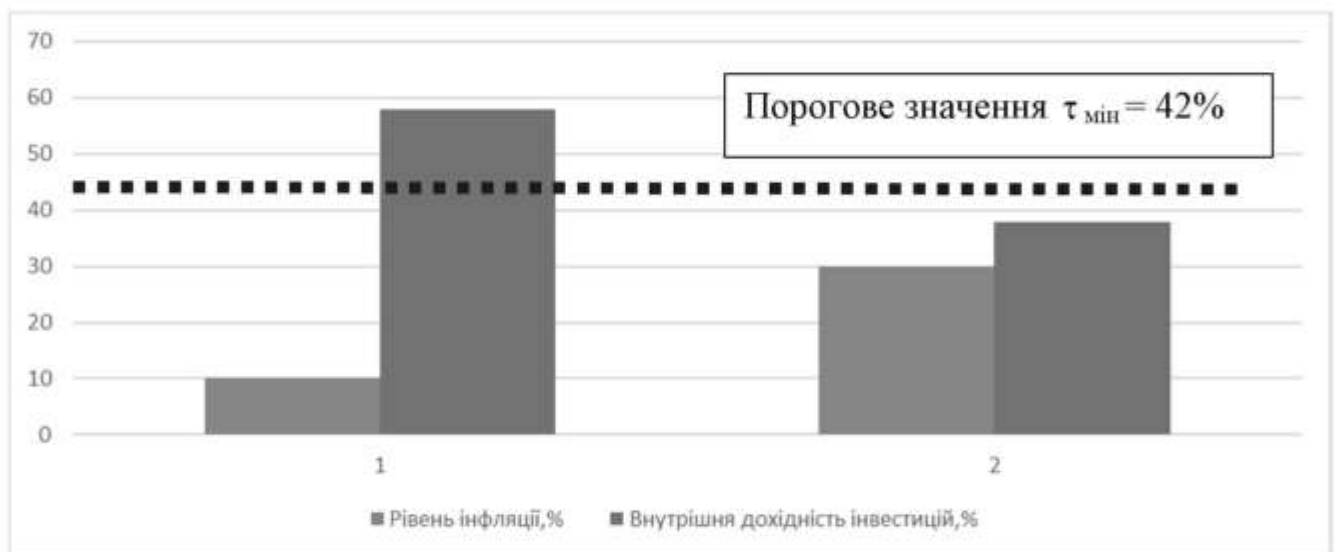


Рисунок 4.1 – Моделювання залежності величини внутрішньої дохідності потенційних інвестицій від рівня інфляції у країні

Аналіз рисунка 4.1 показує, що при рівні інфляції в 10% величина внутрішньої дохідності інвестицій становить $E_v = 58\%$, що більше порогового значення $\tau_{\text{мін}} = 42\%$, а тому комерціалізація нашої розробки може бути доцільною; при рівні інфляції в 30% величина внутрішньої дохідності інвестицій, вкладених в комерціалізацію нашої розробки, становить $E_v = 37,8\%$, що менше порогового значення $\tau_{\text{мін}} = 42\%$, а тому для прийняття остаточного рішення потрібно провести додаткові обґрунтування і розрахунки (наприклад, знизити рівень прийнятого ризику, підняти ціну реалізації розробки тощо).

4.4 Висновки

Аналіз комерційного потенціалу розробки показав, що програмний продукт за своїми характеристиками випереджає можливі аналогічні реалізації і є перспективною розробкою. Він має кращі функціональні показники, а тому є конкурентоспроможним товаром на ринку. Результати виконаної економічної частини зведено у таблицю 4.6.

Таблиця 4.6 – Результати економічної частини

Показники	Задані у технічному завданні	Досягнуті у магістерській кваліфікаційній роботі	Висновок
1. Витрати на розробку, тис. грн.	не більше 180 тис. грн.	168 тис. грн.	Виконано
2. Абсолютний ефект від впровадження розробки (в майбутній вартості грошей), тис. грн.	не менше 3500 тис. грн. (за 3 роки)	3715 тис. грн.	Виконано
3. Внутрішня норма дохідності інвестицій, %	не менше 42%	58%	Виконано
4. Термін окупності інвестицій, роки	не більше 3-ох років	1,72 року	Виконано

Таким чином, основні техніко-економічні показники модифікованого алгоритму виконані.

ВИСНОВКИ

У роботі встановлено, що використання завадостійких кодів дозволяє досягти енергетичного виграшу та знизити частоту виникнення помилок в послідовностях даних після їх проходження через канал із завадами. При високому відношенні сигнал/шум рекомендується використовувати блокові коди турбо-добутки, які забезпечують низькі частоти помилок декодування на високих швидкостях передавання даних.

Описано та формалізовано математичний апарат і структурні схеми для функціонування завадостійкого кодека коду турбо-добутку в рамках автоматизованої системи передавання даних. Використовуючи розроблену математичну модель, алгоритмічні та програмні засоби, було досліджено ефективність роботи такої системи із застосуванням імітаційної моделі.

Експериментальні дослідження підтверджують, що розроблений кодек може бути практично використаний, якщо оптимізувати його роботу під реальні умови. Модифікація незначно поступається за енергетикою, $+0,001-0,094$ дБ для $BER=10^{-5}$, дозволяючи зменшити кількість обчислень.

Аналіз комерційного потенціалу програмного забезпечення показав, що його розробка є перспективною.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Justesen J., Hoholdt T. A Course in Error-Correcting Codes. European Mathematical Society, 2017. 226 p.
2. Codes and Turbo Codes / C. Douillard, M. Jezequel, G. Battail et al. Paris: Springer, 2010. 424 p.
3. Gracie K., Hamon M.-H. Turbo and Turbo-Like Codes: Principles and Applications in Telecommunications. *Proceedings of the IEEE*. 2007. №5. pp. 1228–1254.
4. Channel Coding: Theory, Algorithms, and Applications / D. Declercq et al. Academic Press Library in Mobile and Wireless Communications, 2014. 690 p.
5. Hagenauer J., Offer E., Papke L. Iterative Decoding of Binary Block and Convolutional Codes. *IEEE Transactions on Information Theory*. 1996. Vol. 42. № 2. pp. 429–445.
6. Shami A., Al-Dweik A., Mukhtar H. Turbo Product Codes: Applications, Challenges, and Future Directions. *IEEE Communications Surveys & Tutorials*. 2016. Vol. 18. № 4. pp. 3052–3068.
7. Pyndiah R.M. Near-Optimum Decoding of Product Codes: Block Turbo Codes. *IEEE Transactions on Communications*. 1998. Vol. 46. pp. 1003-1010. URL: <http://departements.imt-atlantique.fr/data/sc/turbo/Pyn98.pdf> (дата звернення 12.03.23).
8. Morgos F.L., Cuc A.-M., Grava C. Performance Analysis of Turbo Codes, LDPC Codes, and Polar Codes over an AWGN Channel in the Presence of Inter Symbol Interference. *Sensors*. Basel, 2023. № 4. 19 p.

9. Suman K. Optimization of Heuristic Algorithms for Improving BER of Adaptive Turbo Codes. *International Journal of Advanced Research in Engineering and Technology*. 2019. № 2. pp. 414–421.

10. Зубенко В.О. Методи побудови і декодування каскадних кодових конструкцій з покращеними властивостями: автореферат дис. ... канд. техн. наук: 05.12.02. Харків, 2013. 23 с.

11. Перов В.О., Крайник Я.М., Програмне та апаратне тестування декодери turbo-product-кодів. *Інформатика та математичні методи в моделюванні*. 2018. Том 8. №3. С. 256–264.

12. Melnyk Ye.S., Ivanov Yu.Yu., Zvuzdetskyi Ye.O. Generalized Swarm Algorithm for Solving the Task of Turbo-like Codes Decoding. *Міжнародна науково-практична конференція "Інтеграція знань і технологій для сталого розвитку: нові горизонти науки, освіти та суспільства"*. Полтава, 2024. С.

13. Денбновецький С.В., Мельник І.В., Писаренко Л.Д. Кодування сигналів в електронних системах. Частина 3. Способи кодування сигналів. Том 2. Групові, ітеративні та згорткові коди. К.: КПІ ім. Ігоря Сікорського, 2021. 634 с.

14. Самохвалов Ю.Я., Бурба О.І. Методичний підхід до передпроектного дослідження об'єктів автоматизації. *Науково-практичний журнал "Сучасна спеціальна техніка"*. 2016. №3 (46). С. 115–124.

15. Особливості оцінювання параметрів процесу передавання даних із використанням турбо-кодів / Р.Н. Кветний, Ю.Ю. Іванов, С.Г. Кривогубченко, О.В. Стукач. *Метрологія і прилади*. К: ВКФ «Фавор ЛТД», 2017. С. 25–32.

16. Al-Dweik A., Le Goff S., Sharif B. A Hybrid Decoder for Block Turbo Codes. *IEEE Transactions on Communications*, 2009. Vol. 57. pp. 1229–1232.
17. Justesen J. Performance of Product Codes and Related Structures with Iterated Decoding. *IEEE Transactions on Communications*, 2011. Vol. 5 (2). pp. 407–415.
18. Chase D. Class of Algorithms for Decoding Block Codes with Channel Measurement Information. *IEEE Transactions on Information Theory*, 1972. Vol. 18. № 1. pp. 170–182.
19. Sidorenko A.A. Study of the Block and Convolutional Code Decoding Efficiency Using a Turbo Code Decoder. *Journal of Physics*. 2024. Vol. 2697. Bukhara. 8 p.
20. Efficient Decoder for Turbo Product Codes Based on Quadratic Residue Codes / J. Dong, Y. Li, R. Liu et al. *Electronics*. 2022. Vol. 11. 13 p. URL: <https://www.mdpi.com/2079-9292/11/21/3598> (дата звернення 20.11.2024).
21. Lu P., Lu E., Chen T. An Efficient Hybrid Decoder for Block Turbo Codes. *IEEE Communication Letters*, 2014. Vol. 18. pp. 2077–2080.
22. Kishore J.H., Yamuna B., Balasubramanian K. Design of a Fast Chase Algorithm based High Speed Turbo Product Code Decoder. *Proceedings of the International Conference on Advances in Computing and Communications*. Kochi (India), 21–23 October 2021. pp. 1–5. URL: https://www.researchgate.net/publication/351012386_Design_of_High-Speed_Turbo_Product_Code_Decoder (дата звернення 24.11.2024).
23. Hardware-Oriented Turbo-Product Codes Decoder Architecture / Y.M. Krainyk, V.O. Perov, M.P. Musiyenko et al. *Intelligent Data Acquisition and*

Advanced Computing Systems: Technology and Applications. Bucharest, 2017. pp. 151–154.

24. Zvuzdetskyi Ye.O., Ivanov Yu.Yu. List Decoding of Block Turbo-Products-Codes. *Proceedings of the 8th International scientific and practical conference "Scientific research in the modern world"*. 2023. pp. 199–201.

25. An Efficient Maximum-Likelihood-Decoding Algorithm for Linear Block Codes with Algebraic Decoder / T. Kaneko, T. Nishijima, H. Inazumi, S. Hirasawa. *IEEE Transactions on Information Theory*. 1994. Vol. 40 (2). pp. 320–327.

26. Оцінювання складності декодування згорткових турбо-кодів та блокових кодів турбо-добутків / Ю.Ю. Іванов, Б.О. Боднарченко, Є.О. Звуздецький, Ю.С. Здітовецький. *Вісник Вінницького політехнічного інституту*. Вінниця: ВНТУ, 2024. № 1. С. 51–55.

27. Jiang Y. A Practical Guide to Error-Control Coding Using MATLAB. London: Artech House, 2010. 293 p.

28. Communication Toolbox. Error Detection and Correction. Turbo-products – Codec Information. URL: <https://es.mathworks.com/help/comm/ref/tpcdecoder.html> (дата звернення 25.11.24).

29. Семенов С.Г., Ліпчанська О.В., Ліпчанський М.В. Аналіз методів управління передачею відеопотоку даних та вимог до якості їх передавання. *Системи управління, навігації та зв'язку*. Полтава: ПНТУ ім. Ю. Кондратюка, 2018. № 3 (49). С. 139–142.

30. Іванов Ю.Ю. Експериментальне дослідження завадостійкості турбо-кодів: числові оцінки та імітаційне моделювання нового субоптимального

алгоритма PL-log-MAP. *Вісник Вінницького політехнічного інституту*. Вінниця: ВНТУ, 2016. № 5. С. 76–84.

31. Козловський В.О., Лесько О.Й., Кавецький В.В. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. Вінниця: ВНТУ, 2021. 42 с.

ДОДАТКИ

Додаток А
(обов'язковий)
Технічне завдання

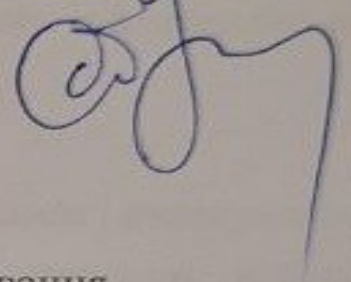
97

ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ

д.т.н., проф. Бісікало О. В.

« 14 » 10 2024 р.



ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу
«Розробка автоматизованого модуля для декодування
блокових кодів турбо-добутків»
08-31.МКР.003.02.000 ТЗ

Керівник роботи:

к.т.н., доц. каф. АІТ

Іванов Ю. Ю.

«22» 09 2024 р.

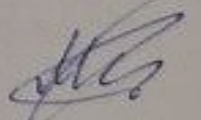


Виконавець:

ст. гр. ІАКІТР-23м

Мельник Є. С.

«22» 09 2024 р.



1. Назва та галузь застосування

У роботі розглядаються турбо-добутки, які ефективні для передавання невеликих кодових блоків на дуже високих кодових швидкостях. Результатом дослідження є математичне, алгоритмічне та програмне забезпечення, яке дозволяє виконати експерименти для системи передавання даних.

2. Підстави для розробки

Розробку здійснювати на підставі наказу по університету № 310 від 17.09.2024 та завдання до магістерської кваліфікаційної роботи, складеного та затвердженого кафедрою «Автоматизації та інтелектуальних інформаційних технологій».

3. Мета та призначення розробки

Метою роботи є зменшення обчислювальної складності декодування блокового коду турбо-добутку за рахунок спрощення процедури вибору ненадійних символів у зашумленій послідовності даних без значного зниження ефективності роботи системи передавання даних.

4. Джерела розробки

1. An Improved Iterative Decoding Algorithm for Block Turbo Codes / M. Lalam, K.A. Cavalec, D. Leroux et al. *IEEE International Symposium on Information Theory*. 2006. pp. 2403—2407.

2. Suman K. Optimization of Heuristic Algorithms for Improving BER of Adaptive Turbo Codes. *International Journal of Advanced Research in Engineering and Technology*. 2019. № 2. pp 414–421.

3. Morgos F.L., Cuc A.-M., Grava C. Performance Analysis of Turbo Codes, LDPC Codes, and Polar Codes over an AWGN Channel in the Presence of Inter Symbol Interference. *Sensors*. Basel, 2023. № 4. 19 p.

5. Показники призначення

Розроблено програмне забезпечення, яке дозволяє провести експериментальні дослідження для системи передавання даних з турбо-добутками.

Основні технічні вимоги та мінімальні системні вимоги до програми: операційна система *Windows 7 / 10 / 11*; наявність клавіатури та миші; процесор

не слабше *Core i5 (Ryzen 5)*; оперативна пам'ять – 8 Гб; відеопам'ять – 1 Гб; жорсткий диск – 200 Гб і більше.

Вхідні дані: кількість ненадійних символів; кількість ітерацій декодування; послідовності інформаційних бітів та мультимедійні дані. Результати роботи програми: експериментальні графічні залежності $BER = f(E_b/N_0)$; декодоване зображення.

6. Економічні показники

До економічних показників входять:

- термін окупності – до 3-х років;
- абсолютний ефект від впровадження розробки – не менше 3500 тис. грн.;
- мінімальна дохідність – не менше 42%;
- інші економічні переваги у порівнянні з аналогами.

7. Стадії розробки

1. Розділ 1 «Теоретична основа та аналіз предметної області роботи» має бути виконаний до 11.10.24.

2. Розділ 2 «Розробка модифікованого математичного апарату для декодування блокових кодів турбо-добутків» має бути виконаний до 26.10.24.

3. Розділ 3 «Розробка програмного забезпечення та експериментальні дослідження» має бути виконаний до 10.11.24.

4. Економічний розділ має бути виконаний до 30.11.24.

8. Порядок контролю та приймання

1. Рубіжний контроль провести до 30.12.24.

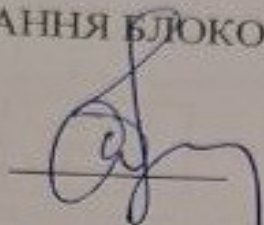
2. Попередній захист магістерської кваліфікаційної роботи провести до 05.12.24.

3. Захист магістерської кваліфікаційної роботи провести до 19.12.24.

ІЛЮСТРАТИВНА ЧАСТИНА

РОЗРОБКА АВТОМАТИЗОВАНОГО МОДУЛЯ
ДЛЯ ДЕКОДУВАННЯ БЛОКОВИХ КОДІВ ТУРБО-ДОБУТКІВ

Зав. кафедри АПТ



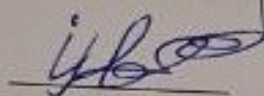
(підпис)

д-р техн. наук, професор каф. АПТ

Бісікало О. В.

(науковий ступінь, вчене звання, ініціали та прізвище)

Керівник роботи



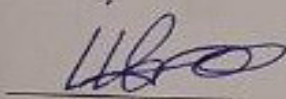
(підпис)

канд. техн. наук, доцент каф. АПТ

Іванов Ю. Ю.

(науковий ступінь, вчене звання, ініціали та прізвище)

Тех. контроль



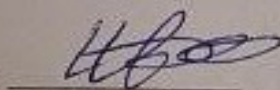
(підпис)

канд. техн. наук, доцент каф. АПТ

Іванов Ю. Ю.

(науковий ступінь, вчене звання, ініціали та прізвище)

Нормоконтроль



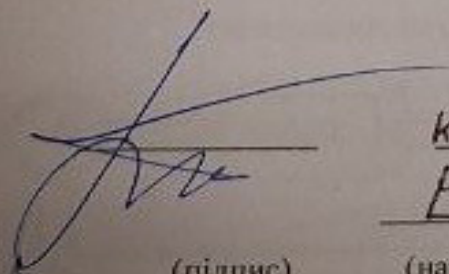
(підпис)

канд. техн. наук, доцент каф. АПТ

Іванов Ю. Ю.

(науковий ступінь, вчене звання, ініціали та прізвище)

Опонент



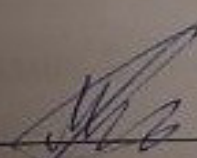
(підпис)

канд. Техн. наук, доцент каф. КСУ

Біцков М. М.

(науковий ступінь, вчене звання, ініціали та прізвище)

Студент гр. 1АКІТР-23м



(підпис)

Мельник Є. С.

(ініціали та прізвище)

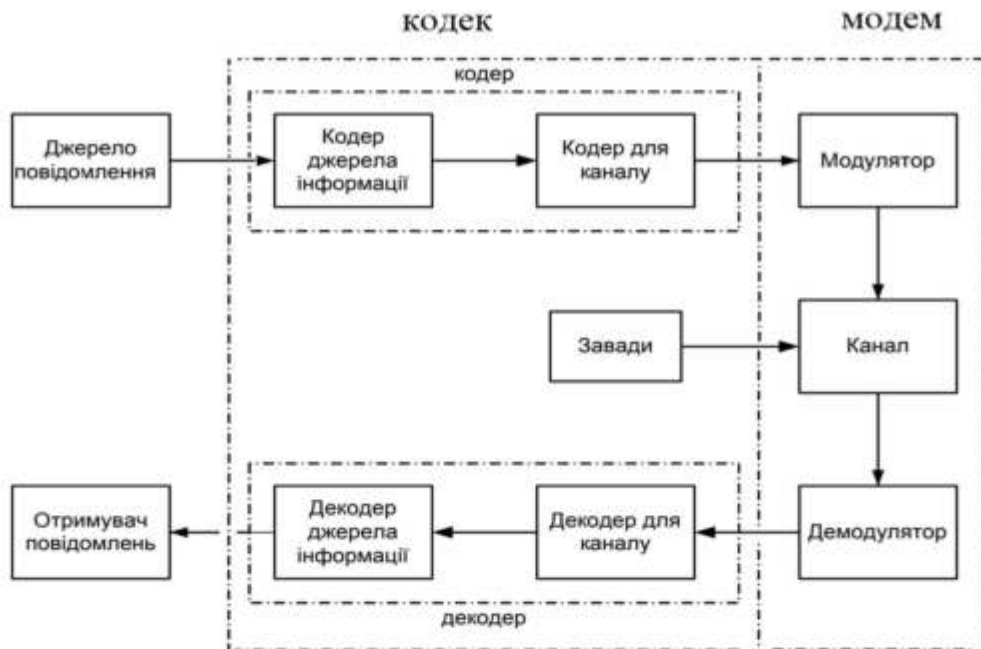


Рисунок Б.1 – Структура системи передавання даних



Рисунок Б.2 – Структура кодера

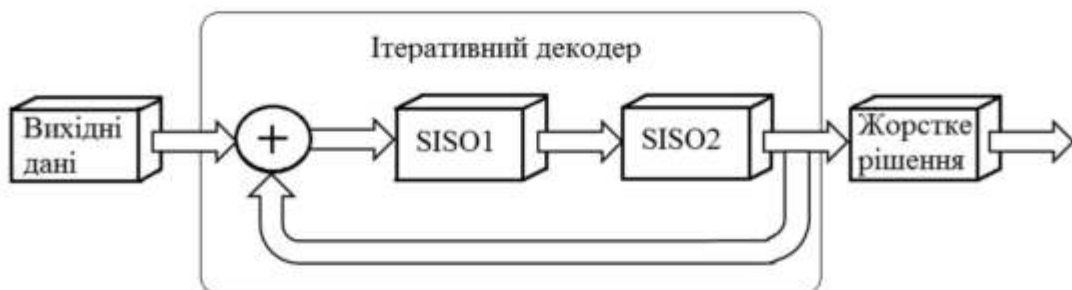


Рисунок Б.3 – Структура декодера

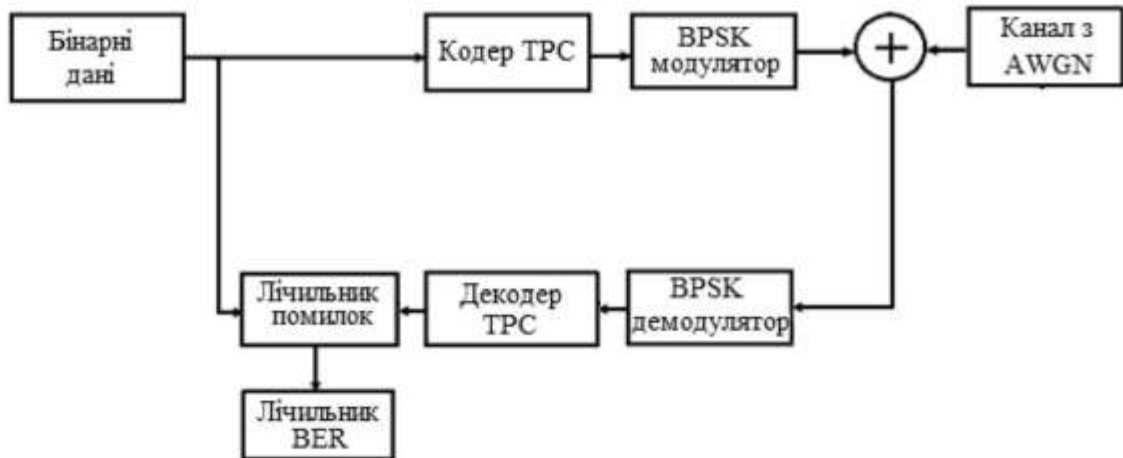


Рисунок Б.4 – Схема експерименту



Рисунок Б.5 – Схема модифікованого методу генерації векторів помилок



Рисунок Б.6 – Схема головної програми для декодера

Додаток В

(обов'язковий)

Лістинг програмного забезпечення

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
alpha = [0.0 0.2 0.3 0.5 0.7 0.9 1.0 1.0 1.0 1.0];
beta = [0.2 0.4 0.6 0.8 1.0 1.0 1.0 1.0 1.0 1.0];
n = 15;
k = 11;
info_word_length = k * k;
SNRdB = 0:2:8;
SNR = 10.^(SNRdB./10);
ber1 = zeros(length(SNR), 1);
ber2 = zeros(length(SNR), 1);
ber3 = zeros(length(SNR), 1);
ber4 = zeros(length(SNR), 1);

for i = 1:length(SNR)
    SNR(1, i)
    xj1 = 0;
    xj2 = 0;
    xj3 = 0;
    xj4 = 0;

    for jj = 1:1000
        [R, msg, v] = received_signal(SNR(i));
        W = zeros(15, 15);
        m = 1;
        R1 = R + W * alpha(2 * (m - 1) + 1);
        for k = 1:15
            R_row = R1(k, :);
            [W1, D] = half_iteration(R_row, beta(2 * (m - 1) + 1));
            W(k, :) = W1;
        end
        R2 = R + W * alpha(2 * (m - 1) + 2);
        for k = 1:15
            R_col = R2(:, k);
            [W1, D] = half_iteration(R_col', beta(2 * (m - 1) + 2));
            W(:, k) = W1;
        end
        out_it1 = W > 0;
        z1 = length(find((msg ~= out_it1(5:15, 5:15))));
        xj1 = xj1 + z1;
    end
end

```

```

m = 2;
R1 = R + W * alpha(2 * (m - 1) + 1);
for k = 1:15
    R_row = R1(k, :);
    [W1, D] = half_iteration(R_row, beta(2 * (m - 1) + 1));
    W(k, :) = W1;
end
R2 = R + W * alpha(2 * (m - 1) + 2);
for k = 1:15
    R_col = R2(:, k);
    [W1, D] = half_iteration(R_col', beta(2 * (m - 1) + 2));
    W(:, k) = W1;
end
out_it2 = W > 0;
z2 = length(find((msg ~= out_it2(5:15, 5:15))));
xj2 = xj2 + z2;

```

```

m = 3;
R1 = R + W * alpha(2 * (m - 1) + 1);
for k = 1:15
    R_row = R1(k, :);
    [W1, D] = half_iteration(R_row, beta(2 * (m - 1) + 1));
    W(k, :) = W1;
end
R2 = R + W * alpha(2 * (m - 1) + 2);
for k=1:15
    R_col = R2(:, k);
    [W1, D] = half_iteration(R_col', beta(2 * (m - 1) + 2));
    W(:, k) = W1;
end
out_it3 = W > 0;
z3 = length(find((msg ~= out_it3(5:15, 5:15))));
xj3 = xj3 + z3;

```

```

m = 4;
R1 = R + W * alpha(2 * (m - 1) + 1);
for k = 1:15
    R_row = R1(k, :);
    [W1, D] = half_iteration(R_row, beta(2 * (m - 1) + 1));
    W(k, :) = W1;
end
R2 = R + W * alpha(2 * (m - 1) + 2);
for k = 1:15
    R_col = R2(:, k);
    [W1, D] = half_iteration(R_col', beta(2 * (m - 1) + 2));

```

```
W(:, k) = W1;
end
out_it4 = W > 0;
z4 = length(find((msg ~= out_it4(5:15, 5:15))));
xj4 = xj4 + z4;
end

[xj1 xj2 xj3 xj4]
ber1(i, 1) = xj1/(repetition * info_word_length);
ber2(i, 1) = xj2/(repetition * info_word_length);
ber3(i, 1) = xj3/(repetition * info_word_length);
ber4(i, 1) = xj4/(repetition * info_word_length);
end

[SNRdB' ber1 ber2 ber3 ber4]
figure
semilogy(SNRdB, ber1(:, 1), 'r-<', 'linewidth', 2.0)
hold on;
semilogy(SNRdB, ber2(:, 1), 'g-<', 'linewidth', 2.0)
semilogy(SNRdB, ber3(:, 1), 'b-<', 'linewidth', 2.0)
semilogy(SNRdB, ber4(:, 1), 'm-<', 'linewidth', 2.0)
theoretical_awgn = 0.5 * erfc(sqrt(SNR));
semilogy(SNRdB, theoretical_awgn, 'ko-', 'LineWidth', 2);
grid on;
legend('1st iteration', '2nd iteration', '3rd iteration', '4th iteration', 'AWGN BPSK');
xlabel('SNR(dB)');
ylabel('Bit Error Rate');
axis([1 8 10^-6 1])
```

Додаток Г
(обов'язковий)
Довідка про впровадження

Додаток Д
(обов'язковий)
Протокол перевірки МКР

ПРОТОКОЛ ПЕРЕВІРКИ ДИПЛОМНОЇ РОБОТИ НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка автоматизованого модуля для декодування блокових кодів турбо-добутків.

(тема роботи)

Тип роботи: магістерська кваліфікаційна робота (МКР)

(БДР, БДП, МКР)

Підрозділ: кафедра автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, ІАКІТР-23м

(кафедра, факультет, навчальна група)

Науковий керівник: Іванов Ю. Ю., доцент каф. АІТ

(прізвище, ініціали, посада)

Показники звіту подібності Turnitin (Unicheck)

Оригінальність 88 % Схожість 12 %

Аналіз звіту подібності (відмітити потрібне):

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

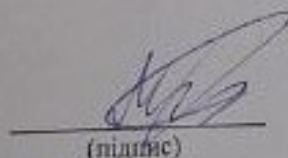
Особа, відповідальна за перевірку



Маслій Р. В.
(прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Turnitin (Unicheck, StrikePlagiarism) щодо роботи.

Автор



(підпис)

Мельник Є. С.
(прізвище, ініціали)

Керівник роботи



(підпис)

Іванов Ю. Ю.
(прізвище, ініціали)