

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет програмування та комп'ютерних і телекомунікаційних систем

Кафедра телекомунікацій та радіотехніки

**ДИПЛОМНА РОБОТА**

Другий ( Магістерський)

Освітній рівень

Галузь знань 17 Електроніка та телекомунікації

Шифр і назва спеціальності

Спеціальність 172 Телекомунікації та радіотехніка

Шифр і назва спеціальності

на тему Модель цифрової системи зв'язку з завадостійким згортковим кодуванням

ДРТР.2017035.01.07.ПЗ

Виконав: студент 2 курсу, група ТР<sub>м</sub>-19-1

  
підпис

Т.С. Біндер  
Ініціали, прізвище

Керівник: канд. техн. наук, доц.

  
підпис

І.С. Пятін  
Ініціали, прізвище

До захисту допускаю:

Зав. кафедри: д-р техн. наук, проф.

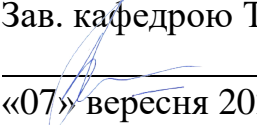
  
підпис

Ю.М. Бойко  
Ініціали, прізвище

05 грудня 2020 р.

Хмельницький, 2020  
Хмельницький національний університет

Факультет програмування та комп'ютерних і телекомунікаційних систем  
Кафедра телекомунікацій та радіотехніки  
Освітній рівень другий (магістерський)  
Галузь знань 17 – Електроніка та телекомунікації  
Спеціальність 172 – Телекомунікації та радіотехніка  
Освітня-професійна програма Телекомунікації та радіотехніка

ЗАТВЕРДЖУЮ  
Зав. кафедрою ТР  
 Ю.М. Бойко  
«07» вересня 2020р.

## ЗАВДАННЯ НА ДИПЛОМНУ РОБОТУ

Біндер Тарас Сергійович

1 Тема роботи: Модель цифрової системи зв'язку з завадостійким згортковим кодуванням

керівник роботи Пятін Ілля Станіславович, к.т.н, доцент

Затверджено наказом по університету від «1» вересня 2020р. № 118.

2 Строк подання студентом роботи на кафедрі: 30.11.2020р.

3 Вихідні дані (характеристика об'єкта, умов дослідження та ін.)

Мета роботи: дослідження цифрових систем зв'язку з згортковими кодами в середовищі Matlab.

Об'єкт дослідження: процес передачі і приймання, сигналів з згортковими кодами у цифрових системах зв'язку.

Предмет дослідження: методи та засоби згорткового кодування і декодування у цифрових системах зв'язку.

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити) Завадостійке кодування. Декодування з м'яким рішенням. Коди що ітеративно декодуються. Дослідження моделей систем зв'язку з згортковими кодами

Завдання отримав 

Науковий керівник 



## ЗМІСТ

Вступ.....	
1 Завадостійке кодування.....	
1.1 Визначення завадостійкого кодування .....	
1.2 Класифікація методів кодування.....	
1.3 Економне кодування.....	
1.4 Код Хемінга.....	
1.5 Згорткові коди.....	
1.6 Особливості кодування і декодування згорткових кодів.....	
Висновки до першого розділу.....	
2 Декодування з м'яким рішенням .....	
2.1 Передача двійкових сигналів по каналам зв'язку .....	
2.2 Алгоритм Вітербі .....	
2.3 Алгоритм Вітербі з Евклідовою метрикою.....	
2.4 Алгоритм Чейза.....	
2.5 Декодування за упорядкованими статистиками.....	
2.6 Декодування за мінімумом узагальненої відстані.....	
2.7 Спискове декодування.....	
2.8 Алгоритм Вітербі з м'яким виходом.....	
2.9 Алгоритм декодування за максимумом апостеріорної ймовірності.....	
Висновки до другого розділу.....	
3 Коди що ітеративно декодуються.....	
3.1 Поняття ітеративного декодування.....	
3.2 Каскадні коди.....	
3.3 Ітеративне декодування паралельного каскадного коду.....	
3.4 Використання пристроїв перемішування.....	
3.5 Каскадні коди послідовного типу.....	
3.6 Ітеративне декодування послідовних схем каскадного кодування.....	
3.7 Добуток блокових кодів.....	

Висновки до третього розділу.....	
4 Дослідження моделей систем зв'язку з згортковими кодами .....	
4.1 Simulink-модель системи зв'язку з послідовним каскадним кодуванням....	
4.2 Згорткові коди .....	
4.3 Коди Ріда-Соломона.....	
4.4 Порівняння коду Ріда-Соломона і згорткового коду.....	
Висновки до четвертого розділу.....	
Висновки.....	
Перелік джерел посилання.....	
Додаток А. Презентація.....	
Додаток Б. Апробація роботи.....	

## ВСТУП

### Актуальність теми

Завадостійким називається кодування, при якому здійснюється виявлення та виправлення помилок в прийнятих кодових комбінаціях. Можливість завадостійкого кодування здійснюється теоремою Шеннона, згідно з якою: якщо продуктивність джерела менше пропускної здатності каналу зв'язку, то існує принаймні одна процедура кодування і декодування при якій ймовірність помилкового декодування як завгодно мала, якщо ж продуктивність джерела більше пропускної здатності каналу, то такої процедури не існує [1].

Завадостійке кодування відбувається додаванням під час передачі в корисні дані спеціальним чином структурованої надлишкової інформації (наприклад, контрольного числа), а при прийомі використання цієї надмірної інформації для виявлення і виправлення помилки. Число помилок, яке можна виправити, обмежена і залежить від конкретного застосовуваного коду.

Коди виявлення помилок (які можуть тільки встановити факт помилки) належать до тих же класів кодів, що і коди, що виправляють помилки. Фактично будь-який код, що виправляє помилки, може бути також використаний для виявлення помилок (при цьому він буде здатний виявити більше число помилок, ніж був здатний виправити). Коди, що виправляють помилки, застосовуються в системах цифрового зв'язку, в тому числі: супутникової, радіорелейної, стільникової, передачі даних по телефонних каналах, а також в системах зберігання інформації, в тому числі магнітних і оптичних. Коди, що виявляють помилки, застосовуються в мережевих протоколах різних рівнів.

За способом роботи з даними коди, що виправляють помилки, діляться на блокові, що ділять інформацію на фрагменти постійної довжини і обробляють кожен з них окремо, і згорткові, що працюють з даними як з безперервним потоком.

Згорткові коди засновані на перетворенні вхідної послідовності двійкових символів в вихідну послідовність двійкових символів, у якій на кожний символ вхідної послідовності формується більше одного символу вихідної послідовності [2]. На відміну від блокових кодів, вони не ділять інформацію на фрагменти і працюють з нею як із суцільним потоком даних.

Згорткове кодування - це ітеративна обробка потоку бітів, що створює залежність кожного біта від кількох попередніх. Згорткове кодування задають за допомогою породжуючих поліномів, які визначають структуру двійкового кодера згорткового коду. Кодове слово на виході такого кодера складається у вигляді двох послідовностей, які у двійковій формі представляють коефіцієнти відповідних породжуючих поліномів.

#### **Мета і задачі дослідження.**

Метою роботи є дослідження цифрових систем зв'язку з згортковими кодами в середовищі Matlab. Для досягнення цієї мети поставлені наступні завдання:

- провести огляд завадостійких кодів у цифрових системах зв'язку;
- розглянути і проаналізувати структурну схему каналу зв'язку з згортковими кодами;
- дослідити модель системи зв'язку з згортковими кодами, оцінити енергетичну ефективність.

**Об'єкт дослідження** – процес передачі і приймання, сигналів з згортковими кодами у цифрових системах зв'язку.

**Предмет дослідження** – методи та засоби згорткового кодування і декодування у цифрових системах зв'язку.

**Методи досліджень.** Для вирішення поставлених наукових завдань використовується математичний апарат теорії цифрової обробки сигналів, статистичного оцінювання, теорії приймання та оброблення сигналів.

#### **Наукова новизна одержаних результатів:**

1. Вперше проведена оцінка енергетичної ефективності цифрових систем зв'язку з згортковими кодами. Згортковий код з швидкістю  $R=0,5$  ефективніше коду Ріда-Соломона (7, 5) на 2 дБ.

2. Отримали подальший розвиток методи завадостійкого кодування і декодування.

**Практичне значення одержаних результатів:**

Проведене дослідження згорткових кодів як одиночних, так і в складі каскадного коду показало їх ефективність і переваги практичного використання в порівнянні з кодом Ріда-Соломона

**Публікації.** Результати дипломної роботи магістра опубліковані у матеріалах науково-практичної конференції МНІС ІІІ-2020.

**Структура та обсяг магістерської атестаційної роботи.** Дипломна робота магістра складається із вступу, чотирьох розділів, висновків, переліку джерел посилання та додатків. Дипломна робота магістра має загальний обсяг сторінок, з яких основний зміст викладений на сторінках друкованого тексту, містить рисунків та таблиць. Перелік джерел посилання складається з джерел.

# 1 ЗАВАДОСТІЙКЕ КОДУВАННЯ

## 1.1 Визначення завадостійкого кодування

Кодування сигналу - його представлення в певній формі, зручній або придатній для передачі повідомлень. У процесі кодування елементи переданого повідомлення перетворюються в відповідні кодові символи. Зокрема, для передачі 32 основних букв алфавіту необхідно передати числа від 0 до 31. Щоб передати будь-яке число, записане в десятковій формі, потрібна передача десяти цифр від 0 до 9 (для даного випадку потрібні десять елементарних сигналів). Реалізація кодування на передавальній стороні завжди передбачає застосування зворотної процедури - декодування - для відновлення повідомлення за прийнятими кодовими символами.

Код - повна сукупність умовних символів, яку застосовують для кодування повідомлень. Взаємозв'язок символів (або комбінацій символів, якщо кодуються не окремі символи) вихідного алфавіту з їх кодовими комбінаціями складає таблицю відповідності (або таблицю кодів). Множина можливих кодових символів (елементарних сигналів) називають кодовим алфавітом, а їх число  $m$  - основою коду. Код з основою  $m = 2$  - бінарний, з іншими основи - багатопозиційний. При підставі коду  $m$  правила кодування  $K$  елементів повідомлення зводяться до правил записи  $K$  різних чисел в системі числення на основі  $m$ .

Найпростішими електричними сигналами, які утворюють символи при кодуванні повідомлень, є імпульси струму або напруги. При кодуванні кожен елемент повідомлення записують певною сукупністю кодових символів, яку називають кодовою комбінацією. Число розрядів  $n$ , що утворюють кодову комбінацію, називають розрядністю коду або довжиною кодової комбінації.

Щоб закодувати повідомлення джерела, необхідно встановити визначені правила. Сукупність правил кодового позначення об'єктів називають системою кодування. Таку систему можна виразити по-різному.

Часто дуже зручною системою кодування є кодова таблиця, в якій наводяться алфавіт кодованих повідомлень і відповідні їм кодові комбінації.

Найбільш широко при кодуванні дискретних повідомлень застосовують двійковий код, який має в техніці особливе значення. Це пояснюється тим, що наявність і відсутність посліжки найбільш впевнено помітні. Крім того, дуже просто практично реалізувати пристрій, що має два можливих стани (відкрито - закрито). Зокрема, пристрій з двома стійкими станами, наприклад електронний перемикач або тригер, здатний зберігати один біт інформації,  $N$  таких пристроїв -  $N$  біт. Тому двійковий код знаходить переважне застосування не тільки в системах зв'язку, а й в комп'ютерах, автоматичі, тощо.

Необхідна кількість розрядів для кодування повідомлення при заданому максимальному числі рівнів шкали квантування  $U_{max}$  визначаються з співвідношення  $n = \log_2 N_{max}$ . Якщо кодова група містить  $n$  символів 0 і 1, то за допомогою подібного  $n$ -розрядного двійкового коду можна закодувати числа до  $N_{max} = 2^n$ .

Якщо все кодові слова мають однакове число символів, то код називають рівномірним, в іншому випадку - нерівномірним. Довжину кодової комбінації визначають кількістю окремих елементів, що входять до неї. Рівномірний код при  $n = 3$  називають трибітним, при  $n = 5$  - п'ятибітним або п'ятиелементним кодом, тощо.

Код Бодо ( $m = 2, n = 5$ ) - це код, елементами якого є посліжка і відсутність посліжки однакової тривалості і абсолютного значення, але різної полярності. Код Бодо рівномірний, кожній букві відповідає п'ять елементарних знаків. Всього можна скласти  $N = 2^5 = 32$  комбінацій. Для збільшення числа знаків в кодї Бодо застосовують другий регістр з 32 знаками.

Базовий і відомий приклад нерівномірних кодів - код Морзе, у якого кодові слова мають різну довжину: знакам, що найбільш часто зустрічаються, присвоюються найменш короткі кодові знаки і навпаки. У кодї Морзе

символи 1 і 0 використовуються тільки в двох поєднаннях - як одиночі (1 і 0) або як потрійні (111 і 000). Сигнал, який відображається однією одиницею (короткою послідовністю струму), відповідає точці (.), Трьом одиницям (втричі довшою послідовністю струму) - тире (нуль струму або пауза). Символ 0 використовується як знак, що відокремлює точку від тире, крапку від точки і тире від тире. Сукупність знаків 000 використовується як розділовий знак між кодовими комбінаціями. Крім цього сукупність знаків 00000 використовується для відокремлення слів. Код Морзе заснований на статистиці англійської мови. Зауважимо, що код Морзе - це кодування не з двійковим вхідним алфавітом (точка і тире), як здається на перший погляд, а з потрійним алфавітом: крапка, тире і пробіл. При радіозв'язку кодом Морзе передавач випромінює радіоімпульси різної довжини, відповідні точкам і тире кодових комбінацій. У приймачі за допомогою спеціального генератора і змішувача радіоімпульси перетворюються в відрізки коливань звукової частоти. Змінюючи частоту коливань генератора, оператор має можливість вибрати найбільш сприятливу для себе частоту звукових коливань. Швидкість передачі знаків при слуховому телеграфному зв'язку залежить від кваліфікації оператора і зазвичай становить близько 100 знаків в хвилину. До переваг коду Морзе слід віднести високу стійкість перед завадами прийому на слух, завдяки чому він знаходить широке застосування в радіозв'язку. При прийомі на слух перетворення сигналу в повідомлення здійснює людина, яка, має певний досвідом, може розрізняти сигнали, значно уражені завадами.

Історія кодування. Зародилося кодування в давні часи і використовувалося як для представлення інформації в символічному вигляді, так і для шифрування повідомлень і тайнопису.

У теорії інформації доводиться, що як для неортогональних, так і для ортогональних сигналів значення ймовірності помилки  $p$  при оптимальному прийомі однозначно визначається числом символів  $m$  і параметром відношення сигнал/шум  $Q_{вихл} = E_0/W_0$ , де  $E_0$  - енергія елемента сигналу;  $W_0$  - спектральна густина потужності білого шуму. Звідси можна зробити

висновок, що необмежене підвищення вірності передачі повідомлень (тобто спадання ймовірності помилки  $p$  до нуля) може бути отримано лише за рахунок необмеженого зростання енергетичного параметра  $Q_{вих0}$  (шляхом збільшення потужності сигналу). Зокрема, якщо задані середня потужність сигналу  $P_c$  і спектральна густина потужності білого шуму  $W_0$ , то це може відбуватися тільки при прагненні до нуля швидкості передачі символів  $\nu$ , оскільки  $E_0 = P_c T_c$ , а тривалість сигналу  $T_c = 1/\nu$ .

К. Шеннон довів принципову можливість безпомилкової передачі сигналів, якщо швидкість передачі менше пропускної здатності каналу зв'язку, яка тим більше, чим вище відношення сигнал/шум на вході приймального пристрою. Це вказувало на те, що енергетика ліній зв'язку визначає тільки їх пропускну здатність, а як завгодно високу завадостійкість приймання повідомлень можна досягти застосуванням спеціальним чином побудованих кодів. Революційні для того часу ідеї Шеннона здійснили переворот у свідомості зв'язківців, адже до створення теорії інформації вважалося само собою зрозумілим, що єдині можливості підвищення завадостійкості прийому сигналів складаються в збільшенні потужності передавача або в багаторазовій передачі по каналу зв'язку одного і того ж повідомлення. Ці способи призводять до дуже низької ефективності використання пропускної здатності каналу зв'язку. Фактично Шеннон стверджував, що потужність сигналу, шум у каналі і смуга частот обмежують лише швидкість передачі, а не її точність. Шеннон знайшов необхідні і достатні умови зменшення ймовірності помилки до нуля. Крім того, він зробив це для різних моделей каналів зв'язку, зокрема з обмеженою смугою частот, для фіксованого числа форм сигналів, для джерел з залежними символами (званих надлишковими джерелами), для джерел безперервних повідомлень, для критеріїв вірності прийому дискретних повідомлень, відмінних від ймовірності помилки символу  $p$ , і т. д. Це і склало сутність нової теорії інформації і теорії кодування. Однак К. Шеннон не вказав, як знайти потрібні коди, а лише довів їх існування.

## 1.2 Класифікація методів кодування

Методи кодування, що використовуються в техніці зв'язку і коди можна класифікувати за рядом специфічних ознак. Наведена класифікація відображає коди, які використовуються (рис. 1.1).

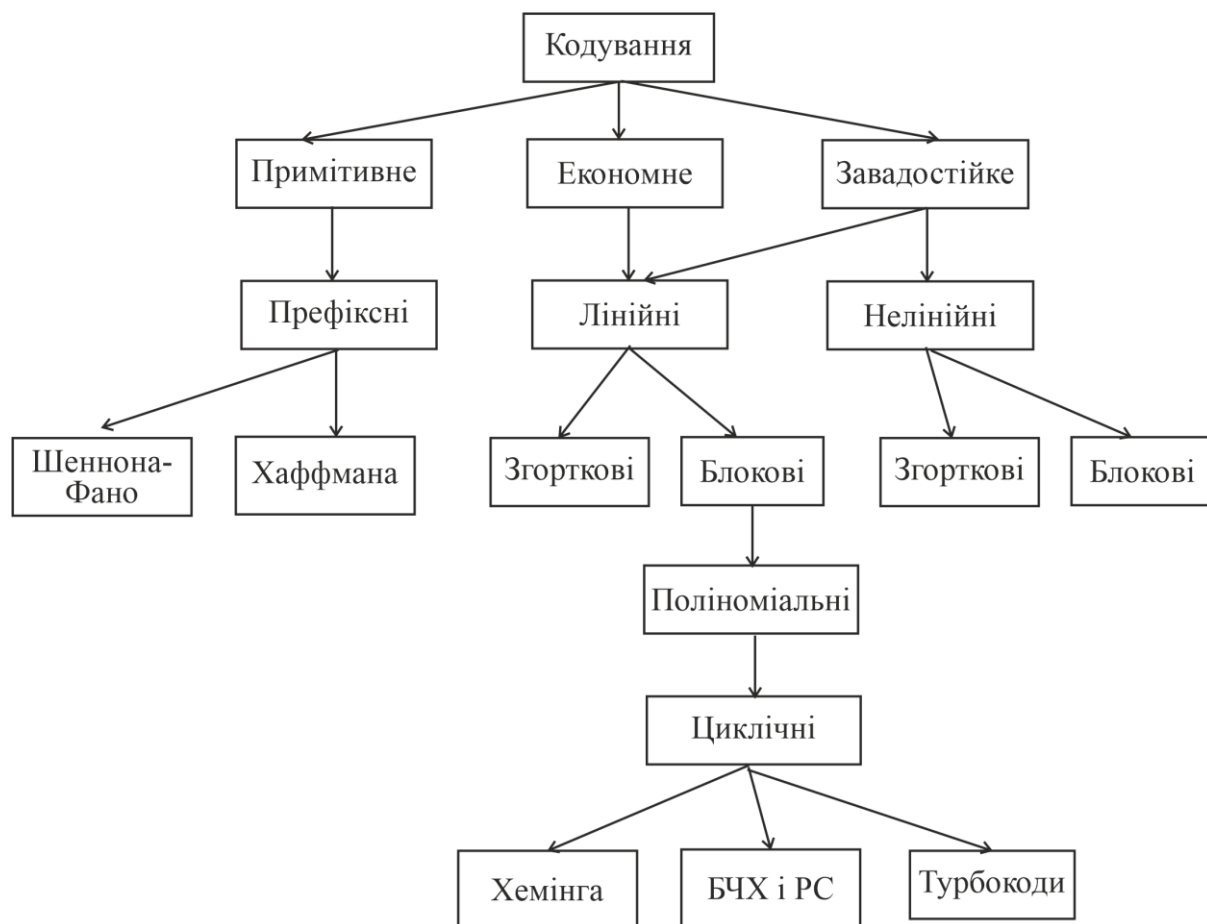


Рисунок 1.1 – Класифікація кодів

За призначенням кодування (як і відповідні коди) поділяють на примітивне, економне і завадостійке.

За допомогою кодування вирішують наступні завдання:

- узгодження алфавіту джерела повідомлень з алфавітом каналу - примітивне (первинне, просте) кодування;

- «стиснення» інформації (зменшення або повне усунення надмірності, що міститься в повідомленні) - економне (ефективне) кодування;

- виявлення (detection) або виправлення (correction) помилок, що виникають в каналі зв'язку через завади і спотворення сигналу, - завадостійке (коригуюче, надмірне) кодування. Завадостійке кодування часто називають канальним кодуванням.

### **1.3 Економне кодування**

Економне кодування методом Шеннона-Фано (запропонований незалежно Шенноном і Фано в 1948-49 роках). Спрощено принцип алгоритму Шеннона-Фано полягає в наступному. Символи алфавіту джерела повідомлення записують в порядку зменшення ймовірностей. Потім їх поділяють на дві частини так, щоб суми ймовірностей символів, що входять до кожної з частин, були приблизно однаковими. Всім символам першої частини в якості першого символу комбінації коду приписують нуль, а символам другої частини в якості першого символу комбінації - одиницю. Потім кожну з цих частин (якщо вона містить більше одного повідомлення) ділять в свою чергу на дві, по можливості, рівноімовірні частини і до них застосовують таке ж правило кодування. Процедуру повторюють до тих пір, поки в кожній з частин не залишиться по одному повідомленню.

Економне кодування методом Хаффмана. Цей метод кодування з'явився дуже давно і набув широкого поширення через простоту реалізації і високу швидкість передачі. Метод не вимагає великих обчислювальних потужностей на відміну від арифметичного кодування.

Для роботи алгоритму необхідно мати таблицю значень ймовірностей різних символів, які можуть зустрітися в даний момент кодування. На основі таблиці будують бінарне дерево Хаффмана наступним чином:

- впорядкувати символи по зростанню ймовірності їх появи;

- перші два символи в отриманому ряді об'єднати в один, зіставивши першому символу нуль, другого символу - одиницю; ймовірності цих двох символів скласти. Якщо в ряді залишився один символ, то закінчити, інакше перейти до першого пункту.

Основним і суттєвим недоліком цього методу є те, що він кодує символи за допомогою цілого числа біт, що знижує степінь стиснення і зводить нанівець точне передбачення ймовірностей, яке дають деякі відмінні сучасні алгоритми моделювання. Відзначимо, що алгоритм створення коду Шеннона-Фано називають зверху вниз, а Хаффмана - від низу до верху. Перевагами даних методів є їх очевидна простота реалізації і, як наслідок цього, висока швидкість кодування і декодування. Основним недоліком є їх неоптимальність в загальному випадку.

#### **1.4 Код Хемінга**

Код Хемінга - самоконтролюючий і самокорегуючий код. Побудований по відношенню до двійкової системі числення. Дозволяє виправляти одиночну помилку і знаходити подвійну. Коди Хемінга почали використовувати з 1950 р.

Коди Хемінга є самоконтролюючими коди, тобто кодами, що дозволяють автоматично виявляти помилки при передачі даних. Одиночна помилка в будь-якому розряді кодового слова змінить парність спільної кількості одиниць. Лічильники за модулем 2, підраховує кількість одиниць, що містяться у двійкових цифрах числа, і сигналізують про наявність помилки.

Коди, де є можливість автоматичного виправлення помилок, називаються самокорегуючими. Для побудови самокорегуючого коду, одного контрольного розряду недостатньо. Кількість контрольних розрядів  $k$  має бути вибрано так, щоб виконувалась нерівність  $2^k \geq k + m + 1$ , або  $k \geq \log_2(k + m + 1)$ , де  $m$  - кількість двійкових розрядів у кодовому слові.

Основні характеристики самокорегуючих кодів наступні:

- число дозволених і заборонених комбінацій. Якщо  $n$  - число символів в блоці,  $r$  - число перевірочних символів в блоці,  $k$  - число інформаційних символів, то  $2^n$  - число можливих кодових комбінацій,  $2^k$  - число дозволених кодових комбінацій,  $2^n - 2^k$  - число заборонених комбінацій;

- надмірність коду. Величина  $r/n$  називається надмірністю коригуючого коду;

- мінімальна кодова відстань. Мінімальною кодовою відстанню  $d$  називається мінімальне число спотворених символів, необхідне для переходу з однієї дозволеної кодової комбінації в іншу;

- число виявлених і виправлених помилок. Якщо  $g$  - кількість помилок, які код здатний виправити, то необхідно і достатньо, щоб  $d \geq 2g + 1$ ;

- коригувальні можливості кодів.

Побудова кодів Хеммінга засноване на принципі перевірки на парність числа одиничних символів: до послідовності додається такий елемент, щоб число одиничних символів у отриманій послідовності було парним:

$$r_1 = i_1 \oplus i_2 \oplus \dots \oplus i_k$$

Знак  $\oplus$  означає підсумовування за модулем два:

$$S = i_1 \oplus i_2 \oplus \dots \oplus i_n \oplus r_1.$$

Якщо  $S = 0$ , то помилки нема, якщо  $S = 1$  - є однократна помилка.

Такий код називається  $(k + 1, k)$  або  $(n, n - 1)$ . Перше число – кількість елементів послідовності, друге – кількість інформаційних символів.

Для кожного числа перевірочних символів  $r = 3, 4, 5, \dots$  існує класичний код Хемінга:

$(n, k) = (2^r - 1, 2^r - 1 - r)$ , тобто  $(7, 4)$ ,  $(15, 11)$ ,  $(31, 26)$ .

При інших значеннях  $k$  отримуємо скорочений код.

Розглянемо класичний код Хемінга  $(7, 4)$ . Згрупуємо перевірочні символи

$$r_1 = i_1 \oplus i_2 \oplus i_3,$$

$$r_2 = i_2 \oplus i_3 \oplus i_4,$$

$$r_3 = i_1 \oplus i_2 \oplus i_4,$$

Отримання кодового слова виглядає наступним чином:

$$(i_1 \ i_2 \ i_3 \ i_4) \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = (i_1 \ i_2 \ i_3 \ i_4 \ r_1 \ r_2 \ r_3)$$

На вхід декодера надходить кодове слово  $V = (i'_1, i'_2, i'_3, i'_4, r'_1, r'_2, r'_3)$  де позначені символи, які можуть бути спотворені в результаті дії завади. В декодері в режимі виправлення помилок будується послідовність синдромів:

$$S_1 = r_1 \oplus i_1 \oplus i_2 \oplus i_3,$$

$$S_2 = r_2 \oplus i_2 \oplus i_3 \oplus i_4,$$

$$S_3 = r_3 \oplus i_1 \oplus i_2 \oplus i_4.$$

Отримання синдрому виглядає наступним чином:

$$(i_1 \ i_2 \ i_3 \ i_4 \ r_1 \ r_2 \ r_3) \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = (S_1 \ S_2 \ S_3).$$

$S = (S_1, S_2, S_3)$  називається синдромом послідовності.

Синдром  $(0,0,0)$  вказує на те, що в послідовності немає спотворень. Кожному ненульовому синдрому відповідає певна конфігурація помилок, яка виправляється декодером. Кодер коду Хемінга приведений на рис. 1.2.

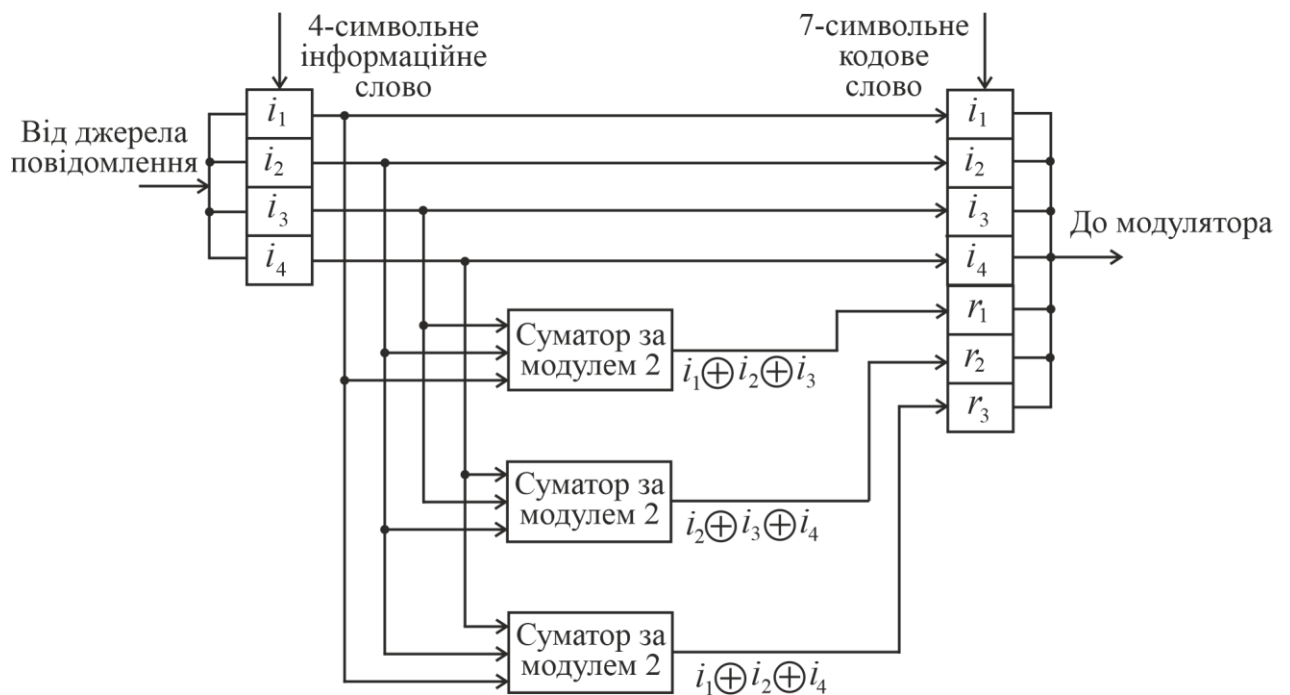


Рисунок 1.2 – Кодер коду Хемінга

Декодер коду Хемінга приведений на рис. 1.3.

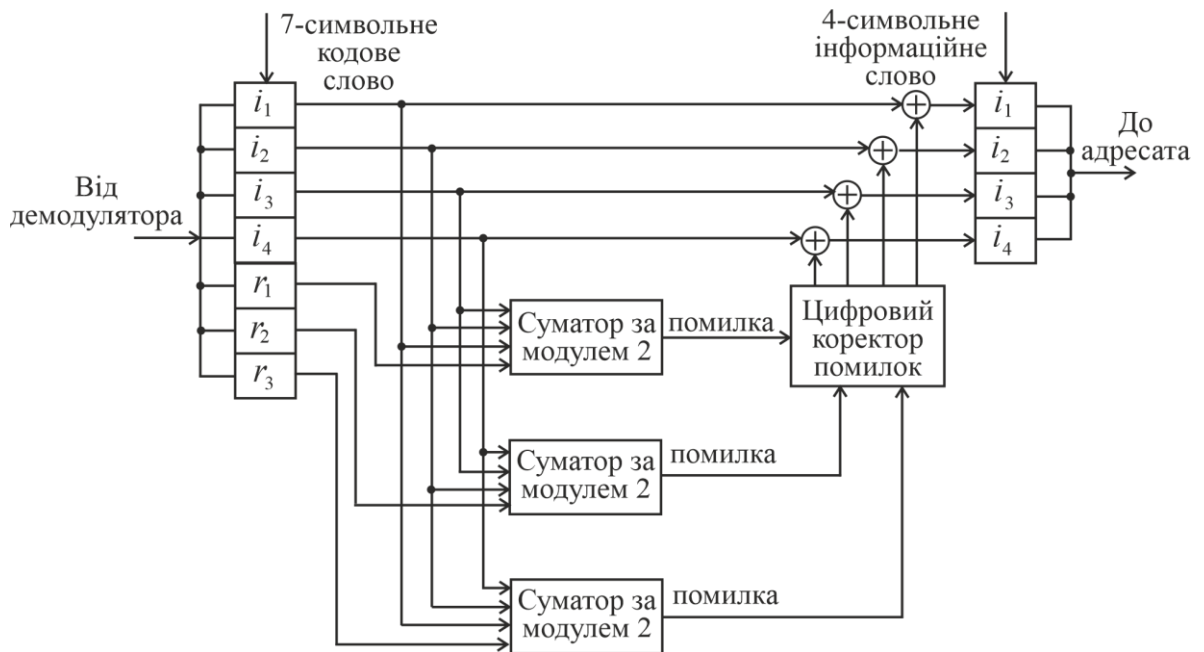


Рисунок 1.3 – Декодер коду Хемінга

У загальному випадку кількість контрольних біт в кодовому слові дорівнює двійковому логарифму числа, на одиницю більшого, ніж кількість біт кодового слова (включаючи контрольні біти); логарифм округляється в більшу сторону. Наприклад, інформаційне слово довжиною 1 біт вимагає двох контрольних розрядів, 2-, 3- або 4-бітове інформаційне слово - трьох, 5 ... 11-бітове - чотирьох, 12 ... 26-бітове - п'яти контрольних розрядів, тощо.

Алгоритм декодування за Хемінгом ідентичний алгоритму кодування. Матриця перетворення потрібної розмірності множиться на матрицю-стовпець кодового слова і кожний елемент отриманої матриці-стовпця береться за модулем 2. Отримана матриця-стовпець має назву «матриця синдромів».

### 1.5 Згорткові коди

Згорткові коди є окремим випадком ґратчастих кодів. Можна також вважати, що решітка є просто іншим (іноді більш зручним) способом представлення і звичайних згорткових кодів, які можуть бути

систематичними і несистематичними. У послідовності кодових символів систематичного коду без зміни міститься послідовність інформаційних символів. У несистематичному коді положення інформаційних символів в кодовій послідовності вказати не можна. При свертковому кодуванні символи коду розподіляються в часі на значному інтервалі, що підвищує його стійкість до одиночних пакетів помилок. Кодер для згорткового коду являє собою пристрій з пам'яттю, і надлишкові символи, що додаються залежать не тільки від поточного фрагмента повідомлення, але і від внутрішнього стану кодера, тобто в кінцевому рахунку від попередніх фрагментів повідомлення. Внутрішня структура кодера утворюється поєднанням багаторозрядних регістрів зсуву і суматорів за модулем 2. До складу кодера може входити один або декілька регістрів. На виході кодера в кожному такті формуються вихідні символи, які також перетворюються в послідовний потік з підвищеною тактовою частотою і надходять в канал зв'язку. Вихідна послідовність кодера може бути представлена як цифрова згортка вхідної інформаційної послідовності і імпульсної характеристики кодера (звідси походить назва кодів - згорткові).

Для представлення згорткових кодів можна використовувати породжуючі многочлени і матриці, кодові дерева, діаграми станів або графчасті структури. З математичної і схемотехнічної точок зору найбільш зручно опис за допомогою породжуючих многочленів, що використовують оператори затримки  $D$ . Породжуючі многочлени визначають для кожного з суматорів за модулем 2 номери розрядів регістра зсуву, до яких підключаєні входи суматорів.

Згорткові коди, на відміну від блокових, не ділять інформацію на фрагменти і працюють з нею як із суцільним потоком даних. Такі коди, як правило, породжуються дискретною лінійною інваріантною в часі системою. Тому, на відміну від більшості блокових кодів, згорткове кодування - дуже проста операція, чого не можна сказати про декодування.

Кодування згортковим кодом проводиться за допомогою регістра зсуву, відводи від якого підсумовуються за модулем два (рис. 1.4). Таких сум може бути дві (найчастіше) або більше.

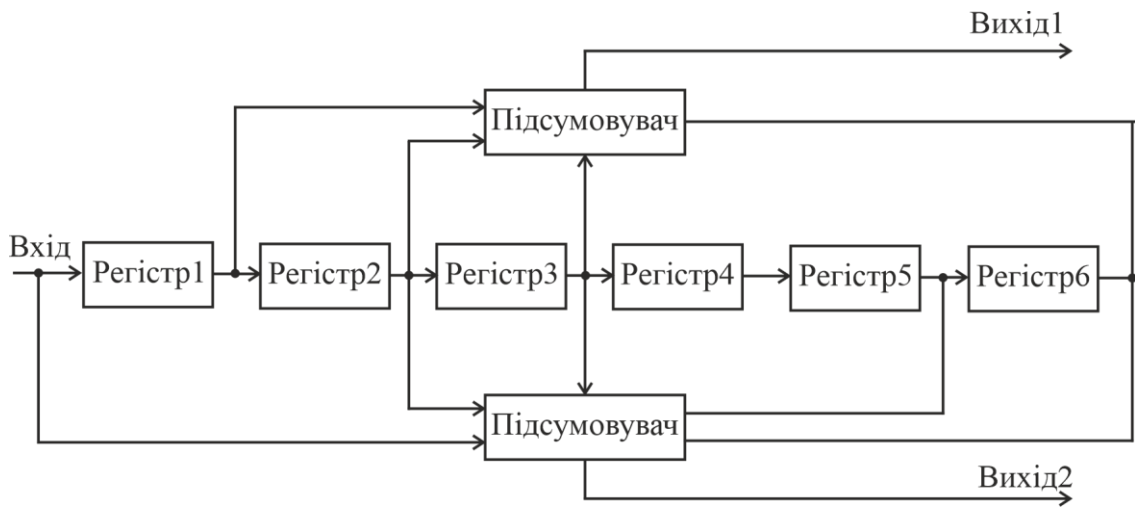


Рисунок 1.4 – Згортковий кодер ( $k=7$ ,  $R=1/2$ )

Декодування згорткових кодів, як правило, проводиться за алгоритмом Вітербі, який намагається відновити передану послідовність згідно з критерієм максимальної правдоподібності.

Згорткові коди ефективно працюють в каналі з білим шумом, але погано справляються з пакетами помилок. Більш того, якщо декодер помиляється, на його виході завжди виникає пакет помилок.

## 1.6 Особливості кодування і декодування згорткових кодів

Згортковий код - це коригуючий помилки код, в якому:

- на кожному такті роботи кодера  $k$  символів вхідної послідовності перетворюються в  $n > k$  символів вихідний послідовності;
- в перетворенні також беруть участь  $t$  попередніх символів;
- виконується властивість лінійності (якщо двом послідовностям  $x$  і  $y$ , що кодуються, відповідають кодові послідовності  $X$  і  $Y$ , то послідовності  $ax + by$  відповідає  $aX + bY$ ).

Згорткове кодування є окремим випадком деревоподібних і ґратчастих кодів. Узагальнена структурна схема згорткового кодера приведена на рис. 1.5.

Для блокових кодів інформаційна послідовність розбивається на блоки довжиною  $k$  символів, блоки кодуються незалежно один від одного

Для згорткових кодів інформаційна послідовність розбивається на кадри довжиною  $k_0$  символів. Кожний кадр кодуються з урахуванням  $m$  попередніх кадрів. Властивості згорткових кодів:

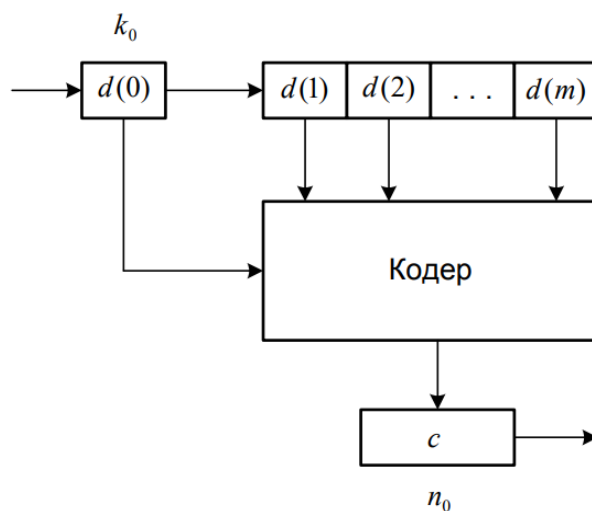


Рисунок 1.5 – Узагальнена структурна схема згорткового кодера

- згорткові коди мають властивості лінійності і сталості в часі;
- довжина кодового обмеження - кількість кадрів, які враховуються при кодуванні  $m$ , може бути нескінченною;
- інформаційна довжина  $k = k_0(m + 1)$ ;
- кодова довжина  $n = n_0(m + 1) = k(n_0 / k_0)$ ;

Згорткове кодування може задаватися схемою кодера (рис. 1.6).

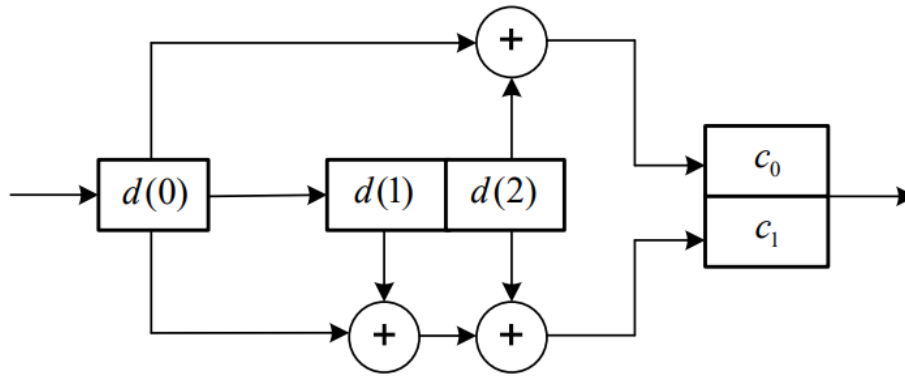


Рисунок 1.6 – Кодер згорткового коду

Розглянемо кодер згорткового (6,3) коду. Довжина інформаційного кадру  $k_0 = 1$ , довжина кодового кадру  $n_0 = 2$ .

Опис згорткових кодів за допомогою многочленів

Можна записати операцію кодування за допомогою многочленів

$$c_j = g_j(x) \cdot d(x)$$

У многочлені  $d(x)$  коефіцієнти перед різними степенями  $x$  відповідають різним розрядами і різним затримкам  $d_i(l)$ . В інформаційному кадру  $k_0$  символів, многочлен  $g_j(x)$  можна представити сумою  $k_0$  многочленів ( $g_{ij}(x)$ ), кожний степені не більше  $m$ .  $g_{ij}(x)$  - визначає  $j$ -й розряд кодового кадру, по  $i$ -им розрядами всіх інформаційних кадрів.

Многочлени  $g_{ij}(x)$  називаються породжуючими многочленами згорткового коду. Число породжуючих многочленів дорівнює  $n_0 \cdot k_0$ , деякі многочлени можуть бути однаковими. Матриця породжуючих многочленів

$$G(x)_{[k_0 \times n_0]} = [g_{ij}(x)]$$

$\text{trellis} = \text{poly2trellis}(\text{ConstraintLength}, \text{CodeGenerator})$  повертає опис ґратчастої структури, відповідне перетворення для кодера з прямим зв'язком  $K/N$ .  $K$  - кількість вхідних бітових потоків для кодера, а  $N$  - кількість вихідних з'єднань.  $\text{ConstraintLength}$  вказує затримку для вхідних бітових потоків у кодері.  $\text{CodeGenerator}$  визначає вихідні з'єднання для вхідних бітових потоків з кодером.

Функція  $\text{poly2trellis}$  приймає поліноміальний опис згорткового кодера і повертає відповідний опис ґратчастої структури. Цей вихід може бути використаний в якості вхідного сигналу до функцій  $\text{convenc}$  і  $\text{vitdec}$ . Він також може бути використаний в якості значення параметра маски для згорткового кодера, декодера Вітербі, і APP Decoder блоків.

$\text{trellis} = \text{poly2trellis}(\text{ConstraintLength}, \text{CodeGenerator}, \text{FeedbackConnection})$  повертає опис ґратчастої структури, відповідне перетворення для кодера зворотного зв'язку швидкості  $K/N$ .  $K$  - кількість вхідних бітових потоків для кодера, а  $N$  - кількість вихідних з'єднань.  $\text{ConstraintLength}$  вказує затримку для вхідних бітових потоків в кодері.  $\text{CodeGenerator}$  визначає вихідні з'єднання для вхідних бітових потоків з кодером.  $\text{FeedbackConnection}$  визначає з'єднання зворотного зв'язку для кожного з  $K$  вхідних бітових потоків для кодера.

Trellis структура для систематичного згорткового кодера зі швидкістю  $1/2$  зі зворотним зв'язком, що показана на рис. 1.7.

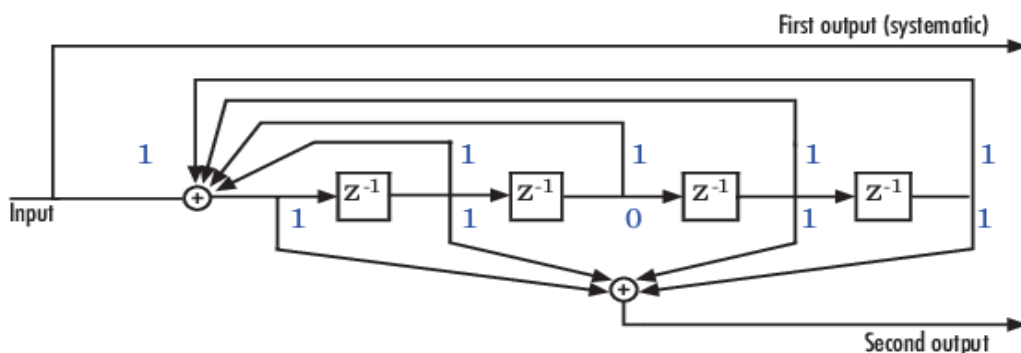


Рисунок 1.7– Trellis структура згорткового кодера швидкості  $1/2$

Цей кодер має властивість обмеження довжини  $\text{ConstraintLength}=5$ , поліноміальну матрицю генератора [37 33] і поліном з'єднання зворотного зв'язку рівний 37. Перший поліном генератора збігається з поліномом з'єднання зворотного зв'язку, оскільки перший вихід відповідає систематичним бітам. Поліном зворотного зв'язку представлений двійковим вектором [1 1 1 1 1], відповідним верхньому ряду двійкових цифр на діаграмі. Ці цифри позначають з'єднання від виходів регістрів до суматора. Початкова 1 відповідає вхідному біту. Вісімкове представлення двійкового числа 11111 є число 37.

Другий многочлен генератора представлений двійковим вектором [1 1 0 1 1], що відповідає нижньому ряду двійкових цифр на діаграмі. Вісімкове число, відповідне бінарного числа 11011, є число 33.

Використовуйте `poly2trellis`, щоб перетворити поліном в гратчасту структуру. При використанні з поліномом зворотного зв'язку, `poly2trellis` встановлює з'єднання зворотного зв'язку з входом решіток.

При згортковому кодуванні перетворення інформаційних послідовностей у вихідні та кодові відбувається безперервно. Кодер двійкового згорткового коду містить зсувний регістр з  $m$  розрядів і суматори за модулем 2 для утворення кодових символів у вихідній послідовності. Входи суматорів з'єднані з певними розрядами регістра.

### **Висновки до першого розділу**

1. Коди, в яких можливе автоматичне виправлення помилок, називаються самокорегуючими.

2. Згорткові коди породжуються дискретною лінійною інваріантною в часі системою. Згорткові коди ефективно працюють в каналі з білим шумом, але погано справляються з пакетами помилок.

3. Згорткові коди використовуються для надійної передачі даних: відео, мобільного зв'язку, супутникового зв'язку. Вони використовуються разом з кодом Ріда - Соломона та іншими кодами.

## 2 ДЕКОДУВАННЯ З М'ЯКИМ РІШЕННЯМ

### 2.1. Передача двійкових сигналів по каналам зв'язку

У цьому розділі розглядається декодування з (додатковою) «м'якою» інформацією, одержуваної з каналу. Розглянемо простий випадок передачі двійкових сигналів по каналу з АБГШ (адитивним білим гаусовим шумом). Щоб обґрунтувати доцільність застосування декодування з м'яким рішенням зауважимо, що шумова компонента в задачі відновлення даних або прийому сигналів є безперервною, тобто не дискретною, за своєю природою. Це означає, що прийняті символи представляються (квантовими) дійсними числами (відповідними напрузі, струму, тощо), а не двійковими символами або символами з кінцевого поля  $GF(2^m)$ .

Коли вибираються жорсткі рішення щодо прийнятих символів, можуть відбуватися помилки. Існують два методи декодування завадостійких кодів, заснованих на прийнятій послідовності дійсних чисел:

#### 1. Декодування з жорстким рішенням (hard decision decoding) (ДЖР).

При формуванні жорстких рішень щодо прийнятих з каналу величин відбуваються помилки. Мета ДЖР полягає у виправленні двійкових помилок, що виникли в процесі вибору жорстких рішень.

#### 2. Декодування з м'яким рішенням (soft-decision decoding) (ДМР).

Прийняті з каналу величини вводяться безпосередньо в декодер для формування оцінок кодової послідовності. Особливим випадком ДМР є декодування за максимумом правдоподібності (maximum-likelihood decoding) (МПД), при якому в якості рішення декодера вибирається найближча (в деякій метриці) кодова послідовність. Тут важливо пам'ятати, що для ДМР необхідно знати статистику шуму в каналі зв'язку.

У загальному випадку ДМР більш трудомістке, ніж ДЖР. Відзначимо дві основні причини. Одна з них полягає в тому, що ДМР виконує операцій з дійсними числами. У практичних застосуваннях ці числа квантуються з

кінцевою точністю (тобто представляються кінцевим числом біт). Для деяких систем передачі двійкових сигналів відомо, що квантування на 8 рівнів (або представлення чисел трьома бітами) забезпечує ефективність системи, близьку до використання обчислень з нескінченною точністю (без квантування).

Інша причина збільшення складності ДМР пов'язана з необхідністю обчислення апостеріорних статистик для кодових символів. Однак, збільшення трудомісткості окупається потенційним підвищенням ефективності системи кодування. Для двійкових сигналів в гаусовому каналі в разі ДМР та ж сама ефективність досягається при відношенні сигнал-шум на 2 - 3 дВ менше, ніж при ДЖР. Це означає, що в разі ДМР потужність, яку випромінює передавач може бути знижена на 50 - 60% в порівнянні з випадком ДЖР. Ця економія потужності перетворюється в менший розмір передавальної антени або в менший розмір приймальної антени при тій самій потужності передавача.

Для каналів з адитивним білим гаусовим шумом (АБГШ) імовірність прийнятого значення  $r_i$  за умовою, що був переданий символ  $v_i$ , дорівнює:

$$p(r_i | v_i) = p_{ni}(r_i - m(v_i)) = \frac{1}{\sqrt{\pi N_0}} e^{-(r_i - m(v_i))^2 / N_0} \quad (2.1)$$

Квадрат Евклідової відстані,  $D^2(r_i, m(v_i)) = (r_i - m(v_i))^2$ , є метрикою для декодування за максимумом правдоподібності (MLD). Позначимо  $E$  енергію переданого сигналу. Для відображення символів в сигнали (або модуляції) використовуємо двійкову фазову модуляцію (BPSK):

$$m(v_i) = \begin{cases} \sqrt{E}, & \text{якщо } v_i = 0 \\ -\sqrt{E}, & \text{якщо } v_i = 1 \end{cases} \quad (2.2)$$

або інакше  $m(v_i) = (-1)^v \sqrt{E}$

У двійковому каналі з фазовою модуляцією і гаусовим шумом метрика декодування може бути спрощена, якщо зауважити, що за умови  $v_i = 1$  маємо:

$$p(r_i | 1) = p_{ni}(r_i + 1) = \frac{1}{\sqrt{\pi N_0}} e^{-(r_i+1)^2/N_0}.$$

Після видалення констант, натуральний логарифм відношення умовних ймовірностей  $p(r_i | v) / p(r_i | 1-v)$ , – метрика логарифма правдоподібності, пропорційна  $-r$ , якщо  $v=1$ , і пропорційна  $r$ , якщо  $v=0$ . Таким чином, корисним є наступне зауваження: у двійковому каналі з гаусовим шумом обчислення метрики зводиться до зміни знака прийнятого з каналу сигналу.

## 2.2 Алгоритм Вітербі

Позначимо  $S_i^{(k)}$  стан (вузол) на решітці в момент  $i$ . Кожному стану решітки поставимо у відповідність метрику (вагу)  $M(S_i^{(k)})$  і певний шлях на решітці  $y^{(k)}$  (що закінчується у даному стані). Ключовою властивістю (для розуміння) алгоритму Вітербі є наступне.

У момент  $i$  найкращі (найближчі до прийнятої послідовності) шляхи  $y^{(k)}$  закінчуються в станах  $S_i^{(k)}$ , мають спільний початок в певний момент  $i - \ell$ .

Для двійкового згорткового коду з швидкістю  $1/2$  величина  $\ell$  повинна задовольняти умові  $\ell > 5m$ . Строго кажучи, ця величина залежить від відношення сигнал-шум. Зазвичай декодер Вітербі працює у вікні розміром  $L$  прийнятих з каналу  $n$  ребер. Ця величина називається глибиною декодування. Очевидно, що умова  $L > \ell$  є необхідною (величина вікна впливає на ймовірність помилки декодера Вітербі).

Розглянемо застосування алгоритму Вітербі до двійкового згорткового коду з пам'яттю  $m$  і швидкістю  $1/n$ . Нехай  $v[i] = (v_0[i] v_1[i] \dots v_{n-1}[i])$  позначає мітку (тобто кодові символи) ребра кодової решітки,  $r[i] = (r_0[i] r_1[i] \dots r_{n-1}[i])$  позначає  $i$ -е ребро на виході каналу. Основні етапи декодування можна сформулювати наступним чином (рис. 2.1).

Початкові умови. Нехай  $i = 0$ .

Встановимо нульові початкові значення метрик і шляхів:

$$M(S_0^{(k)}) = 0,$$

$y_0^{(k)} = ()$  – порожній масив.

1. Обчислення метрик ребер решітки. В  $i$ -му перетині решітки (на  $i$ -му кроці алгоритму) визначаємо метрики ребер:

$$BM_i^{(b)} = d_H(r[i], v[i]), b = \sum_{\ell=0}^{n-1} v_\ell[i] \cdot 2^{n-1-\ell},$$

які визначаються кодovими символами  $v[i]$  на кожному ребрі решітки та прийнятими з каналу символами  $r[i]$ .

2. Додати, порівняти і вибрати (ДПВ)

Для кожного стану  $S^{(k)}$ ,  $k = 0, 1, \dots, 2^{m-1}$  і відповідної пари ребер, що входять в даний стан з двох попередніх  $S^{(k_a)}$ ,  $a \in \{1, 2\}$ , обчислити і порівняти суми  $MS_{i-1}^{(k_a)} + BS_i^{(b_a)}$ , де  $b_a = v_0[a]2^{n-1} + v_1[a]2^{n-2} + \dots + v_{n-1}[a]2^0$ .

Вибрати ребро, яке дає найменшу метрику шляху, і запам'ятати,

$$M(S_i^{(k)}) = \min\{M(S_{i-1}^{(k)}) + BM_i^{(b_1)}, S_{i-1}^{(k)} + BM_i^{(b_2)}\} \quad (2.3)$$

3. Оновлення пам'яті шляхів

Для кожного стану  $S_i^{(k)}$ ,  $k = 0, 1, \dots, 2^{m-1}$  запам'ятати кращі шляхи  $y^{(k)}$ , доповнивши кожен з них кращим ребром  $v_{kj}$ ,  $j \in \{1, 2\}$ .

4. Декодування символів. Якщо  $i > L$ , то видати як оцінку кодової послідовності ребро  $y_{i-L}^{(\gamma)}$ , де  $\gamma$  – індекс стану  $S^{(\gamma)}$  з найменшою метрикою. Встановити  $i = i + 1$  і повернутися на початок процедури.



Рисунок 2.1. – Структурна схема декодера Вітербі

Для реалізації на інтегральних мікросхемах зазвичай застосовується метод зворотного проходу пам'яті шляхів з відновленням інформаційних символів з переходів між станами кодової решітки.

### 2.3 Алгоритм Вітербі з Евклідовою метрикою

Алгоритм Вітербі може бути застосований для декодування даних, які закодовані двійковим згортковим кодом і передані по каналу з двійковою фазовою модуляцією і АБГШ. У порівнянні з алгоритмом Вітербі з жорстким рішенням, необхідні наступні зміни.

#### 1. Блок генерації метрики ребер:

Метрика для каналу з адитивним гаусовим шумом пропорційна кореляції між прийнятою з каналу послідовністю відліків і перевіркою кодовою послідовністю. Таким чином, замість Евклідових відстаней використовуються кореляційні метрики.

#### 2. Блок Додати-порівняти-вибрати (ACS):

Алгоритм Вітербі максимізує кореляційну метрику, а не мінімізує відстань. Алгоритм Вітербі може бути застосований до блокових кодів.

Синдромна решітка для двійкового лінійного блокового  $(N, K)$  коду  $C$  може бути побудована з перевіркою матриці наступним чином. Позначимо  $(v_1, v_2, \dots, v_N)$  кодове слово коду  $C$ . В момент  $i$ ,  $1 \leq i \leq N$ , стан решітки визначається частковими синдромами:

$$s_i = \sum_{j=1}^i v_j h_j \quad (2.4)$$

де сума обчислюється в  $GF(2)$ ,  $h_j$  –  $j$ -ий стовпець перевіркою матриці  $H$  і  $s_0 = (0, 0, \dots, 0)$ . Максимальне число станів синдромної решітки дорівнює  $\min(2^K, 2^{N-K})$ . Синдромна решітка має властивість мінімальності числа станів. Решітка, яка задовольняє умові мінімальності числа станів, називається мінімальною решіткою.

Розглянемо двійковий циклічний  $(7, 4, 3)$  код Хемінга з породжуючим поліномом  $g(x) = x^3 + x + 1$ . Перевірочний поліном дорівнює  $h(x) = 1 + x + x^2 + x^4$  і перевірна матриця коду  $C$  має вигляд:

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Піставимо у відповідність стану  $s_j = (s_0, s_1, s_2)$  наступне число  $I_j$ :

$$I_j = s_0 + 2s_1 + 2^2 \cdot s_2.$$

Решітка цього коду має 8 станів (оскільки  $2^{N-K} = 2^3$ ).

Для деяких класів кодів, таких як розширені коди БЧХ та коди Ріда-Мілера, решітка може бути розділена на секції. В результаті виходить більш

регулярна і симетрична ґратчаста структура з великим числом паралельних подрешіток, які можуть бути використані для побудови дуже швидких декодерів Вітербі для блокових кодів.

## 2.4 Алгоритм Чейза

Алгоритм Чейза, який використовує множину або список найбільш ймовірних послідовностей помилок, забезпечує майже оптимальне декодування. Ця множина помилок вибирається на основі оцінок надійності прийнятих символів. Кожна комбінація помилок додається до слова, прийнятого з жорсткими (посимвольними) рішеннями, і результат декодується з використанням декодера з жорстким рішенням. Для кожного декодованого кодового слова підраховується його метрика (відстань) щодо прийнятої з м'яким рішенням послідовності символів. В якості найбільш ймовірного рішення вибирається декодування кодового слова з найкращою метрикою.

Нехай  $C$  двійковий лінійний блоковий  $(N, K, d)$  код, що виправляє будь-яку комбінацію випадкових помилок ваги  $t = \lfloor (d - 1) / 2 \rfloor$  або менше. Нехай  $r = (r_1, r_2, \dots, r_N)$  отримане на виході каналу слово, де  $r_i = (-1)^{c^{(i)+w_i}}$  і  $w_i$  є Гаусова випадкова величина з нульовим середнім і дисперсією  $N_0 / 2$  для всіх  $i = 1, 2, \dots, N$ . Знакові розряди на виході каналу представляють жорсткі рішення щодо символів прийнятого слова,

$$z = (z_{0,0}, z_{0,1}, \dots, z_{0,N-1}), \quad z_{0,j} = \text{sign}(r_j), \quad 0 \leq j \leq N,$$

$$\text{sign}(x) = \begin{cases} 0, & x > 0 \\ 1, & \text{інше} \end{cases}$$

Надійність символів на виході каналу при прийомі двійкових сигналів з АБГШ є амплітуди  $|r_i|$ . Отримані надійності символів упорядковуються за допомогою будь-якого алгоритму сортування (швидкого сортування). Виходом алгоритму є список індексів  $I_j, j = 1, 2, \dots, N$ , згідно якому

$$|r_{i_1}| \leq |r_{i_2}| \leq \dots \leq |r_{i_k}|$$

На першому етапі декодування прийняте слово з жорсткими посимвольними рішеннями вводиться в декодер з жорстким рішенням. Позначимо  $v_0$  декодоване кодове слово. Тоді метрикою  $v_0$  щодо прийнятого слова  $g$  (з м'якими рішеннями) є величина

$$v_0 = \sum_{j=1}^N (-1)^{r_j} \cdot r_j \quad (2.5)$$

яка обчислюється і запам'ятовується як максимум. Чейз ввів три типи алгоритмів відповідно множинам комбінацій помилок, що виправляються.

Алгоритм типу 1. Перевіряються всі комбінації помилок на відстані не більше  $(d - 1)$  від прийнятого слова.

Алгоритм типу 2. Перевіряються комбінації помилок ваги  $\lfloor (d - 1) / 2 \rfloor$  і менше, які розміщуються на будь-яких позиціях за винятком  $\lfloor d / 2 \rfloor$  позицій з найменшими надійностями. Ефективність цього алгоритму лише трохи поступається алгоритму Типу – I, однак складність його набагато менше за рахунок меншої кількості комбінацій помилок, що перевіряються.

Алгоритм типу 3. Перевіряються ті комбінації помилок, для яких  $i$  помилок розміщуються на  $i$  найменш надійних позиціях,  $i$  непарне,  $1 \leq i \leq d - 1$ . Слід зауважити, що цей алгоритм тісно пов'язаний з алгоритмом декодування по мінімуму узагальненої відстані.

Через невисоку складність і хорошу ефективність алгоритм Чейза типу-ІІ став найбільш популярним з перерахованих вище алгоритмів. Структурна схема алгоритму Чейза показана на рисунку 76.

Розглянемо алгоритм Чейза, тип 2. Для  $i = 1, 2, \dots, 2^t - 1$  додати комбінацію помилок  $e_i$  до прийнятого з жорстким рішенням слову:  
 $z_i = e_i \oplus z_0$ .

Комбінації помилок генеруються на  $t$  найменш надійних позиціях. Такими є позиції з індексами  $\{I_1, I_2, \dots, I_t\}$ , надійності (амплітуди) яких мінімальні.

Кожен вектор  $z_i$  вводиться в декодер з жорстким рішенням, який формує кодове слово  $v_i$ ,  $i = 1, 2, \dots, 2^t - 1$ . Обчислити метрику (2.5) і, якщо вона максимальна, запам'ятати кодове слово  $v_i$  - як найбільш ймовірне.

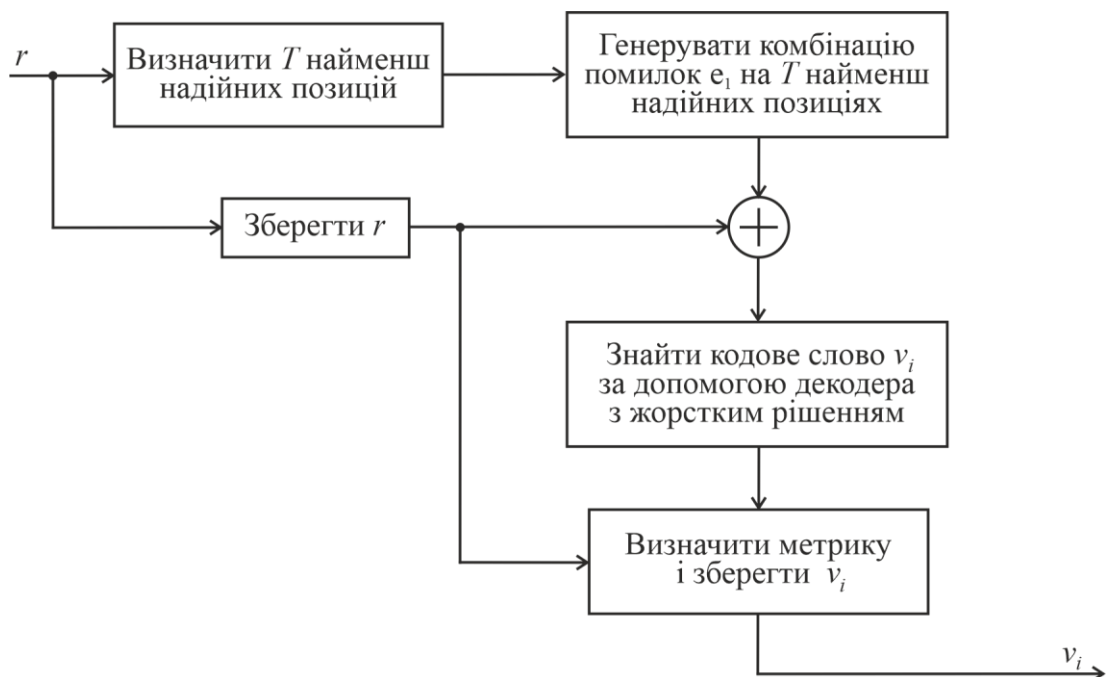


Рисунок 2.2 – Структурна схема алгоритму Чейза, тип 2.

Якщо це необхідно, алгоритм може формувати список кодових слів (в якості можливих рішень). Ця властивість алгоритму Чейза може бути

використана для формування м'якого виходу при ітеративном декодуванні блокових турбо кодів.

## 2.5 Декодування за упорядкованими статистиками

Алгоритм декодування за упорядкованими статистиками (ordered statistics decoding - OSD) аналогічний алгоритму Чейза в тому сенсі, що він теж створює список комбінацій помилок і використовує декодування з жорстким рішенням. Однак, на відміну від алгоритмів Чейза, які працюють на найменш надійних позиціях, OSD алгоритм створює список кодових слів на множині найбільш надійних позицій.

Припустимо, що для передачі двійкових сигналів по каналу з АБГШ використовується лінійний блоковий  $(N, K)$  код  $C$  з породжуючою матрицею  $G$  і мінімальною відстанню  $d_H \geq 2t + 1$ . Нехай  $c = (c_1, c_2, \dots, c_N)$  кодове слово з коду  $C$  і нехай  $r = (r_1, r_2, \dots, r_N)$  прийнята послідовність.

Як і в алгоритмі Чейза декодування починається з упорядкування компонент прийнятої послідовності по спаданню надійностей. Позначимо

$$y = (y_1, y_2, \dots, y_N),$$

впорядковану послідовність, в якій  $|y_1| \geq |y_2| \geq \dots \geq |y_N|$ . Це переупорядочення будемо називати перестановкою  $\lambda_1$  такою, що  $y = \lambda_1(r)$ .

Наступний крок алгоритму полягає в перестановці стовпців матриці:

$$G' = \lambda_1[G] = (g'_1, g'_2, \dots, g'_N)$$

де  $g'_j$  -  $j$ -й стовпець матриці  $G'$ .

Продовження алгоритму полягає в побудові найбільш надійного базису деякого еквівалентного коду. Починаючи з першого стовпця матриці  $G'$ ,

знаходяться перші  $K$  лінійно незалежних стовпців, яким відповідають найбільші надійності. Далі ці  $K$  стовпців використовуються як перші  $K$  стовпців нової матриці  $G''$  в порядку, відповідному їх надійності. Решта  $(N-K)$  стовпців теж упорядковуються в порядку убутання їх надійності. Цей процес назовемо другим перестановочним відображенням  $\lambda_2$  таким, що

$$G'' = \lambda_2[G'] = \lambda_2[\lambda_1[G]]$$

Застосовуючи відображення  $\lambda_2$  до послідовності  $y$ , отримуємо нову переупорядочену послідовність  $z$ :

$$z = \lambda_2[y] = (z_1, z_2, \dots, z_k, z_{k+1}, \dots, z_N),$$

$$\text{де } |z_1| \geq |z_2| \geq \dots \geq |z_k| \text{ і } |z_{k+1}| \geq |z_{k+2}| \geq \dots \geq |z_N|.$$

За допомогою елементарних операцій над рядками матриці  $G''$  отримуємо в систематичній формі матрицю  $G_I$ :

$$G_I = (I, P) = \begin{bmatrix} 1 & 0 & \dots & 0 & p_{1,1} & \dots & p_{1,N-K} \\ 0 & 1 & \dots & 0 & p_{2,1} & \dots & p_{2,N-K} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & p_{k,1} & \dots & p_{k,N-K} \end{bmatrix}$$

Легко бачити, що код, який генерується матрицею  $G_I$ , еквівалентний коду  $C$  в тому сенсі, що генеруються ті ж кодові слова з перестановкою кодових символів.

Наступний крок полягає в реалізації жорсткого декодування, що враховує  $k$  найбільш надійних позицій перевпорядкованої прийнятої послідовності  $z$ .

Позначимо  $u_a = (u_1, u_2, \dots, u_K)$  результат декодування. Відповідне кодове слово з коду  $C_1$  може бути знайдено як

$$v = u_a G_1 = (v_1, v_2, \dots, v_K, v_{K+1}, \dots, v_N)$$

Оцінка  $v_{HD}$  кодового слова коду  $C$  може бути отримана з  $v$  за допомогою зворотного відображення вигляду:

$$v_{HD} = \lambda_1^{-1}[\lambda_2^{-1}[v]] \quad (2.6)$$

На основі жорсткого рішення  $v$  алгоритм виконує повторні проходи (ітерації) доти, поки не буде досягнута оптимальність або достатня ефективність.

Ітеративний процес  $l$ -го порядку визначається наступним чином.

Для  $1 \leq i \leq l$  виконуються всі можливі заміни  $i$  з  $k$  найбільш надійних символів кодового слова  $v$ .

Ітерації можуть бути організовані таким чином, щоб мінімізувати кількість обчислень. Зокрема, для двійкових сигналів в гаусовому каналі достатньо обчислювати кореляцію між кодovими словами, що генеруються і перевпорядкованою прийнятою послідовністю. Таким чином, не потрібно обчислення дійсної послідовності  $x'$ . Все, що потрібно в цьому випадку, це складання прийнятих значень зі зміненими знаками, що відповідають змінам в послідовності  $v$  при генерації  $v''$ .

## 2.6 Декодування за мінімумом узагальненої відстані

У 1966 р Форне ввів декодування по мінімуму узагальненої відстані (МУВ). Основна ідея полягала в тому, щоб розширити поняття стирання, розділяючи прийняті значення за класами (рівнями) надійності. Стратегія

декодування аналогічна алгоритму Чейза, Тип-Ш, з використанням комбінацій стирань. У процесі декодування по МУВ вводяться додаткові стирання на  $d-1$  найменш надійних позиціях і перевіряється умова достатності для вибору оптимального рішення. Ітеративний процес триває до тих пір, поки не виконається ця умова або не буде введено максимальне число стирань.

Нехай  $C$  лінійний блоковий  $(N, K, d)$  код. Припустимо, що є декодер, що виправляє стирання і помилки, здатний виправляти будь-яку комбінацію  $e$  помилок і  $s$  стирань в межах коректуючої здатності коду, тобто  $2e + s \leq d - 1$ .

Нехай  $r = (r_1, r_2, \dots, r_N)$  прийняте слово на виході каналу. На  $i$ -ій позиції для переданого символу  $c_i$ , маємо оцінку  $r_i = (-1)^{c_i} + w_i$  де  $w_i$  - гаусова випадкова величина з нульовим середнім і дисперсією  $N_0/2$ ,  $i = 1, 2, \dots, N$ . Передбачається, що слово  $r$  нормалізовано таким чином, що всі його компоненти обмежені інтервалом  $[-1, +1]$  (за допомогою двобічного обмежувача). Як завжди, знаковий біт представляє собою жорстке (посимвольне) рішення щодо прийнятого слова,

$$z = (z_1, z_2, \dots, z_N), z_j = \text{sign}(r_j), 1 \leq j \leq N.$$

Надійності прийнятих з каналу величин упорядковуються, як в алгоритмах Чейза і OSD, і формується список індексів  $I_j$ ,  $j = 1, 2, \dots, N$ :

$$|r_{I_1}| \leq |r_{I_2}| \leq \dots \leq |r_{I_N}|.$$

На першому етапі декодування прийняте з жорсткими рішеннями слово вводиться в алгебраїчний декодер (МОР декодування застосовується до кодів БЧХ або РС) з виправленням помилок. Нехай  $v$  є результат такого

декодування. Кореляційний метрика слова  $v$  щодо прийнятого слова  $r$  дорівнює

$$v = \sum_{j=1}^n (-1)^{v_j} \cdot r_j$$

Якщо виконується така умова достатності

$$v > n - d, \quad (2.7)$$

то слово  $v$  вибирається як найбільш ймовірне і на цьому декодування закінчується.

В іншому випадку виконується наступний етап декодування. На другому етапі вводяться два стирання,  $s = 2$ , на позиціях з індексами  $I_1$  і  $I_2$  і виконується виправлення помилок і стирань. Далі обчислюється кореляційна метрика (2.6) між  $r$  і результатом декодування  $v$  і перевіряється умова (2.7).

Якщо це необхідно, то виконується черговий етап процедури. На кожному новому етапі додаються по два стирання,  $s = s + 2$ , на найменш надійних позиціях, поки їх кількість не перевищить максимально можливого ( $s_{\max} = d - 1$ ). Якщо в результаті всіх етапів декодування (включаючи  $s = 0$ ) не знайдено жодного кодового слова, то або фіксується відмова від декодування (фатальна помилка), або видається жорстке рішення  $z$ . Зауважимо, що в разі РС коду завжди можна відновити кодове слово по  $k$  найнадійніших позиціях, тобто з виправленням  $d - 1$  стирань.

Умова (2.7), використане в декодуванні по МОР, може бути покращено і застосовано до інших алгоритмів, що формує список рішень, таким як алгоритми Чейза і OSD. Ці алгоритми є прикладами спискового декодування. Критерій (2.7) є занадто жорстким, внаслідок чого часто відкидаються можливі кодові слова, серед яких можуть бути і найбільш ймовірні (тобто такі, які вибрав би декодер по максимуму правдоподібності). Відомі декілька

варіантів поліпшених (оптимізованих) умов достатності. Два з них представлені нижче без доказів. Для пояснення цих умов необхідно ввести деякі визначення.

Нехай  $x$  є двійкове кодове слово з фазовою модуляцією (2.2),  $x = m(v)$ , де  $v \in C$  і  $x_i = (-1)^{v_i}$ ,  $1 \leq i \leq N$ . Введемо наступні множини:  $S_e = \{i : \text{sign}(x_i) \neq \text{sign}(r_i)\}$  - множина позицій з помилками,  $U = [I_j, j = 1, 2, \dots, d]$  - множина найменш надійних позицій і множина  $T = \{i : \text{sign}(x_i) = \text{sign}(r_i), i \in U\}$ . Тоді розширена відстань або кореляційний дефект між кодовим словом  $v$  і прийнятим словом  $r$  визначається виразом:

$$d_e(v, r) = \sum_{i \in S_e} |r_i| \quad (2.8)$$

Покращений критерій вибору (оптимального) кодового слова заснований на верхній межі кореляційного дефекту (7.9) і на збільшенні потужності множин тестованих позицій. Розглянемо поліпшення умов, запропонованих Форні. Існує оптимальне кодове слово таке, що

$$d_e(x_{opt}, r) = \sum_{i \in T} |r_i|.$$

## 2.7 Спискове декодування

Спискове декодування (видача списку можливих рішень декодера) було введено Елайесом і Возенкратом. Спискове декодування використовують для поліноміальних кодів РС за межами їх конструктивної коректуючої здібності. Запропонований метод (алгоритм Судану), використовує інтерполяцію і факторизацію поліномів двох змінних над

розширенням базового поля. Алгоритм Судана можна розглядати як розвиток алгоритму Берлекемпа-Велч.

Алгоритми декодування з м'яким виходом (soft-output).

Існують алгоритми декодування, що формують на виході найбільш імовірну кодову послідовність або кодове слово (або список кодових слів). А також алгоритми, що оцінюють надійність символів для подальшої обробки. В області завадостійкого кодування алгоритми з м'яким виходом були введені в 1962 році за кодами з низькою щільністю перевірок на парність (LDPC - low-density parity-check code), які виконують проходи вперед - назад для обчислення надійності символів.

Передбачається, що лінійний блоковий код побудований за допомогою двійкового згорткового коду з пам'яттю  $m$  і швидкістю  $1/n$  і що сигнали передаються по каналу з АБГШ. Передбачається також, що згортковий кодер стартує з нульового початкового стану  $S_0^{(0)}$  і через  $N$  кроків (трелісних секцій) зупиняється в нульовому стані  $S_N^{(0)}$ .

## 2.8 Алгоритм Вітербі з м'яким виходом

Запропонована модифікація алгоритму Вітербі з видачею побітових оцінок надійності. Алгоритм Вітербі з м'яким виходом (soft-output viterbi algorithm - SOVA) обчислює надійності (або м'який вихід) інформаційних символів як логарифм відношення правдоподібності (log-likelihood ratio - LLR),

$$\Lambda(u_i) = \log \left( \frac{P(u_i = 1 | r)}{P(u_i = 0 | r)} \right),$$

(2.9)

де  $r$  - прийнята кодова послідовність.

Робота декодера SOVA може бути розділена на дві частини. У першій частині декодування виконується так само як і в звичайному алгоритмі Вітербі, який вибирає найбільш імовірну кодову послідовність  $v$ , що відповідає шляху (на ґратах) з максимальною (кореляційною) метрикою на  $n$ -му кроці. Також потрібно збереження метрик шляхів на кожному кроці і для кожного стану декодера. Ці метрики будуть потрібні в останній частині алгоритму для обчислення м'яких виходів. У другій частині SOVA алгоритм Вітербі виконується в зворотному напрямку, тобто обчислюються метрики і шляхи, що починаються з  $N$ -ої секції решітки і закінчуються в  $i$ -ою секцією,  $i \geq 0$ . Зауважимо, що в другому проході не потрібно збереження кращих шляхів, а необхідно тільки збереження метрик для кожного стану решітки. Таким чином, після другого проходу для кожного вузла решітки є дві метрики: одна для лівої частини шляху, а друга для правої частини шляху, що проходить через нього. Потім для кожної секції решітки обчислюються м'які виходи.

Позначимо  $M_{\max}$  (кореляційну) метрику найбільш правдоподібної послідовності  $v$ , знайденої алгоритмом Вітербі. Умовна імовірність відповідної інформаційної послідовності  $i$  при заданій прийнятій послідовності, інакше апостеріорна імовірність (APP), залежить від  $M_{\max}$  оскільки справедлива наступна оцінка:

$$P\{u | r\} = P\{v | r\} = e^{M_{\max}} \quad (2.10)$$

Без втрати загальності можна вважати, що оцінка APP щодо інформаційного біта  $i$ , має вигляд:

$$P\{u_i = 1 | r\} \approx e^{M_i^{(1)}},$$

де  $M_i(1) \triangleq M_{\max}$ . Позначимо  $M_i(0)$  максимальну метрику шляху асоційованого з доповненням інформаційного символу  $u_i$ . Легко показати, що

$$\Lambda(u_i) \approx M_i(1) - M_i(0) \quad (2.11)$$

Таким чином, для  $i$ -ої секції решітки м'який вихід може бути отриманий з різниці між максимальною метрикою шляхів на решітці, для яких  $u_i = 1$ , і максимальної метрикою шляхів з  $u_i = 0$ .

На стадії формування м'якого виходу алгоритму SOVA для  $i$ -ої секції визначається найбільш правдоподібне значення інформаційного символу  $u_i = a$ ,  $a \in \{0, 1\}$ , і відповідна йому максимальна метрика (яка була знайдена на прямому проході алгоритму Вітербі) встановлюється рівною  $M_i(u_i)$ . Метрика найкращого шляху для доповнення  $M_i(u_i \oplus 1)$  визначається наступним чином:

$$M_i(v_i \oplus 1) = \min \left\{ M_f(S_{i-1}^{k_1}) + BM_i^{(b_1)}(v_i \oplus 1) + M_b(S_i^{k_2}) \right\},$$

де  $k_1, k_2 \in \{0, 1, 2, \dots, 2^m - 1\}$

-  $M_f(S_{i-1}^{k_1})$  метрика шляху, що вижив на прямому проході алгоритму для  $(i - 1)$ -ої секції та стану  $S^{k_1}$ ,

-  $BM_i^{(b_1)}(u_i \oplus 1)$  метрика ребра для інвертованого інформаційного символу, асоційованого з переходом зі стану  $S^{k_1}$  в  $S^{k_2}$ ,

-  $M_b(S_i^{k_2})$  метрика шляху, що вижив на зворотному проході моменту  $i$  і стану  $S^{k_2}$ . Остаточню, м'який вихід обчислюється як різниця:

$$\Lambda(u_i) = M_i(1) - M_i(0) \quad (2.12)$$

У декодері, що реалізує алгоритм SOVA, алгоритм Вітербі використовується двічі. На прямому проході він працює як звичайний декодер Вітербі з тією лише відмінністю, що повинні зберігатися метрики шляхів для кожного стану решітки. На зворотному проході алгоритму Вітербі зберігаються тільки метрики для всіх станів декодера без збереження шляхів, що вижили. Зауважимо, що виконання зворотного проходу і обчислення м'яких виходів можна поєднати в одному процесі. Особливу увагу необхідно приділити нормалізації метрик для кожного стану декодера в обох напрямках проходу.

Декодер SOVA може бути використаний в режимі ковзного вікна (sliding window decoder), як звичайний декодер Вітербі. За рахунок збільшення обсягу обчислень декодер може працювати безперервно, тобто не в блоковому режимі, без періодичного примусового повернення декодера в нульовий стан. У цьому варіанті прямий і зворотний проходи декодера Вітербі, прямий і зворотний проходи алгоритму SOVA і обчислення м'яких виходів виконуються з використанням різних блоків пам'яті.

## **2.9 Алгоритм декодування за максимумом апостеріорної ймовірності**

Алгоритм BCJR [BCJR] є оптимальним алгоритмом посимвольного максимуму апостеріорної ймовірності (MAP) декодування лінійних блокових кодів, який мінімізує ймовірність помилки на символ. Мета MAP декодування полягає в обчисленні апостеріорних ймовірностей інформаційних символів при заданій прийнятій послідовності  $r$  за правилом:

$$\Lambda(u_i) = \log \left( \frac{P(u_i = 1 | r)}{P(u_i = 0 | r)} \right). \quad (2.13)$$

Розглянемо алгоритм MAP декодування в версії близькою до оригіналу.

Переходи між станами на решітці мають ймовірності:

$$P\{S_i^{(m)} | S_{i-1}^{(m')}\} \quad (2.15)$$

Для кодових символів  $v_i$

$$q_i(x_i | m', m) \triangleq P\{x_i = x | S_{i-1}^{(m')}, S_i^{(m)}\}, \quad (2.16)$$

Де  $x = \pm 1$ ,  $x_i = m(v_i) = (-1)^{v_i}$ ,  $0 < i < N$ .

Послідовність  $x$  передається по каналу з АБГШ і приймається як послідовність  $r$  з перехідними ймовірностями

$$P\{r | x\} = \prod_{i=1}^N p(r_i | x_i) = \prod_{i=1}^N \prod_{j=0}^{n-1} p(r_{i,j} | x_{i,j}) \quad (2.17)$$

де ймовірність  $p(r_{i,j} | x_{i,j})$  визначена в (2.1).

Позначимо  $B_i^{(j)}$  множину ребер, що зв'язує стан  $S_{i-1}^{(m')}$  з  $S_i^{(m)}$  таким чином, що відповідний інформаційний біт  $u_i = j$ ,  $j \in \{0, 1\}$ . Тоді

$$P(u_i = j | r) = \sum_{(m', m) \in B_i^{(j)}} P(S_{i-1}^{(m')}, S_i^{(m)}, r) \triangleq \sum_{(m', m) \in B_i^{(j)}} \sigma_i(m', m). \quad (2.18)$$

Величина  $\sigma_i(m', m)$  в (7.20) дорівнює

$$\sigma_i(m', m) = \alpha_{i-1}(m') \gamma_i^{(j)}(m', m) \beta_i(m), \quad (2.19)$$

де спільна ймовірність  $\alpha_i(m) \triangleq P\{S_i^{(m)}, r\}$  визначена рекурсивно як

сума

$$\alpha_i(m) = \sum_{m'} \alpha_{i-1}(m') \sum_{j=0}^1 \gamma_i^{(j)}(m', m) \quad (2.20)$$

Зазвичай її називають метрикою прямого проходу.

Імовірність  $\gamma_i^{(j)}(m', m) \triangleq P\{S_i^{(m)}, r | S_{i-1}^{(m')}\}$  задається наступною сумою

$$\gamma_i^{(j)}(m', m) = \sum p_i(m | m') P\{x_i = x | S_{i-1}^{(m')}, S_i^{(m)}\} P\{r_i | x\} \quad (2.21)$$

Імовірність  $p_i(m | m') = P\{S_i^{(m)} | S_{i-1}^{(m')}\}$ . У каналі з АБГШ (2.23) можна записати в наступному вигляді

$$\gamma_i^{(j)}(m', m) = P\{u_i = j\} \delta_{ij}(m, m') \exp\left(-\frac{1}{N_0} \sum_{q=0}^{n-1} (r_{i,q} - x_{i,q})^2\right), \quad (2.22)$$

де  $\delta_{ij}(m, m') = 1$ , якщо  $(m, m') \in B_i^{(j)}$ , і  $\delta_{ij}(m, m') = 0$  в іншому випадку.

Імовірність  $\gamma_i^{(j)}(m', m)$  називають метрикою ребра.

Умовна ймовірність  $\beta_i(m) \triangleq P\{r_f | S_i^{(m)}\}$  визначається як

$$\beta_i(m) = \sum_{m'} \beta_{i+1}(m') \sum_{j=0}^1 \gamma_i^{(j)}(m', m) \quad (2.23)$$

Цю величину називають метрикою зворотного проходу.

Комбінуючи (2.22), (2.21), (2.20), (2.19) і (2.11), отримуємо м'який вихід логарифма відношення правдоподібності для символу  $u_i$  у вигляді:

$$\Lambda(u_i) = \log \left( \frac{P\{u_i = 1 | r\}}{P\{u_i = 0 | r\}} \right) = \log \left( \frac{\sum_m \sum_{m'} \alpha_{i-1}(m') \gamma_i^{(1)}(m', m) \beta_i(m)}{\sum_m \sum_{m'} \alpha_{i-1}(m') \gamma_i^{(0)}(m', m) \beta_i(m)} \right) \quad (2.24)$$

Жорстке рішення дорівнює  $(\hat{u}_i) = \text{sign}(\Lambda(u_i))$ , а надійність рішення  $u_i$  є  $|\Lambda(u_i)|$ . Виведені вище рівняння можна інтерпретувати в такий спосіб. У цьому алгоритмі декодування може бути застосований двохпрохідний алгоритм Вітербі, подібно до того, як це робиться в декодері SOVA. На прямому проході для заданої ймовірності зміни стану в момент  $i$  обчислюється спільна ймовірність прийнятої послідовності до моменту  $i$  і стану в момент  $i$ . На зворотному проході обчислюється ймовірність прийнятої послідовності від моменту  $i + 1$  до моменту  $N$  при заданому стані в момент  $i$ . Тоді м'який вихід залежить від спільної ймовірності зміни станів і прийнятого символу в момент  $i$ .

### Висновки до другого розділу

1. Застосування методів м'якого декодування дозволяє знизити відношення сигнал / шум на 2 дБ при збереженні заданої достовірності передачі інформації в порівнянні з методами жорсткого декодування.

2. Методи м'якого декодування набули широкого вжитку з появою каскадних кодів, що допускають ефективне ітеративне декодування з обміном м'якими рішеннями на кожній ітерації

3. Алгоритми, декодування м'якого рішення демодулятора вимагають більших обчислювальних витрат, але забезпечують зниження ймовірності виникнення помилки в декодованому повідомленні.

## 3 КОДИ ЩО ІТЕРАТИВНО ДЕКОДУЮТЬСЯ

### 3.1 Поняття ітеративного декодування

Ітеративне декодування можна визначити як метод, який багаторазово використовує алгоритм декодування з м'яким виходом для того, щоб зменшити ймовірність помилки схеми завадостійкого кодування і наблизитися до декодування за максимумом правдоподібності при меншій складності обчислень. Якщо правильно обраний (побудований) базовий код, що виправляє помилки, то збільшення числа ітерацій призводить до підвищення завадостійкості.

Технологія ітеративного декодування з'явилася в 1954 році. Пізніше, в 60-х роках, важливий внесок у цю технологію зробили Галлагер [Gal] і Мессі [Masl], ітеративне декодування стали називати імовірнісним декодуванням (probabilistic decoding). Головна ідея була тією ж самою, що і сьогодні - максимізувати апостеріорну ймовірність переданого символу.

Розглянемо коди з ітеративним декодуванням. Такі кодові схеми були названі турбо-подібними кодами. Спираючись на застосування ітеративних алгоритмів декодування, кодові конструкції можна розділити на два класи:

- коди добутки. Прикладами кодів цього класу є паралельні каскадні коди або турбо коди і послідовні каскадні коди;
- коди з низькою щільністю перевірок (low-density parity-check codes).

Це лінійні коди, перевірочні матриці яких мають низьке відношення числа ненульових елементів до загальної кількості елементів. В обох класах компонентні коди можуть бути згортковими або блоковими, з систематичним або несистематичним кодуванням, а також будь-яка їх комбінація. Вивчення ітеративної техніки декодування, почнемо з обговорення фундаментальної структури турбо кодів.

Турбо коди були введені в 1993 році. Кодова конструкція з швидкістю  $1/2$ , що складається з двох рекурсивних систематичних згорткових (RSC)

кодів з 16 станами і перемешувача розміром 256x256, забезпечила ймовірність помилки (BER) порядку  $10^{-5}$  при передачі двійкових сигналів в гаусовому каналі і відношенні сигнал-шум,  $E_b/N_0 = 0,7$  дБ.

Основні елементи схеми турбо кодування, полягають у наступному:

- схема кодування. Схема добутку кодів (паралельного каскадного коду) з компонентними рекурсивними систематичними згортковими кодами;
- оцінка надійності. Для композитних кодів застосовуються декодери (з м'яким входом і м'яким виходом) по максимуму апостеріорної ймовірності (MAP), щоб генерувати логарифмічні відношення правдоподібності;
- ітеративне декодування. Надійності деяких символів (у вигляді зовнішньої інформації) передаються через зворотний зв'язок від зовнішнього декодера (по стовпцях) до внутрішнього (по рядках) і навпаки;
- випадкове перемішування. Між двома кодерами використовується випадковий перемешувач великої довжини. Його призначення полягає, головним чином, в тому, щоб на кожній ітерації забезпечити незалежність оцінок інформаційних символів, одержуваних компонентними MAP декодерами;

Розглянемо апостеріорне логарифмічне відношення правдоподібності (LLR) для інформаційного символу.

$$\Lambda(u_i) = \log \left( \frac{P\{u_i = 1 | r\}}{P\{u_i = 0 | r\}} \right) \quad (3.1)$$

Позначимо  $C_i(0)$  і  $C_i(1)$  множини модульованих послідовностей  $x = m(v) \in m(C)$  таких, що на  $i$ -ій позиції послідовності  $v \in C$  знаходиться символ 0 або 1, відповідно. Нагадаємо, що модуляція є взаємно однозначне відображення двійкових символів,  $v \in \{0, 1\}$ , на  $x \in \{+1, -1\}$ , за виразом (2.1). Тоді LLR (3.1) для символу може бути записана наступним чином

$$\Lambda(u_i) = \log \left( \frac{\sum_{x \in C_i(1)} \prod_{i=1}^N p(r_i, x_i)}{\sum_{x \in C_i(0)} \prod_{i=1}^N p(r_i, x_i)} \right) =$$

$$= \log \left( \frac{p(r_1 | x_1 = -1) P(x_1 = -1) \sum_{x \in C_i(1)} \prod_{i=1}^N p(r_i, x_i)}{p(r_1 | x_1 = +1) P(x_1 = +1) \sum_{x \in C_i(0)} \prod_{i=1}^N p(r_i, x_i)} \right) = \quad (3.2)$$

Звідси випливає, що LLR для інформаційного символу може бути представлено сумою

$$\Lambda(u_i) = \Lambda_{ci} + \Lambda_a(u_i) + \Lambda_{i,e}(C), \quad (3.3)$$

де  $\Lambda_{ci}$  називають LLR каналу, величину  $\Lambda_a(u_i)$  називають апіорним LLR інформаційного символу і  $\Lambda_{i,e}(C)$  називають зовнішнім (extrinsic) LLR, що характеризує обмеження, які накладаються іншими інформаційними символами коду. Значення цих величин визначаються наступними співвідношеннями:

$$\Lambda_{ci} = \log \left( \frac{P\{r_i | x_i = -1\}}{P\{r_i | x_i = +1\}} \right) \quad (3.4)$$

$$\Lambda_{ci} = (-1)^{r_i} \log \left( \frac{1-p}{p} \right), \text{ Для ДСК з параметром } p ;$$

$$\Lambda_{ci} = \frac{4}{N_0} r_i, \text{ для каналу з АБГШ } (\sigma_n^2 = N_0 / 2) ;$$

$$\Lambda_{ci} = \frac{4}{N_0} a_i r_i, \text{ для загальних релєєвських завмирань.}$$

$$\Lambda_a(u_i) = \log \left( \frac{P\{x_i = -1\}}{P\{x_i = +1\}} \right) = \log \left( \frac{P\{u_i = 1\}}{P\{u_i = 0\}} \right) \quad (3.5)$$

$$\Lambda_{i,e}(C) = \log \left( \frac{\sum_{x \in C_i(1)} \prod_{i=1}^N p(r_i, x_i)}{\sum_{x \in C_i(0)} \prod_{i=1}^N p(r_i, x_i)} \right) \quad (3.6)$$

Припустимо, що двійкові сигнали передаються по каналу з АБГШ і що в якості компонентних кодерів використовуються двійкові рекурсивні систематичні кодери з швидкістю  $1/n$ . У ітеративній процедурі декодування зовнішня (iextrinsic) інформація  $\Lambda_{i,e}(C)$  може бути введена в декодер (через зворотний зв'язок) і використана в якості апіорної ймовірності на наступній ітерації. У процедурі декодування, що використовує прямий і зворотний проходи по решітці, зовнішнє LLR може бути записано в наступному вигляді

$$\Lambda_{i,e}(C) = \log \left( \frac{\sum_{(m,m') \in B_i(1)} \alpha_{i-1}(m') \xi_1(m', m) \beta_i(m)}{\sum_{(m,m') \in B_i(0)} \alpha_{i-1}(m') \xi_1(m', m) \beta_i(m)} \right), \quad (3.7)$$

де імовірності  $\alpha_i(m)$  і  $\beta_i(m)$  визначені в (7.22) і (7.25), відповідно.

Для обчислення зовнішнього LLR необхідно ввести модифіковану метрику ребра

$$\xi_1(m', m) = \delta_{ij}(m, m') \exp \left( \frac{E}{N_0} \sum_{q=1}^{n-1} r_{i,q} x_{i,q} \right) \quad (3.8)$$

де, як зазвичай,

$$\delta_{ij}(m, m') = \begin{cases} 1, & (m, m') \in B \\ 0, & \text{інше} \end{cases}$$

Зовнішнє (логарифмічне) відношення правдоподібності (LLR) для  $i$ -ої інформаційної позиції не містить будь-якої змінної, безпосередньо пов'язаної з  $u_i$ ,  $i = 1, 2, \dots, N$ . Оскільки кодування за визначенням систематичне, то всі мітки ребер решітки мають вигляд  $(u_i, u_{i,1}, \dots, u_{i,n-1})$  і, отже, підсумовування в виразі для модифікованої метрики починається з  $q = 1^3$ .

У загальному випадку двовимірному кодування перший декодер (по рядках) обчислює  $\Lambda_{i,j}^{(1)}(C)$  і передає його другому декодеру (по стовпцях) як апріорну інформацію  $\Lambda_a^{(2)}(u_i)$ , необхідну для обчислення LLR інформаційного символу  $u_i$ . Іншими словами, зовнішня інформація дає (м'який) вихід, що залежить від тих (м'яких) входів (надійностей), які не пов'язані безпосередньо з інформаційним символом  $u_i$ . Ітеративне декодування складових кодів є темою наступного розділу.

### 3.2. Каскадні коди

У цьому розділі представлені схеми кодування каскадних кодів з перемежувачами. Ці схеми дозволяють використовувати прості декодери компонентних кодів. У ітеративній процедурі посимвольного MAP декодери компонентних кодів можуть обмінюватися зовнішніми оцінками LLR, істотно підвищуючи надійність оцінки інформаційного символу з збільшенням числа ітерацій.

Розглянемо паралельну схему обробки: турбо коди. На рисунку 79 показана структурна схема кодера паралельного каскадного коду, більш відомого як турбо код. Використовуються два кодера, кожен з яких обчислює тільки перевіірочні символи. Наприклад, якщо використовуються RSC коди

швидкістю  $1/2$ , то кодери відповідають дробовій частині  $g_1(D)/g_0(D)$  поліноміальної породжуючої матриці  $G(D)$ . Входами кодеров є послідовності  $u$  і  $P_u$ , де  $P$  – матриця перестановок, асоційована з перемишувачем. Вихід кодера, що відповідає вхідному символу  $u_i$ , має вигляд  $(u_i, v_{P1,t}, v_{P2,t})$ , де  $v_{P1,t}$  і  $v_{P2,t}$  представляють перевіірочні символи для  $i = 1, 2, \dots, N$ ,  $N$  довжина блоку.

В оригінальній роботі в конструкції турбо коду були використані RSC коди. Для вивчення цієї схеми припустимо, що в якості компонентних кодів використані два двійкових систематичних лінійних блокових  $(n_i, k_i)$  коди,  $i = 1, 2$ . Швидкість каскадного коду дорівнює :

$$R = \frac{K}{N} = \frac{k_1 k_2}{n_1 n_2 - (n_1 - k_1)(n_2 - k_2)} = \frac{R_1 R_2}{R_1 + R_2 - R_1 R_2} \quad (3.9)$$

де  $R_i = k_i / n_i$ ,  $i = 1, 2$ . Позначимо  $G_i = (I_i | P_i)$  породжуючу матрицю коду  $C_i$ . Тоді породжуюча матриця паралельного каскадного коду  $C_{PC}$  може бути приведена до наступного вигляду

$$G_{PC} = \left( I_{k_1 k_2} \begin{pmatrix} P_1 & & & \\ & P_1 & & \\ & & 0 & \\ & & & P_1 \end{pmatrix} \Pi \begin{pmatrix} P_2 & & & \\ & P_2 & & \\ & & 0 & \\ & & & P_2 \end{pmatrix} \right) = (I_{k_1 k_2} | P'_1 | P'_2) \quad (3.10)$$

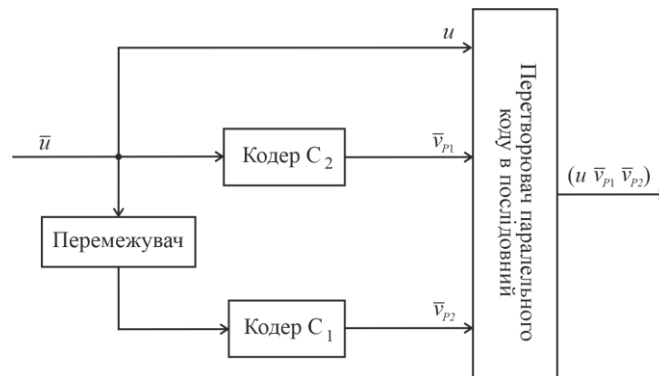


Рисунок 3.1 – Структура кодера паралельного складеного коду.

Матриця перестановок, асоційована з перемежувачем, і  $P_i$  - перевірна частина породжуючої матриці коду  $C_i$ ,  $i=1, 2$ . Підматриця  $P_1$  з'являється в блочно-діагональній підматриці  $P'_1$  матриці  $G_{PC}$  рівно  $k_2$  разів, а підматриця  $P_2$  з'являється  $k_1$  разів в підматриці  $P'_2$ . Інші елементи підматриць  $P'_1$  і  $P'_2$  дорівнюють нулю. Звідси випливає, що кодове слово коду  $C_{PC}$  має вигляд  $(u | uP'_1 | uP'_2)$ .

Розглянемо коди:  $C_1(2, 1, 2)$  і  $C_2(3, 2, 2)$ . Тоді  $k_1k_2 = 2$ . У цьому випадку існує єдина матриця перестановок  $P$ . Перевірочні підматриці і матриця перестановок визначаються виразом:

$$P_1 = (1), P_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \Pi = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

відповідно. Породжуюча матриця  $(5, 2)$  паралельного каскадного коду  $C_{PC}$  має вигляд:

$$\begin{aligned} G_{PC} &= \left( \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} (1) & 0 \\ 0 & (1) \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} (1) \\ (1) \end{pmatrix} \right) = \\ &= \left( \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} (1) & 0 \\ 0 & (1) \end{pmatrix} \begin{pmatrix} (1) \\ (1) \end{pmatrix} \right) = \\ &= \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} \end{aligned} \quad (3.11)$$

Зауважимо, що мінімальна відстань цього коду  $d_{PC} = 3$ , тобто менше ніж для звичайного каскадного коду, що складається з кодів  $C_1$  і  $C_2$ .

Один із способів інтерпретації паралельного каскадного (турбо) коду з точки зору лінійних блокових кодів полягає в тому, що він розглядається як

перфорований код - добуток, у якого видалені перевірочні символи, що відповідають перевіркам. Єдина істотна відмінність між турбо кодом і блоковим кодом-добутком полягає в тому, що його перемежувач не є прямокутним, тобто із записом за рядками і зчитуванням за стовпцями, а вносить достатній безлад в інформаційну послідовність для того, щоб працювала ітеративна процедура. Розвиваючи цю інтерпретацію, можна розглядати турбо код як узагальнений каскадний код (УКК), замінюючи перфорацію еквівалентним множенням на двійкову матрицю з подальшим підсумовуванням кодових слів. Позначимо  $M_1$  матрицю розміру  $1 \times N$  з  $n_1 k_2$  послідовними одиницями, за якими слідують  $N - n_1 k_2$  нулів. Позначимо  $M_2$  іншу матрицю розміром  $1 \times N$

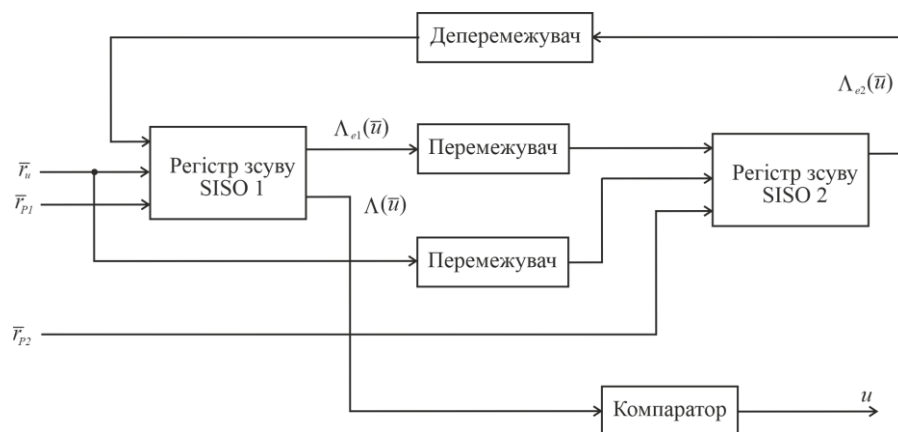


Рисунок 3.2 – Структурна схема ітеративного декодера для паралельного каскадного коду.

з  $N - (n_2 - k_2)k_1$  нулями, за якими ідуть  $(n_2 - k_2)k_1$  одиниць.

### 3.3 Ітеративне декодування паралельного каскадного коду

Базова структура ітеративного декодера для паралельної схеми каскадного кодування з двома компонентними кодами показана на рис. 32. Кожна ітерація складається з двох фаз, по одній на компонентний декодер.

Перша фаза. На першій ітерації, в першій фазі SISO декодер (декодер з м'яким входом і м'яким виходом) першого компонентного коду обчислює апостеріорні LLR в припущенні, що всі символи однаково імовірні, тобто  $\Lambda_a(u) = 0$ . Цей декодер обчислює зовнішню інформацію для кожного інформаційного символу,  $\Lambda_{e1}(u)$ , використовуючи ту частину прийнятої послідовності, яка відповідає перевірочним символам,  $r_{p1}$ , і віддає результат другому SISO декодеру.

Друга фаза. У другій фазі першої ітерації процедури декодування перемішані елементи зовнішньої інформації від першого декодера використовуються як апіорні LLR, тобто  $\Lambda_a(u) = \Pi\Lambda_{e1}(u)$ . Потім обчислюється зовнішня інформація  $\Lambda_{e2}(u)$  на основі тієї частини прийнятої послідовності, яка відповідає перевірочним символам другого компонентного коду,  $r_{p2}$ , завершуючи, таким чином, першу ітерацію декодування. У цей момент може прийматись рішення про інформаційні символи, засноване на значеннях апостеріорних LLR  $\Lambda(u)$ .

На наступних ітераціях перший декодер використовує депережені елементи зовнішньої інформації від другого декодера,  $\Pi^{-1}\Lambda_{e2}(u)$ , як апіорні LLR для обчислення м'якого виходу (тобто апостеріорного LLR)  $\Lambda(u)$ . Ця процедура може повторюватися до тих пір, поки не виконається умова зупинки, або реалізується задане максимальне число ітерацій. Зауважимо, що формування рішень про інформаційні символи після першого декодера дозволяє прибрати один депережувач.

Ітеративне MAP декодування може бути спрощено за рахунок апроксимації функції "log" функцією "шах". Крім того, є ітеративна версія алгоритму SOVA, в якій по суті обчислюється апостеріорне LLR  $\Lambda(u)$  і після віднімання LLR каналу,  $\Lambda_c$ , і зовнішнього LLR,  $\Lambda_{e1}(u)$ , від іншого декодера формується зовнішнє LLR.

Розглянемо ефективність каскадних кодів. На рисунку 78 видно, що зі збільшенням числа ітерацій ймовірність помилки зменшується. Зазвичай для

схем турбо кодування фактом є монотонне зниження швидкості росту виграшу від кодування зі збільшенням числа ітерацій. Збільшення числа ітерацій від 2 до 6 зменшує необхідну SNR на 1,7дБ, тоді як додаткові ітерації з 6 до 18 збільшують виграш від кодування тільки на 0,3 дБ.

З моменту відкриття турбо кодів досягнутий певний прогрес в розумінні поведінки ймовірності помилки цих кодів. В процесі написання цієї книги, мабуть, постала єдина думка серед дослідників про причини, за якими турбо коди мають настільки чудові характеристики: Турбо коди мають розподіл ваг, яке на великих інтервалах збігається з розподілом для випадкових кодів.

Рекурсивне згорткове кодування з правильно вибраним перемеренням перетворює більшу частину повідомлень малої ваги в кодові слова великої ваги.

Систематичні кодери дозволяють ефективно поєднувати ітеративну техніку з MAP декодуванням. Оцінки (початкові) інформаційних символів доступні прямо з каналу зв'язку.

Крім того, зовсім недавно (кінець 2001) з'явилося безліч цікавих напів-аналітичних методів для вивчення збіжності ітеративних алгоритмів декодування, заснованих на еволюції щільності (ймовірності), на гаусовій апроксимації і взаємній інформації або на передавальних характеристиках SNR (після кожної ітерації).

### **3.4 Використання пристроїв перемішування**

Пристрій перемішування - перемерувач (або інтерлівер) є найбільш відповідальною частиною турбо коду для досягнення високої ефективності при ітеративному декодуванні. У конструкції турбо коду перемерувач служить для досягнення трьох цілей: (1) побудова дуже довгих кодів з розподілом ваг, близьким до розподілу для випадкових кодів, (2) забезпечення, на скільки це можливо, декореляції вхідних LLR декодера

SISO (з м'якими входом і виходом) в ітеративній процедурі, (3) правильне термінування кодової решітки (в заданому стані) після передачі коротких або середніх по довжині повідомлень з тим, щоб уникнути крайових ефектів, які могли б збільшити частку шляхів малої ваги на кодовій решітці компонентних кодів. Щоб підкреслити особливу роль інтерлівера, розглянемо блоки даних (фрейми) порівняно невеликої довжини, наприклад, до однієї тисячі символів. Відома величезна кількість публікацій, присвячених конструюванню перемежувачів для турбо кодів. У цьому розділі ми наведемо короткий опис основних типів перемежувачів з посиланнями на літературу.

У 1970 були введені декілька типів оптимальних інтерліверів. Зокрема,  $(n_1, n_2)$  перемежувач був визначений як пристрій, який «перевпорядковує послідовність таким чином, щоб ніякі послідовності суміжних  $n_2$  символів не містили будь-який набір символів, рознесений менш ніж на  $n_1$  символів у початковому порядку».

Позначимо  $\dots a_{l_1}, a_{l_2}, a_{l_3}, \dots$  вихідну послідовність  $(n_1, n_2)$  перемежувача, де  $l_1, l_2, \dots$  індекси (позиції) цих символів у вхідній послідовності. Тоді попереднє визначення зводиться до наступних умов:

$$|l_i - l_j| \geq n_1$$

завжди, коли

$$|i - j| < n_1$$

Затримка інтерлівера і деінтерлівера в сукупності дорівнює затримці, передбаченої для виконання цих операцій. Іншим важливим параметром пристрою перемеження є необхідний обсяг пам'яті. Ремсі ввів чотири типи  $(n_1, n_2)$  перемежувачів, які оптимальні за затримкою і пам'яттю. Ці пристрої відомі як перемежувачі Ремсі. Форне запропонував пристрій такої ж

конструкції, як  $(n_1, n_2)$  інтерлівер Ремсі, що називають згортковим інтерлівером.

Зараз відомо декілька нових підходів до аналізу та побудови інтерліверів. Один з них заснований на випадковому перемешенні і його властивості розсіювання. Ця ідея була вперше запропонована з простим алгоритмом побудови S- випадкових перемешувачів, який полягає в наступному: генерувати випадкові числа в інтервалі  $[1, M]$  і використовувати обмеження на відстань перемешення. Це обмеження, очевидно еквівалентно визначенню  $(S_2, S_1 + 1)$  інтерлівера Ремсі. Далі були введені додаткові обмеження (засновані на емпіричній кореляції між послідовними значеннями зовнішніх LLR) для того, щоб управляти вибором перестановок позицій символів на виході перемешувача, Інший підхід до побудови інтерліверів полягає в тому, що розглядається структура турбо кодера в цілому. Для цього обчислюються його мінімальна відстань і коефіцієнт помилок (число кодових послідовностей на мінімальній відстані). Цей шлях дає точну оцінку зони насичення (error floor) функції ймовірності помилки для середніх і високих значень SNR.

### 3.5 Каскадні коди послідовного типу

Каскадні коди послідовного типу будують за структурною схемою приведеною на рис. 3.3.



Рисунок 3.3 – Структурна схема послідовного каскадного кодера

Приведений кодер для двох лінійних кодів. Послідовна схема з двох лінійних кодів легко інтерпретується як каскадний код (добуток кодів).

Зауважимо, що на відміну від турбокоду в послідовній конструкції не видаляються будь-які перевірочні символи.

Кодер послідовного каскадного коду має ту ж структуру, що і каскадний код. зовнішній  $(n_1 = p_1, k_1 = k, d_1)$  код  $C_1$  швидкості  $R_1 = k / p_1$  і внутрішній  $(n_2 = p_2, k_2 = k, d_2)$  код  $C_2$  швидкості  $R_2 = k / p_2$ . Ці коди пов'язані таким же чином, як і в добутку блокових кодів з включенням блокового інтерлівера довжини  $L = mp$ . Як і раніше,  $m$  кодових слів довжини  $p$  записуються в перемежувач, з якого зчитуються в іншому порядку, заданому матрицею перестановок  $\Pi$ . Послідовність  $L$  біт на виході інтерлівера посилається блоками довжини  $p$  в кодер внутрішнього коду  $C_2$ . Швидкість каскадного  $(N, K, d_1 d_2)$  коду  $C_{SC}$  дорівнює  $R_{SC} = k / n$ , де  $N = nm$  і  $K = km$ .

Породжуюча матриця коду  $C_{SC}$  може бути записана як добуток породжуючої матриці коду  $C_1$ , матриці перестановок  $\Pi$  розміром  $k_2 \times n_1$  і породжуючої матриці коду  $C_2$  наступним чином:

$$G_{SC} = \begin{pmatrix} G_1 & & & \\ & G_1 & & \\ & & 0 & \\ & & & G_1 \end{pmatrix} \Pi \begin{pmatrix} G_2 & & & \\ & G_2 & & \\ & & 0 & \\ & & & G_2 \end{pmatrix} = (G'_1 | \Pi | G'_2)$$

Тут  $G'_i$ , є породжуючою матрицею коду  $C_i$ ,  $i = 1, 2$ . Число появ матриці  $G_1$  в блочно діагональній матриці  $G'_1$  дорівнює  $k_2$ , а число появ  $G_2$  в  $G'_2$  дорівнює  $n_1$ . Всі інші елементи матриць  $G'_i$ , дорівнюють нулю.

Наступні правила були створені для вибору компонентних згорткових кодів в послідовних схемах каскадного кодування:

- Внутрішній код – рекурсивний систематичний згортковий (RSC) код.
- Зовнішній код повинен мати велику мінімальну відстань.
- Зовнішній код може бути несистематичним згортковим кодом.

Останній критерій необхідний для того, щоб мінімізувати число кодових слів мінімальної ваги (ця характеристика відома як експонента помилки (error exponent)) а також вагу вхідних послідовностей, що породжують кодові слова мінімальної ваги.

### **3.6 Ітеративне декодування послідовних схем каскадного кодування**

Повертаючись до рис. 84, відзначимо, що несистематичність зовнішнього згорткового коду означає неможливість отримання зовнішньої інформації з SISO декодера. Отже, в цьому випадку алгоритм ітеративного декодування відрізняється від того, який використовувався для турбо кодів. Якщо для турбо кодів оновлювалися LLR тільки інформаційних символів, то тут неминуче оновлюються LLR як інформаційних, так і кодових символів. Робота м'якого декодера внутрішнього коду не змінюється. Однак, для зовнішнього м'якого декодера апріорні LLR завжди встановлюються нульовими. Значення LLR кодових символів обчислюються і пересилаються м'якому декодеру внутрішнього коду після операції інтерлівінга як апріорні LLR для наступної ітерації. Як і для турбо кодів в цьому випадку існує  $\max$ -log-MAP версія ітеративного алгоритму. Крім того можлива модифікація алгоритму SOVA як апроксимації MAP декодування в ітеративній процедурі декодування кодів-добутків.

### **3.7 Добуток блокових кодів**

Хоча турбо коди і послідовні схеми кодування, мабуть, домінують серед кодових конструкцій, що допускають ітеративне декодування, блокові коди добутки теж можна декодувати ітеративно, як це впливає з попереднього обговорення.

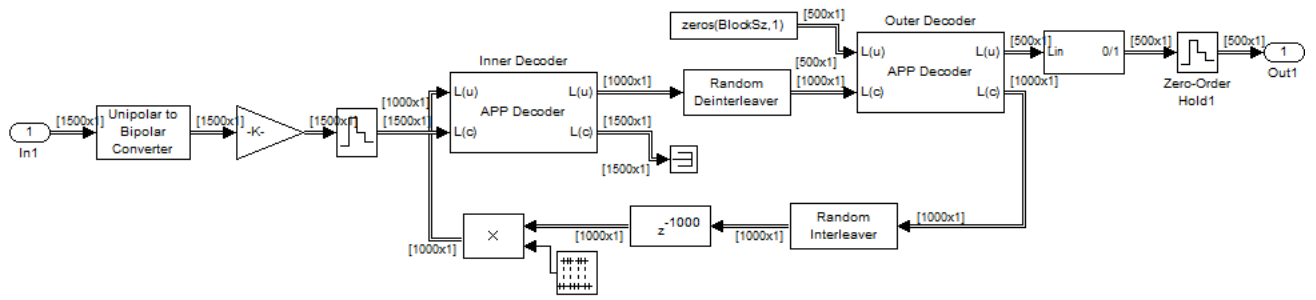


Рисунок 3.4 – Блок-схема декодера послідовного складеного коду.

Розглянемо ітеративне декодування з використанням алгоритму Чейза. Для генерації списку кандидатів (кодових слів, близьких до прийнятого слова) застосовується алгоритм Чейза тип-II. Зовнішні LLR обчислюються на основі двох найближчих кандидатів зі списку. Якщо знайдено лише одне кодове слово (список обсягу 1), то апроксимовані значення LLR надходять на вихід декодера. Нехай  $C$  двійковий лінійний блоковий  $(N, K, d)$  код, здатний виправити будь-яку комбінацію  $t = \lfloor (d-1)/2 \rfloor$  або менше випадкових помилок. Нехай  $r = (r_1, r_2, \dots, r_N)$  прийняте слово на виході каналу,  $r_i = (-1)^{v_i} + w_i$ , де  $v_i \in C$  і  $w_i$  - гауссова випадкова величина з нульовим середнім і дисперсією  $N_0/2$ . Структурна схема алгоритму Чейза тип-2 обробки прийнятого слова  $r$  показана на рисунку 3.5.



Рисунок 3.5 – Структурна схема алгоритму Чейза тип-2

В результаті виконання алгоритму Чейза тип-II можливі наступні події:

1. Знайдено два або більше кодових слів,  $(\hat{v}_1, \hat{v}_2, \dots, \hat{v}_l)$ ,  $l \geq 2$ ;
2. Знайдено одне кодове слово  $\hat{v}_1$ ;
3. Немає жодного кодового слова.

В останньому випадку декодер може підняти прапор про непоправну помилку і видати прийняту послідовність в тому вигляді, в якому вона є. Альтернативним варіантом є збільшення числа комбінацій помилок, що перевіряються до тих пір, поки не знайдеться хоча б одне слово.

Позначимо  $\chi_j(\ell)$  - множину модульованих кодових слів коду  $C$ , у яких  $j$ -а компонента  $x_j = \ell$ ,  $\ell \in \{-1, +1\}$ , для  $1 \leq j \leq N$ . Нехай  $x_j(\ell)$  і  $y_j(\ell)$  з множини  $\chi_j(\ell)$  представляють найближчі (за Евклидовою відстанню) до прийнятого кодового слова  $r$  і наступного модульованого кодового слова, відповідно.

За допомогою  $\log$ - $\max$  апроксимації,  $\log(e^a + e^b) \approx \max(a, b)$ , значення LLR (8.2) для  $j$ -го символу може бути записано в наступному вигляді:

$$\Lambda(u_1) = \Lambda_{c1} + \Lambda_{e1}(u_1) + \Lambda_{e2}(u_1) = +0,41;$$

$$\Lambda(u_2) = \Lambda_{c2} + \Lambda_{e1}(u_2) + \Lambda_{e2}(u_2) = +0,41.$$

Звідси після нормалізації і заміни позначень,  $x_m = x_m(+1)$  і  $y_m = y_m(-1)$ , знаходимо, що м'який вихід дорівнює

$$\Lambda'(u_j) = r_j + \sum_{x_m \neq y_m, m=1}^N r_m x_m$$

$$w_j = x_j \sum_{x_m \neq y_m, m=1}^N r_m x_m$$

(3.12)

інтерпретується як поправочний член м'якого входу  $r_j$  який залежить від двох модульованих кодових слів, найближчих до прийнятого слову  $r$ , і грає ту ж роль, що і зовнішнє LLR. Для кожної позиції  $j$ ,  $1 \leq j \leq N$ , величина  $w_j$  видається наступному декодеру як значення зовнішнього LLR з деяким масштабуючим множником  $\alpha_c$ , а саме, величина

$$r'_j = r_j + \alpha_c w_j$$

(3.13)

обчислюється як м'який вхід для наступного декодера. Множник  $\alpha_c$  вводиться для того, щоб компенсувати різницю дисперсій гаусових випадкових величин  $r_i$  і  $r'_j$ . Структурна схема процедури обчислення м'яких входів і зовнішніх LLR показана на рис. 3.6.

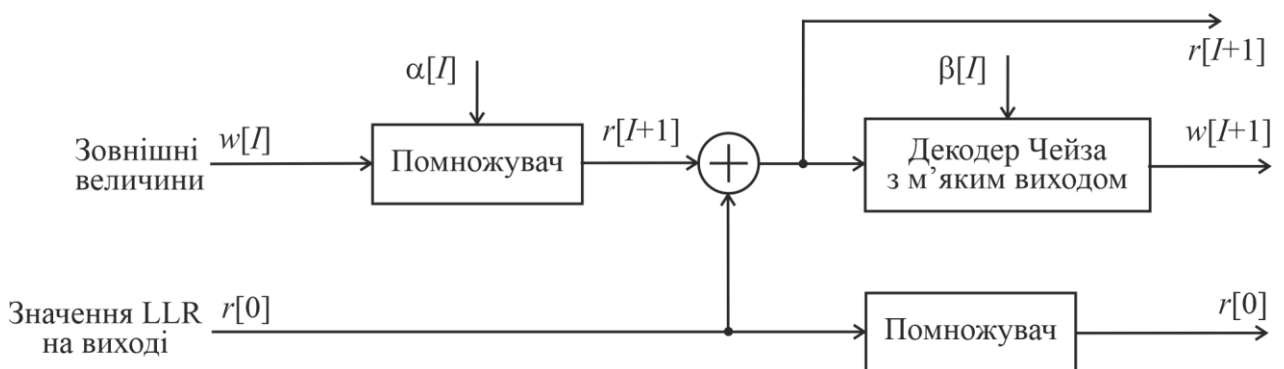


Рисунок 3.6 – Декодер Чейза з м'яким виходом

Якщо для деякої  $j$ -ої позиції,  $1 \leq j \leq N$ , алгоритм Чейза не знаходить жодної пари послідовностей  $x_j(+1)$  і  $y_j(-1)$ , то пропонується використовувати таке значення LLR для символу  $u_j$

$$\Lambda'(u_j) = \beta_c x_j, \quad (3.14)$$

де  $\beta_c$  є коректуючим множником для компенсації помилки апроксимації зовнішньої інформації. За допомогою моделювання цей множник оцінюється як

$$\beta_c = \left| \log \left( \frac{P\{v_j \text{ вірно}\}}{P\{v_j \text{ невірно}\}} \right) \right|, \quad (3.15)$$

тобто в якості значення LLR для експериментальної оцінки ймовірності помилки на символ. У роботах [PP, MT] показано, яким чином можна адаптивно обчислювати коригувальні множники  $\alpha$  і  $\beta$  на основі статистик оброблюваних кодових слів. Алгоритм з м'яким виходом розважений на рис. 3.7.

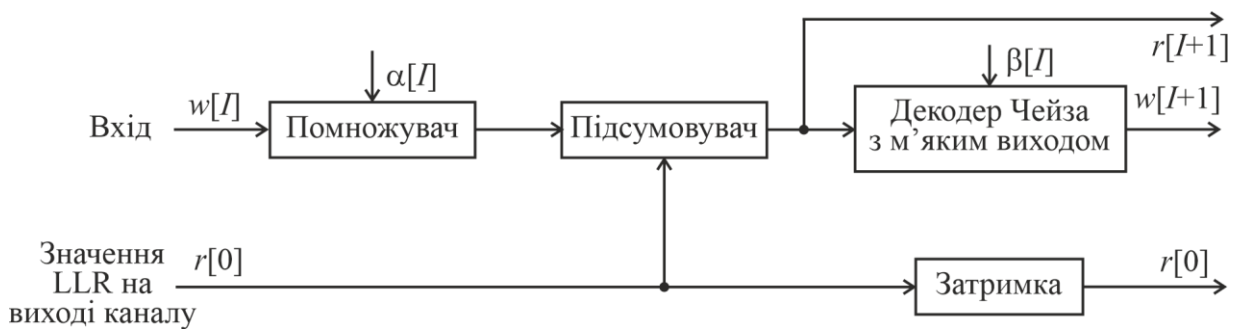


Рисунок 3.7 – Блок схема декодера Чейза з м'яким виходом.

У цьому випадку теж необхідні адаптивні ваги для того, щоб масштабувати значення зовнішніх LLR.

Підсумовуючи сказане вище, отримуємо наступний формалізований опис алгоритму ітеративного декодування з м'яким виходом, що заснований на побудові списку кодових слів за допомогою алгоритму Чейза тип-II.

Крок 0: Початкові установки

Встановити лічильник ітерацій  $I = 0$ . Встановити  $r[0] = r$  (прийнята з каналу послідовність).

Крок 1: М'які входи. Для  $j = 1, 2, \dots, N$ ,

$$r_j[I + 1] = r_j[0] + \alpha_c[I]w_j[I] .$$

Крок 2: Алгоритм Чейза

Виконати алгоритм Чейза тип-II для послідовності  $r[I + 1]$ . Визначити число знайдених слів  $n_c$  (довжину списку). Знайти, якщо це можливо, два найближчих до прийнятої послідовності модульованих кодових слова  $x$  і  $y$ .

Крок 3: Зовнішня інформація

Для  $j = 1, 2, \dots, N$ ,

- Якщо  $n_c \geq 2$ ,

$$w_j[I + 1] = x_j \sum_{m=1}^N r[I + 1]_m x_m$$

- інакше  $w_j[I + 1] = \beta_c[I]x_j$

Крок 4: М'який вихід

Збільшити номер ітерації,  $I = I + 1$ . Якщо  $I < I_{\max}$  (максимальне число ітерацій) або не виповнилася умова зупинки, то повернутися на Крок 1.

Інакше обчислити м'який вихід.

Для  $j = 1, 2, \dots, N$ ,

$$\Lambda(u_j) = \alpha_c[I]w_j[I] + r_j[0]$$

(3.16)

Стоп. У разі двійкової фазової модуляції величини  $\alpha_c$  і  $\beta_c$  були обчислені для чотирьох ітерацій (всього вісім величин для першого і другого декодера).

### Висновки до третього розділу

1. Систематичні згорткові коди, що застосовуються в якості складових кодів турбокодера, характеризуються паралельністю роботи декількох кодерів. При конструктивному поєднанні декількох згорткових кодерів утворюється система, відмінною ознакою якої є потоковий метод кодування і модель системи з багатьма входами-виходами

2. Головним завданням, що вирішується за допомогою алгоритму MAP входить обчислення апостеріорних ймовірностей всіх можливих переходів прийнятого кодового символу по решітці згорткового декодера.

3. У ітеративній процедурі посимвольного MAP декодери компонентних кодів можуть обмінюватися зовнішніми оцінками LLR, істотно підвищуючи надійність оцінки інформаційного символу з збільшенням числа ітерацій.

## 4. ДОСЛІДЖЕННЯ МОДЕЛЕЙ СИСТЕМ ЗВ'ЯЗКУ З ЗГОРТКОВИМИ КОДАМИ

### 4.1. Simulink-модель системи зв'язку з послідовним каскадним кодуванням

На практиці нерідко виникають завдання, коли необхідно використовувати найбільш потужні коди. До числа таких завдань відносяться: забезпечення високої завадостійкості (малу ймовірність помилки в прийманні біта) при дефіциті енергетики радіоліній, що нерідко має місце в супутниковому зв'язку; передача інформації по каналах, в яких помилки з яких-небудь причин групуються в пачки. Застосування в таких випадках більш довгих блокових кодів або згорткових кодів з великою довжиною кодового обмеження неминуче нашоувхується на обмеження, пов'язані з неприпустимим зростанням складності декодера. Вихід з такого суперечливого становища знайшов Форне, який запропонував схему каскадного (багаторівневого) кодування. На практиці найбільш поширеною виявилася каскадна схема з двома рівнями кодування.

При каскадному кодуванні використовуються два коди: зовнішній код, зазвичай блоковий недвійковий, і внутрішній. В якості зовнішніх кодів найбільш часто використовуються код Ріда-Соломона. Ці коди найбільш широко поширені, оскільки є кодами з максимальною кодовою відстанню для фіксованих  $N$  і  $K$  і відносно просто декодуються. Параметри код Ріда-Соломона наступні:

- довжина кодового слова  $N = q - 1 = 2^k - 1$ , де  $k = 1, 2, 3, \dots, n - 1$ ;
- кількість інформаційних символів в кодовому слові  $K = N - 2t$ ;
- мінімальна кодова відстань  $d_{min} = N - K + 1$ ;
- кодова швидкість  $R = K/N$ ;
- кількість помилок в кодовому слові, що виправляється  $t$ .

Серед внутрішніх кодів принципово можуть використовуватися

циклічні коди, ортогональні коди і згорткові. У більшості випадків джерела цифрової інформації є двійковими. До кодування зовнішнім кодом двійкова інформація розбивається на послідовні групи по  $k$  символів кожна. Така група є один  $g$ -ічний символ. Далі послідовності з  $K$   $g$ -ічних символів кодером ставиться у відповідність кодове слово РС-коду довжиною  $N$   $g$ -ічних символів або, що те ж, довжиною  $Nk$  двійкових символів.

Далі починається кодування внутрішнім кодом. При використанні в якості внутрішнього, відносно коротких, наприклад, двійкових циклічних кодів кожної зазначеної вище групи з  $k$  двійкових символів внутрішнім кодером ставиться у відповідність кодове слово внутрішнього коду довжини  $n$  двійкових символів. Довжина отриманого таким чином кодового слова каскадного коду  $N^* = Nn$  двійкових символів, кількість інформаційних біт  $K^* = Kk$ .

При використанні в якості внутрішніх ортогональних (в тому числі біортогональних і симплексних)  $m$ -ічних кодів кожної зазначеної вище групи з  $k$  двійкових символів внутрішнім кодером ставиться у відповідність кодове слово ортогонального  $m$ -ічного коду.

При використанні в якості внутрішніх згорткових кодів потік двійкових символів кодових слів РС-коду кодується згортковим кодом. Довжина отриманого таким чином кодового слова каскадного коду  $N^* = Nn_0$  двійкових символів, кількість інформаційних біт  $K^* = Kk_0$ .

Декодування каскадних кодів здійснюється в зворотному порядку. Спочатку декодується внутрішній код, потім зовнішній. Це дозволяє значно знизити складність декодера каскадного коду, в порівнянні зі складністю декодера, який був би необхідний при одноступеневому декодуванні для забезпечення однакової ймовірності помилки.

Simulink-модель системи зв'язку з послідовним каскадним кодуванням приведена на рисунку 4.1. В якості джерела вхідного сигналу використаний Random Integer generator, що генерує рівномірно розподілені числа в діапазоні  $[0, M-1]$ , де  $M$  задається параметром Set size.

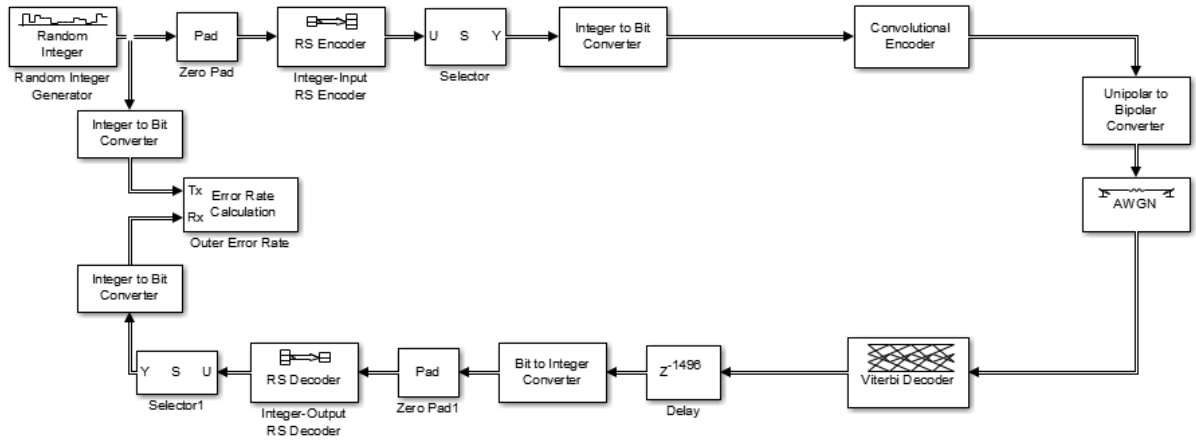


Рисунок 4.1 – Simulink-модель системи зв'язку з послідовним каскадним кодуванням

На рисунку 4.2 зображено вікно налаштувань параметрів блоку Zero Pad.

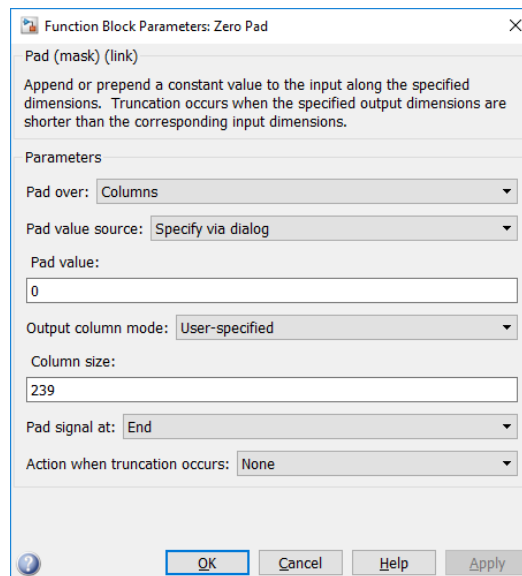


Рисунок 4.2 – Налаштування параметрів блоку Zero Pad

Блок Pad розширює або звужує розміри сигналу на вході шляхом заповнення або видалення даних за його стовпцями або рядками, або будь-

якими зазначеними параметрами. Вилучення даних відбувається, коли ви вказуєте розміри сигналу на виході, які коротше, ніж відповідні розміри сигналу на вході. Ви можете ввести значення блоку Pad в масці блоку або через вхідний порт.

Pad over –розміри сигналу, що необхідно розширити або звузити.

Dimensions to pad –масив, елементи якого треба розширити або звузити.

Pad value source –джерело значень блоку Pad.

Pad value –значення блоку Pad.

Найбільш широко каскадні коди використовуються в конструкції з внутрішніми згортковими кодами.

В супутниковому зв'язку використання таких кодів через їх високу завадостійкість при відносно простій реалізації диктується діючими стандартами. На відміну від конструкцій каскадних кодів з внутрішніми ортогональними конструкції, що розглядаються можна використовувати при середніх і високих швидкостях кодування.

Коди Ріда-Соломона – це недвійкові циклічні коди, символи яких  $m$ -бітові послідовності, де  $m$  - додатне ціле число, більше 1. Коди Ріда-Соломона ефективні при виправленні пакетів помилок в каналах з пам'яттю, а також в каналах з великим набором вхідних символів. Особливістю кодів Ріда-Соломона є те, що до коду довжиною  $n$  можна додати два інформаційних символи, не зменшуючи мінімальної відстані. Такий розширений код має довжину  $n + 2$  і таку ж кількість символів контролю парності, що і початковий код.

До останнього часу каскадні коди з внутрішніми циклічними кодами в системах зв'язку використовувалися не часто. Пояснювалося це тим, що вони забезпечували відносно невисоку стійкість, пов'язану з жорстким декодуванням внутрішнього коду. М'яке декодування циклічних кодів, що забезпечує додатковий енергетичний вигравш кодування близько 2 дБ, практично не використовувалося через складність реалізації декодера.

Каскадні коди з внутрішніми ортогональними кодами застосовуються в системах зв'язку більш широко, особливо в тих випадках, коли канал зв'язку широкосмуговий, тобто його смуга багато ширше смуги, займаної спектром інформаційних сигналів і можна використовувати коди з низькими швидкостями кодування ( $R \ll 1$ ). Ці коди відносно просто декодувати в цілому (тобто м'яко), тому їх застосування в каскадній конструкції досить ефективно. Ця ефективність зберігається і при некогерентному прийманні внутрішнього коду в такий каскадній конструкції. Однак, як уже зазначалося, складність декодера ортогональних кодів зростає пропорційно  $2^k$ , тому реальний інтерес представляють тільки ортогональні коди з  $k \leq 10$ . Simulink-модель системи зв'язку з згортковим кодуванням зображена на рисунку 4.3.

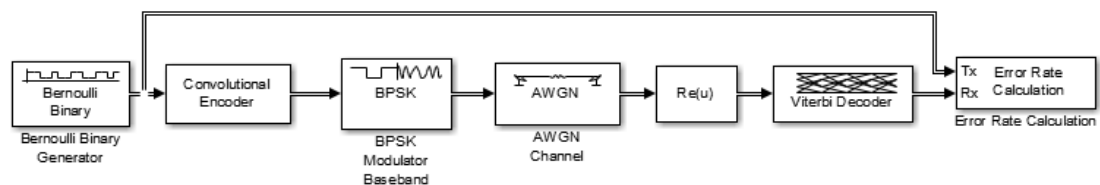


Рисунок 4.3 – Simulink-модель системи зв'язку з згортковим кодуванням

Відношення  $R = k/n$  називають відносною швидкістю коду. Вихідні символи згорткового кодера однозначно визначаються його вхідним сигналом і станом, що залежать від  $m - k$  попередніх символів. На рисунку 4.4 наведені залежності ймовірності помилки на біт від відношення енергії біта до спектральної густини потужності шуму для каскадного коду і згорткового коду.

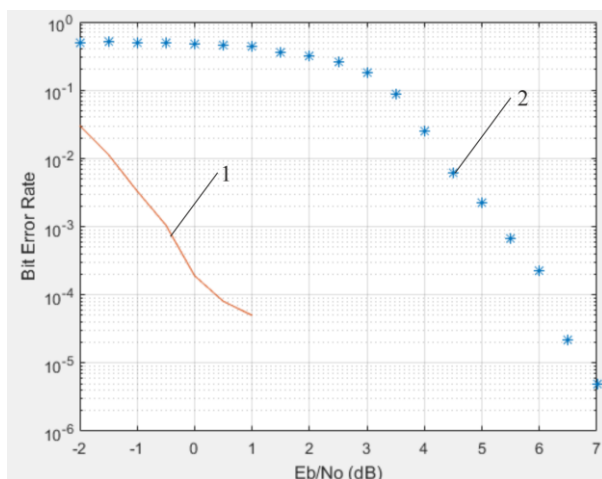


Рисунок 4.4 – Залежності ймовірності помилки на біт (BER) від відношення  $E_b/N_0$ : 1) каскадний код; 2) згортковий код

Основними елементами згорткового кодера є: регістр зсуву, суматори за модулем 2 і комутатор.

Якщо зафіксувати  $BER=10^{-4}$ , то використання каскадного коду у складі зовнішнього згорткового коду (171,133) і внутрішнього коду Ріда-Соломона (7,5) ефективніше використання простого згорткового коду на 5,5 дБ.

Найкращими складовими кодами є рекурсивні систематичні згорткові коди (рис. 4.5), оскільки їх застосування дозволяє значно зменшити число кодових слів низької ваги, що визначають ефективність турбокоду.

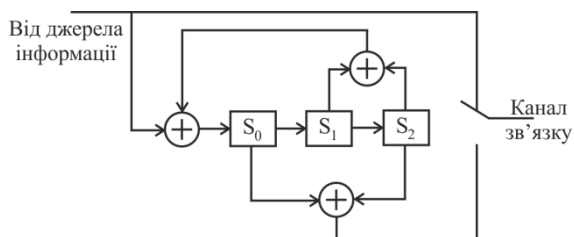


Рисунок 4.5 – Кодер рекурсивного систематичного згорткового коду

## 4.2 Згорткові коди

Згорткові коди засновані на перетворенні вхідної послідовності двійкових символів в вихідну послідовність двійкових символів, у якій на кожний символ вхідної послідовності формується більше одного символу вихідної послідовності. На відміну від блокових кодів, вони не ділять інформацію на фрагменти і працюють з нею як із суцільним потоком даних.

Згорткове кодування - це ітеративна обробка потоку бітів, що створює залежність кожного біта від кількох попередніх. Згорткове кодування задають за допомогою породжуючих поліномів, які визначають структуру двійкового кодера згорткового коду. Кодове слово на виході такого кодера складається у вигляді двох послідовностей, які у двійковій формі представляють коефіцієнти відповідних породжуючих поліномів.

Приклад згорткового кодера наведено на рисунках 4.7 і 4.8. Він містить трирозрядний регістр зсуву, на вхід якого надходить вхідна послідовність двійкових символів. Кодер складається з 3-розрядного регістра зсуву і 2-х суматорів за модулем 2.

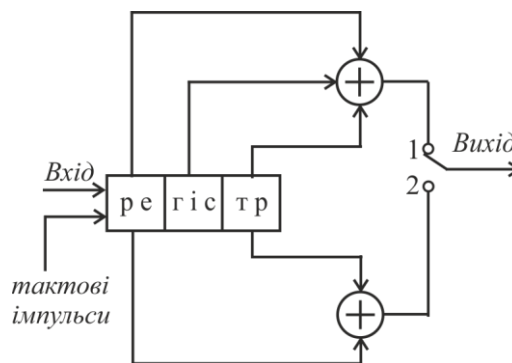


Рисунок 4.7 – Кодер несистематичного згорткового коду швидкістю  $\frac{1}{2}$

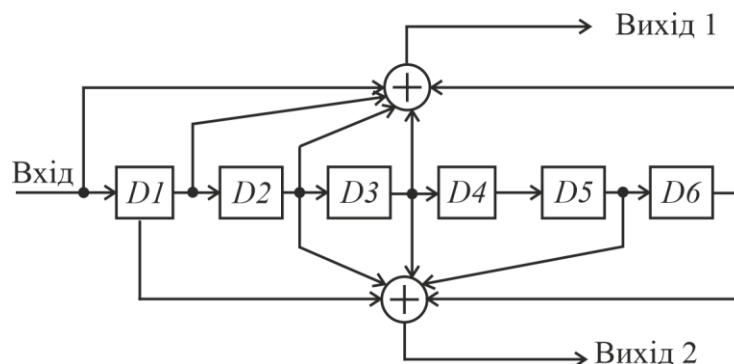


Рисунок 4.8 – Схема згорткового кодера цифрового телебачення DVB-T

Інформаційна послідовність надходить на вхід регістра. Вміст регістра обробляється суматорами, на виходах яких утворюється кодова послідовність, яка надходить на вихід.

Відносна швидкість цього коду зменшується ( $R = 1/2$ ), а тактова частота зростає в два рази.

Розглянемо декодування згорткових кодів. Декодування реалізує алгоритм Вітербі, заснований на принципі максимальної правдоподібності (рисунок 4.9). Алгоритм включає в себе обчислення міри подібності (або відстані Хеммінга), між сигналами на вході декодера і на його виході. Декодер приймає рішення шляхом виключення менш ймовірних шляхів.



Рисунок 4.9 – Декодер згорткових кодів за алгоритмом Вітербі

Згорткові коди ефективно працюють в адитивному каналі з шумом, але погано справляються з пакетами помилок. Якщо декодер помиляється, на його виході виникає пакет помилок, який виправляється кодом РС.

Кодер для згорткового коду представляє собою пристрій з пам'яттю. Символи, що надходять на вхід кодера, за певним законом змінюють його внутрішній стан. Вихідні символи залежать від вхідних символів і від внутрішнього стану кодера. Цей реалізується за допомогою функції convenc.

Декодування згорткового коду проводиться за алгоритмом Вітербі функцією vitdec. Підтримуються м'яке і жорстке декодування.

Розглянемо згортковий кодер зі ступенем кодування  $1/2$ , довжиною кодового обмеження  $K$  і пам'яттю порядку  $K - 1$ . На вхід кодера в момент  $k$ , подається біт  $d_k$ . Відповідним кодовим словом буде бітова пара  $(u_k, v_k)$ , де

$$u_k = \sum_{i=1}^{K-1} g_{li} d_{k-i} \quad \text{по модулю 2, } g_{li} = 0,1 \quad (2.13)$$

$$v_k = \sum_{i=1}^{K-1} g_{2i} d_{k-i}, \text{ по модулю } 2, \quad g_{2i} = 0, 1. \quad (2.14)$$

$G_1 = \{g_{1i}\}$  і  $G_2 = \{g_{2i}\}$  - генератори коду, а  $d_k$  представлений як двійкова цифра. Цей кодер можна представити як лінійну систему з дискретною кінцевою імпульсною характеристикою (finite impulse response - FIR), яка породжує несистематичний згортковий (nonsystematic convolutional - NSC) код, різновид якого показана на рисунку 4.10.

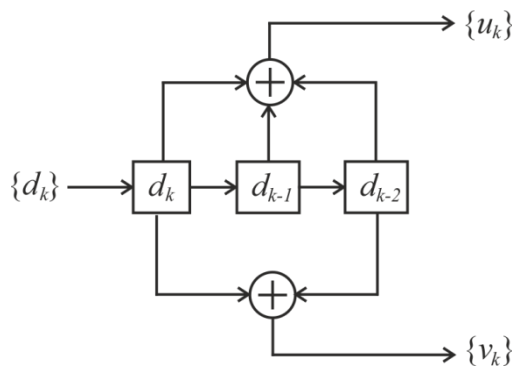


Рисунок 4.10 – Несистематичний згортковий код

Відомо, що при великих значеннях  $E_b/N_0$  достовірність передачі з кодом NSC вище, ніж у систематичного коду з такою ж пам'яттю. При малих значеннях  $E_b/N_0$  існує обхідний шлях. В якості складових компонентів для турбокодів був запропонований клас згорткових кодів з нескінченною імпульсною характеристикою. Такі ж компоненти використовуються в рекурсивних систематичних згорткових (recursive systematic convolutional - RSC) кодах, оскільки в них попередньо кодовані біти даних постійно повинні подаватися на вхід кодера. Двійковий код RSC зі ступенем кодування  $1/2$  виходить з коду NSC шляхом додавання контуру зворотного зв'язку і установки одного з двох виходів ( $u_k$  або  $v_k$ ) рівним  $d_k$ . На рисунку 4.11 показаний приклад такого RSC-коду з  $K = 3$ , де  $a_k$  визначається виразом:

$$a_k = d_k + \sum_{i=0}^{K-1} g'_i a_{k-i} \text{ по модулю } 2, \quad (2.15)$$

де  $g'_i = g_{1i}$ , якщо  $u_k = d_k$ , і  $g'_i = g_{2i}$  - якщо  $v_k = d_k$ .

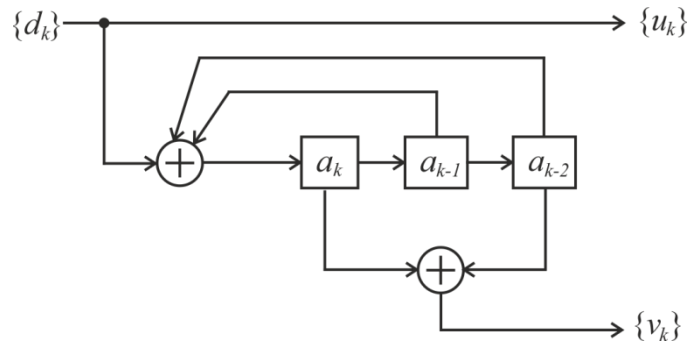


Рисунок 4.11 – Рекурсивний систематичний згортковий код (recursive systematic convolutional — RSC)

На рисунку 4.12 зображена ґратчаста структура RSC-коду, представленого на рисунку 4.11.

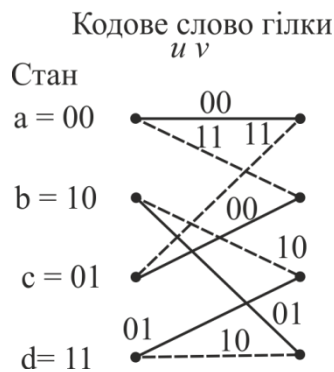


Рисунок 4.12 – Ґратчаста структура RSC- кода, представленого на рисунку 4.11

Вважається, що вхідний біт  $d_k$  з однаковою ймовірністю може приймати як значення 1, так і 0. Крім того,  $\{a_k\}$  показує ті ж статистичні ймовірності, що і  $\{d_k\}$ . Просвіт однаковий у RSC-коду (рисунок 2.5) і NSC-коду. Точно так же збігається їх ґратчаста структура по відношенню до переходів між станами і відповідним вхідним бітам. У RSC- і NSC-кодів дві вихідні послідовності  $\{u_k\}$  і  $\{v_k\}$  не відповідають тій же вхідній послідовності  $\{d_k\}$ . Можна сказати, що при тих же генераторах коду розподіл вагових коефіцієнтів кодових слів RSC-кодера не змінюється, в порівнянні з розподілом вагових коефіцієнтів кодових слів NSC-кодера. Єдина відмінність полягає у відображенні між вхідною і вихідною послідовностями даних.

### 4.3 Коди Ріда-Соломона

Коди Ріда-Соломона РС - недвійкові блокові циклічні коди, що дозволяють виправляти помилки в блоках даних. Недвійкові вони тому, що елементами кодового слова є не біти, а групи бітів (байти). Для восьмибітових символів повна довжина кодового слова становить  $2^8-1 = 255$  символів, з них 239 інформаційних і 16 перевірочних.

У стандарті цифрового телебачення DVB використовується код РС, що записується як (204, 188, 8). Тут 188 - кількість інформаційних байтів в пакеті транспортного потоку MPEG-2, 204 - кількість байт в пакеті після додавання перевірочних символів, 8 - число коректованих байтів. Таким чином, в якості кодових комбінацій беруться цілі пакети транспортного потоку, що містять 188 байт, а перевірочні символи, що додаються, містять 16 байт.

Розглянемо побудову кодового слова, захищеного кодами РС. Формування циклічних кодових слів проводиться за допомогою породжуючого полінома  $g(x)$ , максимальна степінь якого дорівнює числу перевірочних символів  $n-k$ .

Процедура знаходження кодового слова  $c(x)$  полягає в множенні початкового інформаційного полінома  $i(x)$  на породжуючий поліном коду:  $c(x) = g(x) \cdot i(x)$ .

У приймачі для кожного прийнятого транспортного пакета, що містить 204 байта, обчислюються поліноми синдромів помилки. Синдроми помилок виходять діленням кодового слова на породжуючі поліноми. Якщо кодові слова діляться на породжуючий поліном без залишку, то помилок немає. Прийняті слова, які не діляться без залишку на породжуючий поліном, не є дозволеними і, отже, містять помилки. При наявності залишку від ділення - включається процедура виправлення помилки.

Опис кодера Ріда-Соломона. Кодер (рис. 4.13) здійснює розподіл полінома  $v(x)$ , що відображає прийняте слово, на породжуючий поліном  $g(x)$ . Отриманий залишок від ділення приписується до  $v(x)$  в якості молодших розрядів і це кодове слово йде на передачу. Ділення в кодері реалізується за допомогою множення і складання (у двійковій системі числення).

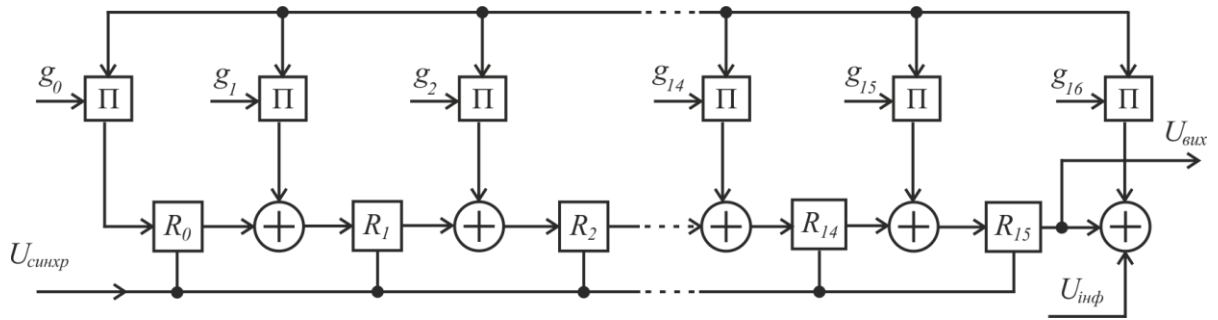


Рисунок 4.13 – Кодер Ріда-Соломона ( $g_0 - g_{16}$  – коефіцієнти породжуючого полінома

$g(x)$ ;  $R_0 - R_{16}$  – комірки пам'яті регістру зсуву;

$+$  - суматор за модулем 2;  $\Pi$  – помножувач сигналів)

Розглянемо декодування кодів РС (рис. 4.14). У приймачі для кожного прийнятого транспортного пакету, що містить 204 байта, обчислюються синдроми помилок і два поліноми: «локатор»  $L(x)$ , корені якого показують положення помилок, і «коректор», що дає значення помилок  $Y_i$ . Обчислення синдрому помилки  $S(x)$ . Синдром помилки є залишком від ділення кодового слова  $c(x)$ , що декодується, на породжуючий поліном  $g(x)$ . Якщо цей залишок дорівнює нулю, кодове слово буде неспотвореним.

Знаходження позицій символічних помилок відбувається шляхом відновлення даних за допомогою розв'язку системи рівнянь з  $t$  невідомими. Для розв'язку використовується швидкий алгоритм Форні.

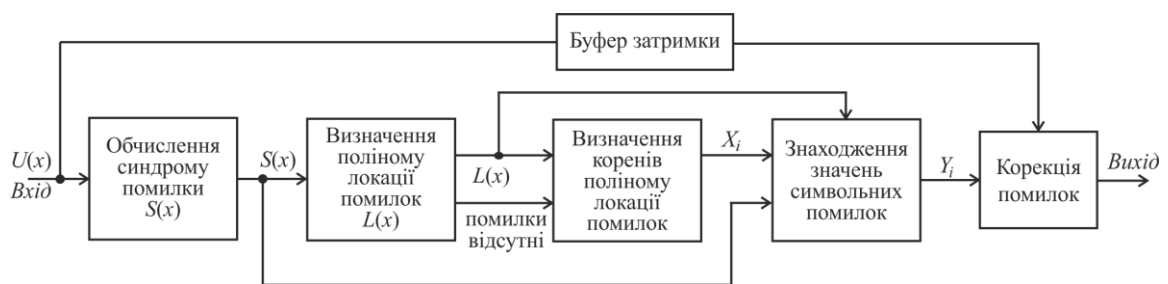


Рисунок 4.14 – Декодер кодів Ріда-Соломона ( $U(x)$  - кодове слово з приймача;  $S(x)$  -

синдроми;  $L(x)$  - поліном локації помилок;

$X_i$  - місце знаходження помилок;  $Y_i$  - значення помилок)

Коди Ріда-Соломона широко використовується в системах відновлення даних з компакт-дисків, при створенні архівів з інформацією для відновлення в разі пошкоджень, в завадостійкому кодуванні.

Відразу після появи (60-і роки ХХ століття) коди Ріда - Соломона стали застосовуватися в якості зовнішніх кодів в каскадних конструкціях, що використовуються в супутниковому зв'язку. У подібних конструкціях  $q$ -і символи РС (їх може бути декілька) кодуються внутрішніми згортковими кодами. На приймальному кінці ці символи декодуються м'яким алгоритмом Вітербі (ефективний в каналах з АБГШ). Такий декодер буде виправляти одиночні помилки в  $q$ -их символах, коли ж виникнуть пакетні помилки і деякі пакети  $q$ -их символів будуть декодовані неправильно, тоді зовнішній декодер Ріда - Соломона виправить пакети цих помилок.

На даний момент коди Ріда - Соломона мають дуже широку сферу застосування завдяки їх здатності знаходити і виправляти багаторазові пакети помилок. Код Ріда - Соломона використовується при запису і читанні в контролерах оперативної пам'яті, при архівації даних, запису інформації на жорсткі диски (ECC), запису на CD/DVD диски. Навіть якщо пошкоджений значний обсяг інформації, зіпсовано кілька секторів дискового носія, то коди Ріда - Соломона дозволяють відновити велику частину втраченої інформації.

Цей алгоритм кодування використовується при передачі даних по мережах WiMAX, в оптичних лініях зв'язку, в супутниковому і радіорелейному зв'язку. Метод прямої корекції помилок в поточному трафіку (Forward Error Correction, FEC) ґрунтується на кодах Ріда - Соломона.

#### **4.4 Порівняння коду Ріда-Соломона і згорткового коду**

Розглянемо код Ріда-Соломона з довжиною кодового слова  $N = 7$  і довжиною інформаційного повідомлення  $K = 5$ .

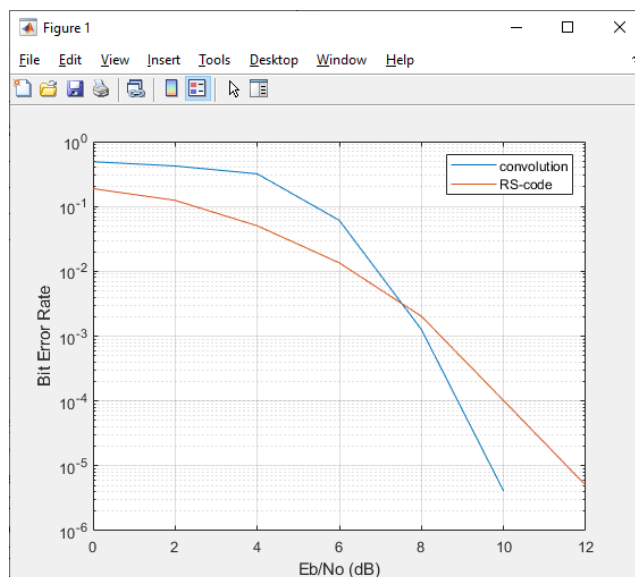


Рисунок 4.15 – Порівняння загорткового коду і кода Ріда-Соломона

На основі отриманої залежності можна зробити висновок, що згортковий код, побудований на основі  $\text{trellis} = \text{poly2trellis}(7, [171 \ 133])$  ефективніше коду Ріда-Соломона (7, 5) на 2 дБ.

### Висновки до четвертого розділу

1. За енергетичним критерієм згортковий код (7, [171 133]) ефективніше коду Ріда-Соломона (7, 5) на 2 дБ.
2. Якщо зафіксувати  $\text{BER}=10^{-4}$ , то використання каскадного коду у складі зовнішнього згорткового коду (171,133) і внутрішнього коду Ріда-Соломона (7,5) ефективніше використання простого згорткового коду на 5,5 дБ.
3. Символи на виході згорткового кодера залежать не тільки від інформаційних символів на вході, а й від попередніх символів на його вході.
4. Згорткове кодування задають за допомогою породжуючих поліномів, які визначають структуру двійкового кодера згорткового коду.

## ВИСНОВКИ

1. За енергетичним критерієм згортковий код (7, [171 133]) ефективніше коду Ріда-Соломона (7, 5) на 2 дБ..
2. Досліджено каскадний код з двома рівнями кодування. Зовнішній код є кодом Ріда – Соломона (7,5); а внутрішній – згортковий код (171,133). Якщо зафіксувати  $BER=10^{-4}$ , то каскадний код ефективніше згорткового на 5 дБ.
3. Систематичні згорткові коди, що застосовуються в якості складових кодів турбокодера, характеризуються паралельністю роботи декількох кодерів. При конструктивному поєднанні декількох згорткових кодерів утворюється система, відмінною ознакою якої є потоковий метод кодування і модель системи з багатьма входами-виходами
4. Головним завданням, що вирішується за допомогою алгоритму MAP входить обчислення апостеріорних ймовірностей всіх можливих переходів прийнятого кодового символу по решітці згорткового декодера.
5. У ітеративній процедурі посимвольного MAP декодери компонентних кодів можуть обмінюватися зовнішніми оцінками LLR, істотно підвищуючи надійність оцінки інформаційного символу з збільшенням числа ітерацій.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Шинкарук О.М. Основи функціонування багатоканальних систем передачі інформації: навч. Посібник /О.М. Шинкарук, Ю.М. Бойко, І.І. Чесановський. – Хмельницький: ХНУ, 2011. – 245 с.

2. Бойко Ю.М. Дослідження способів завадостійкого кодування для захисту від помилок у цифрових каналах передавання інформації /Ю.М. Бойко, Д.А. Макаришкін, С.В. Івчун // Вісник ХНУ. – 2013. - №2. С.188-195.

1. Кларк Д. Кодирование с исправлением ошибок в системах цифровой связи / Д. Кларк, Д. Кейн. Пер. с англ. – М.: Радио и связь, 1987. – 392 с.

2. Скляр Б. Цифровая связь. Теоретические основы и практическое применение/ Б. Скляр. Изд. 2-е, испр.: Пер. с англ. – М. : Издательский дом «Вильямс», 2003.– 1104 с.

3. Прокис Д. Цифровая связь / Д. Прокис. Пер. с англ. под ред. Д.Д. Кловского. – М.: Радио и связь, 2000. – 800 с.

4. Банкет В. Л. Завадостійке кодування в телекомунікаційних системах: навч. посіб. з вивчення модуля 4 дисципліни ТЕЗ / В.Л. Банкет, П.В. Іващенко, М.О. Іщенко. – Одеса: ОНАЗ ім. О. С. Попова, 2011. – 100 с.

5. Пятін І. С. Дослідження послідовного каскадного турбокодування / І. С. Пятін, Д.А. Макаришкін // Вісник ХНУ. Технічні науки № 3, 2017 - С. 142-145.

6. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение. / Р. Морелос-Сарагоса, // М.: Техносфера, 2005. – 320 с.

7. Бойко Ю.М. Текстові документи. Загальні вимоги СОУ 207.01:2017 / Ю.М. Бойко, Г.В. Красильникова, Л.І. Першина, Т.Ф. Косянчук. – Хмельницький : ХНУ, 2017. – 45 с.

8. Стецюк В.І. Дипломне, курсове проектування і магістерська атестаційна робота. Методичні вказівки до розробки та оформлення текстових документів для студентів спеціальності «Радіотехніка, радіоелектронні

- апарати і зв'язок» / В.І. Стецюк, С.К. Підченко, А.А. Таранчук. – Хмельницький: ХНУ, 2007. – 110 с.
9. Шинкарук О.М. Приймання та оброблення сигналів: навч. посібник / О.М. Шинкарук, В.І. Правда, Ю.М. Бойко. – Хмельницький : ХНУ, 2013. – 365 с.
10. Резимов. Л.Т. Естественные радиопомехи / Л.Т, Резимов. – М. : Наука, 1985. – 200 с.
11. Шинкарук О.М. Основи функціонування багатоканальних систем передачі інформації : навч. посібник / О.М. Шинкарук, Ю.М. Бойко, І.І. Чесановський. – Х. : ХНУ, 2011. – 231 с.
12. Рихтер С.Г Цифровое вещание / С.Г. Рихтер. – М. : Горячая линия – Телеком, 2007. – 352 с.
13. Енциклопедичний навчальний довідник : навч. посібник / за ред. Ю.Л. Мазора, Є.А. Мачуського, В. І. Правди. – К. : Вища школа, 1999. – 838 с.

Додаток А  
(довідковий)  
Презентація

Слайд1

**Модель цифрової  
системи зв'язку з  
завадостійким згортковим  
кодуванням**

Біндер Т.С., гр. ТРМ-19-1

Керівник – к.т.н., доц. Пятін І.С.

# Слайд 2

**Мета і задачі дослідження.** Метою роботи є дослідження цифрових систем зв'язку з згортковими кодами в середовищі Matlab .

Для досягнення цієї мети поставлені наступні завдання.

- провести огляд завадостійких кодів у цифрових системах зв'язку;
- розглянути і проаналізувати структурну схему каналу зв'язку з згортковими кодами;
- дослідити модель системи зв'язку з згортковими кодами, оцінити енергетичну ефективність .

**Об'єктом дослідження** є процес передачі і приймання, сигналів з згортковими кодами у цифрових системах зв'язку .

**Предметом дослідження** є методи та засоби згорткового кодування і декодування у цифрових системах зв'язку .

**Наукова новизна одержаних результатів :**

1. Вперше проведена оцінка енергетичної ефективності цифрових систем зв'язку з згортковими кодами. Згортковий код з швидкістю  $R=0,5$  ефективніше коду Ріда-Соломона(7, 5) на 2 дБ.
2. Отримали подальший розвиток методи завадостійкого кодування і декодування.

# Слайд 4

## Цифрова система зв'язку з каскадним кодуванням

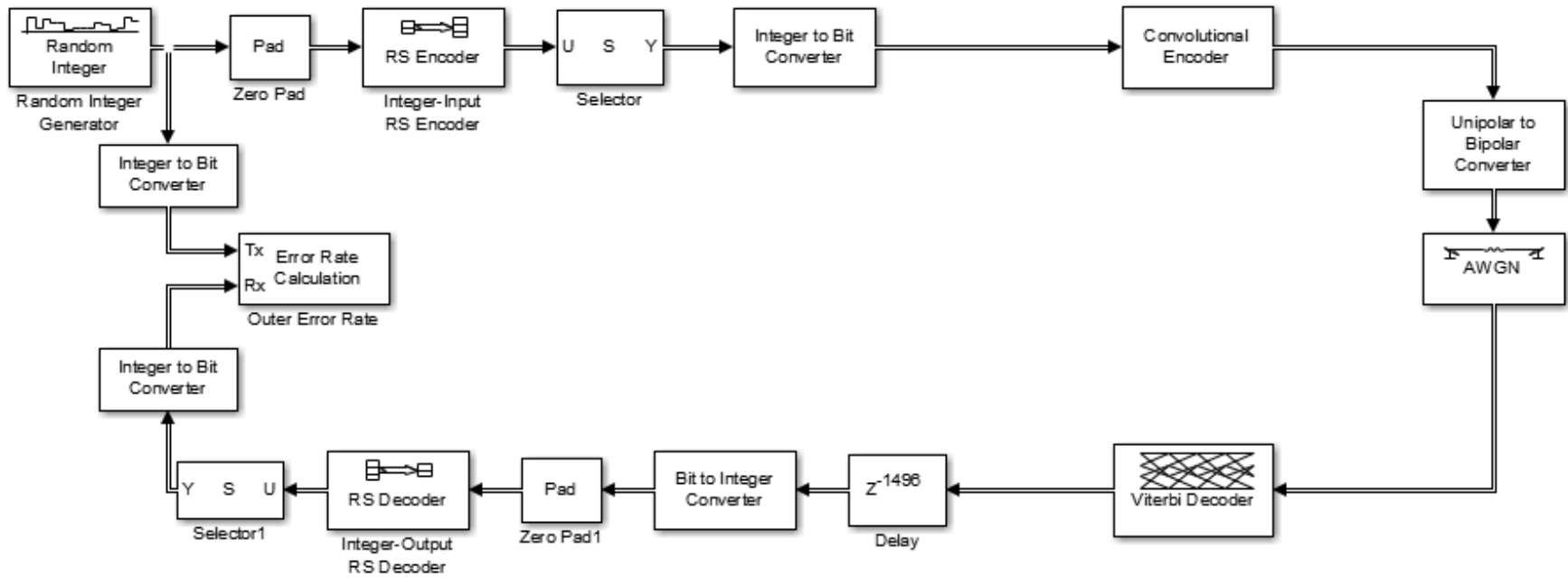


Рисунок 4.1- Simulink-модель системи зв'язку з послідовним каскадним кодуванням

# Слайд 5

## Згортковий код

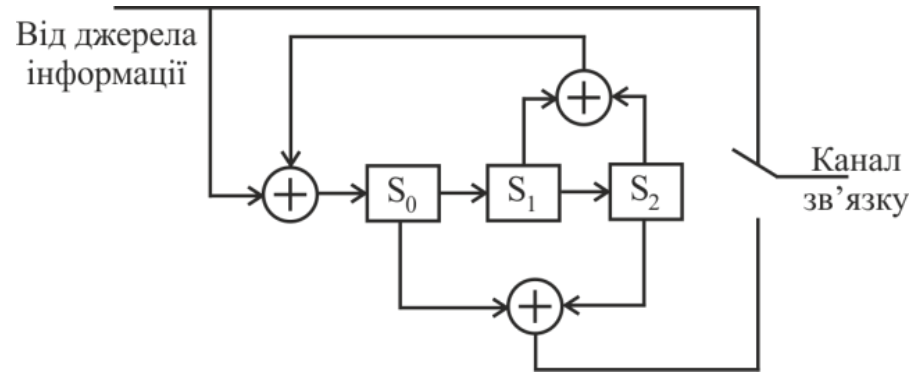
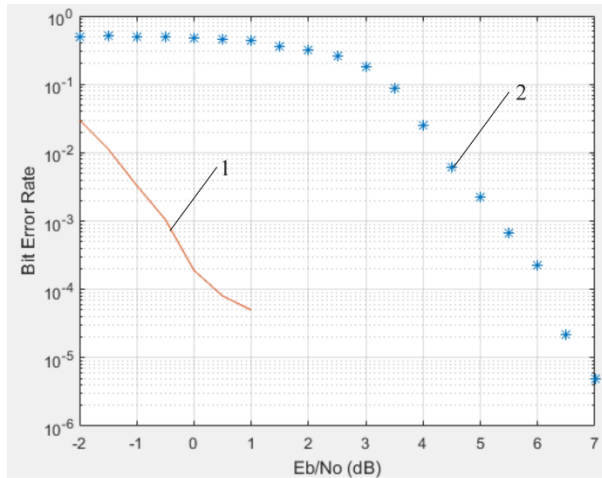


Рисунок 5.2 – Кодер рекурсивного систематичного згорткового коду



Рисунок 5.3 – Декодер згорткових кодів за алгоритмом Вітербі

Рисунок 5.1- Залежності ймовірності помилки на біт (BER) від відношення  $E_b/N_0$ : 1) каскадний код; 2) згортковий код

# Слайд 6

## Рекурсивний систематичний згортковий код

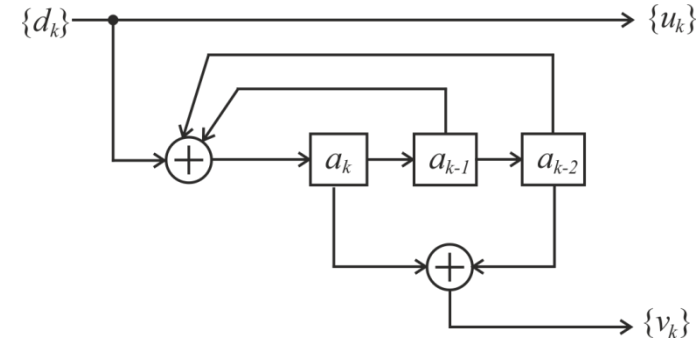


Рисунок 6.1 – Рекурсивний систематичний згортковий код (recursive systematic convolutional — RSC)

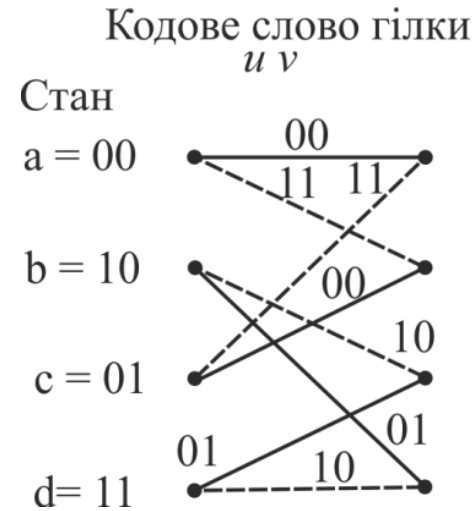


Рисунок 6.2 – Градчаста структура RSC- кода, представленого на рисунку 4.11

# Слайд 7

## Порівняння коду Ріда-Соломона і згорткового коду

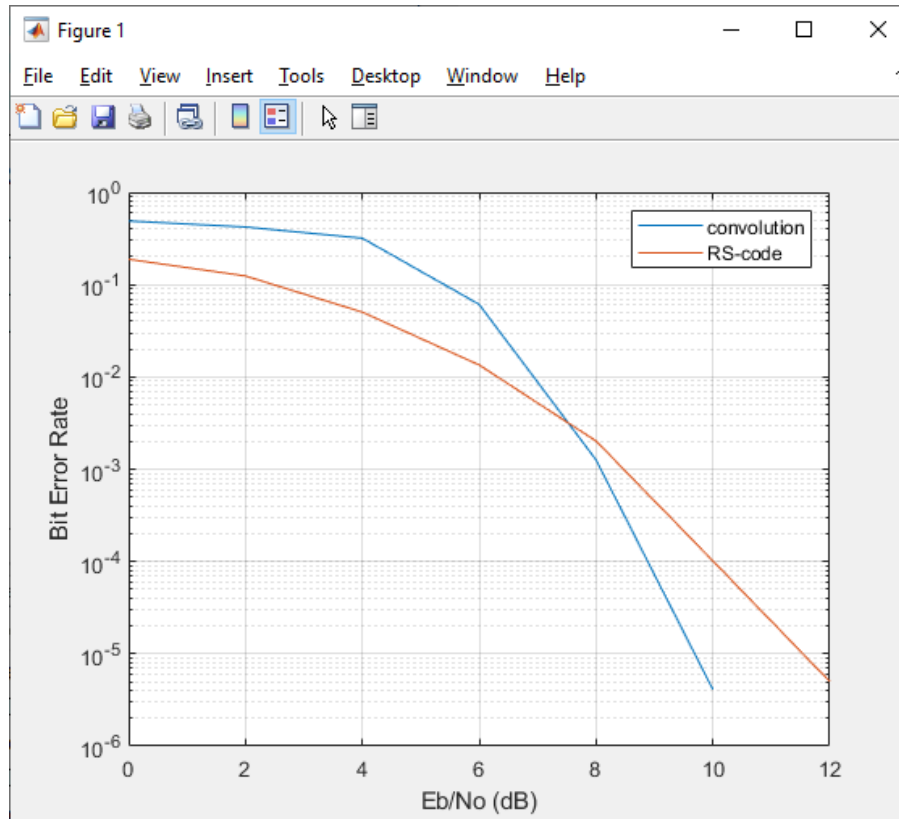


Рисунок 7.1 – Порівняння згорткового коду (171, 133) і кода Ріда-Соломона (7, 5)

# Слайд 8

## Код Ріда-Соломона

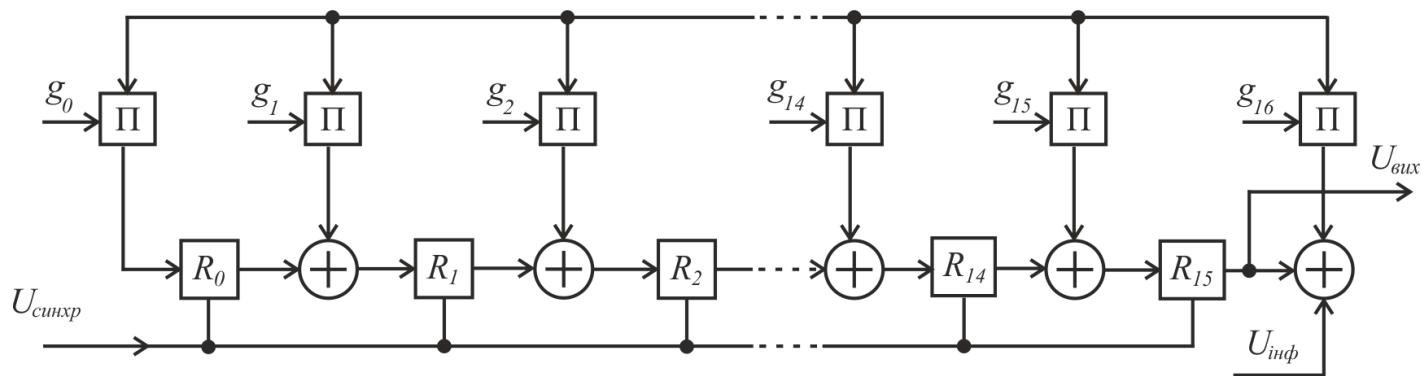


Рисунок 8.1 – Кодер Ріда-Соломона

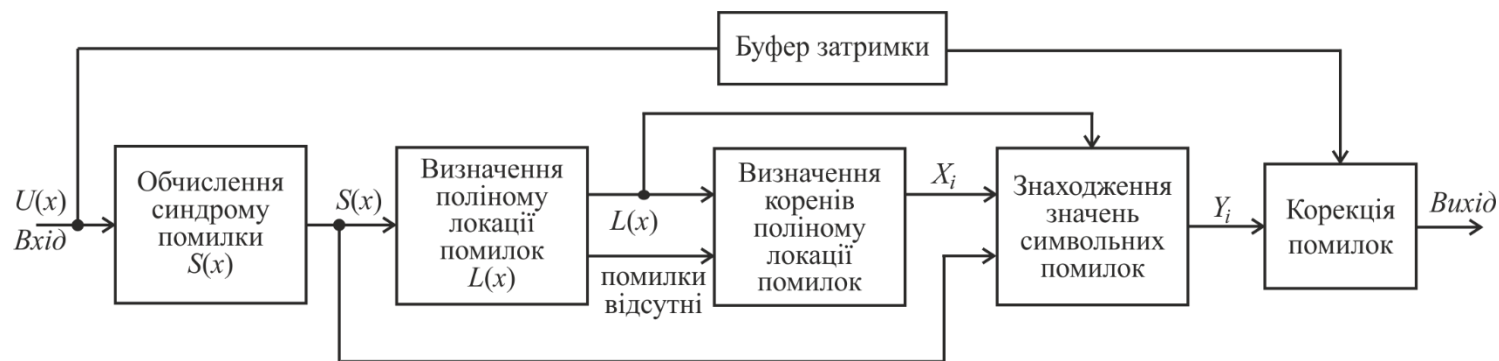


Рисунок 8.2 - Декодер кодів Ріда-Соломона

## ВИСНОВКИ

1. За енергетичним критерієм згортковий код (7, [171 133]) ефективніше коду Ріда-Соломона (7, 5) на 2 дБ..
2. Досліджено каскадний код з двома рівнями кодування. Зовнішній код є кодом Ріда – Соломона (7,5); а внутрішній – згортковий код (171,133). Якщо зафіксувати  $BER=1E-4$ , то каскадний код ефективніше згорткового на 5 дБ.
3. Систематичні згорткові коди, що застосовуються в якості складових кодів турбокодера, характеризуються паралельністю роботи декількох кодерів. При конструктивному поєднанні декількох згорткових кодерів утворюється система, відмінною ознакою якої є потоковий метод кодування і модель системи з багатьма входами-виходами
4. Головним завданням, що вирішується за допомогою алгоритму MAP входить обчислення апостеріорних ймовірностей всіх можливих переходів прийнятого кодового символу по решітці згорткового декодера.
5. У ітеративній процедурі посимвольного MAP декодери компонентних кодів можуть обмінюватися зовнішніми оцінками LLR, істотно підвищуючи надійність оцінки інформаційного символу з збільшенням числа ітерацій.

Додаток Б  
(довідковий)  
Апробація роботи



# ІПК МНІС ІП-2020

**ЗБІРНИК НАУКОВИХ ПРАЦЬ  
МОЛОДИХ  
НАУКОВЦІВ І СТУДЕНТІВ**

**ЧАСТИНА 1**



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Хмельницький національний університет

Військовий інститут Київського національного університету  
ім. Тараса Шевченка

ПВНЗ "Університет економіки і підприємництва"

Вінницький національний технічний університет

Західноукраїнський національний університет

## **Інтелектуальний потенціал - 2020**

збірник наукових праць молодих науковців і студентів

сформовано за матеріалами

Всеукраїнської науково-практичної конференції  
молодих науковців і студентів  
«Інтелектуальний потенціал – 2020»

9-10 листопада 2020 р.

Частина 1

Хмельницький  
2020

ББК 74.480.278  
С.88

«Інтелектуальний потенціал – 2020» - збірник наукових праць молодих науковців і студентів / Колектив авторів – Хмельницький: ПВНЗ УЕП, 2020. – Частина 1. – 100 с.

*Відповідальний редактор: Желавська Н.В.*

*Відповідальний за випуск: Чециун В.М.*

*Редакційна колегія:  
Желавський О.Б.  
Кльоц Ю.П.  
Чециун В.М.  
Тимофеева Л.В.*

© Університет економіки і підприємництва

## ЗМІСТ

Білаш О. Ю., Пятін І.С. Модель визначення спектральної густини потужності сигналу на антені.....	5
Біндер Т. С., Пятін І.С. Модель цифрової системи зв'язку з завадостійким згортковим кодуванням.....	8
Гадомський А.В., Таранчук А.А. Метод моніторингу мережі WLAN WI-FI .....	11
Горбань В.В. Таранчук А.А. Високошвидкісна локальна корпоративна мережа з послугою VoIP – телефонії.....	14
Данілова Л.В., Лавров Є.А., Токар А.С. Оптимізація діалогової людинно-машинної взаємодії в комп'ютерних системах.....	18
Єрмаков М. С., Борисенко О.А. Завадостійкий біноміальний таймер.....	21
Казімірко А.О., Таранчук А.А. Аналіз механізмів захисту мережевого устаткування від хакерської атаки типу TCP SYN Flood .....	23
Ковальчук О.Л., Кучерявий Є.І., Таранчук А.А. Модель «розумної» мережі енергопостачання житлового будинку .....	26
Красильников С.Р. Зміст курсу «Комп'ютерний практикум» у професійній підготовці фахівців спеціальності 015.20 «Професійна освіта. Транспорт» .....	30
Крикун Є. О., Підченко С.К. Технологія побудови сенсорної мережі IoT з використанням протоколу LoRaWAN.....	32
Кубатий Н. О., Таранчук А.А. Пропускна здатність мережі голосової IP-телефонії.....	35
Локазюк В.Ю., Медзатий Д.М. Розробка системи відкритого світу в Unreal Engine 4.....	39
Маниленко М.П., Полікаровських О.І. Обчислювальний метод формування вихідного сигналу синтезатора високих частот.....	42
Матюк Д.С., Мишко О.Є., Деркач М.В. Вплив температури повітря на точність локалізації мобільного робота .....	46
Мельник О. Д., Журавська І. М. Використання технології розпізнавання образів для автоматизації обліку показників побутових лічильників енергії.....	49
Михальський В.М., Полікаровських О.І. Метод нейромережевого керування системою адаптивного радіозв'язку Software Defined Radio.....	53
Ніколайчук І.А., Пятін І.С. Моделювання транспортного каналу з полярними кодами для мобільного зв'язку п'ятого покоління.....	57

электрических сигналов и цепей / Д.Б. Головки, В.В. Белозуб, В.А. Дубровный, Ю.А. Скрипник, В.И. Скрипник. Киев, КНУТД, 2002 – 120с.

2. Афонский А.А. Цифровые анализаторы спектра, сигналов и логики / А.А. Афонский, В.П. Дьяконов. - М.: Солон Пресс, 2009. - 248 с.

3. Афонский А.А. Измерительные приборы и массовые электронные измерения / А.А. Афонский, В.П. Дьяконов. - М.: Солон Пресс, 2007. - 544 с.

### Модель цифровой системы зв'язку з завадостійким згортковим кодуванням

Біндер Т. С.

Науковий керівник – к.т.н., доц. Пятін І.С.  
Хмельницький національний університет

Завадостійким називається кодування, при якому здійснюється виявлення або виявлення та виправлення помилок в прийнятих кодових комбінаціях. Можливість завадостійкого кодування здійснюється теоремою, Шенона, згідно з якою: якщо продуктивність джерела менше пропускної здатності каналу зв'язку, то існує принаймні одна процедура кодування і декодування при якій ймовірність помилкового декодування як завжди мала, якщо ж продуктивність джерела більше пропускної здатності каналу, то такої процедури не існує [1].

Завадостійке кодування відбувається додаванням під час передачі в корисні дані спеціальним чином структурованої надлишкової інформації (наприклад, контрольного числа), а при прийомі використання цієї надмірної інформації для виявлення і виправлення помилки. Число помилок, яке можна виправити, обмежена і залежить від конкретного застосовуваного коду.

Коди виявлення помилок (які можуть тільки встановити факт помилки) належать до тих же класів кодів, що і коди, що виправляють помилки. Фактично будь-який код, що виправляє помилки, може бути також використаний для виявлення помилок (при цьому він буде здатний виявити більше число помилок, ніж був здатний виправити). Коди, що виправляють помилки, застосовуються в системах цифрового зв'язку, в тому числі: супутникової, радіорелейної, стільникової, передачі даних по телефонних каналах, а також в системах зберігання інформації, в тому числі магнітних і оптичних. Коди, що виявляють помилки, застосовуються в мережних протоколах різних рівнів.

За способом роботи з даними коди, що виправляють помилки, діляться на блокові, що ділять інформацію на фрагменти постійної довжини і обробляють кожен з них окремо, і згорткові, що працюють з даними як з безперервним потоком.

Згорткові коди засновані на перетворенні вхідної послідовності двійкових символів в вихідну послідовність двійкових символів, у якій на кожний символ вхідної послідовності формується більше одного символу вихідної послідовності [2]. На відміну від блокових кодів, вони не ділять інформацію на фрагменти і працюють з нею як із суцільним потоком даних.

Згорткове кодування - це ітеративна обробка потоку бітів, що створює залежність кожного біта від кількох попередніх. Згорткове кодування задають за допомогою породжуючих поліномів, які визначають структуру двійкового кодера згорткового коду. Кодове слово на виході такого кодера складається у вигляді двох послідовностей, які у двійковій формі представляють коефіцієнти відповідних породжуючих поліномів.

Приклад згорткового кодера наведено на рис. 1. Він містить трьохрозрядний зсувний регістр, на вхід якого надходить вхідна послідовність двійкових символів. Кодер складається з 3-розрядного регістра зсуву і 2-х суматорів за модулем 2. Схема згорткового кодера для цифрового телебачення DVB-T приведена на рис. 2.

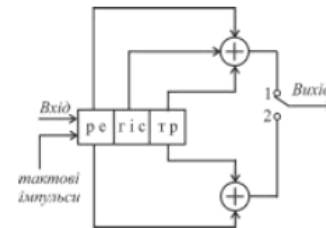


Рисунок 1 – Кодер згорткового коду швидкості 1/2

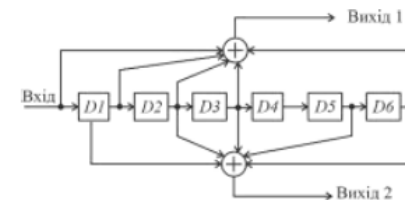


Рисунок 2 – Схема згорткового кодера цифрового телебачення DVB-T

Інформаційна послідовність надходить на вхід регістра. Вміст регістра обробляється суматорами, на виходах яких утворюється кодова послідовність, яка надходить на вихід. Відносна швидкість цього коду зменшується ( $R = 1/2$ ), а тактова частота зростає в два рази.

Розглянемо декодування згорткових кодів. Декодування реалізує алгоритм Вітербі, заснований на принципі максимальної правдоподібності (рис. 3). Алгоритм включає в себе обчислення міри подібності (або відстані Хеммінга), між сигналами на вході декодера і на його виході. Декодер приймає рішення шляхом виключення менш ймовірних шляхів.

Згорткові коди ефективно працюють в каналі з білим шумом, але погано справляються з пакетами помилок. Більш того, якщо декодер помиляється, на його виході виникає пакет помилок, який виправляється кодом РС. Згорткові коди використовуються при низькому відношенні сигнал-шум, коли виправляючої здатності блокових кодів стає недостатньо.



Рисунок 3 – Декодер згорткових кодів за алгоритмом Вітербі

Алгоритм оптимізований за рахунок використання особливостей структури конкретної решітки коду. Перевага декодування Вітербі в порівнянні з декодуванням за методом повного перебору полягає в тому, що складність декодера Вітербі не є функцією кількості символів в послідовності кодових слів.

Алгоритм включає в себе обчислення відстані між сигналом, отриманим в момент часу  $t_i$ , і всіма шляхами решітки, що входять в кожний стан в момент часу  $t_i$ . В алгоритмі Вітербі не розглядаються ті шляхи решітки, які, згідно з принципом максимальної правдоподібності, не можуть бути оптимальними. Якщо в один і той же стан входять два шляхи, вибирається той, який має кращу метрику; такий шлях називається виживаючим. Відбір виживаючих шляхів виконується для кожного стану. Таким чином, декодер заглиблюється в решітку, приймаючи рішення шляхом виключення менш ймовірних шляхів. Попередня відмова від малоімовірних шляхів спрощує процес декодування. Завдання відбору оптимальних шляхів можна виразити як вибір кодового слова з максимальною метрикою правдоподібності або мінімальною метрикою відстані.

На вхід декодера надходить сегмент послідовності  $r$  довжиною  $b$ , що перевищує кодову довжину блоку  $n$ . Назвемо  $b$  вікном декодування. Порівняємо всі кодові слова даного коду (в межах сегмента довжиною  $b$ ) з прийнятим словом і виберемо кодове слово, найближчим до прийнятого. Перший інформаційний кадр обраного кодового слова приймається в якості

оцінки інформаційного кадру декодованого слова. Після цього в декодер вводиться  $n_0$  нових символів, а введені раніше найстаріші  $n_0$  символів скидаються, і процес повторюється для визначення наступного інформаційного кадру. Таким чином, декодер Вітербі послідовно обробляє кадр за кадром, рухаючись по решітці, яка аналогічна використовуваній кодером. У кожен момент часу декодер не знає, в якому вузлі знаходиться кодер, і не намагається його декодувати. Замість цього декодер за прийнятою послідовністю визначає найбільш правдоподібний шлях до кожного вузла і визначає відстань між кожним таким шляхом і прийнятою послідовністю. Ця відстань називається мірою розбіжності шляху. В якості оцінки прийнятої послідовності вибирається сегмент, який має найменшу міру розходження. Шлях з найменшою мірою розбіжності називається вижившим шляхом.

#### Перелік посилань

1. Скляр Б. Цифровая связь. Теоретические основы и практическое применение/ Б. Скляр. Изд. 2-е, испр.: Пер. с англ. – М.: Издательский дом «Вильямс», 2003.– 1104 с.
2. Прокис Д. Цифровая связь / Д. Прокис. Пер. с англ. под ред. Д.Д. Кловского. – М.: Радио и связь, 2000. – 800 с.
3. Банкет В. Л. Завадостійке кодування в телекомунікаційних системах: навч. посіб. з вивчення модуля 4 дисципліни ТЕЗ / В.Л. Банкет, П.В. Іващенко, М.О. Іщенко. – Одеса: ОНАЗ ім. О. С. Попова, 2011. – 100 с.

#### Метод моніторингу мережі WLAN Wi-Fi

Гадомський А.В.

Науковий керівник: к.т.н. доцент Таранчук А.А.

Хмельницький національний університет

Реалізації Wi-Fi дозволяють отримати швидкість передачі даних понад 100 Мбіт/с, при цьому користувачі можуть перемішуватися між точками доступу територією покриття мережі Wi-Fi, використовуючи мобільні пристрої, які оснащені клієнтськими приймально-передавальними (ПП) пристроями Wi-Fi для отримання доступу до мережі Інтернет. Наявність ПП Wi-Fi дозволяє користувачу підключатися до комутаційного шлюзу, а також підтримувати з'єднання декількох кінцевих пристроїв користувача між собою.

В сучасних умовах зростання обсягів трафіку інтернет мереж необхідно забезпечувати користувачів якісним абонентським доступом до мережі Wi-Fi та покращувати і розвивати сферу послуг.

Дальність передавання інформації залежить від потужності передавача Wi-Fi, наявності і характеристик перешкод, типу антени тощо. Ядром

Завідувачу кафедри телекомунікацій  
та радіотехніки  
д.т.н., професору Бойку Ю.М.  
здобувача вищої освіти  
Біндера Т.С.  
ФПКТС, гр. ТРМ-19-1

### ЗАЯВА

З правилами чинного Положення «Про дотримання академічної доброчесності в Хмельницькому національному університеті» від 26.09.2020 (зі змінами від 26.11.2020), згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування заходів дисциплінарної та академічної відповідальності, ознайомлений. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщений (а) та надаю свою згоду на обробку та збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та Anti-Plagiarism) та використання роботи для виявлення плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

23 листопада 2020 р.  
дата

  
підпис

## Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 7.0%

Словари проверки: en\_US, ru\_RU, ua\_UA. Ошибок в документах: 12%

ID: 82572 Название: Модель шифрової системи зв'язку з заводським згортовим кодуванням Добавлено в БД: 2020-12-05 Авторы: Биндер Тарас Сергійович Руководители: Пятин Ілля Станіславович Консультанты: Оponentы: Полікарівських Олексій Ілліч	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	92640	921	8741 (9%)	91 (10%)

## Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы



Ім'я користувача:  
Kafedra telekom\_radlotechniki

Дата перевірки:  
05.12.2020 21:56:10 EET

Дата звіту:  
10.12.2020 12:38:59 EET

ID перевірки:  
1005376000

Тип перевірки:  
Doc vs Internet

ID користувача:  
100005656

Назва документа: Биндер Т.С. \_ ТРМ-19-1

Кількість сторінок: 84 Кількість слів: 15426 Кількість символів: 105528 Розмір файлу: 3.55 MB ID файлу: 1005668609

## 10.8% Схожість

Найбільша схожість: 2.18% з Інтернет-джерелом (<http://elar.khnu.km.ua/jspui/bitstream/123456789/7303/1/7.pdf>)

10.8% Джерела з Інтернету

171

Сторінка 86

Пошук збігів з Бібліотекою не проводився

## 0% Цитат

Вилучення цитат вимкнено

Не знайдено жодних посилань

## 0.75% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 12 слів та 0%)

0.75% Вилучення з Інтернету

271

Сторінка 87

Немає вилучених бібліотечних джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

24

## РІШЕННЯ КАФЕДРИ ТЕЛЕКОМУНІКАЦІЙ ТА РАДІОТЕХНІКИ

### ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Модель цифрової системи зв'язку з завадостійким згортковим кодуванням

Автор: Біндер Тарас Сергійович

Спеціальність: 172 Телекомунікації та радіотехніка

Науковий керівник: к.т.н., доцент Пятін Ілля Станіславович

Після аналізу звіту подібності зроблено такий висновок:


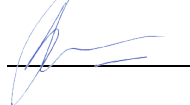
№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних). Робота приймається до захисту.	-
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	відповідає
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	-
4	Робота містить навмисні текстові спотворення, передбачувані спроби укріплення запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	-
5	Інше:	-

Підтвердження: Виявленні запозичення не є плагіатом так як розміщені в розділах, які не описують безпосередньо авторське дослідження (є власні терміни, визначення тощо), складають 10,8% та мають посилання на приведений список літературних джерел.

«07» грудня 2020 р.

Науковий керівник

Завідувач кафедру ТР

Пятін І.С.

Бойко Ю.М.

## Рецензія опонента

на дипломну роботу магістра Біндер Т.С. за темою:

«Модель цифрової системи зв'язку з завадостійким згортковим кодуванням»

Завадостійким називається кодування, при якому здійснюється виявлення та виправлення помилок в прийнятих кодових комбінаціях. Згорткове кодування задають за допомогою породжуючих поліномів, які визначають структуру двійкового кодера згорткового коду, тому тема роботи актуальна.

Наукова новизна полягає у енергетичній ефективності цифрових систем зв'язку з згортковими кодами. Згортковий код з швидкістю  $R=0,5$  ефективніше коду Ріда-Соломона (7, 5) на 2 дБ. Отримали подальший розвиток методи завадостійкого кодування і декодування.

Перший розділ присвячений аналізу методів завадостійкого кодування. Згорткові коди породжуються дискретною лінійною інваріантною в часі системою. Згорткові коди ефективно працюють в каналі з білим шумом, але погано справляються з пакетами помилок

Другий розділ присвячений аналізу принципів декодування з м'яким рішенням. Застосування методів м'якого декодування дозволяє знизити відношення сигнал / шум на 2 дБ при збереженні заданої достовірності передачі інформації в порівнянні з методами жорсткого декодування

Третій розділ присвячений аналізу кодів, що ітеративно декодуються. Систематичні згорткові коди, що застосовуються в якості складових кодів турбокодера, характеризуються паралельністю роботи декількох кодерів. При конструктивному поєднанні декількох згорткових кодерів утворюється система, відмінною ознакою якої є потоковий метод кодування і модель системи з багатьма входами-виходами.

Четвертий розділ присвячений дослідженню моделей систем зв'язку з згортковими кодами. Символи на виході згорткового кодера залежать не тільки від інформаційних символів на вході, а й від попередніх символів на

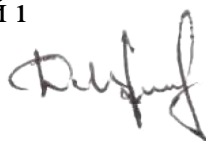
його вході. Згорткове кодування задають за допомогою породжуючих поліномів, які визначають структуру двійкового кодера згорткового коду.

За змістом робота є закінченою, викладення матеріалу послідовне і логічне. Наведені у роботі формули, моделі та висновки мають достатнє обґрунтування та пояснення. Мова викладення роботи є технічно грамотною та зрозумілою.

Серед недоліків роботи можна відмітити недостатньо розгорнутий аналіз можливостей реалізації кодерів і декодерів загорткових кодів і їх відображення у Simulink-моделях.

В цілому, магістерська робота виконана на сучасному науково-технічному рівні і заслуговує оцінку «добре», а Біндер Т.С. – присвоєння кваліфікації магістра з спеціальності «Телекомунікації та радіотехніка».

Доцент кафедри автоматизації,  
комп'ютерно-інтегрованих технологій і  
телекомунікацій, к.т.н., доц.



Макаришкін Д.А.