Вінницький національний технічний університет
Факультет комп'ютерних систем та автоматики
Кафедра автоматизації та інтелектуальних інформаційних технологій

Пояснювальна записка

до бакалаврської дипломної роботи

бакалавр

(освітньо-кваліфікаційний рівень)

на тему: Дослідження методу турбо-декодування на основі списку кодових слів

Виконав: студент 4 курсу, групи 1СІ-16б
Спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології
Нікіфоренко С. В.

Керівник: к.т.н., доцент каф. АІІТ
Кривогубченко С. Г.

Рецензент: _____

Вінниця ВНТУ – 2020 року

Вінницький національний технічний університет

Факультет комп'ютерних систем і автоматики

Кафедра автоматизації та інтелектуальних інформаційних технологій

Освітньо-кваліфікаційний рівень бакалавр

Спеціальність 151 – «Автоматизація та комп'ютерно-інтегровані технології»

ЗАТВЕРДЖУЮ
Завідувач кафедрою АІІТ
д.т.н., проф. Квєтний Р. Н.
«___»_____ 202_ р.

ЗАВДАННЯ

НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Нікіфоренку Сергію Володимировичу

1. Тема роботи: Дослідження методу турбо-декодування на основі списку кодових слів.
   Керівник роботи: к.т.н., доц. каф. АІІТ Кривогубченко Сергій Григорович
   Затверджені наказом ВНТУ від «30» 08 2019 року № 1.
2. Строк подання роботи студентом: «20» 06 2020 р.
3. Вихідні дані до роботи: метод декодування на основі списку кодових слів; кількість найменш надійних символів *t*; турбо-код-добуток; бінарні послідовності та мультимедійні дані; система математичного моделювання *MatLab*.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): теоретична основа завадостійкого кодування даних; розробка математичного апарату методу декодування на основі списку кодових слів; експериментальне дослідження турбо-декодера.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): схема програми, модель системи передавання даних, експериментальні криві.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Кривогубченко С. Г., к.т.н., доц. каф. АІІТ Іванов Ю. Ю., к.т.н., доц. каф. АІІТ		

7. Дата видачі завдання: «10»____09____ 2019 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Теоретична основа завадостійкого кодування даних	до 10.11.2019	
2	Розробка математичного апарату методу декодування на основі списку кодових слів	до 10.02.2020	
3	Експериментальне дослідження турбо-декодера	до 10.05.2020	
4	Оформлення пояснювальної записки і графічного матеріалу	до 30.05.2020	
5	Попередній захист роботи	до 04.06.2020	
6	Остаточний захист роботи	до 20.06.2020	

Студент _____ (підпис) Нікіфоренко С. В. (прізвище та ініціали)

Керівник роботи _____ (підпис) Кривогубченко С. Г. (прізвище та ініціали)

АНОТАЦІЯ

У роботі проаналізовано особливості декодування турбо-кодів-добутків на основі списку кодових слів. З використанням розробленого математичного, алгоритмічного та програмного забезпечення проведено експериментальне дослідження ефективності роботи системи передавання даних.

ABSTRACT

In this work have been analyzed some features of turbo-product-codes augmented list decoding. With the usage of developed mathematical model, algorithms and software an experimental research of the efficiency of the data transmission system with turbo-product-codes has been performed.

ЗМІСТ

ВСТУП

Актуальність теми. Системи цифрового зв'язку складаються з територіально розподіленої сукупності обчислювальних секторів із високошвидкісними каналами передавання даних, інформаційно-обчислювальними комплексами, апаратними та програмними засобами, які виконують автоматизований збір, збереження, оброблення, передавання та отримання даних [1, 2]. Наявність завад різної природи зумовлює актуальність розробки нових методів і засобів, які дозволяють виявляти та коригувати помилки передавання даних. Особливої уваги заслуговують завадостійкі коди, які дозволяють ефективно усувати помилки в каналі зв'язку, забезпечуючи високу вірогідність переданих даних при малих співвідношеннях сигнал/шум. На сучасному етапі розвитку теорії завадостійкого кодування перспективним є турбо-код (ТК) [3], який здатний забезпечувати необхідні характеристики завадостійкості, максимально наближаючись до теоретичної межі Шеннона [4].

Згорткові ТК були представлені на міжнародній конференції зв'язку у 1993 році [5-8]. По аналогії зі згортковими ТК у 1994 році запропоновано використовувати блокові турбокоди або турбокоди-добутки (ТРС – Turbo-Product Codes) [9, 10], які більш ефективно працюють з невеликими кодовими блоками на дуже високих кодових швидкостях. Слід зазначити, що дані коди мають меншу за згорткові ТК складність декодування, а також у них практично відсутній "поріг помилок" (error floor). Але екстремальна задача декодування ТРС залишається важкою.

Саме тому спостерігається збільшення кількості досліджень за цим напрямком у наукових роботах різних авторів, зокрема R. Pyndiah [9, 10], A. Al-Dweik [11], M. Ghnimat [12], А. Архіпкіна [13], Ю.Б. Зубарєва [14] та інших. Вони виділяють метод декодування ТРС на основі списку кодових слів (ALD — augmented list decoding), який є досить ефективним та зручним для використання. Тому дослідження систем передавання даних з подібними декодерами актуальні.

Метою роботи є розширення функціональних можливостей програмного забезпечення для експериментального дослідження ефективності роботи методу декодування на основі списку кодових слів.

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

– проаналізувати роботу сучасних завадостійких кодів;

– проаналізувати математичний апарат методу декодування турбо-кодів-добутків на основі списку кодових слів;

– розробити програмні засоби та дослідити ефективність розробленого турбо-декодера.

Об'єктом дослідження є процес декодування даних у системах цифрового зв'язку.

Предметом дослідження є моделі, методи та інструментальні засоби декодування кодів у системах цифрового зв'язку.

Методи дослідження. У роботі використано положення теорії ймовірності, теорії інформації і кодування, а також алгебру логарифмічного відношення функцій правдоподібності для дослідження методу декодування на основі списку кодових слів; імітаційне моделювання для аналізу та перевірки достовірності отриманих теоретичних результатів. Обробка експериментальних даних виконана за допомогою середовища *MatLab*.

Науково-технічний результат роботи полягає у тому, що на основі теоретичних положень розроблене математичне, алгоритмічне та програмне забезпечення, яке може використовуватися для дослідження ефективності методу декодування на основі списку кодових слів.

У роботі отримані результати, які мають *практичну цінність*, а саме: наведено експериментальні дослідження ефективності роботи методу декодування на основі списку кодових слів.

Апробація результатів та публікації. За результатами даної роботи опубліковано тези доповіді [15] на XLIX науково-технічній конференції Факультету комп'ютерних систем і автоматики (м. Вінниця, 2020).

1 ТЕОРЕТИЧНА ОСНОВА ЗАВАДОСТІЙКОГО КОДУВАННЯ ДАНИХ

1.1 Аналіз методів завадостійкого кодування

Завадостійким кодуванням називається кодування, яке дозволяє здійснити виявлення або виявлення та виправлення помилок в прийнятих кодових комбінаціях. 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. 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 [12-14].

Сучасна теорія завадостійкого кодування надає широкий набір коригувальних кодів різноманітної структури та коригувальної здатності (рисунок 1.1). Такі коди поділяються за способом формування на блокові й неперервні або деревовидні [12].

Блоковий код задає блок з k інформаційних символів n-символьним кодовим словом. Швидкість R блокового коду визначається рівністю $R = k / n$. Деревовидний код складніший. Він відображає нескінченну послідовність інформаційних символів, яка надходить зі швидкістю k_0 символів за один інтервал часу, в неперервну послідовність символів кодового слова зі швидкістю n_0 символів за один інтервал часу.

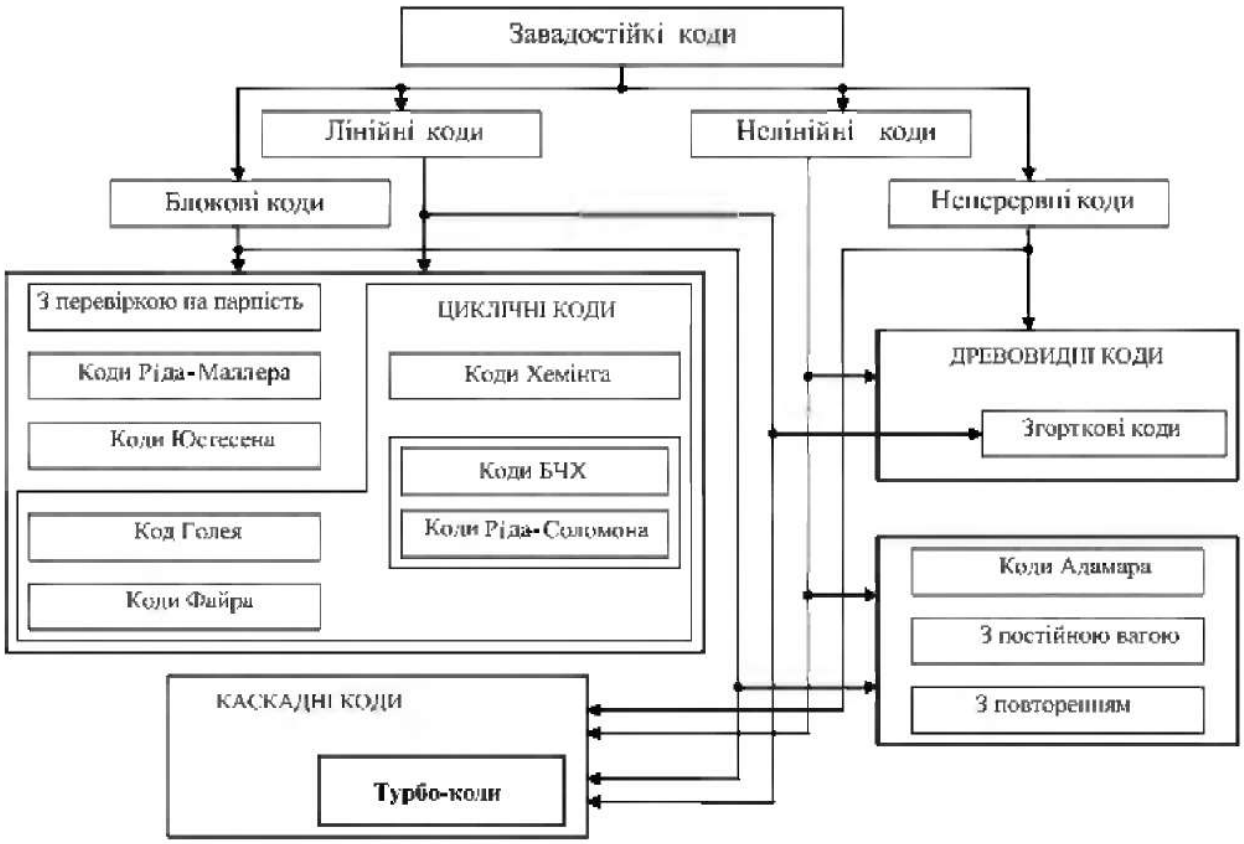

Рисунок 1.1 – Методи завадостійкого кодування

В 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 [12].

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 [13].

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.

Для порівняння кодових комбінацій X та Y довжиною n використовують відстань d, яка показує кількість позицій, в яких дві комбінації відрізняються

$$d = W(X \oplus Y) = \sum_{i=1}^{n} | x_i^X - y_i^Y |, \tag{1.1}$$

де W — вагова функція; x та y — значення розрядів 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. Особливу важливість для характеристики коригувальних властивостей коду має мінімальна кодова відстань d_{min}, яка визначається при попарному порівнянні всіх кодових комбінацій та виборі мінімального значення, тобто [13]

$$d_{\min} = \min_{i=1,\dots q}(\{d_i\}), \tag{1.2}$$

де $q = \dfrac{n^2 + n}{2}$ – кількість порівнянь.

Мінімальна кодова відстань є найважливішою характеристикою завадостійких кодів, яка вказує на гарантовану кількість помилок e, які виправляє даний код, тобто [1]

$$e \le \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor. \tag{1.3}$$

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 C зазвичай оцінюють величиною, пропорційною кількості елементів породжуючої матриці коду [1]

$$C = n^2(1 - R). \tag{1.4}$$

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]

$$C = m^n, \tag{1.5}$$

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.

Важливою складовою завадостійкого кодування є розуміння двох фундаментальних теорем, які фактично визначають достовірність передавання даних.

Теорема Шеннона [1, 16-20]: якщо швидкість передавання даних R не більша пропускної здатності каналу C, то повідомлення можуть передаватися каналом та бути відновлені з малою ймовірністю помилки, при цьому швидкість передавання даних буде близькою до швидкості їх створення джерелом.

Теорема Шеннона-Хартлі [1, 16-20]: верхня теоретична межа пропускної здатності каналу зв'язку з адитивним білим гаусівським шумом (АБГШ) за умови, що $R \le C$, визначається у вигляді

$$C = f(W,S,N) = W \cdot \log_2\left(1+\frac{S}{N}\right). \qquad (1.6)$$

де W – ширина полоси пропускання ($Гц$); S – потужність прийнятого сигналу ($дБ$); N – середня потужність шуму ($дБ$).

Отже, величини W, S та N дозволяють визначити межі швидкості передавання даних, але не ймовірність появи помилки. Формулу (1.6) можна представити із застосуванням нормованого відношення сигнал / шум $\frac{E_b}{N_o}$ замість $\frac{S}{N}$. Потужність шуму визначимо як

$$N = N_0 \cdot W, \qquad (1.7)$$

де N_0 – спектральна щільність потужності шуму.

На основі (1.7) та (1.6)

$$\frac{C}{W} = \log_2\left(1 + \frac{S}{N_0 \cdot W}\right). \tag{1.8}$$

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

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

$$\frac{S \cdot T_b}{N_0} = \frac{S}{R \cdot N_0} = \frac{S}{C \cdot N_0} = \frac{E_b}{N_0}, \tag{1.9}$$

де E_b – енергія некодованого біта; T_b – час передавання символу.

Виконавши ряд перетворень в (1.9), отримаємо

$$\frac{C}{W} = \log_2\left(1 + \frac{E_b \cdot C}{N_0 \cdot W}\right) = \log_2(1 + z), \tag{1.10}$$

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

Тоді з виразу (1.9) можна визначити, що

$$\frac{E_b}{N_o} = \left(\frac{W}{C}\right) \cdot \left(2^{\frac{C}{W}} - 1\right) = \frac{2^{\frac{C}{W}} - 1}{\frac{C}{W}} = \frac{2^{\eta} - 1}{\eta}, \tag{1.11}$$

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

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

$$\lim_{z\to 0}\left(\frac{C}{W}\right) = \lim_{z\to 0}\left(z\cdot \log_2(1+z)^{1/z}\right), \tag{1.12}$$

$$\frac{C}{W} = z\cdot \log_2 e \;\rightarrow\; \frac{E_b}{N_0} = \frac{1}{\log_2 e} = \ln 2 \approx 0{,}693. \tag{1.13}$$

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

$$\frac{E_b}{N_{0\ dB}} = 10\cdot \lg\frac{E_b}{N_o} \approx -1{,}6\ (\partial Б). \tag{1.14}$$

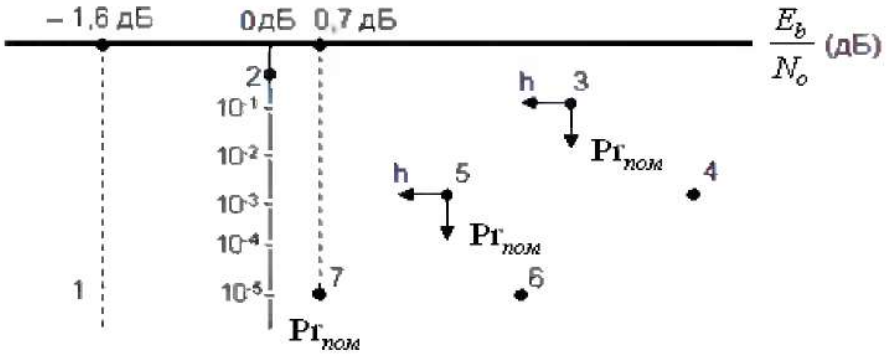

Рисунок 1.2 – Суть поняття кодовий резерв

Наприклад, для досягнення коефіцієнта бітових помилок $BER = 10^{-5}$ з використанням тільки бінарної фазової модуляції (БФМ) потрібно $\frac{E_b}{N_0} = 9{,}6\ (\partial Б)$. Результат теореми говорить, що застосування кодування даних дозволить отримати кодовий резерв у 11,2 ($\partial Б$).

Робоча точка 2 на рисунку 1.2 вказує на відсутність кодування даних. Тоді для двійкових систем обміну даними ймовірність помилки

$$p_{пом}\left(\frac{E_b}{N_0}=0\right)=p_{пр}\left(\frac{E_b}{N_0}=0\right)=0{,}5. \qquad (1.15)$$

Це важливо при розробці та дослідженні імітаційних моделей двійкових каналів зв'язку. Позиції 3-7 формують з точкою 2 траєкторії ефективності системи $T_{234}<T_{256}<T_{276}$. При розробці системи зв'язку головною метою є мінімізація показника $\frac{E_b}{N_0}$ для заданого значення *BER*, що можна представити функцією

$$p\approx BER=f\left(\frac{E_b}{N_0}\right),\ \frac{E_b}{N_0}\to\min. \qquad (1.16)$$

1.2 Аналіз засобів завадостійкого кодування даних

Основна задача будь-якої системи передавання даних – передавання інформації із заданими вірогідністю та швидкістю передавання. На сьогоднішній день головним завданням спеціалістів у галузі теорії кодування є розробка систем передавання даних, в яких задано необхідні швидкість та вірогідність передавання. Дані вимоги знаходяться у протиріччі, оскільки зростання швидкості передавання призводить до зменшення вірогідності. Із теорем Шеннона відомо, що необхідно застосовувати дуже довгі коди, але у реальних умовах присутні обмеження на реалізацію системи зв'язку [17].

Узагальнена структура для роботи з даними представлена на рисунку 1.3, де X – вхідна послідовність; $D=\{D_1,...,D_n\}$ – дані після кодування; $D_i=\{D_k,...,D_{k,v}\}$ – послідовність символів в момент *k*; $x_\xi=\{x_{\xi 1},...,x_{\xi n}\}$ – дані з

виходу каналу з шумом; $x_{\xi_i} = \{x_{\xi_{k,1}},...,x_{\xi_{k,v}}\}$ – послідовність зашумлених символів в момент k; v – перевірочні символи; $\overline{X}$ – інформація з джерела після проходження всіх вузлів системи [18, 19].

Кодер каналу отримує інформаційні символи від джерела даних та додає надлишкові символи, що дозволяє виправити помилки, які з'являються після модуляції сигналу, передавання каналом зв'язку та демодуляції. Декодер каналу приймає рішення та надсилає до приймача. Декодер та кодер називають кодеком каналу зв'язку, а модулятор та демодулятор представляють модем [19].

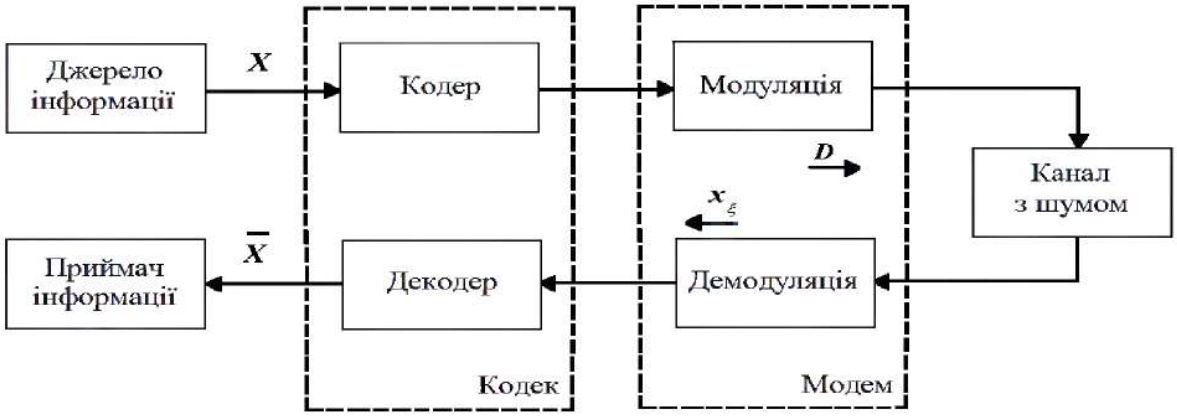

Рисунок 1.3 – Структура обміну даними

У ході модуляції символи $D = \{D_0,...,D_n\}$ стають сигналом

$$\overline{D_{k,v}} = (-1)^{D_{k,v}+1} \cdot \sqrt{E_c}\,, \tag{1.17}$$

де E_c – енергія кодованого символу.

Демодуляція сприяє перетворенню сигналу $x_\xi = \{(x_\xi)_0,...,(x_\xi)_n\}$ на символи (м'які рішення)

$$\overline{D_{k,v}} = \begin{cases} -\sqrt{E_c} \Rightarrow & D_{k,v} = 0; \\ \sqrt{E_c} \Rightarrow & D_{k,v} = 1, \end{cases} \tag{1.18}$$

Зашумлений сигнал в каналі зв'язку визначається у формі

$$x_{\xi k,i} = \overline{D_{k,i}} + \xi . \qquad (1.19)$$

де ξ – величини шуму з каналу зв'язку.

Робота декодерів базується на використанні жорсткого або м'якого прийняття рішення. Для першого варіанту кожному бінарному символу в демодуляторі присвоюються значення 0 або 1, залежно від того, більше або менше прийняте значення від певного порогу. Декодер із м'яким прийняттям рішення отримує, крім бінарної величини 0 чи 1 ще довірчу величину, яка зв'язана із заданим бітом. Коли демодулятор визначився (не впевнений), то біту надається велике (мале) дійсне число, тобто ступінь упевненості високий (низький). Декодер з м'яким входом може надавати дані з жорстким або м'яким рішенням. Перспективні декодери з м'яким входом та м'яким виходом (*soft-input soft-output*) [20-22].

1.3 Структурні особливості турбо-кодів-добутків

Розглянемо кодер двовимірного ТРС. Така кодова структура вимагає використання двох ідентичних компонентних блокових кодів (наприклад код з перевіркою на парність, Хеммінга, Боуза-Чоудхурі-Хоквінгема, Ріда-Соломона тощо) 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 переміщування бітів непотрібна.

При реалізації операції кодування ТРС вихідні дані записуються в двовимірний масив по рядках, після чого кодуються по рядках за допомогою першого коду, а потім дані і перевірочні біти (або тільки дані) першого коду кодуються за стовпцями другим кодом. Для передавання в канал зв'язку дані з

масиву зчитуються по рядках. Мінімальна відстань такого коду $d_{min} = d_1 d_2$, де d_i – мінімальна відстань горизонтального та вертикального кодів. Швидкість коду відповідно становить $R = R_1 R_2$. Аспекти кодування / декодування компонентних кодів описані у роботах [9, 13-15, 20].

Наприклад, на рисунку 1.4 зображено структуру ТРС, 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. 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 стовпець теж представляє собою кодове слово.

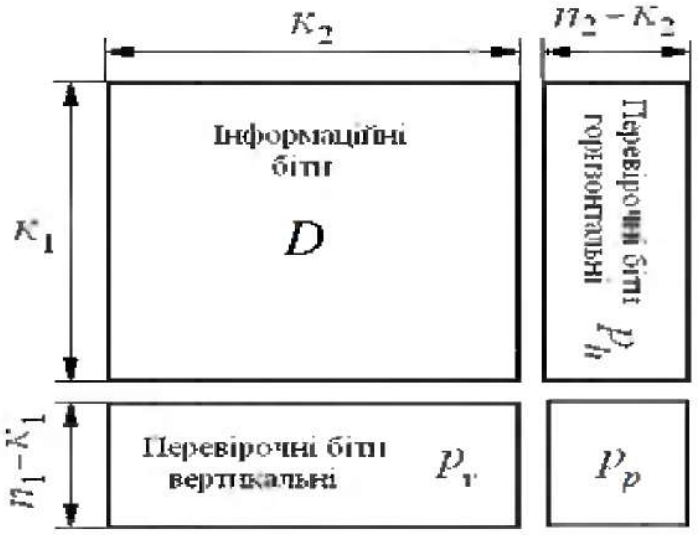

Рисунок 1.4 – Структура двовимірного ТРС:
D – інформаційні біти; p_h, p_v – перевірочні біти горизонтального та вертикального коду; p_p – блок перевірок від перевірочних бітів

ТРС 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. В 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 з м'яким входом і м'яким виходом (Soft-Input Soft-Output, SISO).

Ітеративний декодер двовимірного ТРС представляє собою послідовне з'єднання двох елементарних декодерів SISO (рисунок 1.5), які виконують декодування даних у протилежному до кодування порядку. Такі декодери ефективні, але мають високу складність реалізації. Також можна використовувати методи декодування ТРС з «жорсткими» рішеннями (Hard-Input Hard-Output, HIHO), але з втратою у ефективності декодування [22].

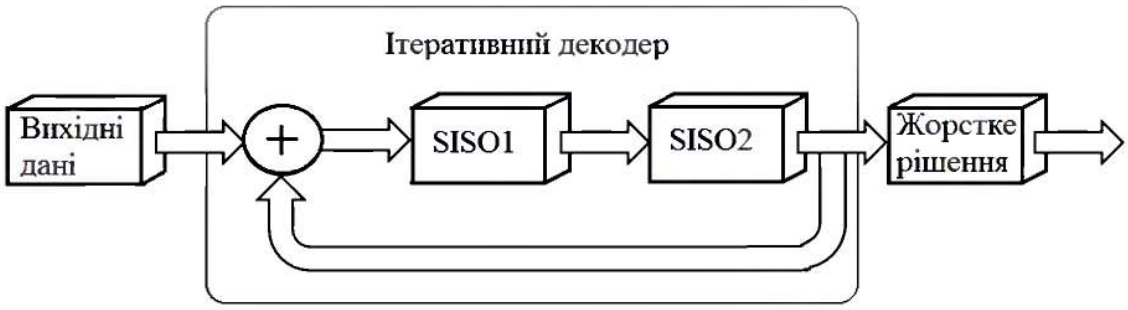

Рисунок 1.5 – Структура ітеративного ТРС декодера

Надійність м'яких рішень визначається їх абсолютним значенням, причому чим воно далі від нуля, тим вищою буде надійність рішення D_k. Відповідно знак дійсного числа a, що представляє м'яке рішення, показує, яке жорстке рішення. 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» (magic GENIE).

Турбокоди 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, SDR системах, мобільних (3G, 4G LTE, Inmarsat) і космічних (CCSDS) комунікаціях. Наприклад, дослідження NASA JPL (Mars Reconnaissance Orbiter + Mars Science Laboratory) дозволили отримати 30 Гб даних, включаючи високоякісні знімки і відеофайли поверхні Марсу за допомогою марсохода Curiosity та турбокода (швидкість $R = 1/6$, пакет з 8920 бітів) [18, 23-25].

Отже, ТРС є перспективним методом завадостійкого кодування, тому в даній роботі досліджуються методи та засоби кодування / декодування ТРС, причому увага акцентується саме на процедурах декодування, що пов'язано з описаними вище особливостями.

1.4 Висновки

У даному розділі показано, що завадостійкі коди є невід'ємною складовою сучасних систем зв'язку. Визначено, що при великому відношенні сигнал/шум застосовують ТРС, який дозволяє отримати дуже малі ймовірності помилки декодування при високій швидкості передавання даних. Розглянуто систему передавання цифрових даних, проаналізовано особливості її складових частин. Представлено структурні особливості кодера та декодера ТРС. Визначено, що найбільш складним компонентом є декодер, тому потрібно розробляти нові або модифікувати існуючі методи декодування, досягаючи компромісу між складністю та ефективністю декодування даних.

2 РОЗРОБКА МАТЕМАТИЧНОГО АПАРАТУ МЕТОДУ ДЕКОДУВАННЯ НА ОСНОВІ СПИСКУ КОДОВИХ СЛІВ

2.1 Особливості алгебри для декодування даних

Для декодування ТРС можна застосовувати НІНО методи: метод Редді-Робінсона [13, 25]; метод інверсії бінарного символу [14]. Більш ефективно застосовувати методи декодування ТРС, 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 ТРС:

1. Декодування за списком слів [11-13, 15, 22, 26];
2. Метод Хартмана-Назарова зі швидким перетворенням Адамара [13, 27].
3. Декодування на графах Таннера.
4. Декодування на основі принципів Вольфа [4, 8, 9, 15, 18, 28].

У даній роботі будемо розглядати ”м’які” методи декодування, математичний апарат яких грунтується на алгебрі логарифму відношення функцій правдоподібностей. Усі операції в ній виконуються над кінцевим полем $GF(2)$ з елементами {-1, +1}. Нехай +1 – це логічна одиниця, а -1 – логічний нуль. Операція додавання за модулем два (XOR) для роботи з цими значеннями подана в таблиці 2.1 [24].

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 імовірності може бути знайдена за формулою

$$\begin{aligned} p(D_1 \oplus D_2 = +1) &= p(D_1 = +1) \cdot p(D_2 = -1) + p(D_1 = -1) \cdot p(D_2 = +1) = \\ &= p(D_1 = +1) \cdot (1 - p(D_2 = +1)) + (1 - p(D_1 = +1)) \cdot p(D_2 = +1) = \\ &= p(D_1 = +1) - 2 \cdot p(D_1 = +1) \cdot p(D_2 = +1) + p(D_2 = +1). \end{aligned} \quad (2.1)$$

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

$$LLR(D_k) = \ln\left(\frac{p(D_k = +1)}{1 - p(D_k = +1)}\right) = \left(\frac{1 - p(D_k = -1)}{p(D_k = -1)}\right). \quad (2.2)$$

Тоді розв'язком рівняння (2.2) відносно $p(D_k = +1)$ буде вираз

$$p(D_k = +1) = \frac{\exp(LLR_{апр}(D_k))}{1 + \exp(LLR_{апр}(D_k))}. \quad (2.3)$$

Підставивши рівняння (2.3) до (2.1), можна отримати формулу

$$p(D_1 \oplus D_2 = +1) = \frac{\exp(LLR(D_1)) + \exp(LLR(D_2))}{(1 + \exp(LLR(D_1))) \cdot (1 + \exp(LLR(D_2)))}. \quad (2.4)$$

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

$$\begin{aligned} p(D_1 \oplus D_2 = -1) &= 1 - p(D_1 \oplus D_2 = +1) = \\ &= \frac{1 + \exp(LLR(D_1) + LLR(D_2))}{(1 + \exp(LLR(D_1))) \cdot (1 + \exp(LLR(D_2)))}. \end{aligned} \quad (2.5)$$

Застосувавши рівняння (2.2), замінивши $D_k = D_1 \oplus D_2$, а також виконавши підстановки (2.4) та (2.5), можна отримати наступний вираз

$$LLR(D_1 \oplus D_2) = \ln\left(\frac{p(D_1 \oplus D_2 = +1)}{p(D_1 \oplus D_2 = -1)}\right) = \ln\left(\frac{\exp(LLR(D_1)) + \exp(LLR(D_2))}{1 + \exp(LLR(D_1) + LLR(D_2))}\right). \quad (2.6)$$

Також можна використати операцію *box-plus* [25, **Ошибка! Источник ссылки не найден.**]

$$LLR(D_1)[+]LLR(D_2) = LLR(D_1 \oplus D_2). \quad (2.7)$$

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

$$p\left(D_k = +1 \mid x_\xi\right) \underset{H_2}{\overset{H_1}{\gtrless}} p\left(D_k = -1 \mid x_\xi\right). \quad (2.8)$$

Вибирається одна з гіпотез H_1 (D_k = +1), якщо апостеріорна ймовірність $p(D_k = +1|x_\xi) > p(D_k = -1|x_\xi)$, інакше приймаємо гіпотезу H_2 (D_k = -1).

За допомогою логарифмування обох частин формули (2.8) м'який вихід декодера буде визначатися наступним чином [9]

$$LLR_{MAP}(D_k) = \ln\frac{p(D_k = +1 \mid x_\xi)}{p(D_k = -1 \mid x_\xi)} \underset{H_2}{\overset{H_1}{\gtrless}} 0. \quad (2.9)$$

У загальному випадку на практиці для ТРС застосовується наближене обчислення $LLR_{MAP}(D_k)$.

2.2 Метод декодування на основі списку кодових слів

Розглянемо обчислення $LLR_{MAP}(D_k)$, засноване на використанні ефективного методу ALD [11-13, 15, 22], 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 [10] розроблено математичний апарат та декодер ТРС, який дозволяє оцінити надійність кожного бінарного символу після декодування Чейза [29], а також здійснити обмінний ітеративний процес.

Алгоритм методу включає наступні кроки [9-13, 20, 29-32]:

1. Нехай $r = (r_1, \ldots, r_n)$ 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).

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

$$h_{ij} = h_{0j}, \text{ якщо } j \notin \{P_k\}, \tag{2.10}$$

$$h_{iP_k} = h_{0P_k} \oplus Num\,(i, k) = h_{0P_k} \oplus e, \tag{2.11}$$

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

3. Декодувати слова h_0, ..., h_{2^t-1} за допомогою жорсткого декодера лінійного коду. Таким чином, можна одержати конкурентні кодові слова c_0, ..., c_{2^t-1}.

4. Обчислити метрики конкурентних слів M_i за допомогою лінійного представлення

$$M_i = \sum_{j=1}^{n} r_j(1 - 2c_{ij}), \tag{2.12}$$

або квадрата евклідової відстані

$$M_i = \sum_{j=1}^{n}(r_j - (2c_{i,j} - 1))^2. \tag{2.13}$$

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

$$M_{pp} = min\{M_i\}. \tag{2.14}$$

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

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

$$LLR(D_j) = \frac{1}{4} \cdot \left(\min\{M_i, c_{ij} \neq c_{pp,j}\} - M_{pp}\right) \cdot (2c_{pp,j} - 1). \tag{2.15}$$

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

$$LLR(D_j) = \beta \cdot (2c_{pp,j} - 1), \tag{2.16}$$

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

7. Обчислити значення зовнішньої інформації E з декодера для кожного біта j та оновити інформацію r на ітерації q за виразами

$$E(D_j) = LLR(D_j) - r_j. \quad (2.17)$$

$$w_j^q = r_j + \alpha \cdot E(D_j^q). \quad (2.18)$$

8. Виконати ітеративний процес декодування, який схематично зображено на рисунку 2.1, на якому декодери рядків та стовпчиків ТРС обмінюються зовнішньою інформацією про біти. У даній схемі q – номер ітерації декодування, а $\alpha = [0.0;\ 0.2; 0.3;\ 0.5;\ 0.7;\ 0.9;\ 1.0;\ 1.0;\ \ldots]$ – множник, який зменшує вплив зовнішньої інформації з декодера.

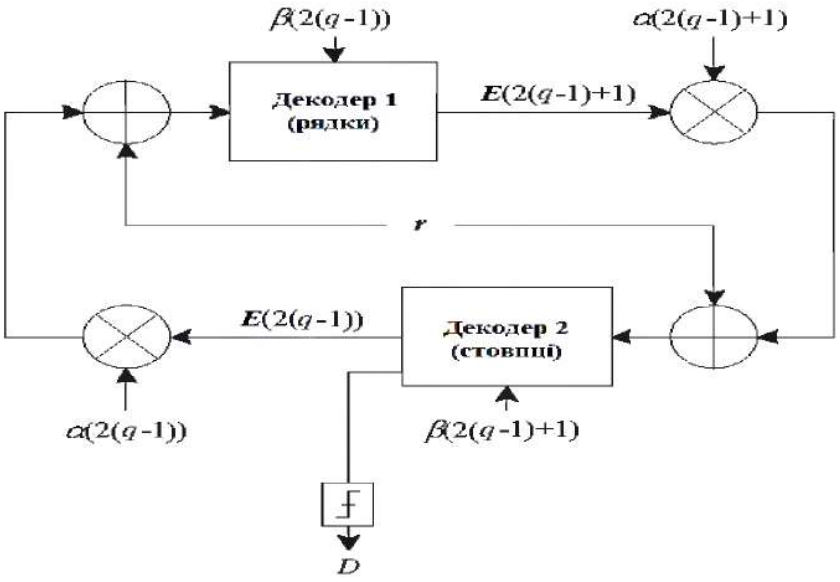

Рисунок 2.1 – Структурна схема ітеративного ТРС декодера

9. Використати критерій зупинки процесу декодування 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 ТРС, використавши метод ALD.

Розглянемо перший рядок: r = (0,5; 0,7; -0,9; 0,2; -0,3; 0,1; 0,6). Нехай кількість інвертованих позицій t у алгоритмі Чейза дорівнює 3 (t = 3), кількість ітерацій q = 1, множники $\alpha_1 = 0$, $\beta_1 = 0{,}2$. Вибираємо найменш надійні символи $P_1 = 6$, $P_2 = 4$, $P_3 = 5$. Знайдемо жорстке рішення D = (1; 1; 0; 1; 0; 1; 1).

Формуємо вектори помилок з трьох символів. Формуємо вектор «жорстких» можливих рішень (таблиця 2.2), враховуючи вектор помилок e. Підкреслені біти відповідають інверсіям, які виникли після застосування алгоритму Чейза. Декодуємо код Хеммінга "жорстким" декодером (таблиця 2.2). Біти із зірочкою у стовпчику кодових слів c_i відповідають позиціям, скоригованим декодером у слові h_i. Обчислюємо метрики конкурентних слів (таблиця 2.3), використовуючи формулу (2.12). Тоді індекс $pp = 1$, а $M_{pp} = 2{,}7$, а відповідне декодоване слово $D = (1; 1; 0; 0; 0; 0; 1)$, причому його можна зустріти декілька разів у списку конкурентних слів.

Таблиця 2.2 – Результат декодування коду Хеммінга

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

Таблиця 2.3 – Метрики конкурентних слів

I	c_i	M_i
0	(1; 1; 0; 1; 0; 1; 0*)	(0,5)(-1)+(0,7)(-1)+(-0,9)(1)+(0,2)(-1)+(-0,3)(1)+(0,1)(-1)+(0,6)(1)=-2,1
1	(1; 1; 0; 0*; 0; 0; 1)	(0,5)(-1)+(0,7)(-1)+(-0,9)(1)+(0,2)(1)+(-0,3)(1)+(0,1)(1)+(0,6)(-1)=-2,7
2	(1; 1; 1*; 1; 1; 1; 1)	(0,5)(-1)+(0,7)(-1)+(-0,9)(-1)+(0,2)(-1)+(-0,3)(-1)+(0,1)(-1)+(0,6)(-1)=-0,9
3	(0*; 1; 0; 1; 1; 0; 1)	(0,5)(1)+(0,7)(-1)+(-0,9)(1)+(0,2)(-1)+(-0,3)(-1)+(0,1)(1)+(0,6)(-1)=-1,5
4	(1; 1; 0; 0; 0; 0*; 1)	(0,5)(-1)+(0,7)(-1)+(-0,9)(1)+(0,2)(1)+(-0,3)(1)+(0,1)(1)+(0,6)(-1)=-2,7
5	(1; 1; 0; 0; 0; 0; 1)	(0,5)(-1)+(0,7)(-1)+(-0,9)(1)+(0,2)(1)+(-0,3)(1)+(0,1)(1)+(0,6)(-1)=-2,7
6	(1; 0*; 0; 0; 1; 1; 1)	(0,5)(-1)+(0,7)(1)+(-0,9)(1)+(0,2)(1)+(-0,3)(-1)+(0,1)(-1)+(0,6)(-1)=-0,9

7	(1; 1; 0; 0; 0*; 0; 1)	(0,5)(-1)+(0,7)(-1)+(-0,9)(1)+(0,2)(1)+(-0,3)(1)+(0,1)(1)+(0,6)(-1)=-2,7

Розраховуємо м'які рішення з виходу декодера для кожного біта j послідовності D (таблиця 2.4) Обчислюємо значення зовнішньої інформації з декодера для кожного біта j (таблиця 2.5). Збільшуємо лічильник ітерацій $q = 2$ та оновлюємо показники $\alpha_2 = 0,2$, $\beta_2 = 0,4$.

Таблиця 2.4 – М'які рішення

j	$LLR(D_j)$	D_j
1	(+1)·(-1,5 - (-2,7)) / 4 = +0,30	1
2	(+1)·(-0,9 - (-2,7)) / 4 = +0,45	1
3	(-1)·(-0,9 - (-2,7)) / 4 = -0,45	0
4	(-1)·(-2,1 - (-2,7)) / 4 = -0,15	0
5	(-1)·(-1,5 - (-2,7)) / 4 = -0,30	0
6	(-1)·(-2,1 - (-2,7)) / 4 = -0,15	0
7	(+1)·(-2,1 - (-2,7)) / 4 = +0,15	1

Таблиця 2.5 – 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 r (таблиця 2.6).

Повторити дану процедуру для всіх рядків. Виконати декодування стовпців, використавши значення w, отримане після декодування рядків. Потім

знову перейти до рядків. Таким чином, інформація обертається петлею між двома компонентними декодерами, що зображено на рисунку 2.1. Виконати $q \leq v$ ітерацій та розрахувати остаточні "жорсткі" рішення для символів D_j.

Таблиця 2.6 – Оновлені значення отриманої послідовності символів

j	w
1	+0,5 + 0,2·(-0,20) = +0,46
2	+0,7 + 0,2·(-0,25) = +0,65
3	-0,9 + 0,2·(+0,45) = -0,81
4	+0,2 + 0,2·(-0,35) = +0,13
5	-0,3 + 0,2·(+0,00) = -0,30
6	+0,1 + 0,2·(-0,25) = +0,05
7	+0,6 + 0,2·(-0,45) = +0,51

2.3 Висновки

У даному розділі розглянуто методи декодування ТРС. Визначено, що декодування з м'якими рішеннями за списком кодових слів є досить ефективним для турбо-кодів-добутків. Розглянуто математичний апарат даного методу, представлено ітеративний ТРС декодер та наведено приклад його роботи. У наступному розділі необхідно дослідити ефективність роботи розробленого декодера у складі системи цифрового зв'язку.

3 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ТУРБО-ДЕКОДЕРА

3.1 Обгрунтування середовища розробки

Система MatLab складається з мови M-language; середовища оброблення даних; керованої графіки; бібліотеки математичних функцій; програмного інтерфейсу користувача. Мова програмування MatLab є досить простою та практичною, незначна кількість операторів компенсується великою кількістю процедур і функцій, які є зрозумілими для інженерів [33]. MatLab представляє собою спеціально створений пакет прикладних програм для проведення інженерних розрахунків. Він дає можливість зменшити час, який затрачується на розв'язання різноманітних наукових і прикладних задач, і значно спростити розробку нових алгоритмів у таких областях як: моделювання об'єктів та систем управління, проектування комунікаційних систем, оброблення сигналів і зображень, вимірювання сигналів, обчислювальна біологія, математична фізика тощо.

Враховуючи дані особливості, для моделювання роботи системи цифрового зв'язку з використанням ТРС кодів та методу декодування ALD було використано систему MatLab.

3.2 Експериментальні дослідження

Ефективність завадостійких кодів покращується завдяки збільшенню довжини кодованої послідовності символів n, яка в свою чергу, залежить від кількості інформаційних символів k. Для великих значень k буде зростати складність декодування. Це означає, що потрібно побудувати довгий код із набагато коротших кодових компонентів, які можна легко декодувати.

Для ТРС з довжиною k отримати такий код можна шляхом об'єднання кодів Хеммінга або Боуза-Чоудхурі-Хоквінгема (БЧХ) [34]. Останні є

узагальненням кодів Хеммінга і представляють великий клас кодів із заздалегідь визначеними довжиною блока n, швидкістю коду R і коригувальними властивостями, а саме, мінімальною кодовою відстанню d_{min}, що дозволяє виправляти кратні помилки. 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 $i < 2^{h-1}$ існує бінарний БЧХ код з параметрами, які наведено у таблиці 3.1 [9].

Таблиця 3.1 – Параметри бінарного коду БЧХ

Параметри коду	Розрахункова формула
Довжина блоку, n	$n = 2^h - 1$
Кількість перевірочних бітів, u	$u = n - k \leq hi$
Мінімальна кодова відстань, d_{min}	$d_{min} \geq 2i + 1$

Виконаємо дослідження ефективності роботи системи цифрового зв'язку з ТРС та знайдемо експериментальну функціональну залежність частоти виникнення бітових помилок BER від нормованого співвідношення сигнал/шум E_b/N_0 ($дБ$). Застосовано таку модель: пакети даних з виходу генератора псевдовипадкової послідовності, ТРС код, компонентні БЧХ-кодери з параметрами коду (n, k, d_{min}) = (7, 4, 3) та (15, 11, 3) з швидкостями $R = (R_{комп})^2 \approx 0{,}327$ та 0,538, метод декодування ALD з кількістю ненадійних бітів $t = 4$ та 10, максимум 8 ітерацій декодування, *BPSK*-модем, канал з *AWGN* [36]. Моделювання проводилося від 0 до 8 $дБ$ з кроком 1 $дБ$. Критерієм закінчення ітеративного процесу для кожного значення E_b/N_0 є час (не більше 40 хв). Результати експерименту показані на рисунках 3.1 та 3.2.

Експериментальна залежність BER = f (E_b/N_0) для системи з ітеративним декодером ТРС $(7, 4)^2$ при $t = 4$ наведена на рисунку 3.1 показує, що для перших 4-ох ітерацій BER гірша, ніж для BPSK, але при подальшому збільшенні ітерацій BER зменшується. Наприклад, на показнику BER = 10^{-3} ітеративний

декодер TPC(7, 4)^2 буде вигравати приблизно 1 *дБ* у кривої BPSK (5,6 *дБ* на 8 ітераціях проти 6,6 *дБ*).

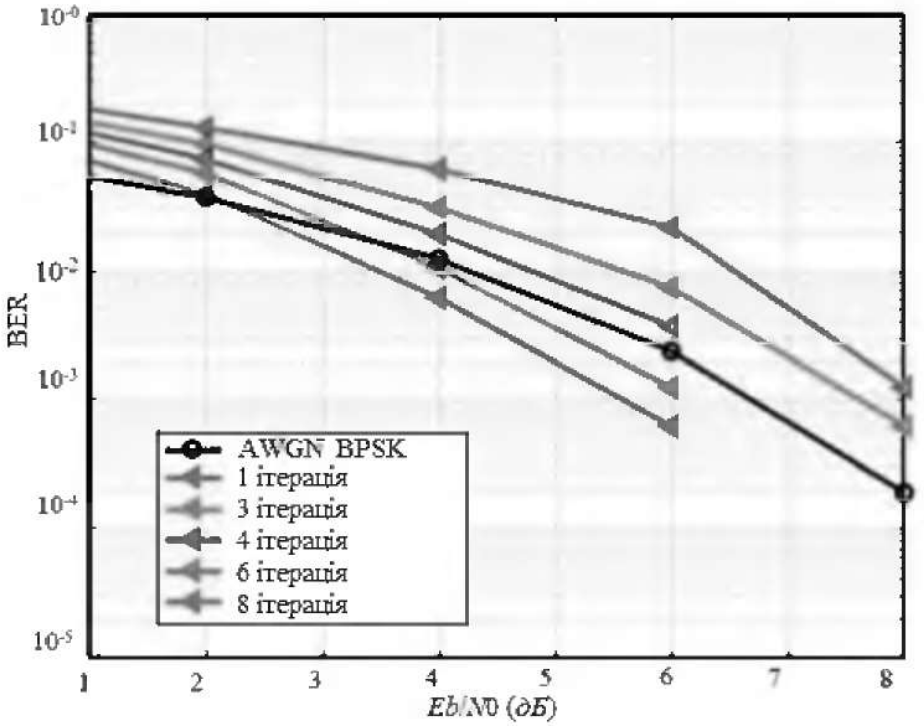

Рисунок 3.1 – Залежність BER = f (E_b/N_0) для TPC (7, 4)^2, t = 4

Аналогічна експериментальна залежність BER = f (E_b/N_0) для системи з ітеративним декодером TPC (15, 11)^2 при t = 10 представлена на рисунку 3.2 показує, що криві BER на 6-ій та 8-ій ітераціях майже однакові, тобто декодер асимптотично збігається. Це означає, що на 8-ій ітерації BER = 10^{-5} можна отримати при 6 *дБ*. Порівнюючи коди TPC (7, 4)^2 та TPC(15, 11)^2 отримаємо, що для BER = 10^{-3} енергетичний виграш при використанні останнього кода становить приблизно 1,6 *дБ* (5,6 *дБ* проти 4 *дБ*). Взагалі використання TPC(15, 11)^2 замість TPC (7, 4)^2 в середньому дозволяє отримати енергетичний виграш близько 1,6 *дБ*. Таким чином, використання більших показників k та t вигідно.

Виконаємо передавання кольорового зображення (1553 фрейма по 2025 інформаційних бітів та 1122 додаткових біта для останнього пакета) з

використанням ТРС(63, 45, 7)^2 зі швидкістю $R \approx 0{,}51$ (компонентний код БЧХ) на показнику $E_b/N_0 = 4$ *дБ* з використанням методу декодування ALD ($t = 10$).

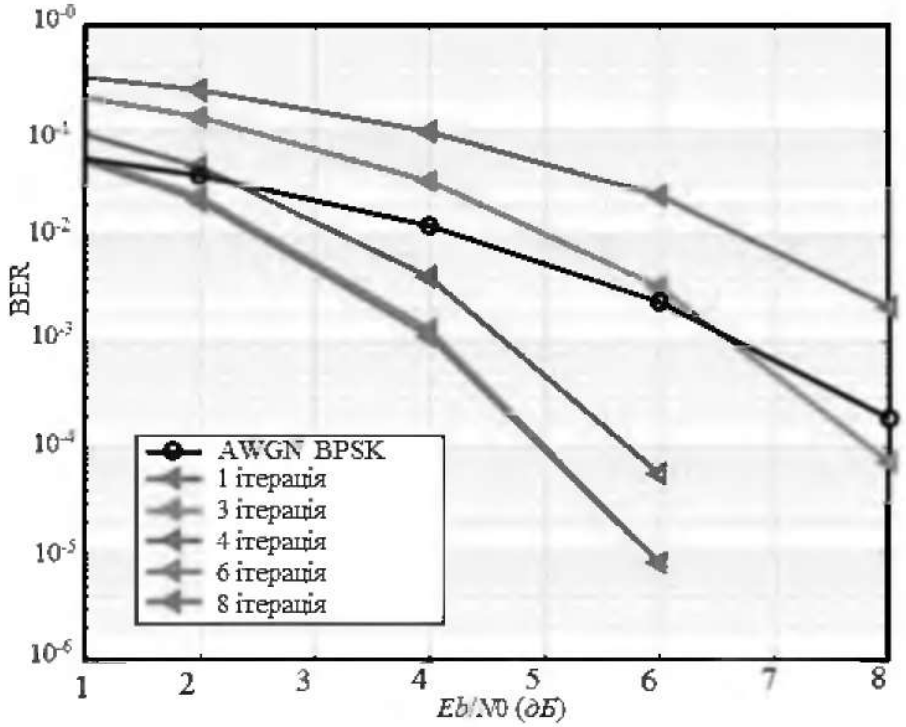

Рисунок 3.2 – Залежність BER = $f(E_b/N_0)$ для ТРС $(15, 11)^2$, $t = 10$

Результати роботи показані на рисунку 3.3 на 0 (без кодування), 2, 4 та 8 ітераціях відповідно, перше число знизу вказує кількість помилок у бітах, а число в дужках – у символах. Чисельні показники наведені в таблиці 3.1, де *BEN* / *SEN* – кількість бітових та символьних помилок, а *BER* / *SER* – коефіцієнт бітових та символьних помилок.

Рисунок 3.2 – Результати передавання мультимедійних даних

Таблиця 3.2 – Чисельні показники

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	$6{,}006 \cdot 10^{-3}$	$6{,}924 \cdot 10^{-2}$
	4	60	52	$1{,}907 \cdot 10^{-5}$	$2{,}645 \cdot 10^{-4}$
	8	0	0	0	0

Можна побачити, що якість переданого зображення покращується, а кількість помилок зменшується.

3.3 Висновки

У даному розділі знайдено експериментальну функціональну залежність частоти виникнення бітових помилок від нормованого співвідношення сигнал/шум для турбо-кодів-добутків з методом декодування на основі списку кодових слів. Енергетичний виграш при використанні ТРС $(15, 11)^2$ замість ТРС $(7, 4)^2$ становить приблизно 1,6-2 *дБ*, а BER = 10^{-5} можна отримати при 6 *дБ*. Результати експериментів на мультимедійних даних підтверджують, що метод декодування ALD досить ефективний та дозволяє досягти хорошої коригуючої здатності.

ВИСНОВКИ

У даній роботі визначено, що застосування завадостійких кодів дає можливість отримати енергетичний виграш та зменшити частоту виникнення помилок. При великому відношенні сигнал/шум у системах передавання цифрових даних застосовують ТРС, який працює на високих швидкостях та дозволяє отримати дуже малі ймовірності помилки декодування. Найскладнішим компонентом кодека такої системи є ТРС декодер.

У результаті проведеного аналізу визначено, що метод декодування на основі списку кодових слів є ефективним і найбільш поширеним на практиці. Тому у роботі проаналізовано та формалізовано математичний апарат даного методу декодування. Представлено структурну схему ітеративного ТРС декодера та наведено приклад його роботи.

Розроблено модель передавання даних, реалізовано алгоритмічні та програмні засоби за допомогою яких досліджено ефективність роботи системи цифрового зв'язку. Знайдено експериментальні функціональні залежності частоти виникнення бітових помилок від нормованого співвідношення сигнал/шум для ТРС$(7, 4)^2$ при $t = 4$ та ТРС$(15, 11)^2$ при $t = 10$ з компонентним БЧХ кодом. Енергетичний виграш на BER = 10^{-3} при використанні ТРС$(15, 11)^2$ замість ТРС$(7, 4)^2$ становить приблизно 1,6-2 *дБ*, причому BER = 10^{-5} можна отримати при 6 *дБ*. Ітеративний ТРС$(7, 4)^2$ декодер на показнику BER = 10^{-3} виграє 1 *дБ* у кривої BPSK.

Експериментальні дослідження із передаванням мультимедійних даних підтверджують, що метод декодування на основі списку кодових слів можна застосовувати на практиці.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

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

2. Иванов Ю.Ю. О некоторых аспектах итеративной стратегии декодирования турбо-кодов: ретроспектива и «турбо»-принцип: матер. IV международной научно-практической конференции «Информационные технологии и компьютерная инженерия» / Ю.Ю. Иванов, А.Я. Кулик. – Винница: ВНТУ, 28–30 мая, 2014 г. – С. 157-160.

3. Варгаузин В.А. Вблизи границы Шеннона / В.А. Варгаузин // ТелеМультиМедиа. – 2005. – № 3 (31). – С. 3-10.

4. Королев А.И. Коды и устройства помехоустойчивого кодирования информации / А.И. Королев. – Минск, 2002. – 286 с.

5. Владимиров С.С. Математические основы теории помехоустойчивого кодирования / С.С. Владимиров. – СПб: СПбГУТ, 2016. – 91 с.

6. Іванов Ю.Ю. Експериментальне дослідження завадостійкості турбо-кодів: числові оцінки та імітаційне моделювання нового субоптимального алгоритма PL-log-MAP / Ю.Ю. Іванов // Вісник Вінницького політехнічного інституту. – Вінниця: ВНТУ, 2016. – № 5. – С. 76-84.

7. Berrou C. Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes / C. Berrou, A. Glavieux, P. Thitimajshima // Proceedings of the ICC'93. – Switzerland: 1993. – P. 1064-1070.

8. Виноградов В.С. Использование турбо-кодека для безопасной передачи данных / В.С. Виноградов, В.В. Коробицын, М.Н. Московцев // Математические структуры и моделирование. – 2015. – № 4(36). – С. 134-144.

9. Pyndiah R. Near Optimum Decoding of Product Codes / R. Pyndiah, A. Glavieux, A. Picart, S. Jacq // In IEEE Global Telecommunications Conference GLOBECOM '94. – 1994. – P. 339-343.

10. Pyndiah R.M. Near-Optimum Decoding of Product Codes: Block Turbo Codes / R. Pyndiah // IEEE Transactions on Communications. – 1998. – V. 46. – P. 1003-1010.

11. A. Al-Dweik A. Turbo Product Codes: Applications, Challenges, and Future Directions / A. Al-Dweik, H. Mukhtar, A. Shami // IEEE Communications Surveys & Tutorials. – 2016. – V. 18. – № 4. – P. 3052-3068.

12. Ghnimat M.G. Iterative Decoding of Turbo Product Codes (TPCs) Using the Chase-Pyndiah Turbo Decoder / M. G. Ghnimat. – North Cyprus, 2017. – 63 p.

13. Архипкин А. Турбо-коды – мощные алгоритмы для современных систем связи / А. Архипкин // Беспроводные технологии. – 2006. – № 1. – С. 63-64.

14. Зубарев Ю.Б. Помехоустойчивое кодирование в цифровых системах передачи данных [Электронный ресурс] / Ю.Б Зубарев, Г.В. Овечкин. – Режим доступа: http://www.mtdbest.iki.rssi.ru/pdf/ obzor_dvoichnie_kodi_2.pdf.

15. Методи декодування блокових і згорткових турбо-кодів [Електронний ресурс]: матер. XLIX науково-технічної конференції факультету комп'ютерних систем і автоматики / С. В. Нікіфоренко, О. М. Солонина, С. Я. Середа, Ю. Ю. Іванов. – Вінниця: ВНТУ, 11-13 березня, 2020. – Режим доступу: https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2020/paper/view/8723/7858.

16. Codes and Turbo Codes: edited by C. Berrou / C. Douillard, M. Jezequel, G. Battail and others. – Paris: Springer, 2010. – 424 p.

17. Morelos-Zaragoza R. The Art of Error Correction Coding / R. Morelos-Zaragoza. – Chippenham: John Wiley & Sons, 2006. – P. 143-168.

18. Johnson S.J. Iterative Error Correction. Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes / S.J. Johnson. – Cambridge, 2009. – 356 p.

19. Channel Coding in Communication Networks. From Theory to Turbocodes / C. Berrou, C. Douillard, M. Jezequel, A. Picart. – Chippenham: ISTE, 2007. – 437 p.

20. Hanzo L. Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Wireless Channels / L. Hanzo, T.H. Liew, – Southampton: Department of Electronics and Computer Science of UK, 2002. – 746 p.

21. Soleymani M.R. Turbo Coding for Satellite and Wireless Communications / M.R. Soleymani, Y. Gao, U. Vilaipornsawai. – USA (New York), 2002. – 231 p.

22. Варгаузин В.А. Турбо-коды и итеративное декодирование: принципы, свойства, применение / В.А. Варгаузин, Л.Н. Протопопов // ТелеМультиМедиа. – 2000. № 4. – С. 33-38.

23. Буданов А.В. Декодирование блочных турбокодов / А.В. Буданов, И.А. Небаев. – СПб: СПбГУТ, 2013. – 26 с.

24. Королев А.И. Турбо-коды и итеративное декодирование / А.И. Королев, В.К. Конопелько. – Минск: БГУИР, 2015. – 74 с.

25. Hagenauer J. Iterative Decoding of Binary Block and Convolutional Codes / J. Hagenauer, E. Offer, L. Papke // IEEE Transactions on Information Theory. – 1996. – V. 42. – № 2. – P. 429-445.

26. Reddy S.M. Random Error and Burst Corrections by Iterated Codes / S.M. Reddy, J.P. Robinson // IEEE Transactions on Information Theory. – 1972. – №. 18(1). – P. 182-185.

27. Nazarov L.E. Use of Fast Walsh-Hadamard Transformation for Optimal Symbol-by-Symbol Binary Block-Code Decoding / L.E. Nazarov, V.M. Smolyaninov // Electronics Letters. – 1998. – №. 34. – P. 261-262.

28. Wolf J. Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis / J. Wolf // IEEE Transactions on Information Theory. – 1978. – V. 24. – P. 76-80.

29. Chase D. Class of Algorithms for Decoding Block Codes with Channel Measurement Information / D. Chase // IEEE Transactions on Information Theory. – 1972. – V. 18. – № 1. – P. 170-182.

30. Cuevas J. Information Theory. Turbo Decoding of Product Codes for Gigabit per Second Applications and Beyond / J. Cuevas, P. Adde, S. Kerouedan // European Transactions on Telecommunications. – № 17 (1). – 2006. – P. 345-355.

31. Jego C. Turbo Decoding of Product Codes based on the Modified Adaptive Belief Propagation Algorithm / C. Jego, W.J. Gross // IEEE International Symposium on Information Theory ISIT 2007. – France (Nice), 2007. – P. 641-644.

32. Goalic A. Real-time Turbo Decoding of Block Turbo Codes Using the Hartmann-Nazarov Algorithm on the DSP Texas tms320c6201 / A. Goalic, K. Cavalec-Amis, V. Kerbaol // In Proceedings of International Conference on Communications (ICC'2002). – New-York, 2002. – V. 3. – P. 1716-1720.

33. Simulating the Operation of Turbo Codes through the Monte Carlo Method, Comparison between MATLAB, C and C# / H. Balta, A. Isar, D. Isar, M. Balta // Advances in Intelligent and Soft Computing. – Springer Berlin Heidelberg, 2012. – V. 133. – P. 1127-1139.

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

ДОДАТКИ

					08-02.БДР.008.00.000 СП		
					Дослідження методу турбо-декодування на основі списку кодових слів.	Літ.	Маса	Масштаб
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Нікіфоренко С.В.					
Перевір.		Іванов Ю.Ю.			*Схема програми*	Арк.	Аркушів 1
Т. Контр.		Іванов Ю.Ю.					
Реценз.						ВНТУ, гр. 1СІ-16б	
Н. Контр.		Іванов Ю.Ю.					
Затверд.		Квєтний Р.Н.					

Додаток А

(обов’язковий)

Схема програми

Рисунок А.1 – Схема програми

					08-02.БДР.008.00.000 ПД		
					Дослідження методу турбо-декодування на основі списку кодових слів.	*Літ.*	*Маса*	*Масштаб*
Змн.	*Арк.*	*№ докум.*	*Підпис*	*Дата*				
Розроб.		*Нікіфоренко С.В.*					
Перевір.		*Іванов Ю.Ю.*			*Модель системи передавання даних*	*Арк.*	*Аркушів* 1
Т. Контр.		*Іванов Ю.Ю.*					
Реценз.						*ВНТУ, гр. 1СІ-16б*	
Н. Контр.		*Іванов Ю.Ю.*					
Затверд.		*Квєтний Р.Н.*					

Додаток Б

(обов'язковий)

Модель системи передавання даних

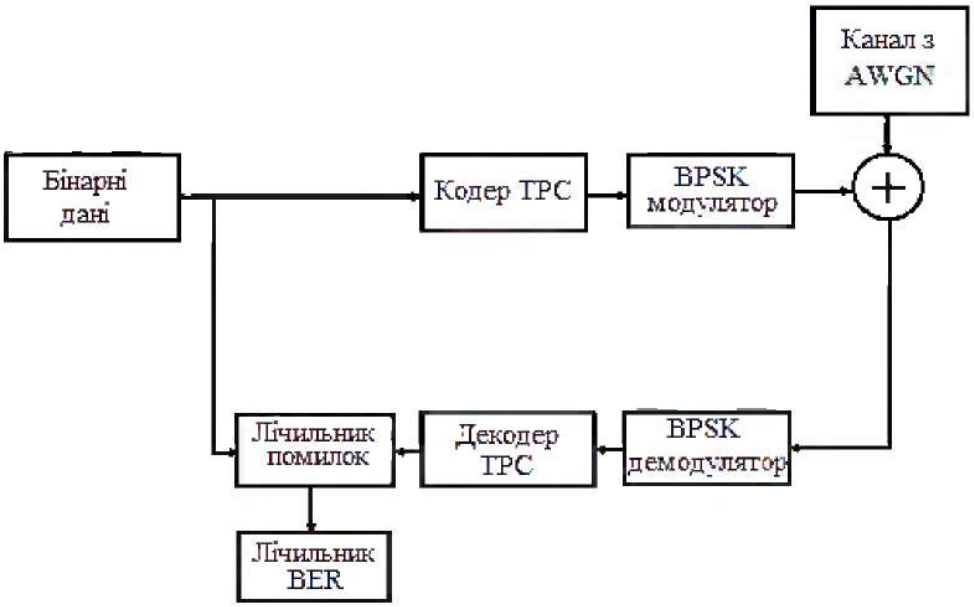

Рисунок Б.1 – Модель системи передавання даних

					08-02.БДР.008.00.000 ПД		
					Дослідження методу турбо-декодування на основі списку кодових слів.	*Літ.*	*Маса*	*Масштаб*
Змн.	*Арк.*	*№ докум.*	*Підпис*	*Дата*				
Розроб.		*Нікіфоренко С.В.*			*Експериментальні криві*	*Арк.*	*Аркушів* 1
Перевір.		*Іванов Ю.Ю.*					
Т. Контр.		*Іванов Ю.Ю.*				*ВНТУ, гр. 1СІ-16б*	
Реценз.							
Н. Контр.		*Іванов Ю.Ю.*					
Затверд.		*Квєтний Р.Н.*					

Додаток В

(обов'язковий)

Експериментальні криві

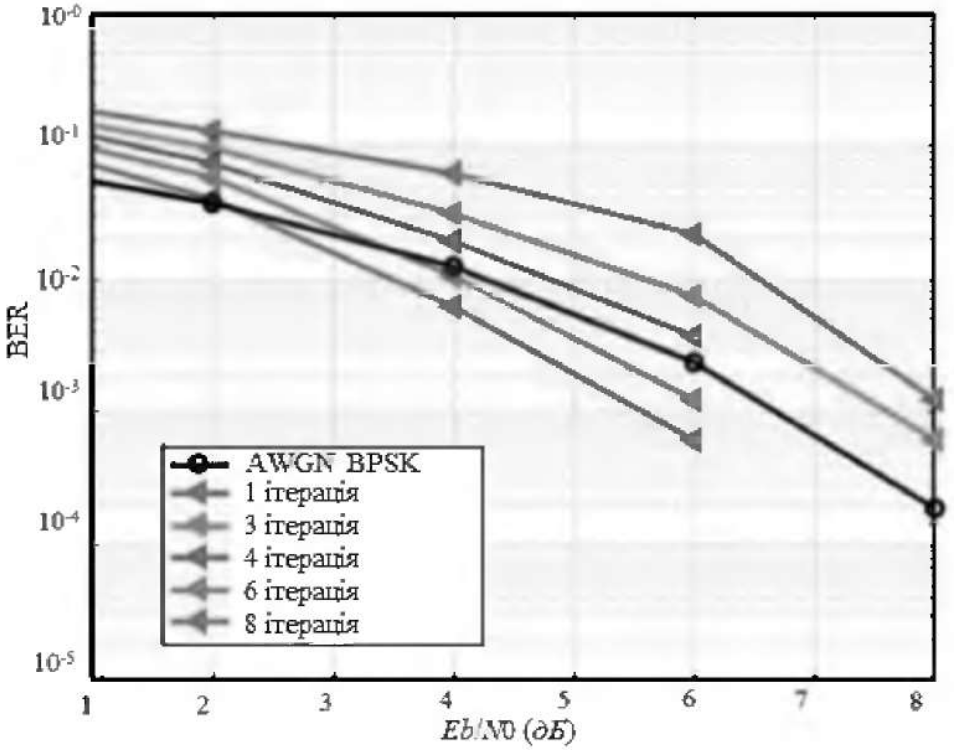

Рисунок В.1 – Залежність BER = f (E_b/N_0):

ТРС$(7, 4)^2$, кількість ненадійних символів $t = 4$

Додаток Г

(обов'язковий)

Лістинг програми

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Main program
clear all
close all
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 in dB
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;
  repetition = 1000;
   for jj = 1:repetition

  [R,msg,v] = received_signal(SNR(i));
  % initializing W:
  W = zeros(15,15);
  m = 1; % iteration-1
  R1 = R+W*alpha(2*(m-1)+1); % Input of row decoding for 1st iteration
    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); %Input of column decoding for 1st iteration
    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;

  m = 2;
  R1 = R+W*alpha(2*(m-1)+1); % Input of row decoding for 2nd iteration
```

```
  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); %Input of column decoding for 2nd iteration
  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); % Input of row decoding for 2nd iteration
  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); %Input of column decoding for 2nd iteration
  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); % Input of row decoding for 2nd iteration
  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); %Input of column decoding for 2nd iteration
  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)); %Theoretical uncoded BPSK AWGN
semilogy(SNRdB,theoretical_awgn,'ko-','LineWidth',2);
grid on;
legend('1st iteration','2nd iteration','3rd iteration','4th iteration', 'AWGN Uncoded BPSK');
xlabel('SNR(dB)');
ylabel('Bit Error Rate');
axis([1 8 10^-6 1])
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Algorithm
function [W,D] = half_iteration(R_row,beta)
  [parmat00,genmat,n,k] = hammgen(4);
  p = 10;
  % step1
  R_abs = abs(R_row);
  Y = (R_row>0);
  % step2
  [AA,index] = sort(R_abs);
  Pos = index(1:p);
  % step3
  Tq = calc_test_patterns(R_row,p,Pos);
  % step4
  Zq = gen_perturbed_sequence(Y,Tq);
  % step5
  [C,Si] = calc_SyndromesandCodewords(Zq);
  % step6
  [M,Mpp,D,loc] = calc_metrics(R_row,C);
  % step7
  F = find_rel(M,C,D,Mpp,beta);
  % step8
  W = calc_ExtrinsicInfo_W(D,F,R_row);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```