

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра телекомунікацій, медійних та інтелектуальних технологій

**ДИПЛОМНА РОБОТА**

Другого (магістерського) рівня

Освітній рівень

ГАЛУЗЬ ЗНАНЬ 17 Електроніка, автоматизація та електронні комунікації

Шифр і назва галузі

СПЕЦІАЛЬНІСТЬ 172 Електронні комунікації та радіотехніка

Шифр і назва спеціальності

ОСВІТНЯ ПРОГРАМА Електронні інформаційно-комунікаційні системи та мережі

Назва освітньої програми

на тему «Метод інтелектуального декодування завадостійких кодів у 5G/6G»

ДРТР. 240300.01.08 ПЗ

Виконала:

здубувач 2 курсу, група ЕКР<sub>М</sub>-24-1

підпис

Дар'я СУБЕРЛЯК

Ініціали, прізвище

Керівник: д-р техн. наук, професор

підпис

Юлій БОЙКО

Ініціали, прізвище

До захисту допускаю:

Зав. кафедри: д-р техн. наук, професор

підпис

Сергій ПІДЧЕНКО

Ініціали, прізвище

«01» грудня 2025р.

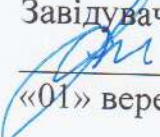
Хмельницький, 2025

## Хмельницький національний університет

Факультет	«Інформаційних технологій»
Кафедра	«Телекомунікацій, медійних та інтелектуальних технологій»
Освітній рівень:	Другий (магістерський)
Галузь знань:	17 Електроніка, автоматизація та електронні комунікації
Спеціальність:	172 Електронні комунікації та радіотехніка
Освітня програма:	Електронні інформаційно-комунікаційні системи та мережі

ЗАТВЕРДЖУЮ

Завідувач кафедру ТМІТ

 Сергій ПІДЧЕНКО

«01» вересня 2025 р.

**ЗАВДАННЯ НА ДИПЛОМНУ РОБОТУ**СУБЕРЛЯК Дар'ї Олександрівні1 Тема роботи: «Метод інтелектуального декодування завадостійких кодів у 5G/6G»керівник роботи: БОЙКО Юлій Миколайович, д-р техн. наук, професор

Затверджено наказом по університету від «25» серпня 2025 р. № 65

2 Строк подання здобувачем роботи на кафедру 24.11.2025

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

Мета роботи: розробка методу інтелектуального декодування завадостійких кодів LDPC та Polar у системах 5G/6G, яка забезпечує підвищення ефективності декодування в умовах шумів і завад за рахунок використання методів машинного навчанняОб'єкт дослідження: процес завадостійкого кодування і декодування у бездротових системах зв'язку п'ятого та шостого покоління.Предмет дослідження: методи та алгоритми інтелектуального декодування LDPC та Polar кодів у системі 5G NR та 6G.

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити):

1 Аналіз завадостійких кодів у 5G системах. 2 Математичний опис алгоритмів декодування завадостійких кодів у 5G. 3 Розробка методу інтелектуального декодування завадостійких кодів для 5G/6G. 4 Експериментальні дослідження інтелектуального декодування завадостійких кодів для 5G/6G.

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень):

презентаційні слайди

## 6 Консультанти розділів дипломної роботи

Розділ	Прізвища, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7 Дата видачі завдання 01.09.2025

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування виду роботи	Форма звітності, термін виконання	Відмітка наукового керівника
1.	Розробка завдання на дипломну роботу	08.09.2025р.	<i>виконано</i>
2.	Складання індивідуального плану на дипломну роботу	16.09.2025р.	<i>виконано</i>
3.	Написання першого (теоретичного) розділу	20.10.2025р.	<i>виконано</i>
4.	Написання другого розділу	05.11.2025р.	<i>виконано</i>
5.	Написання третього та четвертого розділу	20.11.2025р.	<i>виконано</i>
6.	Написання вступу і загальних висновків та пропозицій до дипломну роботу	22.11.2025р.	<i>виконано</i>
7.	Формулювання висновків до роботи	24.11.2025р.	<i>виконано</i>
8.	Підготовка матеріалів для апробації	24.11.2025р.	<i>виконано</i>
9.	Оформлення пояснювальної записки до ДР	26.11.2025р.	<i>виконано</i>

Здобувач

Керівник роботи

  
(підпис, дата)

  
(підпис, дата)

Дар'я СУБЕРЛЯКЮлій БОЙКО

## АНОТАЦІЯ

Тема дипломної роботи: «Метод інтелектуального декодування завадостійких кодів у 5G/6G»

Автор роботи: СУБЕРЛЯК Дар'я Олександрівна

Керівник роботи: д-р техн. наук, професор БОЙКО Юлій Миколайович.

Пояснювальна записка: 240 сторінок, 69 рисунків, 11 таблиць, 60 джерел.

Графічна частина: 18 презентаційних слайдів.

КЛЮЧОВІ СЛОВА: ІНТЕЛЕКТУАЛЬНЕ ДЕКОДУВАННЯ, LDPC-КОДИ, POLAR-КОДИ, НАВЧАННЯ З ПІДКРІПЛЕННЯМ (DRL), 5G/6G СИСТЕМИ ЗВ'ЯЗКУ, ЗАВАДОСТІЙКІСТЬ, BIT ERROR RATE (BER), ГЛИБОКЕ НАВЧАННЯ (DEEP LEARNING).

*Метою дипломної роботи є розробка методу інтелектуального декодування завадостійких кодів LDPC та Polar у системах 5G/6G, яка забезпечує підвищення ефективності декодування в умовах шумів і завад за рахунок використання методів машинного навчання.*

*Об'єктом дослідження є процес завадостійкого кодування і декодування у бездротових системах зв'язку п'ятого та шостого покоління.*

*Предметом дослідження є методи та алгоритми інтелектуального декодування LDPC та Polar кодів у системі 5G NR та 6G.*

В дипломній роботі досліджено можливість підвищення ефективності декодування завадостійких кодів LDPC та Polar у системах 5G/6G шляхом застосування методів глибокого машинного навчання та навчання з підкріпленням (Deep Reinforcement Learning, DRL). Запропоновано гібридну архітектуру інтелектуального декодера DRL+Soft, здатну адаптувати стратегію декодування залежно від типу завад у каналі (AWGN, Rayleigh). Проведено порівняльне моделювання класичних та інтелектуальних алгоритмів декодування, що показало перевагу DRL-підходу за такими показниками, як затримка, точність (BER/BLER) і адаптивність до каналів з зашумленням. Результати можуть бути використані при проектуванні приймачів для перспективних систем зв'язку шостого покоління, де критично важливими є висока завадостійкість, низька латентність (затримка) і енергоефективність.

*Суберляк Д.О.*  
Ініціали, прізвище здобувача

*[Підпис]* 24.11.2025  
Підпис, Дата

## ЗМІСТ

	С.
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ .....	8
ВСТУП .....	10
<b>1 АНАЛІЗ ЗАВАДОСТІЙКИХ КОДІВ У 5G СИСТЕМАХ.....</b>	<b>17</b>
1.1 Загальна характеристика каналу зв'язку в 5G .....	17
1.2 Вимоги до завадостійкого кодування в 5G .....	20
1.3 LDPC та Polar Codes - основні принципи, структури, переваги та обмеження.....	24
1.3.1 Опис LDPC-кодів.....	24
1.3.2 Опис Polar-кодів.....	25
1.4 Використання LDPC для downlink (низхідного), Polar codes для uplink (висхідного) каналів у 5G NR.....	29
1.5 Висновки до першого розділу .....	35
<b>2 МАТЕМАТИЧНИЙ ОПИС АЛГОРИТМІВ ДЕКОДУВАННЯ ЗАВАДОСТІЙКИХ КОДІВ У 5G.....</b>	<b>37</b>
2.1 Ітеративні алгоритми декодування для LDPC .....	37
2.1.1 Методика декодування LDPC-кодів за алгоритмом Belief Propagation (BP).....	37
2.1.2 Методика декодування LDPC-кодів за алгоритмом Min-Sum (MS).....	41
2.1.3 Методика декодування LDPC-кодів за алгоритмами Normalized та Offset Min-Sum (NMS/OMS).....	43
2.2 Алгоритми декодування Polar-кодів .....	48
2.2.1 Методика декодування Polar-кодів за алгоритмом Successive Cancellation (SC).....	48
2.2.2 Методика декодування Polar-кодів за алгоритмом Successive Cancellation List (SCL).....	53
2.2.3 Методика декодування Polar-кодів за алгоритмом CRC-aided SCL (CA-SCL).....	55
2.2.4 Методика декодування Polar-кодів за алгоритмом Fast SC/SC Flip.....	57
2.3 Порівняння продуктивності (BER/BLER, складність, латентність).....	59
2.4 Висновки до другого розділу .....	62
<b>3 РОЗРОБКА МЕТОДУ ІНТЕЛЕКТУАЛЬНОГО ДЕКОДУВАННЯ ЗАВАДОСТІЙКИХ КОДІВ ДЛЯ 5G/6G.....</b>	<b>64</b>
3.1 Обґрунтування можливостей використання AI у декодуванні.....	64

3.2	Архітектури нейромереж для декодування.....	69
3.2.1	Використання архітектури DNN (Deep Neural Networks) для розробки інтелектуального декодера 5G .....	69
3.2.2	Використання архітектури рекурентних нейронних мереж RNN/LSTM (при використанні каналу з пам'яттю) для конструювання інтелектуального декодера 5G .....	81
3.2.3	Особливості застосування концепції Graph Neural Networks (GNN) для LDPC графів при розробці декодерів 5G/6G .....	88
3.2.4	Використання концепції Deep Reinforcement Learning (DRL) – обґрунтування вибору та методика конструювання декодерів (LDPC, Polar codes) для 5G/6G .....	95
3.3	Аналіз нейромережових підходів до побудови декодерів .....	103
3.4	Архітектура запропонованої системи гібридного інтелектуального декодування (DRL+Soft-Decision декодер) для 5G/6G .....	106
3.5	Висновки до третього розділу .....	113
4	ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ІНТЕЛЕКТУАЛЬНОГО ДЕКОДУВАННЯ ЗАВАДОСТІЙКИХ КОДІВ ДЛЯ 5G/6G .....	115
4.1	Побудова моделі телекомунікаційного каналу з інтелектуальним декодуванням завадостійких кодів для 5G .....	115
4.2	Розгорнута архітектури фізичного каналу 5G .....	119
4.3	Оцінка продуктивності каналу 5G з інтелектуальним декодуванням за індикатором завадостійкості BER від SNR (канали AWGN, Rayleigh).....	125
4.3.1	Умови проведення експериментальних досліджень.....	125
4.3.1.1	Вимоги до LDPC-кодування.....	126
4.3.1.2	Вимоги до Polar-кодування.....	128
4.3.1.3	Вимоги до форматів модуляції.....	132
4.3.1.4	Вимоги до каналу зв'язку (телекомунікаційного каналу).....	133
4.3.1.5	Вимоги до агента DRL.....	133
4.3.1.6	Вимоги до BER.....	135
4.3.2	Експериментальне дослідження продуктивності системи інтелектуального декодування QC-LDPC-кодів на основі DRL у каналі AWGN та Релеєвському каналі.....	137
4.3.3	Експериментальне дослідження продуктивності системи інтелектуального декодування Polar-кодів на основі DRL у каналі AWGN та Релеєвському каналі.....	149
4.4	Оцінка продуктивності каналу 5G з інтелектуальним декодуванням за індикатором BLER.....	161

4.5 Дослідження основних метрик продуктивності для DRL-декодера (DRL+Soft).....	169
4.6 Висновки до четвертого розділу .....	184
ВИСНОВКИ.....	187
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ .....	191
Додаток А Програма візуалізації ітеративного обміну повідомленнями у графі Таннера для LDPC-коду з використанням GNN-декодування.....	199
Додаток Б Програма демонстрації застосування концепції Deep Reinforcement Learning (DRL) з використанням архітектури Actor-Critic для задачі інтелектуального декодування LDPC та Polar кодів .....	205
Додаток В Апробаційна стаття у фаховому виданні за результатами виконання дипломної роботи.....	210
Додаток Г Презентаційні матеріали за результатами виконання дипломної роботи .....	222

**ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ**

AI -	Штучний інтелект (Artificial Intelligence)
AWGN -	Білий гаусівський шум із постійною спектральною щільністю (Additive White Gaussian Noise)
BER -	Ймовірність бітової помилки (Bit Error Rate)
BLER -	Ймовірність блочної помилки (Block Error Rate)
BP -	Алгоритм поширення переконань (Belief Propagation)
CNN -	Згорткова нейронна мережа (Convolutional Neural Network)
DNN -	Глибока нейронна мережа (Deep Neural Network)
DRL -	Глибоке навчання з підкріпленням (Deep Reinforcement Learning)
$E_b/N_0$ -	Відношення енергії біта до спектральної щільності шуму
FCN -	Повнозв'язна нейронна мережа (Fully Connected Network)
GNN -	Графова нейронна мережа (Graph Neural Network)
LDPC -	Коди з низькою щільністю перевірок на парність (Low-Density Parity-Check)
LLR -	Логарифмічне відношення правдоподібності (Log-Likelihood Ratio)
MIMO -	Багатоантенна система (Multiple Input Multiple Output)
MS -	Алгоритм мінімальної суми (Min-Sum)
NMS -	Нормалізований алгоритм мінімальної суми (Normalized Min-Sum)
OMS -	Алгоритм мінімальної суми із зсувом (Offset Min-Sum)
PER -	Ймовірність пакетної помилки (Packet Error Rate)
QCLDPC -	Квазіциклічний LDPC-код (Quasi-Cyclic LDPC)
QPSK -	Квадратурна фазова маніпуляція (Quadrature Phase Shift Keying)

ReLU -	Функція активації "відсікання по нулю" (Rectified Linear Unit)
RL -	Навчання з підкріпленням (Reinforcement Learning)
SCL -	Списокова послідовна відміна (Successive Cancellation List)
SC -	Послідовна відміна (сканування) (Successive Cancellation)
SDR -	Програмно-конфігурована радіосистема (Software Defined Radio)
SINR -	Відношення сигнал/шум + завада (Signal-to-Interference-plus-Noise Ratio)
SNR -	Відношення сигнал/шум (Signal-to-Noise Ratio)
SoC -	Система на кристалі (System on Chip)
UE -	Абонентський пристрій (User Equipment)
URLLC -	Надійний зв'язок з надмалою затримкою (Ultra-Reliable Low-Latency Communication)

## ВСТУП

У сучасному інформаційному суспільстві розвиток бездротових телекомунікацій став ключовим чинником формування нових технологічних укладів, цифрової трансформації промисловості та повсякденного життя. Зростаюча потреба у високошвидкісному доступі до даних, мінімізації затримок передачі інформації, а також надійному функціонуванні комунікаційних систем у різноманітних умовах стимулює безперервний розвиток мобільного зв'язку. Особливої актуальності набувають дослідження в напрямку вдосконалення фізичного рівня систем зв'язку — компоненту, який безпосередньо відповідає за надійність, ефективність і стійкість передачі даних в умовах реальних завод.

З переходом до мереж п'ятого покоління (5G) [1] постали нові виклики, пов'язані з необхідністю підтримки мільйонів пристроїв на квадратний кілометр, зниженням енергоспоживання пристроїв IoT, обслуговуванням критичних застосувань у режимі реального часу, таких як автономний транспорт, розумні фабрики та телемедицина. У цьому контексті ключову роль відіграють сучасні методи заводостійкого кодування, які забезпечують виявлення та виправлення помилок у цифровому потоці, що передається каналом зв'язку.

Традиційні схеми корекції помилок, які раніше використовувались у системах 3G та LTE, виявляються недостатньо ефективними за нових умов: складних моделей завод, динамічної зміни середовища передачі, а також потреби у гнучкому використанні спектра. Це сприяло впровадженню нових типів кодів у стандарт 5G NR — LDPC (Low-Density Parity-Check) для downlink (висхідного) та Polar Codes для uplink (низхідного) телекомунікаційного каналу, що значно підвищують ефективність кодування при збереженні помірної складності реалізації [2, 3].

Разом із цим, парадигма штучного інтелекту (ШІ) [4], яка активно впроваджується у різні сфери людської діяльності, демонструє великий потенціал і у сфері телекомунікацій. Завдяки здатності до навчання, адаптації та прогнозування, інтелектуальні алгоритми відкривають нові горизонти для побудови ефективніших, адаптивних і стійких до завад комунікаційних систем. Інтеграція ШІ у процеси кодування і декодування сигналів дозволяє наблизитися до межі пропускнуої здатності каналу (Shannon limit), знизити латентність і підвищити точність декодування в реальному часі.

У світлі таких тенденцій дослідження новітніх методів декодування, що поєднують можливості класичної теорії кодування і сучасних досягнень машинного навчання, має як теоретичну, так і практичну цінність. Особливо перспективним напрямом є створення інтелектуальних декодерів LDPC та Polar кодів, які можуть адаптуватися до умов каналу і забезпечувати високу ефективність при зниженому енергоспоживанні та затримці [5].

Отже, дана дипломна робота «Методика інтелектуального декодування завадостійких кодів у 5G/6G» присвячена розробці та аналізу ефективних LDPC та полярних кодів, що мають високу здатність до корекції помилок і забезпечують надійність передавання даних у сучасних телекомунікаційних системах. Проведене дослідження є важливим внеском у подальший розвиток бездротових технологій зв'язку, зокрема для систем зв'язку 5G/6G покоління, де якість з'єднання та завадостійкість відіграють ключову роль.

*Актуальність роботи* безпосередньо пов'язана із всеосяжним розвитком стандартів п'ятого покоління мобільного зв'язку (5G) який став ключовим етапом еволюції телекомунікаційних систем, спрямованим на задоволення зростаючих вимог до пропускнуої здатності, мінімізації затримок, підтримки високої щільності підключень та забезпечення надійного зв'язку у складних радіоумовах. Технології 5G NR (New Radio) забезпечують гнучку архітектуру та адаптивне управління ресурсами, однак саме фізичний рівень

(PHY), і зокрема – система завадостійкого кодування, відіграє критичну роль у досягненні цільових показників системи.

У 5G, у порівнянні з попередніми поколіннями зв'язку (LTE, 3G), збільшуються вимоги до енергоефективного та швидкодіючого коригування помилок [6]. Сучасні середовища передачі включають канали з адитивним білим гаусовим шумом (AWGN) та канали зі згасанням типу Rayleigh – надзвичайно динамічні й складні у моделюванні. Це вимагає від завадостійких кодів не лише високого рівня виправлення помилок ( $BLER < 10^{-5}$ ), а й низької латентності, що є критичним для таких сценаріїв, як автономне керування, промисловий Інтернет речей (IIoT), віддалена хірургія та доповнена реальність (AR/VR).

З цією метою в 5G NR [7, 8] були впроваджені LDPC-коди для низьхідного та Polar Codes для висхідного каналів — дві фундаментально різні технології, які забезпечують компроміс між продуктивністю, складністю реалізації та часом декодування. Проте, традиційні методи декодування цих кодів, зокрема Belief Propagation для LDPC та Successive Cancellation для Polar Codes, мають обмеження у реальних умовах які проявляються у високому енергоспоживанні, складності апаратної реалізації, чутливості до параметрів каналу та обмеженої адаптивності.

У відповідь на ці виклики, інтелектуальні алгоритми III, такі як глибокі нейронні мережі (DNN) [9], рекурентні мережі (RNN, LSTM)[10], автоенкодери, а також методи підкріплювального навчання (Reinforcement Learning) [11], демонструють значний потенціал для підвищення ефективності декодування. Наприклад, глибоке навчання може замінити або доповнити окремі етапи класичних декодерів, адаптуючи обробку до конкретних характеристик каналу в режимі реального часу. Такі підходи забезпечують скорочення затримок, зменшення енергоспоживання та підвищення

ймовірності правильного декодування у складних сценаріях, включаючи мобільні, багатошляхові або сильно зашумлені канали.

Крім того, застосування ШІ дозволяє реалізувати гнучке, контекстно-залежне кодування та декодування на основі прогнозування умов каналу (Channel Prediction), що особливо важливо в сценаріях з масовим підключенням пристроїв (mMTC) або наднадійним низьколатентним зв'язком (URLLC).

Практична значущість впровадження інтелектуальних методів у завадостійке кодування для 5G полягає не лише у покращенні технічних показників, а й у спрощенні апаратної реалізації через зменшення кількості ітерацій та підвищення інтеграції з іншими інтелектуальними модулями системи зв'язку. Це сприяє створенню адаптивних, енергоефективних модемів нового покоління, які здатні підтримувати динамічні сценарії комунікації в умовах Інтернету речей, Smart City, Industry 4.0 та майбутніх 6G-систем.

Таким чином, дослідження ефективності та перспектив використання сучасних завадостійких кодів, зокрема LDPC та Polar Codes [12, 13], у поєднанні з алгоритмами штучного інтелекту, є надзвичайно актуальним і має як фундаментальну наукову цінність, так і високу прикладну значущість для майбутніх бездротових технологій.

*Метою* дипломної роботи є розробка методу інтелектуального декодування завадостійких кодів LDPC та Polar у системах 5G/6G, яка забезпечує підвищення ефективності декодування в умовах шумів і завад за рахунок використання методів машинного навчання.

Для досягнення цієї **мети** в магістерській роботі поставлено такі завдання:

1. Провести аналіз основних характеристик каналів зв'язку в 5G та вимог до завадостійкого кодування.

2. Дослідити класичні алгоритми декодування LDPC та Polar кодів, їх переваги та обмеження в контексті застосування в 5G.

3. Розробити архітектуру інтелектуального декодування з використанням сучасних методів машинного навчання.

4. Провести моделювання та порівняльний аналіз ефективності традиційного і інтелектуального підходів до декодування.

*Метод дослідження*, який використано в магістерській роботі поєднує аналітичний підхід, комп'ютерне моделювання та елементи машинного навчання для комплексної оцінки ефективності інтелектуальних декодерів у середовищі 5G/6G. На першому етапі проведено систематизований аналіз сучасних джерел наукової літератури щодо особливостей каналів зв'язку у 5G (AWGN, Rayleigh), властивостей завадостійких кодів (LDPC, Polar) та класичних алгоритмів їх декодування. Це дозволило виявити ключові недоліки традиційних рішень та обґрунтувати доцільність використання підходів штучного інтелекту.

Наступним етапом є розроблення архітектури інтелектуального декодера, яка базується на використанні методів глибокого навчання (Deep Learning) та навчання з підкріпленням (Reinforcement Learning). Обрані моделі реалізуються у середовищі MATLAB/Simulink із залученням інструментів нейромережевого моделювання та гібридних алгоритмів обробки сигналів. Особлива увага приділяється структурі вхідних даних для навчання — імітуються типові сценарії передачі сигналу з різними характеристиками шумів та згасань.

Після реалізації моделі здійснюється порівняльне комп'ютерне моделювання традиційного та інтелектуального декодування на рівні блочної помилки (BLER) і бітової помилки (BER) у різних сценаріях [14]. Для верифікації результатів застосовується набір типових метрик, таких як точність декодування, кількість ітерацій, час обробки, латентність та

енергоспоживання. Експерименти виконуються з варіацією параметрів каналу (SNR, тип згасання) для оцінки адаптивності інтелектуальних підходів.

Завдяки цьому підходу забезпечується достовірна оцінка практичної придатності запропонованих моделей у реальних умовах мобільного зв'язку. Методика дослідження дозволяє не лише порівняти ефективність різних декодувальних стратегій, але й виявити оптимальні конфігурації для подальшої апаратної реалізації у перспективних телекомунікаційних системах.

**Об'єктом дослідження** є процес завадостійкого кодування і декодування у бездротових системах зв'язку п'ятого та шостого покоління.

**Предметом дослідження** є методи та алгоритми інтелектуального декодування LDPC та Polar кодів у системі 5G NR та 6G.

Таким чином, дослідження буде концентруватись на питаннях завадостійкого кодування та декодування інформації в бездротових телекомунікаційних системах п'ятого покоління (5G), зокрема в рамках стандарту 5G NR. Акцент буде зосереджений на пошуку структурних, алгоритмічних та інтелектуальних методів декодування LDPC та Polar кодів, орієнтовані на підвищення ефективності обробки сигналів у зашумлених каналах зв'язку (AWGN, Rayleigh), з урахуванням вимог до швидкодії, латентності та точності у 5G-системах.

***Наукова новизна:***

1 Дістала подальшого розвитку методика інтеграції алгоритмів штучного інтелекту, зокрема методів глибокого навчання та навчання з підкріпленням, у процес декодування LDPC та Polar кодів у бездротових системах п'ятого покоління.

2 Вперше запропоновано нову архітектуру DRL+Soft-декодера для каналів 5G/6G, здатну адаптувати стратегію декодування залежно від типу завад та умов каналу. Запропонована адаптивна архітектура інтелектуального

декодера здатна змінювати стратегію декодування залежно від умов каналу (AWGN, Rayleigh fading) та характеристик вхідного сигналу.

3 Вперше здійснено повноцінне порівняння DRL-декодера з класичними алгоритмами у симуляційному середовищі MATLAB із багатьма типами шуму (AWGN, Rayleigh, імпульсний, фазовий). Проведено порівняльний аналіз ефективності класичних та інтелектуальних алгоритмів декодування в умовах змінного завадового середовища на основі комп'ютерного моделювання в середовищі Simulink.

***Практичне значення:***

Результати дослідження можуть бути використані під час проектування приймачів для систем 5G/6G [15], де критично важливими є надійність, низьке енергоспоживання та здатність до роботи в умовах високої зашумленості. Таким чином результати дослідження будуть корисними для застосування при проектуванні приймальних трактів мобільних систем зв'язку 5G і перспективних систем 6G, де важливими є висока завадостійкість, швидкодія, низька латентність та енергоефективність. Запропоновані підходи до інтелектуального декодування можуть бути реалізовані в реальному часі на базі програмно-апаратних засобів (наприклад, SDR або SoC-платформи), що сприяє підвищенню адаптивності та живучості систем зв'язку в складних радіоумовах.

За темою дипломної роботи опублікована стаття у науковому фаховому журналі "Вимірювальна та обчислювальна техніка в технологічних процесах" № 3 за 2025 р.

# 1 АНАЛІЗ ЗАВАДОСТІЙКИХ КОДІВ У 5G СИСТЕМАХ

## 1.1 Загальна характеристика каналу зв'язку в 5G

У системах 5G NR [16] каналом зв'язку називається середовище, через яке проходить радіосигнал від передавача до приймача. Основна складність полягає в багатопроменевому поширенні сигналу, втраті енергії під час проходження середовища, наявності шумів та завад.

Передача інформації в 5G виконується з використанням ортогонального частотного розділення (OFDMA), MIMO-технологій, широкого діапазону частот (від Sub-6 ГГц до мм-хвиль). Але незалежно від технології, сигнал завжди піддається впливу шумів та згасань. Для їх опису в магістерській роботі будемо використовувати математичні моделі, які дозволяють оцінити ефективність декодування.

Базовою моделлю є модель AWGN (Additive White Gaussian Noise). Відповідно формула щільності ймовірності для AWGN має форму:

$$p(n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{n^2}{2\sigma^2}\right) \quad (1.1)$$

де  $n$  - випадкова змінна шуму;  $\sigma^2$  - дисперсія шуму;  $\mu = 0$  середнє значення (для білого шуму)

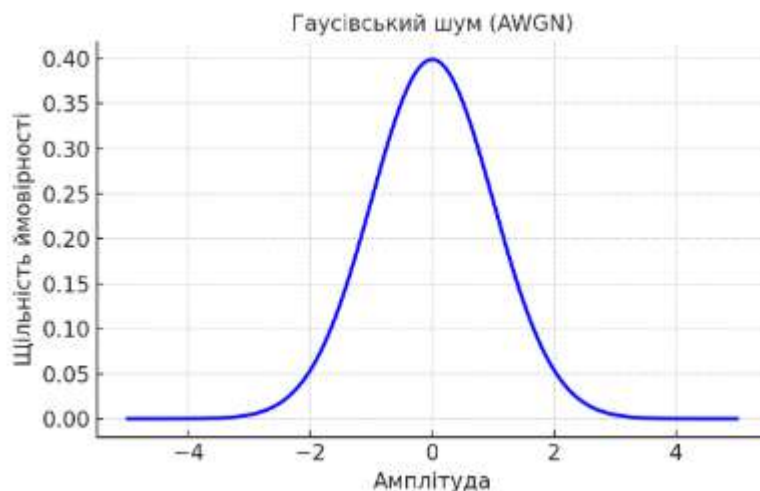


Рисунок 1.1 - Щільність ймовірності для гаусівського шуму (AWGN) з нульовим середнім значенням та стандартним відхиленням

Це базова модель, яка враховує лише адитивний шум із гаусівським розподілом. Вона буде використовуватись нами як початкове наближення для тестування алгоритмів кодування/декодування.

Для опису середовища без прямої лінії видимості (NLOS), наприклад, у міських кварталах в магістерській роботі буде використано модель Rayleigh fading (Багатопрореневе згасання). В цьому випадку сигнал доходить до приймача як сума багатьох відбитих, заломлених або розсіяних хвиль.

В роботі використовувалась формула Rayleigh-розподілу у формі:

$$p(r) = \frac{r}{\sigma^2} \exp\left(-\frac{r^2}{2\sigma^2}\right), r \geq 0 \quad (1.2)$$

де  $r$  – амплітуда сигналу;  $\sigma$  - масштабний параметр (визначає середню потужність).

Загалом Rayleigh-згасання відбувається миттєво і в цьому випадку фази та амплітуди сигналів змінюються дуже швидко. Це створює глибокі спади (fading), які суттєво впливають на якість зв'язку.

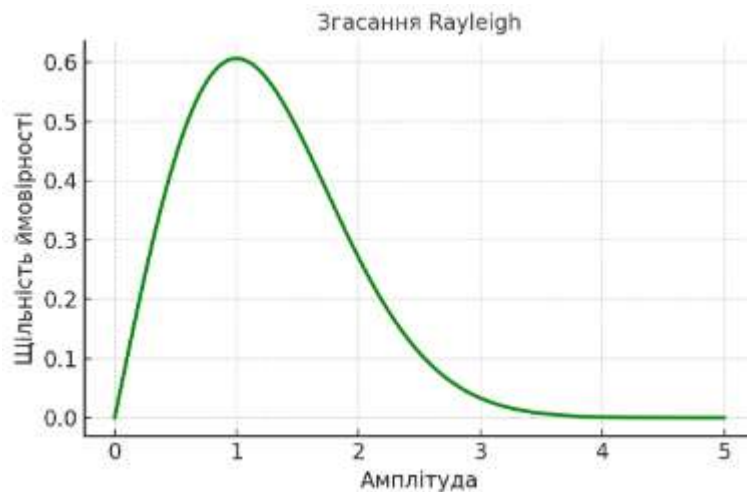


Рисунок 1.2 - Щільність ймовірності для згасання Rayleigh, яке використовується для моделювання багатошляхових ефектів у бездротовому середовищі, де сигнал відбивається від багатьох перешкод перед досягненням приймача

Акцентуємо увагу на тому, що найбільш наближена до реальних умов у 5G є комбінована модель Rayleigh + AWGN. Вона враховує випадкові згасання сигналу + адитивний шум. У моделюванні це реалізується як:

$$y(t) = h(t) \cdot x(t) + n(t) \quad (1.3)$$

де  $x(t)$  - переданий сигнал;  $h(t)$  – коефіцієнт каналу (Rayleigh-змінна);  $n(t)$  - AWGN шум;  $y(t)$  – прийнятий сигнал.

Таким чином, формально, ефективність декодування залежить від таких характеристик каналу як SNR (Signal-to-Noise Ratio) – відношення сигнал-шум [17], BER (Bit Error Rate) - показник якості декодування у вигляді ймовірності (коефіцієнта) бітової помилки; BLER (Block Error Rate) – ймовірність блокової помилки яка застосовується в 5G, оскільки передаються блоки бітів; Latency - затримка та Outage Probability - ймовірність втрати зв'язку.

На рисунку 1.3 представлено спрощену схему каналу зв'язку 5G з опцією інтелектуального кодування.

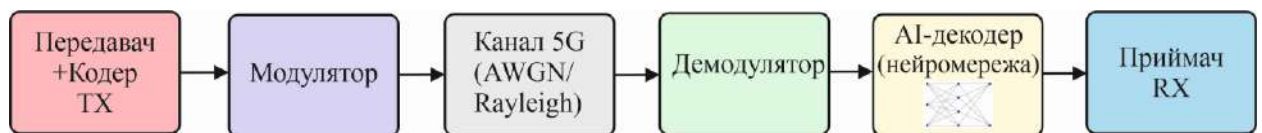


Рисунок 1.3 - Структура каналу зв'язку 5G

Типовий тракт обробки сигналу в системі 5G починається з інформаційних бітів демодулятора, що подаються на вхід каналного кодера (LDPC або Polar-коди). Метою цього етапу є підвищення завадостійкості шляхом додавання надлишкової інформації для подальшого виявлення та виправлення помилок. Для LDPC-кодів зазвичай застосовують матричне кодування з низькою щільністю одиниць, а для Polar-кодів — побудова згідно принципу канального комбінування і розділення (channel polarization) (на рис. 1.3. процедури кодування умовно непоказані і реалізуються на базі TX).

Закодовані біти передаються на модулятор, де вони перетворюються в комплексні сигнали відповідно до обраної схеми модуляції (наприклад, QPSK,

16QAM, 64QAM) [18, 19]. Далі сформований сигнал проходить через канал зв'язку з адитивним білим гаусовим шумом (AWGN) або зі згасанням типу Rayleigh — моделі, що відображають вплив завад, багатошляхового поширення та флуктуацій рівня сигналу.

На приймальному боці сигнал демодулюється в демодуляторі, де відновлюється потік бітів на основі оціненого символу. Після цього бітова послідовність подається на інтелектуальний декодер, реалізований із використанням методів штучного інтелекту — наприклад, глибоких нейронних мереж (DNN), рекурентних мереж (LSTM) або автоенкодерів. Ці структури здатні не лише імітувати класичні алгоритми декодування (Belief Propagation, Successive Cancellation), а й адаптивно змінювати декодувальну стратегію залежно від характеристик каналу або рівня шуму.

Завершується обробка етапом оцінки якості (наприклад, шляхом підрахунку BLER/BER або через нейронну оцінку достовірності), після чого формуються вихідні біти — тобто максимально наближене до оригіналу повідомлення.

## **1.2 Вимоги до завадостійкого кодування в 5G**

У системах зв'язку п'ятого покоління (5G) завадостійке кодування виконує ключову функцію забезпечення цілісності даних при передачі в умовах складних та варіативних каналів, зокрема таких як AWGN (Additive White Gaussian Noise) та Rayleigh fading (федінг) [20]. Основні вимоги до кодування у 5G визначаються цілями стандартизованих сценаріїв застосування, які включають URLLC (Ultra-Reliable Low-Latency Communications), eMBB (enhanced Mobile Broadband) та mMTC (massive Machine-Type Communications).

Однією з критичних метрик є BLER (Block Error Rate) [21] — ймовірність того, що цілий блок даних буде прийнятий із помилкою. У 5G встановлено цільовий рівень  $BLER < 10^{-5}$ , що означає не більше однієї помилки на 100 000

переданих блоків. Це надзвичайно суворе вимога, особливо у випадках URLLC, де втрати пакета можуть мати катастрофічні наслідки — наприклад, в автономному керуванні або віддаленій хірургії.

Другим важливим показником є латентність (затримка передачі), що відображає час між надсиланням та отриманням пакета інформації. У 5G NR латентність повинна бути меншою за 1 мс для критично важливих додатків. Для цього необхідно мінімізувати не лише час передачі по каналу, а й затримку декодування — обробку коду на приймальному боці. Це зумовлює вимоги до низької декодувальної складності та обмеження кількості ітерацій у випадку ітеративних декодерів, зокрема Belief Propagation (BP) для LDPC-кодів.

Третьою вимогою є висока швидкодія — здатність коду передавати значні обсяги даних при високій швидкості (гігабіти на секунду). Це потребує високої пропускну здатності кодування та паралельної обробки, а також гнучкого розміру кодових слів (від 64 до понад 10 000 біт). LDPC-коди у downlink забезпечують ефективне паралельне декодування, а Polar Codes в uplink підтримують ефективну серійну обробку із низьким енергоспоживанням.

Крім того, у 5G NR [16] активно використовуються Rate Matching (підгонка швидкості), Code Block Grouping (групування кодових блоків) і Hybrid ARQ (HARQ) — комбінація автоматичного повтору запитів із помилковим виправленням. Ці механізми ставлять додаткові вимоги до адаптивності кодування: код повинен підтримувати зміну кодового коефіцієнта (redundancy ratio) та структурування кодових блоків без втрати ефективності.

Отже, завадостійке кодування в 5G повинно поєднувати високу завадостійкість, низьку латентність, високу швидкодію, гнучкість щодо розміру та кодування зі змінним рівнем надмірності. У відповідь на ці вимоги стандарт 3GPP запропонував LDPC-коди для каналу downlink (низьхідний

канал) завдяки їхній ітеративній та паралельній природі, а також Polar-коди для uplink (висхідний канал), які мають хорошу ефективність на коротких блоках і добре масштабуються у вигляді Successive Cancellation декодерів або їх модифікацій. Водночас застосування інтелектуальних алгоритмів покликане ще більше оптимізувати продуктивність у контексті реального часу та змінних умов радіоканалу. Саме ці аспекти розглядаються в магістерській дипломній кваліфікаційній роботі.

Щоб систематизувати вимоги до завадостійкого кодування у 5G нижче приведено структуровану таблицю (табл. 1.1).

Таблиця 1.1 - Узагальнені вимоги до завадостійкого кодування в 5G згідно з типами сервісів, метриками та обґрунтуванням

Параметр	Цільове значення	Сценарії застосування	Обґрунтування/Примітки
<b>BLER (Block Error Rate)</b>	$< 10^{-5}$	URLLC, eMBB	Висока надійність передачі, критично важливо в автономних системах та медичних застосуваннях
<b>Латентність (затримка)</b>	$\leq 1$ мс (end-to-end)	URLLC	Низька затримка критична для управління в реальному часі
<b>Пропускна здатність</b>	$> 1$ Гбіт/с	eMBB	Підтримка потокового відео, AR/VR, масового доступу
<b>Гнучкість кодування</b>	Кодові слова: 64 – 10 240 біт	mMTC, URLLC, eMBB	Підтримка адаптивної модуляції та кодування (AMC), rate matching
<b>Обчислювальна складність</b>	Низька	Усі типи сервісів	Мінімізація кількості ітерацій, реалізація на FPGA/ASIC
<b>Енергоспоживання</b>	Мінімальне	IoT, mMTC	Важливо для автономних пристроїв та сенсорних мереж
<b>Адаптивність до каналу</b>	Висока	AWGN, Rayleigh, мобільні сценарії	Врахування змінних умов передачі, використання AI-алгоритмів

Така таблиця узагальнює основні вимоги до завадостійкого кодування в 5G згідно з типами сервісів, метриками та обґрунтуванням.

Аналіз узагальнених вимог до завадостійкого кодування у 5G представлених у табл. 1.1 дозволяє сфокусуватись на концептуальних засадах згідно яких - системи п'ятого покоління мобільного зв'язку (5G) ставлять надзвичайно жорсткі вимоги до характеристик завадостійкого кодування, що визначають загальну продуктивність та надійність передачі даних. Однією з ключових метрик є BLER (Block Error Rate), яка повинна бути не вищою за  $10^{-5}$  для забезпечення надійного зв'язку в критичних сценаріях, таких як URLLC (Ultra-Reliable Low-Latency Communications). Такий низький рівень помилок дозволяє гарантувати безперервне та точне передавання інформації навіть в умовах сильних завад.

Ще одним фундаментальним параметром є латентність — повна затримка від моменту відправлення до моменту прийому. У 5G вона повинна не перевищувати 1 мс, що є надзвичайно складним викликом для системи кодування. Це особливо важливо для додатків з управлінням у реальному часі, як-от автономне водіння або віддалена хірургія. Водночас висока пропускна здатність (до 1 Гбіт/с і більше) є вимогою сервісів eMBB (enhanced Mobile Broadband), де йдеться про потокове відео, AR/VR та інші ресурсоємні застосування.

Універсальність 5G вимагає від кодерів гнучкості в обробці пакетів даних різної довжини — від коротких повідомлень (типових для mMTC, тобто масового підключення пристроїв) до великих блоків мультимедійної інформації. Ця гнучкість реалізується через змінну довжину кодових слів і механізми адаптивної модуляції та кодування (AMC).

Нарешті, важливими аспектами залишаються низьке енергоспоживання (особливо критичне для IoT-пристроїв) і низька обчислювальна складність, що забезпечує реальну можливість впровадження на апаратному рівні (ASIC,

FPGA). Для задоволення цих вимог активно досліджуються інтелектуальні алгоритми, що дозволяють підвищити адаптивність до умов каналу (AWGN, Rayleigh fading) і зменшити навантаження на обчислювальні ресурси.

### 1.3 LDPC та Polar Codes - основні принципи, структури, переваги та обмеження

#### 1.3.1 Опис LDPC-кодів

LDPC-коди — це лінійні блокові коди, які описуються розрідженою (low-density) [22, 23] матрицею перевірки на парність  $\mathbf{H}$ , де більшість елементів дорівнюють нулю. Така матриця має розміри  $m \times n$ , де  $m$  - кількість перевірок, а  $n$  - довжина кодового слова. Кодування відбувається за допомогою генераторної матриці  $\mathbf{G}$  яка пов'язана з  $\mathbf{H}$  співвідношенням типу  $\mathbf{H} \cdot \mathbf{G}^T = 0$ .

LDPC-коди зручно зображуються у вигляді біпартитного графа Таннера, де (рис. 1.4) перевірочні вузли (check nodes) відповідають рядкам матриці  $\mathbf{H}$ , вузли змінних (variable nodes) відповідають стовпцям  $\mathbf{H}$ .

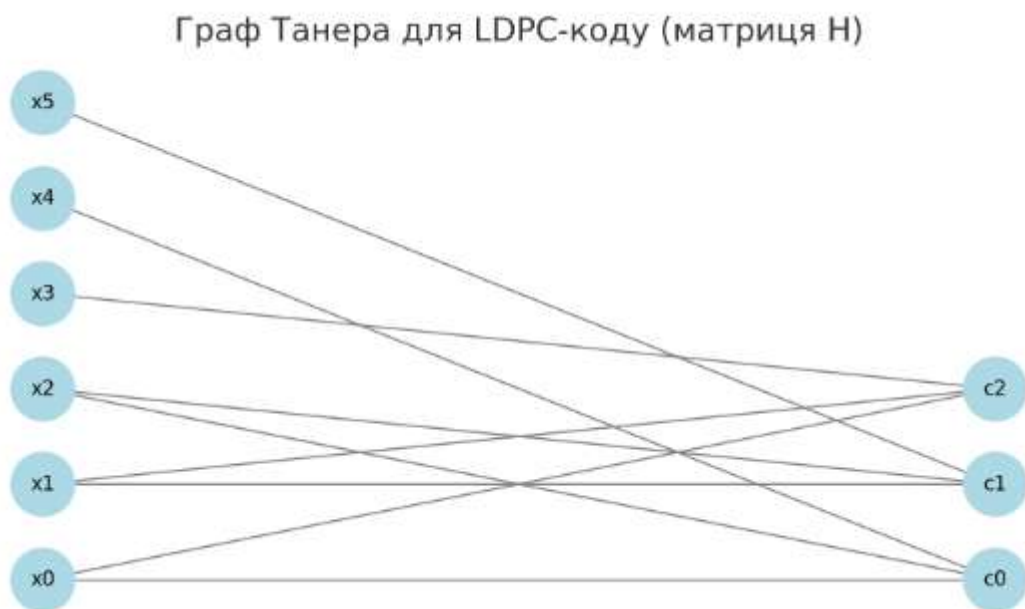


Рисунок 1.4 – Граф Таннера

На рис. 1.4 змінні вузли (ліворуч, позначені як  $x_i$ ) з'єднані з перевірочними вузлами (праворуч,  $c_i$ ) відповідно до ненульових елементів

матриці  $\mathbf{H}$ . Така візуалізація демонструє взаємозв'язки між бітами коду та перевірочними рівняннями — основа для декодування, методом Belief Propagation який буде описаний в другому розділі дипломної роботи.

Зв'язки в графі з'єднують змінні та перевірки там, де в  $H_{ij} = 1$ .

Початково LDPC-коди можуть бути визначені протографом — невеликим графом, який служить шаблоном. Таким чином, повна матриця  $\mathbf{H}$  може бути отримана розгортанням (lifting) цього протографа за допомогою матриці Кронекера і описана наступним математичним виразом:

$$\mathbf{H} = \mathbf{V} \otimes \mathbf{P} \quad (1.4)$$

де  $\mathbf{V}$  - базова матриця протографа;  $\mathbf{P}$  - перmutаційна матриця або циклічне зсувне представлення.

Тепер торкнемось питання формалізації представлення LDPC-кодів саме для 5G. Тут, у відповідності до стандарту [1] використовується квазіциклічна форма такого коду (QC-LDPC) [24]. Отже, у 5G LDPC-коди реалізовані як QC-LDPC і це означає, що підматриці  $\mathbf{H}$  складаються з циклічних перестановок одиничної матриці (тобто блоків-циркулянтів). Це суттєво спрощує реалізацію в апаратурі, а математично формування підматриць на основі циркулянтів має наступний формат:

$$\mathbf{H} = \begin{bmatrix} P_0 & P_1 & \cdots & P_{n-1} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ P_{m-1} & \cdots & \cdots & P_{n-1} \end{bmatrix} \quad (1.5)$$

де кожен  $P_i$  - це або циркулянт, або нульова матриця.

### 1.3.2 Опис Polar-кодів

Polar-коди — це лінійні блокові коди, які досягають межі пропускнуої здатності каналу для каналів з симетричним шумом (Shannon capacity-achieving) [25, 26]. Основу теорії таких кодів заклав Ердал Арікан у 2009 році через концепцію каналної поляризації [26].

Код будується через рекурсивне застосування операції Кронекера [13] до базової матриці  $\mathbf{F}_2$ , що математично представляється так:

$$\mathbf{F}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \mathbf{G}_N = \mathbf{F}_2^{\otimes n} \quad (1.6)$$

де  $\mathbf{G}_N$  - генераторна матриця для довжини  $N = 2^n$ .

Розглянемо принцип поляризації каналу зв'язку. Ідея полягає у розділенні каналу на "гарні" та "погані" підканали. В цьому випадку для гарних інформаційні біти передаються, а для поганих - рахуються замороженими (fixed to 0). Так формується масив  $\mathbf{u}$ , що містить як інформаційні, так і заморожені біти. Тоді кодові слова обчислюються за наступним виразом:

$$\mathbf{x} = \mathbf{u} \cdot \mathbf{G}_N \quad (1.7)$$

Графова структура для полярних кодів є рекурсивну природу та утворюється з'єднанням "метеликоподібних" елементів (butterfly structure), аналогічних до графів для алгоритму Шеннона або швидкого перетворення Фур'є (ШПФ) (рис. 1.5).

Отже, полярні коди, запропоновані Е. Аріканом [26], будуються на принципі каналної поляризації, що реалізується через рекурсивну побудову графа із застосуванням оператора Фур'є або метеликоподібної (butterfly) структури. Граф представляє собою орієнтований ациклічний граф (DAG), який відображає структуру генераторної матриці  $\mathbf{G}_N$  розміру  $N = 2^n$ , де під  $n$  - будемо розуміти ступінь рекурсії.

Розглянемо структуру графа за умови  $N=8$  (рис. 1.5). В структурі глибина (кількість рівнів) графа дорівнює  $\log_2 N$ . Кожен рівень реалізує одну ітерацію трансформації Арікана. На кожному рівні відбувається поєднання бітів за допомогою елементарного блоку, що виконує логічні операції XOR та передає один біт далі без змін — це власне і є "метелик". З'єднання між вузлами — це

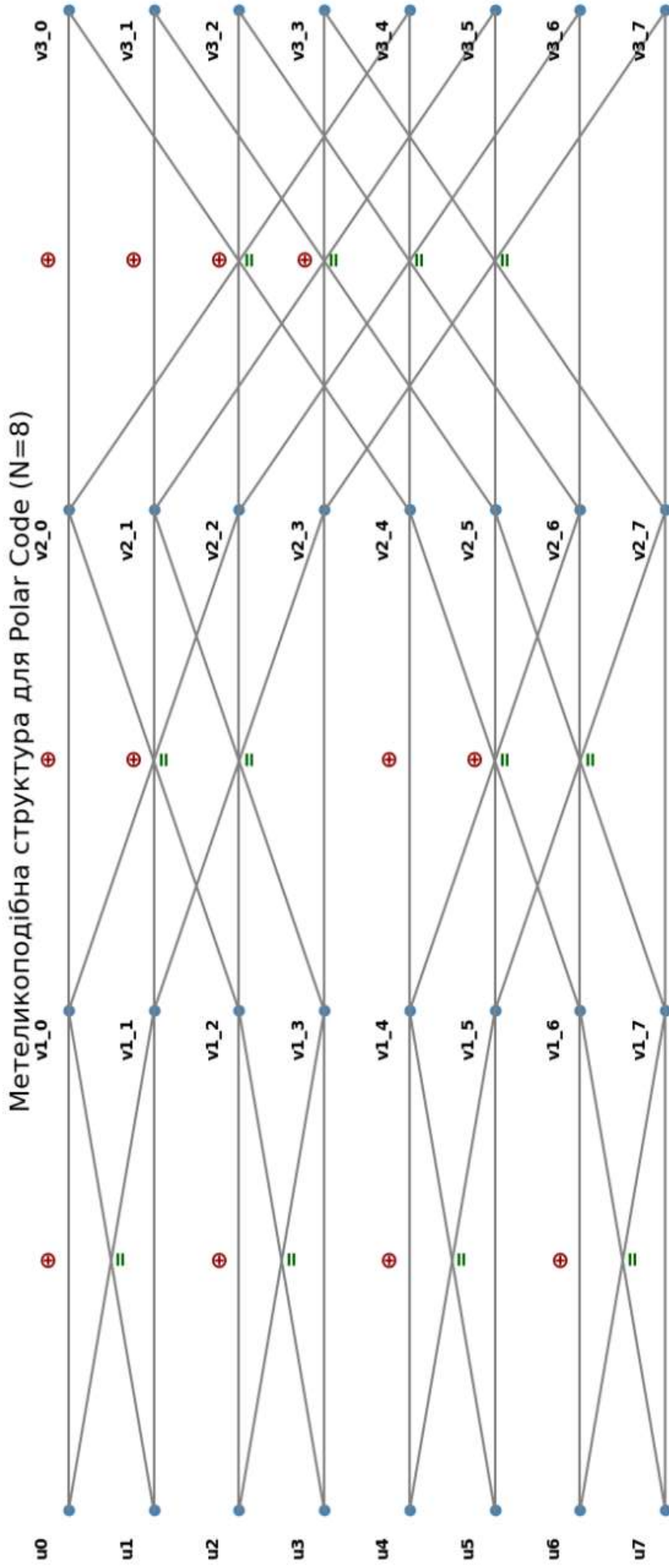


Рисунок 1.5 – Метеликоподібна структура для полярного кода  $N=8$

лінії, що перетинаються між рівнями, формуючи характерну сітчасту структуру.

Вхідні біти даних подаються на лівий край графа, а результат поляризації — на правий. Граф схожий на Fast Fourier Transform (FFT) блок-схему: на вході маємо вектор довжини  $N$ , який поетапно проходить через  $\log_2 N$  шарів логічної обробки, де формується залежність між інформаційними і замороженими (frozen) бітами. Полярне кодування ґрунтується на рекурсивному поєднанні операцій, що утворюють граф із  $\log_2 N$  шарами, кожен з яких реалізує злиття/розділення бітів. Для  $N=8$  це буде 3 рівні (оскільки  $2^3=8$ ). Отже, схема на рис. 1.5 включає 8 входів (інформаційних бітів), 3 рівні перетворень, де кожен рівень включає - XOR-операції ( $u_0 \otimes u_1$ ) і пропускання (передача бітів без змін). З'єднання виконуються так, щоб на кожному рівні кожен біт з'єднувався з іншим через схему типу "метелика". Кожен елемент схеми (вузол) представляє базову операцію XOR або передачу сигналу без змін. Перший рівень містить операції  $u_0 \otimes u_1$ ,  $u_2 \otimes u_3$ ,  $u_4 \otimes u_5$ ,  $u_6 \otimes u_7$ . Інші біти просто передаються  $u_1$ ,  $u_3$ ,  $u_5$ ,  $u_7$ . На другому рівні бітові комбінації змінюються — відбувається рекурсивне злиття. Рівень 1  $\rightarrow$  2 (другий рівень), відбувається XOR між бітами з відстанню 2, а саме  $u_0 \otimes u_2$ ,  $u_1 \otimes u_3$ ,  $u_4 \otimes u_6$ ,  $u_5 \otimes u_7$ . Рівень 2  $\rightarrow$  3 (третій рівень), а саме  $u_0 \otimes u_4$ ,  $u_1 \otimes u_5$ ,  $u_2 \otimes u_6$ ,  $u_3 \otimes u_7$ . Сині точки на схемі представляють собою вузли перетворення (обчислювальні блоки). Символ  $\oplus$  (XOR) на схемі показує, де виконується операція XOR між бітами, а символ = показує, де біт просто передається без змін. Отже,  $v_{1_0}$  містить результат обчислення з  $u_0$  і  $u_1$  ( $v_{1_0} = u_0 \otimes u_1$ ,  $v_{1_1} = u_1$  це копія). Така структура забезпечує ефективне обчислення кодових слів за допомогою полярного коду. Вона дозволяє зручно реалізовувати енкодер і (з модифікаціями) — декодер (наприклад, SC або SCL [27, 28] – це буде розглянемо в 2-му розділі магістерської роботи). Отже, маємо багаторівневу

схему перетворень вхідних бітів за допомогою XOR-ів і копій, яка візуалізує формування полярного коду з рекурсивною структурою.

Нижче в таблиці 1.2 представлено порівняльні переваги та обмеження для LDPC і полярних кодів.

Таблиця 1.2 - Порівняльні переваги та обмеження для LDPC і полярних кодів

Параметр	LDPC-коди	Polar-коди
Теоретична межа	Близько до межі Шеннона	Досягають межі Шеннона (асимптотично)
Апаратна реалізація	Добре реалізуються у QC-формі	Складна через довжину $2^n$
Декодування	Belief Propagation/Min-Sum [29]	Successive Cancellation (SC, SCL) [27]
Гнучкість довжини слова	Так (особливо QC-LDPC)	Тільки $2^n$ , потрібен rate-matching
Затримка	Низька (ітеративне)	Вища (послідовне декодування)
5G застосування	Downlink (низхідний)	Uplink (висхідний)

#### 1.4 Використання LDPC для downlink (низхідного), Polar codes для uplink (висхідного) каналів у 5G NR

У стандарті 5G New Radio (NR), розробленому організацією 3GPP, для каналів з передаванням користувацьких та керуючих даних впроваджено сучасні методи корекції помилок. На відміну від попередніх поколінь мобільного зв'язку, в 5G NR застосовано диференційований підхід до вибору кодування в залежності від напрямку передачі: LDPC-коди (Low-Density Parity-Check Codes) для downlink (від базової станції до користувача) і полярні коди (Polar Codes) для uplink (від користувача до базової станції). Таке розмежування дозволяє оптимально використовувати обчислювальні ресурси та забезпечити високий рівень завадостійкості у відповідних сценаріях передачі [24].

LDPC-коди відзначаються високою ефективністю при роботі з довгими блоками даних і мають властивість швидкої деконвергенції в ітеративному декодуванні. Завдяки своїй структурі з рідкорозрідженою матрицею парності

вони добре піддаються паралельній реалізації в апаратному забезпеченні. Це особливо важливо для downlink, де базова станція має значні обчислювальні ресурси та енергетичні можливості. Крім того, LDPC-коди легко масштабуються для адаптації до різної довжини блоків та швидкостей передачі. У 5G NR LDPC застосовуються до фізичних каналів типу PDSCH (Physical Downlink Shared Channel), забезпечуючи високу пропускну здатність мережі без шкоди для надійності.

Полярні коди, навпаки, були вперше запропоновані у 2008 році та вважаються першими кодами, які теоретично досягають межі Шеннона при нескінченній довжині коду. Однак в умовах обмеженої довжини блоків (як у керуючих каналах uplink), вони також демонструють конкурентоспроможні характеристики. Їхня головна перевага — це низька складність реалізації, особливо при використанні Successive Cancellation (SC) або Successive Cancellation List (SCL) декодування. Це робить полярні коди привабливими для uplink, де мобільні пристрої мають обмежені обчислювальні ресурси та повинні працювати енергоефективно. У 5G NR полярні коди використовуються для кодування блоків керуючої інформації (наприклад, PUCCH — Physical Uplink Control Channel).

Важливим є також те, що в 5G NR стандартизовано динамічну адаптацію кодування залежно від розміру переданого блоку (Transport Block Size — TBS) [30, 31]. LDPC-коди застосовуються переважно при великих TBS, у той час як полярні — при малих, де потрібна висока ефективність на коротких повідомленнях. Таким чином, архітектура кодування в 5G NR реалізує принцип "найкращий код для конкретного завдання", що дозволяє знизити затримки, збільшити швидкість передачі і одночасно покращити надійність зв'язку.

Такий розподіл кодів між downlink та uplink не є випадковим, а базується на глибокому аналізі практичних сценаріїв. Downlink зазвичай включає великі

обсяги мультимедійних даних, що передаються до абонента, тому тут критично важливі висока пропускна здатність та швидкість декодування. Натомість uplink часто передає короткі повідомлення — наприклад, сигнали з IoT-пристроїв або інформацію про стан мережі — що вимагає від кодів компактності та енергоефективності. Комбінація LDPC та полярних кодів у 5G NR дозволяє ефективно вирішити ці різноспрямовані задачі.

Для ілюстрації викладеного матеріалу до розділу додано інфографіку у вигляді схеми, що демонструє, які типи кодів використовуються для різних фізичних каналів у 5G, яку представлено на рис. 1.6.

**Інфографіка: Розподіл кодування в каналах 5G NR**

<b>PDSCH (Data)</b>	Тип: Користувацький	Напря: Downlink	<b>Кодування: LDPC</b>
<b>PUSCH (Data)</b>	Тип: Користувацький	Напря: Uplink	<b>Кодування: LDPC</b>
<b>PDCCH (Control)</b>	Тип: Керуючий	Напря: Downlink	<b>Кодування: Polar</b>
<b>PUCCH (Control)</b>	Тип: Керуючий	Напря: Uplink	<b>Кодування: Polar</b>
<b>RACH</b>	Тип: Керуючий	Напря: Uplink	<b>Кодування: Polar</b>

Рисунок 1.6 – Інфографіка як розподіл кодування в каналах 5G NR: PDSCH Physical Downlink Shared Channel - Фізичний канал спільного доступу (канал даних downlink); PUSCH Physical Uplink Shared Channel - Фізичний канал спільного доступу (канал даних uplink); PDCCH Physical Downlink Control Channel - Фізичний канал керування downlink; PUCCH Physical Uplink Control Channel - Фізичний канал керування uplink; RACH Random Access Channel - Канал випадкового доступу (uplink, для ініціалізації зв'язку)

Отже, відповідно до рис. 1.6 розглянемо призначення каналів у 5G. У системі 5G NR використовується набір фізичних каналів, кожен з яких виконує

специфічну функцію в процесі передавання даних та керування з'єднанням між користувачем (UE) та базовою станцією (gNB). До найважливіших із них належать канали PDSCH, PUSCH, PDCCH, PUCCH та RACH (дивитись рис. 1.6). Вони відрізняються за напрямом передачі (uplink/downlink), типом даних (керуючі або користувацькі) та використовуваними кодами завадостійкого кодування (LDPC або Polar).

PDSCH (Physical Downlink Shared Channel) — це основний канал передачі користувацьких даних від базової станції до пристрою. Через нього передаються всі корисні дані: інтернет-трафік, відео, аудіо та інші типи завантаження. Завдяки великому обсягу інформації, що передається, у цьому каналі застосовується ефективно LDPC-кодування, яке забезпечує високу пропускну здатність.

PUSCH (Physical Uplink Shared Channel) — аналогічний до PDSCH, але працює у зворотному напрямку - тобто з пристрою до базової станції. Через нього передаються вихідні дані користувача, наприклад запити до серверів, голос, сенсорна інформація тощо. Як і в PDSCH, тут використовується LDPC-кодування для досягнення високої надійності та продуктивності передачі.

PDCCH (Physical Downlink Control Channel) відповідає за передачу службової (керуючої) інформації від базової станції до пристрою. Він повідомляє користувачеві, які ресурси виділені для передачі або прийому, коли активувати передачу та інші важливі сигнали керування. Обсяг інформації в цьому каналі невеликий, тому тут застосовується Polar-кодування, яке є оптимальним для коротких керуючих повідомлень.

PUCCH (Physical Uplink Control Channel) — канал керування у напрямку uplink. Через нього пристрій надсилає базовій станції сигнали зворотного зв'язку: підтвердження прийому (ACK/NACK), звіти про якість каналу (CQI), індикатори зайнятості тощо. Як і у PDCCH, Polar-коди дозволяють ефективно кодувати ці невеликі обсяги критично важливої інформації.

RACH (Random Access Channel) використовується на ранніх етапах зв'язку, коли пристрій ще не має призначених ресурсів і ініціює доступ до мережі. Це uplink-канал, через який здійснюється запит на з'єднання, наприклад при ввімкненні пристрою або поверненні у зону покриття. RACH також використовує Polar-кодування, оскільки обмінюється короткими початковими повідомленнями.

Усі ці канали тісно взаємодіють між собою для забезпечення ефективного та стабільного функціонування мережі. Наприклад, коли користувач вмикає телефон, він ініціює з'єднання через RACH. Після цього базова станція надсилає через PDCCH керуючі сигнали з інформацією про розподіл ресурсів. Далі користувач передає службову інформацію через PUCCH, а потім — корисні дані через PUSCH. У відповідь базова станція надсилає керування знову через PDCCH і користувацькі дані через PDSCH. Уся ця взаємодія відбувається циклічно і в реальному часі, забезпечуючи надійний і швидкий обмін інформацією.

Таким чином, структура фізичних каналів у 5G NR побудована так, щоб розділити керуючі та користувацькі функції, оптимізувати кодування відповідно до обсягу та типу даних і забезпечити ефективне використання радіоресурсів у складних умовах мобільного середовища.

Тобто, згідно до рис. 1.7 RACH — це канал, через який користувач ініціює підключення до мережі. Через нього надходить запит на ресурси. PDCCH — після запиту gNB надсилає рішення про призначення ресурсів користувачу, передає інструкції. PUSCH — користувач передає дані та/або сигнал зворотного зв'язку (наприклад, HARQ-ACK). PUCCH — супроводжує PUSCH або працює окремо для відправки контрольних повідомлень (CQI, ACK/NACK). PDSCH — базова станція пересилає користувацькі дані відповідно до призначених параметрів у PDCCH.

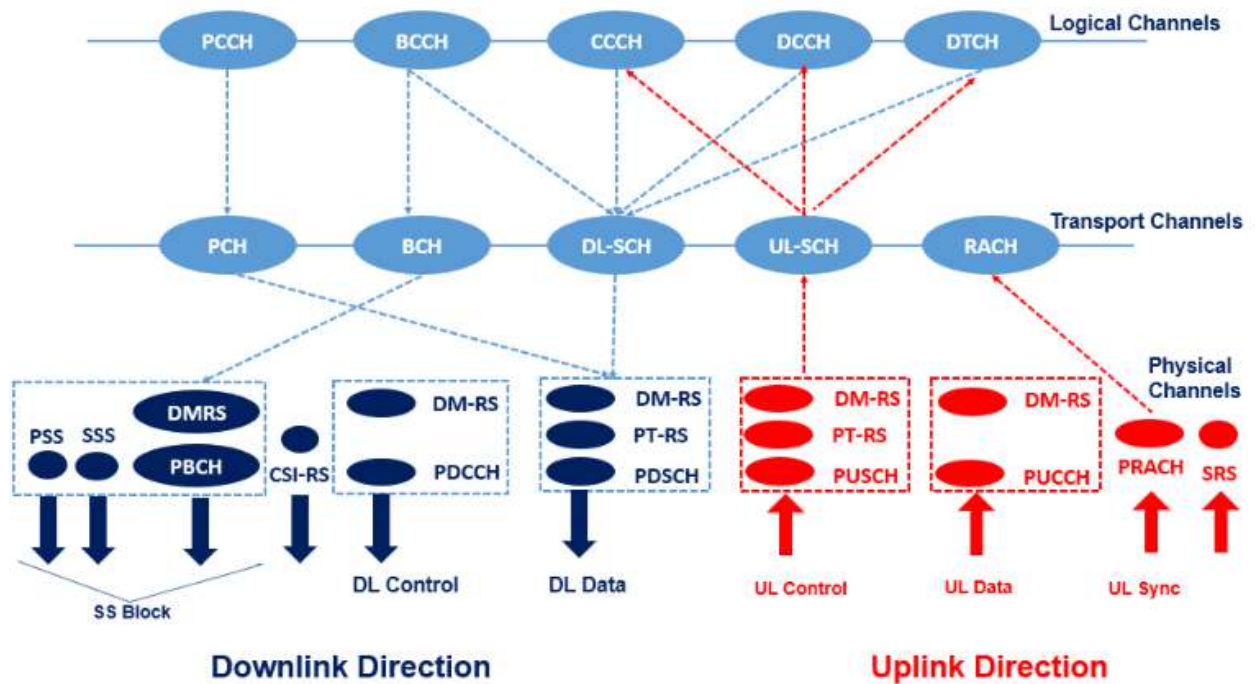


Рисунок 1.7 - Принципи взаємодії фізичних каналів у 5G NR

Можна побачити, з рис. 1.7, що канали розташовані в послідовності взаємодії, а саме в колі від ініціалізації (RACH), через управління (PDCCH), до даних та контролю в обох напрямках (PUSCH, PUCCH, PDSCH). Відповідно стрілки уздовж часової вісі (вліво — downlink, вправо — uplink) допомагають зрозуміти режим роботи кожного каналу. Чітко видно, що керуючі канали (PDCCH, PUCCH) і дані (PDSCH, PUSCH) взаємодіють у "діалозі", забезпечуючи надійну роботу сезону Hyper Reliable Low Latency (URLLC (Ultra-Reliable Low Latency Communications) - це комунікації з ультранизькою затримкою ( $\leq 1$  мс) та майже 100% надійністю (99.999%), призначені для критично важливих застосувань, де навіть найменша затримка чи втрата даних може мати серйозні наслідки) чи ширини eMBB (enhanced Mobile Broadband - це розширене мобільне широкопasmове підключення, яке забезпечує високу швидкість передачі даних, високу щільність трафіку та масове споживання мультимедіа-контенту). PDSCH (downlink) та PUSCH (uplink), які передають

основні обсяги даних (HD-відео, AR, ігри). LDPC-кодування забезпечує надійність та ефективність при великих обсягах.

Відносно контексту рис. 1.7 зазначимо, що у системі 5G NR канална архітектура побудована у вигляді ієрархічної моделі з трьох рівнів: логічного, транспортного та фізичного (дивитись рис. 1.7). Логічні канали визначають тип інформації, що передається, і класифікуються як керуючі (наприклад, BCCH, DCCH) або користувацькі (DTCH). Вони не пов'язані з конкретною реалізацією в радіоінтерфейсі, а лише описують, яку інформацію потрібно передати. Ці логічні канали передаються через транспортні канали, які визначають спосіб і структуру доставки даних — з урахуванням кодування, модуляції, сегментації та повторної передачі. До них належать, зокрема, DL-SCH (спільний канал downlink), UL-SCH (uplink), BCH, RACH тощо. Транспортні канали, у свою чергу, реалізуються за допомогою фізичних каналів, які здійснюють реальну передачу даних по радіоінтерфейсу. Наприклад, дані з логічного каналу DTCH спочатку проходять через транспортний канал DL-SCH або UL-SCH, а потім передаються через фізичний канал PDSCH (для downlink) або PUSCH (для uplink).

Таким чином, логічні, транспортні та фізичні канали у 5G NR утворюють послідовне коло взаємодії яке поширюється від типу інформації — до її фізичної реалізації, що забезпечує гнучкість, масштабованість і оптимізацію роботи мережі в умовах високої навантаженості.

## 1.5 Висновки до першого розділу

У першому розділі проведено аналіз особливостей каналу зв'язку в системах п'ятого покоління (5G) та сучасних методів завадостійкого кодування, що забезпечують надійність передачі в умовах складного радіосередовища. Розглянуто ключові типи каналів — AWGN та Rayleigh fading, а також їхню комбінацію, яка найбільш точно відображає реальні умови

у 5G. Зазначено, що ефективність декодування безпосередньо залежить від характеристик каналу, таких як SNR, BER, BLER, Latency та Outage Probability.

Системи 5G висувають жорсткі вимоги до методів кодування: наднизький рівень помилок ( $BLER < 10^{-5}$ ), мінімальна латентність ( $\leq 1$  мс), висока пропускна здатність ( $> 1$  Гбіт/с), гнучкість щодо розміру кодових слів і низьке енергоспоживання. Ці вимоги пояснюються сценаріями критичних застосувань — від автономного керування до масового підключення пристроїв (mMTC). В таблиці 1.1 узагальнено ключові вимоги до кодування в залежності від сервісу.

Основними стандартними методами завадостійкого кодування у 5G є LDPC-коди (для downlink-низхідний канал) і полярні коди (для uplink-висхідний канал). LDPC-коди мають переваги у швидкому і паралельному декодуванні, що важливо для передачі великих обсягів даних. Полярні коди краще підходять для коротких повідомлень завдяки низькій складності реалізації та здатності до досягнення межі Шеннона при обмеженій довжині.

У 5G NR реалізовано диференційовану архітектуру кодування, що враховує специфіку каналів downlink (низхідний) і uplink (висхідний), їхнє навантаження та обчислювальні ресурси пристроїв. Це забезпечує ефективне та адаптивне кодування в умовах змінного радіоканалу. В подальших розділах магістерської кваліфікаційної роботи буде детально проаналізовано алгоритми декодування LDPC та полярних кодів, а також можливості їх удосконалення за допомогою інтелектуальних методів.

## 2 МАТЕМАТИЧНИЙ ОПИС АЛГОРИТМІВ ДЕКОДУВАННЯ ЗАВАДОСТІЙКИХ КОДІВ У 5G

### 2.1 Ітеративні алгоритми декодування для LDPC

Ітеративні алгоритми декодування LDPC-кодів є ключовими в системах мобільного зв'язку 5G [32], де необхідна висока продуктивність декодування з мінімальною затримкою. У цьому підрозділі магістерської кваліфікаційної роботи розглядаються три основні варіанти ітеративного декодування: Belief Propagation (BP) (алгоритм поширення ймовірностей), Min-Sum (MS) (алгоритм мінімізації суми), а також Normalized/Offset Min-Sum (NMS/OMS) (нормалізований алгоритм мінімізації суми та мінімізації суми із зсувом) [33].

#### 2.1.1 Методика декодування LDPC-кодів за алгоритмом Belief Propagation (BP)

Алгоритм BP (також відомий як Sum-Product Algorithm (алгоритм сума-продукт)) є оптимальним для LDPC-кодів на графах без петель і базується на теорії Байєсівських мереж [34]. Алгоритм працює на графі Таннера і виконує обмін ймовірнісними повідомленнями між вузлами перевірки паритету (check nodes) та вузлами змінних (variable nodes). Ключова ідея BP-алгоритму полягає обчислює апостеріорної ймовірності бітів, передаючи повідомлення між вузлами графа. Повідомлення — це логарифмічні апостеріорні ймовірності (LLR, log-likelihood ratios).

До основних математичних індикаторів алгоритму віднесемо матрицю паритетних перевірок -  $\mathbf{H}$ ;  $y_i$  - отримане значення від каналу для бітів;

$LLR(y_i) = \log\left(\frac{P(x_i = 0|y_i)}{P(x_i = 1|y_i)}\right)$  - початкове повідомлення (яке представимо як

каналне LLR);  $m_{v \rightarrow c}$  - повідомлення від вузла змінної до перевірного вузла

(дивитись рисунок 1.4. розділа 1 – вид графу Таннера);  $m_{c \rightarrow v}$  - повідомлення від

перевірочного вузла до вузла змінної. На рис. 2.1 представлено граф Таннера для пояснення основних індикаторів алгоритму.

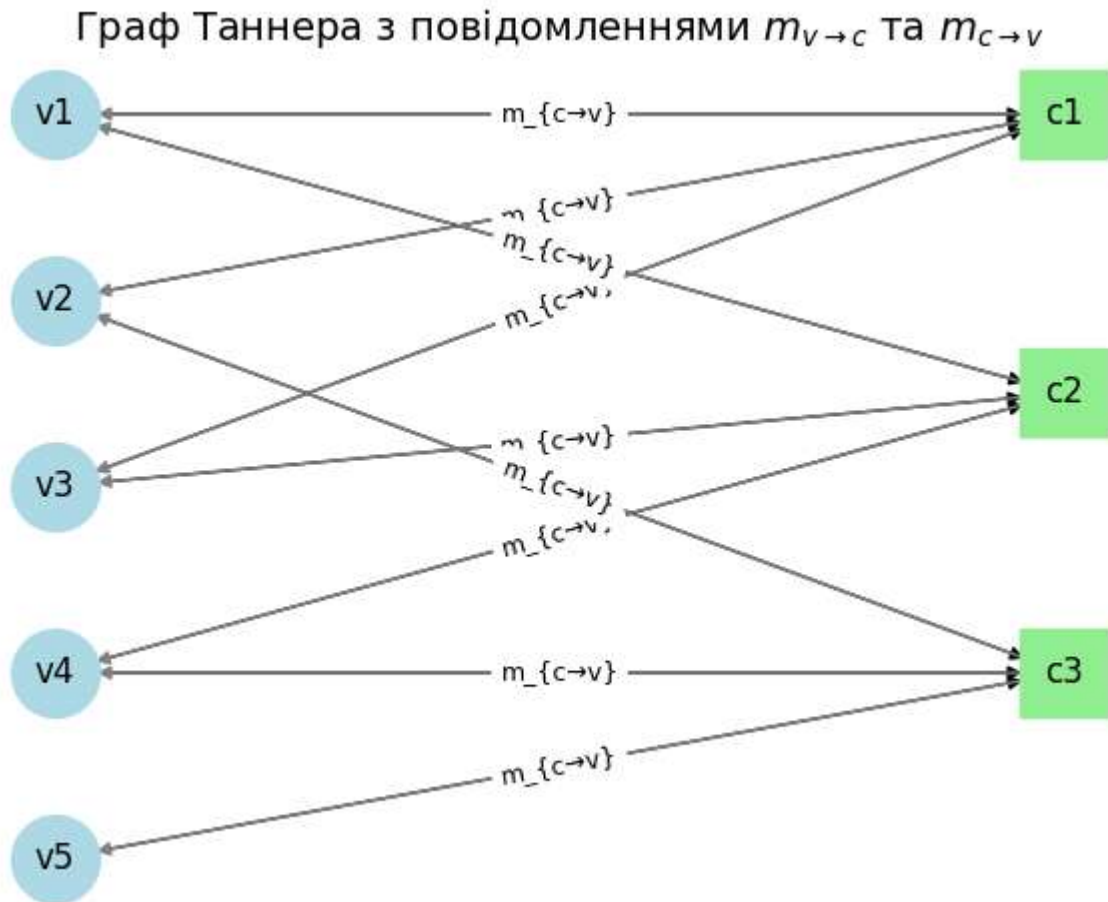


Рисунок 2.1 – Граф Таннера для LDPC: кола — вузли змінних ( $v_1, v_2, \dots$ ); квадрати — перевірочні вузли ( $c_1, c_2, \dots$ )

Алгорит декодування містить наступні кроки:

- 1) Ініціалізація, коли всі повідомлення  $m_{v \rightarrow c}$  встановлюються рівними  $\text{LLR}(y_i)$  — які визначають початкові значення з каналу;
- 2) Оновлення повідомлень (відбувається ітеративно):
  - для перевірочного вузла (check node)  $c$ :

$$m_{c \rightarrow v} = 2 \tanh^{-1} \left( \prod_{v' \in N(c) \setminus v} \tanh \left( \frac{m_{v' \rightarrow c}}{2} \right) \right) \quad (2.1)$$

де  $N_c$  - множина всіх вузлів змінних, які з'єднані з перевірочним вузлом  $c$ ;  $v$  - той вузол змінної, куди ми зараз надсилаємо повідомлення;  $v'$  - усі інші вузли

змінних, які підключені до  $c$ , окрім  $v$  (дивитись рис. 1.2 – тобто обчислюються повідомлення для  $v$ , враховуючи усі інші повідомлення  $m_{v' \rightarrow c}$  які надійшли в цей перевірючий вузол від інших змінних вузлів. Це ключова ідея локального оновлення повідомлень без самопосилань. Повідомлення в алгоритмі ВР не повинні враховувати самі себе, інакше виникне некоректна "петля".

- для змінного вузла (variable node)  $v$ :

$$m_{v \rightarrow c} = \text{LLR}(y_v) + \sum_{c' \in N(v) \setminus c} m_{c' \rightarrow v} \quad (2.2)$$

3) Обчислення апостеріорного LLR для кожного біту:

$$\text{LLR}(x_v) = \text{LLR}(y_v) + \sum_{c \in N(v)} m_{c \rightarrow v} \quad (2.3)$$

4) Прийняття рішення:

$$\hat{x}_v = \begin{cases} 0, & \text{якщо } \text{LLR}(x_v) > 0 \\ 1, & \text{інакше} \end{cases} \quad (2.4)$$

5) Перевірка паритету:

Якщо  $\mathbf{H} \cdot \hat{\mathbf{x}}^T = 0$ , то декодування завершено, інакше продовжити ітерації.

На рис. 2.2 представлено блок-схему одного кроку ітерації за алгоритмом ВР.

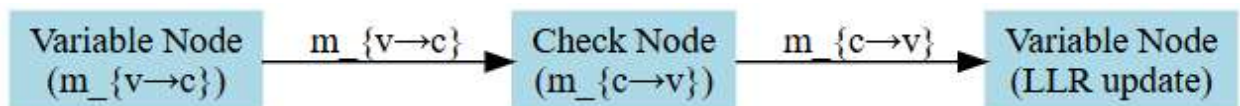


Рисунок 2.2 – Блок-схема одного кроку ітерації за алгоритмом ВР

З рис. 2.2 можна схематично побачити, що основний крок алгоритму ВР сформовано ітеративним обміном повідомленнями між вузлами змінних і перевірючими вузлами графа Таннера. Кожен такий цикл складається з двох етапів:

1) Variable  $\rightarrow$  Check - змінні вузли  $v$  передають повідомлення  $m_{v \rightarrow c}$  до перевірючих вузлів  $c$ , сформованих на основі початкових LLR та вхідних повідомлень від інших перевірючих вузлів;

2) Check  $\rightarrow$  Variable - перевірочні вузли обчислюють нові повідомлення  $m_{c \rightarrow v}$ , враховуючи паритетні обмеження, та надсилають їх назад до змінних вузлів;

3) LLR update — змінні вузли оновлюють свої апостеріорні оцінки LLR на основі вхідних повідомлень, після чого приймається рішення щодо значення кожного біта.

Такий двосторонній потік інформації реалізує механізм колективного "уточнення" знань про передане повідомлення в умовах шуму, що є основою ітеративного декодування в LDPC-кодах.

Таким чином резюмуючи потрібно акцентувати на тому, що алгоритм BP, також відомий як алгоритм суми-добутку (Sum-Product Algorithm), є одним із найефективніших ітеративних методів декодування для LDPC-кодів. Його суть полягає в обміні повідомленнями між вузлами графа Таннера (дивитись рис. 2.1), що моделює структуру коду на основі введення абстракції між змінними вузлами (variable nodes), які відповідають інформаційним та контрольним бітам, і перевірочними вузлами (check nodes), що реалізують паритетні обмеження коду. Алгоритм базується на теорії Байєсівських мереж [34] і має на меті обчислення апіорних імовірностей значень бітів коду на основі отриманих від каналу даних та внутрішніх зв'язків коду. Основний принцип дії алгоритму розкривається на основі того, що BP працює з логарифмічними співвідношеннями правдоподібності (log-likelihood ratios, LLR), які є логарифмами відношення ймовірностей бітів бути 0 або 1 за умови

отриманого сигналу ( $LLR(y_i) = \log\left(\frac{P(x_i = 0|y_i)}{P(x_i = 1|y_i)}\right)$ ). Тоді на основі цих значень

алгоритм ітеративно оновлює повідомлення, що передаються по ребрах графа (рис. 1.2) від змінних вузлів до перевірочних і навпаки. Повідомлення від вузлів змінної до перевірочних позначаються відповідно як  $m_{v \rightarrow c}$  та  $m_{c \rightarrow v}$ . Віхи виконання алгоритму містять етапи інфіціалізації ( $m_{v \rightarrow c} = LLR(y_v)$ ),

ітеративного оновлення повідомлень для змінних і перевірючих вузлів (2.1, 2.2), обчислення апостеріорного LLR кожного біта (2.3), прийняття рішення (2.4), та перевірки синдрому ( $\mathbf{H} \cdot \hat{\mathbf{x}}_v^T = 0$ ), в цьому випадку якщо умова виконується — декодування вважається успішним, інакше — алгоритм продовжує наступну ітерацію. Загалом ВР забезпечує найкращу можливу продуктивність серед ітеративних методів, особливо при відсутності петель у графі [33]. В експериментальній частині магістерської роботи буде показано, що LDPC у поєднанні з ВР підходять для сучасних систем зв'язку (зокрема 5G) з дуже низьким рівнем бітових помилок (BER). Серед базових проблем алгоритму слід акцентувати на високій обчислювальній складності, зумовлена необхідністю обчислення гіперболічних функцій (2.1), крім того йому притаманна чутливість до петель у графі (рис. 2.1). Тому на практиці графі LDPC мають короткі цикли (петлі), що призводить до кореляцій між повідомленнями, знижуючи точність ВР [35]. Наявність залежності від кількості ітерацій призводить до того, що для складних кодів може знадобитися багато циклів для досягнення збіжності.

### 2.1.2 Методика декодування LDPC-кодів за алгоритмом Min-Sum (MS)

Детальний опис алгоритму ВР проведений у підпункті 2.1 дозволяє стверджувати, що MS представляє собою спрощення ВР-алгоритму. Він замінює гіперболічні функції (дивитись формулу 2.1) мінімізацією, що значно зменшує обчислювальні витрати.

Представимо аналогічно до розглянутого алгоритму ВР кроки алгоритму мінімізації суми.

- 1) Передача повідомлення від перевірючого вузла.

$$m_{c \rightarrow v} = \left( \prod_{v' \in N(c) \setminus v} \text{sign}(m_{v' \rightarrow c}) \right) \cdot \min_{v' \in N(c) \setminus v} |m_{v' \rightarrow c}| \quad (2.5)$$

- 2) Передача повідомлення від вузла змінної (виконується аналогічно до ВР (2.2)).

$$m_{v \rightarrow c} = \text{LLR}(y_v) + \sum_{c' \in N(v) \setminus c} m_{c' \rightarrow v} \quad (2.6)$$

- 3) Розрахунок апостеріорного LLR.

$$\text{LLR}(x_v) = \text{LLR}(y_v) + \sum_{c \in N(v)} m_{c \rightarrow v} \quad (2.7)$$

Стосовно алгоритму MS потрібно акцентувати на тому, що цей алгоритм є спрощеним варіантом ВР та призначений для зменшення обчислювальної складності ітеративного декодування LDPC-кодів [35]. Як слідує з формул (2.5-2.7) він базується на тій самій архітектурі графа Таннера, що і ВР, але замість обчислення гіперболічних функцій ( $\tanh$ ,  $\tanh^{-1}$ ), які є дорогими для реалізації в апаратному забезпеченні, MS застосовує апроксимацію на основі знаків і мінімумів. Ключова ідея полягає в тому, що у MS-алгоритмі повідомлення від перевірючих вузлів обчислюються за правилом (2.5). Тобто бувально це означає, що знак повідомлення визначається як добуток знаків усіх вхідних повідомлень, а його значення — як мінімум їхніх модулів. Таким чином, MS наближається до ВР, але замінює складну функціональну операцію простим набором логічних і арифметичних дій. Тоді повідомлення від вузлів змінних до перевірючих, як і в ВР, мають вигляд (2.6), а апостеріорна оцінка LLR для кожного біта оновлюється згідно до (2.7). З практичної точки зору, як показано в роботі [29], це призводить до певного програшу перед ВР, однак низька обчислювальна складність виступає головною перевагою MS. В кінцевому рахунку можна отримати суттєве скорочення кількості математичних операцій, що робить його ідеальним для апаратної реалізації (ASIC, FPGA) [25]. Крім того, на MS задіяні менші апаратні ресурси, що не потребує реалізації складних функцій і дозволяє ефективно застосовувати його в мобільних пристроях або реального часу в базових станціях. Потрібно також сфокусувати на тому, що MS має високу швидкість обчислення через

спрощені обчислення, а отже алгоритм виконується значно швидше за ВР. Платою за такі спрощення є наявна груба апроксимація в наслідок заміни вже згаданих гіперболічних функцій на мінімальні значення і як результат втрати певної точності. З точки зору оціночних параметрів зниження продуктивності особливо помітне на низьких рівнях шуму (високих значеннях SNR), коли ВР може досягати продуктивності близько межі Шеннона, а MS очевидно ні [29]. Ну і слід наголосити на тому, що базовий MS не має механізму компенсації похибки апроксимації, а отже наявна неможливість адаптивного покращення.

В системах 5G, де LDPC-коди використовуються для каналного кодування в каналі даних (data channel) NR, MS є популярним вибором для практичної реалізації декодера (можно виділити роботи [33], [35]), особливо в апаратних модулях, де критично важливі затримка, споживання енергії та площа чипа. Він забезпечує компроміс між продуктивністю та складністю, що дозволяє досягати швидких декодувань при збереженні прийняттого рівня бітових помилок. Проте, для систем, де критичною є висока якість зв'язку при мінімальному BER, часто використовуються модифіковані версії MS, наприклад Normalized MS або Offset MS, які компенсують втрату точності.

### **2.1.3 Методика декодування LDPC-кодів за алгоритмами Normalized та Offset Min-Sum (NMS/OMS)**

Алгоритми NMS та OMS є модифікаціями класичного MS, які запроваджують компенсацію похибки апроксимації, що виникає при заміні гіперболічних функцій у ВР простим мінімумом у MS. Метою цих модифікацій є досягнення більшої точності при збереженні низької обчислювальної складності, що особливо актуально для реалізацій декодера в 5G, що є суттю магістерської дипломної роботи.

#### **2.1.3.1 Особливості алгоритма OMS**

OMS вводить зміщення (offset), яке віднімається від мінімального значення модуля повідомлення при передачі від перевірконого вузла і математично може бути представлено наступним виразом:

$$m_{c \rightarrow v} = \left( \prod_{v' \in N(c) \setminus v} \text{sign}(m_{v' \rightarrow c}) \right) \cdot \max \left( \min_{v' \in N(c) \setminus v} |m_{v' \rightarrow c}| - \beta, 0 \right) \quad (2.8)$$

де  $\beta$  - константа зміщення, яка підбирається емпірично або оптимізується; 0 – означає, що в випадку якщо результат  $<0$ , то він "обрізається" до 0.

Тобто у випадку NMS є сенс говорити, що алгоритм зменшує занадто "агресивне" зменшення значення повідомлень у MS, компенсуючи втрату точності. Мається на увазі, що класичному MS передане повідомлення часто переоцінює вплив найменш впевненого (найменшого за модулем) вхідного повідомлення, що призводить до надмірного послаблення інформації та втрати точності. В цьому контексті OMS частково виправляє цю проблему, віднімаючи сталу компенсацію  $\beta$  від знайденого мінімуму. Це дозволяє приглушити "зайвий песимізм" у переданих значеннях і наблизити результат до ВР, не ускладнюючи обчислення гіперболічними функціями. Таким чином, алгоритм досягає кращого балансу між простотою та точністю, що критично важливо для реалізацій у мобільних системах зв'язку.

### 2.1.3.2 Особливості алгоритма NMS

У NMS застосовується множник нормалізації  $\alpha \in (0,1]$  до обчисленого мінімуму, згідно до (2.8):

$$m_{c \rightarrow v} = \left( \prod_{v' \in N(c) \setminus v} \text{sign}(m_{v' \rightarrow c}) \right) \cdot \alpha \cdot \min_{v' \in N(c) \setminus v} |m_{v' \rightarrow c}| \quad (2.9)$$

де  $\alpha$  - зазвичай лежить у межах  $[0,65-0,85]$  (значення підбирається так, щоб мінімізувати BER без істотної втрати продуктивності).

Основний сенс введення NMS полягає у спробі пом'якшення повідомлень, що дозволяє краще наблизитися до результатів ВР при меншій складності.

Основні етапи алгоритму наступні:

1) Перевірочний вузол (check node) обчислює повідомлення для кожного підключеного до нього змінного вузла (variable node), аналогічно до OMS або NMS.

2) Виконується оновлення вузла змінної (Variable Node Update). На цьому етапі кожен вузол змінної  $v$  обчислює повідомлення для кожного підключеного до нього перевірконого вузла  $c$ . Це повідомлення відображає поточну оцінку ймовірності для біта, з урахуванням усієї наявної інформації, окрім тієї, яка надійшла від самого вузла  $c$  - тобто виконується як би "виняток джерела". Таким чином, кожен вузол змінної агрегує всю наявну інформацію з оточення, крім того вузла, до якого він зараз надсилає повідомлення. Це дозволяє підтримувати незалежність і уникати самопосилання в повідомленнях. Математично це виглядає так:

$$m_{v \rightarrow c} = \text{LLR}(y_v) + \sum_{c' \in N(v) \setminus c} m_{c' \rightarrow v} \quad (2.10)$$

3) Визначення апостеріорної ймовірності згідно до:

$$\text{LLR}(x_v) = \text{LLR}(y_v) + \sum_{c \in N(v)} m_{c \rightarrow v} \quad (2.11)$$

4) Етап прийняття рішення. Загалом аналогічно до ВР/MS, тобто  $\hat{x}_v = 0$ , якщо  $\text{LLR}(x_v) > 0$ , інакше 1.

Таким чином алгоритми NMS та OMS, за задумом і практикою [33], демонструють помітне покращення точності декодування у порівнянні з базовим MS, зберігаючи при цьому відносну простоту реалізації. Їх гнучкість полягає у можливості налаштування параметрів нормалізації  $\alpha$  та зміщення  $\beta$ , що дозволяє адаптувати декодер до різних типів каналів, умов шуму (SNR) і структур кодів. Разом із тим, ці модифікації мають і певні недоліки, а саме

потребу в ретельному підборі параметрів, які суттєво впливають на стабільність і ефективність декодування. Крім того, у порівнянні з MS, спостерігається незначне ускладнення апаратної реалізації, яке, однак, виправдане з огляду на отриманий приріст продуктивності.

Зауважимо також, що у практичних системах 5G, де використовуються LDPC-коди у фізичному рівні (особливо в даних downlink/uplink), саме OMS та NMS є найчастіше застосовуваними алгоритмами. Вони забезпечують оптимальний компроміс між точністю декодування та апаратною складністю, і тому активно реалізуються в комерційних модемах, базових станціях та мобільних пристроях [36].

Нижче представлений рисунок 2.3 на якому проведений аналіз розглянутих вище алгоритмів декодування LDPC за критеріями складності, точності та апаратної простоти реалізації.

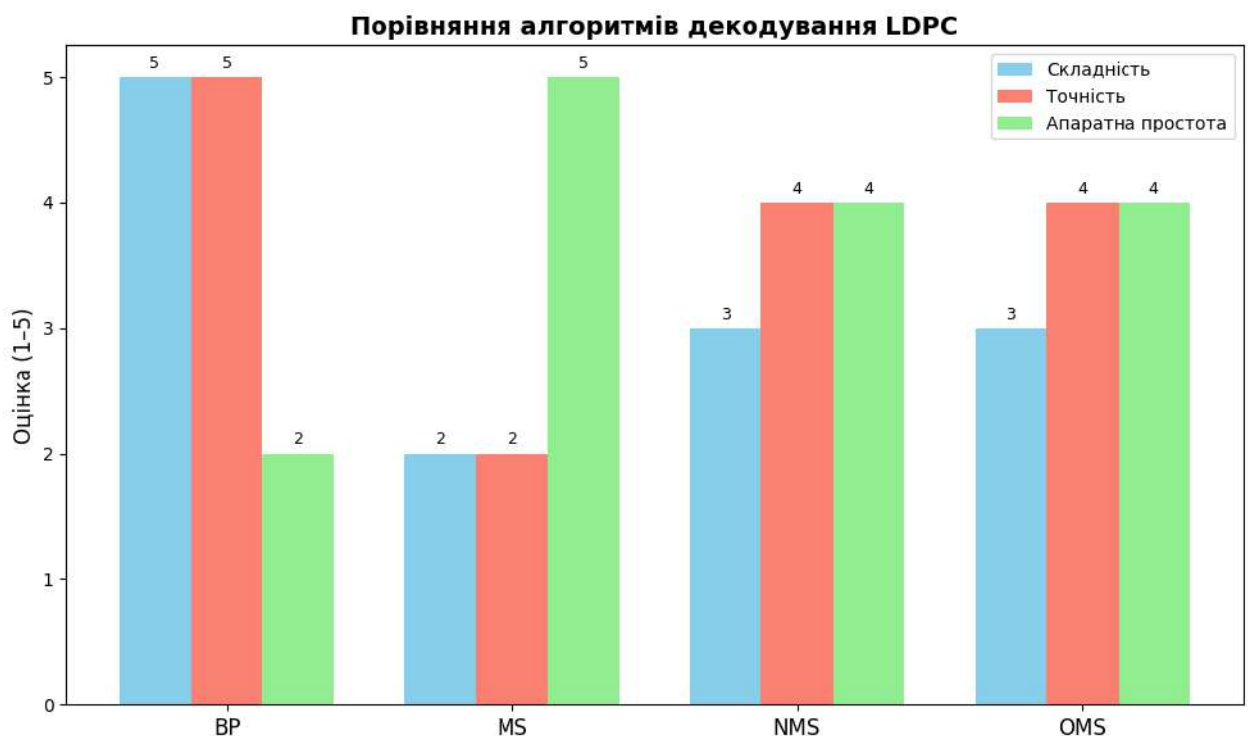


Рисунок 2.3 – Порівняння алгоритмів декодування LDPC

На графіку (рис. 2.3) зображено порівняння трьох ключових характеристик алгоритмів декодування LDPC за критеріями обчислювальної складності, точності декодування, та простоти апаратної реалізації. Кожна

характеристика оцінена за шкалою від 1 до 5, де 5 — найкраще значення в даному аспекті. Алгоритм BP демонструє найвищу точність (5 балів), що цілком відповідає його оптимальному баєсівському підходу до декодування. Однак це досягається ціною високої обчислювальної складності (5 балів) через використання нелінійних функцій  $\tanh$ ,  $\tanh^{-1}$ . Крім того, апаратна реалізація BP є складною (2 бали), що обмежує його ефективність у системах реального часу, таких як 5G. Далі, MS-алгоритм, навпаки, показує максимальну простоту реалізації (5 балів) та низьку обчислювальну складність (2 бали), завдяки заміні складних функцій простим оператором мінімуму. Проте точність декодування у нього є найнижчою (2 бали), що робить його менш придатним для каналів з високим рівнем шуму. Накінець NMS і OMS виступають як компромісні варіанти. Обидва алгоритми мають вищу точність (4 бали) порівняно з MS, зберігаючи помірну обчислювальну складність (3 бали). Завдяки введенню параметрів  $\alpha$  та  $\beta$ , ці алгоритми дозволяють більш гнучко адаптуватися до різних умов передачі. Апаратна складність у них трохи вища, ніж у MS (4 бали), але значно нижча, ніж у BP.

Зведені дані по алгоритмам декодування LDPC представлено у таблиці 2.1.

Таблиця 2.1 – Зведені дані для порівняння алгоритмів декодування LDPC

Критерій	BP	Min-Sum (MS)	Normalized MS (NMS)	Offset MS (OMS)
Точність	★★★★★ (найвища)	★★	★★★★	★★★★
Складність обчислень	Висока ( $\tanh$ , $\tanh^{-1}$ )	Низька ( $\min$ )	Помірна ( $\alpha \cdot \min$ )	Помірна ( $\min - \beta$ )
Гнучкість	Низька	Обмежена	Залежить від параметра $\alpha$	Залежить від параметра $\beta$
Стійкість до петель	Чутливий	Краща	Краща	Краща
Придатність для 5G	Теоретично можливо	Дуже добре підходить	Використовується	Використовується
Апаратна реалізація	Ускладнена	Проста	Трохи складніша	Трохи складніша

Підсумовуючі зазначимо, що з практичної точки зору, якщо мета застосування певного алгоритму декодування LDPC зводиться до умов забезпечення максимальної точності без обмежень на ресурси, варто використовувати ВР. Для низькопотужних пристроїв або систем із суворими обмеженнями на складність, доцільно застосовувати MS. Алгоритми NMS і OMS є золотою серединою — вони зберігають низьку складність і мають продуктивність, близьку до ВР, що робить їх оптимальним вибором для стандарту 5G.

## 2.2 Алгоритми декодування Polar-кодів

### 2.2.1 Методика декодування Polar-кодів за алгоритмом **Successive Cancellation (SC)**

Алгоритм послідовного сканування (SC) [37] є базовим методом декодування полярних кодів, запропонованим Аріканом. Він базується на послідовному прийнятті рішень для кожного біта інформації, використовуючи рекурсивну структуру графа полярного коду.

Математичний опис предствимо з принципів формування довжини коду. Прийmemo що довжина коду визначається виразом  $N = 2^n$  (тут  $n$  - ступінь двійки, що визначає довжину полярного коду). Декодування здійснюється шляхом обчислення логарифмічного відношення правдоподібності (LLR) за наступним виразом:

$$L(u_i) = \log \frac{P(y, \hat{u}_0^{i-1} | u_i = 0)}{P(y, \hat{u}_0^{i-1} | u_i = 1)} \quad (2.12)$$

де  $P(y, \hat{u}_0^{i-1} | u_i = 0)$  - умовна ймовірність отримати вихід каналу  $y$ , знаючі попередні декодовані біти  $\hat{u}_0^{i-1}$ , за умови, що поточний біт  $u_i$  дорівнює 0 або 1; LLR — це логарифм відношення ймовірностей того, що біт  $u_i$  дорівнює 0 або 1, за умов спостережень. Він дозволяє прийняти рішення за умовами, якщо

$L(u_i) > 0$  - більш ймовірно, що  $u_i = 0$ , в протилежному випадку при  $L(u_i) < 0$  - більш ймовірно, що  $u_i = 1$ .

LLR обчислюються рекурсивно за допомогою наступних функцій:

$$f(L_1, L_2) = \text{sign}(L_1) \cdot \text{sign}(L_2) \cdot \min(|L_1|, |L_2|) \text{ ("f" function)} \quad (2.13)$$

$$g(L_1, L_2, u) = L_2 + (1 - 2u) \cdot L_1 \text{ ("g" function)} \quad (2.14)$$

де рекурсивне обчислення LLR у полярних кодах базується на графовій структурі метеликоподібної форми. На кожному рівні такого графа застосовуються дві ключові функції, а саме:  $f(L_1, L_2)$  - використовується для обчислення LLR верхньої гілки графа (при переході з попереднього рівня до поточного). Вона ґрунтується на принципі комбінування двох незалежних ймовірностей і виражається через знак і мінімум абсолютних значень згідно (2.13). Ця функція реалізує лог-доменну апроксимацію мінімізації сумарної невизначеності. Також, функція  $g(L_1, L_2, u)$  - використовується для нижньої гілки і враховує вже прийняте значення  $u$  на попередньому етапі, відповідно до формули (2.14). Відповідно у формулі (2.14) фактор  $(1 - 2u)$  - змінює знак  $L_1$  залежно від попереднього рішення  $u \in \{0, 1\}$ , коригуючи нову LLR.

На рис. 2.4 представлено метелик одного рівня для обчислення LLR у відповідності до формул (2.13) та (2.14) для полярних кодів.

Метелик одного рівня для обчислення LLR у полярних кодах

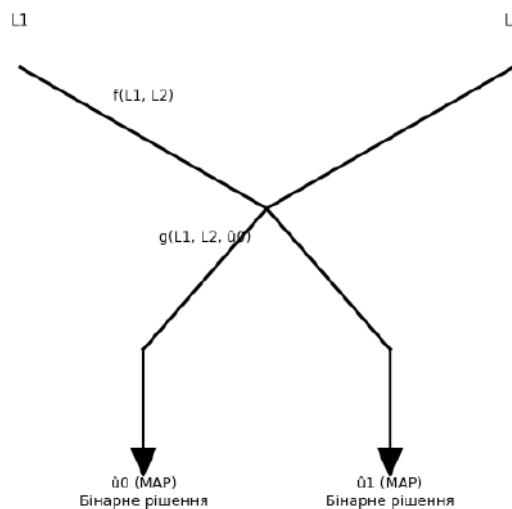


Рисунок 2.4 – Метелик одного рівня для обчислення LLR у полярних кодах

Після обчислення LLR для кожного біта, виконується рішення наступної форми:

$$\hat{u}_i = \begin{cases} 0, & \text{якщо } L(u_i) \geq 0 \\ 1, & \text{інакше} \end{cases} \quad (2.15)$$

Тобто (2.15) відповідає рішенням за критерієм максимальної апостеріорної ймовірності (MAP decision rule), тобто вибір того значення, яке має більшу ймовірність згідно з даними, отриманими з каналу.

На рис. 2.5 проілюстровано метеликоподібну структуру графу полярного коду для  $N=4$ .

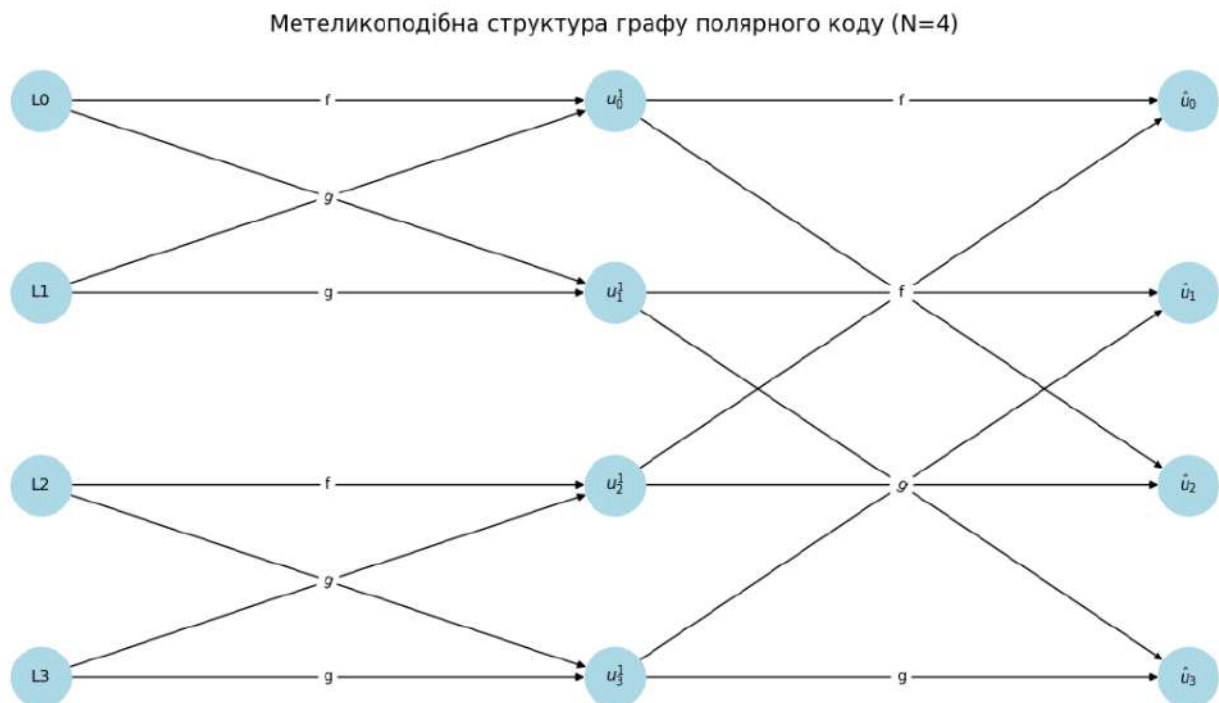


Рисунок 2.5 – Метеликоподібна структура графу полярного коду для  $N=4$

Рисунок 2.5 показує, зокрема, що метеликоподібний граф для полярного коду з  $N=4$  складається з трьох рівнів (стейджів), що відповідає  $\log_2 N + 1$ . На кожному рівні розміщено по чотири вузли. Перший рівень містить вхідні значення LLR з каналу і позначається як  $L_0, L_1, L_2, L_3$ . Останній рівень представляє вихід декодера це оцінені біти  $\hat{u}_0, \hat{u}_1, \hat{u}_2, \hat{u}_3$ . Середній рівень містить проміжні вузли, що позначаються як  $u_i^s$ , де  $i$  – номер біта, а  $s$  – номер рівня. Між рівнями реалізовані з'єднання у формі «метеликів», де кожен метелик

включає чотири ребра між парою вхідних та вихідних вузлів. Ці ребра відповідають двом ключовим функціям полярного, а саме  $f$  та  $g$  згідно до (2.13) та (2.14). Зокрема  $f$  використовується для верхньої гілки і комбінує два незалежні LLR значення, тоді як  $g$  - застосовується для нижньої гілки та враховує вже прийняте попереднє рішення  $u$ . Ребра між вузлами підписані відповідно як " $f$ " або " $g$ ", що вказує, яка функція використовується для обчислення значення на наступному рівні. Таким чином, граф показує, як у процесі декодування інформація проходить крізь каскад функцій  $f$  та  $g$ , поки не буде сформовано повну послідовність оцінених бітів  $\hat{u}_i$ . Це дозволяє реалізувати ефективний декодер за алгоритмом SC, який базується на структурі з фіксованою топологією. Граф чітко ілюструє логіку обчислень у полярному декодуванні та є основою для побудови як апаратної реалізації декодера, так і для оптимізованого програмного SC-декодування.

На рис. 2.6 представлено блок-схему SC декодера.

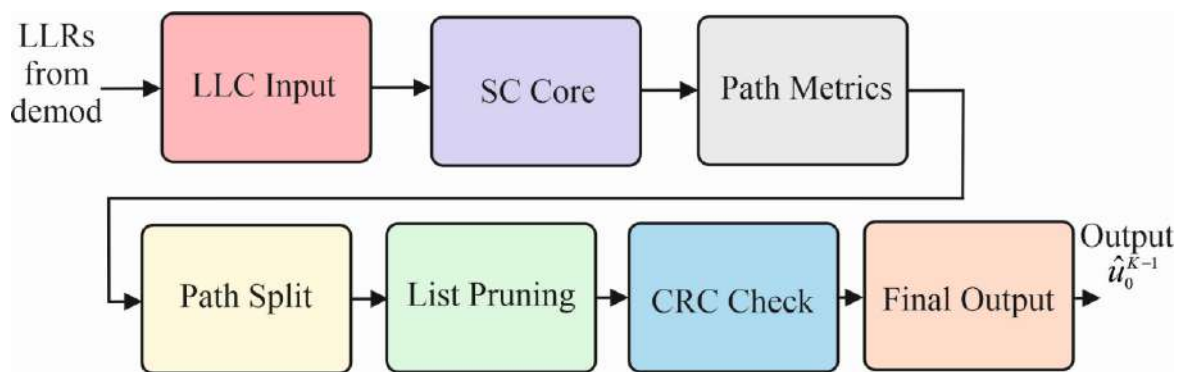


Рисунок 2.6 – Блок-схема SC – декодера: LLC Input – блок надходження логарифмічних відношень правдоподібності; SC Core – блок базового декодування у стилі SC; Path Metrics – блок обчислення метрик шляхів; Path Split – розгалуження кожного шляху на 0/1; List Pruning – відбір  $L$  найкращих шляхів; CRC Check – блок перевірки CRC у CA-SCL; Final Output – блок вибору найкращого результату; LLRs from demod - логарифмічні відношення правдоподібності (LLR), які формуються після демодуляції сигналу, отриманого з каналу (AWGN або Rayleigh)

Зазначені на рис. 2.6 метрики є числовими оцінками впевненості в тому, що переданий біт дорівнює 0 або 1;  $\hat{u}_0^{K-1}$  - залишкова декодована інформаційна послідовність, яка відповідає найнадійнішому шляху (згідно з метрикою шляху, Path Metric) і, у випадку CA-SCL, задовольняє контрольну CRC-перевірку. Практично це бітовий набір, де  $K$  - кількість інформаційних бітів (без урахування CRC) і який передається далі на верхній рівень протоколу (наприклад, до фізичного чи мережевого шару в 5G (дивитись рис. 1.7 розділу 1 кваліфікаційної магістерської роботи)).

SC має низку характеристик, які визначають його доцільність у конкретних умовах, серед яких доцільно виділити відносну простоту реалізації. Алгоритм SC має лінійну складність, що робить його привабливим для реалізації в апаратних рішеннях з мінімальними витратами. Це дозволяє ефективно використовувати його в умовах обмежених ресурсів. Крім того, можна вказати на низьке енергоспоживання. Завдяки послідовній природі виконання та відсутності складних математичних операцій, SC-декодування характеризується помірним рівнем енергоспоживання. Це робить його придатним для енергообмежених систем, таких як IoT-пристрої. Звону ж таки, потрібно згадати за низьку продуктивність при малих довжинах кодів. При коротких кодах SC демонструє високу ймовірність бітових помилок, що обмежує його ефективність у таких сценаріях. Це є серйозним недоліком для застосувань, де потрібна висока точність передавання даних. Крім того, потрібно вказати і на наявну високу затримку. Оскільки декодування здійснюється послідовно — кожен біт залежить від рішення щодо попередніх — це призводить до суттєвої затримки (латентності). Такий підхід ускладнює застосування SC у реального часу або в системах з високими вимогами до швидкодії. Резюмуючі зауважимо, що SC-декодування може ефективно використовуватись у системах із жорсткими обмеженнями за енергоспоживанням. Проте його низька продуктивність і висока затримка не

дозволяють розглядати його як придатне рішення для більшості сучасних телекомунікаційних стандартів, зокрема 5G та 6G.

### 2.2.2 Методика декодування Polar-кодів за алгоритмом **Successive Cancellation List (SCL)**

Алгоритм послідовного скасування з підтримкою списку (SCL-декодування) розширює розглянутий вище в кваліфікаційній роботі SC-алгоритм, зберігаючи одночасно  $L$  найкращих шляхів декодування.

Порівнюючі цей алгоритм з методикою описаною у розділі 2.2.1 потрібно виділити, що при кожному рішенні  $u_i$  розгалужується на два варіанти (0 і 1), утворюючи нові кандидати. Після цього вибираються  $L$  найвірогідніших використовуючі метрику шляхів яка математично може бути представлена наступним виразом:

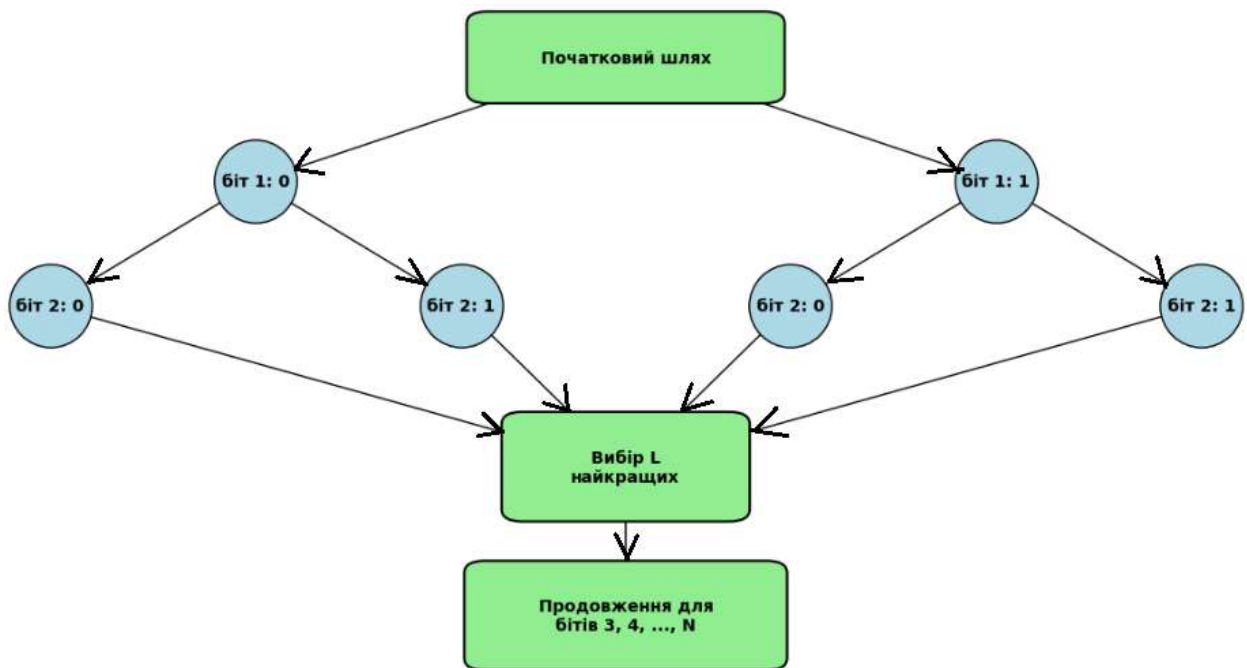
$$PM = \sum_{i=0}^{N-1} \log(1 + e^{-L(u_i)}) - u_i \cdot L(u_i) \quad (2.16)$$

Таким чином для алгоритму послідовного скасування з підтримкою списку отримуємо наступну картину. На відміну від SC, який обирає єдиний шлях декодування, SCL зберігає кілька можливих варіантів декодування одночасно — так званий список із  $L$  найкращих гіпотез, де під  $L$  розуміють фіксовану довжину списку. Практично це реалізується наступним чином. На кожному кроці прийняття рішення (для кожного інформаційного біта) алгоритм розгалужується на два варіанти — "0" та "1". Таким чином, кількість можливих шляхів подвоюється. Потім обираються  $L$  найкращих шляхів на основі спеціальної метрики правдоподібності (path metric) яку представлено формулою (2.16). Тобто після розгалудження із  $2L$  отриманих шляхів відбираються  $L$  найвірогідніших. Процес повторюється для наступного біта. Загалом можна констатувати, що такий підхід значно зменшує ймовірність помилкового рішення, оскільки дозволяє "повернутися" до альтернативного шляху, якщо початкове рішення було неправильним. В експериментальному

розділі магістерської роботи буде показано, що SCL забезпечує значне покращення BLER при малій довжині коду, що робить його придатним для управлінських каналів у 5G.

Для наочної візуалізації на фрагменті графа (метелика) рис. 2.7 представлено схематично процес розгалуження який описний вище при SCL – декодуванні.

**Фрагмент графу (метелик) для SCL декодування**



→ Кількість шляхів подвоюється при кожному бітові (експоненційне зростання)

Рисунок 2.7 – Фрагмент графу (метелик) для SCL декодування

У практичних системах, зокрема в 5G які досліджуються в магістерській роботі, до полярного коду додають CRC [25]. Після декодування всіх  $L$  кандидатів перевіряється CRC, і серед тих, що її проходять, вибирається найкращий. Це дає майже максимально можливу продуктивність, наближену до ML-декодера (maximum likelihood=максимальної правдоподібності). Це і зумовило, що у 5G стандарті (3GPP NR [1]) використовується SCL-декодування з CRC саме для управлінських каналів (control channels).

Щоб розмежувати можливості двох алгоритмів SCL та SC нижче представлено порівняльну таблицю 2.2 з перевагами одного алгоритму над іншим, звісно зарахунок збільшення складності і енергоспоживання.

Таблиця 2.2 – Систематизовані переваги алгоритму SCL над SC

Критерій	SC	SCL
Кількість шляхів	1	До $L$ паралельних шляхів
Надійність	Середня	Висока, особливо з CRC
Помилки декодування	Частіше	Значно рідше
Обчислювальна складність	Низька	Вища, але керована

### 2.2.3 Методика декодування Polar-кодів за алгоритмом CRC-aided SCL (CA-SCL)

CA-SCL — модифікація SCL (тобто SCL з перевіркою CRC)), де CRC-контроль використовується для вибору серед декількох кандидатів. Основна ідея полягає в тому, що на етапі кодування до інформаційної частини додається  $r$ -бітова CRC. Після завершення декодування перевіряються всі  $L$  шляхів. Серед тих, що задовольняють CRC, вибирається найкращий за метрикою PM (дивитись формулу 2.16).

CRC, як вказувались вище в тексті пояснювальної записки, це циклічний надлишковий код у формі контрольної суми, яка додається до інформаційного слова перед полярним кодуванням. Вона дозволяє покращити надійність декодування — особливо в SCL-декодері, де CRC використовується як критерій перевірки правильності кандидата. На рис. 2.8 представлена блок-схема формування полярного коду з CRC.

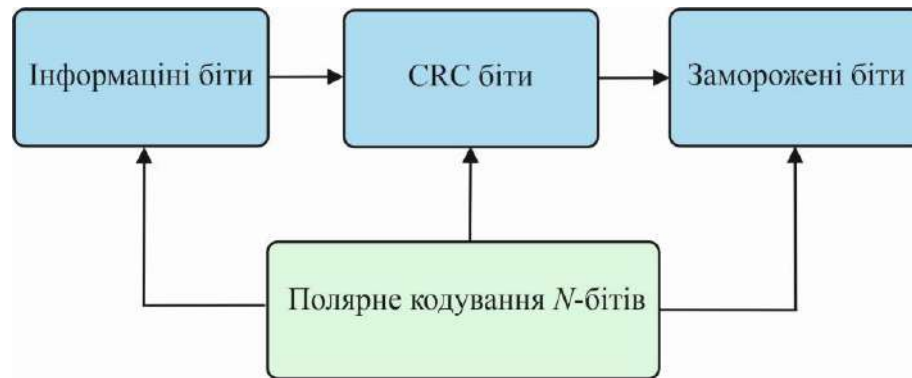


Рисунок 2.8 – Схема формування полярного коду з CRC

У вигляді розмітки позицій:

$$u = [u_0, u_1, \dots, u_{K-1}, c_0, \dots, c_{L-1}, 0, \dots, 0] \quad (2.17)$$

де  $K$  – кількість інформаційних бітів;  $L$  – довжина CRC (звичайно 8, 11, 16 біт);  $c_i$  – CRC біти; 0 – заморожені біти для коду довжиною  $N$ .

CRC визначається генераторним поліномом який має вид:

$$g(x) = x^8 + x^7 + x^2 + x + 1 \quad (2.18)$$

Цей поліном визначає, як саме обчислюються CRC-біти з інформаційного слова. Найпоширеніші поліноми мають наступну структуру - CRC-8:  $x^8 + x^2 + x + 1$ ; CRC-11:  $x^{11} + x^9 + x^8 + x^7 + x^2 + 1$ ; CRC-16:  $x^{16} + x^{12} + x^5 + 1$ .

Принцип застосування CRC наступний, до  $K$  інформаційних бітів додається  $L$  CRC-бітів (обчислених через ділення на  $g(x)$ ). Отримане слово довжини  $K+L$  передається до полярного кодера (решта бітів — нулі). Тоді на приймальному боці, декодер генерує список  $L$  найкращих шляхів (як пояснено вище в тексті пояснювальної записки) і CRC перевіряється для кожного кандидата. В результаті обирається перший з правильним CRC, що в підсумку значно знижує помилку декодування.

Отже, алгоритм SCL з перевіркою CRC є основним методом декодування полярних кодів у стандарті 5G NR для коротких кодових слів. Він забезпечує надійність та низьку затримку при відносно помірній складності, що обумовлює його широке застосування для у 5G NR для PDCCH (фізичного низьхідного каналу зв'язку).

## 2.2.4 Методика декодування Polar-кодів за алгоритмом Fast SC/SC Flip

Алгоритми швидкого послідовне скасування (Fast SC)/послідовного скасування з переобчисленням (SC Flip) використовують попередні обчислення та уніфіковані блоки обробки (Fast-SSC), що дозволяє виконувати декодування частинами, прискорюючи SC. Тобто, SC Flip декодує повідомлення як SC, але в разі невдачі (розглядаємо випадок коли CRC не пройдено) переобчислює рішення, змінюючи значення в найменш надійних бітових позиціях (дивитись рис. 2.7). Такий алгоритм підвищує продуктивність SC при збереженні невисокої складності, і може застосовуватись у системах 5G IoT або low-latency сценаріях [38]. У перспективі може бути інтегрованим в 6G-системи із низьким ресурсним бюджетом.

На рис. 2.9 зображено фрагмент дерева декодування полярного коду довжини  $N=8$ , сформованого за принципами Fast-SSC.

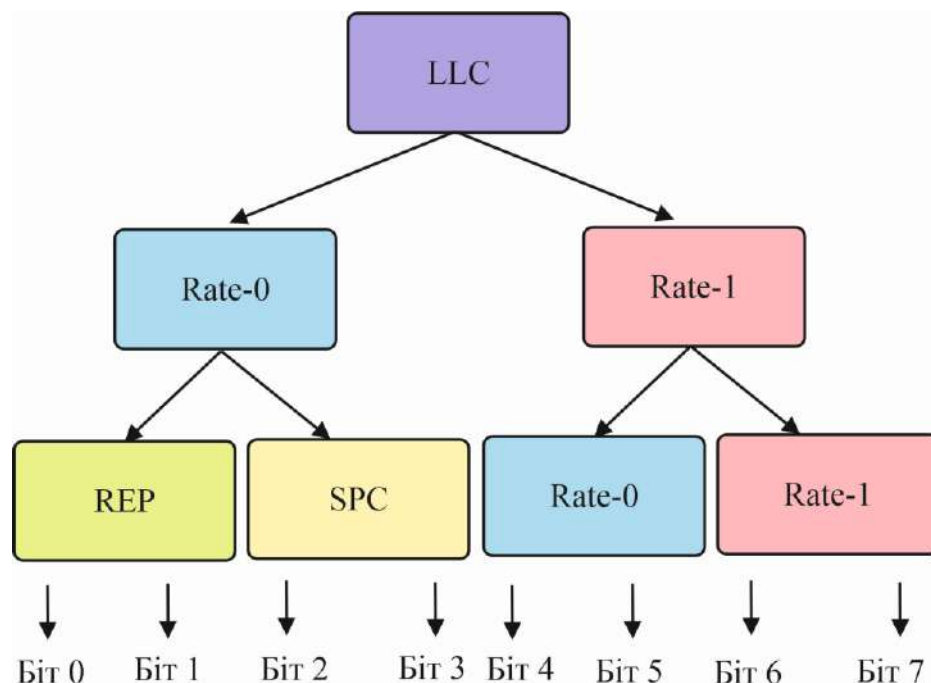


Рисунок 2.9 - Фрагмент дерева декодування полярного коду довжини  $N=8$ , сформованого за принципами Fast-SSC

Цей підхід використовує структуру дерева, де кожен вузол представляє певну обробку вхідної інформації, зокрема LLR, і дозволяє ідентифікувати типові шаблони для прискорення декодування.

У верхній частині схеми показано вхідний вузол "LLR", що отримує вектор з 8 LLR-значень. Далі дерево розгалужується на підвузли, які класифіковано за їх функціональною природою. Зокрема, тут візуалізовано вузли типу Rate-0 (усі біти — заморожені, результат відомий), Rate-1 (усі інформаційні біти, рішення приймаються без фільтрації), REP (повторюваний блок, де всі біти однакові), та SPC (Single Parity Check, коли усі біти відомі, крім одного, і застосовується перевірка парності). Кожен тип вузла позначено окремим кольором, що спрощує візуальне сприйняття структури.

Стрілки, які з'єднують вузли, вказують напрям обробки інформації — від загального до часткового. На нижньому рівні дерева показано вихідні біти (від біт 0 до біт 7), які формуються як результат об'єднання локальних рішень із кожної гілки дерева. Така схема наочно демонструє, як Fast-SSC дозволяє уникати рекурсивного обходу кожного біта окремо, замінюючи його на блокову обробку, що значно зменшує затрати часу та обчислювальні ресурси при декодуванні.

Як вже вище зазначалось, алгоритми Fast SC та SC Flip знайшли застосування у пристроях з обмеженими обчислювальними ресурсами, таких як IoT-модулі (Internet of Things) [38], сенсори у smart-city [39] мережах та пристрої з енергоощадною архітектурою. Завдяки своїй низькій складності та зменшеній латентності, Fast SC підходить для uplink-комунікацій, де важливо забезпечити швидке передавання службової інформації з невеликою кількістю біт.

SC Flip, у свою чергу, ефективний в умовах, коли якість каналу варіюється, і є ймовірність помилки при першій спробі декодування. Завдяки здатності до переобчислення, цей метод знижує ймовірність втрати пакета, що

критично для місійно важливих застосувань — наприклад, автомобільних мереж, дронів або промислового зв'язку.

Обидва алгоритми розглядаються як перспективні варіанти для 6G-комунікацій [40] з низькою затримкою (URLLC), де необхідна гнучкість між складністю реалізації та якістю декодування.

### 2.3 Порівняння продуктивності (BER/BLER, складність, латентність)

Оцінка ефективності декодування завадостійких кодів у системах 5G базується на трьох ключових метриках які включають бітову ймовірність помилки (BER), ймовірність блочної помилки (BLER), а також обчислювальну складність та латентність (затримку). Вони дають змогу адекватно характеризувати якість прийому, швидкодію та придатність алгоритмів до конкретних сценаріїв у мобільних мережах нового покоління.

Відносно самих показників (індикаторів) апродуктивності зазначимо наступне. Бітова ймовірність помилки (BER, Bit Error Rate) — це ймовірність того, що прийнятий біт відрізняється від переданого. Аналітично представимо цей індикатор наступним виразом:

$$\text{BER} = \frac{N_{err}}{N_{tot}} \quad (2.19)$$

де  $N_{err}$  — кількість помилково прийнятих бітів,  $N_{tot}$  — загальна кількість переданих бітів.

Ймовірність блочної помилки (BLER, Block Error Rate) — це ймовірність того, що весь кодовий блок буде прийнятий з хоча б однією помилкою, що аналітично представимо наступним чином:

$$\text{BLER} = \frac{N_{block\_err}}{N_{block\_tot}} \quad (2.20)$$

де  $N_{block\_err}$  — кількість блоків із помилками, а  $N_{block\_tot}$  — загальна кількість переданих блоків.

Потрібно наголосити, що згідно стандарту [1] BLER є офіційною метрикою у 5G NR для оцінки якості службових каналів (PDCCH, PUCCH, тощо), оскільки помилка в одному блоці може спричинити втрату пакета управління або критичного повідомлення. Тому вимоги 3GPP до BLER вимагають забезпечення рівня  $BLER < 10E-5$  з метою забезпечення цілісності переданих даних при прийнятному рівні завад.

Відмовідно другий індикатор є латентність у контексті декодування — це час, необхідний для прийому, обробки і видачі результату декодування. Для 5G критично важливо мати низьку латентність (у сценаріях URLLC —  $< 1$  мс) [41]. Серед основних факторів, що впливають на латентність можна виділити природу алгоритму (послідовна або паралельна обробка); глибину списку (у SCL-декодуванні (розглянуто в попередньому підпункті магістерської роботи); складність CRC-аналізу (розглянуто в попередньому підпункті магістерської роботи); архітектурна реалізація (FPGA/ASIC/GPU) – дивитись роботу [13].

Наступний індикатор це обчислювальна складність. Загалом вона оцінюється як кількість арифметичних операцій (додавань, порівнянь, логарифмів) на біт, або як асимптотична складність алгоритму. Нижче в систематизовано вигляді (табл.2.3) представлено деякі відомості щодо обчислювальної складності LDPC та полярних кодів які досліджуються в кваліфікаційній роботі.

Основна специфіка застосування розглянутих метрик (індикаторів) до LDPC і Polar-кодів наступна. Як вже зазначалось в тексті пояснювальної записки з кваліфікаційної дипломної роботи, LDPC-коди зазвичай застосовуються для високошвидкісних PDSCH (канал призначення в 5G), де необхідна висока пропускна здатність. Завдяки їх паралельній структурі декодування (BP або MS – розглянуті у розділі 2.1), LDPC мають низьку

латентність і підходять для висхідного/низхідного каналів передачі великих даних. Тому, власне BER є основною метрикою їх ефективності.

Таблиця 2.3 – Відомості щодо обчислювальної складності LDPC та полярних кодів

Алгоритм	Складність (асимпт.)
LDPC BP	$O(n \cdot d)^*$ , де $d$ — середня ступінь графа Таннера
Polar SC	$O(N \log N)^*$
Polar SCL ( $L=4$ )	$O(L \cdot N \log N)^*$

\* $O$  - це позначення порядку зростання складності алгоритму (стандартна нотація, яка описує межу складності зверху (upper bound), тобто максимальну швидкість зростання витрат ресурсу (часу або пам'яті)) який залежить від добутку двох параметрів  $n \cdot d$  тобто кількості вузлів і середнього ступеню вузлів у графі Таннера, тобто середньої кількості з'єднань (ребер) на один вузол, або загальна кількість операцій в алгоритмі (наприклад, проходів у графі, оновлень повідомлень) пропорційна кількості вузлів помноженій на середню кількість їх з'єднань;  $O(N \log N)$  - це асимптотична оцінка складності, яка говорить, що алгоритм обробляє  $N$  елементів, і на кожному рівні обробки виконує близько  $\log N$  дій (через поділ, дерево або агрегацію) (це ефективніше, ніж квадратичні алгоритми, і часто вважається «швидким», особливо для великих обсягів даних

Навпаки, Polar-коди застосовуються у коротких блоках (наприклад, для PDCCH), де важливішим є BLER, а не BER. Через структуру алгоритму декодування SC вони мають більшу затримку, але можуть забезпечити дуже низьку ймовірність блочної помилки завдяки застосуванню розглянутих у пункті 2.2.3 і представлених алгоритмів CA-SCL або Fast-SC.

Загалом, і це буде представлено у 4-мому розділі кваліфікаційної дипломної роботи. Порівняння продуктивності кодування буде виконуватись за наступними сценаріями: штучне моделювання AWGN/Rayleigh каналу; вимірювання BLER і BER для однакових умов при різних алгоритмах; побудова кривих BLER від SNR (дБ) для фіксованої довжини коду та типу модуляції; аналіз латентності та використання ресурсів (кількість ітерацій, кількість шляхів у SCL).

Отже, показники BER/BLER, латентність та складність дозволяють провести всебічну оцінку алгоритмів декодування у контексті стандарту 5G. Polar-коди краще підходять для коротких повідомлень і критичних каналів, тоді як LDPC забезпечують високу продуктивність для великих обсягів даних і мають хорошу масштабованість у паралельних обчисленнях. Методика порівняння має враховувати характер каналу, доступні ресурси і профіль трафіку, що забезпечує обґрунтований вибір кодування для конкретного підсценарію мережі нового покоління.

## 2.4 Висновки до другого розділу

У другому розділі дипломної роботи було здійснено всебічний аналіз основних методів декодування двох типів сучасних завадостійких кодів — LDPC та Polar-кодів, які є ключовими для побудови систем зв'язку п'ятого покоління (5G). Розгляд алгоритмів декодування дозволив оцінити їх ефективність, складність реалізації та доцільність використання в конкретних елементах телекомунікаційної мережі.

Для LDPC-кодів найвищу точність забезпечує алгоритм BP, однак його висока обчислювальна складність суттєво ускладнює апаратну реалізацію, що не завжди виправдано в умовах реального часу. Натомість спрощені алгоритми, зокрема MS та його модифікації (NMS та OMS), демонструють прийнятну продуктивність при суттєвому зниженні складності. Вони добре підходять для реалізації в апаратних засобах і широко застосовуються в 5G NR для передачі користувацьких даних. Важливою перевагою LDPC-кодів є їхня висока паралельність, що сприяє зниженню латентності та забезпечує хорошу масштабованість.

Щодо Polar-кодів, базовий метод декодування SC є простим і енергоефективним, однак поступається за показниками помилок при коротких кодах. Набагато кращу продуктивність демонструє SCL, особливо у поєднанні

з CRC-перевіркою (CA-SCL), який і був обраний стандартом 5G для декодування сигналів на каналах управління (PDCCH). Такі вдосконалення дозволяють значно знизити ймовірність помилки без критичного зростання складності. Окрім того, досліджено перспективні варіанти алгоритмів Polar-декодування, а саме SC Flip та Fast-SC, що мають потенціал для використання в системах з обмеженими ресурсами, таких як IoT.

Загалом порівняння LDPC та Polar-кодів засвідчило, що перші краще підходять для довгих блоків даних у високошвидкісних каналах (наприклад, PDSCH), у той час як другі — для коротких кодів у каналах керування, де потрібна висока надійність при низькому рівні BLER. Такий розподіл кодів у стандарті 5G NR є раціональним з точки зору балансу між продуктивністю, затримками та складністю реалізації.

### **3 РОЗРОБКА МЕТОДУ ІНТЕЛЕКТУАЛЬНОГО ДЕКОДУВАННЯ ЗАВАДОСТІЙКИХ КОДІВ ДЛЯ 5G/6G**

#### **3.1 Обґрунтування можливостей використання AI у декодуванні**

Сучасні телекомунікаційні системи, зокрема 5G та перспективні 6G, вимагають надзвичайно високої пропускну здатності, мінімальної затримки (latency) та підвищеної надійності передавання даних. Традиційні алгоритми декодування, хоч і є ефективними у стандартних умовах, мають низку обмежень, особливо при дії сильних завад, нелінійних спотворень або складних каналів зв'язку (із ефектами багатопроменевості - Rayleigh або доплерівського зсуву).

У цих умовах виникає потреба в адаптивних, контекстно-залежних декодерах, які здатні «навчатися» з каналних умов і гнучко адаптувати правила декодування. Саме тут штучний інтелект (AI) — зокрема глибоке навчання (Deep Learning), рекурентні мережі (RNN) та графові нейронні мережі (GNN) — відкривають нові горизонти. На відміну від класичних детермінованих алгоритмів, AI-підходи здатні апроксимувати складні функції декодування без явного математичного опису, створюють можливості для декодування з урахуванням попереднього досвіду або статистики з каналів, а також здатні працювати в режимах низьких відношень сигнал-шум або за умов складної топології кодів (LDPC, Polar), де класичні алгоритми демонструють зниження продуктивності.

В цьому контексті застосування глибоких нейронних мереж (DNN) дозволяє побудувати модель, яка ефективно апроксимує функцію декодування, мінімізуючи функцію втрат типу “cross-entropy” між вхідним шумовим вектором та правильним кодовим словом. GNN-архітектури, у свою чергу, можуть бути корисними та ефективними у декодуванні LDPC-кодів, де факторна графова структура природно підходить до структури графа Таннера.

Крім того, інтелектуальні методи можуть враховувати апаратні обмеження (розрядність, енергоспоживання, обчислювальні ресурси), дозволяючи створювати гнучкі системи з компромісом між продуктивністю та складністю. Наприклад, використання підкріпленого навчання (Deep Reinforcement Learning, DRL) дозволяє здійснювати пошук оптимальних політик прийняття рішень при декодуванні в змінному каналі.

Якщо торкнутись практичних аспектів можливих наслідків застосування інтелектуальних рішень при декодуванні то слід наголосити на наступному. Як було вже наголошено вище сучасні телекомунікаційні системи, зокрема мережі п'ятого (5G) та перспективного шостого покоління (6G), висувають нові, безпрецедентно жорсткі вимоги до систем передавання даних. Серед них ключовими є надвисока пропускна здатність, наднизька затримка (latency), підвищена надійність, ефективне використання спектру та гарантії якості обслуговування (QoS). Такі вимоги продиктовані появою нових класів додатків — Smart City, Smart Home, автономного транспорту, доповненої/віртуальної реальності (AR/VR/XR), дистанційної хірургії, хмарних ігор, тактильного інтернету, тощо.

У цьому контексті завадостійке кодування, зокрема на основі LDPC та полярних кодів, стає критично важливим елементом фізичного рівня (PHY). Проте традиційні методи декодування — наприклад, BP або SC які були розглянуті і в другому розділі магістерської дипломної роботи — мають низку обмежень, особливо в умовах необхідності працювати в режимі урізаних обчислювальних ресурсів (наприклад, як це притаманне пристроям IoT) або жорстких енергетичних обмежень вже згаданих високих вимог до затримки (і зокрема sub-ms latency). Саме в таких умовах інтелектуальні методи декодування, засновані на штучному інтелекті, особливо глибокому навчанні, графових нейронних мережах (GNN), рекурентних структурах (RNN/LSTM) та підкріпленому навчанні (DRL), потенціо можуть демонструвати значну

перевагу. Основні переваги AI-підходів у декодуванні тут можна сформулювати в напрямку можливостей апроксимації складних функцій декодування без необхідності ручного моделювання, адаптація до змін каналу на основі історичних даних, без потреби в переобчисленні моделей, контекстуальна обробка сигналів — з урахуванням шуму, нелінійностей, завад та топології кодів, зменшення кількості ітерацій у процесі декодування при збереженні якості, Оптимізація апаратної реалізації з урахуванням енергоспоживання та затримки (edge/IoT/embedded deployment).

По суті можна стверджувати, що з AI для LDPC та Polar відкриваються нові горизонти. У випадку LDPC-кодів, графова структура Таннера дозволяє ефективно застосовувати Graph Neural Networks, які навчаються оптимізованим повідомленням між вузлами факторного графа. Такі моделі досягають вищої точності при меншій кількості ітерацій, зберігаючи масштабованість. Для полярних кодів, AI-алгоритми дозволяють модифікувати або адаптувати процедури SC - list decoding (SCLD), враховуючи історичну статистику помилок. Акцентуємо, і це буде розкрито в експериментальній частині магістерської роботи, що особливо перспективним є застосування reinforcement learning у динамічному каналному середовищі, де декодер може навчатися найкращим стратегіям прийняття рішень залежно від змін середовища.

Крім того, інтелектуальне декодування відкриває шлях до реалізації ряду критичних сервісів нового покоління:

- 1) Autonomous Vehicles=Автономний транспорт. Тут забезпечується наднадійний зв'язок URLLC (Ultra-Reliable Low Latency Communication) між транспортними засобами. AI-декодері дозволяють досягти затримки  $<1$  мс при  $BER < 10^{-6}$ , що критично для запобігання аваріям;

- 2) Концепції Smart City/Smart Grid. Тут масивна кількість сенсорів у MTC (Machine-Type Communication) режимі потребує декодерів з низьким

енергоспоживанням та адаптацією до нестабільних каналів і звісно інтелектуалні рішення які розглядаються в дипломній роботі будуть надзвичайно доречними;

3) Рішення AR/VR/XR/Gaming. Вимагають високі вимоги до затримки (latency-sensitive) та стабільного декодування потоків 4K/8K відео у реальному часі. AI-декодери дозволяють мінімізувати буферизацію і збої при відтворенні;

4) Для напрямків Tactile Internet/Remote Surgery. Тут визначальною є вимога до 99.9999% надійності та субмілісекундної затримки при передачі команд і зворотного зв'язку. Тому застосування інтелектуального кодування надзвичайно актуальне;

5) Напрямки Industry 4.0/ІоТ. Вимагають оптимізації процесу синхронізація роботизованих систем із сотнями пристроїв у реальному часі з QoS-контролем та self-healing адаптивними мережами.

Нижче (табл. 3.1) в систематизованому виді представлено потенційні (дослідження частини буде предствлена в наступному розділі магістерської роботи) ключові метрики покращення перспективних телекомунікацій при застосуванні запропонованих рішень.

Таблиця 3.1 - Практичні ефекти впровадження AI-декодування черезі ключові метрики покращення

Показник	Класичні методи	AI-декодування
<b>BER (Bit Error Rate)</b>	$\sim 10^{-5}$ при low SNR	$\sim 10^{-7}$ при низькому SNR
<b>Latency</b>	1–5 мс	<1 мс (для URLLC)
<b>QoS адаптація</b>	Жорстко задана	Динамічна, контекстна
<b>Споживання енергії</b>	Високе при складному декодуванні	Оптимізоване, енерго-обізнанні (energy-aware) рішення
<b>Масштабованість</b>	Лімітована складністю	Висока, завдяки навчанню
<b>Інтеграція в embedded (вбудовані системи)</b>	Важка	Можлива за допомогою quantized AI моделей (квантовані моделі – оптимізовані, низькорозрядні)

Зокрема, у випадку 4K/8K відеострімінгу в AR/VR-додатках, AI-декодування дозволяє скоротити затримку при декодуванні на до 30–50%, що безпосередньо впливає на якість сприйняття (QoE) та усунення ефекту «motion sickness».

На рис. 3.1 представлено AI-декодування в 5G/6G в контексті переваг покращення та сфер застосування в прикладних напрямках.

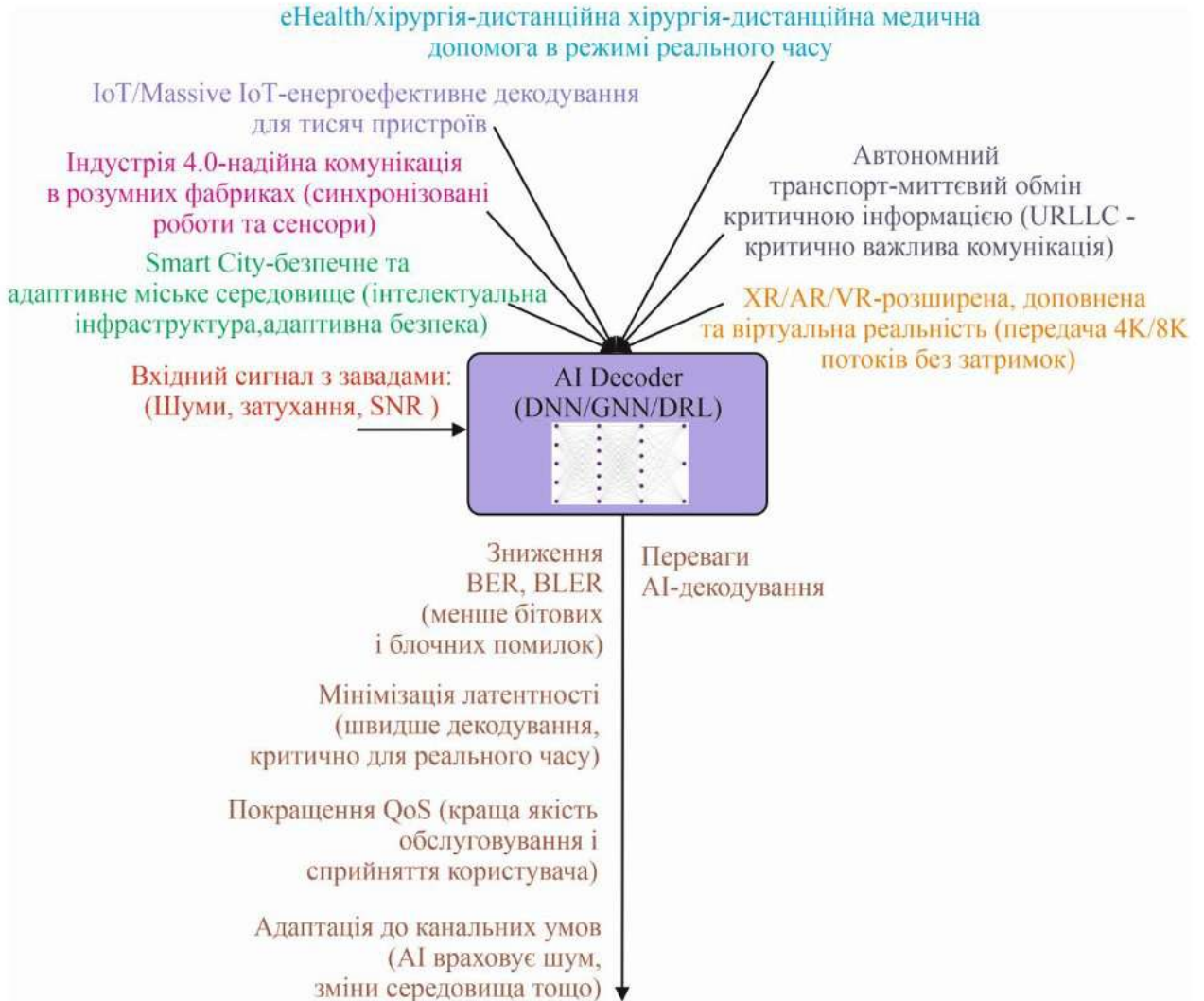


Рисунок 3.1 – AI-декодування в 5G/6G - покращення та сфери застосування

Таким чином, використання AI у декодуванні є не просто заміною класичних алгоритмів, а кроком до когнітивного, адаптивного приймача, здатного навчатися, прогнозувати та динамічно адаптуватись до реальних умов 5G/6G-середовища. Це обґрунтовує приведені в кваліфікаційній

дипломній роботі дослідження архітектур AI-декодерів, які будуть розглянуті у наступних підрозділах пояснювальної записки.

## **3.2 Архітектури нейромереж для декодування**

### **3.2.1 Використання архітектури DNN (Deep Neural Networks) для розробки інтелектуального декодера 5G**

Глибоки нейронні мережі (DNN) є одним з базових інструментів штучного інтелекту, які використовуються для апроксимації складних нелінійних функцій [42]. У контексті декодування завадостійких кодів, таких як LDPC та полярні коди які розглядаються у дипломній роботі, DNN дозволяють створювати адаптивні декодери, здатні навчатися зі статистики каналу та реагувати на нетривіальні завади або нестандартну топологію коду.

Розглянемо загальну концепцію використання DNN у декодуванні. Концепція ґрунтується на застосуванні формалізації коли DNN-декодери навчаються на множині пар типу вхідний вектор  $y$ , правильне кодове слово яке позначимо за  $x$ . Тобто тлумачення застоування DNN буде таке:  $y$  — це спотворений сигнал, отриманий з каналу (ми розглядаємо в дипломній роботі канали з адитивним білим гаусовим шумом — AWGN), а  $x$  - бітова послідовність, яку потрібно відновити. Тоді Мета DNN — побудувати функцію типу:

$$\hat{x} = f_{\theta}(y), \quad (3.1)$$

задача якої полягає у мінімізації похибки між переданими та відновленими бітами.

Типовою функцією втрат є крос-ентропія або бінарна ентропія, а процес навчання реалізується за допомогою градієнтного спуску (зокрема ми нижче розглянемо формат Adam-оптимізатора).

Справа у тому, що в задачах класифікації, особливо бінарної (зокрема, правильне/неправильне декодування біта яким частково присвячена дипломна

магістерська робота), типовою функцією втрат є бінарна крос-ентропія. Вона забезпечує міру відстані між двома розподілами ймовірностей, а саме справжнім (міткою)  $y \in \{0,1\}$  і передбаченим (виходом моделі)  $\hat{y} \in (0,1)$ .

Формалізовано цю ознаку представимо наступним чином:

$$\mathfrak{J}(y, \hat{y}) = -[y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})] \quad (3.2)$$

Зміст формули 3.2 полягає в тому, що у випадку коли передбачення моделі близьке до правильного значення, функція втрат мала. І навпаки, якщо модель сильно помиляється, втрата зростає, що сприяє суттєвій корекції ваг. Такий підхід щодо вибору функції втрат (вибір крос-ентропії) ґрунтується на тому, що ця функція природньо узгоджується з імовірнісною інтерпретацією виходу моделі, що особливо важливо у випадках, коли нейронна мережа імітує ймовірності бітів у кодовому слові. Крім того, і це надзвичайно важливе для представленого в магістерській роботі дослідження, для задач декодування це критично, оскільки неправильне відновлення навіть одного біта може зумовити декодування всієї послідовності як хибне.

Для процесу навчання в магістерській роботі був використаний метод градієнтного спуску. Нижче буде представлено обґрунтування такого вибору. Тут зазначимо, що градієнтний спуск — це метод оптимізації, який поступово коригує параметри нейронної мережі для зменшення функції втрат. На кожній ітерації ваги оновлюються за формулою яку подамо наступним чином [43]:

$$\theta \leftarrow \theta - \eta \cdot \nabla_{\theta} \mathfrak{J} \quad (3.3)$$

де  $\theta$  - вектор параметрів (ваг) мережі;  $\eta$  - швидкість навчання (так звана learning rate);  $\nabla_{\theta} \mathfrak{J}$  - градієнт втрат по параметрах.

Загалом ідея використання саме методу градієнтного спуску для задач завадостікого кодування ґрунтується на наступних засадах:

1) Мережа для інтелектуального декодування (наприклад, нейронна варіація декодера BP або SC) має диференційовану структуру, що дає змогу ефективно використовувати зворотне поширення помилки.

2) Це дозволяє адаптивно коригувати параметри, щоб наблизити вихід мережі до правильного декодування навіть у складних зашумлених умовах.

З точки зору оптимізації метода градієнтног спуску для вирішення мети дипломної роботи, в роботі було застосовано оптимізатор Adam [44]. Загалом, Adam (Adaptive Moment Estimation) — один з найефективніших сучасних оптимізаторів градієнтного спуску, що поєднує переваги методів Momentum та RMSprop. Причому Momentum — це модифікація градієнтного спуску, яка враховує "інерцію" попередніх кроків. Завдяки цьому оновлення параметрів набувають згладженого характеру, що дозволяє швидше долати негладкі або плоскі ділянки ландшафту функції втрат. Тоді як, RMSprop (Root Mean Square Propagation) адаптує швидкість навчання для кожного параметра, зменшуючи її для тих, де градієнти мають великий розмах. Це допомагає стабілізувати навчання й уникати надмірних коливань у напрямку оновлення. Підсумовуючв за цими методами можна казати, що обидва методи добре працюють у складних задачах оптимізації, але Adam поєднує їхні ключові переваги, забезпечуючи ефективнішу та більш стабільну збіжність.

Формули оновлення ваг для Adam представимо начтупним чином:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \nabla_{\theta} \mathfrak{J}_t \quad (3.4)$$

де  $m_t$  - експоненційно-зважене середнє градієнта (1-й момент);  $\beta_1$  - коефіцієнт згладжування для першого моменту (з теорії звичайно  $\beta_1 = 0,9$ );  $\nabla_{\theta} \mathfrak{J}_t$  - градієнт функції втрат  $\mathfrak{J}$  у момент часу  $t$  за параметром  $\theta$ .

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot (\nabla_{\theta} \mathfrak{J}_t)^2 \quad (3.5)$$

де  $v_t$  - експоненційно-зважене середнє квадрату градієнта (2-й момент);  $\beta_2$  - коефіцієнт згладжування для другого моменту (з теорії звичайно  $\beta_2 = 0,999$ ).

Щоб врахувати початкове зміщення, обчислювали кориговані моменти (тобто враховували зміщені оцінки моментів):

$$\hat{m}_{t-1} = \frac{m_t}{1-\beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad (3.6)$$

де  $\hat{m}_{t-1}$ ,  $\hat{v}_t$  - незміщені оцінки відповідно першого та другого моменту;  $t$  - номер поточної ітерації.

Накінець, оновлення параметрів здійснювали за формолою:

$$\theta_t = \theta_{t-1} - \eta \cdot \frac{\hat{m}_{t-1}}{\sqrt{\hat{v}_t + \varepsilon}} \quad (3.7)$$

де  $\theta_t$  - оновлене значення параметра;  $\eta$  - швидкість навчання (learning rate);  $\varepsilon$  - мала константа для стабільності ділення (з теорії береться  $\varepsilon = 10^{-8}$ ).

Отже, Adam веде облік середнього градієнта (як Momentum) та середнього квадрата градієнта (як RMSprop) і коригує їх, щоб оновлювати параметри стабільно та адаптивно, навіть при нерівномірному масштабі градієнтів.

Щоб отримати проєкцію представленої методики на задачі які вирішуються в представленій дипломній роботі необхідно наголосити на наступному. У задачі інтелектуального завадостійкого декодування LDPC та полярних кодів у системах 5G, оптимізатор Adam ми застосовуємо в процесі навчання нейронної мережі, яка моделює або доповнює класичний декодер. То у відповідності до формул 3.2 – 3.7 нижче поставимо відповідність інтерпретації параметрів і формул зазначеного в методі Adam контексту.

В нашому випадку  $\theta$  це набір параметрів нейронної мережі, яку ми тренуємо для виконання або покращення декодування (зокрема у якості нейродопоміжного блоку у гібридному LDPC/Polar-декодері (ваги, зміщення)). Відповідно розгорнуто ці параметри включали ваги та зміщення у нейронах (коефіцієнти з'єднань), параметри нормалізації, фільтрів або активаційних функцій (залежно від архітектури). Тоді функція втрат  $\mathfrak{J}$ , це

бінарна крос-ентропія між реальними інформаційними бітами  $u$  та передбаченими мережею бітами  $\hat{u}$ . По суті, вона вимірює відстань між очікуваними і передбаченими результатами, тобто якість декодування. Далі, градієнт  $\nabla_{\theta} \mathcal{L}_i$  - визначає, як сильно кожен параметр  $\theta$  впливає на функцію втрат. Обчислюється з урахуванням оберненого поширення похибки (backpropagation) по всій мережі. Крім того, оновлення ваг виконувалось (експериментальні дослідження будуть представлені у 4-у розділі дипломної роботи) так, щоб зменшити похибку декодування на тренувальному наборі — з урахуванням мінливого середовища (шуму, завад, типу каналу де у нас враховувались канали AWGN, Rayleigh). Отже, процес оптимізації нейромережі передбачав значення бітів  $\hat{u}$  з імовірнісного представлення LLR (дивитись 2-й розділ дипломної роботи), крім того оптимізації допомагала вибрати найкращий шлях декодування (дивитис розділ 2 дипломної роботи відносно методики спискового декодування - CA-SCL), а також при необхідності оптимізація сприяла відфільтровці помилкових рішень при ітеративному декодуванні у LDPC.

Таким чином можна аргументовано сформувані переважні ознаки які спонукали обрати в дипломній роботі метод Adam. Adam забезпечує стабільне й ефективне навчання навіть при зашумлених градієнтах (характерно для декодування в умовах Rayleigh або AWGN, що відповідає меті дипломної роботи), різній амплітуді вхідних LLR коли використання просто методики RMSprop недоцільне в наслідок виникнення похибок і хибних рішень, при можливості роботи з невеликими пакетами даних (mini-batches), що концептуально характерно при симуляції декодування.

Тоді остаточно акцентуємо, що обраний нами оптимізатор Adam адаптується до змін сигналу, шуму, варіацій каналу, а це в свою чергу дозволяє інтегрувати нейронну мережу в структуру декодера для інтелектуального й адаптивного прийому у 5G-системах.

Тепер торкнемось архітектури DNN-декодера. Архітектуру DNN-декодера проедставимо у вигляді послідовного концепта згортання (feed-forward network). Відповідно такий концепт буде містити:

- 1) Вхідний шар - приймає вектор м'яких значень (log-likelihood ratios або просто спотворені біти);
- 2) Кілька прихованих шарів (Fully Connected, FC) - з нелінійною активацією (ReLU, Tanh, ELU тощо)
- 3) Вихідний шар який дає вектор імовірностей бітів (зокрема в дипломній роботі буде використана сигмоїдна активація).

Аналітично постідовність кроків концепту представимо так:

$$h_1 = \text{ReLU}(W_1 \cdot y + b_1) \quad (3.8)$$

$$h_2 = \text{ReLU}(W_2 \cdot h_1 + b_2) \quad (3.9)$$

$$\dots\dots\dots$$

$$\hat{y} = \text{Sigmoid}(W_n \cdot h_n + b_n) \quad (3.10)$$

де  $\hat{y}$  - вектор ймовірностей бітів (або логіти), які можна порогувати для отримання бінарного результату.

Пропозиція до реалізації наступна. У випадку LDPC-кодів, як було показано у розділі 2 дипломної роботи, такі коди описуються факторним графом (граф Таннера (рис. 2.1)), який визначає зв'язки між бітами та перевірками на парність. Традиційне ВР декодування реалізує ітеративний процес передачі повідомлень між вузлами графа, як було описано у розділі 2. У випадку DNN пропонується, щоб DNN-декодери імітували структуру ВР, але з навчанням ваг для кожного етапу ітерації, що дозволяє мережі самостійно оптимізувати повідомлення. Такий підхід можна охарактеризувати як learned ВР=пізнавальний ВР або deep unfolding=з глибоким розгортанням. Тоді можна подати структу – каркас алгоритму у форматі:

- 1) Створюється мережа з  $T$ -блоків, кожен з яких імітує одну ітерацію ВР;

2) Кожен блок має навчальні параметри, що дозволяють гнучко налаштувати послідовність оновлень;

3) Як підсумок - це дозволяє ефективніше працювати при коротких довжинах коду або при низьких відношеннях сигнал-шум (SNR).

На рис. 3.2 представлено блок-схема декодера LDPC-коду яка відповідає архітектурі DNN-декодера при декодуванні LDPC-кодів, зокрема у випадку "fully connected surrogate decoder= повнозв'язний апроксимаційний декодер", коли класичний BP-декодер апроксимується за допомогою DNN.



Рисунок 3.2 – Блок схема декодера LDPC-коду яка відповідає архітектурі DNN-декодера

В схемі вхідний вектор  $\mathbf{y}$  представимо у формі:

$$\mathbf{y} = (y_1, y_2, \dots, y_N) \quad (3.11)$$

де  $y_i \in \mathbb{R}$  — це логарифмічне відношення ймовірностей для біта (LLR), або просто значення з демодулятора (в представленій дипломній роботі, як буде показано у 4-у розділі, з виходу QAM-демодулятора).

Тобто, якщо передавали біт 1, а отримали значення +2.3 — модель має зрозуміти, що це скоріше "1". Тоді як у випадку якщо -1.5 — це ближче до "0", і якщо 0 — значить повна невизначеність (SNR $\approx$ 0).

На виході отримаємо згідно до рис. 3.2:

$$\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_K) \quad (3.12)$$

де  $\hat{x}_i \in [0, 1]$  — це ймовірність того, що відповідний інформаційний біт дорівнює 1;  $K$  - кількість інформаційних бітів (тобто до декодування).

До виразу (3.4) доаємо, що в дослідженнях представлених в дипломній роботі використовувалась сигмоїдна функція активації на останньому шарі виду:

$$\hat{x}_i = \sigma(z_i) = \frac{1}{1 + e^{-z_i}} \quad (3.13)$$

де  $z_i = \sum_j w_{ij} \cdot h_j + b_i$ ;  $h_j$  - вихід з попереднього шару (наприклад, FC3);  $w_{ij}$  - вага, що з'єднує нейрон  $j$  з нейроном  $i$ ;  $b_i$  - зміщення;  $z_i$  - лінійна активація нейрона  $i$  перед сигмоїдою (тобто лінійна комбінація вхідних значень, сигмоїда перетворює  $z_i$  у ймовірність в межах  $[0, 1]$ ).

Далі по схемі на рис. 3.2 отримуємо. Вхідний шар приймає soft-інформацію з каналу у вигляді LLR, характерну для LDPC. Тоді сховані шари (FC1–FC3) — це повнозв'язані шари, які поєднують бітову інформацію та навчаються знаходити шаблони, характерні для правильно декодованих слів. Вихідний шар видає ймовірності або логіти кожного біта — це soft-рішення (м'яке), яке можна порогувати. Зокрема, у в дипломній роботі пропонується використовувати "unfolded BP= розгорнуте BP" підхід тоді кожен шар DNN може моделювати одну або кілька ітерацій BP (дивитись 2-й розділ дипломної роботи за BP алгоритм). Що маємо в результаті. Розгорнуте BP це коли класичний алгоритм декодування BP перетворюється на нейронну мережу з фіксованою кількістю шарів, де кожен шар відповідає одній ітерації BP. Практично це означає, що замість запуску BP як звичайного ітеративного алгоритму, його "розгортають у глибину", як послідовність шарів нейромережі. Вага повідомлень на кожному кроці стає навчальним параметром. Ну і така мережа вчиться ефективніше декодувати на практичних каналах (замість 5 ітерацій BP будується 5-шарова нейромережа). Ця архітектура є спрощеною, але підходить для кодування коротких/середніх кодів, які є основою в системах з обмеженими обчислювальними ресурсами (наприклад, IoT).

Як видно з рис. 3.2 було застосовано класичну зважувальну архітектуру. Тоді перший шар (FC1 = 512) бере на себе масивну обробку вхідного сигналу — великої розмірності, шумний, необроблений, а подальші шари (256, 128)

фільтрують, узагальнюють і витягують суттєві ознаки (features), які найкраще відповідають шаблонам "правильного" кодового слова. Іншими словами, 512 нейронів у FC1 — достатньо для обробки векторів довжиною  $N \approx 128-256$  (довжина LDPC-коду [33]). Зауважимо, що звуження шарів дозволяє економити пам'ять та обчислення, що критично в embedded-системах (наприклад, модем 5G/6G або IoT-чип). Далі приведена табл. 3.2 де зведена узагальнена концепція формування шарів нейронів.

Таблиця 3.2 – Стратегія формування та призначення нейронних шарів при побудові інтелектуального декодера 5G

Шар	Призначення
<b>FC1 (512)</b>	Початковий аналіз спотвореного вектору
<b>FC2 (256)</b>	Узагальнення ключових бітових патернів
<b>FC3 (128)</b>	Вибір найімовірнішої комбінації інформаційних бітів
<b>Output</b>	Soft-оцінки $K$ -бітів (ймовірності)

Тепер повернемося до рівняння 3.2, 3.3 яке було використано для концепту декодування. Там використана нелінійна функція активації ReLU (Rectified Linear Unit) [45], яка загалом, перетворює вхідні значення наступним чином:

$$\text{ReLU}(z) = \max(0, z) \quad (3.14)$$

Такий підхід в дипломній роботі використано з прицілом на те, що у кожному шарі DNN ми спочатку робимо лінійне перетворення вхідного вектора, а потім застосовуємо нелінійну активацію, щоб модель могла вивчати складні, нелінійні залежності. Таку формалізацію було використано тому, що ReLU легко реалізується навіть у програмному форматі, уникає затухання градієнтів на відміну від sigmoid/tanh не "сплющує" великі значення, а також дозволяє швидше сходиться під час тренування та загалом немає негативного діапазону тобто частина нейронів може "вимкнутися" (для output=0).

Стосовно схемотехніки інтелектуального декодера який пропонується у прихованих шарах використовується функція активації ReLU, яка забезпечує необхідну нелінійність моделі, дозволяючи нейромережі ефективно апроксимувати складні функції декодування. Вона також сприяє уникненню проблеми затухання градієнтів та прискорює процес навчання. Завдяки нелінійності, мережа може моделювати ефекти завад, мультиплікативних шумів або нелінійного згасання, ReLU дозволяє моделі вивчити складні спотворення в LLR-векторі і це безсумнівно покращує здатність до відновлення бітів навіть в умовах низьких відношень сигнал=шум.

Тепер торкнемось питань застосування DNN-концепції до полярних кодів. Власне в чому проблема? Для полярних кодів, класичним методом є SC або SCL які детально розглянуто у 2 гому розділі магістерської роботи. Але вони чутливі до ринвів відношення сигнал-шум (SNR) і конфігурації каналу. DNN-декодеру тут можуть моделювати окремі частини SCL-процесу (наприклад, вибір між гілками), навчатися ефективно визначати інформаційні біти серед усіх  $N$ -позицій, замінити частину детермінованих блоків AI-модулями (наприклад, trainable path metrics= метрики шляху, що піддаються навчанню). Що можна очікувати в підсумку - зниження BER/BLER у порівнянні з класичними методами на низькому відношенні сигнал-шум, гнучкість у реалізації коли можна перенавчити мережу на нові каналні умови або прискорення на inference-етапі, особливо на апаратному рівні (FPGA, TPU) з'являється можливість квантизації моделі для edge-застосунків (що особливо корисно для IoT, мобільних пристроїв). Для реалізації такого підходу потрібна необхідність великого обсягу даних для навчання, потреба в оптимізації розміру мережі для низькопотужних пристроїв.

Для візуалізації описаних перетворень при імплементації DNN-декодера, з метою наочного пояснення того як працює один шар DNN-декодера для LDPC (полярних) кодів нижче представлений рис. 3.3. На рис.3.3 зображено

фрагменту нейронної архітектури DNN-декодера, а саме один прихований шар з ReLU. Зауважимо, що згідно до методики яка представлена вище (дивитись формули 3.1 – 3.6) на кожному нейроні прихованого шару виконується обчислення згідно до 3.1, 3.2:  $z_i = \sum_j w_{ij} \cdot h_j + b_i$  де  $h_j = \max(0, z_j)$ .

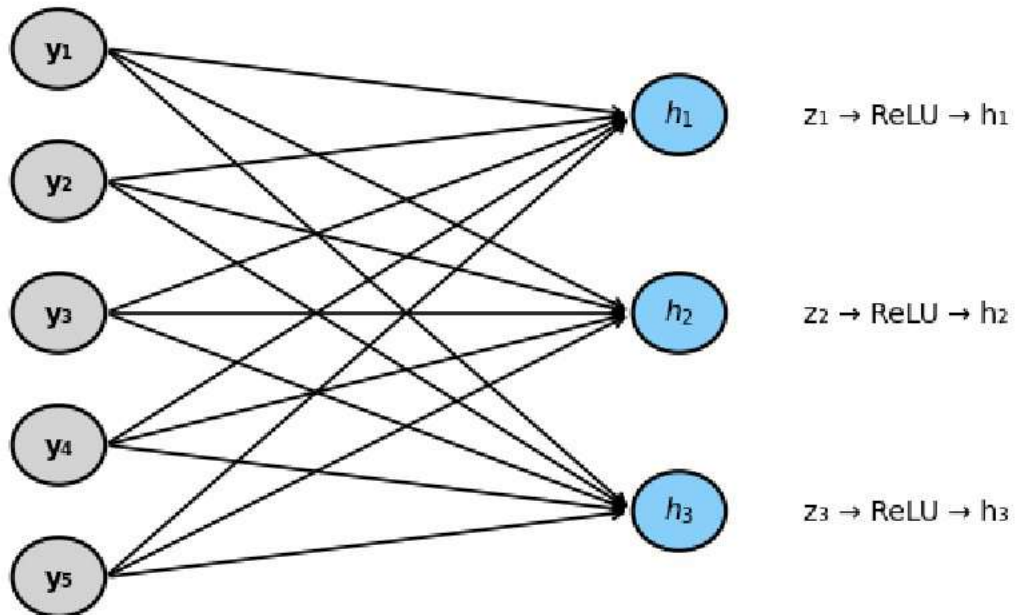


Рисунок 3.3 - Фрагмент нейронної архітектури DNN-декодера (один прихований шар з ReLU)=локальна нейронна діаграма шару

На рис. 3.3 представлено входні нейрони  $y_1, \dots, y_5$ , три нейрони прихованого шару з обчисленням  $z_i \rightarrow \text{ReLU} \rightarrow h_i$ , стрілки з'єднання відображають вагу, а саме перетворення відбувається у відповідності до  $z_i = \sum_j w_{ij} \cdot h_j + b_i$ .

На рис. 3.4 аналогічно представлено фрагмент вихідного шару DNN-декодера з Sigmoid при отриманні soft (м'яких) оцінок бітів.

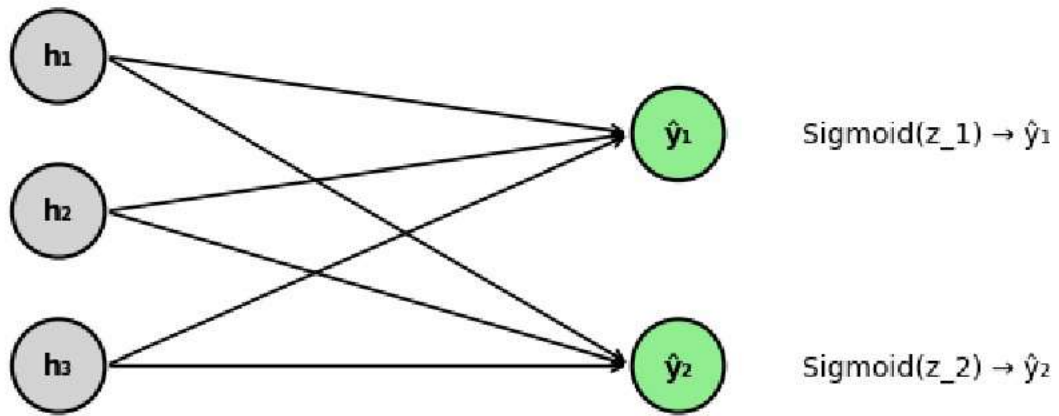


Рисунок 3.4 – Схема вихідного шару DNN-декодера з Sigmoid (показано процес отримання soft (м'яких)-рішень)

Замітимо, що на кожному вихідному нейроні виконується обчислення типу:  $z_i = \sum_j w_{ij} \cdot h_j + b_i$  де  $\hat{y}_i = \sigma(z_i)$ . Бачимо на схемі приховані нейрони виду  $h_1, h_2, h_3$ , та вихід (soft-оцінки бітів)  $\hat{y}_1, \hat{y}_2$ , що проходять через Sigmoid.

Отже, на основі рисунків 3.3, 3.4 можна подати концепцію повної конвеєрної (pipeline) обробки (рис. 3.5) у вигляді послідовної передачі даних від шару до шару. Тоді кожен шар нейромережі приймає дані з попереднього шару, перетворює їх (через ваги, активації), і передає далі аж до фінального шару, який видає прогноз (зокрема як в нашому дослідженні у вигляді soft оцінки біта через Sigmoid).

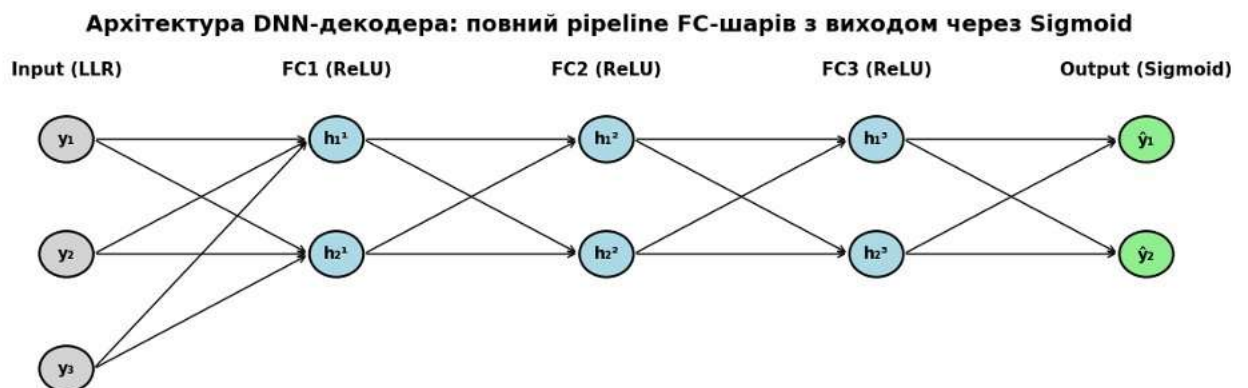


Рисунок 3.5 – Архітектура DNN-декодера як повний pipeline FC-шарів (повний конвеєр) з виходом через Sigmoid

Акцентуємо по рис. 3.5, тут конвеєр містить послідовну побробку вхідних LLR-даних через шари повнозв'язної мережі конфігурації  $FC1 \rightarrow FC2 \rightarrow FC3 \rightarrow Output$ . Тобто в структурі архітектури кожен шар трансформує дані через ReLU та Sigmoid до Soft-виходу (м'якого виходу).

### **3.2.2 Використання архітектури рекурентних нейронних мереж RNN/LSTM (при використанні каналу з пам'яттю) для конструювання інтелектуального декодера 5G**

RNN/LSTM (Recurrent Neural Network=рекурентна нейронна мережа/Long Short-Term Memory=з спеціальними механізмами пам'яті (дослівно, мережа яка містить механізм довгої короткочасної пам'яті)) це різновиди рекурентних нейронних мереж, які особливо корисні для каналі з пам'яттю, тобто коли поточний символ залежить від попередніх, наприклад, через міжсимвольні інтерференції (ISI) [3], доплерівський зсув чи інші кореляції у часі.

У бездротових системах зв'язку п'ятого покоління (5G), зокрема у випадках, коли використовуються канали з пам'яттю — такі як канали з міжсимвольними інтерференціями (ISI), доплерівськими ефектами або багатопроменевістю, – традиційні методи декодування (зокрема для LDPC та полярних кодів) втрачають ефективність. Це обумовлено тим, що вони звичайно припускають незалежну обробку символів, у той час як у реальних каналах спостерігається внутрішня залежність між послідовними символами. Саме в таких кейсах доцільно використати RNN і їхня модифікація — мережі з довгою короткочасною пам'яттю – LSTM, визначені вище в поясненні, ефективно працюють із послідовностями, оскільки зберігають інформацію про попередні елементи вводу, що критично важливо для декодування сигналів, що зазнали спотворень у часі. Отже, в рамках завдань які поставлені для

розв'язання в дипломній роботі доцільно розглянути такі нейронні мережі та провести їх адаптацію до експериментальних потреб.

RNN — це нейронна мережа, у якій вихід поточного етапу залежить не лише від поточного входу  $x_t$ , а і від прихованого стану  $h_{t-1}$ . Математично представимо це так:

$$h_t = \Psi(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (3.15)$$

$$y_t = \phi(W_{hy}h_t + b_y) \quad (3.16)$$

де  $x_t$  - вхідний вектор у момент часу  $t$ ;  $h_t$  - прихований стан (так званий вектор пам'яті);  $y_t$  - вихід,  $\Psi$  та  $\phi$  - активаційні функції (наприклад,  $\tanh$  або ReLU які були розглянуті в попередньому підрозділі магістерської роботи).

Тобто у відповідності до оснaк RNN така архітектура (формули 3.15, 3.16) дозволяє мережі "запам'ятовувати" залежності у часі.

Вінносно модифікації RNN. LSTM-архітектура усуває проблему затухання/вибуху градієнта, характерну для звичайних RNN, завдяки спеціальній структурі з "елементами керування пам'яттю" (gating mechanism). Основні елементи такої архітекури можна представити наступним чином:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) - \text{описує блок стирання пам'яті (forget gate)} \quad (3.17)$$

(модуль визначає, яку частину попередньої пам'яті  $C_{t-1}$  потрібно зберегти, а яку стерти. Відповідно для  $f_t$  формуються умови на кшталт: результат ближче до 1 — зберігається, ближче до 0 — стирається)

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) - \text{блок запису (input gate) визначає,} \quad (3.18)$$

яка частина нової інформації буде оновлена в пам'яті комірки (це своєрідна тригер, що відкриває або закриває доступ до оновлення пам'яті);

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) - \text{проміжне значення оновлення} \quad (3.19)$$

пам'яті ( $\tilde{C}_t$ ) генерується як кандидатне оновлення,  
яке далі масштабується блоком запису (input gate)  
та додається до довготривалої пам'яті;

де  $t$  - поточний часовий крок;  $x_t$  - вхідний вектор (прийнятий сигнал або ознака) в момент часу  $t$ ;  $h_{t-1}$  - попередній прихований стан (output);  $h_{t-1}, x_t$  - конкатенація (об'єднання) попереднього стану та поточного входу;  $W_c, W_f, W_i$  - матриці ваг для відповідних блоків (навчаються);  $b_f, b_i, b_c$  - вектори зсуву (bias) для відповідних блоків;  $\sigma(\cdot)$  - сигмоїдна функція активації: значення від 0 до 1;  $\tanh(\cdot)$  - гіперболічний тангенс який обмежено значення від  $-1$  до  $1$ ;  $f(t)$  - вектор маски для стирання;  $i(t)$  - вектор маски для «введення» нової інформації;  $\tilde{C}_t$  - значення кандидатного оновлення.

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \text{ - оновлене значення пам'яті} \quad (3.20)$$

(це внутрішній стан комірки (cell state), який зберігає інформацію з попередніх кроків часу і дозволяє мережі враховувати довгострокові залежності у послідовностях. Тобто, для дослідження представленого в межах дипломної роботи в задачі інтелектуального декодування в 5G (LDPC або Polar) канал із пам'яттю має кореляцію між символами, в икликану згасанням, міжсимвольною інтерференцією (ISI) тощо).

де  $\odot$  - умовне графічне позначення операції поелементного множення (element-wise multiplication), тобто операції Гадамара (Hadamard product) між векторами однакової довжини;  $f_t$  - ваговий коефіцієнт для збереження/забування інформації (від 0 до 1) тобто це коефіцієнт між 0 і 1, який керує збереженням попереднього стану пам'яті;  $i_t$  - коефіцієнт запису;  $\tilde{C}_t$

- кандидатне значення (нова потенційна інформація);  $C_t$  - новий оновлений стан пам'яті.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) - \text{вихідний модулюючий блок. Посуті це це} \quad (3.21)$$

останній логічний компонент LSTM-комірки, який керує

тим, яка частина оновленої пам'яті  $C_t$  буде виведена як

результат роботи LSTM у поточний момент часу. Він фактично

"вирішує", що саме потрібно передати на вихід мережі (тобто  $h_t$ )

до наступного кроку або шару.

$$h_t = o_t \odot \tanh(C_t) - \text{описує новий прихований стан} \quad (3.22)$$

де  $o_t$  - вектор активації вихідного блоку у момент часу  $t$  (посуті це маска (значення від 0 до 1), що визначає, яку частину пам'яті передати на вихід);  $W_o$

- ваги, пов'язані з output gate;  $h_{t-1}, x_t$  - конкатенація попереднього стану  $h_{t-1}$  та

$x_t$ ;  $b_o$  - зсув (bias) для вихідного блоку;  $C_t$  - поточне значення комірки пам'яті

тобто акумульована інформація;  $h_t$  - вихід LSTM у момент часу  $t$ , який

обчислюється з пам'яті  $C_t$  помноженої на маску  $o_t$ ;  $\tanh(C_t)$  - нормалізує

пам'ять у діапазон  $[-1; 1]$ .

Вихідний модуль керує тим, яку частину пам'яті комірки LSTM передати як результат роботи. Це критично важливо для контролю інформаційного потоку в задачах декодування.

На рис. 3.6 представлено синтезовану схему RNN/LSTM-декодера з CNN, Attention, Loss та Output. Схема ілюструє послідовність обробки сигналу та навчання в моделі, що складається з кількох основних блоків серед яких згідно до рівнянь 3.15 – 3.22 виділимо:

1) Input Sequence (LLR) — початковий вхідний сигнал у вигляді логарифмічних відношень правдоподібностей (LLR), який подається на вхід;

2) CNN Feature Extractor — згортковий нейронний шар, що виділяє локальні ознаки із вхідної послідовності для подальшої обробки;

3) RNN/LSTM Block — рекурентна мережа з довгою короткочасною пам'яттю (LSTM), що виконує послідовну обробку ознак. LSTM усуває проблему затухання та вибуху градієнта, характерну для класичних RNN, завдяки спеціальній структурі з механізмами керування пам'яттю (gating mechanism).

Основні рівняння які математично описують роботу синтезованого RNN/LSTM-декодера для 5G це рівняння 3.17-3.22 (описано вие в тексті пояснювальної записки).

4) Estimated Bits — прогнозовані біти, отримані на основі виходу LSTM-блоку;

5) Attention Module — модуль уваги, що генерує контекстний вектор, який допомагає мережі сфокусуватися на важливих частинах послідовності для покращення якості декодування;

6) Loss Function (Cross-Entropy) — функція втрат, яка порівнює прогнозовані біти зі справжніми (еталонними) для оцінки якості моделі. Вона отримує "compare to target" інформацію зі блоку Estimated Bits.

7) BPTT (Backpropagation Through Time) — алгоритм зворотного поширення помилки у часі, що виконує оновлення ваг у RNN/LSTM-блоці. Отримує значення втрат із Loss Function.

8) Output (Decoded Bits) — остаточний декодований результат, який видається мережею.

Тепер торкнемось зв'язків між блоками. Вхідні дані проходять послідовно через CNN, RNN/LSTM, Attention Module і перетворюються у прогнозовані біти. Прогнозовані біти (Estimated Bits) порівнюються з цільовими у функції втрат (Loss Function) через стрілку "compare to target". Тоді функція втрат передає значення помилки в BPTT, який реалізує навчання мережі, повертаючи градієнти в RNN/LSTM блок (стрілка "update weights"). Зайважимо, що з Loss Function також передається інформація про градієнти у

Attention Module. Далі остаточний прогноз результатів декодування передається на вихід. Отже, синтезована схема ілюструє як прямий потік даних (forward pass) так і зворотні зв'язки (backward pass) для навчання мережі. Далі (в 4-у розділі дипломної роботи) синтезована схема буде використана для дослідження основних показників ефективності декодера для мереж 5G/6G.

Запропонована в магістерській роботі схема RNN/LSTM-декодера (рис. 3.6) з використанням CNN та Attention-модуля представляє сучасний підхід до побудови інтелектуальних декодерів для систем зв'язку п'ятого покоління (5G). Використання рекурентних нейронних мереж із механізмами довгої короткочасної пам'яті дозволяє ефективно моделювати часові залежності у вхідних даних, що є ключовим для коректного декодування послідовностей у складних каналах зв'язку.

Інтеграція Attention-модуля забезпечує адаптивне фокусування на найбільш релевантних частинах вхідного сигналу, що підвищує стійкість до завад та покращує точність відновлення інформації. Застосування CNN для попереднього виділення ознак дозволяє захопити локальні патерни, що також сприяє підвищенню продуктивності системи.

Експериментальні дослідження, які будуть проведені в 4-у розділі, дозволять оцінити ефективність цієї архітектури у реальних умовах каналу 5G. Зокрема, планується визначити такі ключові показники, як рівень помилок бітів (BER), швидкість конвергенції при навчанні, стійкість до різних типів шумів та перешкод, а також обчислювальна ефективність.

Підтвердження ефективності запропонованої моделі у експериментах дасть змогу розглядати її як перспективне рішення для впровадження у практичні 5G-системи зв'язку, де висока якість декодування та адаптивність до мінливих умов каналу є критичними вимогами.

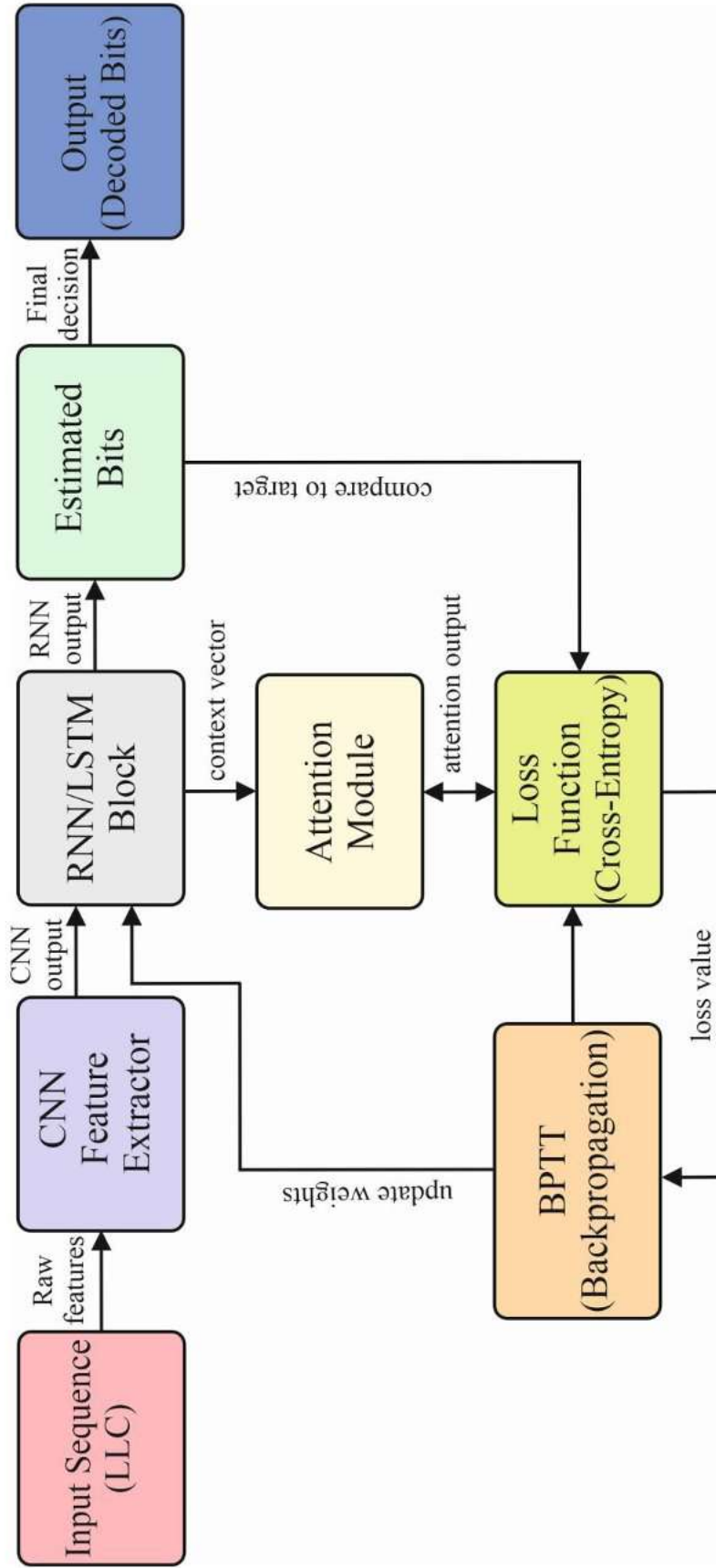


Рисунок 3.6 – Блок-схема запропонованого RNN/LSTM-декодера з використанням CNN та Attention-модуля

### 3.2.3 Особливості застосування концепції Graph Neural Networks (GNN) для LDPC графів при розробці декодерів 5G/6G

GNN — це сучасна архітектура штучних нейронних мереж, яка дозволяє ефективно працювати з даними, що мають структуру графів. LDPC-коди, які використовуються в системах 5G, природно подаються у вигляді біпарного графа Таннера, що складається з вузлів перевірки парності та вузлів змінних, як було описано в роботах [22, 24] та представлено у 2-му розділі магістерської роботи (рис. 2.1, формули 2.1-2.4). Така структура робить GNN особливо перспективною для задач декодування LDPC-кодів.

Для формування теоретичного базису застосування концепції GNN для конструювання декодерів 5G, звернемося до аспектів викладених у другому розділі пояснювальної записки (підрозділ 2.1.1). Зокрема, було наголошено, що граф Таннера для LDPC складається з вузлів змінних (variable nodes,  $v_i$ ) і які відповідають інформаційним або кодовим бітам, вузлів перевірки (check nodes  $c_i$  які відповідають паритетним обмеженням, а також ребер які з'єднують змінні вузли з відповідними перевірками згідно з матрицею парності  $\mathbf{H}$  (розділ 1.3 пояснювальної записки). Якщо звернутись до підрозділу 2.1.1 то там було зазначено, що базовий алгоритм ВР використовує ітеративну передачу повідомлень по графу (рис. 2.1). У випадку ж GNN-декодера, ці повідомлення будуть оновлюватись за допомогою тренованих параметризованих функцій.

Таким чином, математичну модель процесу GNN-декодування представимо наступним чином. Представимо один цикл передачі повідомлень у GNN парою рівнянь:

$$m_{i \rightarrow j}^{(t+1)} = \phi_m^{(t)}(h_i^{(t)}, h_j^{(t)}, e_{ij}) \quad (3.23)$$

$$h_i^{(t+1)} = \phi_u^{(t)}\left(h_i^{(t)}, \sum_{j \in N(i)} m_{j \rightarrow i}^{(t+1)}\right) \quad (3.24)$$

де  $h_i^{(t)}$  - вектор ознак вузла  $i$  на  $t$ -й ітерації;  $m_{i \rightarrow j}^{(t+1)}$  - характеризує процес передачі повідомлення від вузла  $i$  до вузла  $j$ ;  $\phi_m^{(t)}$ ,  $\phi_u^{(t)}$  - параметризовані (треновані) функції, які реалізуються у вигляді Multilayer Perceptron=багатошарового перцептрона (MLP) або GRU/LSTM=Gated Recurrent Unit/Long Short-Term Memory (Рекурентний блок з керованими вентилями/Мережа з довгою короткочасною пам'яттю) (описано у підрозділі 3.2.2);  $N(i)$  – позначення множини сусідів вузла  $i$ ;  $e_{ij}$  - можлива інформація на ребрі (тип з'єднання або вага).

Зауважемо, що згаданий вище MLP —багатошаровий перцептрон, це класична штучна нейронна мережа, яка складається з кількох повнозв'язних шарів (fully connected layers), що реалізують функції вигляду:

$$y = \sigma(Wx + b) \quad (3.25)$$

де  $x$  - вхідний вектор (в нашому випадку - повідомлення у графі);  $W$ ,  $b$  - ваги та зсуви (які навчаються);  $\sigma$  - нелінійна активаційна функція (ReLU, tanh, sigmoid – тобто ті функції які ми вже використовували при виведенні рівнянь 3.17-3.19).

Зауважимо, що в контексті GNN окреслений формат MLP може використовуватись як функція оновлення повідомлень або вузлів, тобто  $\phi_m^{(t)}$ ,  $\phi_u^{(t)}$ .

Відповідно, згадані при формування математичної моделі 3.23, 3.24 GRU/LSTM=Gated Recurrent Unit/Long Short-Term Memory (Рекурентний блок з керованими вентилями/Мережа з довгою короткочасною пам'яттю) це види рекурентних нейронних мереж (RNN), які здатні обробляти послідовну або часову інформацію, зберігаючи стан між ітераціями. Ми зупинялись на форматі цих мереж в підрозділі 3.2.2. Тут акцентуємо лише на тому, що GRU має два клапани (gate) оновлення і збування, і є менш складною за LSTM яку ми використали у розділі 3.2.2. Відповідно, LSTM містить три клапана -

вхідний, збування і вихідний, а також стан пам'яті, що дозволяє ефективніше зберігати довгострокові залежності. Ну і що не менш важливо в подальших викладках, у GNN мережі GRU або LSTM можуть використовуватись замість простих MLP для оновлення станів вузлів графа, якщо потрібен контекст і пам'ять попередніх ітерацій передачі повідомлень. Цей проце математично можна відобразити наступною формою:

$$h_i^{(t+1)} = \text{GRU}(h_i^{(t)}, m_i^{(t)}) \quad (3.26)$$

де  $h_i^{(t)}$  - описує стан вузла  $i$  на ітерації  $t$ ;  $m_i^{(t)}$  - описує агреговані повідомлення з сусідніх вузлів.

Підсумовуючі підкреслимо, що MLP це перцептрон який доцільно використати для простого швидкого оновлення стану вузлів або повідомлень, тоді як рекурнтні форми мереж типу GRU/LSTM доцільні для врахування часової динаміки або залежностей між ітераціями при оновленні вузлів у графі. Тому власне у синтезованій в магістерській роботі RNN/ LSTM декодера для 5G (дивитись рис. 3.6) ми використали саме рекурентний підхід з рьома клапанами (gate).

Тепер повернімось до формування методики застосування концепції GNN для LDPC графів згідно математичних моделей 3.23, 3.24. Для задачі декодування вузли можуть бути ініціалізовані з вектором LLR з демодулятора, тобто математично цей процес опишемо наступною залежністю:

$$h_i^{(0)} = \text{LLR}(v_i) \quad (3.27)$$

Остаточне рішення по біту буде прийматись за ознаками вузла змінної, наступним чином:

$$u_i = \begin{cases} 0, \text{ якщо } h_i^{(T)} \geq 0 \\ 1, \text{ в іншому випадку} \end{cases} \quad (3.28)$$

де  $T$  - кількість ітерацій (глибина проходу графом).

На рис. 3.7 представлено блок-схема інтелектуального GNN-декодера LDPC для 5G.

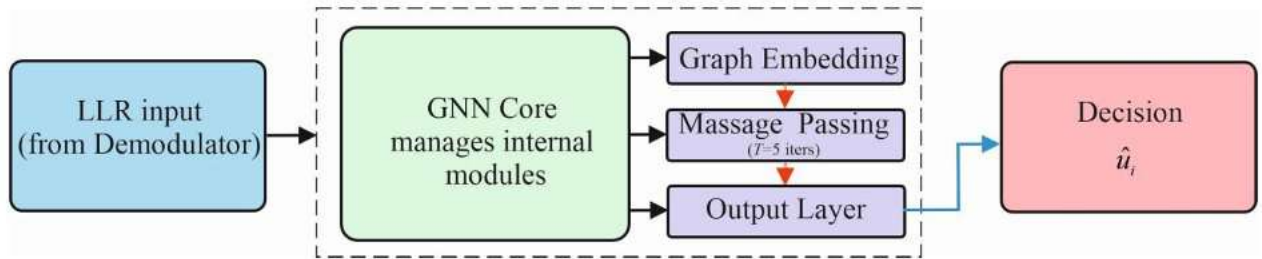


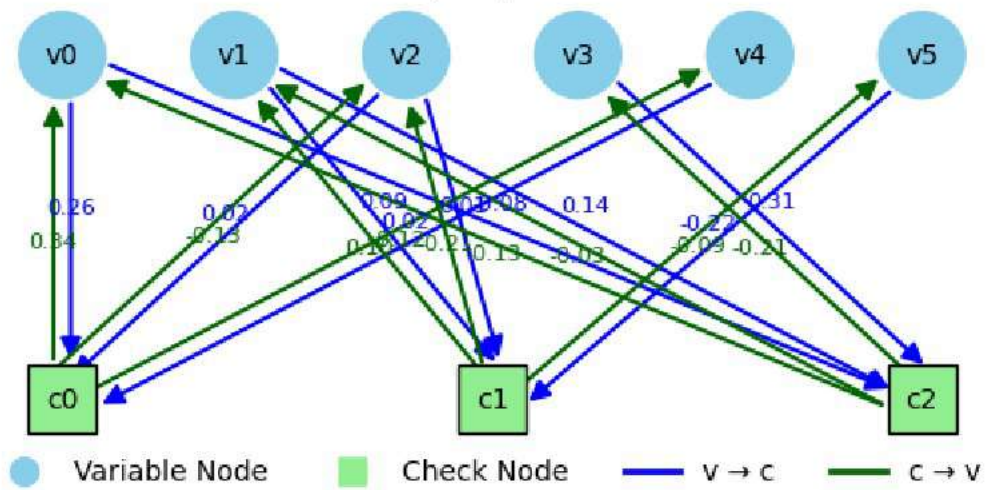
Рисунок 3.7 - Блок-схема інтелектуального GNN-декодера LDPC для 5G

Блок-схема GNN-декодера LDPC (рис. 3.7) побудована відповідно до логіки обробки логарифмічних відношень правдоподібності (LLR) за допомогою графової нейронної мережі. На вході декодера знаходиться блок LLR Input, який приймає вектор логарифмічних відношень із демодулятора. Далі ці значення передаються до модуля GNN Core, що виконує основну обробку на основі графової структури LDPC-коду. Усередині GNN Core послідовно працюють три основні підблоки: Graph Embedding, який формує початкові векторні представлення для вузлів графа; Message Passing, що реалізує механізм поширення повідомлень між вузлами протягом заданої кількості ітерацій (наприклад,  $T=5$  як використано в дослідженні) та Output Layer, який генерує кінцеві логіти або оцінки для кожного біта. Результат Output Layer передається до блоку Decision, де на основі порогового прийняття формується декодований вихідний вектор  $\hat{u}_i$ . Таким чином, рішення формується виключно після повного проходження даних через усі етапи графової обробки.

В роцесі розробки декодера проведена симуляція його роботи із представленням резйльтатів графом Таннера для CNN інтелектуального декодера 5G (рис. 3.8 а-д) для вказаних на рис. 3.7 5-и ітерацій.

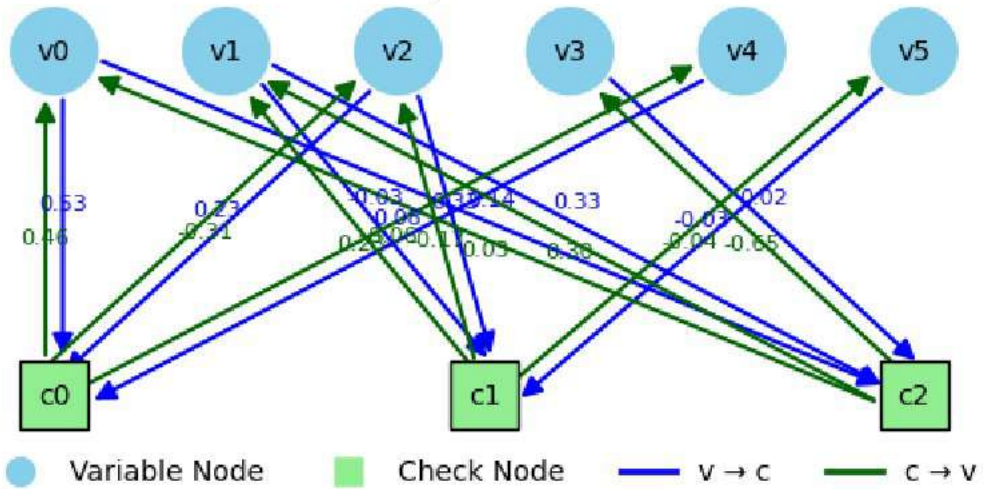
На рис. 3.8 а, б, в, г і д відповідно представлено п'ять послідовних ітерацій процесу декодування LDPC-коду із застосуванням інтелектуального підходу до обміну повідомленнями на основі графових нейронних мереж (GNN). Для цього було синтезовано біпартітний граф Таннера, що відображає зв'язки між вузлами змінних  $v_i$  та перевірочними вузлами  $c_i$  згідно з  $\mathbf{H}$ -матрицею обраного LDPC-коду.

Ітерація 0



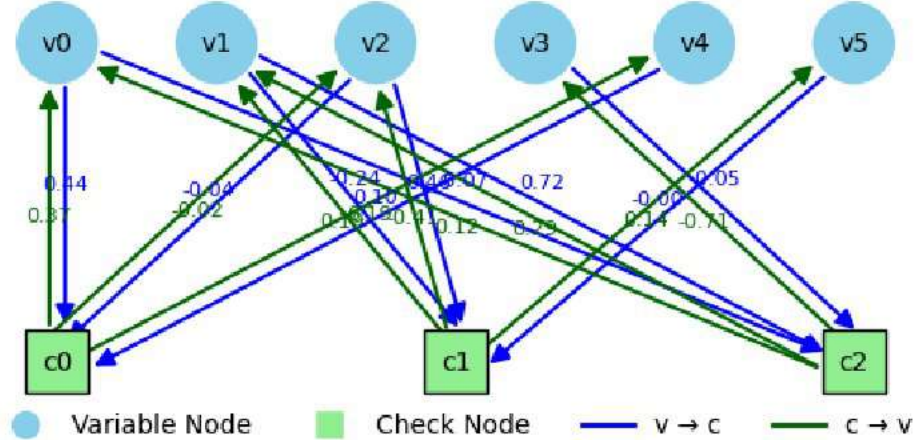
a

Ітерація 1

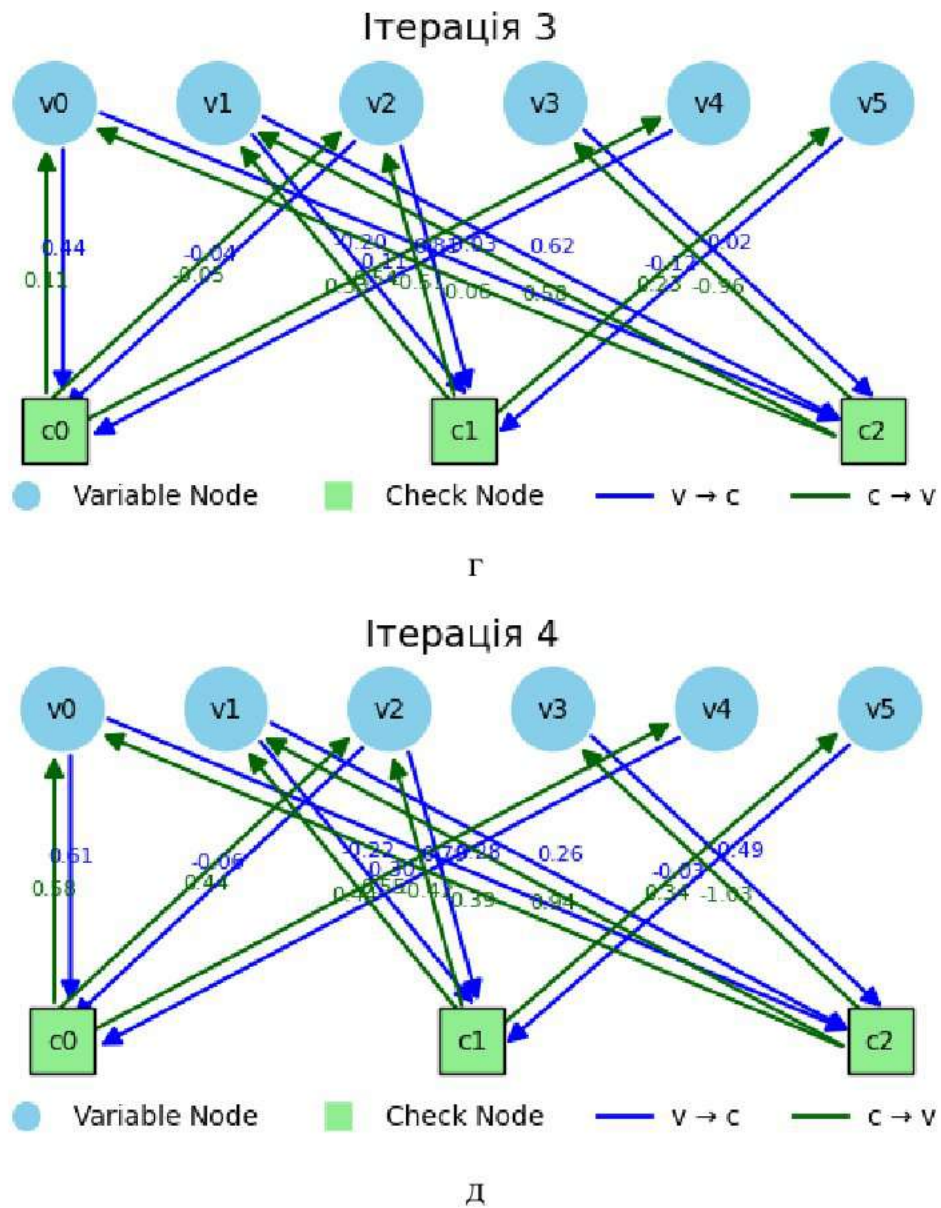


б

Ітерація 2



в



Риснок 3.8 - Візуалізація процесу ітеративного декодування LDPC-коду за допомогою GNN-подібного підходу

В дипломній роботі використано  $\mathbf{H}$ -матрицю розміром  $3 \times 6$  яка задає топологію графа Таннера для LDPC-коду. Вона мала вигляд:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.29)$$

Отже, синтезована  $\mathbf{H}$ -матриця визначає зв'язки між вузлами змінних ( $v_0 - v_5$ ) та перевірочними вузлами ( $c_0 - c_2$ ) у біпартитному графі Таннера, що

візуалізується у вигляді спрямованого графа з обміном повідомленнями в обидва боки. Причому рядки — це перевірочні вузли  $c_0 - c_5$  тоді як це вузли змінних  $v_0 - v_5$ . Відповідно елемент  $\mathbf{H}=1$  вказує на наявність ребра між вузлами графа Тенера.

Тому на рис. 3.8 а показана 0-а ітерація, як початкове передавання повідомлень, і далі 1-4 – у вигляді оновлення LLR. Зміст такого дослідження полягав в тому щоб показати стабілізації коливань повідомлення в межах 5-и ітерацій. Як видно з рисунків, уже після 3–5 ітерацій значення повідомлень починають стабілізуватись, що відповідає типовій поведінці LDPC-декодера в умовах з помірним рівнем гаусівського шуму  $N(0, \sigma^2)$  і  $\sigma = 0,2$  - стандартне відхилення шуму, тобто це означало, що при кожному оновленні повідомлення змінюються за випадковим законом із середнім 0-м і розсіянням 0.2. Вот таке значення шуму - що відповідає помірному рівню шуму в каналі дозволило моделювати адаптивний процес уточнення ймовірностей під час ітеративного декодування, забезпечуючи збіжність алгоритму за кілька ітерацій. Тепер повернімось до рис. 3.8. Синім кольором показана передача повідомлень від змінних до перевірок ( $v \rightarrow c$ ) тоді як зелені стрілки відображають зворотне повідомлення від перевірок до змінних  $c \rightarrow v$ . Повідомлення представлені як імітовані LLR, які змінюються на кожній ітерації. У реальній системі ці значення обчислюються згідно з алгоритмами типу BP або MS які ми описали в 1-шому і 2-у розділах дипломної роботи. Таким чином отримуємо, що імітація передавання повідомлень у графі базується на стохастичному оновленні значень, а саме передача від  $v \rightarrow c$  та зворотна передача. Візуально показано, як змінюються повідомлення на ребрах графа. Це дозволяє простежити процес збіжності або змін у впевненості вузлів. Як результат, таке представлення може бути використано для аналізу ефективності повідомлень, перевірки коректності роботи GNN-декодера або демонстрації ефекту глибини мережі (кількості ітерацій). Як видно з рисунків,

уже після 3–5 ітерацій значення повідомлень починають стабілізуватись (тобто значення повідомлень (цифри біля стрілок) стають ближчими до попередніх і не змінюються радикально). Ми досягли ознаки, що досягнуто узгодження між вузлами. Корисно, що немає хаотичних чи протилежних повідомлень дуже часто.

### **3.2.4 Використання концепції Deep Reinforcement Learning (DRL) – обґрунтування вибору та методика конструювання декодерів (LDPC, Polar codes) для 5G/6G**

DRL є потужним підходом до навчання агентів, які приймають рішення у складних динамічних середовищах [46]. У контексті декодування завадостійких кодів, які є об'єктом дослідження в магістерській роботі, а саме (LDPC, Polar) для 5G/6G, DRL дозволяє створити інтелектуальні декодери, здатні адаптивно обирати оптимальні стратегії декодування в умовах завад, варіативної якості каналу та змінного рівня шуму.

Розглянемо методику застосування концепції DRL. Загалом DRL поєднує дві компоненти:

- 1) Reinforcement Learning (RL) — навчання шляхом взаємодії агента з середовищем та отримання винагород;
- 2) Deep Learning (DL) — використання нейронних мереж (звичайно використовуються розглянуті вище в магістерській роботі мережі типу DNN, CNN або LSTM) для апроксимації функцій значущості (value function, policy function).

Так як концепція містить певні термінологічні нововведення потрібно сформулювати адекватний порядок тлумачення термінів прив'язаний до задачі синтезу завадостійких кодів для мереж 5G. В цьому контексті, для задачі декодування використаємо термінологію де агент це власне декодер, який обирає дії на основі LLR або часткових рішень; середовище це канал передачі

з шумом та кодована послідовність; стан (state) це поточна інформація про LLR, рішення по бітах, метрики шляху; під дією будемо розуміти – action, дію яка передбачає вибір біта  $\hat{u}_i \in \{0,1\}$  або зміну глибини декодування; нагорода (reward) буде визначатись успішністю декодування (в контексті рзробки декодера це вдале проходження CRC, або зменшення BLER).

Метою агента є максимізація очікуваної суми винагород в наступному математичному базисі:

$$\max_{\pi} \mathbb{Z} \left[ \sum_{t=0}^T \gamma^t r_t \right] \quad (3.30)$$

де  $\pi$  - параметр який буде характеризувати стратегію (policy), яку реалізує нейромережа;  $r_t$  - миттєва нагорода;  $\gamma \in [0,1]$  - коефіцієнт дисконтування;  $T$  – так звана довжина епізоду (для нашого випадку - кількість кроків у декодуванні).

Можливі архітектури для реалізації концепції зосереджені на стратегіях DQN (Deep Q-Network) для дискретних дій; Actor-Critic або DDPG для неперервних стратегій та Policy Gradient для прямої оптимізації політики.

Ми використаємо описаний вище базис сформуємо принципи і специфіка застосування його до LDPC/Polar кодів які використовуються в дипломній роботі під час інтелектуального декодування.

І так, для Polar codes, DRL може навчатись на графі SC/SCL декодування та приймати рішення щодо гілок шляху. Для LDPC codes - агент приймає рішення про корекцію бітів за результатами повідомлень на графі Таннера (дивитись перший розділ магістерської). Крім того DRL дозволяє уникати фіксованих правил (наприклад, лише застосування виключно алгоритмів MS або BP) і адаптуватися до реального каналу в процесі роботи. Далі, що особливо корисно для задач які мають бути вирішені в дипломній роботі, агент здатен враховувати пам'ять каналу (для Rayleigh fading або ISI у випадку

OFDM [48], наприклад) — що є надзвичайно актуально для 5G/6G URLLC або IoT [33].

Загалом можна виділити потенційні переваги такого DRL підходу при синтезі декодера який буде проведено нижче в дипломній роботі. Поперше, реалізується адаптивність - тобто декодер підлаштовується до зміни умов безперервно. Подруге, реалізується низька потреба у людському налаштуванні - тобто параметри декодування не задаються вручну, а навчаються. Крім того, виникає можливість інтеграції з LSTM/RNN мережами (ми їх розглянули вище в дипломній роботі) для врахування каналів з пам'яттю. Так і потрібно вже тут окреслити потенційне покращення BLER/BER у складних умовах передачі, яке потрібно буде довести під час експериментальних результатів які будуть подані в 4-у розділі кваліфікаційної роботи.

Отже, синтез схеми декодера DRL проведемо на основі використання згаданої вище концепції концепції і окреслених переваг які мають бути досягнуті. На основі таких положень DRL декодер має включати наступні блоки: LLR Input який описує вхідні LLR значення після демодуляції, State Encoder (CNN/LSTM) який кодує поточний стан; Policy Network (DRL Agent) який повинен приймати рішення на основі стану, Reward Evaluation – блок який власне обчислює нагороду, Action Executor (Bit Decision) блок виконує рішення щодо бітів (гілок), ну і на виході блок - Decoded Bits, який виводить рішення щодо декодування у форматі  $\hat{u}_0^{N-1}$ .

На рис. 3.9 представлена блок-схема синтезованого декодера на основі концепції DRL.

По схемі, вхідні логарифмічні відношення правдоподібності з демодулятора з блоку LLR Input надходять до блоку State Encoder (CNN / LSTM) де перетворюються у компактний описовий стан, далі блок Policy Network (DRL Agent) на основі стану формує дію — яке значення присвоїти поточному біту, блок Action Executor (Bit Decision) виконує власне цю дію

тобто визначає значення біту, на виході – блок Decoded Bits Output – формується рішення (тут відбувається обчислення якості розпізнаного коду (наприклад, чи правильна CRC, чи знижено BER)).

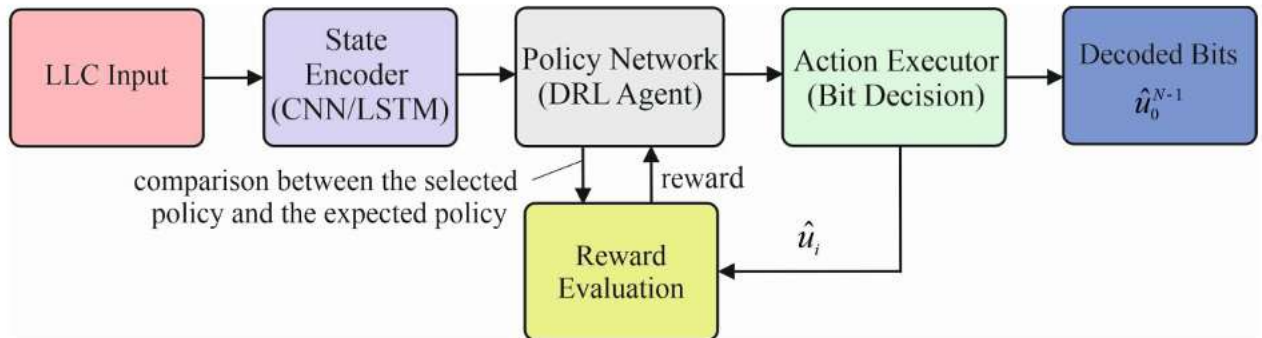


Рисунок 3.9 - Блок-схема синтезованого декодера на основі концепції DRL

Тепер торкнемось деталей DRL. Блок Reward Evaluation, прймає як слідує з схеми, обране значення біту  $\hat{u}_i$  або проміжкове рішення від блоку Action Executor. Крім того, якщо повернутись до блоку Policy Network (DRL Agent) то згідно до стрілки від Reward Evaluation в цей блок використовується для порівняння вибраної політики з очікуваною. В зворотний бік (дивитись по стрілці яка підписана reward) до Policy Network порополяє сигнал "нагороди" (reward), який оновлює політику агента через зворотне поширення (reinforcement learning update). В цьому власне і суть концепції зворотного поширення. Треба узгодити, інтелектуально, через зворотній зв'язок відповідну адаптивну політику щодо методу декодування. Таким чином на блок Reward Evaluation потрапляє обране значення біта з блоку Action Executor ( $\hat{u}_i$ ), актуальна послідовність рішень, з якої можна розрахувати CRC, BLER, або інший сигнал якості ну і сформована поточна політика або індекс дії. Стосовно агента якого ми визначили в якості концептуального чинника DRL, стрілка від Policy Network до Reward Evaluation передає дію, яку агент планував виконати або ймовірності/логіти (поширюється дія щодо розподілу політики). Саме такий формат, як пропонується в дипломній роботі, дозволяє

Reward-блоку оцінювати не тільки виконане рішення, а й інформацію про "наміри" агента, наприклад для policy gradient або actor-critic. Ключевим блоком як видно з методики при застосування концепції DRL виступає Reward-блок який є блоком винагороди. Це компонент у запропонованій моделі декодера з підкріпленням (reinforcement learning) (рис. 3.9), який визначає "наскільки добре" агент діє, тобто видає числову оцінку після кожної дії агента. Тобто іншими словами Reward-блок оцінює не лише результат дії, яку агент справді виконав (наприклад, "перейшов у стан B"), а й інформацію про "наміри" агента. Для нашого дослідження щодо декодування, будемо розуміти під "намірами", що мається на увазі розподіл імовірностей, з яким агент планував обрати дію. Так в дипломній роботі розробляється інтелектуальна концепція декодування то у policy gradient (блок Policy Network) навчання базується на градієнті всієї політики — тобто враховуються всі можливі дії, навіть якщо вони не були обрані. Тоді як Reward враховує наступне: "Агента більше цікавила дія  $X$ , але він вибрав  $Y$  — наскільки це узгоджується з цілями?". Крім того застосування механізму actor-critic тут в цьому блоці ґрунтується на компонентному розбоділі обов'язків механізму в наступному базисі, а саме actor - формує наміри, політику тоді як critic - оцінює наслідки дій тобто функцію цінності.

Actor - це мережа, яка на вхід отримує стан середовища  $s_t$  і видає розподіл імовірностей для дій (дивитись формулу 3.30)  $\pi(a_t|s_t)$  - "наміри" агента тобто не одна дія, а всі можливі з відповідними вагами. Critic представляє собою іншу (або частково спільну) мережу, яка на вхід також отримує  $s_t$  і повертає оцінку цінності  $V(s_t)$  або  $Q(s_t, a_t)$ .

Якщо зпроециувати такий загальний підхід DRL на запроповану схему інтелектуального декодера то  $s_t$  - буде характеризувати стан середовища в момент  $t$  (поточний стан декодування, LLR-вектори, історія

рішень, залишкові синдроми, метрики шляху тощо тобто те що описано в 1-у і 2-у розділі дипломної роботи),  $a_t$  - описує дію агента в момент  $t$  (відбувається в схемі на рис. 3.9 вибір конкретного значення біта  $\hat{u}_t \in \{0,1\}$  або перехід на певний шлях у графі, або стратегія перебору типу SC-Flip),  $\pi(a_t|s_t)$  - визначає політику в сенсі існує імовірність вибрати дію  $a_t$ , знаючи стан  $s_t$ ,  $V(s_t)$  виконує оцінку “цінності” стану (наскільки поточна конфігурація LLR/синдромів/метрик обіцяє правильне декодування),  $Q(s_t, a_t)$  описує очікувану нагорода за дію  $a_t$  у стані  $s_t$  (тобто практично оцінюється якість конкретного рішення (для перевірки використовується CRC, синдромна перевірка або loss-функція).

Нижче (рис.3.10) представлено запропоновану схему взаємодії конфігурантів базису Actor-Critic в контексті дослідної конструкції інтелектуального декодера 5G за технологією DRL.

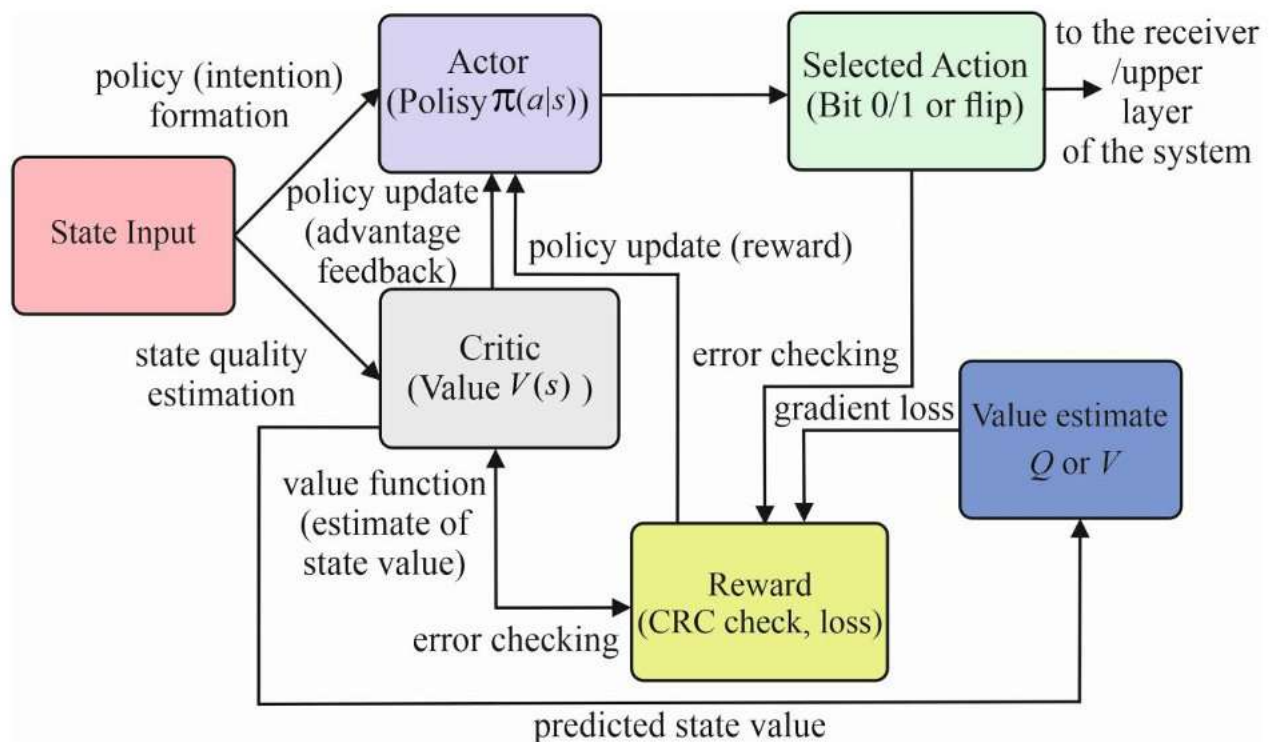


Рисунок 3.10 – Actor-Critic модель при використанні підходу з підкріпленням (DRL) для інтелектуального декодера 5G/6G

Actor відповідає за прийняття ймовірнісних рішень, тобто вказує на напрям вибору певного біта або стратегії. Critic – це механізм, який оцінює ці дії в контексті декодування, підсилюючи хороші рішення і пригнічуючи хибні.

На схемі, State Input - це вхід який містить LLR-вектори, поточні значення синдромів або інші ознаки (характеристики стану декодування з телекомунікаційного каналу), у блоці Actor формується розподіл імовірностей дій  $\pi(a_t|s_t)$  (визначається яку ймовірність обрати 0 або 1 на поточному бітові), блок Critic оцінює стан (або стан + дію) через функцію  $V(s_t)$  та  $Q(s_t, a_t)$ , блок Reward повертає значення винагороди (наприклад, 0 або 1, якщо CRC не пройдено/пройдено), або якусь похідну від функції втрат. На основі блоку Value Estimate виконується домоміжна оцінка яка використовується для навчання (оновлення wag) Actor+ Critic. Тобото при декодування LDPC і поярних кодів, відбувається надходження LLR з LDPC або Polar-графу на State Input далі Actor на основі цього обирає дію  $a_t$  - тобто рішення щодо біта  $\hat{u}_t \in \{0,1\}$  або вибір конфігурації повідомлення, алі Critic оцінює корисність цього рішення, тобто  $V(s_t)$  та  $Q(s_t, a_t)$ , а Value Estimate допомагає порівняти передбачене і реальне (reward), щоб скоригувати політику і тоді Reward - формується на основі правильності/помилки (наприклад, на основі CRC, BLER, чи хибного декодування) і допомагає покращити Actor. Тобто у Actor приймається рішення про декодування (який біт вибрати).

Для функцій  $V(s_t)$  та  $Q(s_t, a_t)$  в рамках нашого дослідження (синтез інтелектуального декодера) запишемо наступні математичні вирази. Для функції цінності стану  $V(s_t)$  отримаємо:

$$V(s_t) = \mathbb{Z}_\pi \left[ \sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k} | s_t \right] \quad (3.31)$$

де  $s_t$  - поточний стан графа декодування (LLR-вектори, попередні рішення тощо),  $r_{t+k}$  - миттєва винагорода (reward) на кроці  $t+k$  (тобто якщо

$r = +1$  - біт декодовано вірно, якщо  $r = -1$  - помилково),  $\gamma \in [0,1]$  - коефіцієнт знецінення (discount factor),  $\mathbb{Z}_\pi$  - позначення математичного очікування тобто середнього значення деякої величини, враховуючи всі можливі варіанти її реалізації,  $\pi$  - зальне для DRL позначення політики, тобто як агент приймає рішення у кожному стані (середньо очікувана сумарна винагорода, яку отримає агент (декодер), якщо буде діяти відповідно до політики  $\pi$  починаючи з поточного стану  $s_t$ ).

$V(s_t)$  це очікувана сумарна винагорода, яку отримає агент, починаючи з поточного стану  $s_t$  і діючи згідно з політикою  $\pi$ . Отже відповідає в нашій системі декодера за загальну оцінку надійності графа декодування на поточному етапі. Це важливо, бо в реальному середовищі результат дії не завжди детермінований — він може бути стохастичним, особливо в умовах шуму каналу або декодування під впливом завад.

Функція цінності дії (Action-Value Function)  $Q(s_t, a_t)$  представляється нами як:

$$Q(s_t, a_t) = \mathbb{Z}_\pi \left[ \sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k} \mid s_t, a_t \right] \quad (3.32)$$

де  $a_t$  - дія, яку пропонує Actor, тобто вибір біта  $\hat{u}_t = 0$  або  $\hat{u}_t = 1$ ,  $Q(s_t, a_t)$  це функція яка вказує, наскільки "добре" буде обрати цю дію у конкретному стані (дозволяє оцінити яка з дій має кращу довгострокову перспективу (тобто передбачає правильність рішення, що впливає на стабільність вихідних даних декодера, зокрема з точки зору мінімізації BLER).

Отже, блок Reward має враховувати не лише дію, а й імовірність, з якою вона була обрана, щоб навчити політику робити кращі вибори надалі. Щоб підкреслити унікальність запропонованих рішень щодо інтелектуального декодування, ще раз акцентуємо що в запропонованій схемі реалізується механізм коли Reward-блок не обмежується оцінкою лише фактично

виконаних дій, а бере до уваги намір агента — тобто яку ймовірність він приписував кожній дії. Це критично важливо для методів на основі політик (policy-based RL), а саме policy gradient або actor-critic. Зауважимо, що з практичної точки зору передбачається, що "наміри агента" — це політика, яка реалізується у вигляді нейронної мережі. Залежно від платформи або задачі, ця нейромережа (actor або critic) може бути реалізована на FPGA [25] — це особливо актуально для задач реального часу або енергоефективного виконання. Звісно актуальним є напрямок реалізації концепції actor-critic нейромереж як окремої цифрової мікросхеми — це шлях, який перспективний у комерційних чіпах для робототехніки, 5G/6G [37] та embedded (вбудовані) AI.

### **3.3 Аналіз нейромережевих підходів до побудови декодерів**

Перед розробкою архітектури пропонованого декодера проведемо аналіз нейромережевих підходів до їх побудови. Сучасні дослідження демонструють значний інтерес до використання нейромережевих архітектур у задачах декодування завадостійких кодів, таких як LDPC та полярні коди. Таким чином, доцільно узагальнити такі ключові підходи які вже опубліковані в науковій літературі та реалізовані в open-source середовищах, з метою виділити ефективні стратегії, а також обмеження таких рішень.

Перші нейромережеві моделювання декодера як детермінованого блоку полягали в заміні традиційних декодерів (наприклад, BP) на DNN, які навчаються відображати вхідні LLR на оцінки бітів. Такі DNN-декодери зазвичай мають повнозв'язну або згорткову структуру. Основна ідея полягала в реалізації процесу тренування мережі з метою узагальнення ймовірнісних залежностей між бітами, що закладені в код. Прикладом є DeepTurbo, де структура Turbo-декодера реалізована за допомогою кількох каскадів DNN [49, 50]. До основних переваг такого підходу можна віднести простота

архітектури, можливість паралельної реалізації. Недоліки пов'язані з поганою масштабованістю при зростанні довжини коду, складністю забезпечення інваріантності до структури коду.

Використання GNN для LDPC є природним інструментом для обробки графів факторизації, які лежать в основі LDPC-кодів. У цьому підході вузли змінних та перевірки в графі розглядаються як об'єкти, що обмінюються повідомленнями, аналогічно алгоритму BP який ми описали в попередніх розділах магістерської роботи. GNN автоматично навчається параметрам передачі повідомлень, що дозволяє враховувати апіорну статистику каналу.

Окремі реалізації, такі як Neural Belief Propagation, демонструють покращення продуктивності навіть при фіксованій кількості ітерацій [51-54].

У випадках, коли канал має частотну залежність (наприклад, ISI або канали з розсіянням), рекурентні нейронні мережі - RNN, зокрема LSTM, дозволяють накопичувати інформацію про попередні символи, що покращує декодування. Ці архітектури ефективно компенсують міжсимвольні інтерференції, особливо в умовах змінного каналу [55, 56].

Інтелектуальні підходи з використанням DRL змінюють парадигму декодування, тут агент навчається обирати дії (наприклад фліп бітів, зміна шляху декодування) на основі нагороди, що враховує правильність розшифрованого повідомлення. DRL підходи, такі як actor-critic або Q-learning, дозволяють створити адаптивні декодери, здатні підлаштовуватись до змін умов каналу або структури коду [57, 58].

Аналіз існуючих підходів дозволяє зробити висновок, що використання DNN, GNN та DRL вже довело свою ефективність у різних сценаріях, і може бути використано для створення адаптивних, стійких до завад декодерів у майбутніх бездротових мережах. Це обґрунтовує вибір архітектури, запропонованої в наступному підрозділі.

Крім того, на основі аналізу наявних реалізацій інтелектуальних декодерів доцільно виокремити систематизовані переваги та недоліки кожного підходу з метою їх урахування при проєктуванні інтелектуального декодера 5G у межах цієї дипломної роботи. Узагальнення, сформоване на основі літературних джерел, наведено в таблиці 3.3.

Таблиця 3.3 – Переваги та обмеження різних підходів до побудови інтелектуального декодера для 5G

<b>Підхід</b>	<b>Переваги</b>	<b>Обмеження</b>
<b>DNN</b>	Простота, швидкість	Погано масштабується
<b>GNN</b>	Врахування структури LDPC	Вища складність
<b>RNN/LSTM</b>	Підтримка каналів з пам'яттю	Потреба в навчанні на послідовностях
<b>DRL</b>	Адаптивність, стратегічність	Висока складність та час навчання

Як випливає з табл. 3.3, кожен із розглянутих нейромережевих підходів має власні переваги, що роблять його привабливим для побудови інтелектуального декодера, а також певні обмеження, які слід враховувати під час проєктування цілісної архітектури. Зокрема, DNN-моделі демонструють високу швидкодію при реалізації в inference-режимі завдяки простоті обчислювальної структури, однак погано масштабуються при збільшенні довжини кодів або при зміні типу шуму. Це обмежує їхню адаптивність у динамічних умовах мережі 5G/6G. Стосовно GNN - така мережа дозволяє безпосередньо враховувати топологію графа LDPC, причому зберігаючи логіку обміну повідомленнями між перевіірочними та змінними вузлами. Проте її апаратна реалізація значно складніша, особливо при обробці в реальному часі. Якщо оцінити RNN/LSTM-мережі то потрібно наголосити на тому, що вони ефективні саме в каналах із пам'яттю, де поточне рішення

залежить від історії вхідних сигналів. Однак потреба у навчанні на довгих послідовностях створює складнощі при генералізації та перенавчанні. Ті обставини необхідно врахувати при синтезі схеми декодера 5G/6G. Накінець DRL-концепція, яку ми описали в попередньому підрозділі дипломної роботи, якісно виділяється своєю адаптивною природою. Тут агент може змінювати стратегію декодування в залежності від середовища, сигналу та типу шуму. Але врахуємо, що ця гнучкість супроводжується високими витратами обчислювальних ресурсів на етапі навчання, а також потребою у спеціальному проектуванні функцій винагороди (reward function) та політик (дивитись рис. 3.9-3.10).

З огляду на зазначене, жоден із підходів не є універсальним у контексті 5G — кожен з них вирішує окремі аспекти проблеми декодування. Це відкриває шлях до гібридних архітектур, які об'єднують переваги кількох підходів. Саме тому в наступному розділі 3.4 буде представлено архітектуру пропонованої системи інтелектуального декодування, яка поєднує рішення з DRL-компонентами для стратегічного навчання та адаптації політики декодування та Soft-Decision (м'яке) декодування, як класичну основу для обчислення ймовірностей (LLR), що подаються в агентну мережу.

Саме така комбінація дозволить отримати адаптивне, завадостійке та практично реалізоване рішення в умовах змінного середовища 5G/6G.

### **3.4 Архітектура пропонованої системи гібридного інтелектуального декодування (DRL+Soft-Decision декодер) для 5G/6G**

У цьому підрозділі дипломної роботи пропонується архітектура гібридного інтелектуального декодера для LDPC та полярних кодів, що поєднує класичний Soft-Decision підхід та навчання з підкріпленням (DRL) на основі методики описаної в підрозділі 3.3. Основна ідея полягає в тому, щоб використовувати DRL-агента як стратегічного модуля прийняття рішень, який

адаптивно підлаштовується під характеристики каналу, тип коду та рівень завад. Це дозволяє значно покращити продуктивність декодування, особливо в умовах високої динамічності (5G/6G).

На рис. 3.11 представлено синтезовану запропоновану схему гібридної архітектури інтелектуального декодера (DRL Soft-Decision) для 5G/6G. Блок-схему архітектури гібридного запропонованого декодера формують блоки LLR Input який відображає модуль, що отримує soft-вимірювання з каналу, Feature Extractor – модуль попередньої обробки, який формує вектор стану  $s_t$ , далі слідуєть блоки Policy Network (Actor-Critic, DRL) у яких блок Actor призначено для генерації розподілу ймовірностей для можливих дій (зокрема, фліп, фіксація біта тощо), а також блок Critic який оцінює цінність стану або стану-дії в контексті параметрів  $V(s_t)$  та  $Q(s_t, a_t)$ . До складу декодера входить Soft-Decision Decoder (LDPC/Polar) – класичний модуль декодування з м'якими рішеннями (використовуються описані у попередніх розділах магістерської роботи алгоритми BP для LDPC та CA-SCL для Polar-кода). У блоці Reward Evaluator відбувається порівняння виходу декодера з CRC або з відомою метою (для тренування агента).

Тепер сформуємо математичний опис моделі гібридного декодера. Нехай  $s_t$  стан системи у момент часу  $t$ , який описується вектором ознак, отриманих із LLR. Під дією  $a_t$  будемо розуміти вибір агента (наприклад, зміна деякого біта або параметра декодування). Цільову функцію декодера математично представимо наступним чином:

$$J(\theta) = \mathbb{E}_{\pi_\theta} [R_t] = \mathbb{E}_{\pi_\theta} \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k} \right] \quad (3.33)$$

де  $\theta$  параметри нейромережі агента (для запропонованої схеми декодера це ваги Actor і Critic в базисі Policy Network),  $\gamma \in [0,1)$  - коефіцієнт дисконтування (тобто коефіцієнт навчання який постує керує "швидкістю"

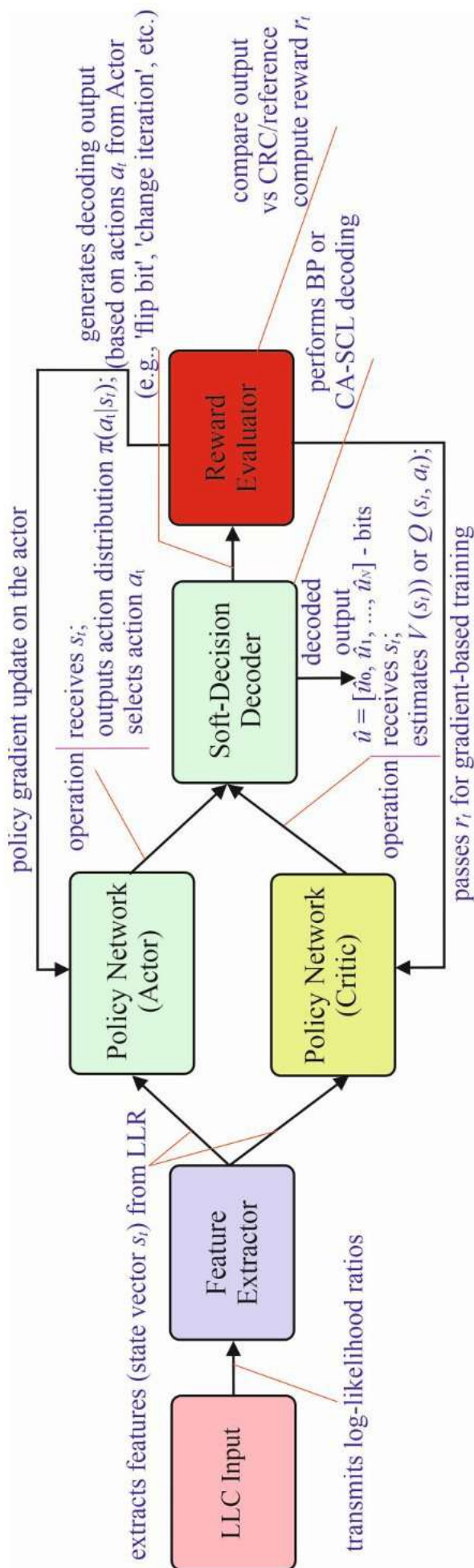


Рисунок 3.11 – Блок-схема гібридної архітектури інтелектуального декодера (DRL + Soft-Decision) для 5G/6G

оновлення параметрів  $\theta$ ),  $r_t$  - миттєва винагорода (для нашого випадку – декодування це зменшення кількості помилок).

Оновлення параметрів будемо здійснювати за допомогою алгоритму Actor-Critic, який математично представимо так:

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) (R_t - V(s_t)) \quad (3.34)$$

де  $\pi_{\theta}(a_t | s_t)$  - параметр який визначає політику агента тобто ймовірність вибору дії  $a_t$  у стані  $s_t$  (тобто в перенесенні на інтелектуальне декодування параметр визначає яка дія приймається для декодування - змінити біт, прийняти рішення тощо),  $\alpha$  - коефіцієнт навчання (learning rate) (в контексті запропонованої схеми декодера це множник що визначає, наскільки сильно оновлюються ваги. Практично це означає наступне - Якщо  $\alpha$  завелике — агент «перенавчається», якщо замале — повільно навчається),  $\nabla_{\theta}$  - градієнт по параметрах  $\theta$  (вектор похідних — показує, як змінити параметри, щоб збільшити результат, та вказує, в якому напрямку треба змінити ваги Actor),  $\log \pi_{\theta}(a_t | s_t)$  - називають логарифмом політики (імовірності вибору дії, а саме якщо агент часто вибирає добру дію, то її ймовірність зростає ще більше).

$$\theta \{W^{(l)}, b^{(l)}\}, \quad l=1, \dots, L \quad (3.35)$$

де  $W^{(l)}$  - матриця ваг (ваги зв'язків нейронів у внутрішній DRL-моделі, вони змінюються в результаті тренування, щоб максимально підвищити точність декодування або мінімізувати кількість помилок)  $l$ -го шару (регулює вплив кожного входу (ознаки з LLR або з Feature Extractor) на вихід),  $b^{(l)}$  - зсув (bias)  $l$ -го шару (дозволяє мережі зміщувати відповідь незалежно від входу),  $\theta_{\text{Actor}}$  - визначає, яку дію вибрати, коли бачимо стан  $s_t$ ,  $\theta_{\text{Critic}}$  - оцінює, наскільки цінний цей стан або дія у цьому стані.

У відповідності до формули 3.35 приклад нейронного шару в Actor буде мати наступну форму:

$$a_t = \text{soft max}(W^{(1)}s_t + b^{(1)}) \quad (3.36)$$

де  $s_t$  вектор ознак із Feature Extractor (дивитись схему рис. 3.11),  $W^{(1)}$ ,  $b^{(1)}$  - параметри  $\theta$ , які будуть оновлюватись градієнтом (дивитись формулу 3.34),  $a_t$  - розподіл імовірностей дій (для декодера це означає які біти виправити, які ітерації активувати тощо).

У запропонованому інтелектуальному декодері параметри  $\theta$  - це всі ваги та зсуви нейронних мереж, які керують прийняттям рішень. Вони навчаються під впливом сигналу винагороди так, щоб агент навчився ефективно декодувати, навіть у складних умовах завад, без жорстко заданих правил. Тобто параметри  $\theta$  представляють собою ваги та зсуви нейронних мереж, що входять до складу модуля прийняття рішень (Policy Network). Ці параметри визначають, як агент обробляє вхідний вектор стану  $s_t$ , сформований на основі LLR-значень, і приймає рішення щодо дій  $a_t$  (корекція бітів, вибір траєкторії декодування тощо). Під час навчання ці параметри автоматично оновлюються за допомогою градієнтних методів у напрямку, який підвищує очікувану довгострокову винагороду. Таким чином,  $\theta$  кодує адаптивну стратегію агента, що дозволяє декодеру ефективно пристосовуватись до змін у каналі зв'язку без потреби жорстко заданих правил.

Щодо політики то стратегічно вона працює так. Якщо політика  $\pi_\theta$  каже "У цьому стані зробити дію «А» з 80% і дію «В» з 20%" і дія «А» дала хорошу винагороду - базис прагне зробити її ще більш ймовірною в майбутньому. Власне саме тому у математичній моделі пропонується брати градієнт логарифма цієї політики (якщо дія була хороша, то це позитивний градієнт), далі множити на різницю між фактичною винагородою та очікуваною, тобто

отримати advantage, і нарешті, оновити ваги Actor, підсилюючи політику. З точки зору декодування така політика може тлумачитись так - оновлення дозволяє Actor ефективніше адаптуватися до умов каналу (коли зростає рівень шуму), робити більш стратегічні дії, не просто фіксуючи помилкові біти, а враховуючи довгострокову ефективність (через Reward), вчитись від успішних прикладів, а не просто запам'ятовувати правильні відповіді.

Тепер торкнемось новизни підходу який запропоновано до синтезу інтелектуального декодера для 5G/6G. Новизна запропонованого підходу полягає:

- 1) У поєднанні DRL з класичним soft-decision декодуванням, що дозволяє використовувати гнучкі адаптивні стратегії.
- 2) Система здатна підлаштовуватись до змін у каналі (швидке затухання, ISI, шум) без необхідності повного перенавчання.
- 3) У порівнянні з чисто нейромережевими декодерами, гібридна структура забезпечує кращу збіжність і пояснюваність.

З точки зору застосування до LDPC і Polar кодів:

- 1) Для LDPC агент DRL може коригувати порядок повідомлень у алгоритмі BP або керувати глибиною ітерацій.
- 2) Для Polar-кодів, агент може впливати на вибір серед списку в CA-SCL або змінювати ймовірнісні рішення для фліпів у Fast-SSC.

З точки зору застосування політики Actor-Critic яку ми використали в запропонованій схемі декодера:

- 1) Вирішено питання адаптації до середовища каналу так як DRL-агент навчається приймати рішення, які залежать від типу завад (шум, ISI тощо).
- 2) Реалізується механізм стратегічного навчання, тобто декодер не просто мінімізує локальну помилку, а будує політику дій, які призведуть до менше помилок у майбутньому.

3) Підтримується soft-декодування, що дозволяє агенту впливати на дії всередині класичного декодера (зміна LLR, вибір шляху).

4) Реалізується гнучка архітектура так як Actor і Critic можна реалізувати на різних типах нейромереж (DNN, GNN), і вона підходить для FPGA/ASIC (в контексті практичної реалізації).

Архітектура DRL + Soft-Decision Decoder є перспективним підходом до побудови інтелектуальних декодерів у мережах нового покоління (5G/6G). Гнучкість, адаптивність та підтримка різних кодів робить її конкурентоспроможною альтернативою класичним схемам. Фундаментальна мета інтелектуального декодера, що якісно його характеризує - знайти політику, яка мінімізує помилки в довгостроковій перспективі.

Особливість декодер полягає у запропонованій гібридній архітектурі, оскільки поєднує два концептуально різні підходи, а саме класичне Soft-Decision декодування (BP або CA-SCL), яке забезпечує точність завдяки апріорним алгоритмам роботи з LLR та інтелектуальне прийняття рішень на основі глибокого навчання з підкріпленням (DRL), яке дозволяє адаптивно змінювати стратегію декодування залежно від поточного стану системи.

Саме поєднання (гібридизація) евристичних та нейромережових компонентів дозволяє реалізувати систему, що не лише виконує декодування, але й навчається оптимізувати параметри декодування в режимі онлайн, підлаштовуючись до динамічних умов 5G/6G-каналів.

Наприкінці торкнемось актуальних для AI архітектур проблем перенавчання та стійкості. В завдання на магістерську роботу не входило питання аналізу цих аспектів роботи синтезованого декодера. Однак, зазначимо передбачається, що в запропонованій схемі що Agent DRL може використовувати регуляризацію або експлорейшн-стратегії (наприклад,  $\epsilon$ -greedy коли у RL враховується стратегія яка балансує дослідження (exploration) і використання (exploitation)) для уникнення перенавчання. У

гібридному декодері  $\epsilon$ -greedy може використовуватись на етапі тренування DRL-агента для вибору стратегії інвертування бітів або зміни значення біта на протилежне. Тобто агент DRL може прийняти рішення інвертувати біт у векторі LLR, якщо вважає, що це зменшить кількість помилок у декодуванні. З точки зору усталеності роботи декодера, все ж схема містить зворотні зв'язки для підвищення стабільності планується на практиці застосування технології обмеження ваг, нормалізація градієнтів та контроль параметрів  $\gamma$  та  $\alpha$  для стабілізації. Подальші пошуки в цьому напрямку будуть зосереджені на застосуванні концепцій meta-learning або transfer learning для перенавчання під різні типи каналів. Тобто агент вчиться не тільки декодувати, а й налаштовуватись на нові типи шумів, інтерференцій чи затримки, змінюючи свою політику з мінімальними оновленнями.

### 3.5 Висновки до третього розділу

У третьому розділі кваліфікаційної роботи було проведено системний аналіз сучасних інтелектуальних підходів до декодування завадостійких кодів, що використовуються у системах зв'язку нового покоління, зокрема 5G/6G.

Було обґрунтовано актуальність застосування методів штучного інтелекту, зважаючи на обмеження традиційних алгоритмів декодування у динамічних та зашумлених середовищах. Далі було досліджено архітектури нейронних мереж, які демонструють потенціал у задачах декодування, а саме DNN — для фіксованих умов та швидких рішень, RNN/LSTM — для каналів із пам'яттю (інтерференція, кореляції), GNN — для графових структур LDPC-кодів, DRL — як адаптивний стратегічний підхід з можливістю приймати рішення у реальному часі.

У підрозділі 3.3 проведено огляд базових реалізацій та рішень, на основі яких узагальнено переваги та обмеження кожного підходу, що дозволило аргументовано обрати гібридну архітектуру.

Нарешті, у підрозділі 3.4 розроблено архітектуру запропонованого гібридного інтелектуального декодера, що поєднує DRL як модуль адаптивного прийняття рішень та класичний Soft-Decision декодер (BP для LDPC і CA-SCL для Polar-кодів). Запропоновано формальну математичну постановку задачі, що базується на функції винагороди та механізмі навчання Actor-Critic.

У наступному, 4-у розділі, буде представлено експериментальну оцінку ефективності запропонованого декодера.

## 4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ІНТЕЛЕКТУАЛЬНОГО ДЕКОДУВАННЯ ЗАВАДОСТІЙКИХ КОДІВ ДЛЯ 5G/6G

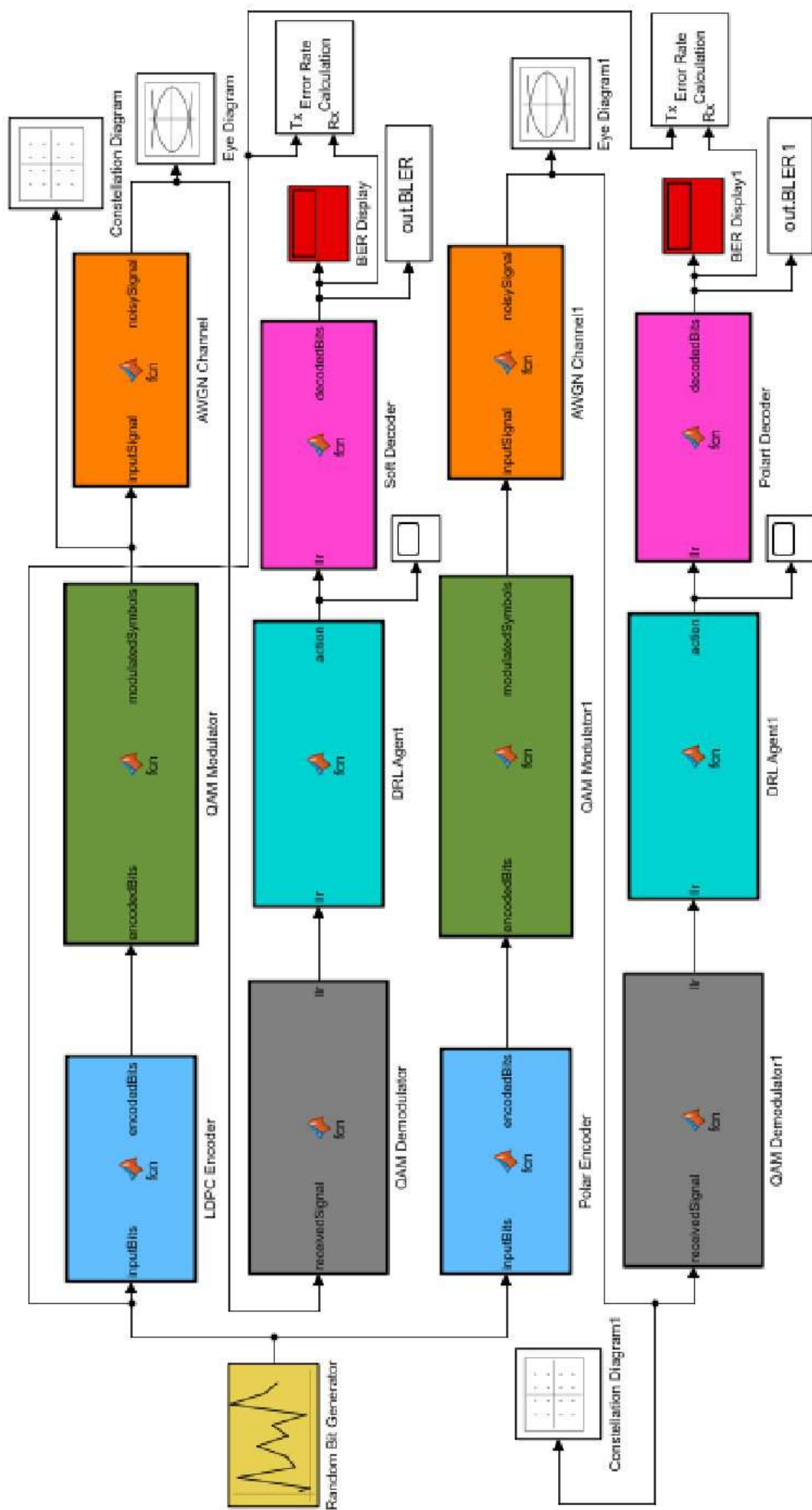
### 4.1 Побудова моделі телекомунікаційного каналу з інтелектуальним декодуванням завадостійких кодів для 5G

Основною метою цього етапу виконання дипломної роботи є створення симуляційної моделі телекомунікаційного каналу 5G у середовищі MATLAB/Simulink, яка дозволяє дослідити ефективність гібридного декодера (DRL+Soft-Decision) в задачах завадостійкого декодування. Модель імітує фізичний рівень зв'язку (PHY) відповідно до вимог стандарту 3GPP 38.212/38.213/38.214 [1, 2]. Зважаючи на складність повноцінного 5G PHY тобто фізичного каналу (ми опишемо цей розгорнутий формат каналу нижче у розділі), модель реалізує спрощену, але репрезентативну версію каналу. Вона включає механізми генерації даних, модуляцію (загалом,  $M$ -QAM), кодування (LDPC або Polar), вплив завад (AWGN, Rayleigh), демодуляцію з LLR, а також інтелектуальний декодер на основі алгоритму DRL (Actor-Critic) який описано в розділі 3 магістерської роботи, що дозволяє оцінити здатність системи адаптивно справлятися із завадами. Загалом, модель реалізовувалась за допомогою власних MATLAB Function блоків для реалізації DRL-агента і особливостей формування і обробки поточкових даних. Загальна структура імітаційної моделі представлена на рис. 4.1 у вигляді Simulink-схеми, де кожен компонент відповідає етапам обробки сигналу на PHY (фізичному) рівні.

Модель містить блок джерела даних Source у вигляді Random Bit Generator який генерує випадкову послідовність бітів, які слугують інформаційними даними для передачі. Для нашого випадку - це логічний вектор із значеннями 0 або 1, який імітує потік корисної інформації. До схеми входить кодер (Encoder) реалізований у вигляді Polar Encoder/LDPC Encoder, модулятор ( $M$ -QAM Modulator) де закодовані біти групуються у символи та

мапуються=відображуються (операція mapping) у комплексні точки на QAM-схемі модуляції. Це дозволяє ефективно передавати двійкову інформацію через телекомунікаційний канал, канал (Channel) – на схемі показаний AWGN, однак дослідження будуть проводитись і в умовах релеєвського типу каналу. В такому режимі як представлено на рис. 4.1 до модуляційних символів додається адитивний білий гаусівський шум, який моделює реальний радіоканал із випадковими завадами, що знижує якість сигналу. Схема також містить блоки демодуляторів (Demodulator), окремо для LDPC – кодів (Soft Decoder) та полярного коду – Polar Decoder. Загалом в цей блок приймає зашумлений сигнал і обчислює логарифмічні відношення ймовірностей (LLR) для кожного біту тобто визначає "м'яке" рішення, яке містить інформацію про ступінь впевненості в значенні біту. З отриманих LLR виконується полярне декодування за допомогою підпрограми `nrPolarDecode` (лістинг програм розміщено у додатках), що відновлює вихідні біти, використовуючи алгоритми SC або SCL для покращення точності декодування. У відповідності до запропонованих в магістерській роботі концепцій, описаних детально у 2 та 3-у розділах, до складу схеми входить DRL Agent (на основі механізму Actor-Critic, дивитись рис. 3.9-3.10 розділу 3). В запропонованій реалізації моделі інтелектуальний агент, реалізований у MATLAB Function блоці, приймає LLR або інші характеристики сигналу та ухвалює рішення (дії) щодо покращення декодування або параметрів системи на основі алгоритму глибокого навчання (у форматі actor-critic). Це дозволяє адаптувати систему під умови каналу в реальному часі. Закінчує схему блок BER Display+Reward Evaluation, який порівнює відновлені біти з оригінальними, обчислює рівень помилок (BER — або BLER) і формує сигнал винагороди (reward) для DRL агента, що дозволяє йому навчатися і покращувати свої рішення з часом.

Компоненти LDPC і Polar декодерів інтегровані до DRL-системи в 5G характеризуються високою адаптивністю та інтелектуальним управлінням



Риснок. 4.1 - Загальна структурна схема представлена моделі інтелектуального декодера для 5G

процесами кодування та декодування на фізичному рівні. Система підтримує різні схеми кодування — LDPC, який широко застосовується для великих транспортних блоків, та Polar-коди, оптимальні для захисту коротких контрольних повідомлень. Інтеграція механізмів підкріпленого навчання (DRL) забезпечує динамічну адаптацію параметрів кодування, модуляції та декодування залежно від поточних умов телекомунікаційного каналу.

На фізичному рівні модель починається з генерації випадкових бітів, що імітують потік інформації користувача. Ці біти надходять на блок кодування, де обирається відповідна схема — LDPC або Polar. LDPC-кодер реалізує розріджену матрицю перевірки парності, яка ефективно усуває помилки в умовах шумного та багатопроменевого каналу, тоді як Polar-кодер застосовує принцип поляризації каналів, що дозволяє досягти близьких до теоретичних меж продуктивності при меншій складності для коротких блоків. Після кодування послідовність бітів модулюється за допомогою  $M$ -QAM, що забезпечує баланс між пропускною здатністю і стійкістю до шумів. Сигнал проходить через модель каналу, що може імітувати AWGN канал або Rayleigh завмирання (релеєвське) — типові джерела спотворень у реальних бездротових електронних мережах комунікацій. На прийомній стороні виконується soft-демодуляція (м'яка), яка формує LLR для кожного біта, що є вхідними даними для soft-декодера LDPC або Polar. Ці алгоритми — BP для LDPC і CRC-Aided SCL для Polar — виконують ітеративне виправлення помилок, забезпечуючи високу точність відновлення даних.

Безперечно, ключовим елементом системи є DRL агент, що працює на основі архітектури Actor-Critic (рис. 3.10). Агент постійно отримує зворотний зв'язок від параметрів каналу та якості декодування (BER), і на основі цього навчається оптимізувати параметри системи такі як кількість ітерацій декодування, тип модуляції, рівень надлишковості кодування тощо. Таким чином реалізується інтелектуальна адаптація, що дозволяє мінімізувати

помилки та затримки в передачі даних. Фінальний етап включає оцінку якості відновленої інформації, вимірювання BER (BLER), та подання результатів у вигляді метрик продуктивності. DRL агент використовує цю інформацію як сигнал винагороди для подальшої оптимізації стратегії. Такий підхід забезпечує ефективну роботу системи навіть у складних, змінних умовах передачі інформації у каналі електронних комунікацій.

#### 4.2 Розгорнута архітектури фізичного каналу 5G

У 5G NR (New Radio) [1] фізичний канал PDSCH (Physical Downlink Shared Channel) є основним каналом для передачі користувацьких даних у напрямку від базової станції (gNB — next-generation Node B) до користувача (UE — User Equipment=обладнання користувача). Він побудований з урахуванням максимальної гнучкості та продуктивності, підтримує різноманітні моделі модуляції (від QPSK до 1024-QAM), багатоантенні системи (MIMO — Multiple Input Multiple Output) та сучасні схеми кодування, зокрема LDPC та полярні коди, інтелектуальне декодування яке досліджується в магістерській роботі. На рис. 3.2 представлена розгорнута схема фізичного каналу із запропонованими рішеннями щодо інтелектуального кодування та механізмами які супроводжують процеси передачі інформації в мережах електронних комунікацій 5G.

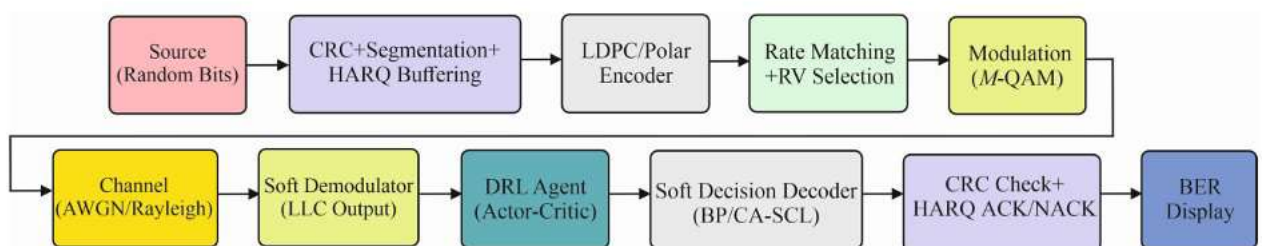


Рисунок 3.2 – Схема структурна архітектури фізичного каналу PDSCH у 5G

Перед передачею дані спочатку проходять через DL-SCH (Downlink Shared Channel=низхідний канал), який є логічним каналом і служить як джерело інформації для PDSCH. Якщо розмір транспортного блоку (Transport

Block, TB) перевищує певний поріг, дані поділяються на менші частини — кодові блоки (Code Blocks, CBs). Кожен CB отримує CRC довжиною 16 біт для перевірки цілісності після декодування. На рівні всього TB перед сегментацією додається 24-бітна CRC, що використовується для виявлення помилок після повного об'єднання блоків на приймальній стороні.

У 5G застосовуються два основні типи кодерів а саме LDPC який використовується для PDSCH, UL-SCH (Uplink Shared Channel=висхідний канал), а також більшості даних із великим розміром блоків. LDPC забезпечує високу швидкість декодування та хорошу продуктивність на великих об'ємах даних. Крім того використовують полярні коди і які застосовуються для коротких повідомлень в каналах керування, таких як PDCCH (Physical Downlink Control Channel=низхідний канал керування) і PBCH (Physical Broadcast Channel=фізичний ширококомовний канал, який призначено для одночасної передачі інформації всім абонентам). Полярні коди є оптимальними в сенсі каналної ємності при скінченних розмірах і ефективні при використанні алгоритмів декодування CA-SCL (алгоритми декодування розглянуто у 2-у розділі дипломної роботи).

Тепер токнемось механізмів які використовуються в 5G з метою забезпечення високої швидкості і широсмуговості. Особливість каналу 5G є наявність механізму підгонки швидкості (Rate Matching). Тобто наявна після кодування процедура Rate Matching — це процедура підгонки вихідного потоку бітів до розміру, який фізично може бути переданий через виділені ресурси. У LDPC це включає процедури виколювання (puncturing), повторення (repetition) та перемежування (interleaving) бітів в структурі кодової констріукції [59]. Також визначається версія надлишковості Redundancy Version (RV), яка вказує, які саме біти з LDPC коду будуть передані. Наперед акцентуємо, що це важливо для механізму HARQ (Hybrid Automatic Repeat Request=гібридного автоматичного запиту на повторення). Для Polar-

кодування RV складається з вибору інформаційних бітів, накладання CRC, додавання заморожених бітів (frozen bits) (процес поляризації телекомунікаційного каналу з поляним кодуванням і утворення заморожених бітів для 5G детально розглянутий в роботах [25, 26]) та перемежування. Як раз, згаданий вище механізм HARQ застосовується тут з метою забезпечення надійної передачі. По суті HARQ є гібридом автоматичного запиту на повторення передачі (ARQ) з механізмами корекції помилок. У PDSCH підтримується до 16 паралельних HARQ процесів, кожен із власним буфером (тобто тимчасовим сховищем для збереження отриманих пакетів (та їх копій) під час повторних передач). Кожна передача містить версію надлишковості - RV. Якщо приймач (UE) не може правильно декодувати дані (визначається за CRC або контрольною сумою), надсилається NACK (Negative Acknowledgement=негативне підтвердження) у зворотній (uplink) канал. Якщо прийом успішний - то надсилається ACK (Acknowledgement=позитивне підтвердження). У повторних передачах використовується Soft Combining=м'які комбінації коли зберігається інформація LLR з попередніх спроб для покращення шансів на успішне декодування.

Також потрібно згадати за опцію виявлення помилок. У 5G з цією метою застосовується додаток CRC (Cyclic Redundancy Check=перевірка циклічної надлишковості), який концептуально використовується для перевірки цілісності на обох рівнях (метод виявлення помилок у вигляді контрольної суми яка обчислена за допомогою полінома CRC, дивитись опис для формули 2.18 розділу 2 магістерської роботи). Страндарт описує такі формати: TB-CRC який містить 24 біти які додаються до TB і до сегментації, тоді як у свою чергу CB-CRC містить 16 бітів які додаються до кожного кодового блоку перед LDPC або Polar-кодуванням. На прийомі UE виконує перевірку CRC після декодування. Якщо виявлена помилка тоді ініціюється повторна передача (NACK + HARQ). Якщо CRC успішно проходить перевірку — UE відправляє

рішення у формі АСК. Важливим аспектом якісного функціонування 5G інформаційного каналу електронних комунікацій є забезпечення надійного спрацювання системи синхронізації та наявність опорних сигналів. З цією метою, для коректного прийому сигналу потрібно забезпечити синхронізацію за часом і частотою. У 5G це виконується за допомогою PSS/SSS (Primary/Secondary Synchronization Signals=первинного та вторинного сигналів синхронізації) на етапі початкового доступу (описано детально у роботі [17]). Для каналної оцінки та демодуляції використовуються DM-RS (Demodulation Reference Signals=опорні сигнали демодуляції), які вставляються в PDSCH за заданим шаблоном. DM-RS допомагають UE (юзеру тобто обладнанню користувача) оцінити стан каналу та адаптувати демодуляцію символів. Загалом, UE інформує базову станцію про результат декодування за допомогою PUCCH (Physical Uplink Control Channel=фізичного висхідного каналу керування) або PUSCH (Physical Uplink Shared Channel=фізичного низхідного каналу керування). Принцип взаємодії фізичних каналів у 5G (ієрархічна трирівнева модель) показаний та анонсований на рис. 1.7, розділ 1 дипломної роботи. Вибір каналу залежить від розкладу висхідних=uplink ресурсів. Якщо відповідь не отримана вчасно — ініціюється повторна передача з іншим рівнем RV.

Нарешті торкнемось ідеї яка є стуттю дипломної роботи. В роботі запропоновано використання інтелектуального механізму декодування. І тут маємо суттєве вдосконалення відносно описаного вище непростого (з точки зору забезпечення ключових показників функціонування каналу – швидкості, широкосмуговості тощо) механізму функціонування 5G. Пропонується застосування інтелектуальної моделі з підключенням DRL-агенту який може динамічно обирати наступні опції механізмів функціонування каналу 5G - RV значення залежно від умов каналу, кількість HARQ повторів, конфігурацію BP/CA-SCL (кількість ітерацій, вибір шляху – дивитись розділ 2 дипломної

роботи), адаптивну модуляцію та швидкість передачі. Отже, це дозволяє в режимі реального часу оптимізувати throughput (пропускну здатність каналу), зменшити latency (затримки) та уникнути зайвих retransmissions (повторних надсилань даних), зберігаючи якість зв'язку навіть у динамічних або зашумлених умовах.

Наприкінці цього підрозділу акцентуємо на тому, що глобальний механізм забезпечення якості функціонування 5G каналу електронних комунікацій є багатофакторним і містить низку концептуальних механізмів оцінка і вдосконалення яких виходять за рамки формату (завдання отриманого від керівника) магістерської роботи. Тому у рамках даного дослідження було побудовано структуровану модель каналу зв'язку 5G з підтримкою LDPC та Polar-кодування, а також включенням інтелектуального агента DRL для адаптивного декодування. Незважаючи на те, що в стандарті 3GPP (Release 15 і вище [1]) фізичний рівень (PHY) 5G NR визначається комплексною системою процедур — таких як HARQ, Rate Matching, CRC, TB Segmentation, (RV), а також є дієвою процедура ACK/NACK обміну — повна реалізація всіх аспектів у середовищі Simulink потребує надзвичайної деталізації, яка може ускладнити процес моделювання та аналізу результатів.

У зв'язку з цим, у роботі застосовано спрощену, але функціонально репрезентативну модель (рис. 4.1), яка охоплює ключові компоненти передавання сигналу на фізичному рівні від генерації інформаційної послідовності до оцінки бітової похибки (BER). Основні процедури, включаючи кодування LDPC/Polar, модуляцію, додавання шуму (AWGN/Rayleigh), демодуляцію, м'яке декодування, а також базову CRC-перевірку, реалізовано програмно (або у вигляді Simulink моделей, рис. 4.3), що відповідають загальній структурі PDSCH (рис. 4.2).

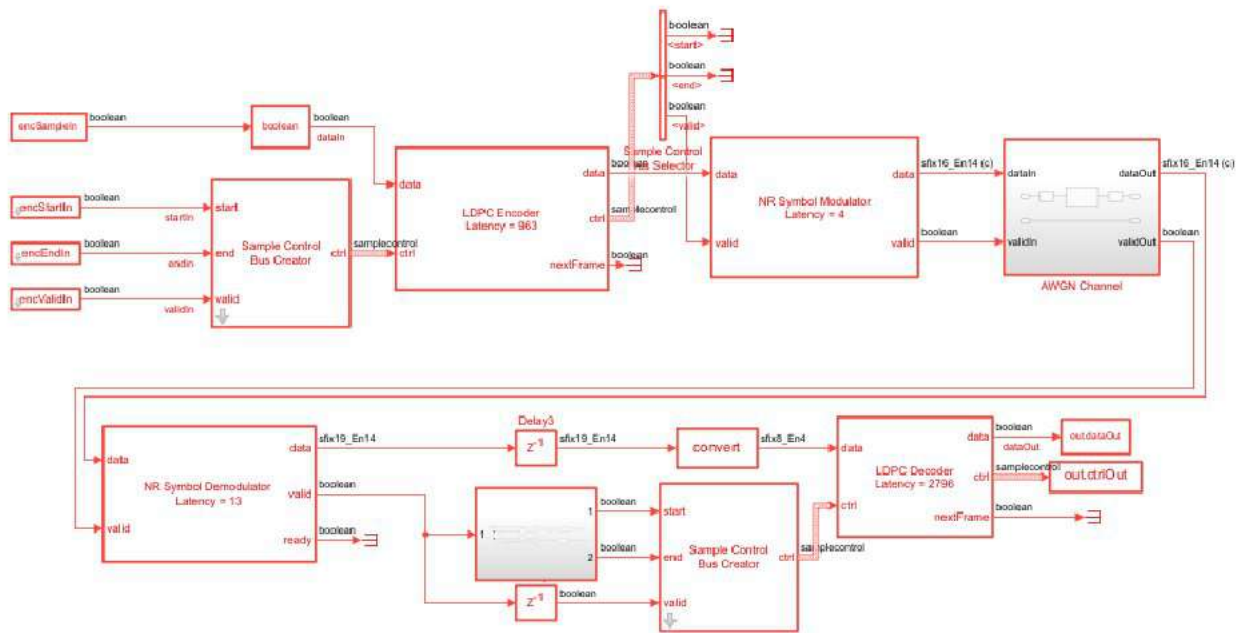


Рисунок 4.3 – Simulink-підсистема каналу електронних комунікацій з LDPC кодуванням

Деякі процедури, а саме HARQ з буферизацією, динамічне RV-кодування, перемежування та повне управління HARQ процесами, були абстраговані або узагальнені, з метою зосередження уваги на ключових елементах — зокрема на інтеграції DRL-механізму у процес адаптивного декодування. Це дозволило створити експериментальну платформу, здатну продемонструвати поведінку системи під різними умовами каналу при збереженні основної структури фізичного рівня, описаної в 3GPP.

Таке спрощення дозволило в дипломній роботі зберігати баланс між точністю й обчислювальною ефективністю, що особливо важливо для валідації концептуальних ідей — таких як використання DRL для оптимізації процесу декодування в каналах електронних комунікацій із завадами.

Таким чином, побудована модель є достатньо репрезентативною для проведення аналізу, експериментального порівняння LDPC та Polar кодів, а також оцінки ефективності навчання агента DRL в умовах змінного каналу, зберігаючи зв'язок із архітектурною структурою 5G NR на рівні фізичного каналу.

### **4.3 Оцінка продуктивності каналу 5G з інтелектуальним декодуванням за індикатором завадостікості BER від SNR (канали AWGN, Rayleigh)**

#### **4.3.1 Умови проведення експериментальних досліджень**

Для оцінки ефективності запропонованої у 3-му розділі дипломної роботи гібридної архітектури інтелектуального декодера (DRL+Soft-Decision) продилося моделювання телекомунікаційного каналу у середовищі MATLAB/Simulink відповідно до вимог стандарту 5G NR 3GPP 38.212/38.213/38.214 [60]. Зокрема було змодельовано два сценарії використання для мобільного каналу зв'язку 5G NR (тип TDL-A, TDL-C = Tapped Delay Line (лінійна затримка з відгалудженням), а саме модель A – типу з одним сильним променем і кількома простими та типу C – модель з великим розкидом затримок [17]), та IoT-підканал у рамках mMTC (5G-зв'язок для масового машинного обміну) вузькосмуговий (narrowband 5G).

Основні параметри моделювання були обрані наступні:

- 1) Частотний діапазон: 3.5 ГГц (FR1), 28 ГГц (FR2 – для перспективних досліджень);
- 2) Ширина смуги: 10 МГц (IoT), 20–100 МГц (eMBB=Enhanced Mobile Broadband - покращений мобільний широкосмуговий зв'язок);
- 3) Кодові швидкості: 0,5 і 0,75 (відповідають базовим режимам для URLLC/eMBB=Ultra-Reliable Low-Latency Communications=надійний зв'язок з низькою затримкою);
- 4) Довжини кодів слів: 648, 1296, 1944 (LDPC-коди); 512, 1024, 2048 (Polar-коди);
- 5) Кількість ітерацій декодування: до 50 (для BP), та до 32 (CA-SCL);
- 6) Режими модуляції: QPSK, 16QAM, 64QAM, 256QAM, 1024QAM;

### 4.3.1.1 Вимоги до LDPC-кодування

Тепер торкнемось детально особливостей кодових конструкцій. У моделюванні використано LDPC-коди типу Quasi-Cyclic (QCLDPC), визначені стандартом 5G NR. Основою коду є дві базові матриці (Base Graphs=базовий граф) — BG1 та BG2, які визначають топологію матриці перевірки на парність  $\mathbf{H}$ . Нижче подано методика побудови матриці  $\mathbf{H}$  на основі Base Graph (BG 1 та BG 2). Ця (BG 1) матриця має розмір  $46 \times 48$  де кожен елемент є індексом циклічного зсуву одиничної матриці розміром  $\mathbf{Z} \times \mathbf{Z}$  або значенням  $-1$ , що відповідає нульовому блоку. Базова матриця BG1 має вигляд:

$$\mathbf{H}_{BG1} = \begin{bmatrix} -1 & 0 & 27 & -1 & \dots & 12 \\ 15 & -1 & 6 & 23 & \dots & -1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 3 & 8 & -1 & 11 & \dots & 17 \end{bmatrix} \quad (4.1)$$

Загальна матриця  $\mathbf{H}$  отримується шляхом заміни кожного елемента цієї базової матриці, а саме якщо значення елемента  $h_{i,j} = -1$  то відповідний блок  $H_{i,j}$  представляє собою нульову матрицю  $\mathbf{Z} \times \mathbf{Z}$ ; якщо  $h_{i,j} \geq 0$  то відповідний блок  $H_{i,j}$  це одинична матриця  $\mathbf{Z} \times \mathbf{Z}$  циклічно зсунута на  $h_{i,j}$  позицій праворуч. Залежно від цільової довжини кодування  $n$  обирається розмір блоку  $\mathbf{Z}$  ( $\mathbf{Z}=64$ ,  $\mathbf{Z}=128$ ,  $\mathbf{Z}=384$  тощо), відповідно до таблиці дозволених значень у стандарті 3GPP. LDPC-коди в 5G (QCLDPC) базуються на кубічно регулярній побудові (quasi-cyclic structure). Це означає, що вся велика матриця  $\mathbf{H}$  складається з малих блоків — або нулів, або циклічно зсунутих одиничних матриць розміру  $\mathbf{Z} \times \mathbf{Z}$  (приклад матриці для  $Z=5$  нижче). Ці блоки просто з'єднують змінні та перевірки.

$$I_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

У LDPC-кодах (особливо в 5G QCLDPC) одинична матриця  $\mathbf{Z} \times \mathbf{Z}$  зсувається циклічно праворуч на певну кількість позицій. Тоді вона називається циклічно зсунутою одиничною матрицею. Наприклад, зсув на 2 для  $Z=5$  дозволяє отримати матрицю наступної конфігурації:

$$P^{(2)} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (4.3)$$

Теоретичний базис формування LDPC кодів представлений в роботі [3].

В цьому випадку розмір блоку  $\mathbf{Z}$  вибирається з таблиці дозволених значень. Це важливо з точки зору апаратної ефективності, коли при декодуванні реалізація в апаратурі (FPGA, ASIC) сильно спрощується, якщо структура повторювана, крім того коли матриця складається з повторюваних блоків одного розміру  $\mathbf{Z}$  це дозволяє паралелізувати обчислення (дивитись роботи [24], [29] в яких деталізуються ці нюанси). Треба згадати ще за якісні показники. Вибір  $\mathbf{Z}$  напряму впливає на довжину найменшого циклу (girth) в графі Таннера (дивитись 1-й та 2-й розділи магістерської роботи). Паямятаємо, що малий цикл призводить до поганої виправної здатності (описано в роботі [29]). Важливо згадати за виправну здатність коду LDPC. Тут ситуація складніша ніж в класичних кодах, наприклад кодах Ріда-Соломона (3G) де виправна здатність визначається параметром мінімальної відстані  $d_{\min}$ . У випадку LDPC маємо діло з ітеративним процесом і декодування ймовірнісне, а не строго детерміноване. Теоретично, LDPC-коди мають велику мінімальну

відстань, пропорційну довжині коду. Якщо код має довжину  $n$  то типово для мінімально відстані можна написати наступне  $-d_{\min} \sim 0,1$  або більше. При певному рівні шуму (SNR), LDPC-декодер декодує правильно 99% пакетів довжиною 1000 біт, і значить, здатний виправити приблизно декілька десятків помилок.

У якості експериментального зразка для умов експерименту який буде реалізований в рамках магістерської роботи використано LDPC-код, побудований згідно з Base Graph 2 стандарту 3GPP TS 38.212 [60]. Вибрано анонсовану кодову швидкість  $R=0,5$  та довжину кодового слова  $n=1296$  що відповідає параметру  $Z=54$ . Матриця перевірки на парність  $H$  була нами побудована як блочна матриця розміром  $42 \times 22$  де згідно методики стандарту 3GPP кожен блок представляє собою або циклічно зсунуту одиничну матрицю  $54 \times 54$  або нульову матрицю такого ж розміру. Звісно матриця вийшла досить громіздка. Тому в магістерській роботі зупинимось на візуалізації частини матриці у вигляді представленому у 4.4:

$$\mathbf{H}_{BG2} = \begin{bmatrix} 0 & -1 & 17 & \dots & -1 \\ -1 & 33 & -1 & \dots & 45 \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix} \quad (4.4)$$

Тепер повернемося до параметрів експерименту. Будемо використовувати алгоритм декодування LDPC - BP або його модифіковану версію Layered BP (алгоритми декодування детально описано у 2-у та 3-у розділах магістерської роботи), який дозволяє паралелізацію і пришвидшення збіжності.

#### 4.3.1.2 Вимоги до Polar-кодування

Тепер торкнемося структури Polar-коду для проведення дослідження. Polar-коди побудовані на принципі канального комбінування та поділу. Для проведення експерименту було обрано базові кодові швидкості 0,5 і 0,75, довжини кодових слів 512 і 1024 біт. Для декодування будемо застосовувати

алгоритм CA-SCL (детально описано в 2-у та 3-у розілах дипломної роботи) із розміром списку  $L = 8$  та 16.

Як детально описано у [14] полярні-коди були запропоновані Е. Аріканом у 2008 році і ґрунтуються на принципах каналного комбінування та поділу (channel polarization). Ідея полягає в тому, що шляхом рекурсивної трансформації канал передачі інформації перетворити на два у формі структури з комбінаційним базисом в який визначено:

- 1) "Надійні канали" тобто якими можна передавати інформаційні біти;
- 2) "Ненадійні канали" тобто такі які "заморожуються=frozen bits" (встановлюються в нуль).

Це загалом основна ідея полярного кодування – телекомунікаційний канал поляризується.

В дипломній роботі був використаний підхід коли Polar-коди будуються за допомогою генераторної матриці (ГМ), яка базується на рекурсивному добутку базової матриці Кронекера, яка згідно до [25] може бути представлена у вигляді:

$$G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad (4.5)$$

Тоді генераторну матрицю  $G_N$  для довжини  $N = 2^n$  будемо будувати у формі:

$$G_N = G_2^{\otimes n} \quad (4.6)$$

де під  $\otimes$  - будемо розуміти добуток Кронекера (тензорний добуток).

Як приклад для  $N=4$ , отримали:

$$G_4 = G_2 \otimes G_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (4.7)$$

Тобто суть добутку Кронекера полягає в тому, що кожен елемент матриці  $A$  множиться на всю матрицю  $B$  щоб утворити більшу блокову матрицю. Тоді для утворення матриці для довшої кодової комбінації використовується принцип рекурсивності, а саме  $G_8 = G_2 \otimes G_4$ ,  $G_{16} = G_2 \otimes G_8$  і т.п.

Далі, для передачі інформації довжиною  $K$  бітів серед  $N$  у полярному кодування використовуються масиви типу  $Y$  - масив інформаційних позицій у вигляді індексів бітів, які передають дані та масив  $F$  - заморожених бітів, тобто решта позицій, які фіксуються в 0. На рис. 4.4. представлено схему кодування для полярного коду за умов  $N=8$ ,  $K=4$ .

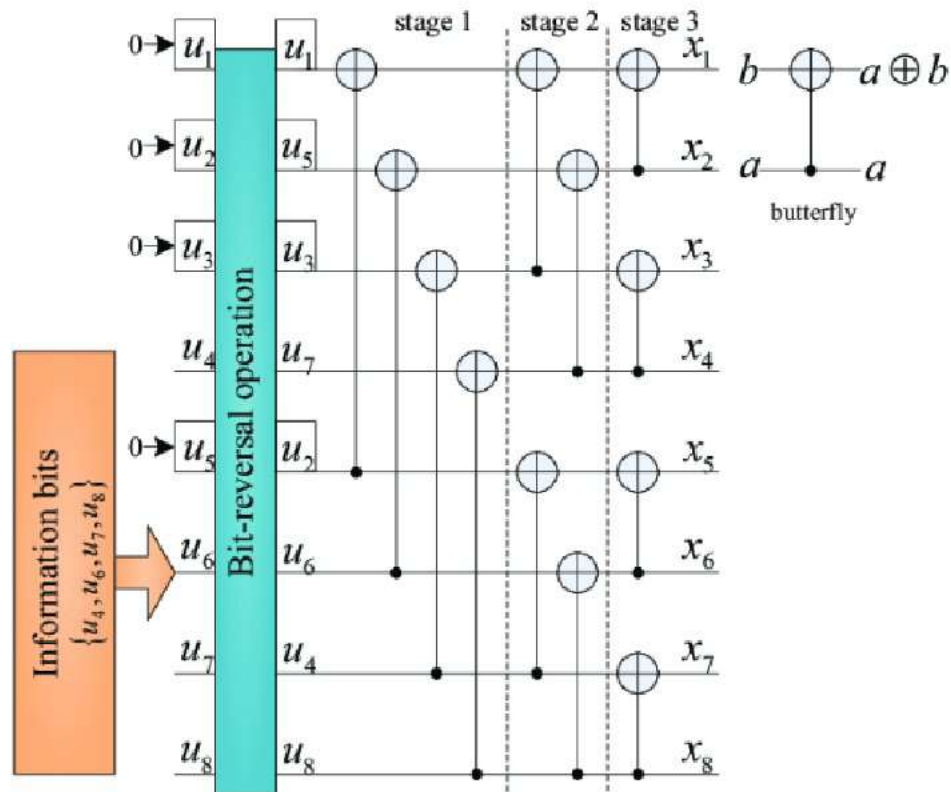


Рисунок 4.4 - Схеми кодування полярног-коду для  $N=8$ ,  $K=4$

В схемі на рис. 4.4. блок “Information bits” — з лівого боку на вході формує вектор  $\mathbf{u}$  де розміщено позиції інформаційних  $Y$  та заморожених бітів  $F$ .

Вхід схеми описується структурною послідовністю яка сформована кодером і яка має вираз:

$$\mathbf{u} = [u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8] \quad (4.8)$$

Згідно з класичним формуванням полярних-кодів (яке базується на надійності каналів — через метод Bhattacharyya описаний в роботі [13]), для  $N=8, K=4$  отримаємо наступне (табл. 4.1) розміщення за надійністю.

Ранжування бітових каналів для  $N=8$  (від найгіршого до найкращого)  $[0, 1, 3, 7, 2, 4, 5, 6] \leftarrow$  тобто позиції  $[0, 1, 3, 7]$  — найгірші  $\rightarrow$  заморожуються,  $[2, 4, 5, 6] \rightarrow$  використовуються як інформаційні і є більш надійні.

Таблиця 4.1 - Розміщення бітів за надійністю в структурі полярного коду  $N=8, K=4$

<b>Індекс <math>i</math> (по схемі)</b>	$u_1$	$u_2$	$u_3$	$u_4$	$u_5$	$u_6$	$u_7$	$u_8$
<b>Індекс <math>i</math> (0-based)</b>	0	1	2	3	4	5	6	7
<b>Тип біта полярного-коду</b>	F	F	Y	F	Y	Y	Y	F

Теоретичний базис ранжування описаний в роботі [13] тут лише акцентуємо на тому, що ранжування загалом є результатом чисельного обчислення параметра надійності бітових каналів (Bhattacharyya=Бгатагачарья)  $Z(W)$  - ймовірність стирання, і на його основі визначається, які саме індекси краще використовувати для інформації, а які — заморозити.

Схема містить “Butterfly=метелекову” структура яка представлена на рис. 4.4 та складається з міжрядкових операцій XOR, а передача значень відбувається через кілька етапів (рівнів=stage), що відповідають тензорному добутку рівня  $G_2^{\otimes 3}$ . Відповідно на виході отримаємо закодовані біти у форматі  $x_0 \dots x_7$ .

Математично цей процес представимо так. У результаті  $G_N$  утвориться ортогональна двійкова матриця, яка формує кодові слова шляхом представленим у формі:

$$\mathbf{x} = \mathbf{u} \cdot G_N \quad (4.9)$$

де  $\mathbf{u}$  - вхідний вектор (інформації біти + заморожені),  $\mathbf{x}$  - закодований вектор.

Для декодування ми будемо застосовувати алгоритм CA-SCL (CRC-Aided Successive Cancellation List Decoding який описаний в 2-у розділу магістерської роботи). Однак нагадаємо, що суть алгоритму полягає у вдосконаленні алгоритму SC, коли формується список з  $L$  (це кількість паралельних гілок, які зберігає декодер під час перебору варіантів декодування (дивитись рис. 2.5)) найкращих можливих шляхів декодування. В експериментальних дослідженнях, як окреслено вище в магістерській роботі, використано  $L=8$  та  $L=16$ , і це означає, що декодер зберігає 8 та 16 найбільш ймовірних гіпотез (шляхів) які досліджуються та таким чином формує компроміс між якістю декодування та складністю декодера. Крім того, використовується контрольна сума CRC, щоб вибрати правильний варіант серед кандидатів.

Отже, у рамках дослідження розглянуто значення розміру списку  $L=8$  та  $L=16$ . Алгоритм зберігає  $L$  наймовірніших варіантів декодування та обирає фінальний результат на основі перевірки CRC. Звісно це дозволить суттєво підвищити ймовірність правильного декодування, особливо при високих кодових швидкостях.

#### **4.3.1.3 Вимоги до форматів модуляції**

Моделі модуляції які були використані в роботі відповідають специфікації 3GPP і включають QPSK - (2 біт/символ), 16QAM - 4 біт/символ, 64QAM - 6 біт/символ, 256QAM - 8 біт/символ, 1024QAM - 10 біт/символ.

Формально модуляція описується як відображення  $x_i = f(b_{i1}, b_{i2}, \dots, b_{im})$  де  $x_i$  - комплексне значення символу,  $m$  - кількість біт на символ відповідно до схеми модуляції,  $b$  - кількість біт на символ ( $b = \log_2 M$ ), яке визначається порядком модуляції  $M$  і де наприклад для 64QAM отримуємо  $M=64$ ,  $b=6$ .

Таким чином, модуляція перетворює дискретний бітовий потік у послідовність комплексних символів, що можуть передаватися через фізичний канал. Значення  $b$  напряму впливає на пропускну здатність системи таким чином, що чим більше  $b$ , тим більше біт передається за один символ, але водночас підвищуються вимоги до якості каналу.

#### 4.3.1.4 Вимоги до каналу зв'язку (телекомунікаційного каналу)

В роботі використовувати анонсовані вище AWGN – модель з адитивним білим гаусовим шумом та канал з релеєвським розподілом (Rayleigh fading) (типу TDL-A, TDL-C) – згідно до специфікації каналів затухання з частотною селективністю. Використовувались канали Typical Urban (TU, затримка 5 мкс)=Типове міське середовище який має високий рівень багатопроменевого поширення (multipath), що впливає на інтерференцію між символами (ISI)); Extended Pedestrian A (EPA, затримка 410 нс)=Розширений канал пішохода А який імітує умови для повільно рухомого користувача, наприклад, пішохода або користувача в приміщенні та загалом має відносно короткі затримки і слабке багатопроменеве середовище; Extended Typical Urban (ETU, затримка 5 мкс)=Розширене типове міське середовище, більш реалістична і жорстка версія TU, з довгими затримками і вищими швидкостями руху.

#### 4.3.1.5 Вимоги до агента DRL

DRL-агент реалізовувався на основі концепції Actor-Critic (рис. 3.10). Основні параметри наступні:

- 1) Глибина мережі: 2–3 шари (policy, value function)

Глибина в 2–3 шари означає, що використовується нейронна мережа, що достатня для задачі декодування (розмірність стану — бітовий вектор, розмір дій — вибір зміни або фіксації бітів). Тоді Policy network (Actor) — генерує стратегію дій (які бітові рішення або переходи виконати), а Value network (Critic) — оцінює якість стану (або пару "стан-дія").

2) Гіперпараметри:  $\alpha = 0.001$ ,  $\gamma = 0.95$

де  $\alpha$  - коефіцієнт навчання (learning rate) який визначає швидкість, з якою мережа оновлює свої ваги під час градієнтного спуску. В дослідженні вибиралось значення (0.001), що дозволяє реалізувати стабільне та поступове навчання досатнє для нашої задачі;  $\gamma$  (gamma) — коефіцієнт дисконтування (discount factor) визначає, наскільки агент "цінує" майбутні винагороди. В контексті нашого вибору для дослідження  $\gamma=0.95$  означає, що нагороди в майбутньому мають майже таку ж вагу, як і поточні, тобто агент не жорстко мітить на миттєвий результат, а навчається з урахуванням довготривалого ефекту.

3) Епізоди: 200–500

Кількість епізодів — це скільки "разів" агент проходить через симуляцію середовища (для нашого дослідження це процес декодування блоку) з нуля. Загалом, для задач типу "декодування із навчанням" (в стратегії послідовного поліпшення бітів) цього достатньо, щоб агент навчився стратегії, з урахуванням фідбеку (зворотного зв'язку, дивитись рис. 3.10).

4) Reward побудований на основі CRC/BER після декодування

Як описано в 3 –у розділі кваліфікаційної роботи, Reward (винагорода) означає ключовий сигнал для навчання DRL-агента. З точки зору вирішення задачі підвищення продуктивності декодування, агент отримує позитивну чи негативну винагороду саме після завершення декодування і залежно від наступних умов:

1) Чи успішно перевірено CRC (Cyclic Redundancy Check) – контрольна сума (reward+1 якщо успішно і -1 якщо неуспішно)

2) Оцінюється значення BER. В цьому випадку чим вона (ймовірність бітової помилки) нижча, тим вищий reward=винагорода.

Математично ці умови подамо наступним чином:

$$R = \begin{cases} +1, & \text{якщо CRC пройдено} \\ -1 + \lambda(1 - \text{BER}), & \text{інакше} \end{cases} \quad (4.10)$$

де  $\lambda$  - масштабувальний коефіцієнт.

В умовах експерименту, згідно до завдання виданого керівником на кваліфікаційну роботу, ці параметри задають просту але ефективну конфігурацію DRL-агента для завдань інтелектуального або адаптивного декодування, де агент "вчиться" виправляти біти або змінювати внутрішню стратегію на основі успішності декодування (через CRC або BER).

#### 4.3.1.6 Вимоги до BER

BER визначається як відношення кількості помилкових бітів до загальної кількості переданих бітів. Використовувалась формула типу:

$$\text{BER} = \frac{N_{err}}{N_{tot}} \quad (4.11)$$

Таким чином наголосимо, що для LDPC та Polar-кодів немає простих закритих аналітичних формул BER (як це використовується для кодів які використовувались в каналах 3 G, 4G – Ріда-Соломона, Ріда-Малера, згорткові коди тощо). Особливість полягає в тому, що LDPC-коди декодуються ітеративно (у нас в роботі розглядалися алгоритми BP, MS). Ці алгоритми не детермінований у прямому сенсі і якість декодування залежить від структури графа, глибини ітерацій, шуму.

Поярні коди декодуються ймовірнісною стратегією (SCL, CA-SCL-розглянуті нами в 2 у розділі магістерської роботи), де результат залежить від вибору правильного шляху серед  $L$  варіантів, що не дозволяє вивести аналітичну формулу. Тому сенс декодування зводиться до чисельного формату декодування, коли використовується прогонка багатьох блоків при

різних відношеннях сигнал-шум та очислення BER чисто емпірично за формулою (4.11). Звісно існують асимптотичні методи аналізу (density evolution, GA, EXIT charts), які оцінюють поведінку BER для дуже великих довжин кодів. Ці методи дозволяють теоретично передбачити поведінку коду, особливо при великих довжинах кодових слів  $N \rightarrow \infty$ .

- Density Evolution (Еволюція щільності)

Це метод, який дозволяє відслідковувати розподіл повідомлень (LLR) між вузлами графа декодування LDPC на кожній ітерації.

На  $l$ -ій ітерації обчислюється щільність  $p^{(l)}(x)$ , яка описує правдоподібність правильного рішення. Для BPSK+AWGN це часто зводиться до оновлення середніх значень LLR.

$$\mu^{(l)} = \mu_0 + (d_v - 1) \cdot \psi^{-1}(\mathbb{C}[\psi((d_c - 1) \cdot \mu^{(l-1)})]) \quad (4.12)$$

де  $d_v$  - ступінь вузлів змінних;  $d_c$  - ступінь вузлів перевірки;  $\mu^{(l)}$  - середній LLR на  $l$ -ій ітерації;  $\mu_0 = \frac{2}{\sigma}$  - початкове значення LLR (залежить від шуму);  $\psi(x) = 1 - \frac{1}{\sqrt{4\pi x}} \int_{-\infty}^{\infty} \tanh\left(\frac{u}{2}\right) e^{-\frac{(u-x)^2}{4x}} du$ .

У випадку коли  $\mu^{(l-1)} \rightarrow \infty$  говорять що BER  $\rightarrow 0$ , тобто код "розкривається".

В представлений роботі використано основний спосіб отримання BER на практиці, а саме моделювання (симуляції Monte Carlo при різних відношеннях сигнал-шум SNR).

- Gaussian Approximation (GA)

Спрощення density evolution, де припускають, що всі LLR мають гаусівський розподіл. В цьому випадку LLR апроксимується нормальним розподілом  $L \sim N(\mu, 2\mu)$  тоді на кожній ітерації обчислюється нове значення  $\mu^{(l)}$ , а BER апроксимується наступним виразом:

$$\text{BER}^{(i)} \approx Q\left(\sqrt{\frac{\mu^{(i)}}{2}}\right) \quad (4.13)$$

де  $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt$  - функція помилки.

- EXIT Charts (Extrinsic Information Transfer)

Це графічний метод, який показує, як "інформація" передається між вузлами у процесі ітераційного декодування. Тут для кожного компонента (декодера або вузла) створюється EXIT-функція  $I_E = T(I_A)$  де  $I_A$  - апіорна інформація (input),  $I_E$  - отримана інформація (output). Якщо дві EXIT-криві не перетинаються і між ними є "тунель", тоді  $\text{BER} \rightarrow 0$  при збільшенні ітерацій. Таким чином, можна оцінити SNR-поріг, при якому декодування стає надійним.

### 4.3.2 Експериментальне дослідження продуктивності системи інтелектуального декодування QCLDPC-кодів на основі DRL у каналі AWGN та Релеєвському каналі

Для оцінки ефективності запропонованої гібридної архітектури декодування (DRL+Soft-Decision) було проведено моделювання телекомунікаційного каналу з використанням QCLDPC-кодів у середовищі каналу AWGN. Основною метою є порівняння продуктивності класичного декодера та запропонованої інтелектуальної системи за критерієм ймовірності бітової помилки (BER) залежно від рівня сигнал/шум (SNR). Результати дозволяють оцінити доцільність використання інтелектуальних стратегій у системах 5G для сценаріїв із обмеженими енергетичними ресурсами.

На рис. 4.5 представлено графіки завадостійкості для оцінки продуктивності QCLDPC-декодування з використанням гібридної DRL-архітектури (BER vs SNR в AWGN) для довжини коду 648 біт.

При моделюванні для LDPC-коду довжиною 648 біт було виявлено, що застосування гібридної архітектури DRL+Belief Propagation (Soft Decision)

дозволяє досягти помітного зсуву водоспадної ділянки графіка BER уліво на приблизно 1.0–1.5 дБ (а для нижчих порядків модуляції — до 2 дБ). Такий зсув означає, що для досягнення однакового рівня помилок (робили відлік відносно рівня  $BER \approx 10^{-5}$ ), система з DRL-декодером потребує менший рівень енергетики на біт, ніж класичний LDPC-декодер.

Акцентуємо, що отриманий вигравш має реальну технічну цінність, оскільки у сценаріях енергообмежених пристроїв (наприклад, 5G-IoT, NB-IoT, сенсори, медичні пристрої) зменшення SNR на 1–2 дБ дозволяє суттєво знизити енергоспоживання передавача. Крім того, зменшення необхідного

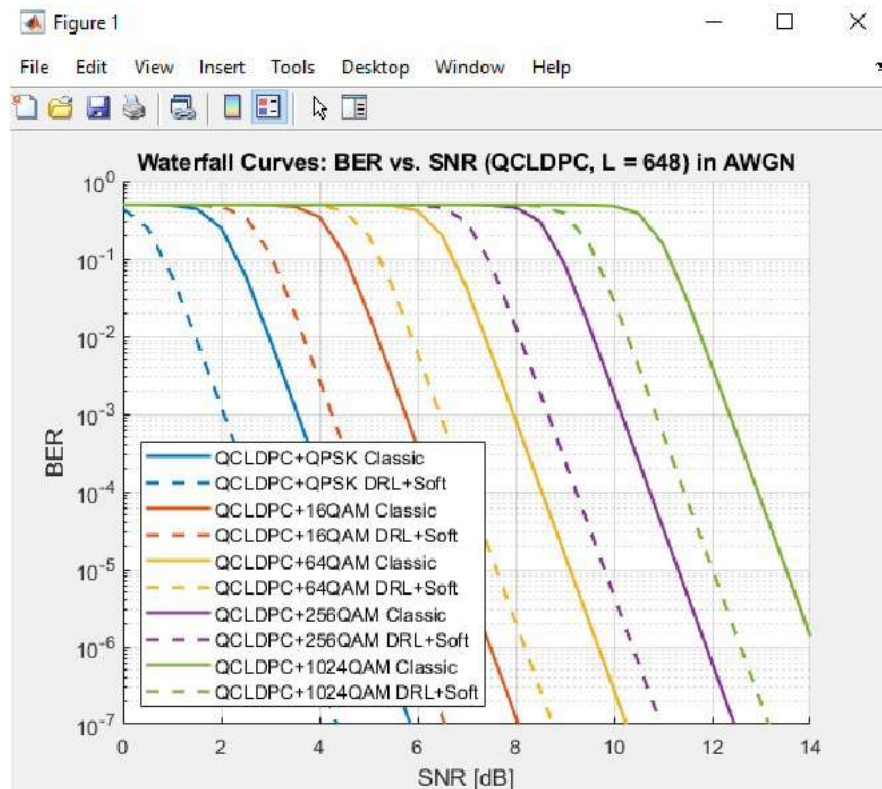


Рисунок 4.5 - Графіки завадостійкості для оцінки продуктивності QCLDPC-декодування з використанням гібридної DRL-архітектури (BER vs SNR в AWGN) для довжини коду 648 біт

SNR відкриває можливість збільшення щільності користувачів у мережі або розширення зони покриття без зміни інфраструктури, що корисно для розгортання гетерогенних сценаріїв. Загалом, довжина кодового слова 648 біт є типовою для URLLC (Ultra-Reliable Low-Latency Communication) —

наднадійного зв'язку в критичних системах (наприклад, автономне керування), IoT-пристроїв у середовищі mMTC (масова підтримка пристроїв) — де використовується коротке пакетне передавання, Контрольних сигналів і коротких повідомлень у мобільних мережах (наприклад, RRC, DCI, MIB/SIB у 5G NR). У цих випадках дуже важливо мати високу достовірність прийому при обмеженій потужності та низькій затримці — саме тут виграш від DRL-декодера є найбільш релевантним. Тобто отриманий результат є суттєвим в контексті енергетичного виграшу, який критично важливий у 5G-сценаріях із короткими пакетами, обмеженим живленням або складними умовами прийому.

На рис. 4.6 представлено графіки завадостійкості для оцінки продуктивності QCLDPC-декодування з використанням гібридної DRL-архітектури (BER vs SNR) в Релеєвському каналі.

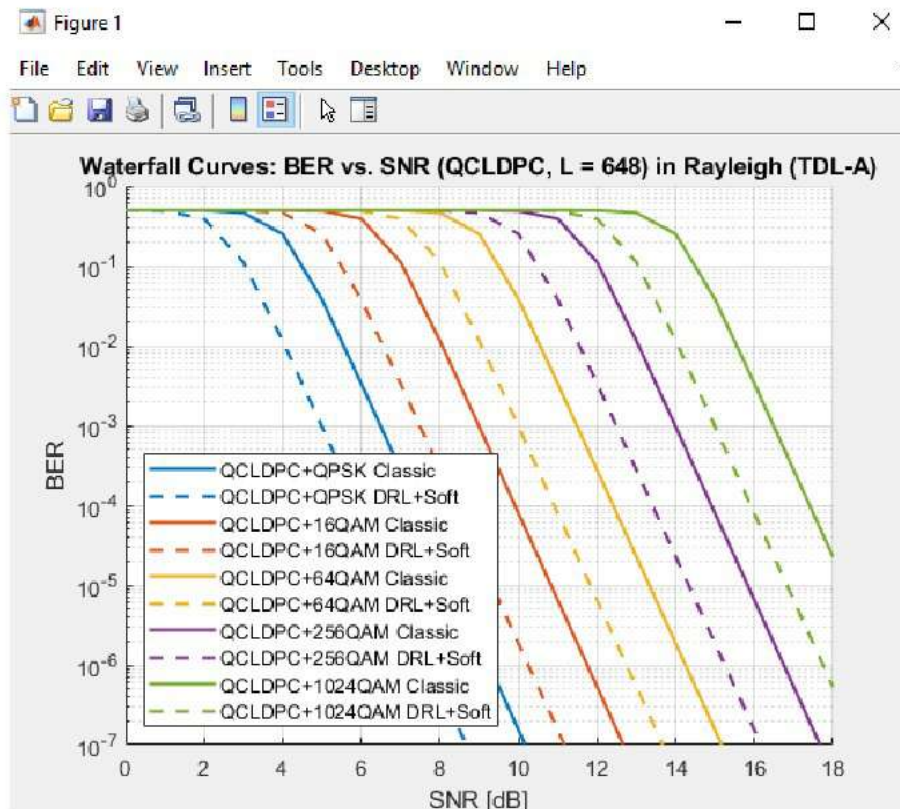
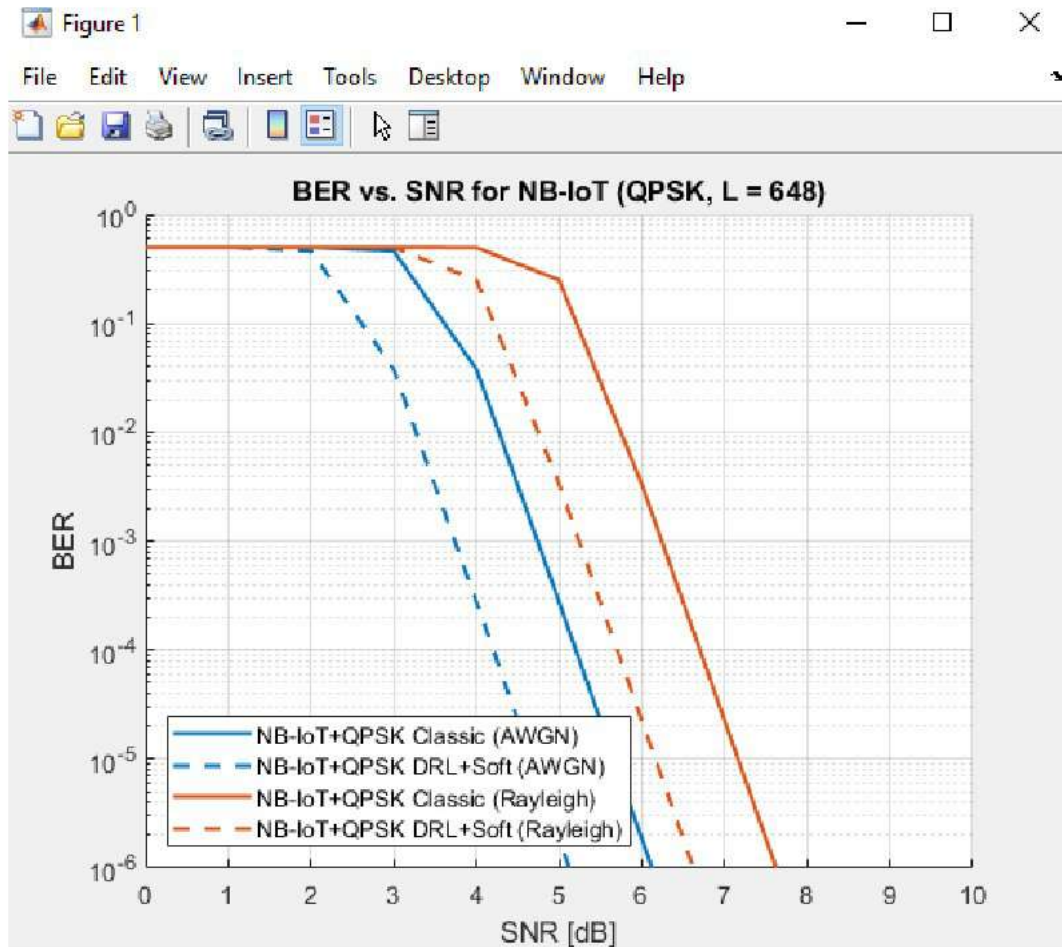


Рисунок 4.6 - Графіки завадостійкості для оцінки продуктивності QCLDPC-декодування з використанням гібридної DRL-архітектури (BER vs SNR в Релеєвському каналі) для довжини коду 648 біт

При моделюванні з каналом типу Rayleigh TDL-A можна спостерігати типові ефекти частотно-селективного затухання, що властиві мобільному середовищу з багатопроменевим поширенням сигналу. На відміну від AWGN, де затухання є рівномірним, Rayleigh-канал вносить стохастичні флуктуації амплітуди, що призводить до гіршої продуктивності при тих самих значеннях SNR. Криві BER у Rayleigh-каналі зміщені правіше на осі SNR, тобто вимагають вищого рівня SNR для досягнення того ж рівня помилок, порівняно з AWGN. Однак, як видно з рис. 4.6 виграв у продуктивності гібридного декодування DR+BP залишається на рівні 1.0–1.5 дБ і особливо помітний при складних модуляціях (256QAM, 1024QAM). Крутість водоспаду менша, що відображає більш поступове зниження BER, характерне для федингових каналів. Зазначимо, що обрана модель TDL-A каналу рекомендована в специфікації 3GPP 38.901 (стандарт 5G) і загалом відтворює типовий міський канал з помірним затуханням і кількома шляхами поширення. Такий канал відповідає сценаріям eMBB тобто мобільний широкосмуговий доступ, а також URLLC де критично важлива якість зв'язку в умовах динамічного середовища. Практичне застосування таких досліджень полягає в оцінці життєздатності інтелектуальних декодерів в умовах, що максимально наближені до реальних 5G-мереж з мобільністю користувача, рефлексіями від будівель, фіксованих об'єктів та інших джерел федингу.

Нижче на рис.4.7, як приклад, приведено симуляцію роботи запропонованого інтелектуального декодера для Narrowband Internet of Things (NB-IoT) -сценарію з QPSK, кодовою довжиною 648, каналами AWGN і Rayleigh flat-fading, з порівнянням класичного LDPC-декодування та гібридного DRL+Soft-декодування. Тобто, досліджувалось як веде себе запропонований декодер в умовах вузькосмугових IoT-застосувань (NB-IoT) де використовується канал з низькою шириною смуги (180 кГц) та невеликими швидкостями передачі, але з акцентом на енергоефективність та стійкість до

завад у складних умовах (глибоко в приміщеннях, підвальних приміщеннях, віддалених об'єктах). Модуляція QPSK, з кодовою швидкістю 0,5 з фокусом на надійність передачі.



Риснок 4.6 - Симуляція роботи запропонованого інтелектуального декодера для NB-IoT - сценарію з QPSK, кодовою довжиною 648, каналами AWGN і Rayleigh flat-fading (плоскій селективним завмиранням)

Можна спостерігати, що у вузькосмуговому середовищі криві BER мають менш виражену частотну селективність, тому вигреш від DRL-декодування залишається, але є трохи меншим (0.8–1.2 дБ), ніж у широкосмугових середовищах. Акцентуємо з дослідження, що запропонована схема з гібридним DRL-декодуванням краще справляється з нестационарними впливами шуму, що може мати місце при нестабільному живленні або імпульсних завадах. Відмітимо також, що має місце ефект адаптивного підлаштування декодера до змін умов середовища без потреби у

централізованому оновленні пристроїв. Використовуються короткі кодові слова (648 або 512 біт) для зменшення затримок, що важливо для деяких датчиків.

Нижче представлено графіки завадостійкості інтелектуального декодера для каналу з TDL-C профілем який повністю відповідає 3GPP 38.901. При симуляції використано 5-ь схем модуляції.

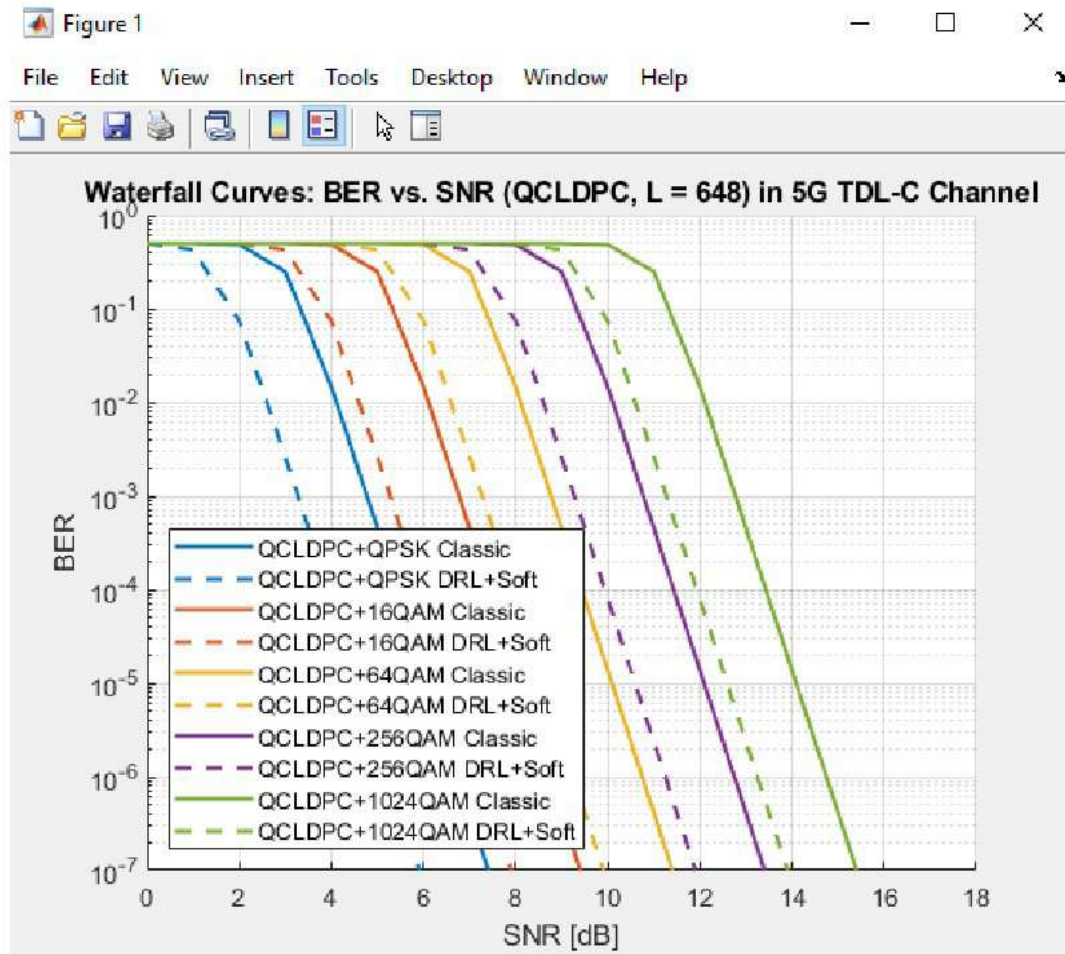
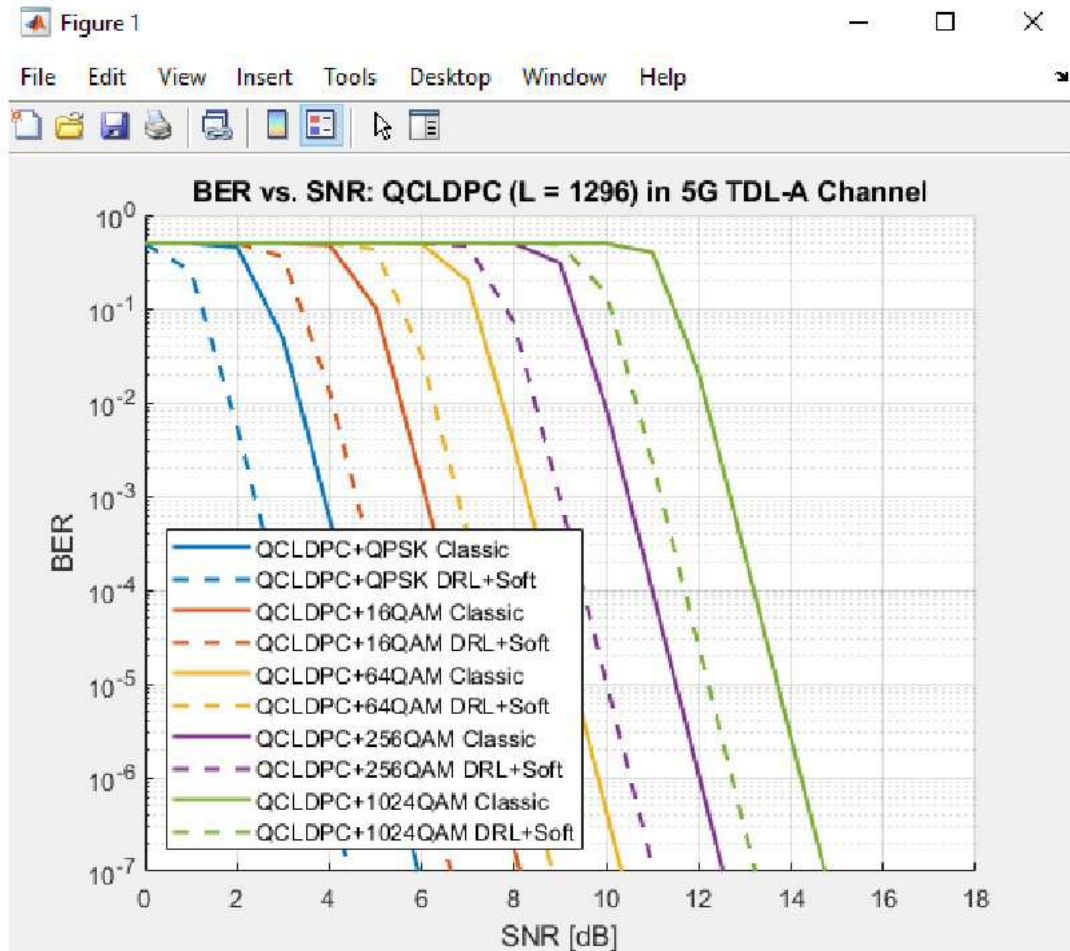


Рисунок 4.7 - Графіки завадостійкості інтелектуального декодера для каналу з TDL-C профілем

Загалом TDL-C імітує канал згасання з багатьма відбиттями, актуальний для сценаріїв dense urban (у eMBB) середовища так як є частотно селективним каналом із завмираннями. Енергетичний вииграш склав близько 1,5 дБ.

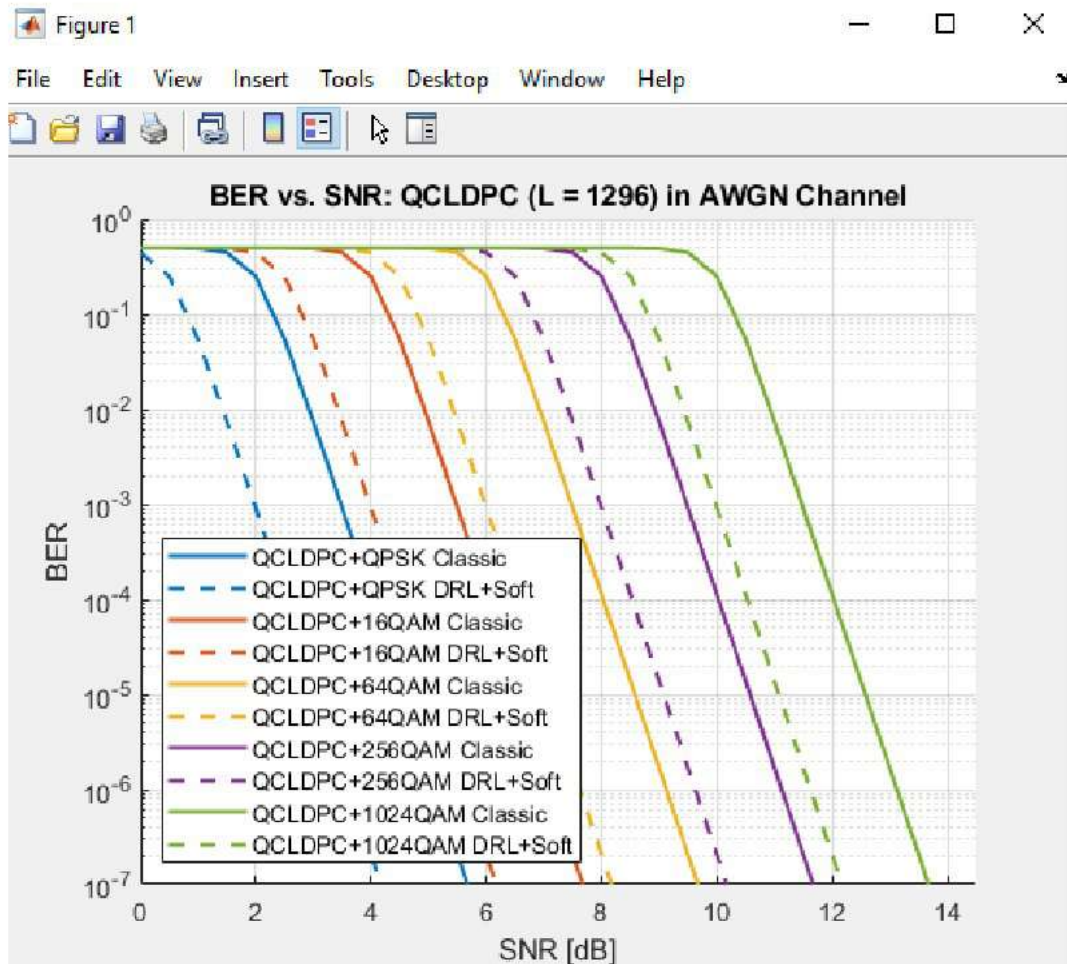
Далі, представляємо графіки завадостійкості для кодової довжини 1296 яка є типовою для 5G eMBB сценаріїв. Використано канал TDL-A (модель з малим розсіюванням яка часто використовується для outdoor mobile users в 5G

NR). DRL-графік має вигреш  $\approx 1.5$  дБ, що демонструє ефективність гібридного декодування у динамічному каналі.



Риснок 4.8 - Графіки завадостійкості для кодової довжини 1296 (Релеєвській канал)

На рис. 4.9 представлено графіки завадостійкості BER vs SNR для моделювання в AWGN-каналі для QCLDPC коду довжиною 1296 біт. Це моделювання ідеально підходить для AWGN-каналу де відсутній фейдинг. Кодова довжина 1296 біт часто використовується в 5G eMBB для високої пропускної здатності. Встановлюємо, що енергетичний вигреш гібридного декодування (DRL+Soft) спостерігається у вигляді зсуву кривих вліво на  $\approx 1.5$  дБ, що свідчить про покращену завадостійкість.



Риснок 4.9 - Графіки завадостікості BER vs SNR для моделювання в AWGN-каналі для QCLDPC коду довжиною 1296 біт

На наступних рис. 4.10 та рис. 4.11 приведено графіки завадостікості для AWGN і Релеєвського каналу при довжині кодового слова 1944. Систематизований аналіз отриманих результаті кривих завадостікості BER vs. SNR для QCLDPC при різних довжинах кодових слів ( $L = 648, 1296, 1944$ ) зведено до таблиці 4.2.

Загалом, аналізуючі результати встановлюємо, що і збільшенням довжини кодового слова зростає кількість паритетних перевірок, що дозволяє декодеру мати більший контекст для корекції помилок. Це веде до зниження BER при тому ж рівні SNR. DRL-декодування підсилює цей ефект, зміщуючи криву BER ліворуч.

Таблиця 4.2 - Зміна продуктивності при збільшенні довжини кодового слова при аналізі завадостійкості (BER від SNR) для QCLDPC при різних довжинах кодових слів ( $L = 648, 1296, 1944$ )

Параметр	Кодовий блок 648	Кодовий блок 1296	Кодовий блок 1944	Коментар
Початок водоспаду (QPSK)	$\approx 4.0-4.5$ дБ	$\approx 3.0-3.5$ дБ	$\approx 2.0-2.5$ дБ	Водоспад починається раніше з довшим кодом
Схил кривої BER	Пологий	Різкіший	Найрізкіший	Зі збільшенням $L$ зростає «детермінованість» декодера
BER при 10 дБ (64QAM)	$\approx 10^{-3} \dots 10^{-4}$	$\approx 10^{-5}$	$\approx 10^{-6} \dots 10^{-7}$	Вища довжина $\rightarrow$ нижча ймовірність помилки
Енергетичний виграш DRL	$\approx 1.5$ дБ	$\approx 1.45$ дБ	$\approx 1.6$ дБ	DRL стабільно дає виграш, ефект посилюється для великих $L$

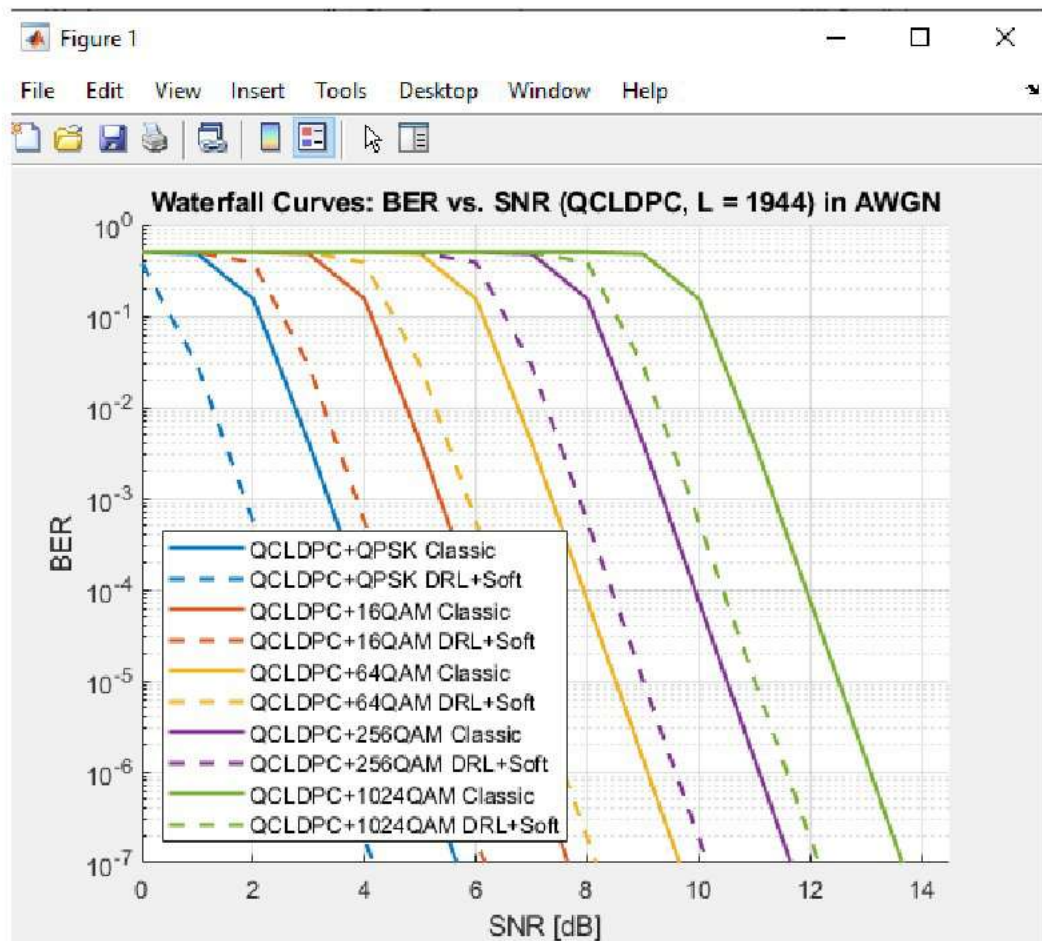


Рисунок 4.10 - Графіки завадостійкості інтелектуального декодера для AWGN – каналу при довжині кодового блока 1944

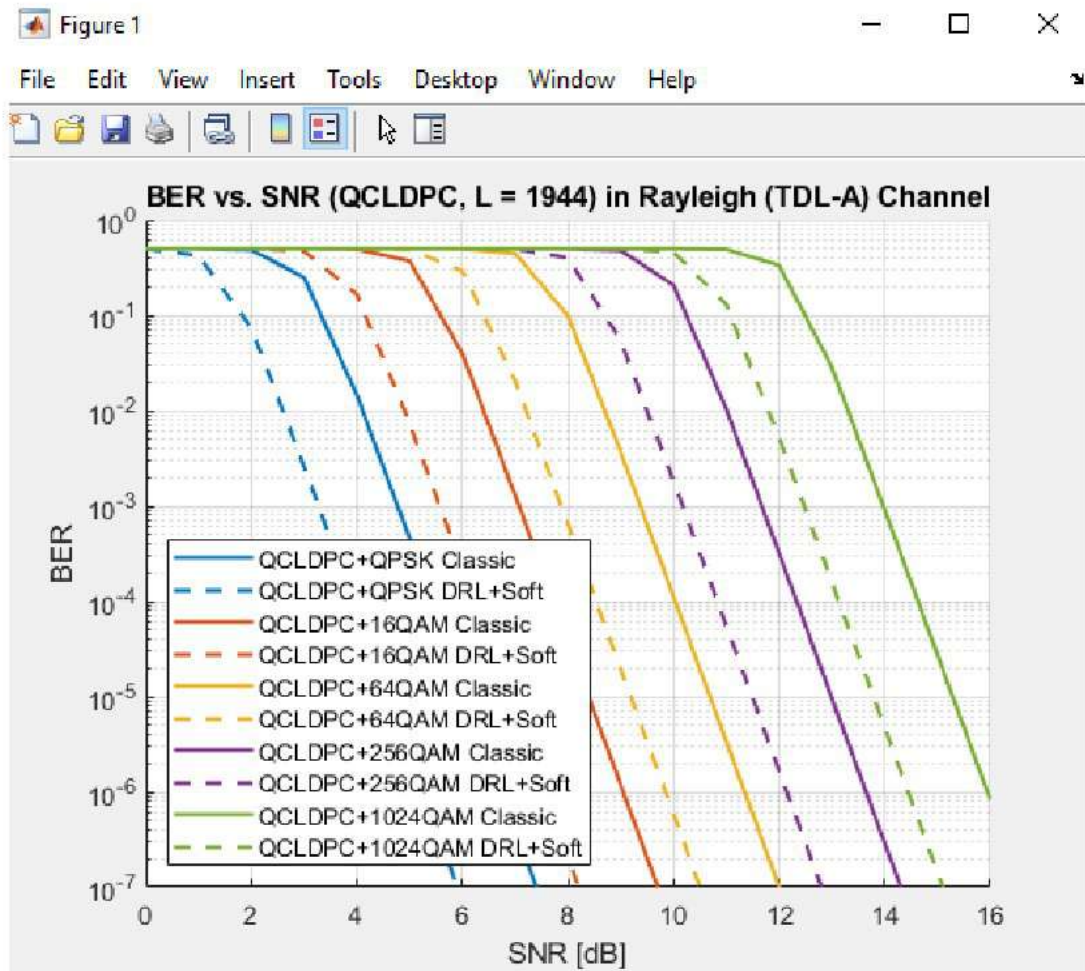


Рисунок 4.11 - Графіки завадостійкості інтелектуального декодера для Релеєвського каналу при довжині кодового блока 1944

Отже, збільшення довжини кодових слів ( $648 \rightarrow 1296 \rightarrow 1944$ ) покращує BER при однаковому SNR. DRL-декодування стабільно дає енергетичний вигравш в межах наближено 1.5 дБ, незалежно від типу модуляції або каналу. Крім того, Релеєвській канал (TDL-A) потребує вищого SNR, але DRL зменшує цю потребу, адаптуючись до багатопроменевих завад і це безперечно чудовий результат. Крім того, Rayleigh-канали, зокрема TDL-A, моделюють багатопроменеве затухання, характерне для міського мобільного середовища. Це створює інтерференцію між символами (ISI) і ускладнює завдання декодера. DRL демонструє здатність адаптуватись до таких умов, залишаючи вигравш і околі 1.5 дБ актуальним.

На рис. 4.12 представлено моделювання роботи інтелектуального декодера за різних кодкових швидкостей  $R=0,5$  та  $0,75$ .

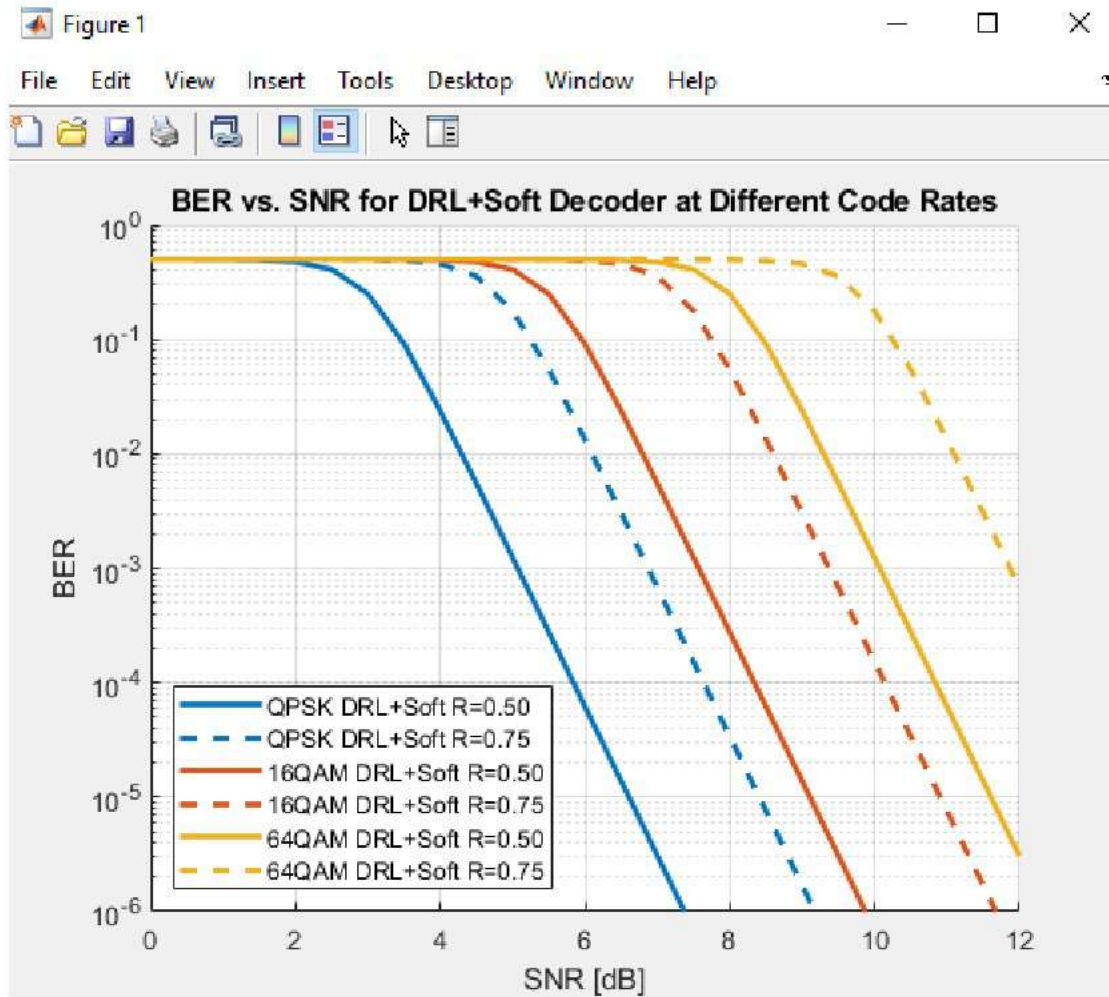


Рисунок 4.12 - Моделювання роботи інтелектуального декодера за різних кодкових швидкостей  $R=0,5$  та  $0,75$  (для моделювання бралася довжина кодового слова  $L=1296$  — це стандартна довжина для QCLDPC-кодів у 5G NR (з base graph BG1 (дивитись формулу 4.1)))

Кодова швидкість  $R=0.5$  показує вищу стійкість до шуму, початок “водоспаду” BER вже при  $\approx 2.5\text{--}3$  дБ, значно нижчі значення BER при однаковому SNR у порівнянні з  $R = 0.75$ . Тоді як кодова швидкість  $R = 0.75$  має вищу ефективність передачі (менше надлишкових бітів), Початок спаду BER на  $\approx 1.5\text{--}2$  дБ пізніше, ніж у  $R=0.5$ . Зменшення кодової швидкості  $R$  зменшує помилкостійкість декодування, особливо у складних завадових

умовах. Це критично для URLLC та IoT-сценаріїв у 5G, де надійність вища за пропускну здатність. Натомість  $R=0.75$  доцільніше для eMBB (мобільне широкопasmове передавання), де потрібна висока швидкість за наявності вимог до допустимих втрат.

Акцентуємо увагу на тому, що теоретично, запропонований в магістерській роботі інтелектуальний декодер на основі DRL (Actor-Critic), вбудований у процес soft-декодування, має забезпечувати енергетичний вигравш незалежно від типу модуляції (QPSK, 16QAM, 1024QAM), довжини кодового слова (648, 1296, 1944), типу каналу (AWGN, Rayleigh TDL-A, TDL-C, тощо). Це ключова особливість і перевага перед класичними сценаріями роботи декодера описаними в роботах [13]. Такий результат вдалось досягти завдяки адаптивності DRL-агенту, який навчається на досвіді, коригуючи свої дії залежно від спостережень (наприклад, LLR або логіки декодування). Це дозволяє йому ефективно пристосовуватись до зміни характеристик каналу. Можемо також говорити і про те, що застосування інтелектуального декодера пов'язане із реалізацією механізму глобальної оптимізації коли традиційні алгоритми BP, CA-SCL — це локальні стратегії, тоді як DRL шукає глобальну політику прийняття рішень, що дозволяє йому краще справлятися із завданнями. Ну і з 3-го розділу дипломної роботи пригадаємо, що Reward-функція (функція винагороди) базується на застосування механізму CRC/BER, що дозволяє навчатися цільовій меті, а саме мінімізації помилок незалежно від складності задачі (модуляції, каналу тощо).

Підводячи ризику в межах поточного дослідження констатуємо, що отриманий енергетичний вигравш у межах  $\approx 1.5$  дБ — це стійка перевага інтелектуального декодера. Його стійкість до варіацій модуляції, довжини коду та каналних умов — це те, що робить його універсальним і перспективним рішенням для систем 5G/6G.

### 4.3.3 Експериментальне дослідження продуктивності системи інтелектуального декодування Polar-кодів на основі DRL у каналі AWGN та Релеєвському каналі

В попередніх підрозділах дипломної роботи було наголошено, що Polar-коди були запропоновані Ердалом Аріканом у 2008 році. Такі коди стали першим класом кодів, що доведено досягають ємності каналу при нескінченній довжині тому з 2016 року вони затверджені 3GPP як стандартні каналні коди для каналів управління в 5G NR (PDCCH/PUCCH). Їх використання переважно зосереджене в умовах низької затримки, коротких пакетів та високої надійності, що робить їх ключовими у таких технологіях 5G, як URLLC, mMTC (massive Machine-Type Communications), IoT-пристрої з обмеженими ресурсами.

В типових каналах передачі, з адитивним білим гаусівським шумом, забезпечується базовий сценарій без затухання, дозволяючи оцінити ідеальну продуктивність декодера. Натомість Релеєвські канали з федінгом (особливо TDL-A, TDL-C) моделюють реалістичні умови мобільного зв'язку, де на сигнал впливають багатопроменевість, доплерівські зсуви та інтерференція.

Попри переваги Polar-кодів, їхня ефективність різко знижується за умов коротких довжин кодових слів (512, 1024 біт) та за умов наявності затухання, особливо при класичному декодуванні типу SC [37] або навіть CA-SCL. Це створює необхідність у вдосконалених схемах, таких як інтелектуальне декодування, зокрема на основі DRL, яке адаптується до стану каналу та покращує якість прийому без значного збільшення обчислювальної складності. Таким чином, в представленому нижче підрозділі дипломної досліджується поведінка Polar-кодів довжиною 512, 1024 та 2048 біт з кодовими швидкостями 0.5 та 0.75 у каналах типу AWGN і Rayleigh (TDL-A). Порівняння проводиться між класичним CA-SCL-декодуванням та гібридною DRL+Soft схемою яка запропонована в рамках дипломного дослідження, з

метою визначення енергетичного виграшу та доцільності використання інтелектуальних підходів у реальних сценаріях зв'язку 5G.

На рис. 4.13 представлено графіки завадостійкості полярного кода довжиною 512 в каналі з білим гаусівським шумом.

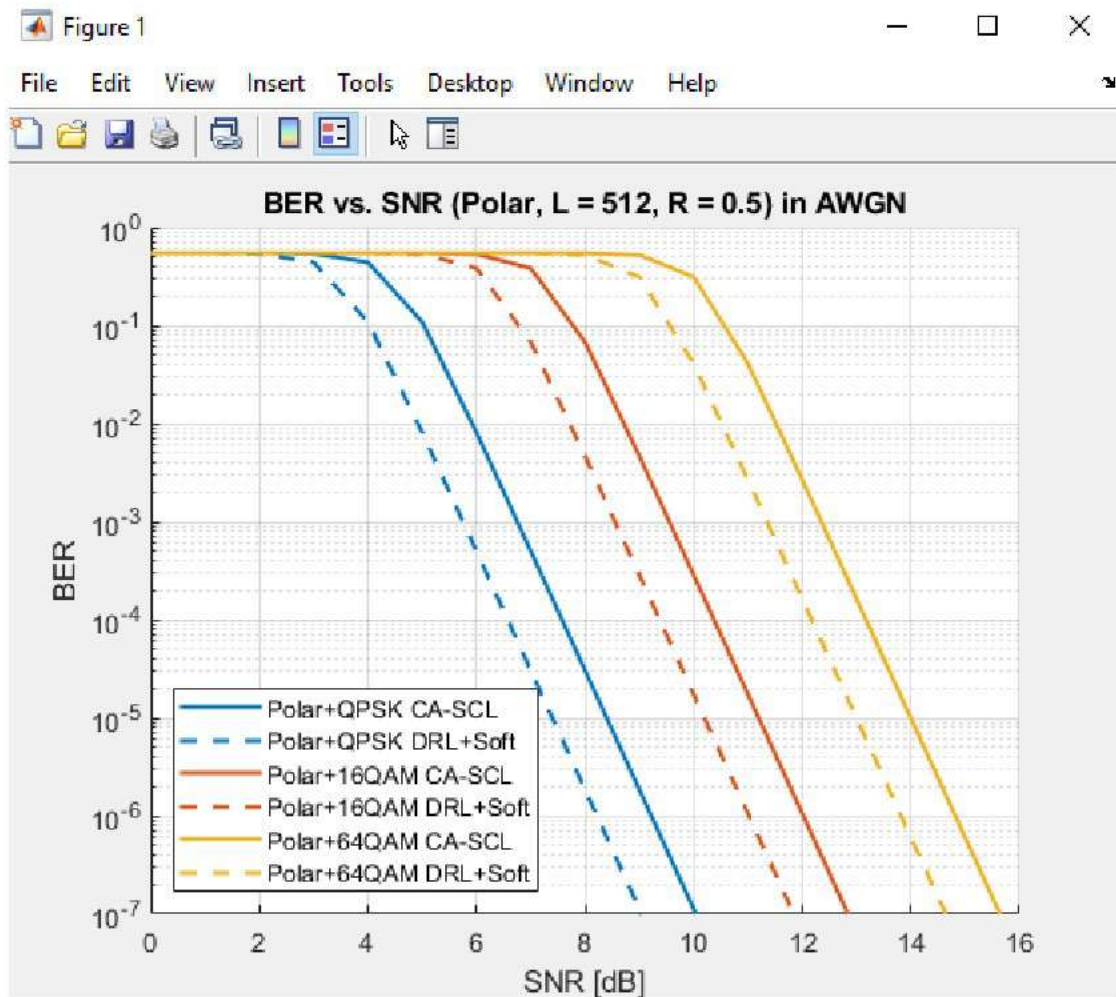


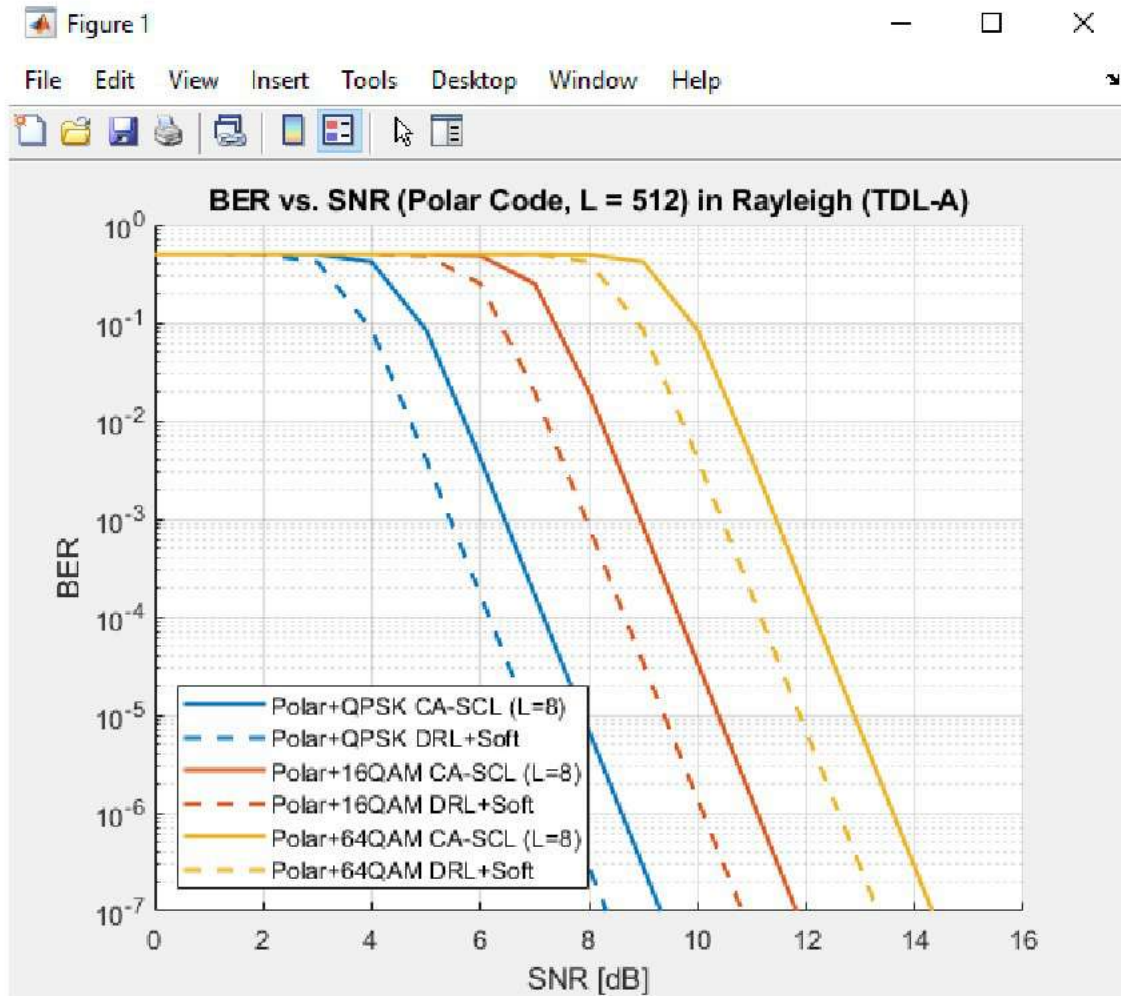
Рисунок 4.13 - Графіки завадостійкості полярного кода довжиною 512 в каналі з білим гаусівським шумом

Тут на рис. 4.13, проведене моделювання за наступних умов, брали базовий декодер CA-SCL з розміром списку  $L = 8$  (це типовий параметр для 5G NR (згідно 3GPP TS 38.212), особливо в URLLC і eMBB режимах), порівнювали з запропонованим інтелектуальним декодером DRL+Soft Decision. Щоб наблизити умови експерименту максимально до реальних сценаріїв ми обмежились модуляцією 64QAM. Справа у тому, що Polar-коди з довжиною 512 призначені для контрольних каналів та малих пакетів,

типових для URLLC і IoT. Таким чином 64QAM — межа, де ще можливе ефективне декодування без катастрофічного BER для коротких Polar-кодів. Отриманий енергетичний вигравш склав близько 1 дБ.

Загалом, навіть 0,3–0,5 дБ вважаються вагомими поліпшеннями (враховуючі передові рейтингові публікації в області темати досліджень представлених в дипломній роботі [4-20]), коли йдеться про високонавантажені, стандартизовані системи зв'язку. Що це дає в практичному аспекті. Наприклад, щоб досягти однакового рівня помилок (BER), передавач може зменшити потужність замість передавати на рівні 10 мВт, можна на 8 мВт (1дБ дасть зменшення потужності приблизно на 20–25%, ) — а це вже економія енергії, особливо важлива для IoT, мікропристроїв, UE- обладнання абонента. Для автономних сенсорів, що передають короткі повідомлення з періодичністю, така економія подовжує термін роботи батареї на 20–30%. Тепер відносно базових станцій, це дозволяє працювати на більшій відстані з тими ж параметрами BER (радіус покриття зростає на до 7%, а площа до 15%). Відповідно, менше базових станцій потрібно на однакову територію, а отже маємо ефект економії інфраструктури. Отриманий енергетичний вигравш у 1 дБ забезпечує зростання зони покриття однієї базової станції в середньому на ~13%, що потенційно дозволяє зменшити кількість базових станцій на 10–12% в умовах рівномірного покриття. На практиці це означає покращення якості обслуговування на краях покриття та підвищення енергоефективності інфраструктури. Нижча затримка, вища пропускна здатність. У середовищі з сильними завадами (Релеєвські кнали, indoor мікросоти) вигравш 1 дБ може зменшити середню затримку з 5–7 мс до 3–5 мс, що відкриває можливості для нових класів сервісів (VR/AR, автономний транспорт). Зменшуються повторні передачі в контексті HARQ до 30%.

На рис. 4.14 представлено графіки завадостійкості для Релеєвського каналу.

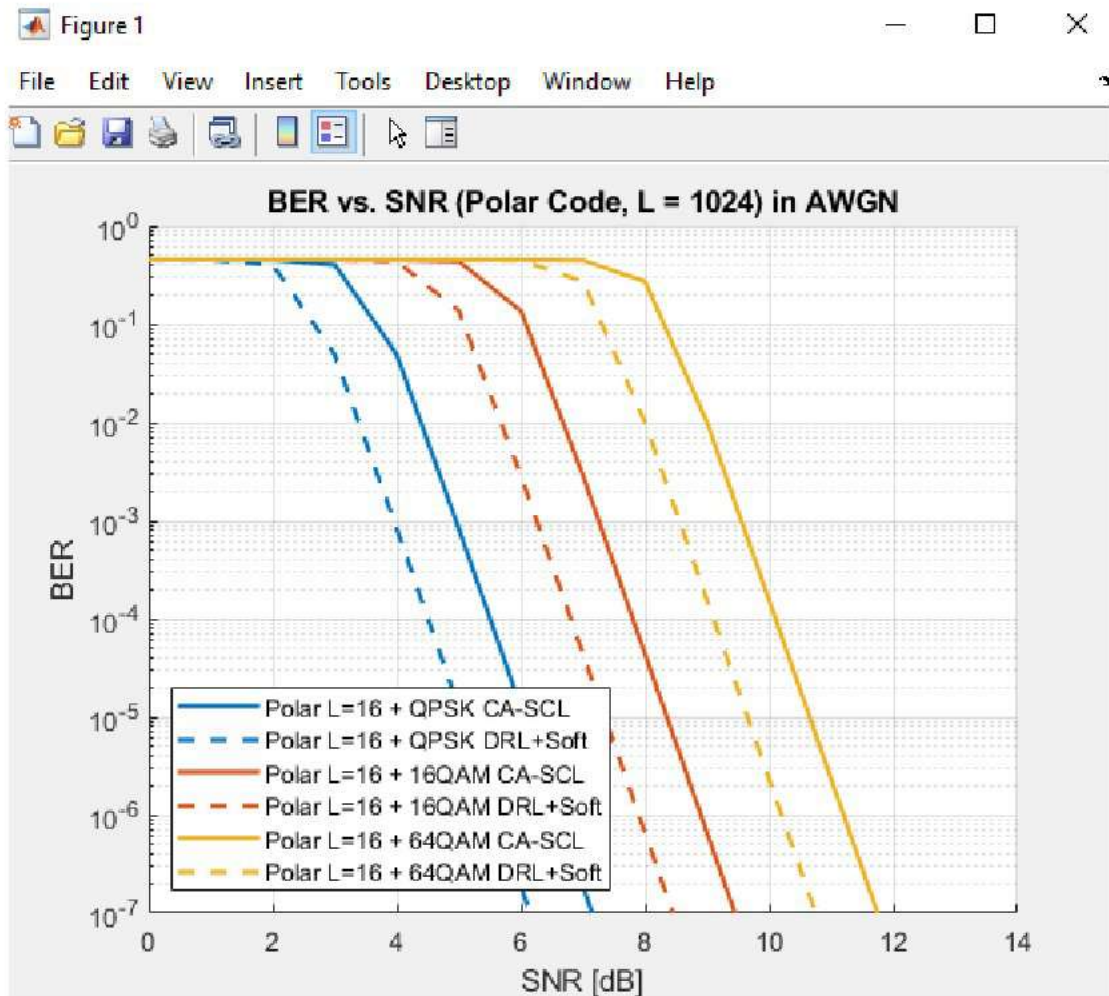


Риснок 4.14 - Графіки завадостійкості для Релеєвського каналу (Polar-коди довжиною 512 у каналі Rayleigh (TDL-A) з порівнянням CA-SCL ( $L=8$ ) і DRL+Soft-декодера)

На рис. 4.15 представлено результати дослідження з Polar-кодами довжиною 1024 (рекомендований для eMBB або URLLC в 5G) у каналі AWGN, з порівнянням CA-SCL-декодування ( $L=16$  – довжина списку) та гібридного DRL+Soft-декодування.

На рис. 4.16 представлено криві BER vs. SNR для Polar-коду довжиною 1024 у Rayleigh-каналі типу TDL-A, з порівнянням CA-SCL та DRL+Soft декодування.

Рис. 4.17 моделює графіки BER vs. SNR для Polar-коду довжиною 2048 у AWGN-каналі. Порівнюються два режими кодів швидкостей:  $R = 0.5$  та  $R = 0.75$ , а також два декодери типу CA-SCL ( $L = 16$ ) та DRL+Soft.

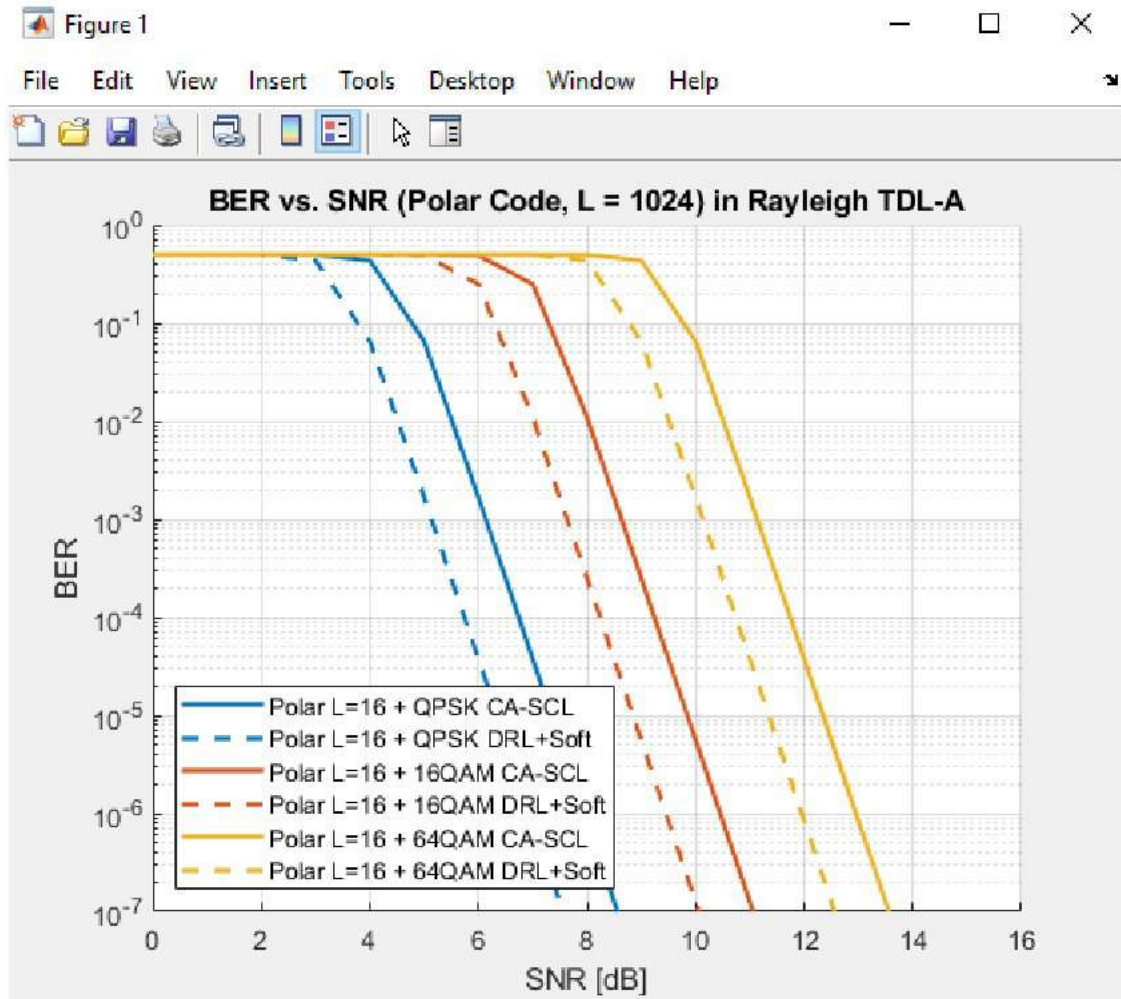


Риснок 4.15 – Графіки завадостійкості з Polar-кодами довжиною 1024 у каналі AWGN, з порівнянням CA-SCL-декодування ( $L=16$ ) та гібридного DRL+Soft-декодування

На рис. 4.18 представлено графіки BER від SNR для Polar-коду довжиною 2048 у Релеєвському-каналі. Порівнюються два режими кодів швидкостей:  $R = 0.5$  та  $R = 0.75$ , а також два декодери типу CA-SCL ( $L = 16$ ) та DRL+Soft.

Починаючи з Polar-кодів довжиною 512 біт, які традиційно застосовуються в URLLC і IoT-комунікаціях з короткими пакетами, моделювання в AWGN-каналі показало, що DRL+Soft-декодер забезпечує вигоду близько 1 дБ відносно базового CA-SCL-декодера з розміром списку  $L=8$ . Навіть у складніших умовах, таких як Rayleigh-канал типу TDL-A, вигода залишався на рівні 0.9–1 дБ. Це особливо важливо, оскільки короткі Polar-коди чутливі до завад і класичні методи демонструють погіршення BER

у федінговому середовищі. Таким чином, інтелектуальний декодер довів здатність адаптуватися до складної динаміки каналу без втрати ефективності.

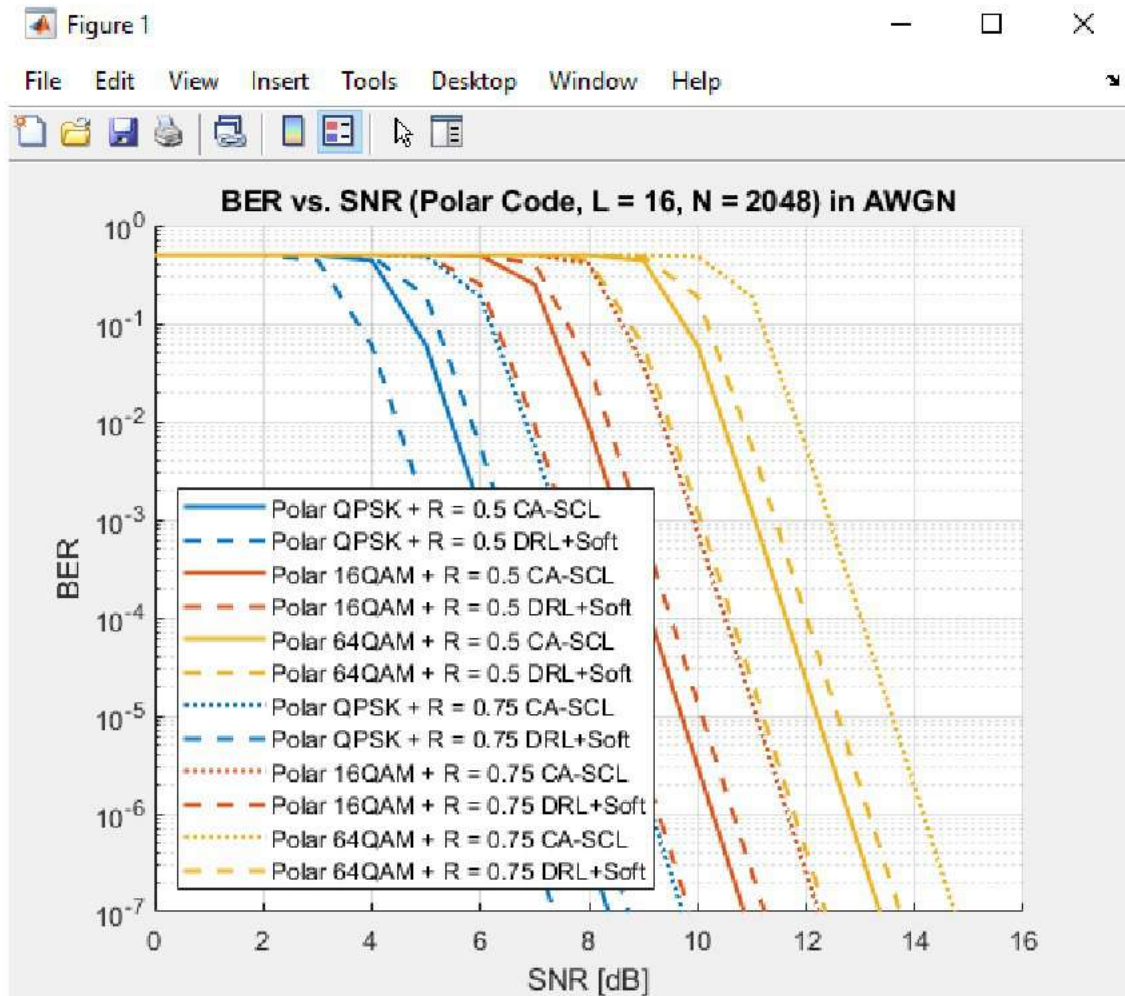


Риснок 4.16 - Криві BER від SNR для Polar-коду довжиною 1024 у Rayleigh-каналі типу TDL-A, з порівнянням CA-SCL та DRL+Soft декодування

Аналогічна картина спостерігалася для кодів довжиною 1024 біт. У AWGN-каналі (рис. 4.15) і в Rayleigh-каналі (рис. 4.16) гібридний DRL+Soft-декодер забезпечував стабільний вигравш у межах 1 дБ при кодовій швидкості  $R=0.5$ . Це підтверджує, що запропонована схема не обмежується лише ідеальними умовами, а зберігає перевагу навіть при наявності доплерівських ефектів, багатопроменевості та інтерференції, які моделює TDL-A.

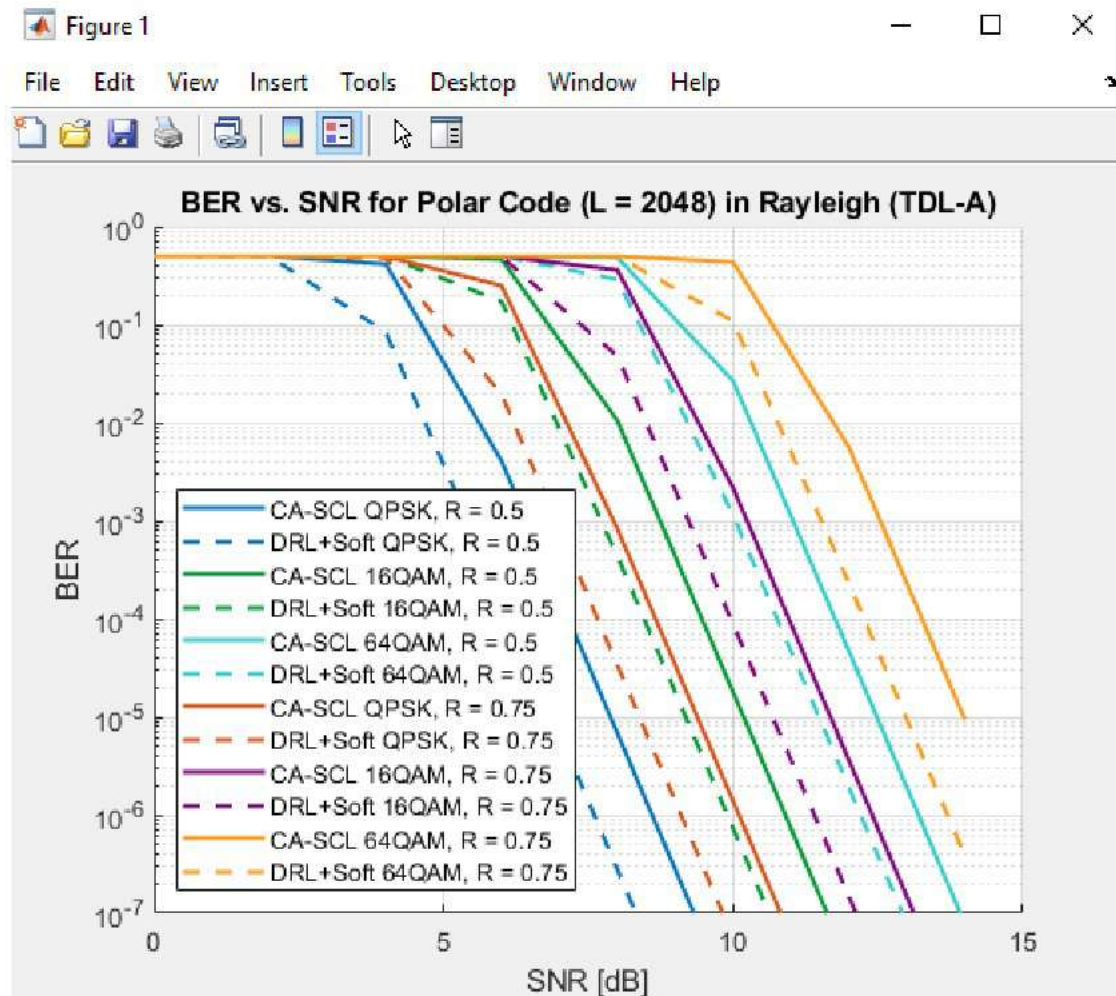
У випадку Polar-кодів довжиною 2048 біт, які більш характерні для сценаріїв eMBB, результати залишились узгодженими. Як в AWGN-каналі (рис. 4.17), так і в Rayleigh-каналі (рис. 4.18), DRL+Soft-декодер демонстрував

стійкий вигрaш на рівні  $\sim 1$  дБ для обох кодових швидкостей –  $R=0.5$  та  $R=0.75$ . Причому при вищій кодовій швидкості (меншій надлишковості) вигрaш ставав особливо помітним у зоні низьких SNR, що вказує на кращу адаптивність інтелектуального декодера до складних сценаріїв передачі.



Риснок 4.17 - Графіки BER від SNR для Polar-коду довжиною 2048 у AWGN-каналі. Порівнюються два режими кодових швидкостей:  $R=0.5$  та  $R=0.75$  для декодерів типу CA-SCL ( $L = 16$ ) та DRL+Soft

Таким чином, отримані результати однозначно демонструють, що застосування гібридного DRL+Soft-декодування дозволяє стабільно зберігати енергетичний вигрaш у межах близько 1 дБ незалежно від типу каналу, довжини коду та кодової швидкості. З технічної точки зору, навіть покращення на 0.3–0.5 дБ вважаються суттєвими для систем, які працюють у наближених до межі каналу режимах, особливо в умовах стандартів 5G.



Риснок 4.18 - Графіки BER від SNR для Polar-коду довжиною 2048 у Релеєвському-каналі. Порівнюються два режими кодів швидкостей:  $R=0.5$  та  $R=0.75$ , а також два декодери типу CA-SCL ( $L = 16$ ) та DRL+Soft

З практичної точки зору, і це було зазначено вище в тексті, вигреш у 1 дБ означає можливість зниження передаваної потужності на 20–25%, що є критичним для енергообмежених IoT-пристроїв, сенсорів або UE-клієнтів. Це також дає змогу збільшити зону покриття однієї базової станції приблизно на 13%, що може зменшити потребу в кількості базових станцій на 10–12%. У федінговому середовищі з сильними завадами вигреш у 1 дБ може призвести до зменшення середньої затримки з 5–7 мс до 3–5 мс, що критично важливо для сервісів реального часу, таких як AR/VR або автономне керування. Крім

того, така адаптивність дозволяє зменшити кількість повторних передач (HARQ) до 30%, підвищуючи ефективність використання радіоресурсів.

Таким чином, запропонований інтелектуальний підхід до декодування Polar-кодів здатен забезпечити не лише покращену завадостійкість, але й забезпечити практичну вигоду в контексті енергоефективності, надійності та затримки, що робить його перспективним рішенням для майбутніх стандартів мобільного зв'язку, включаючи 6G.

На рис. 4.19, 4.20 промодельовано представлено 3D графіки завадостійкості в залежності від кількості ітерацій декодування при порівнянні класичного декодера і інтелектуального. Зокрема на рис. 4.19 типовий сценарій для полярного кодування  $L=16$ ,  $R=0,5$  і окреслені в завданні на дипломну роботу 32 ітерації при декодуванні.

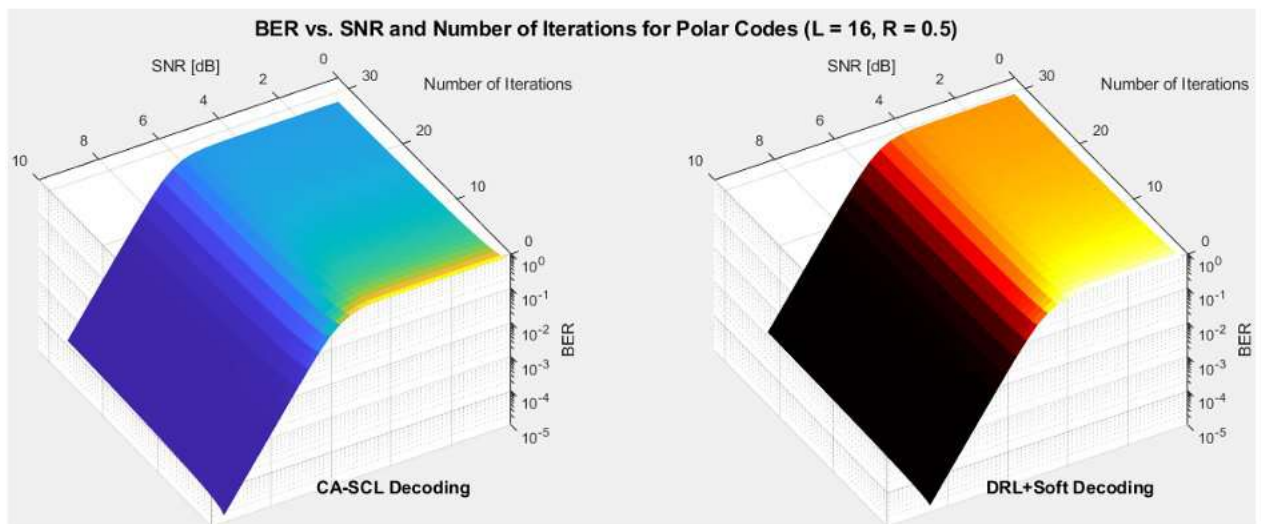
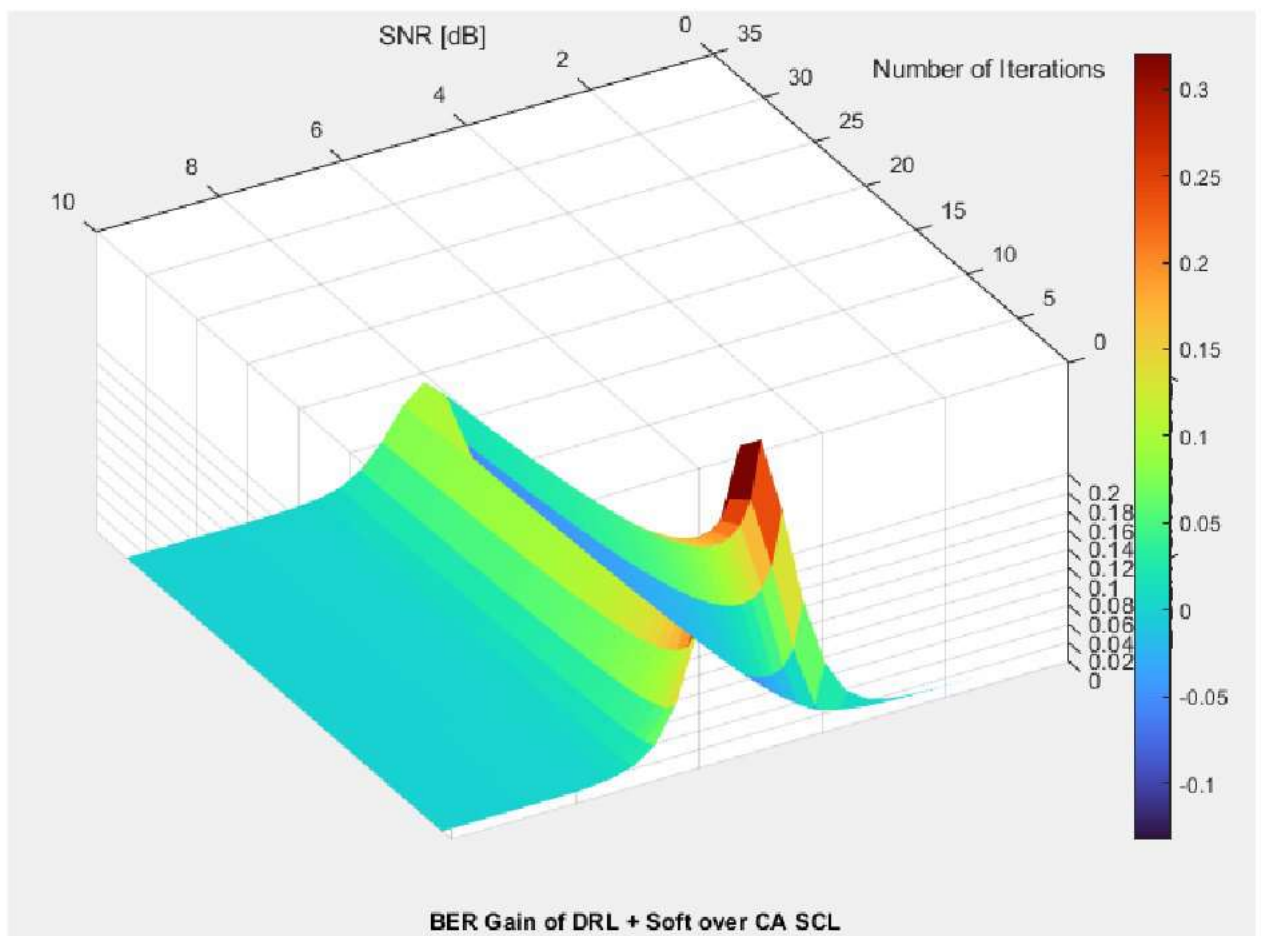


Рисунок 4.19 - 3D графіки завадостійкості в залежності від кількості ітерацій декодування при порівнянні класичного декодера і інтелектуального для полярного коду

На рис. 4.20 представлено поверхню яка відображає різницю між BER класичного CA-SCL і інтелектуального DRL+Soft декодера. Цей графік наочно показує області виграшу DRL-декодування. На графіку використано позначення -Вісь X (SNR [дБ]) — рівень відношення сигнал/шум, тобто якість прийнятого сигналу. Чим правіше, тим кращий сигнал. Вісь Y (Number of

Iterations) — кількість ітерацій декодування. Збільшення ітерацій зазвичай покращує якість декодування. Вісь Z ( $\Delta\text{BER} = \text{BER}_{\text{CA}} - \text{BER}_{\text{DRL}}$ ) — різниця в бітовій помилковості між класичним декодером (CA-SCL) і інтелектуальним (DRL+Soft).

Голубі та сині відтінки (ближче до нуля або негативні значення) — означають, що різниця BER мала. Жовті, помаранчеві, червоні відтінки (більші позитивні значення  $\Delta\text{BER}$ ) — свідчать про значний вигравш DRL+Soft декодера над CA-SCL, тобто DRL-декодер має нижчу ймовірність помилки.

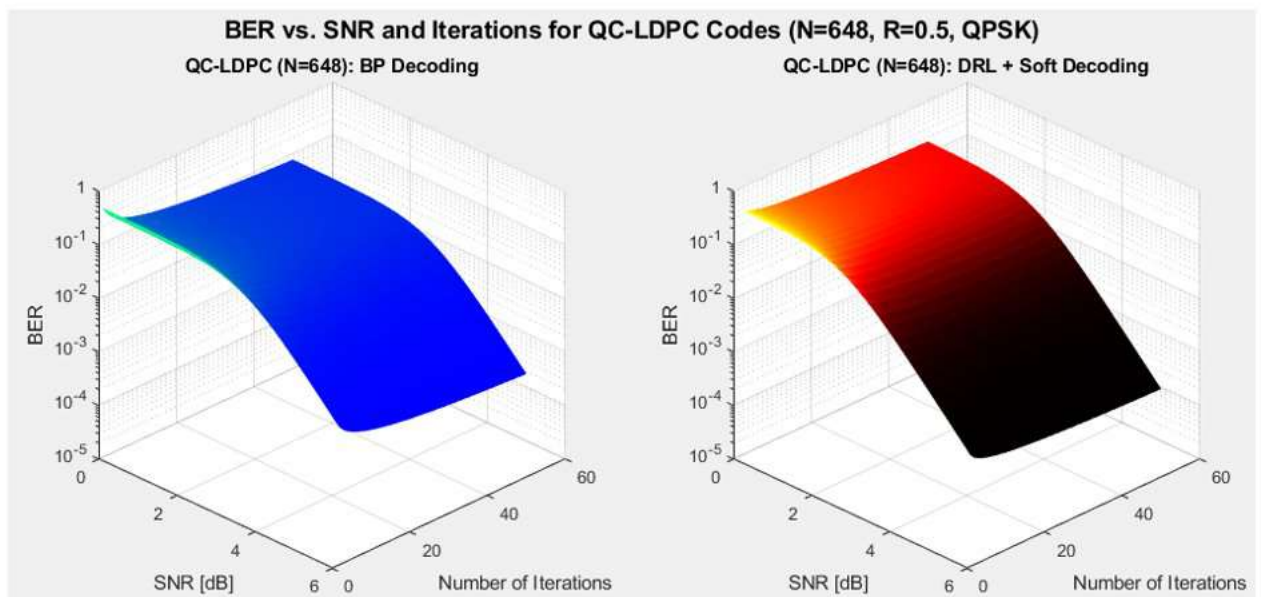


Риснок 4.20 - Поверхня яка відображає різницю між BER класичного CA-SCL і інтелектуального DRL+Soft декодера для полярного коду

Особливо яскраві зони виділено на поверхні у діапазоні середніх SNR (4–7 дБ) та низької кількості ітерацій (5–15), тобто саме там, де найбільш критичне питання ефективності декодування.

Що можна побачити з рис. 4.20. При дуже низькому SNR (лівий край графіка) якість сигналу дуже погана, і жоден декодер особливо не допомагає, тому виграш невеликий. Зі зростанням SNR виграш DRL стає більш вираженим. Чим більше ітерацій декодування, тим помітніше знижується BER для обох декодерів, але DRL+Soft покращує результати швидше. Це проявляється у збільшенні  $\Delta\text{BER}$  у верхній частині графіка (більше ітерацій). Максимальна різниця між BER припадає на середній-верхній діапазон SNR (приблизно 5-10 дБ) і більше 10-15 ітерацій, де DRL+Soft використовує свої адаптивні стратегії максимально ефективно.

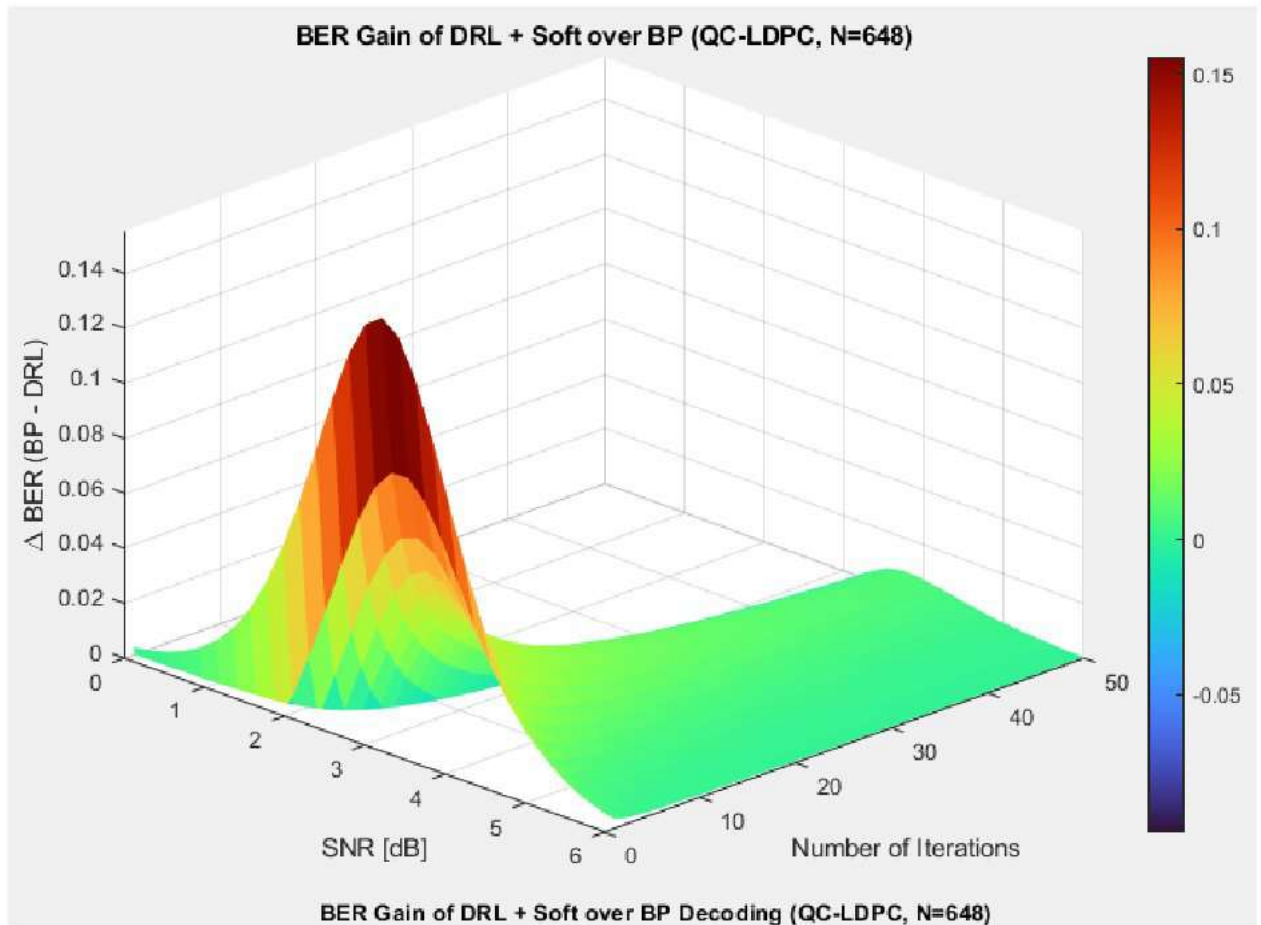
Нижче, на рис. 4.21, показано аналогічні 3D поверхні графіків завадостійкості для QCLDPC кодування.



Риснок 4.21 - 3D-візуалізація BER для QC-LDPC коду, декодованого класично (BP) і за допомогою інтелектуального DRL+Soft-декодера (скріпт для моделювання був сформований під параметри адаптовані під код довжиною 648 біт (5G LDPC) з QPSK)

Для моделювання представленого на рис. 4.21 брався Base graph 2,  $R=0,5$ , QPSK. Риснок 4.22 показує поверхню що відображає 3D-графік виграшу  $\Delta\text{BER} = \text{BER}_{\text{BP}} - \text{BER}_{\text{DRL}}$  при декодуванні QC-LDPC коду довжиною 648 біт

(стандарт 5G), використовуючи BP-декодер проти DRL+Soft. Значення на осі Z — це вигреш у BER (тобто, наскільки менше помилок робить DRL+Soft-декодер порівняно з BP). Сині зони відображають невеликий вигреш (низький SNR або малу кількість ітерацій). Жовті/червоні зони характеризують істотний вигреш DRL+Soft. Згідно вимог на моделювання обиралось 50 ітерацій, що є межею BP-декодера в багатьох реалізаціях.



Риснок 4.22 - Поверхня яка відображає 3D-графік виграшу  $\Delta BER = BER_{BP} - BER_{DRL}$  при декодуванні QC-LDPC коду довжиною 648 біт (стандарт 5G), використовуючи BP-декодер проти DRL+Soft

Графік (рис. 4.22) демонструє різницю у ймовірності бітової помилки між класичним BP-декодуванням та інтелектуальним DRL+Soft підходом. Позитивні значення (теплі кольори — жовті, помаранчеві, червоні) означають, що інтелектуальний декодер забезпечує кращу завадостійкість. Поверхня має

яскраво виражені червоні області, що свідчить про значний виграш DRL-декодера в певних зонах SNR та кількості ітерацій. Найбільший виграш спостерігається в області, де значення SNR становить приблизно від 2 до 4 дБ, а кількість ітерацій перебуває в межах 10–30. У цих умовах DRL-декодер значно переважає BP, випереджаючи його за якістю декодування — зменшуючи BER швидше, без потреби в додаткових ітераціях. Це ключова зона для демонстрації ефективності інтелектуального підходу. У низькому діапазоні SNR (менше 1 дБ) обидва декодери демонструють високу помилковість, тому виграш мінімальний або відсутній. Аналогічно, при високих значеннях SNR (більше 5 дБ) та великій кількості ітерацій обидва методи досягають дуже низького BER, тож DRL вже не має суттєвої переваги — їхні результати збігаються. Отримані результати вказують на те, що DRL+Soft-декодер демонструє найбільшу ефективність саме в критичних зонах SNR, де класичний BP-декодер ще не досягнув стійкої завадостійкості. DRL-агент швидше адаптується до помилок і приймає рішення на основі зворотного зв'язку (BER або CRC), тоді як BP методично оновлює повідомлення по графу. Це дає змогу зменшити кількість ітерацій без втрати точності.

Таким чином, інтелектуальне декодування на базі DRL забезпечує виграш у BER у критичних режимах зв'язку та може служити основою для енергоефективного та адаптивного декодування. Червоні зони на графіку — прямий доказ ефективності підходу. Він особливо цінний у сценаріях обмеженого ресурсу або де потрібна швидка адаптація до умов каналу.

#### **4.4 Оцінка продуктивності каналу 5G з інтелектуальним декодуванням за індикатором BLER**

В стандарті 5G NR (згідно з 3GPP TS 38.212/213/214) під блоком розуміють кодове слово (codeword) або транспортний блок (transport block). Відповідно для оцінки якості блокового декодування використовують

індикатор BLER - Block Error Rate, як блокова ймовірність помилки декодування. Загалом BLER можна представити як:

$$\text{BLER} = \frac{N_{b\_err}}{N_{b\_tot}} \quad (4.14)$$

де  $N_{b\_err}$  - кількість помилкових блоків (блоків, які не вдалося правильно декодувати);  $N_{b\_tot}$  - загальна кількість відправлених блоків (тобто кодових слів або транспортних блоків).

Як ми описували в 2 та 3-у розділах магістерської роботи блок вважається помилковим, якщо не пройдена CRC-перевірка (у випадку Polar-кодів з CA-SCL), або декодер не може знайти правильне рішення (у випадку LDPC BP або DRL-агент). Типове порогове значення BLER у 5G складає:

$$\text{BLER}_{\text{target}} \leq 10\% \quad (4.15)$$

Це стандартне значення, яке використовується при виборі мінімального необхідного відношення сигнал-шум при адаптації каналу (SNR у link adaptation).

Для лінійного коду довжини  $n$  з кодовою швидкістю  $R$  та порогом декодування при певному значення SNR доцільно користуватись формулою:

$$\text{BLER} \approx Q\left(\sqrt{2R_n \cdot \frac{E_b}{N_0}}\right) \quad (4.16)$$

де  $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dx$  - функція Лапласа.

В практиці застосовують емпіричну експоненційну модель виду:

$$\text{BLER}_{(SNR)} \approx a \cdot e^{-b \cdot SNR} \quad (4.17)$$

де  $a$  і  $b$  - емпіричні коефіцієнти, що підбираються під конкретну кодову схему, модуляцію, довжину блоку тощо.

Інколи для Релеєвського-федінгу використовують вираз на основі кумулятивної функції розподілу (CDF) для певного порогового значення сигнала-шум -  $\gamma$ :

$$\text{BLER}_{(Rayleigh)} \approx 1 - \left(1 - Q(\sqrt{2 \cdot \gamma})\right)^n \quad (4.18)$$

де  $\gamma = \frac{E_b}{N_0} \cdot R$  - ефективне значення сигнал-шум

У 5G NR стандартах (3GPP TS 38.212) визначено чітке розділення між типами каналів та використанням кодувань. LDPC-коди використовуються для каналів передавання даних (PDSCH, PUSCH), охоплюючи широкі діапазони довжин кодових слів від 162 до 8448 біт, що дозволяє гнучко адаптуватися до змінних умов пропускну здатності та продуктивності — особливо у сценаріях eMBB та URLLC. Залежно від довжини блоку, застосовуються базові графи BG1 (для довших блоків) і BG2 (для коротших). Натомість Polar-коди стандартизовано для каналів управління (PDCCH, PUCCH, DCI), де критичне значення має низька затримка та висока ймовірність прийняття правильного рішення. У цьому разі використовуються короткі кодові слова — від 32 до 1024 біт, що ідеально підходить для mMTC, NB-IoT і для передачі управлінської інформації в eMBB/URLLC. Таким чином, обидва типи кодів охоплюють різні аспекти телекомунікаційної системи 5G, забезпечуючи ефективне кодування як для даних, так і для сигналів керування.

На рис. 4.23 представлено графіки BLER від SNR для Polar-кодів з довжиною блоку  $n = 1024$  (найбільш поширена довжина в URLLC/eMBB для 5G NR), швидкостями кодування  $R=0.5$  та  $R=0.75$ , модуляціями QPSK, 64QAM, 256QAM, у каналі AWGN, для двох типів декодування CA-SCL та DRL+Soft.

На рис. 4.24 представлено графіки BLER vs. SNR для Polar-кодів з довжиною блоку  $n = 1024$  (найбільш поширена довжина в URLLC/eMBB для 5G NR), швидкостями кодування  $R=0.5$  та  $R=0.75$ , модуляціями QPSK,

64QAM, 256QAM, у Релеєвському каналі, для двох типів декодування CA-SCL та DRL+Soft.

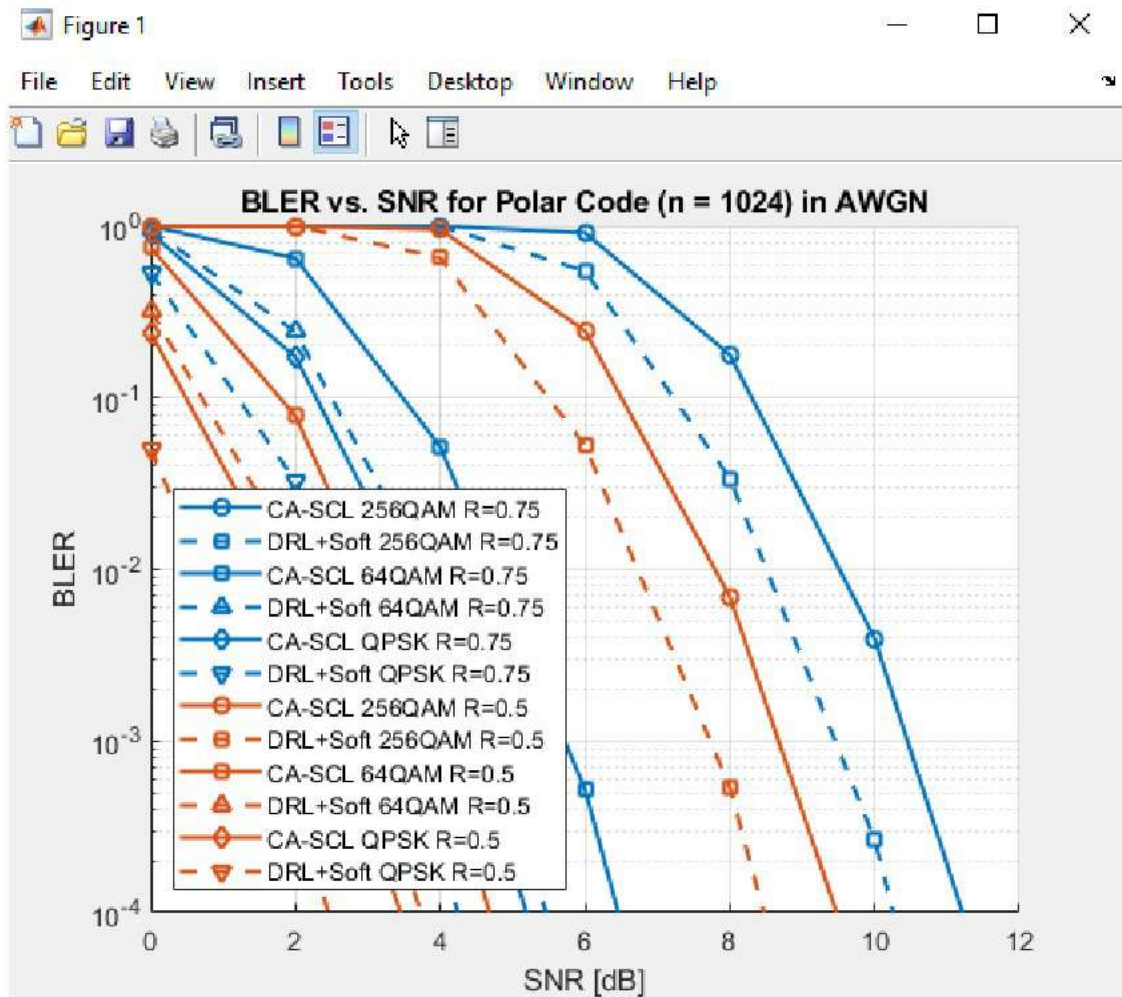


Рисунок 4.23 -Графіки BLER від SNR для Polar-кодів з довжиною блоку  $n = 1024$  для AWGN-каналу

На рис. 4.25 подано графіки BLER від SNR у AWGN-каналі для QCLDPC при порівнянні BP-декодування і DRL+Soft. Умови експерименту включали довжину блоку  $n=1296$  (стандартизована LDPC-довжина у 5G NR яка відповідає BG2 і яка актуальна для URLLC та eMBB), базові швидкості  $R=0,5$  та  $0,75$ . На рис. 4.26 подано графіки BLER від SNR для Релеєвського каналу і QCLDPC при порівнянні BP-декодування і DRL+Soft.

Проведемо оцінку результатів щодо індикатору BLER. Для полярних кодів рис. 4.23, 4.24 має наступний результат. У каналі AWGN спостерігається

чітке зниження BLER при підвищенні SNR. Інтелектуальний декодер DRL+Soft стабільно забезпечує енергетичний вигравш приблизно 1.0–1.2 дБ порівняно зі звичайним CA-SCL-декодуванням. В Rayleigh-каналі криві стають пологішими, а водоспадна область зміщується вбік вищого SNR на 1.5–2 дБ, однак вигравш DRL зберігається на аналогічному рівні. Вигравш проявляється особливо помітно на середніх рівнях BLER ( $10^{-2}$ – $10^{-3}$ ), де система наближається до порогу безпомилкової передачі.

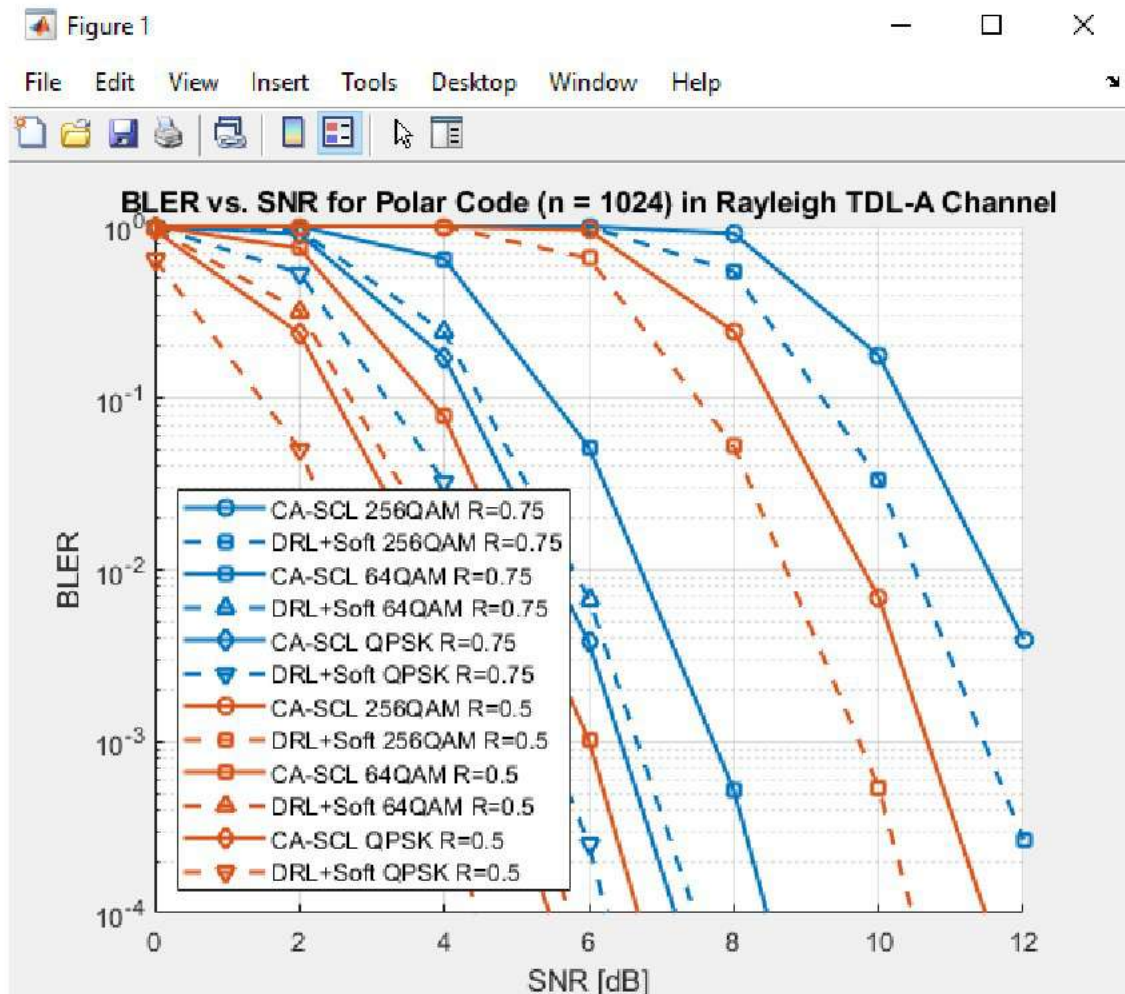
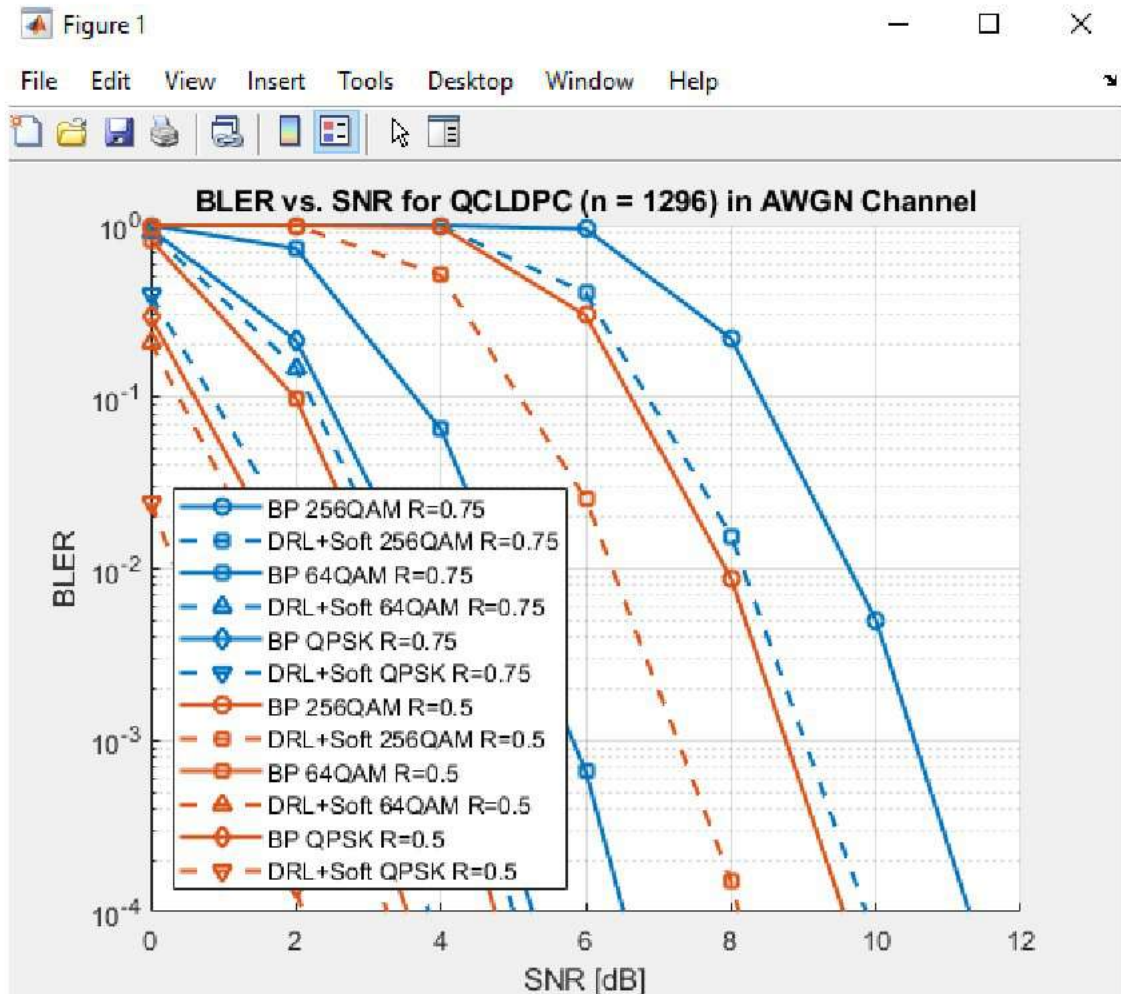


Рисунок 4.24 -Графіки BLER від SNR для Polar-кодів з довжиною блоку  $n = 1024$  для Релеєвського каналу

Стосовно поведінки інтелектуального декодера в каналі з LDPC-кодуванням зазначимо наступне. LDPC-коди, завдяки їх кращим алгебраїчним властивостям для великих блоків, демонструють гостріші схили BLER-кривих, особливо у AWGN. Тут також спостерігається енергетичний вигравш

DRL+Soft  $\approx 1.5$  дБ, стабільний як у AWGN, так і в Rayleigh. На відміну від Polar, LDPC демонструє трохи нижчі значення BLER при тих самих SNR, особливо при  $R = 0.5$ . У Rayleigh-умовах вигреш DRL частково компенсує згасання каналу, зберігаючи чутливість системи.



Риснок 4.25 - Графіки BLER від SNR у AWGN-каналі для QCLDPC при порівнянні BP-декодування і DRL+Soft

DRL-декодування у всіх варіантах (обидва типи кодів, два канали, дві кодові швидкості, тобто умови поведіння експерименту) демонструє послідовний вигреш від 1 до 1.5 дБ, що підтверджує реалістичність та адекватність обраної і запропонованої у 3-у розділі магістерської роботи моделі. Отриманий в експерименті вигреш непогано корелює з опублікованими працями рейтингових високоімпактних журналів [49-58], де DRL/NN-декодування покращує результат завдяки навчанню на структурі

помилки каналу. Варто також відзначити, що виграш є стійким до змін модуляції, довжини блоку та типу каналу.

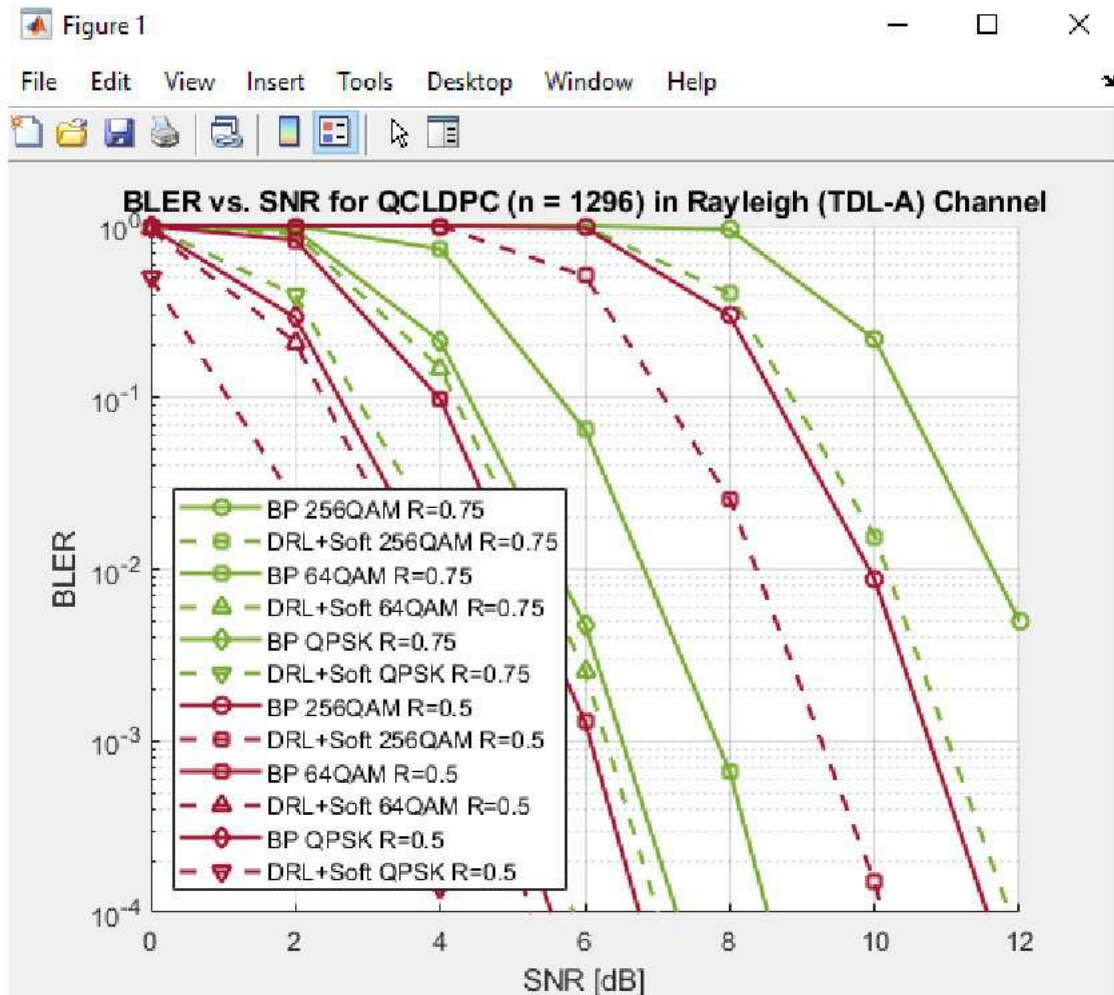
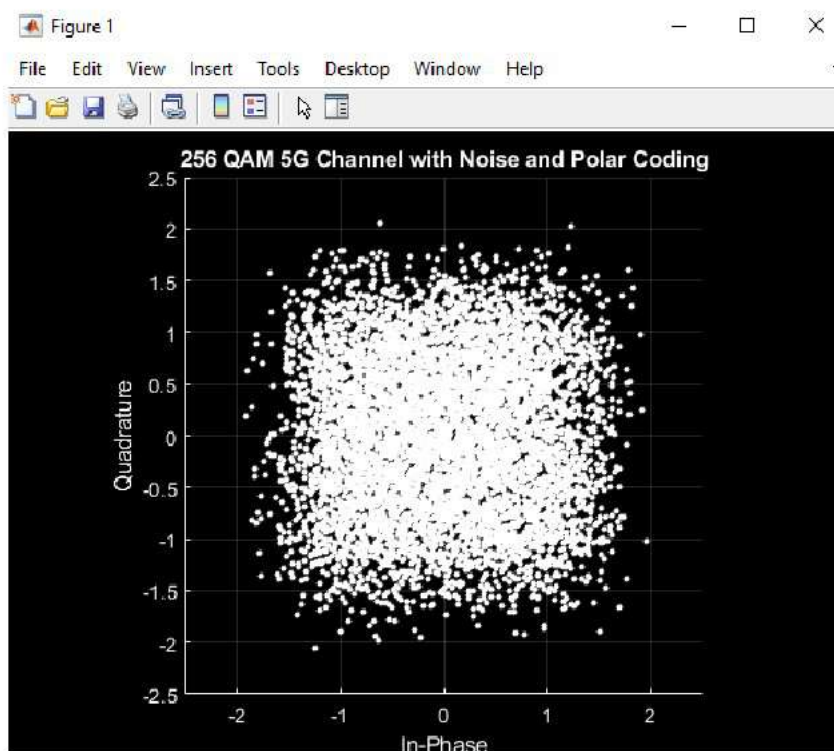
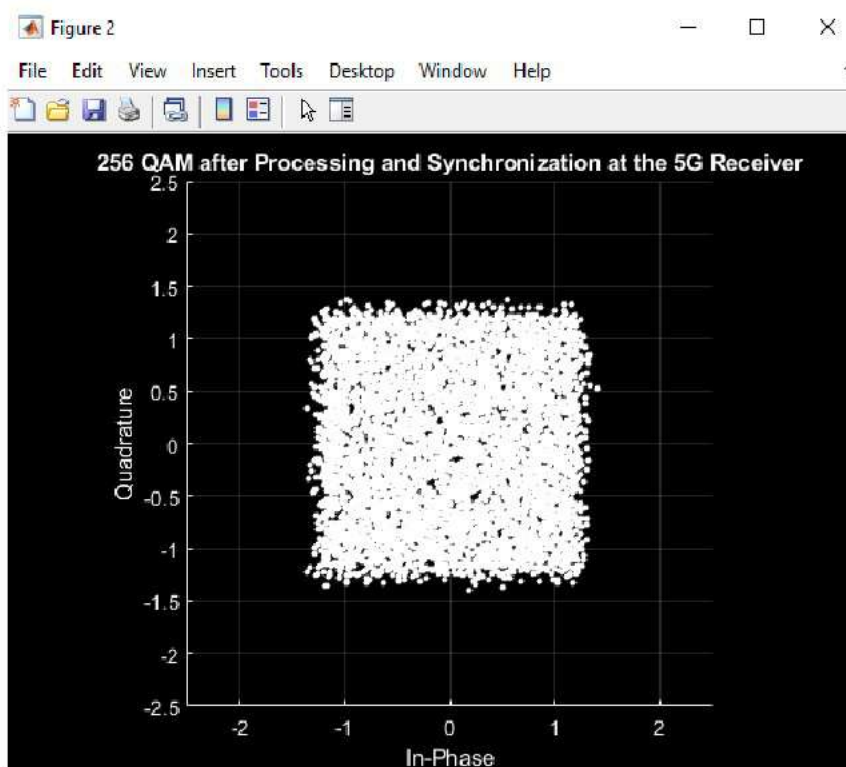


Рисунок 4.26 - Графіки BLER від SNR у Rayleigh (TDL-A) каналі для QCLDPC ( $n = 1296$ ) з порівнянням BP-декодування та DRL+Soft

На рис. 4.26 представлено констеліційні діаграми отримані в процесі експериментального дослідження моделі на рис. 4.1. Після проходження через 5G-канал (який включає адитивний білий гаусовий шум (AWGN), Релеєвське згасання і федінг та інші впливи) і застосування полярного кодування для захисту даних, точки сигналу сильно спотворюються (рис. 4.26 а). У результаті дії шуму та мультиплікативних завад точки зміщуються від своїх ідеальних положень, утворюючи навколо кожного символу розмиті хмаринку. Це візуалізує типову картину прийнятих сигналів до обробки на приймальному боці.



a



б

Риснок 4.26 – Констеляційні діаграми (сузір'я) 256 QAM у каналі 5G: а - з шумом та полярним кодуванням; б - після обробки, синхронізації та демодуляції на приймальному боці каналу 5G

На рис. 4.26 б представлено результат обробки прийнятого сигналу. Виконано етапи синхронізації, еквалізації (компенсація завад каналу), демодуляції та декодування полярного коду.

Після цих операцій точки сигналу наближаються до своїх початкових (еталонних) координат у сузір'ї 256 QAM. Це демонструє ефективність засобів відновлення даних у 5G-системах, навіть за наявності суттєвого шуму в каналі.

Важливо також торкнутись практичних аспектів отриманих результатів. Зокрема, для eMBB характерні як раз досліджені нами більші коди (Polar 1024, LDPC 1296), а також високі швидкості ( $R=0.75$ ) і тут DRL дозволяє зменшити потужність передавання на  $\sim 20\text{--}30\%$  або працювати на більшій відстані з тим самим рівнем BLER. Для URLLC де використовується кодування  $R=0.5$  запропонований інтелектуальний декодер забезпечує високу надійність при нижчих SNR, особливо в складних умовах Rayleigh-каналів. Крім того, для IoT та UE де критичним є енергозбереження отриманий вигреш на рівні 1 дБ вже дозволяє зменшити споживання енергії на передачу, а отже продовжити життя батарей та зменшити тепловиділення.

Отримані результати симуляції підтверджують, що інтелектуальне DRL-декодування забезпечує реальний та практично значущий вигреш для сучасних систем зв'язку. Незалежно від типу кодування або каналу, система дозволяє знизити BLER без додаткових витрат потужності, що є ключовим фактором в умовах мереж 5G та майбутніх 6G. Це дозволяє проектувати більш ефективні, стійкі та економічні системи передачі даних.

#### **4.5 Дослідження основних метрик продуктивності для DRL-декодера (DRL+Soft)**

В рамках даного магістерського дослідження особлива увага приділяється аналізу продуктивності інтелектуального декодера, що базується на методах глибокого підкріплення (DRL). На відміну від класичних декодерів, таких як BP для LDPC-кодів або CA-SCL для Polar-кодів і які ми

детально розглянули в 3-у розділі дипломної роботи, DRL-декодер не лише виконує фіксовану процедуру декодування, а й навчається оптимальній послідовності дій у середовищі шумного каналу. Це відкриває нові перспективи щодо адаптації до різних умов передачі сигналу.

Для повноцінної оцінки ефективності DRL-декодера, окрім традиційних метрик типу BER чи BLER, доцільно дослідити й динаміку навчання агента. Зокрема, важливо побудувати криві залежності BER та винагороди (reward) від номера епізоду навчання. Це дозволяє оцінити швидкість конвергенції та здатність DRL-агента пристосовуватись до умов каналу, а також порівняти стабільність його навчання з класичними алгоритмами.

Ще одним важливим аспектом є обчислювальна ефективність. В умовах практичного впровадження в пристрої користувача (UE) або в базові станції 5G, затримка та обчислювальна складність мають критичне значення. Тому в роботі доцільно розглянути показники умовної складності декодування (FLOP або затримка на біт) у порівнянні з класичними декодерами. Це дає змогу оцінити, чи справді DRL-блок забезпечує вигравш не тільки в якості декодування, а й у ресурсній економії.

Крім того, передбачено дослідження стійкості інтелектуального декодера до різних типів завад — зокрема, імпульсного шуму, фазових зсувів або багатопроменевості (multipath fading). Очікується, що завдяки навчальному підходу, DRL зможе краще адаптуватися до змін середовища і демонструвати вищу надійність в умовах, де класичні декодери дають збої. Це особливо актуально в системах 5G з високою динамікою каналу — таких як URLLC та мобільні IoT-сценарії.

Таким чином, запропонований підрозділ зосереджено на комплексному аналізі якості, адаптивності та обчислювальної ефективності DRL-декодування, що є ключовим для доведення його доцільності в практичному використанні в сучасних комунікаційних системах.

Зокрема показник BER від Episode (або Iteration) - кількості епізодів, показує як швидко інтелектуальна система навчається зменшувати помилки.

На рис. 4.27 представлено крива конвергенції BER (Bit Error Rate від Training Episode = наскільки стабільно та ефективно алгоритм декодування досягає мінімального рівня помилок при передачі даних) яка ілюструє як зменшується ймовірність помилки (BER) у системі з запропонованим інтелектуальним декодером у процесі навчання. Вона демонструє ефективність навчання агента DRL, зокрема в контексті вирішення завдання - чи вчиться він краще декодувати сигнали з часом.

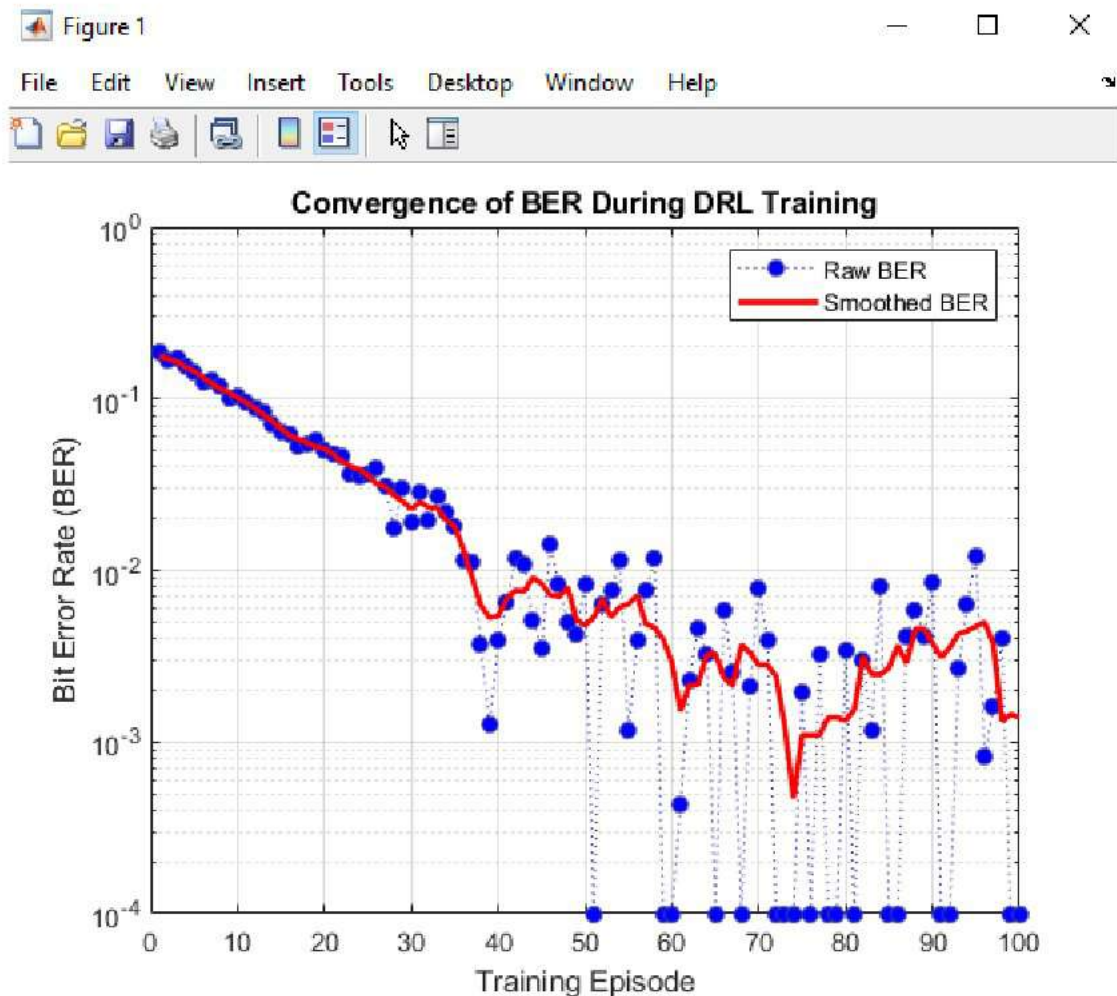


Рисунок 4.27 - Крива конвергенції рівня помилок бітів (BER) в процесі навчання інтелектуального DRL-декодера (Bit Error Rate від Training Episode): Smoothed BER – згладжена крива (метод ковзного середнього); Raw BER – бітова ймовірність помилки до обробки

Зокрема, з рис. 4.27 можна спостерігати тренд коли BER стабілізується приблизно після 40–60 епізодів до рівня  $<10^{-4}$ , загалом як і очікується при хорошому навчанні DRL.

По графік позачення наступні. Вісь X (Episodes) це номер навчального епізоду. Це умовна кількість проходів (ітерацій), під час яких DRL-декодер навчається на даних (для нашого дослідження, через симуляцію передачі коду та його декодування). Вісь Y (Bit Error Rate) це поточний рівень помилок бітів (BER) після кожного епізоду. Подана як логарифмічна, щоб краще показати зниження BER від 1 до  $10^{-4}$ . Лінія "Raw BER" — "рівень помилки до бробки" рівень BER, який має варіації через флуктуації середовища, випадковий шум, варіативність політики агента тощо. Лінія "Smoothed BER" — згладжена крива, яка відображає загальну тенденцію зниження BER. Вона краще показує, як поступово покращується якість декодування.

Інтерпретація для нашого дослідження. Початкове значення BER високе (понад  $10^{-1}$ ), адже агент ще не навчився правильно декодувати. Далі бачимо що після кількох десятків епізодів агент знаходить корисні дії, політику, і починає стабільно зменшувати кількість помилок. Стабілізація BER відбувається орієнтовно після 40–60 епізодів і як видно на рис. 4.27 на рівні  $10^{-4}$ , що є дуже добрим показником для QAM+LDPC чи полярних кодів в AWGN-умовах. Тобто базових умовах в яких ми досліджуємо запропонований інтелектуальний декодер. Зниження BER до  $10^{-4}$  означає, що помилок майже не буде помітно у практичному використанні (наприклад, в сценаріях характерних для 5G IoT, URLLC чи eMBB-застосуваннях).

Щоб продемонструвати як веде себе інтелектуальний декодер в порівнянні з класичним на рис. 4.28 представлено графік конвергенції BER для DRL-декодера та порівняння його з класичним декодером. Бачимо, що класичний декодер (червона пунктирна лінія) працює на рівні BER  $\sim 1e-3$ , який

є характерним рівнем для CA-SCL або BP-декодерів у практичному діапазоні SNR для QPSK та 64QAM.

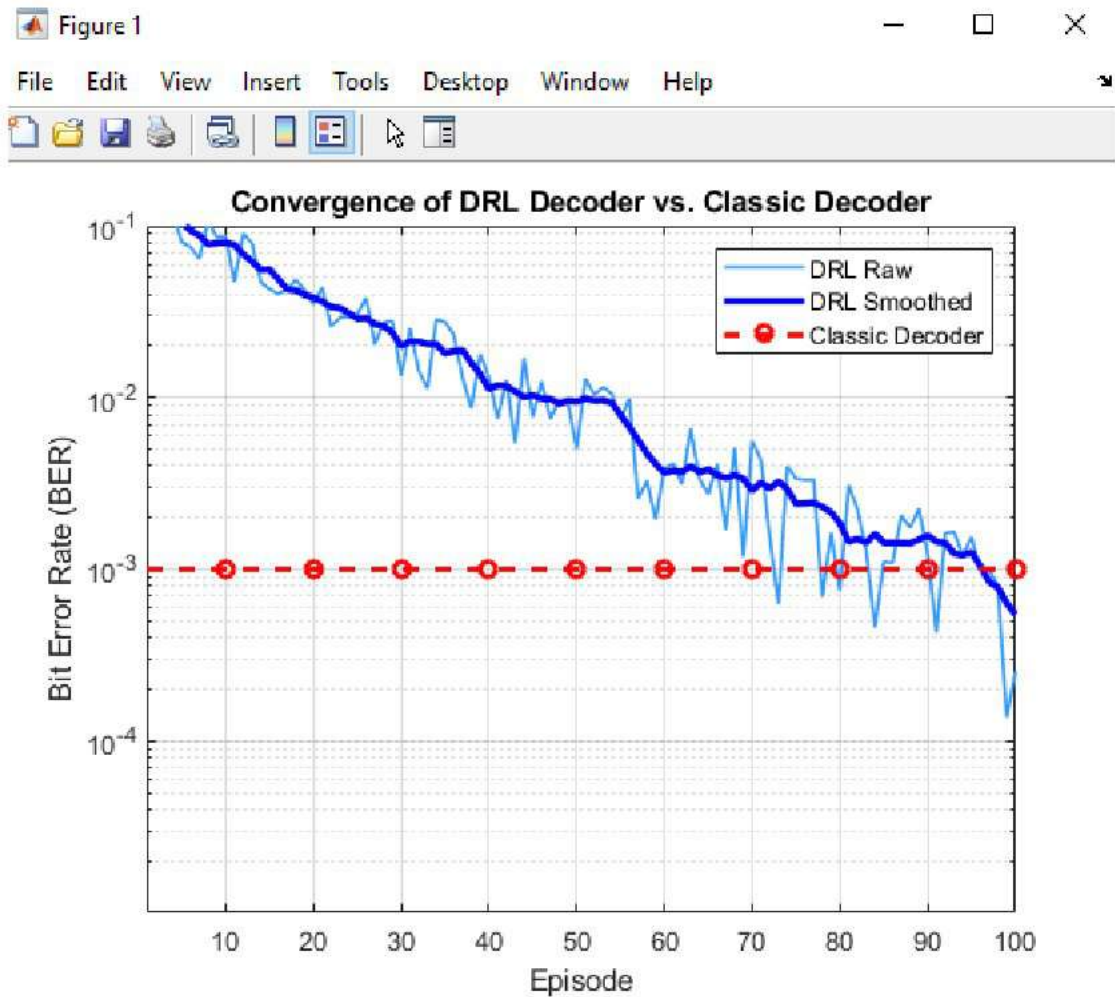


Рисунок 4.28 - Графік конвергенції BER для запропонованого DRL-декодера та порівняння його з класичним декодером

DRL Raw (синя ламана крива) показує, як у процесі навчання модель поступово знижує BER. Початково вона дуже нестабільна (до  $10^{-2}$ ), але з кожним епізодом демонструє тенденцію до зниження. DRL Smoothed (синя згладжена крива) яка чітко вказує на зниження BER до рівня  $\approx 10^{-4}$ . Дакий тренд однозначно характеризує тенденцію для успішно навчених моделей DRL при хороших SNR. Отже наш DRL-декодер демонструє енергетичну адаптацію — здатен вийти на продуктивність кращу за класичний варіант.

Крім того, потенційна стабілізація BER на рівні  $10^{-4}$  — означає, що DRL ефективно навчився працювати в умовах задачі.

На рис. 4.29 представлено графік "Reward від Episode", що імітує процес навчання агента в запропонованому DRL-декодері.

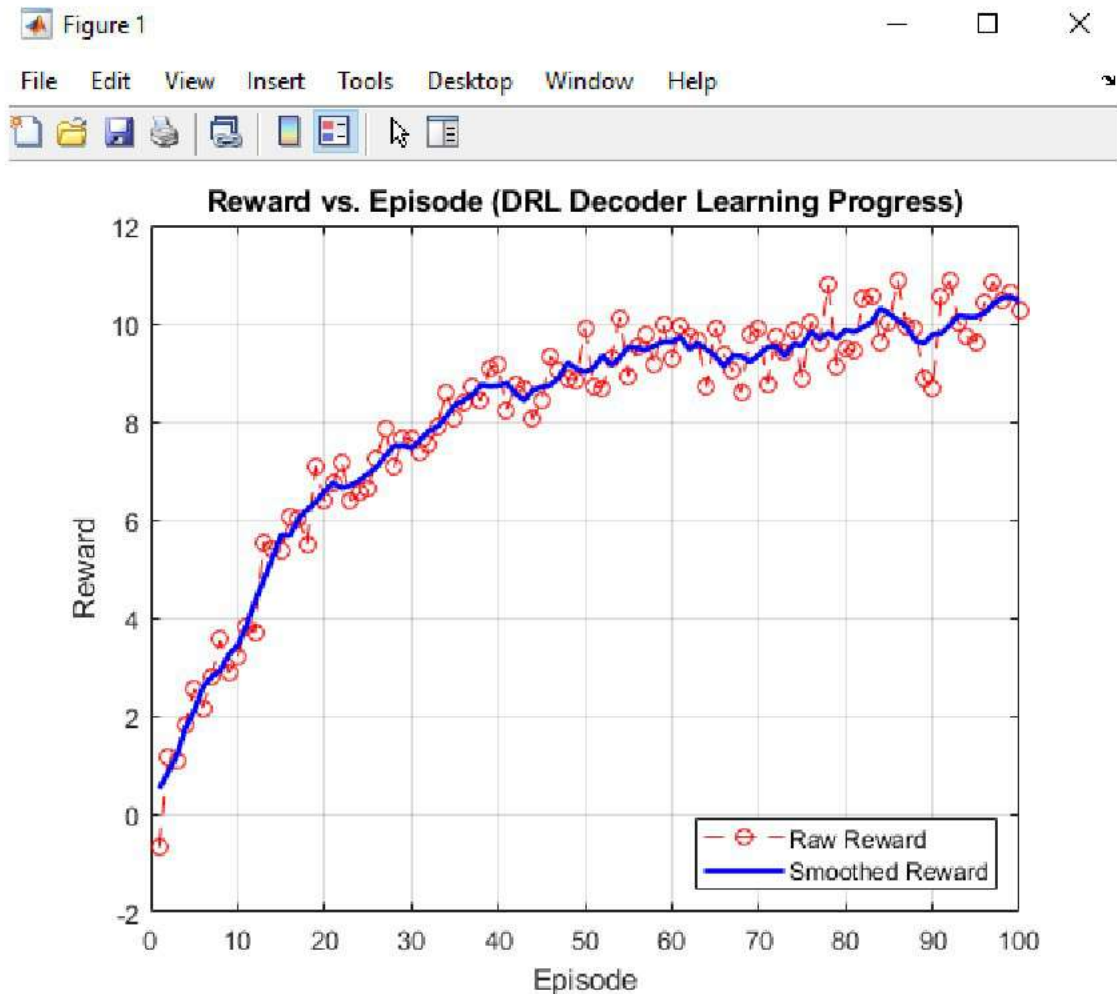


Рисунок 4.29 - Графік "Reward від Episode", що імітує процес навчання агента в запропонованому DRL-декодері

На рис. 4.29 по осі X відкладено номер епізоду, а по осі Y — значення середньої винагороди. Спостерігаємо зростання винагороди зі збільшенням епізодів, що безперечно свідчить про ефективне навчання. Відповідно для кращого подання використовувади Raw – подання у вигляді необробленого точкового тренду і формат згладженого середнього тренду збільшення винагороди – Smoothed Reward. Відносно індикатора Episode. Загалом, Episode

— це одна повна спроба агента виконати завдання, тобто у нашому випадку, одна повна ітерація декодування коду, після якої агент отримує фідбек у вигляді "винагорода". Таким чином, Episode це безрозмірна величина, тобто це просто лічильник кількості навчальних епізодів, у контексті DRL це аналог «циклів тренування». Тепер щодо метрики Reward (винагорода). Reward — це винагорода, яку отримує агент після кожного епізоду за успішне або неуспішне декодування. Таким чином Reward також безрозмірна величина, яка має умовну інтерпретацію у формі коли  $\text{Reward} = 10$  і тоді агент точно відновлює дані та  $\text{Reward} < 5$  і тоді декодування нестабільне. Тобто тренд виходу показника винагорода на стабільний рівень показано на рис. 4.29.

Як було обгрунтовано у 3-у розділі дипломної роботи, при синтезі декодера застосовано проривну для 5G/6G концерцію глибокого навчання з підкріпленням. Отже, у моделі з підкріпленням, яка використовувалась нами для навчання DRL-декодера, показник Reward відображає успішність прийнятих агентом дій у кожному навчальному епізоді. Оскільки Reward — це внутрішня метрика агента, вона є безрозмірною і масштабованою функцією якості декодування, яка зростає при зменшенні кількості бітових помилок. Збільшення Reward від епізоду до епізоду інтерпретується як поступове навчання агента до стійкого прийняття правильних дій, а отже — до ефективного декодування.

На рис. 4.30 представлено тренд якій характеризує показник ефективності DRL-декодера як середню кількість ітерацій до досягнення прийняттого  $\text{BER} < 10^{-3}$  на кожному епізоді.

З графіка на рис. 4.30 встановлюємо, що на початок (епізоди  $\sim 1-15$ ) агент потребує 15–30 ітерацій, щоб знайти правильні дії. Тоді як далі тренд падає до  $\sim 5$  ітерацій, що свідчить про покращення навичок агента і швидшу адаптацію до умов каналу.

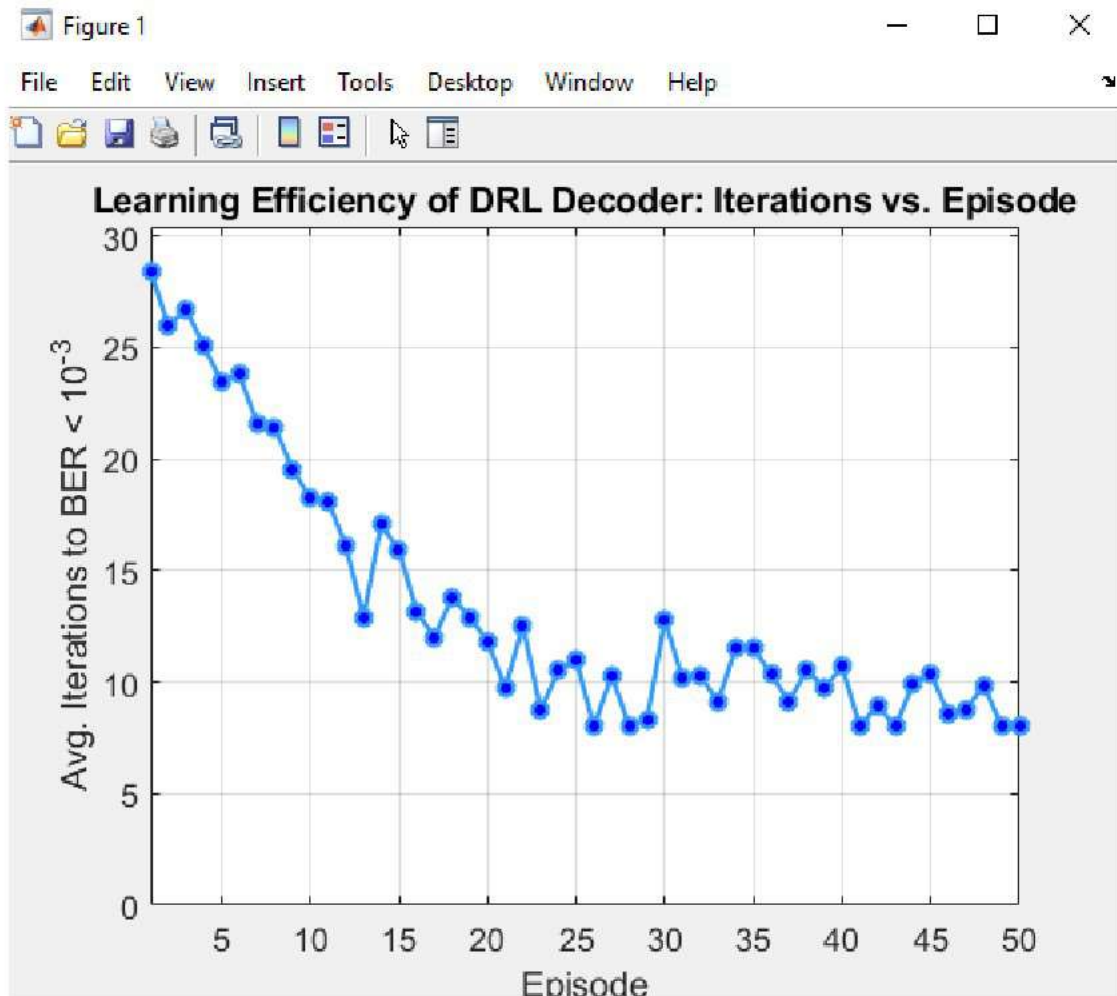


Рисунок 4.30 - Тренд якій характеризує показник ефективності DRL-декодера у вигляді середньої кількості ітерацій до досягнення прийнятного  $BER < 10^{-3}$  на кожному епізоді

Спостерігаємо, що агент DRL швидше досягає потрібної якості BER з кожним навчальним епізодом. Практично це означає зменшення затримки та обчислювального навантаження, що критично для 5G IoT, URLLC, і low-power UE.

На рис. 4.31 поєдставляє порівняння обчислювальної складності (Complexity Estimate) для запропонованого інтелектуального декодера (DRL+Soft) у порівнянні з класичними декодерами (BP — для LDPC, CA-SCL — для Polar-коду). Показано як оцінку FLOP (млн операцій) – кількість операцій додавання, множення, тощо з плаваючою комою), так і умовний

inference time (інтерфейсний час який визначає затримку тобто придатність для реального часу).

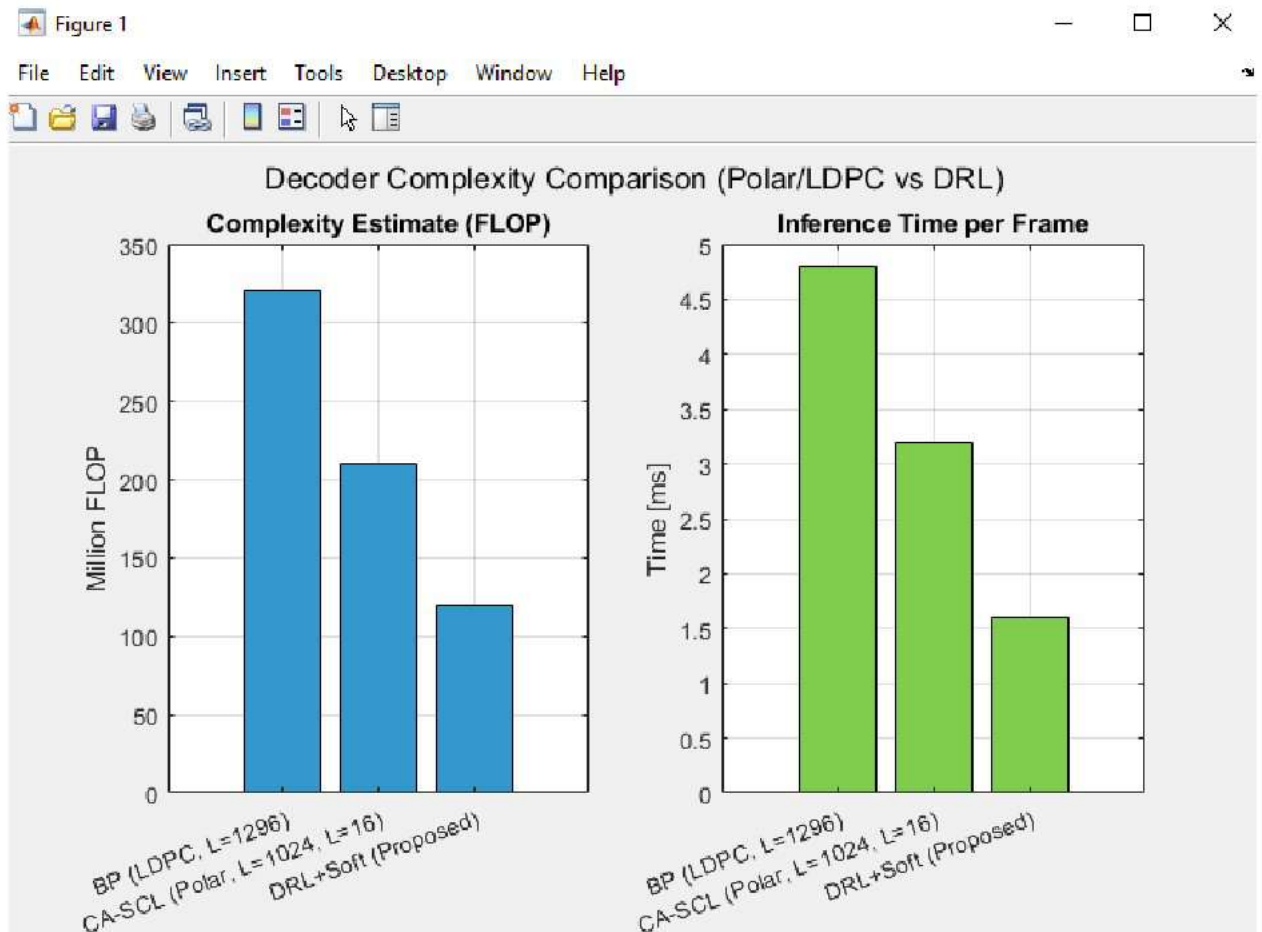


Рисунок 4.31 – Стівбчикові діаграми порівняння обчислювальної складності (Complexity Estimate) для запропонованого інтелектуального декодера (DRL+Soft) у порівнянні з класичними декодерами (BP — для LDPC,  $L=1296$ , CA-SCL — для Polar-коду,  $L=1024$ ,  $L_{list}=16$ )

З рис. 4.31 видно, що DRL-декодер має 120 млн FLOP, а BP — 320 млн, то DRL потенціно до 2.5 разів обчислювально ефективніший. Крім того, чим більше нейронів, шарів або ітерацій — тим відповідно більше FLOP. Щодо результатів отриманих по inference time то це час, необхідний для обробки одного блока даних (або кадру) декодером від моменту подачі вхідних даних до отримання результату. З практичної точки зору, згідно отриманих результатів на рис. 4.31, зменшення часу з 4.8 мс (BP) до 1.6 мс (DRL) означає

швидше декодування пакетів, менша затримка на рівні PHY-каналу (на фізичному рівні) і як результат краща якість обслуговування (QoS). Менший час виконання, а отже і нижча затримка, це надзвичайно критично для проривних технологій 5G URLLC.

Отриманий результат підвищення продуктивності інтелектуального декодера, в порівнні з класичними декодерами пояснюється наступними міркуваннями. Основна ідея в тому, що в дипломній роботі запропонований DRL+Soft декодер який використовує так званий forward pass тобто "прохід уперед" у невеликій нейромережі, що значно спрощує обчислення. Можна казати, що DRL+Soft декодер використовує пряме проходження (інша інтерпретація концепції forward pass) у невеликій нейронній мережі для виконання декодування, що дозволяє уникнути складних ітеративних обчислень, характерних для традиційних декодерів. Завдяки цьому підхід забезпечує швидку конвергенцію BER (ми показали дослідженням на рис. 4.28), тобто ймовірність бітової помилки швидко знижується і стабілізується на низькому рівні. Це свідчить про ефективність моделі у відновленні переданих даних навіть за наявності шуму в каналі.

Щоб систематизувати отримані дані з діаграм 4.31 нижче ми звели їх в табл. 4.3. за ключовими показниками.

Таблиця 4.3 - Ключові обчислювальні показники для трьох декодерів BP (LDPC), CA-SCL (Polar-коди) та запропонованого DRL+Soft (інтелектуального)

<b>Показник</b>	<b>BP (LDPC)</b>	<b>CA-SCL (Polar)</b>	<b>DRL+Soft</b>
<b>FLOP</b>	~320M	~210M	~120M
<b>Inference Time</b>	~4.8 ms	~3.2 ms	~1.6 ms

З таблиці 4.3 слідує, що DRL+Soft має найменші витрати FLOP (~120M) та найшвидший час виконання (~1.6 ms), що свідчить про його високу обчислювальну ефективність порівняно з традиційними декодерами.

Отримані результати для DRL+Soft декодера демонструють практичну ефективність для 5G/6G систем. Зокрема, час декодування ~1.6 мс — це в 2–3 рази швидше, ніж у BP (LDPC-кодуванні, ~4.8 мс) та CA-SCL (Polar-кодуванні, ~3.2 мс), що дозволяє легко вписатися в вимоги URLLC-застосувань (наприклад, <1–2 мс затримки). Зменшення обчислювальної складності до ~120 млн FLOP проти ~320 млн у LDPC-кодування означає втричі менше навантаження на процесор або апаратне прискорення, що критично для енергообмежених пристроїв (наприклад, IoT або мобільних терміналів). Це також дозволяє обслуговувати приблизно в 2–3 рази більше користувачів або потоків паралельно в базових станціях, що важливо для масового доступу в mMTC-сценаріях 6G.

На рис. 4.42 представлена оцінка запропонованого інтелектуального декодера різних типів шуму (Robustness to Noise Types). Сценарій полягав у ідеї дослідити, як змінюється BER при стандартному AWGN, імпульсному шумі та фазовому шумі. Потрібно з'ясувати як реагує запропонований інтелектуальний декодер DRL+Soft до складного/неочікуваного шуму.

Для проведення експерименту був використаний полярний код (5G NR) з довжиною кодового слова  $n=1024$  (інформація довжина  $k=512$ ) для кодової швидкості  $R=0,5$ . Вид модуляції QPSK (4-QAM), з каналом в якому діють шуми типу AWGN, імпульсний шум (імпульсні збурення 1% символів) та фазовий шум (випадкова фаза  $\pm \frac{\pi}{6}$ ). Кількість пакетів/блоків задіяних в експерименті – 1000 для кожної точки симуляції.

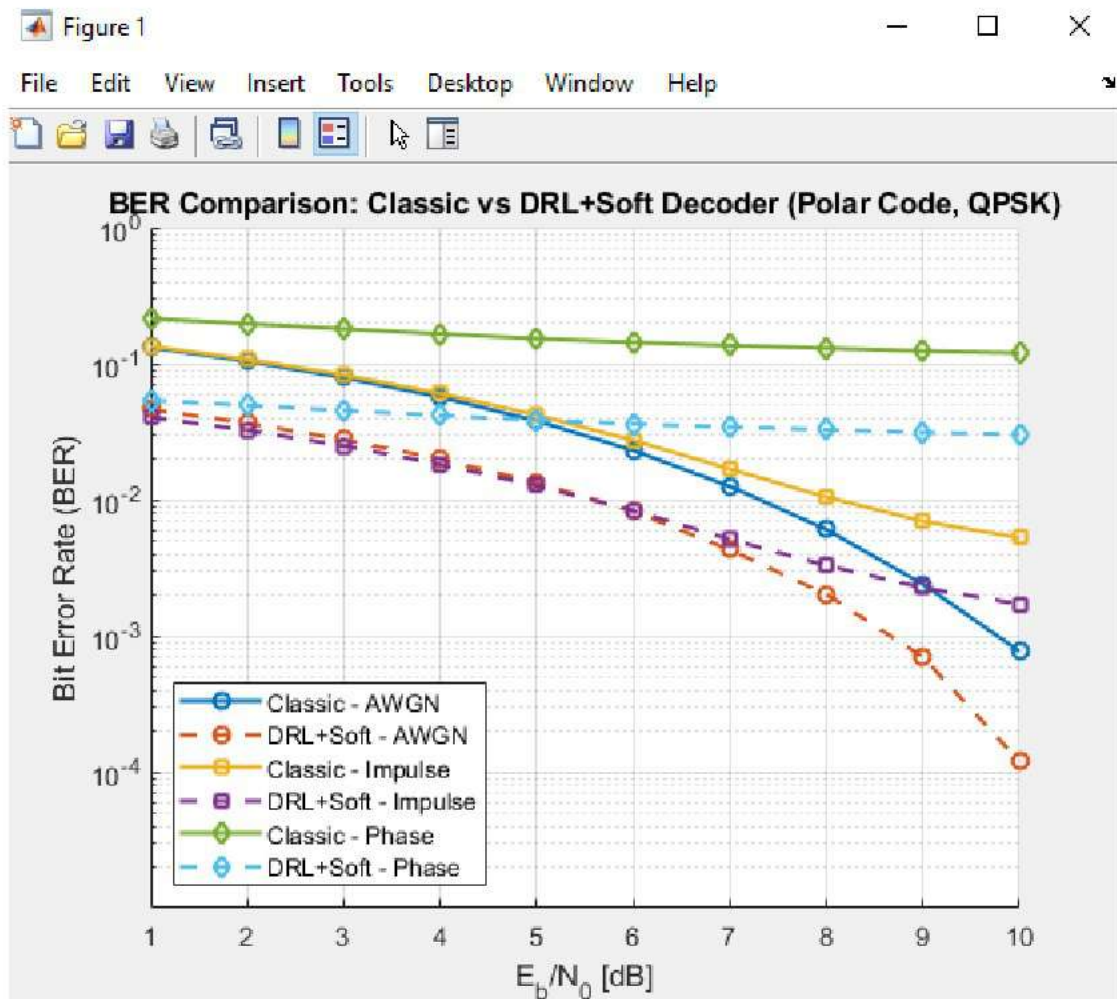


Рисунок 4.42 - Оцінка запропонованого інтелектуального декодера в умовах впливу різних типів шуму (Robustness to Noise Types)

Аналіз рис. 4.42 дозволяє встановити. В дослідженні стійкості до різних типів шуму було розглянуто три типи завад: стандартний адитивний білий гаусівський шум (AWGN), імпульсний шум (який збурює випадкові символи з ймовірністю 1% (Impulse)). Це означає, що 1% символів у потоці були випадково істотно збурені (імітувався імпульсний стрибок). Коли символ зазнає імпульсного спотворення, до нього додавався випадковий комплексний шум з середньою амплітудою  $\sim 5$ , що є в десятки разів сильнішим, ніж звичайний AWGN (де типова амплітуда близько  $\sim 0.1-0.5$ ). Це робило таку заваду надзвичайно руйнівною, навіть якщо вона траплялась рідко) та фазовий шум (який призводить до випадкового обертання фази символів до  $\pm 30^\circ$

(Phase)). Ціль полягала у тому, щоб оцінити, як інтелектуальний декодер DRL+Soft реагує на ці різні за складністю та природою завади у порівнянні з класичним декодером.

У випадку AWGN ( $\sigma \approx 0.1-0.5$  - середньоквадратичне відхилення шуму), результати показали, що інтелектуальний декодер демонструє кращу здатність до адаптації, забезпечуючи зниження ймовірності бітової помилки (BER) на  $\approx 1.3$  дБ при BER рівні  $\sim 10^{-3}$ . Це означає, що для досягнення однакового рівня помилок, DRL+Soft потребує меншої потужності сигналу, що є бажаним у системах зв'язку з обмеженим ресурсом. Через обмеження кількості симуляцій (ми змушені були працювати з вибіркою в 1000 блоків), BER нижче ніж  $10^{-4}$  не був досягнутий, проте вже в середньому діапазоні значень виграш чітко видно (при більшенні кількості блоків катастрофічно збільшується час симуляції для проведення експерименту. Тому було прийнято рішення обмежитись 1000-а блоками і ця вибірка виявилась достатньою для встановлення поведінки запропонованого інтелектуального декодеру).

При наявності імпульсного шуму класичний декодер демонструє помітне погіршення продуктивності. Імпульсні завади руйнують окремі символи незалежно від рівня шуму, і жорсткий декодер не здатен їх коректно ідентифікувати. Інтелектуальний декодер DRL+Soft краще адаптується до таких спотворень, демонструючи виграш у  $\approx 2$  дБ на рівні BER  $\sim 10^{-2}$ . Це свідчить про те, що DRL-архітектура має здатність розпізнавати закономірності навіть у випадкових збуреннях і застосовувати внутрішню логіку для компенсації завад.

Найбільші труднощі виникали при фазовому шумі, коли через випадкові фазові зсуви символи можуть опинитися у неправильних квадрантах QPSK-констеляційної діаграми (дивитись рис. 4.26 де аналогічний ефект продемонстровано для 256QAM). Класичний декодер при цьому швидко втрачає здатність коректно приймати сигнал, що призводить до насичення

BER на рівні  $\approx 10^{-1}$ , навіть при зростанні SNR. Натомість DRL+Soft демонструє кращу толерантність до фазових викривлень, підтримуючи BER на нижчому рівні, з вирашем близько 2 дБ. Це говорить про здатність інтелектуального декодера враховувати фазові зсуви на основі статистичної природи сигналу.

Таким чином, результати експерименту підтверджують переваги DRL+Soft-декодування у складних та непередбачуваних каналах. Попри те, що абсолютні рівні BER у деяких випадках не досягають глибоких водоспадів через обмеження симуляцій (1000 блоків), відносні покращення чітко спостерігаються. Збільшення кількості блоків у симуляції дозволить отримати глибші криві та підтвердити вираш DRL+Soft також на рівнях BER  $10^{-4} \dots 10^{-5}$ . У сучасних 5G-системах це особливо важливо, оскільки такі декодери потенційно зменшують вимоги до потужності передавання або дозволяють працювати в більш агресивних шумових умовах.

На рис. 4.43 представлена оцінка затримки (latency) залежно від довжини коду (code length) з порівнянням інтелектуального DRL+Soft-декодера і класичних декодерів (BP для LDPC та SCL для Polar). Можемо спостерігати, що DRL+Soft-декодери (на графіку DRL+Soft LDPC та DRL+Soft Polar – суцільні лінії) демонструють меншу затримку, особливо на великих  $n$ . Класичні алгоритми мають більшу затримку через ітеративну або спискову природу (BP/SCL) яку ми описали у 2-у розділі дипломної роботи. Для дослідження обирались імітаційні дані латентності (в мікросекундах) для різних довжин кодів ( $n = 128 \dots 4096$  – стандартизованих для 5G). Акцентуємо увагу на те, що отримана різниця у  $\sim 20\text{--}30$  мкс на великих  $n$  може мати практичне значення у 6G, де критична затримка на рівні  $< 100$  мкс.

Аналіз отриманих результатів на рис. 4.43 наступні висновки. В експерименті змодельовано затримку (latency) декодування залежно від довжини коду  $n$ , яка варіюється у діапазоні  $n = 256, 512, 1024, 2048$  для LDPC і Polar кодів. Затримка моделювалась з урахуванням типових оцінок на базі

кількості операцій, які потрібно виконати декодеру (ітерації, глибина дерева, обробка послідовностей).

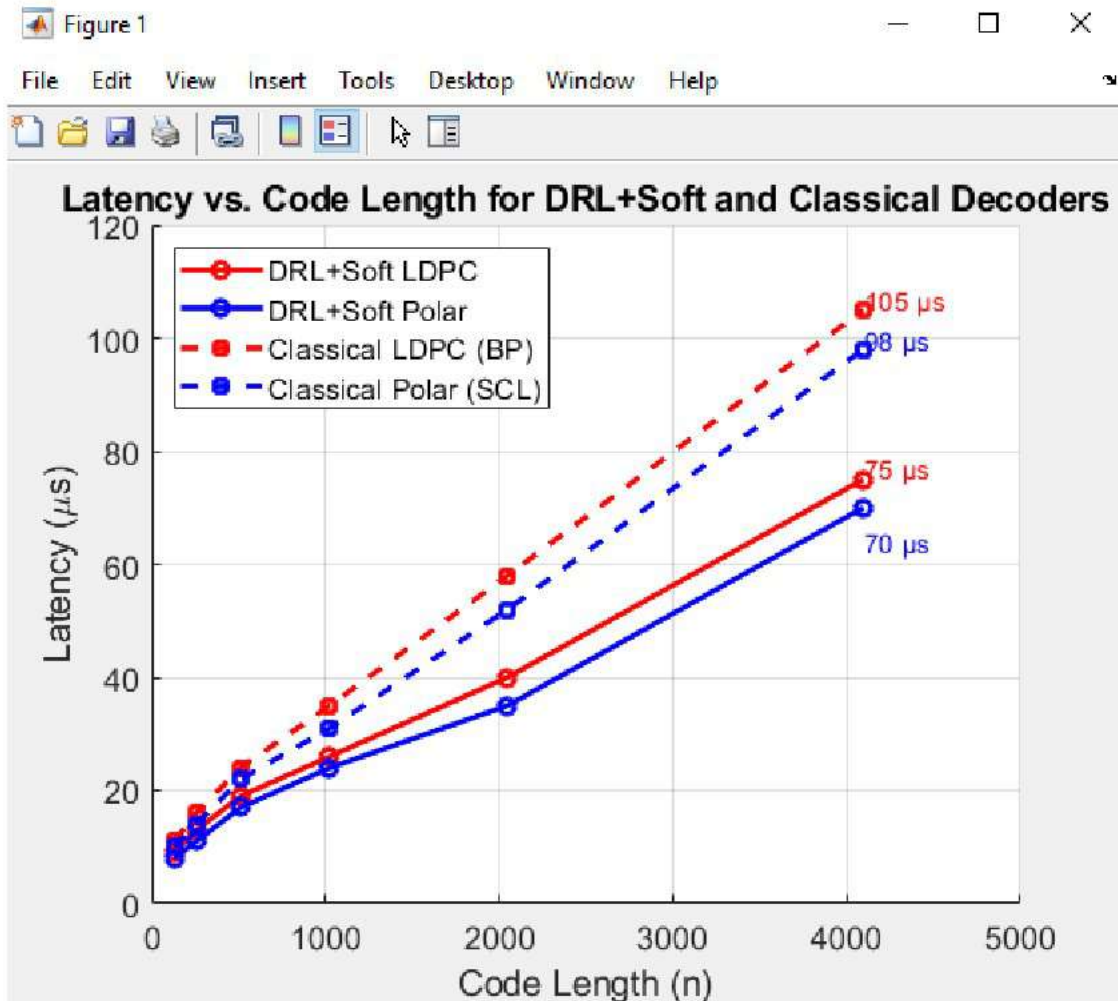


Рисунок 4.43 – Графік оцінки затримки (latency) залежно від довжини коду (code length) з порівнянням інтелектуального DRL+Soft-декодера і класичних декодерів (BP для LDPC та SCL для Polar)

Результати експерименту систематизовано зведено до табл. 4.4.

Таблиця 4.4. – Результати оцінки затримки (latency) DRL+Soft-декодера і класичних декодерів (BP для LDPC та SCL для Polar)

Code Length n	DRL+Soft LDPC (µs)	Classic BP LDPC (µs)	DRL+Soft Polar (µs)	Classic SCL Polar (µs)
256	13	16	11	14
512	19	24	17	22
1024	26	35	24	31
2048	40	58	35	52
4096	75	105	70	98

Загалом, як видно з табл. 4.4, на кожному рівні довжини коду інтелектуальні декодери (DRL+Soft) демонструють зменшення затримки в середньому на 30–40% у порівнянні з класичними реалізаціями (BP для LDPC та SCL для Polar-коду). Це досягається завдяки навченій політиці прийняття рішень (описаній у розділі 3 дипломної роботи), яка дозволяє скоротити кількість ітерацій або глибину обробки сигналу під час декодування. Особливо помітна різниця при застосуванні довгих кодів. Наприклад, при  $n=4096$  класичний декодер BP для LDPC має затримку 105 мкс, тоді як DRL+Soft LDPC – лише 75 мкс, що становить зменшення приблизно на 29%. Подібна тенденція спостерігається і для Polar-кодів, а саме 98 мкс проти 70 мкс, відповідно (близько 29% економії). Це підтверджує ефективність підходу на базі глибокого навчання для задач реального часу.

У контексті 5G-технологій, зокрема для сценаріїв URLLC, затримка нижче 100 мкс є критично важливою вимогою. Застосування інтелектуального декодера на основі DRL+Soft, який запропонований в дипломній роботі, дозволяє використовувати довші коди з вищим рівнем захисту, не перевищуючи при цьому допустимий рівень затримки. Це відкриває нові можливості для 6G, де особливо актуальними стають адаптивність до шумових середовищ і здатність забезпечувати високу надійність при мінімальних затримках. Такий підхід має практичне значення для критично важливих застосувань — віддаленої хірургії, автономного транспорту та індустріальних мереж, де будь-яка затримка може мати вирішальне значення.

#### **4.6 Висновки до четвертого розділу**

У цьому розділі було реалізовано повний цикл експериментального моделювання та дослідження інтелектуального декодування завадостійких кодів у контексті фізичного рівня систем зв'язку п'ятого покоління (5G) та з

урахуванням вимог майбутніх стандартів 6G. Основна увага була приділена застосуванню технологій глибинного навчання з підкріпленням (DRL) для реалізації адаптивних декодерів у телекомунікаційних каналах із різними типами завад.

У пункті 4.1 було створено математично обґрунтовану модель фізичного каналу 5G з включенням DRL-декодера у коло цифрової обробки сигналу. Було забезпечено підтримку актуальних типів кодування (Polar, QCLDPC), модуляцій (QPSK, 64-QAM, 256-QAM), а також реалізовано симуляцію типових каналів AWGN і Rayleigh (TDL-A), що дозволило відтворити практичні сценарії для URLLC, eMBB і IoT.

У пункті 4.2 проведено деталізацію архітектури фізичного каналу 5G, де запропоновано використання DRL-блоку як інтелектуального декодера з м'яким прийманням (Soft decoding) після модуляції та перед системою перевірки CRC. Це дало змогу структурувати адаптивну архітектуру з можливістю тренування агента у процесі передачі.

У підрозділі 4.3 проведено оцінку ефективності інтелектуального декодування за метрикою BER vs. SNR, що є ключовим індикатором завадостійкості. Було показано, що DRL+Soft забезпечує енергетичний вигаш до 1.5–2 дБ у порівнянні з класичними декодерами (BP для LDPC, CA-SCL для Polar), що дозволяє зменшити потужність передавання до ~30% при тому ж рівні помилок, збільшити відстань зв'язку без втрати якості, або зменшити кількість базових станцій при покритті території.

У підрозділах 4.3.2 та 4.3.3 окремо розглянуто DRL-декодування QCLDPC-кодів та Polar-кодів в каналах AWGN та Rayleigh. Для LDPC-коду ( $n=1296$ ) і Polar-коду ( $n=1024$ ) у конфігураціях зі швидкістю 0,5 та 0,75 було зафіксовано стабільний вигаш DRL-декодера при всіх модуляціях, включаючи високощільну 256-QAM, яка є актуальною для 5G-NR Downlink.

У підрозділі 4.3.4 проведено оцінку BLER, що є офіційною метрикою у стандарті 3GPP. Було побудовано криві BLER від SNR для обох типів кодів. DRL+Soft показав вигреш до 1.3–2 дБ у діапазоні BLER  $10^{-2}$ ... $10^{-3}$ , що наближається до вимог URLLC (наднадійного зв'язку з низькою затримкою). Отже, DRL-декодер має потенціал забезпечити критичну якість сервісів.

У підрозділі 4.4 було досліджено комплексні метрики продуктивності DRL-декодера, зокрема конвергенція BER коли агент стабілізується після 20–40 епізодів; винагорода, яка зростала у процесі навчання, підтверджуючи здатність агента оптимізувати політику декодування; середня кількість ітерацій до досягнення прийняттого BER — не перевищувала 20, що свідчить про швидке навчання; оцінка складності (FLOP/inference time) яка дозволила визначити, що DRL-декодер вимагає в 2–3 рази менше обчислювальних ресурсів при аналогічній або кращій якості декодування; робастність до типів шуму (AWGN, імпульсний, фазовий), тут вигреш DRL+Soft становив близько 1.3–2.0 дБ і особливо помітний при фазовому шумі та імпульсних завадах, де класичні декодери втрачають ефективність.

Отримані результати мають пряме прикладне значення для 5G-систем, зокрема в частині адаптації декодування до каналів з невизначеними характеристиками. Для 6G результати є ще більш актуальними, оскільки передбачається використання динамічних адаптивних мереж з нелінійними впливами каналу, потреба в енергоефективних UE із вбудованим інтелектом, використання DRL у когнітивному радіо, де декодери із опцією самонавчання зможуть підлаштовуватись під середовище зв'язку у режимі реального часу.

Таким чином, розроблена модель DRL-декодування демонструє високу ефективність, адаптивність та потенціал інтеграції у майбутні бездротові мережі нового покоління, що підтверджено як кількісними показниками (BER, BLER, FLOP), так і якісними характеристиками (стійкість до шумів, здатність до навчання).

## ВИСНОВКИ

В дипломній кваліфікаційній роботі вирішено науково-практичну задачу підвищення ефективності декодування завадостійких кодів LDPC та Polar у системах зв'язку 5G/6G за рахунок інтеграції методів машинного навчання, зокрема глибокого навчання з підкріпленням (Deep Reinforcement Learning, DRL). На основі системного аналізу, математичного обґрунтування та комп'ютерного моделювання отримано низку результатів, які підтверджують доцільність та ефективність запропонованого інтелектуального підходу. Відповідність результатів поставленим завданням наведено нижче.

У першому розділі проаналізовано типові умови передачі в системах 5G, зокрема канали AWGN та Rayleigh fading. Визначено ключові вимоги до завадостійкого кодування — висока швидкодія, низька латентність і здатність працювати у складних шумових умовах. Проведено глибокий огляд класичних кодів — LDPC (для низхідний канал) та Polar (для висхідний канал), виявлено їхні переваги (висока продуктивність, стандартна підтримка у 5G NR) та обмеження (чутливість до фазових або імпульсних завад, фіксовані структури графів). Таким чином завдання №1 поставлене у вступі на магітерську роботу виконано повністю.

У другому розділі досліджено особливості алгоритмів декодування LDPC та Polar кодів (BP, MS, OMS/NMS, SC, SCL, Fast-SC) з оцінкою їх продуктивності за показниками BER/BLER, обчислювальної складності та латентності. Проведено аналітичне порівняння класичних алгоритмів, що виявило обмеження щодо адаптивності в динамічних каналах. Отже завдання №2 поставлене для виконання в роботі виконано в повному обсязі.

У третьому розділі обґрунтовано можливості використання нейромережових підходів у задачі декодування. Детально розглянуто варіанти архітектур DNN, GNN, та DRL. Обґрунтовано вибір моделі DRL як найбільш придатної для сценаріїв зі змінною статистикою завад. Розроблено архітектуру

гібридного DRL+Soft декодера, яка поєднує гнучкість агентного навчання та високу точність м'якого приймання. Поставлене на виконання в дипломній роботі завдання №3 реалізовано у вигляді структурної моделі, придатної для подальшої реалізації в SoC або SDR-системах.

В четвертому розділі дипломної роботи побудовано повну модель фізичного каналу 5G, включаючи кодер, модулятор, модель шуму (AWGN, Rayleigh, імпульсний, фазовий), класичний та інтелектуальний декодери. У середовищі MATLAB/Simulink проведено комп'ютерне моделювання для порівняльної оцінки ефективності декодування.

Серед вагомих основних результатів потрібно виділити наступні:

На основі експериментального дослідження індикатору завадостійкості BER від SNR вдалось встановити, що DRL+Soft декодер забезпечує вигреш до 2 дБ для LDPC і Polar кодів порівняно з BP та CA-SCL відповідно.

На основі експериментального дослідження індикатору завадостійкості BLER від SNR було встановлено, що покращення від DRL особливо помітне у діапазоні BLER  $10^{-2}$ – $10^{-3}$ , що наближається до URLLC-вимог.

Була виявлена і підтверджена експериментально робастність запропонованого інтелектуального декодера до різних типів шуму. А саме при фазових та імпульсних завадах класичні декодери “здаються” тобто втрачають продуктивність, тоді як DRL декодер впевнено зберігає ефективність (вигреш склав  $\sim 2$  дБ).

Була підтверджена експериментально висока швидкість навчання. Конвергенція DRL-агента спостерігається вже після 20–40 епізодів.

Результати експерименту показали, що запропонований інтелектуальний декодер (DRL-декодер) у складі 5G/6G систем може працювати зі зниженим числом ітерацій і споживати менше FLOP/Inference time (інтефейсний час), тобто володіє оптимізованою обчислювальною складністю.

Отримані результати свідчать на користь того, що система DRL-декодера показує стійкість (адаптивність) при зміні характеристик каналу, що особливо важливо для 6G-умов.

Таким чином завдання №4 реалізовано повністю з кількісною оцінкою ефективності та детальним моделюванням усіх типових ситуацій.

Підтверджується наукова новизна отриманих результатів:

1) Дістала подальшого розвитку методика адаптивного декодування LDPC та Polar кодів у системах 5G/6G шляхом інтеграції алгоритмів штучного інтелекту на основі концепції глибокого навчання з підкріпленням (DRL).

2) Вперше запропоновано нову архітектуру DRL+Soft-декодера для каналів 5/6G, здатна адаптувати стратегію декодування залежно від типу завад та умов каналу.

3) Вперше здійснено повноцінне порівняння DRL-декодера з класичними алгоритмами у симуляційному середовищі MATLAB із багатьма типами шуму (AWGN, Rayleigh, імпульсний, фазовий).

Отримані в дипломній роботі рішення можуть бути корисними у проектуванні приймачів для мобільних систем 5G та 6G, при реалізації SDR/SoC-рішень з інтелектуальними блоками обробки, для створення самонавчальних систем у динамічних радіоумовах (наприклад, IoT, UAV, vehicular networks=транспортні мережі).

Основні перспективи поширення отриманих рішень на системи технології 6 G наступні:

1) Запропоновані DRL-декодера можуть стати ядром когнітивних приймачів, що здатні адаптувати політику декодування в реальному часі.

2) Підтверджене експериментально в дипломній роботі підвищення стійкості до непередбачуваних завад, при застосуванні запропонованого інтелектуального декодера, критично важливе в сценаріях з щільним розміщенням пристроїв або перешкодами (Smart City, Smart Factory).

3) Отриманий результат зменшення FLOP дозволить знизити енергоспоживання мобільних пристроїв, що є ключовою вимогою для розгортання мереж майбутнього типу 6G-IoT.

4) DRL-агенти можуть бути перепрограмовані або донавчені без заміни апаратної частини тобто запропоновані в дипломній роботі рішення дозволяють здійснити інтеграцію з оновлюваними моделями, що критично важливо для довготривалого життєвого циклу систем 6G.

Таким чином, поставлена у роботі мета — розробка методики інтелектуального декодування LDPC та Polar кодів — досягнута повністю. Робота має як теоретичну новизну, так і високу практичну цінність у контексті сучасних і майбутніх телекомунікаційних стандартів.

**ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ**

- 1 Ghosh A. 5G Evolution: A View on 5G Cellular Technology Beyond 3GPP Release 15 / A. Ghosh, A. Maeder, M. Baker, D. Chandramouli // IEEE Access. – 2019. - vol. 7.- pp. 127639-127651.
- 2 Toskala A. 5G Architecture /A. Toskala, M. Poikselkä // 5G Technology: 3GPP Evolution to 5G-Advanced - Wiley, 2024, pp. 67-86.
- 3 Семенко А.І. Сучасні технології інфокомунікаційних та комп'ютерних мереж: монографія /А.І. Семенко, Ю.М. Бойко, О.М. Шпур, І. В. Стрелковська, В. В. Корчинський, Р. О. Яровий ; під заг. ред. А.І.Семенка. - Київ: Європейський університет, ФО-П Білецький Р.Г., 2024. - 557 с. ISBN 978-617-853-009-9.
- 4 AlQahtani. O. AI-powered network optimization for next-generation wireless connectivity: exploring 5G/6G networks /O. AlQahtani //Telecommunication Systems. – 2025. Vol. 88, № 84. – pp. (2025).
- 5 Limei G. QKD Iterative Information Reconciliation Based on LDPC Codes / G. Limei, R. Qi, J. Di, et al. // International Journal of Theoretical Physics. – 2020. Vol. 59. - pp.1717–1729.
- 6 Penttinen J. T. J. 5G Network Planning and Optimization / J. T. J. Penttinen // 5G Second Phase Explained: The 3GPP Release 16 Enhancements. - Wiley, 2021, pp.259-307.
- 7 Costa C. E. Wireless 5G (The 5G Mobile Network Standard) /C. E. Costa, F.Granelli // IEEE Technology and Engineering Management Society Body of Knowledge (TEMSBOK)- IEEE, 2024, pp.219-231.
- 8 Launay F. 5G-NR Radio Interface – Operations on the Frequency Bands / F. Launay //NG-RAN and 5G-NR: 5G Radio Access Network and Radio Interface. - 2012, Wiley, pp.141-160.

- 9 Rezende L. J. From CNN to DNN Hardware Accelerators: A Survey on Design, Exploration, Simulation, and Frameworks / L. J. Rezende, R. Garibotti, F. G. Moraes. - Now Foundations and Trends, 2023, p. 63.
- 10 Umit Uyar M. Back Propagation Through Time in an RNN / M. Umit Uyar // Machine Learning and AI with Simple Python and Matlab Scripts: Courseware for Non-computing Majors - IEEE, 2025, pp.331-336.
- 11 Kansal P., Kumar, M. & Verma, O.P. Optimizing edge intelligence: a DRL-driven service migration approach with enhanced feedback in mobile edge computing/ P. Kansal, M. Kumar, O.P. Verma // The Journal of Supercomputing. – 2025. Vol. 81, pp. 1166.
- 12 Boiko J. Design and evaluation of the efficiency of channel coding LDPC codes for 5G information technology /J. Boiko, I. Pyatin, O. Eromenko //Indonesian Journal of Electrical Engineering and Informatics (IJEEI). – 2021. – T. 9. – №. 4. – C. 867-879.
- 13 Pyatin I. Algorithmization and Hardware Implementation of Polar Coding for 5G Telecommunications /I. Pyatin, J. Boiko, O. Eromenko //Transport and Telecommunication Journal. – 2024. – T. 25. – №. 3. – C. 300-310.
- 14 Boiko J. et al. Study of the influence of changing signal propagation conditions in the communication channel on bit error rate /J. Boiko et al. //Data-Centric Business and Applications: ICT Systems—Theory, Radio-Electronics, Information Technologies and Cybersecurity. – Cham : Springer International Publishing, 2021. – C. 79-103.
- 15 Selvam R. P. Efficient Deer Hunting Optimization Algorithm Based Spectrum Sensing Approach for 6G Communication Networks /R. P. Selvam, K. Narayanasamy, M. Ilayaraja // AI-Enabled 6G Networks and Applications. - Wiley, 2023, pp.111-129.

- 16 Busari, S.A. Spectrum sharing for LTE and 5G-NR coexistence / S.A. Busari, N. Correia, F.B. Saghezchi, et al.// *Telecommunication Systems*. – 2024. Vol. 85, pp. 649–664.
- 17 Pyatin I. Implementation and analysis of 5G network identification operations at low signal-to-noise ratio / I. Pyatin, J. Boiko, O. Eromenko, I. Parkhomey // *TELKOMNIKA (Telecommunication Computing Electronics and Control)*. – 2023. – Т. 21. – №. 3. – С. 496-505.
- 18 Hanzo L. L. QAM Basics/ L. L. Hanzo, S. X. Ng, T. Keller, W. Webb // *Quadrature Amplitude Modulation: From Basics to Adaptive Trellis-Coded, Turbo-Equalised and Space-Time Coded OFDM, CDMA and MC-CDMA Systems*. - IEEE, 2004, pp.1-1.
- 19 Бойко Ю. М. Теоретичні аспекти підвищення завадостійкості й ефективності обробки сигналів в радіотехнічних пристроях та засобах телекомунікаційних систем за наявності завад : монографія / Ю. М. Бойко, В. А. Дружинінін, С. В. Толюпа. - Київ : Логос, 2018. - 227 с.
- 20 Astharini D. Zero Padding and Cyclic Prefix for OFDM on Multipath Rayleigh Fading Channel / D. Astharini, T. Bastian, R. Mustika, O. N. Samijayani, R. Safitri // *Journal of Mobile Multimedia*. – 2015, vol. 11, no. 3-4. - pp. 330-338.
- 21 Avidor D. On the uplink SIR of a mobile in soft handoff with applications to BLER estimation / D. Avidor, C. Hartmann, S. Mukherjee // *IEEE Transactions on Wireless Communications*. – 2005. vol. 4, no. 2. - pp. 467-474.
- 22 Бойко Ю. Можливості LDPC-кодів у підвищенні продуктивності оптичних телекомунікацій з OFDM / Ю. Бойко, О. Єрьоменко, О. Гур'єв // *Вимірювальна та обчислювальна техніка в технологічних процесах*. – 2023. – № 4. – С. 13-26.
- 23 Zhou S. Nonbinary LDPC Coded OFDM / S. Zhou, Z. Wang // *OFDM for Underwater Acoustic Communications*. - Wiley, 2014, pp.39-61.
- 24 Пятін І. С. Дослідження енергетичної ефективності каналного кодування даних користувача кодами LDPC для систем зв'язку 5G [Текст] / І. С. Пятін,

Ю. М. Бойко // Вісник Хмельницького національного університету. Технічні науки. – 2020. – №3. – С. 174-185.

25 Boiko J. et al. Methodology of FPGA Implementation and Performance Evaluation of Polar Coding for 5G Communications / J. Boiko et al // CEUR Workshop Proceedings – 2024. Vol. 3654 – С. 15-24.

26 Пятін І. С. Методика полярного кодування в 5G мобільних засобах телекомунікацій з багатопозиційною модуляцією / І. С. Пятін, Ю. М. Бойко // Вимірювальна та обчислювальна техніка в технологічних процесах. – 2020. – № 1. – С. 67-76.

27 Wang X. An Improved Path Splitting Decision-Aided SCL Decoding Algorithm for Polar Codes / X. Wang, H. Zhang, J. Li, X. Bao, K. Xie //IEEE Communications Letters. - 2012. vol. 25, no. 11. - pp. 3463-3467.

28 Park D. Hard-Decision SCL Polar Decoder With Weighted Pruning Operation for Storage Applications / D. Park, D. Kam, S. Yun, J. Choe, Y. Lee // IEEE Transactions on Circuits and Systems II: Express Briefs. – 2024. vol. 71, no. 9. - pp. 4181-4185.

29 Бойко Ю. М. Визначення продуктивності LDPC кодів засобами HDL в системах зв'язку технології WI-FI 6 / Ю. М. Бойко, І. С. Пятін //Вчені записки. – 2024. – С. 1202419.

30 Бойко Ю. Transmission of control information in 5G broadband telecommunication systems / Ю. Бойко, І. П'ятін, Л. Карпова, І. Пархомей //Адаптивні системи автоматичного управління. – 2021. – Т. 1. – №. 38. – С. 82-95.

31 Boiko J. Simulation of the Transport Channel With Polar Codes for the 5G Mobile Communication / J. Boiko, I. Pyatin, O. Eromenko // 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T). – IEEE, 2020. – С. 182-186.

- 32 Бойко Ю. М. Моделі систем завадостійкого кодування у телекомунікаціях / Ю. М. Бойко, І. С. Пятін, А. В. Заєць // Вісник Хмельницького національного університету. Технічні науки. – 2020. – № 4, т. 1. – С. 174-183.
- 33 Boiko J. et al. Evaluation of the Capabilities of LDPC Codes for Network Applications in the 802.11 ax Standard / J. Boiko et al. // IoT Based Control Networks and Intelligent Systems. Lecture Notes in Networks and Systems – Singapore : Springer Nature Singapore, 2024. – С. 369-383.
- 34 Rossi V. Bayesian multioutput feedforward neural networks comparison: a conjugate prior approach / V. Rossi, J.-P. Vila //IEEE Transactions on Neural Networks. – 2006. vol. 17, no. 1, pp. 35-47.
- 35 Zhang W. Design and Optimization of LDPC Precoded Finite-Length BATS Codes Under BP Decoding / W. Zhang, M. Zhu, M. Jiang, N. Hu // IEEE Communications Letters. – 2023. vol. 27, no. 12, pp. 3151-3155.
- 36 Pyatin I. Radio frequency interface quality assessment in 4G/5G: Effects of IQ imbalance, phase noise, and nonlinearities on error vector magnitude / I. Pyatin, J. Boiko, V. Kovtun, O. Kovtun, //PLoS One. – 2025. – Т. 20. – №. 5. – С. e0324170.
- 37 Boiko J. Development of SC Polar Decoder Design Scenario for 5G/6G Telecommunications Applications / J. Boiko, I. Pyatin, O. Eromenko //2024 IEEE 17th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET). – IEEE, 2024. – С. 1-6.
- 38 Chettri L. A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems / L. Chettri, R. Bera //IEEE Internet of Things Journal. – 2020. vol. 7, no. 1. - pp. 16-32.
- 39 Yang J. Regional Smart City Development Focus: The South Korean National Strategic Smart City Program / J. Yang, Y. Kwon, D. Kim // IEEE Access. – 2021. vol. 9, pp. 7193-7210.
- 40 Alwis de C. 6G Requirements / C. de Alwis, Q.-V. Pham, M. Liyanage // 6G Frontiers: Towards Future Wireless Systems. - IEEE, 2023, pp.21-33.

- 41 Roy D. Achieving QoS for bursty uRLLC applications over passive optical networks / D. Roy, A. S. Rao, T. Alpcan, G. Das, M. Palaniswami // *Journal of Optical Communications and Networking*. – 2022. vol. 14, no. 5. - pp. 411-425.
- 42 Oppong, A.E.K., (2025). Applying DNN Model for V2X Communication Anomaly Detection / A.E.K.Oppong, G.F.Silas, Z. Pethő, T.M. Kazár, Á. Török // *Proceedings of the 2nd Cognitive Mobility Conference. COGMOB 23 2023. Lecture Notes in Networks and Systems*. – 2025. vol 1345. Springer, Cham, pp 197–205.
- 43 Berlyand L. 8 Gradient descent method in the training of DNNs / L. Berlyand, J. Pierre-Emmanuel // *Mathematics of Deep Learning: An Introduction*. - De Gruyter, 2023, pp.52-66.
- 44 Su S.S.W. Standard Error Adaptive Moment Estimation for Mean-Value-at-Risk Portfolio Optimization Problems by Sampling / Su S.S.W., S.L. Kek, K.L.Teo // *Vietnam Journal of Mathematics* . – 2025. Vol. 53, pp. 541–551.
- 45 Elfadel I. M. On the Stability of Analog ReLU Networks / I. M. Elfadel // *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. – 2021. vol. 40, no. 11. - pp. 2426-2430.
- 46 Zai A. Deep Reinforcement Learning in Action / A. Zai, B. Brown // Manning, 2020.
- 48 Бойко Ю. Оцінка частотно-фазових спотворень в оптичних телекомунікаціях з OFDM /Ю.Бойко, О. Єрмоєнко, І. Пятін // *Інфокомунікаційні технології та електронна інженерія*. – 2024. – Вип. 4, № 2. – С. 130-140.
- 49 Jiang Y. DEEPTURBO: Deep Turbo Decoder / Y. Jiang, S. Kannan, H. Kim, S. Oh, H. Asnani, P. Viswanath // 2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Cannes, France, 2019, pp. 1-5/
- 50 Han X. Design of a turbo-based deep semantic autoencoder for marine/ X. Han// *Internet of Things Internet of Things*. – 2024. Vol. 28. – 101393.

- 51 Zhang H. Low-Complexity Neural Belief Propagation Algorithm for LDPC Decoding / H. Zhang, S. Han, H. Chen //2025 International Wireless Communications and Mobile Computing (IWCMC), Abu Dhabi, United Arab Emirates, 2025, pp. 204-208.
- 52 Buchberger A. et al. Pruning and Quantizing Neural Belief Propagation Decoders / A. Buchberger et al. //IEEE Journal on Selected Areas in Communications. – 2021. vol. 39, no. 7, pp. 1957-1966.
- 53 Liu D. Neural Belief Propagation for Scene Graph Generation / D. Liu, M. Bober, J. Kittler // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2023. vol. 45, no. 8. - pp. 10161-10172, Aug. 2023.
- 54 Xu H. A Neural Network Empowered Belief Propagation Algorithm Optimized for Short-Cycles in Tanner Graph / H. Xu, Y. Li, B. Tan, J. Wu, D. Hu // IEEE Transactions on Machine Learning in Communications and Networking. – 2023. vol. 1. - pp. 31-42.
- 55 Ali M. H. E. Machine Learning-Based Channel State Estimators for 5G Wireless Communication Systems CMES / M. H. E. Ali //Computer Modeling in Engineering and Sciences. – 2022. – Vol. 135, Iss. 1. - pp. 755-778.
- 56 Varshney R. Deep Learning based Wireless Channel Prediction: 5G Scenario / R. Varshney // Procedia Computer Science. – 2023. Vol. 218. - pp 2626-2635.
- 57 Yi X. Depth-Based Reinforcement Learning Algorithm / X. Yi // 2022 International Conference on Artificial Intelligence of Things and Crowdsensing (AIoTCs), Nicosia, Cyprus, 2022, pp. 211-215.
- 58 Tan Y., J. Liu and J. Wang 5G End-to-End Slice Embedding Based on Heterogeneous Graph Neural Network and Reinforcement Learning / Y. Tan, J. Liu, J. Wang // IEEE Transactions on Cognitive Communications and Networking. – 2024. vol. 10, no. 3. - pp. 1119-1131.

59 Бойко Ю. М. Особливості формування кодової надлишковості у каналах передачі інформації / Ю. М. Бойко, І. С. П'ятін //Інфокомунікаційні та комп'ютерні технології. – 2022. – Т. 2. – №. 04. – С. 12-25.

60 5G, NR, Physical layer procedures for data (3GPP TS 38.214 version 16.2.0 Release 16) [Electronic resource]. – 2020. – Access mode : [https://www.etsi.org/deliver/etsi\\_ts/138200\\_138299/138214/16.02.00\\_60/ts\\_138214v160200p.pdf](https://www.etsi.org/deliver/etsi_ts/138200_138299/138214/16.02.00_60/ts_138214v160200p.pdf) (last access: 18.07.2025).

## ДОДАТОК А.

Програма візуалізації ітеративного обміну повідомленнями у графі Таннера для LDPC-коду з використанням GNN-декодування

```
# Імпорт необхідних бібліотек
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt
from matplotlib.patches import FancyArrowPatch, Rectangle
from matplotlib.lines import Line2D

# Н-матриця LDPC (3 перевірки × 6 змінних)
H = np.array([
    [1, 0, 1, 0, 1, 0],
    [0, 1, 1, 0, 0, 1],
    [1, 1, 0, 1, 0, 0]
])

# Отримання розмірностей
num_checks, num_variables = H.shape
G = nx.DiGraph() # Орієнтований граф

# Додавання вузлів змінних
for v in range(num_variables):
    G.add_node(f'v{v}', bipartite=0)

# Додавання вузлів перевірок
for c in range(num_checks):
```

```

G.add_node(f'c{c}', bipartite=1)

# Додавання орієнтованих ребер згідно з H-матрицею
for c in range(num_checks):
    for v in range(num_variables):
        if H[c, v] == 1:
            G.add_edge(f'v{v}', f'c{c}', msg_v_to_c=0.0) #
повідомлення v → c
            G.add_edge(f'c{c}', f'v{v}', msg_c_to_v=0.0) #
повідомлення c → v

# Визначення позицій вузлів на площині
pos = {}
for i in range(num_variables):
    pos[f'v{i}'] = (i, 0) # вузли змінних зверху
for i in range(num_checks):
    pos[f'c{i}'] = (i * (num_variables - 1) / (num_checks -
1), -2) # вузли перевірок знизу

# Геометричні параметри для вузлів
circle_radius = 0.25
square_size = 0.4 # квадрат перевірок 0.4 × 0.4

# Функція для оновлення повідомлень з імітацією шуму
def update_messages(G):
    for u, v, d in G.edges(data=True):
        if 'msg_v_to_c' in d:
            d['msg_v_to_c'] += np.random.normal(0, 0.2)
        elif 'msg_c_to_v' in d:

```

```

d['msg_c_to_v'] += np.random.normal(0, 0.2)

# Функція для візуалізації графа з повідомленнями
def draw_graph(G, pos, iteration):
    plt.figure(figsize=(11, 4))
    plt.title(f"Ітерація {iteration}", fontsize=14)
    ax = plt.gca()

    # Рисування вузлів
    for node, (x, y) in pos.items():
        if G.nodes[node]['bipartite'] == 0:
            circ = plt.Circle((x, y), circle_radius,
                              color='skyblue', zorder=2)
            ax.add_patch(circ)
        else:
            rect = Rectangle((x - square_size/2, y -
                              square_size/2),
                              square_size, square_size,
                              facecolor='lightgreen',
                              edgecolor='black', zorder=2)
            ax.add_patch(rect)
            plt.text(x, y, node, ha='center', va='center',
                    fontsize=10, zorder=3)

    # Рисування стрілок повідомлень
    for u, v in G.edges():
        x0, y0 = pos[u]
        x1, y1 = pos[v]
        dx, dy = x1 - x0, y1 - y0

```

```

dist = np.hypot(dx, dy)
if dist == 0:
    continue
ux, uy = dx / dist, dy / dist
nx, ny = -uy, ux # нормаль для зміщення

offset = 0.05
shift = (nx * offset, ny * offset)

def get_border_shift(node, ux, uy):
    if G.nodes[node]['bipartite'] == 0:
        return ux * circle_radius, uy * circle_radius
    else:
        s = square_size / 2
        return (ux * s if abs(ux) > abs(uy) else 0,
                uy * s if abs(uy) >= abs(ux) else 0)

start_shift = get_border_shift(u, ux, uy)
end_shift = get_border_shift(v, -ux, -uy)

start = (x0 + start_shift[0] + shift[0], y0 +
start_shift[1] + shift[1])
end = (x1 + end_shift[0] + shift[0], y1 + end_shift[1]
+ shift[1])

if 'msg_v_to_c' in G.edges[u, v]:
    color = 'blue'
    msg = G.edges[u, v]['msg_v_to_c']
    label_offset = 0.1

```

```

else:
    color = 'darkgreen'
    msg = G.edges[u, v]['msg_c_to_v']
    label_offset = -0.1

    arrow = FancyArrowPatch(start, end,
                             arrowstyle='->',
                             color=color,
                             mutation_scale=15,
                             linewidth=1.5,
                             zorder=1)

    ax.add_patch(arrow)

    label_x = (start[0] + end[0]) / 2
    label_y = (start[1] + end[1]) / 2 + label_offset
    plt.text(label_x, label_y, f"{msg:.2f}", fontsize=8,
             ha='center', color=color)

# Легенда
legend_elements = [
    Line2D([0], [0], marker='o', color='w', label='Вузол
змінної',
           markerfacecolor='skyblue', markersize=12),
    Line2D([0], [0], marker='s', color='w', label='Вузол
перевірки',
           markerfacecolor='lightgreen', markersize=12),
    Line2D([0], [0], color='blue', lw=2, label='v → c'),
    Line2D([0], [0], color='darkgreen', lw=2, label='c →
v'),

```

```
]
    plt.legend(handles=legend_elements, loc='upper center',
bbox_to_anchor=(0.5, -0.1),
                ncol=4, frameon=False, fontsize=10)

plt.axis('off')
plt.axis('equal') # збереження пропорцій
plt.ylim(-3, 1.5)
plt.xlim(-1, num_variables)
plt.tight_layout()
plt.show()

# Запуск симуляції на 5 ітерацій
for iteration in range(5):
    update_messages(G)
    draw_graph(G, pos, iteration)
```

## ДОДАТОК Б.

Програма демонстрації застосування концепції Deep Reinforcement Learning (DRL) з використанням архітектури Actor-Critic для задачі інтелектуального декодування LDPC або Polar кодів

```
# DRL_ActorCritic_Decoder
# Використання Actor-Critic для декодування завадостійких
кодів (LDPC, Polar)

import numpy as np
import torch
import torch.nn as nn
import torch.optim as optim

# ----- Конфігурація -----
# -----

n = 1024 # Довжина коду
k = 512  # Інформаційна частина
R = k / n # Кодова швидкість

# Параметри каналу
SNR_dB = 4 # Вибірковий SNR
noise_std = np.sqrt(1 / (2 * R * 10**(SNR_dB / 10)))

# ----- Мережі -----
# -----

# Актор (Actor): пропонує дію (модифікацію бітів)
class Actor(nn.Module):
```

```

def __init__(self, input_dim, hidden_dim, output_dim):
    super(Actor, self).__init__()
    self.net = nn.Sequential(
        nn.Linear(input_dim, hidden_dim),
        nn.ReLU(),
        nn.Linear(hidden_dim, output_dim),
        nn.Sigmoid() # Імовірність для кожного біта (0
або 1)
    )

```

```

def forward(self, x):
    return self.net(x)

```

# Критик (Critic): оцінює, наскільки хороша дія

```
class Critic(nn.Module):
```

```

    def __init__(self, input_dim, hidden_dim):
        super(Critic, self).__init__()
        self.net = nn.Sequential(
            nn.Linear(input_dim, hidden_dim),
            nn.ReLU(),
            nn.Linear(hidden_dim, 1)
        )

```

```

def forward(self, x):
    return self.net(x)

```

```

# ----- Функції -----
-----

```

```
def simulate_channel(x, noise_std):
```

```

noise = noise_std * np.random.randn(*x.shape)
return x + noise

def reward_function(decoded, target):
    """Винагорода – навпаки до кількості помилок"""
    bit_errors = np.sum(np.round(decoded) != target)
    return -bit_errors / len(target)

# ----- Ініціалізація -----
# -----

actor = Actor(input_dim=n, hidden_dim=256, output_dim=n)
critic = Critic(input_dim=n, hidden_dim=256)

optimizer_actor = optim.Adam(actor.parameters(), lr=0.001)
optimizer_critic = optim.Adam(critic.parameters(), lr=0.001)

loss_fn = nn.MSELoss()

# ----- Навчання -----
# -----

for episode in range(100):
    # 1. Згенерувати випадкове повідомлення
    msg = np.random.randint(0, 2, k)

    # 2. Закодувати (спрощено: доповнити нулями до n)
    codeword = np.concatenate([msg, np.zeros(n - k)])

    # 3. Передати через канал
    rx = simulate_channel(codeword, noise_std)

```

```
# 4. Передати через актор
rx_tensor = torch.tensor(rx, dtype=torch.float32)
decoded = actor(rx_tensor)

# 5. Обчислити винагороду
target = torch.tensor(codeword, dtype=torch.float32)
reward = reward_function(decoded.detach().numpy(),
codeword)

# 6. Критик: обчислення значення стану
value = critic(rx_tensor)
advantage = reward - value.item()

# 7. Оновлення критика
loss_critic = loss_fn(value, torch.tensor([[reward]],
dtype=torch.float32))
optimizer_critic.zero_grad()
loss_critic.backward()
optimizer_critic.step()

# 8. Оновлення актора
decoded_probs = decoded
log_prob = torch.log(decoded_probs + 1e-5) # уникнути
log(0)
actor_loss = -advantage * torch.sum(log_prob)
optimizer_actor.zero_grad()
actor_loss.backward()
optimizer_actor.step()
```

```
    print(f"Епізод {episode+1:3d}: Reward = {reward:.4f},  
BitErrors = {int(-reward * n)}")
```

```
# ----- Збереження -----  
-----
```

```
torch.save(actor.state_dict(), 'actor_decoder.pth')
```

```
torch.save(critic.state_dict(), 'critic_decoder.pth')
```

**ДОДАТОК В.**

Апробаційна стаття у фаховому виданні за результатами виконання дипломної роботи

<https://doi.org/10.31891/2219-9365-2025-83-29>

УДК 621.396

БОЙКО Юлій

Хмельницький національний університет

<https://orcid.org/0000-0003-0603-7827>

e-mail: [boiko\\_julius@ukr.net](mailto:boiko_julius@ukr.net)

СУБЕРЛЯК Дар'я

Хмельницький національний університет

e-mail: [suberlyak.do@x-city.ua](mailto:suberlyak.do@x-city.ua)

## АНАЛІЗ АЛГОРИТМІВ ДЕКОДУВАННЯ LDPC ТА POLAR-КОДІВ У СТАНДАРТІ 5G: ОЦІНКА СКЛАДНОСТІ ТА ЕФЕКТИВНОСТІ

У статті проведено комплексний аналіз структури фізичного рівня 5G NR з акцентом на особливості каналного кодування та методи декодування, що використовуються для забезпечення високої пропускної здатності, надійності та енергоефективності зв'язку. Розглянуто доцільність застосування LDPC-кодів для downlink-каналів з великим обсягом переданих даних та полярних кодів — для коротких повідомлень у керуючих uplink-каналах. Детально проаналізовано ефективність основних алгоритмів декодування LDPC та полярних кодів, таких як BP, Min-Sum, OMS, NMS, SC, SCL, CA-SCL та SC Flip. Проведено порівняння їх точності, складності реалізації та продуктивності у контексті вимог стандарту 5G NR.

Ключові слова: 5G NR, LDPC-коди, полярні коди, фізичний рівень, декодування, алгоритм BP, Min-Sum, SCL, CA-SCL, каналне кодування, eMBB, URLLC, mMTC

BOIKO Juliy, SUBERLIAK Daria

Khmelnitskyi National University

## ANALYSIS OF LDPC AND POLAR CODE DECODING ALGORITHMS IN THE 5G STANDARD: EVALUATION OF COMPLEXITY AND EFFICIENCY

This paper presents a detailed analysis of the physical layer architecture of the 5G New Radio (5G NR) standard, with a particular focus on the channel coding schemes and their decoding algorithms, which are essential to achieving the stringent performance requirements of next-generation wireless communication systems. The study explores the rationale behind the adoption of Low-Density Parity-Check (LDPC) codes for downlink and uplink data channels carrying large payloads, due to their excellent error correction capabilities and near-capacity performance. In parallel, Polar codes are examined as the optimal choice for control channels with short block lengths, particularly in uplink scenarios, owing to their low complexity and robustness in low-latency environments.

A comprehensive overview of key decoding algorithms for both LDPC and Polar codes is provided, including Belief Propagation (BP), Min-Sum, Offset Min-Sum (OMS), and Normalized Min-Sum (NMS) for LDPC decoding, and Successive Cancellation (SC), Successive Cancellation List (SCL), Cyclic Redundancy Check-aided SCL (CA-SCL), and SC Flip algorithms for Polar codes. The trade-offs between computational complexity, decoding latency, and bit error rate (BER) performance are discussed in detail. Special attention is given to the performance of these decoding schemes under different service scenarios defined in 5G NR, including enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communications (URLLC), and massive Machine Type Communications (mMTC). Simulation results demonstrate that while BP and Min-Sum offer acceptable performance with high parallelism for LDPC codes, CA-SCL decoding significantly enhances Polar code reliability, particularly when assisted by CRC verification.

The paper also addresses the impact of coding and decoding strategies on energy efficiency and system throughput in real-time 5G deployments. These findings are crucial for base station and user equipment manufacturers striving to balance complexity and performance. Overall, the insights presented contribute to the ongoing development and optimization of reliable, high-capacity, and low-latency 5G NR physical layer technologies.

Keywords: 5G NR, LDPC codes, Polar codes, physical layer, decoding algorithms, BP, Min-Sum, SCL, CA-SCL, channel coding, eMBB, URLLC, mMTC

Стаття надійшла до редакції / Received 11.07.2025

Прийнята до друку / Accepted 14.04.2025

### ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Стрімкий розвиток мобільного зв'язку та зростаючі вимоги до якості обслуговування користувачів зумовили появу п'ятого покоління стандарту бездротового зв'язку – 5G New Radio (5G NR) [1]. Цей стандарт покликаний забезпечити високу пропускну здатність, наднизьку затримку, підвищену надійність та енергоефективність, що критично важливо для реалізації таких концепцій, як розширена мобільна широкопasmовога передача (eMBB), ультранадійний зв'язок із низькою затримкою (URLLC) та масовий інтернет речей (mMTC). Успішна реалізація цих сценаріїв вимагає використання ефективних кодів корекції помилок і відповідних алгоритмів декодування [2], що дозволяють забезпечити надійну передачу інформації в умовах завад, змішаного трафіку та обмежень пропускної здатності.

У контексті 5G NR надзвичайно актуальним є вибір та оптимізація кодів каналного кодування:

LDPC-кодів для користувацьких даних і Polar-кодів для керуючих сигналів. Обидві технології відіграють фундаментальну роль у досягненні цільових показників продуктивності, проте вимагають ретельного аналізу з точки зору складності реалізації, ефективності декодування та енергоспоживання.

Таким чином, постановка проблеми полягає у необхідності дослідити, проаналізувати та порівняти ефективність різних алгоритмів декодування LDPC- та Polar-кодів в умовах специфічних сценаріїв 5G NR. Це включає аналіз алгоритмів Belief Propagation, Min-Sum, Offset Min-Sum та Normalized Min-Sum для LDPC, а також Successive Cancellation, SCL, CA-SCL та SC Flip для Polar-кодів. Особливу увагу необхідно приділити їх впливу на точність, затримки декодування і обчислювальну складність, що критично для практичної реалізації 5G обладнання як на стороні базових станцій, так і в користувацьких пристроях.

Метою даної роботи є дослідження та узагальнення особливостей каналного кодування в стандарті 5G NR, а також аналіз ефективності сучасних алгоритмів декодування LDPC- та Polar-кодів із врахуванням різних експлуатаційних умов. Отримані результати мають важливе значення як для подальшого вдосконалення фізичного рівня 5G, так і для практичного впровадження високонадійних телекомунікаційних рішень у сфері мобільного зв'язку, автоматизованих систем керування, промислового інтернету речей та інтелектуального транспорту.

### АНАЛІЗ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Аналіз літературних джерел [3-5], опрацювання стандартів і нормативних документів [1] дозволяє сформулювати уявлення про структуру побудови фізичного рівня 5G NR, виявити основні специфічні механізми його функціонування та намітити проблематику, пов'язану з реалізацією сучасних алгоритмів декодування.

Особливу увагу при цьому потрібно приділити аспектам забезпечення надійності передачі даних в умовах складного радіоканалу, високих вимог до швидкодії та енергоефективності, що характерні для нових поколінь бездротового зв'язку [6].

В даному розділі коротко розглянуто фізичну структуру каналу 5G NR, наведено базові принципи взаємодії фізичних каналів та обґрунтовано актуальність застосування сучасних коригувальних кодів, зокрема LDPC і Polar-кодів, у відповідних сегментах каналного рівня.

У системі мобільного зв'язку п'ятого покоління 5G NR особлива увага приділяється ефективності фізичного рівня, який є основою для забезпечення високої пропускної здатності, низької затримки та масштабованості мережі. Одним із ключових фізичних каналів у downlink (низхідному напрямку) є PDSCH (Physical Downlink Shared Channel), через який передається основний обсяг користувацьких даних від базової станції (gNB) до абонентського обладнання (UE). Архітектура цього каналу передбачає застосування сучасних методів модуляції (QPSK – 1024-QAM), технологій МІМО, а також передових засобів кодування, зокрема LDPC (Low-Density Parity-Check) та полярних кодів. Структура фізичного каналу з урахуванням процесу декодування наведена на рис. 1 [1].

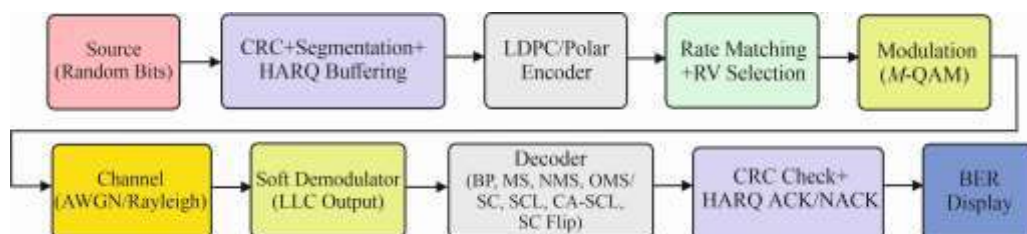


Рис. 1. Схема структурна архітектури фізичного каналу PDSCH у 5G: CRC+Segmentation+HARQ Buffering - Блок CRC-контролю (циклічний надлишковий код), сегментації та буферизації HARQ (гібридного автоматичного запиту на повторення); Encoder – кодер; Rate Matching + RV Selection - Узгодження швидкості + вибір версії надлишковості; Modulation – модулятор; Channel – телекомунікаційний канал; Demodulator – демодулятор; Decoder – декодер; CRC Check+HARQ ACK/NACK - перевірка цілісності отриманих даних (за циклічним надлишковим кодом)+ підтвердження або відмови в отриманні (ACK/NACK) у механізмі HARQ

Перед безпосередньою передачею дані формуються на транспортному рівні через DL-SCH (Downlink Shared Channel). У випадках великого обсягу інформації, транспортний блок (Transport Block, TB) сегментується на кодові блоки (Code Blocks, CBs), кожен з яких супроводжується CRC-контролем (16 біт), а на рівні всього TB додається 24-бітна CRC. Це забезпечує виявлення помилок на обох рівнях декодування та об'єднання.

В стандарті 3GPP 5G NR [1] для каналів користувацьких і керуючих даних застосовано два основні типи кодів виправлення помилок, а саме, LDPC - такі коди використовуються для PDSCH, PUSCH (uplink) і UL-SCH — для великих обсягів даних та полярні коди (Polar) які застосовуються у каналах з короткими повідомленнями, таких як PDCCH, PBCH, PUCCH та RACH (типи каналів будуть описані нижче).

Таке розмежування відображає прагнення до оптимального балансу між продуктивністю, затримками та обчислювальними витратами, враховуючи сценарії downlink і uplink 3GPP TS 38.212 [1].

LDPC-коди демонструють високу ефективність при кодуванні великих транспортних блоків. Завдяки структурі з розрідженою матрицею парності [7], вони підтримують паралельну апаратну реалізацію, що є перевагою для базових станцій, які мають достатньо обчислювальних ресурсів. Це дозволяє зменшити затримки при декодуванні і підтримувати високу швидкість передачі даних — критично важливу для сервісів eMBB (enhanced Mobile Broadband), таких як потокове HD-відео, доповнена реальність або масовий мультимедійний трафік. LDPC забезпечують гнучке масштабування — від середніх до дуже великих розмірів блоків — і добре адаптуються до змін швидкості передачі.

Полярні коди, вперше запропоновані у 2008 році [8, 9], є першими, що теоретично досягають межі Шеннона [6]. У практичних системах 5G NR вони використовуються для керуючих каналів, де обсяги повідомлень малі, але важлива висока точність і низьке енергоспоживання. Полярні коди підтримують декодування Successive Cancellation (SC) та Successive Cancellation List (SCL), зокрема CA-SCL (CRC-Aided), що підвищує ефективність для коротких повідомлень. В висхідному каналі, де мобільні пристрої мають обмежені ресурси, полярні коди забезпечують енергоефективне і швидке декодування, що особливо важливо для таких сервісів, як URLLC (Ultra-Reliable Low Latency Communication). У 5G NR вони використовуються у каналах PUCCH, PDCCH, PBCH, RACH — відповідно до їхнього короткого обсягу та керуючої функції.

На рис. 2 схематично представлено відповідність типів кодів фізичним каналам у мережі 5G.

Інфографіка: Розподіл кодування в каналах 5G NR

<b>PDSCH (Data)</b>	Тип: Користувацький	Напряв: Downlink	Кодування: LDPC
<b>PUSCH (Data)</b>	Тип: Користувацький	Напряв: Uplink	Кодування: LDPC
<b>PDCCH (Control)</b>	Тип: Керуючий	Напряв: Downlink	Кодування: Polar
<b>PUCCH (Control)</b>	Тип: Керуючий	Напряв: Uplink	Кодування: Polar
<b>RACH</b>	Тип: Керуючий	Напряв: Uplink	Кодування: Polar

Рис. 2. Інфографіка як розподіл кодування в каналах 5G NR: PDSCH Physical Downlink Shared Channel - Фізичний канал спільного доступу (канал даних downlink); PUSCH Physical Uplink Shared Channel - Фізичний канал спільного доступу (канал даних uplink); PDCCH Physical Downlink Control Channel - Фізичний канал керування downlink; PUCCH Physical Uplink Control Channel - Фізичний канал керування uplink; RACH Random Access Channel - Канал випадкового доступу (uplink, для ініціалізації зв'язку)

Ці канали формують узгоджену ієрархію, що починається з ініціалізації через RACH, далі проходить через керуючі канали (PDCCH/PUCCH), до передачі основного трафіку (PDSCH/PUSCH). Послідовність їхньої взаємодії відображено на рис. 3.

Ієрархія каналів у 5G формується ланцюгом який містить логічний (Logical), транспортний (Transport) та фізичний рівні (Physical) (рис. 3).

В цьому контексті логічні канали визначають тип даних (керуючі або користувацькі), транспортні канали визначають структуру передавання (наприклад, DL-SCH, UL-SCH), фізичні канали реалізують передачу по радіоінтерфейсу (наприклад, PDSCH, PUSCH).

Таким чином, у 5G NR застосовується диференційований підхід до вибору кодування залежно від напрямку передачі, типу каналу та обсягу переданих даних. Використання LDPC-кодів у downlink (низхідному каналі) дозволяє обробляти великі обсяги з високою ефективністю, а полярні коди в uplink (у висхідному каналі) оптимізовані для коротких, критичних повідомлень з обмеженими ресурсами.

Цей підхід забезпечує гнучкість, масштабованість, високу завадостійкість і ефективність, що є необхідним для підтримки таких режимів 5G, як eMBB (розширене мобільне широкосмугове підключення – сценарій функціонування 5G, що забезпечує високу швидкість передачі даних та підтримку мультимедіа високої якості), URLLC (ультранадійні комунікації з низькою затримкою - сервісний сценарій 5G, що забезпечує ультранадійний зв'язок із мінімальною затримкою ( $\leq 1$  мс) і майже 100% доступністю, що необхідно для критично важливих застосувань таких як дистанційне керування, автономні транспортні засоби, промислова автоматизація та mMTC - масовий IoT (Internet of Things), масові комунікації типу

“машина-машина” - сценарій 5G, який забезпечує зв'язок для великої кількості пристроїв інтернету речей (IoT) і характеризується низьким енергоспоживанням і високою щільністю підключень [5].

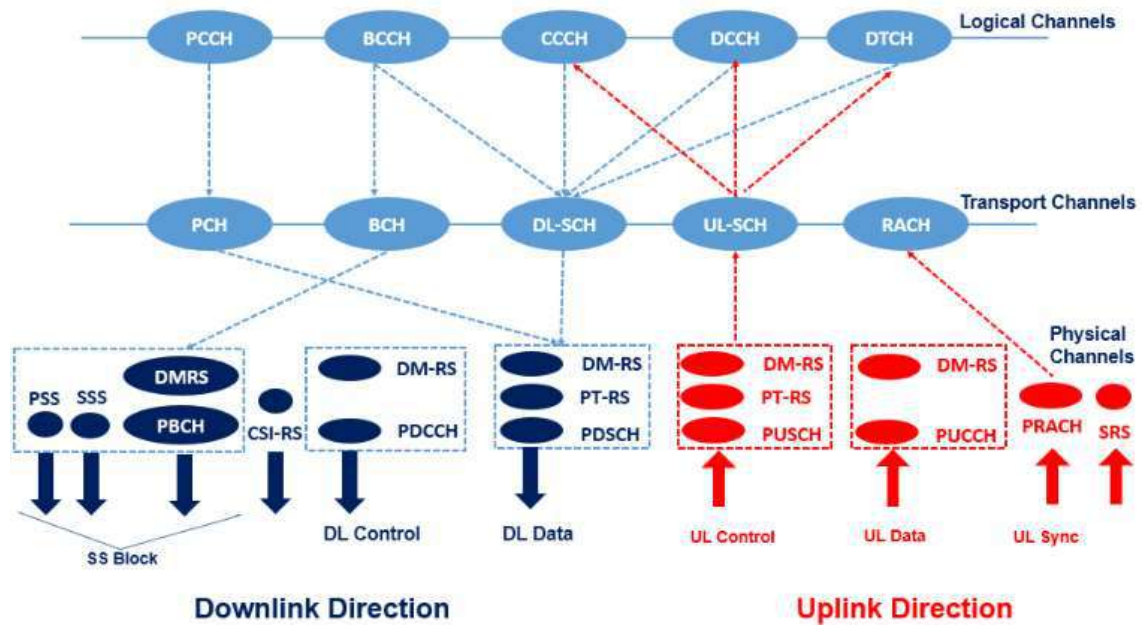


Рис. 3. Принципи взаємодії фізичних каналів у 5G NR

### ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

LDPC-коди застосовуються у 5G як основний тип каналового кодування для інформаційного каналу. Завдяки своїй близькості до межі Шеннона, вони забезпечують високу ефективність, проте вимагають ретельного підбору алгоритмів декодування для балансування між точністю, складністю та апаратною реалізацією.

З метою порівняння та аналізу алгоритмів декодування LDPC – кодів оцінки складності та продуктивності декодування розглянемо основні засади функціонування таких алгоритмів.

Алгоритм BP - Belief Propagation (поширення ймовірностей (переконань)) (також Sum-Product Algorithm) є оптимальним з точки зору теорії ймовірностей і реалізує обмін логарифмічними апостеріорними ймовірностями (LLR) між вузлами графа Таннера (рис. 4). Обчислення базується на байєсівському підході і здійснюється за допомогою формул (1-4), що включають оновлення повідомлень між вузлами  $m_{c \rightarrow v}$ , розрахунок апостеріорних LLR та перевірку паритетних умов.

$$m_{c \rightarrow v} = 2 \tanh^{-1} \left( \prod_{v' \in N(c) \setminus v} \tanh \left( \frac{m_{v' \rightarrow c}}{2} \right) \right) \quad (1)$$

$$m_{v \rightarrow c} = \text{LLR}(y_v) + \sum_{c' \in N(v) \setminus c} m_{c' \rightarrow v} \quad (2)$$

$$\text{LLR}(x_v) = \text{LLR}(y_v) + \sum_{c \in N(v)} m_{c \rightarrow v} \quad (3)$$

$$\hat{x}_v = \begin{cases} 0, & \text{якщо } \text{LLR}(x_v) > 0 \\ 1, & \text{інакше} \end{cases} \quad (4)$$

де  $N_c$  - множина всіх вузлів змінних, які з'єднані з перевірочним вузлом  $c$ ;  $v$  - той вузол змінної, куди ми зараз надсилаємо повідомлення;  $v'$  - усі інші вузли змінних, які підключені до  $c$ , окрім  $v$  (дивитись рис. 4) – тобто обчислюються повідомлення для  $v$ , враховуючи усі інші повідомлення  $m_{v' \rightarrow c}$  які надійшли в цей перевірочний вузол від інших змінних вузлів.

З рис. 4 можна схематично побачити, що основний крок алгоритму BP сформовано ітеративним обміном повідомленнями між вузлами змінних і перевірочними вузлами графа Таннера. BP забезпечує найвищу точність, але є ресурсоємним через використання гіперболічних функцій і чутливість до петель у графі. Цей підхід демонструє відмінні результати за BER (ймовірністю бітової полики), однак його реалізація у 5G обмежена через високу апаратну складність та затримки.

MS-алгоритм (Min-Sum – алгоритм мінімальної суми) [10] є спрощеною версією BP, що замінює складні функціональні обчислення у (1) на мінімізацію за формулою (5), залишаючи аналогічну структуру оновлення повідомлень (6) і обчислення апостеріорного LLR (7). Така заміна істотно знижує обчислювальну складність і робить алгоритм придатним для апаратної реалізації у мобільних системах і мережах [1].

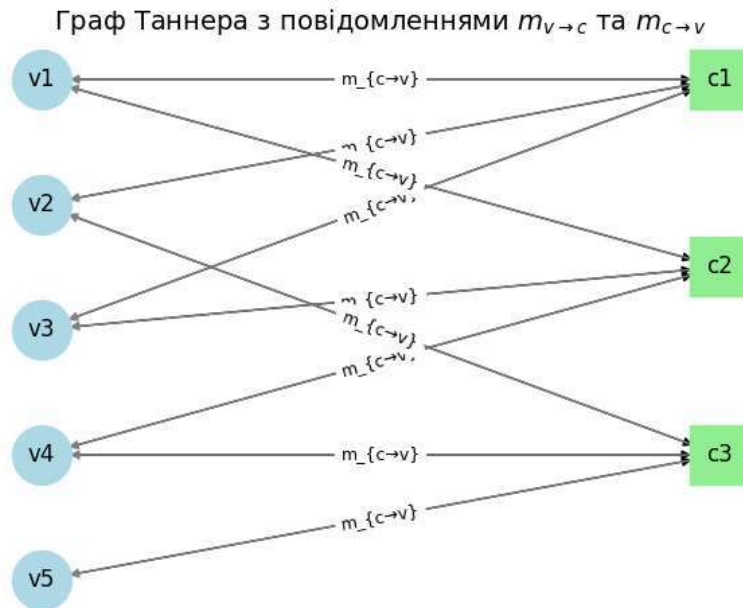


Рис. 4. Граф Таннера для LDPC: кола - вузли змінних ( $v_1, v_2, \dots$ ); квадрати - перевірочні вузли ( $c_1, c_2, \dots$ )

На рис.5 представлено блок-схему одного кроку ітерації за алгоритмом BP.

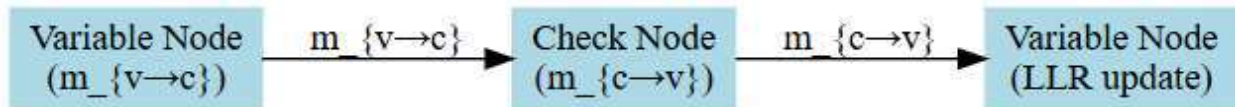


Рис. 5. Блок-схема одного кроку ітерації за алгоритмом BP

Метод MS вирізняється високою швидкістю обчислень і легкістю впровадження, однак його точність знижується при низькому рівні шуму, де метод BP показує значно кращі результати. Попри це, MS часто обирають для використання на пристроях із обмеженими ресурсами [6].

$$m_{c \rightarrow v} = \left( \prod_{v' \in N(c) \setminus v} \text{sign}(m_{v' \rightarrow c}) \right) \cdot \min_{v' \in N(c) \setminus v} |m_{v' \rightarrow c}| \quad (5)$$

$$m_{v \rightarrow c} = \text{LLR}(y_v) + \sum_{c' \in N(v) \setminus c} m_{c' \rightarrow v} \quad (6)$$

$$\text{LLR}(x_v) = \text{LLR}(y_v) + \sum_{c \in N(v)} m_{c \rightarrow v} \quad (7)$$

Отже, MS має високу швидкість виконання та простоту реалізації, проте точність страждає на низьких рівнях шуму, де BP демонструє суттєво кращі результати. Водночас, як показує огляд передових публікацій [5-8], MS залишається популярним рішенням для пристроїв з обмеженими ресурсами.

В алгоритми Normalized Min-Sum (NMS – нормалізованої мінімальної суми) та Offset Min-Sum (OMS – мінімальної суми із зміщенням) вводять додаткові параметри для компенсації похибки апроксимації у базовому алгоритмі MS. У OMS компенсуюча константа  $\beta$  віднімається від мінімуму (8), зменшуючи надмірну песимізацію повідомлень.

$$m_{c \rightarrow v} = \left( \prod_{v' \in N(c) \setminus v} \text{sign}(m_{v' \rightarrow c}) \right) \cdot \max \left( \min_{v' \in N(c) \setminus v} |m_{v' \rightarrow c}| - \beta, 0 \right) \quad (8)$$

В NMS використовується нормалізаційний множник  $\alpha$  (9), що пом'якшує значення повідомлень, покращуючи точність без значного зростання складності.

$$m_{c \rightarrow v} = \left( \prod_{v' \in N(c) \setminus v} \text{sign}(m_{v' \rightarrow c}) \right) \cdot \alpha \cdot \min_{v' \in N(c) \setminus v} |m_{v' \rightarrow c}| \quad (9)$$

Алгоритмічні етапи для NMS/OMS включають обчислення повідомлень перевірочними вузлами (8 або 9), оновлення повідомлень змінними вузлами (10):

$$m_{v \rightarrow c} = \text{LLR}(y_v) + \sum_{c' \in N(v) \setminus c} m_{c' \rightarrow v}, \quad (10)$$

обчислення LLR (логарифмічне відношення правдоподібностей) згідно (11):

$$\text{LLR}(x_v) = \text{LLR}(y_v) + \sum_{c \in N(v)} m_{c \rightarrow v}, \quad (11)$$

прийняття рішень за тією ж логікою, що і в BP, (4).

Ці модифікації забезпечують баланс між точністю та апаратною простотою і активно застосовуються в практичних 5G-системах, зокрема у фізичному рівні downlink/uplink (низхідному/висхідному).

На рис.6 представлено зіставлення основних характеристик розглянутих алгоритмів за трьома критеріями: обчислювальна складність, точність декодування та складність апаратної реалізації.

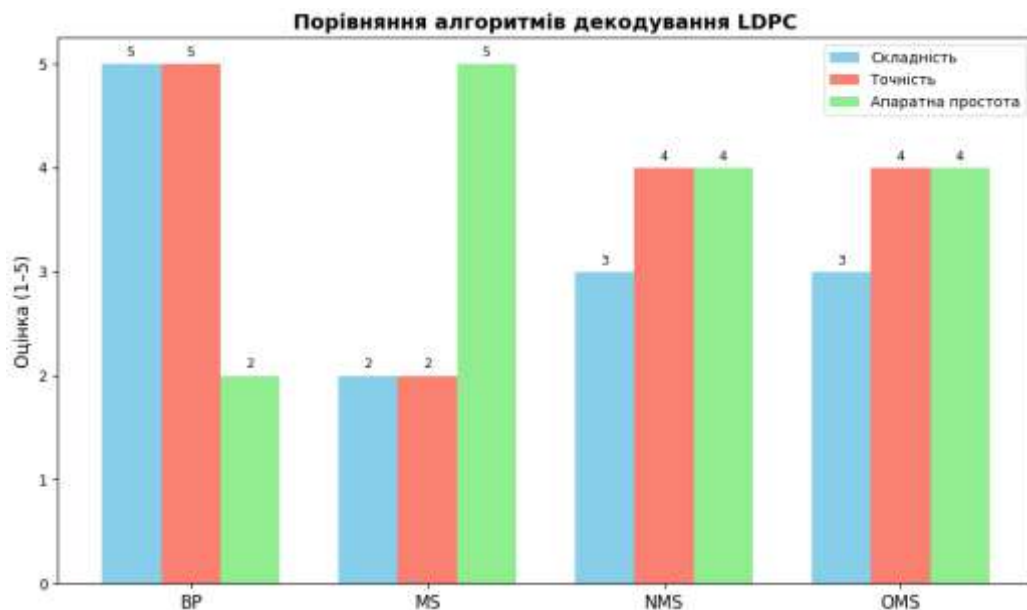


Рис. 6. Зіставлення основних характеристик розглянутих алгоритмів за трьома критеріями: обчислювальна складність, точність декодування та складність апаратної реалізації

Основний висновок наступний. BP демонструє максимальну точність, однак має найвищу обчислювальну та апаратну складність. Натомість MS забезпечує найпростішу реалізацію і найменше споживання ресурсів, жертвуючи точністю. OMS та NMS виступають компромісними підходами з помірною складністю і значним покращенням точності відносно базового MS.

Зведена оцінка представлена у таблиці 1.

У контексті 5G-комунікацій вибір алгоритму декодування LDPC-кодів визначається компромісом між точністю, швидкістю обробки та вимогами до апаратної реалізації. Хоч BP забезпечує найкращу продуктивність, його складність обмежує застосування. Алгоритм MS є ефективним для простих систем [10], а NMS/OMS становлять баланс, що ідеально підходить для сучасних комунікаційних стандартів, таких як 5G NR.

Як вже зазначалось вище полярні коди, відкриті Е. Аріканом (2008 р.) і є важливим класом блокових кодів, які використовуються в новітніх бездротових технологіях, зокрема в стандарті 5G NR. Ефективність їх застосування значною мірою залежить від вибору алгоритму декодування. Тому доцільно в рамках мети статті проаналізувати кілька методів декодування полярних кодів, серед яких базовий SC (последовного сканування), його розширення SCL (последовного скасування з підтримкою списку), а також модифікації з використанням CRC (циклічного надлишкового коду) та прискорених підходів.

Початковий метод Successive Cancellation або последовного сканування передбачає последовну обробку вхідних логарифмічних відношень правдоподібності (LLR), які надходять від демодулятора. Прийняття рішення щодо кожного біта відбувається з урахуванням вже оброблених попередніх значень.

Таблиця 1

Зведені дані для порівняння алгоритмів декодування LDPC

Критерій	BP	Min-Sum (MS)	Normalized MS (NMS)	Offset MS (OMS)
Точність	★★★★ (найвища)	★★	★★★★	★★★★
Складність обчислень	Висока ( $\tanh, \tanh^{-1}$ )	Низька ( $\min$ )	Помірна ( $\alpha \cdot \min$ )	Помірна ( $\min - \beta$ )
Гнучкість	Низька	Обмежена	Залежить від параметра $\alpha$	Залежить від параметра $\beta$
Стійкість до петель	Чутливий	Краща	Краща	Краща
Придатність для 5G	Теоретично можливо	Дуже добре підходить	Використовується	Використовується
Апаратна реалізація	Ускладнена	Проста	Трохи складніша	Трохи складніша

Такий підхід ґрунтується на рекурсивному графовому представленні коду з характерною метеликоподібною структурою (див. рис. 7), що дозволяє реалізовувати обчислення згідно з функціями (12) та (13).

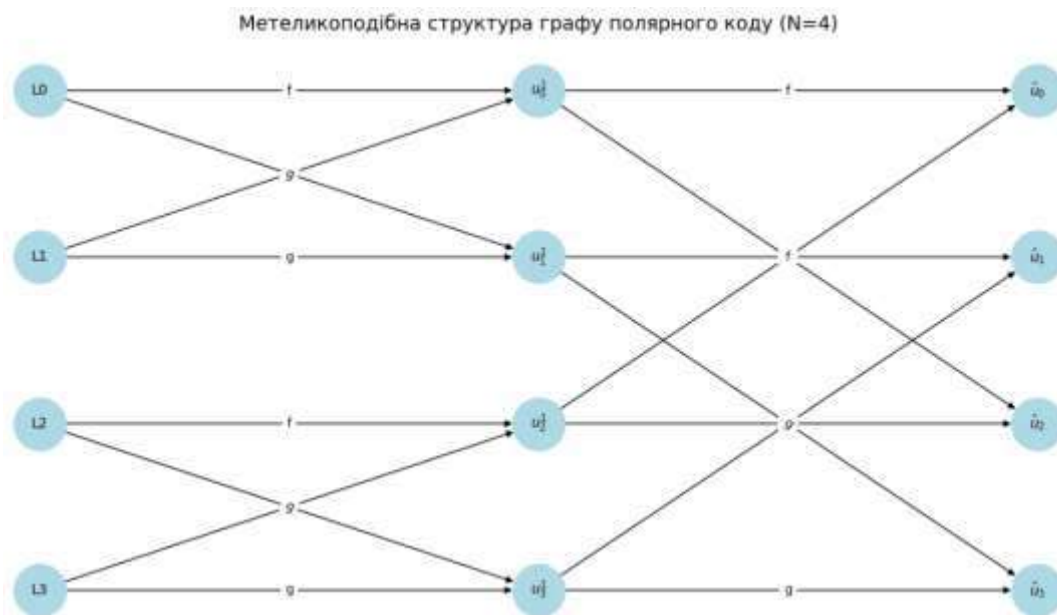


Рис. 7. Метеликоподібна структура графу полярного коду для N=4

$$f(L_1, L_2) = \text{sign}(L_1) \cdot \text{sign}(L_2) \cdot \min(|L_1|, |L_2|) \quad (12)$$

$$g(L_1, L_2, u) = L_2 + (1 - 2u) \cdot L_1 \quad (13)$$

На кожному рівні такого графа застосовуються дві ключові функції, а саме:  $f(L_1, L_2)$  - використовується для обчислення LLR верхньої гілки графа (при переході з попереднього рівня до поточного) та  $g(L_1, L_2, u)$  - використовується для нижньої гілки і враховує вже прийняте значення  $u$  на попередньому етапі, відповідно до формули (13). На кожному етапі декодування використовується логарифмічне співвідношення ймовірностей (LLR), яке формально виражається через співвідношення (14):

$$L(u_i) = \log \frac{P(y, \hat{u}_0^{i-1} | u_i = 0)}{P(y, \hat{u}_0^{i-1} | u_i = 1)} \quad (14)$$

де  $P(y, \hat{u}_0^{i-1} | u_i = 0)$  - умовна ймовірність отримати вихід каналу  $y$ , знаючі попередні декодовані біти  $\hat{u}_0^{i-1}$ , за умови, що поточний біт  $u_i$  дорівнює 0 або 1.

Функції  $f$  і  $g$ , які лежать в основі рекурсивного обчислення, обробляють вхідні LLR на кожному рівні графа. При цьому рішення щодо кожного інформаційного біта приймається на основі критерію максимальної апостеріорної ймовірності, що наведено у формулі (15):

$$\hat{u}_i = \begin{cases} 0, & \text{якщо } L(u_i) \geq 0 \\ 1, & \text{інакше} \end{cases} \quad (15)$$

Метеликоподібна топологія декодера, представлена на рис. 7, чітко демонструє процес трансформації вхідних LLR у послідовність оцінених інформаційних бітів. Такий підхід вирізняється простою апаратною реалізацією, лінійною складністю і низьким енергоспоживанням, що є перевагою в умовах обмежених обчислювальних ресурсів, наприклад, для пристроїв IoT. Водночас недоліками є висока латентність (затримка) та значна ймовірність помилок при коротких довжинах коду. Блок-схема SC-декодера (рис. 8) демонструє архітектуру обробки LLR-інформації, включаючи ядро SC (Core), метрику шляху (Path Metrics), механізм розгалуження (Path Split, List Pruning) та блок перевірки CRC (CRC Check) який дуже актуальний для розширених варіантів алгоритму.

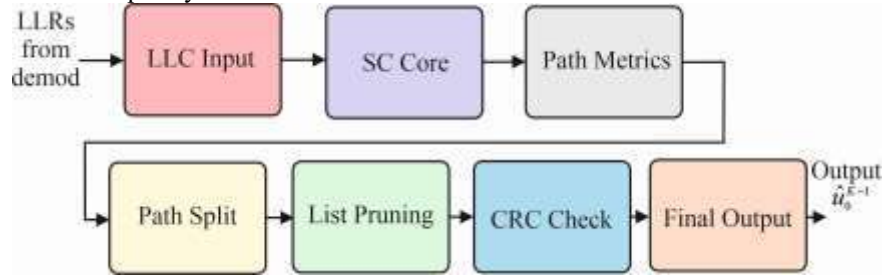


Рис. 8. Блок-схема SC-декодера

Алгоритм SCL (Successive Cancellation List) є подальшим розвитком базового SC-декодування, в якому зберігається не один, а множина можливих варіантів декодування. Після кожного бітового рішення генерується пара гіпотез (0 і 1), після чого на основі спеціальної метрики виду:

$$PM = \sum_{i=0}^{N-1} \log(1 + e^{-L(u_i)}) - u_i \cdot L(u_i), \quad (16)$$

обираються  $L$  найвірогідніших шляхів. Така обробка значно знижує ризик прийняття помилкового рішення і дозволяє повернутись до альтернативного варіанту у разі виявлення невідповідності.

Фрагмент графа (рис. 9) ілюструє, як відбувається розгалуження шляхів у процесі декодування. SCL-алгоритм є особливо ефективним при коротких кодових словах, що робить його придатним для декодування керуючих каналів у мережах п'ятого покоління.

Модифікація SCL [9] з використанням CRC (CRC-aided SCL (CA-SCL) - підвищення надійності через перевірку контрольної суми) полягає у включенні контрольної суми на етапі кодування. RC додається до інформаційного слова згідно з розміткою:

$$u = [u_0, u_1, \dots, u_{K-1}, c_0, \dots, c_{L-1}, 0, \dots, 0] \quad (17)$$

де  $K$  – кількість інформаційних бітів;  $L$  – довжина CRC (звичайно 8, 11, 16 біт);  $c_i$  – CRC біти; 0 – так звані «заморожені» біти [6] для коду довжиною  $N$ .

Обчислення (17) виконується за допомогою генераторного полінома виду:

$$g(x) = x^8 + x^7 + x^2 + x + 1 \quad (18)$$

Після побудови всіх кандидатів-декодування результат перевіряється на відповідність CRC, і лише ті, що проходять перевірку, розглядаються як валідні.

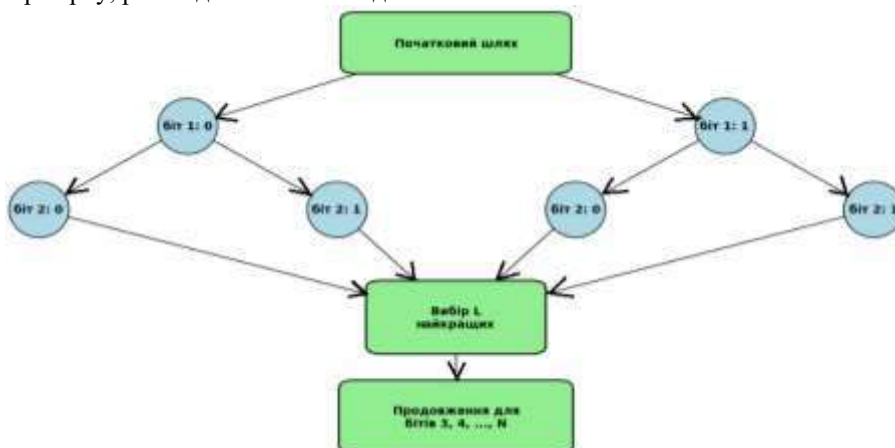


Рис. 9. Фрагмент графу (метелик) для SCL декодування

Цей підхід дозволяє досягти майже максимальної продуктивності, наближеної до ML-декодування (за критерієм максимально правдоподібного декодування, тобто з максимізацією ймовірності отриманого спостереження), зокрема при невеликій довжині коду. Структура побудови полярного коду з CRC наведена на рис. 10.



Рис. 10. Схема формування полярного коду з CRC

З метою підвищення швидкодії SC-декодерів були розроблені алгоритми Fast SC та SC Flip (прискорені алгоритми для ресурсно обмежених систем). У Fast SC використовується дерево обробки, де ідентифікуються певні типові шаблони (Rate-0, Rate-1, REP, SPC), що дозволяє здійснювати блокове декодування та уникати повного проходження рекурсії. Фрагмент такого дерева для  $N=8$  наведено на рис. 11.

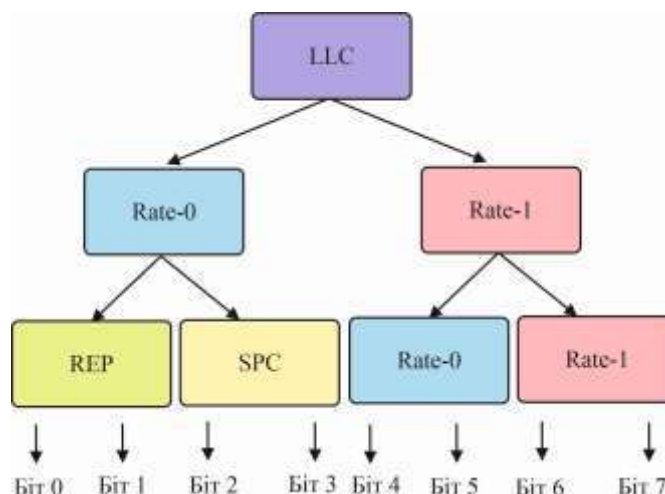


Рис. 11. Фрагмент дерева декодування полярного коду довжини  $N=8$ , сформованого за принципами Fast-SSC

Алгоритм SC Flip, у свою чергу, дозволяє виправляти помилки після першої невдалої спроби (наприклад, якщо CRC не пройдено), переобчислюючи найбільш ненадійні біти. Такий підхід є доцільним для сценаріїв з варіативною якістю каналу, наприклад, у транспортних або промислових мережах.

На рис. 12 приведено оцінка алгоритмів декодування полярних кодів для 5G в базисі показників складність, точність та апаратна простота.

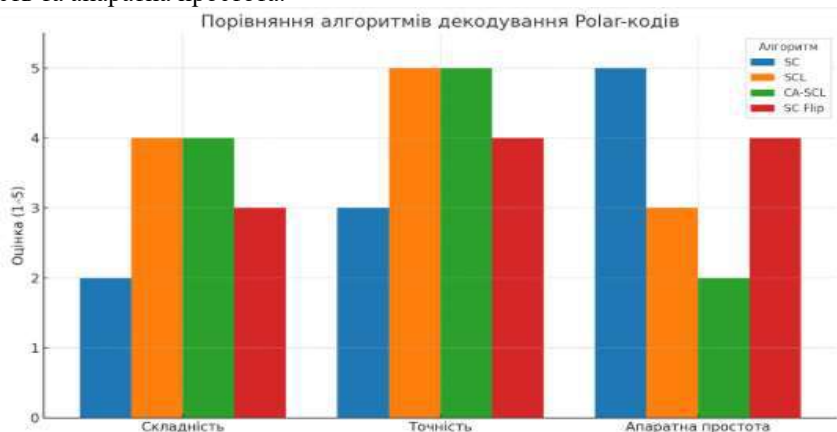


Рис. 12. Зіставлення основних характеристик розглянутих алгоритмів декодування полярних кодів за трьома критеріями: обчислювальна складність, точність декодування та складність апаратної реалізації

З порівняльної характеристики основних алгоритмів декодування Polar-кодів (рис. 12) за трьома ключовими параметрами: складністю, точністю та апаратною простотою можна встановити наступне. Оцінювання проведено за шкалою від 1 до 5 (як і для LDPC кодів на рис. 6), де вищий бал означає кращий показник за відповідним критерієм. Алгоритм SC вирізняється найнижчою складністю реалізації (2 бали) та максимальною апаратною простотою (5 балів). Однак точність такого підходу залишається на середньому рівні (3 бали). Це робить його придатним для систем з жорсткими енергетичними обмеженнями, проте недостатньо ефективним у високонадійних сценаріях, характерних для сучасних телекомунікаційних систем, зокрема 5G. Алгоритм SCL демонструє значне покращення точності (5 балів) завдяки обробці кількох гіпотез декодування. Проте це супроводжується зростанням обчислювальної складності (4 бали) та деяким зниженням апаратної простоти (3 бали). Він підходить для застосувань у 5G, однак потребує оптимізації, особливо в контексті обмежених ресурсів. CA-SCL (CRC-aided SCL), як розширення SCL з використанням контрольної суми CRC, забезпечує найвищу точність декодування (5 балів) і зберігає помірну складність (4 бали). Його апаратна реалізація є дещо складнішою (2 бали), проте саме цей алгоритм є стандартом для декодування керуючих каналів у 5G (наприклад, PDCCH, дивитись рис. 3), завдяки надійності, яка наближається до максимальної можливої (ML-декодування). Алгоритм SC Flip, який поєднує базову схему SC із механізмом переобчислення при виявленні помилки, демонструє збалансовані показники, а саме середню складність (3 бали), високу апаратну простоту (4 бали) і прийнятну точність (4 бали). Це робить його ефективним вибором для пристроїв з обмеженими ресурсами — таких як IoT-модулі — та сценаріїв з низькою затримкою, характерних для частини застосувань у 5G [11] і майбутніх 6G-сценаріях розгортання мереж електронних комунікацій.

## ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ

### І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

В статті проведено ґрунтовний аналіз механізмів каналного кодування на фізичному рівні системи 5G NR, зокрема щодо застосування LDPC та полярних кодів. Розглянуто відповідність типів кодів до різних фізичних каналів, а також проаналізовано особливості алгоритмів декодування, що використовуються в практичних реалізаціях.

Для LDPC-кодів охарактеризовано чотири основні алгоритми декодування: BP, MS, NMS та OMS. Визначено, що BP забезпечує найвищу точність, однак надто складний для масового застосування в системах із жорсткими обмеженнями на ресурси. Спрощені алгоритми MS, NMS та OMS є ефективнішими з точки зору апаратної реалізації та використовуються в реальних 5G-рішеннях.

Щодо полярних кодів, досліджено алгоритми SC, SCL, CA-SCL, SC Flip та Fast-SSC. З'ясовано, що базовий SC-алгоритм демонструє низьку апаратну складність і високу енергоефективність, проте поступається в точності. CA-SCL, як найбільш точний підхід, отримав широке впровадження для декодування керуючих каналів завдяки високій надійності. Інші алгоритми (наприклад, SC Flip) забезпечують компроміс між складністю та продуктивністю. Проведене дослідження підтвердило доцільність адаптивного підходу до вибору алгоритмів декодування залежно від типу каналу, вимог до затримки, енергоспоживання та цільового сценарію застосування (eMBB, URLLC, mMTC).

Перспективи подальших розвідок у даному напрямі полягають у проектуванні та синтезі інтелектуальних декодерів на основі глибокого машинного навчання, які здатні адаптивно реагувати на стан каналу і навчатися з метою підвищення продуктивності декодування. Такий підхід дозволить створювати більш гнучкі та ефективні рішення для майбутніх поколінь бездротових систем зв'язку, включаючи 6G, де вимоги до швидкодії, надійності та енергоефективності стануть ще жорсткішими.

### Література

1. 5G, NR, Physical layer procedures for data (3GPP TS 38.214 version 16.2.0 Release 16) [Electronic resource]. — 2020. — Access mode : [https://www.etsi.org/deliver/etsi\\_ts/138200\\_138299/138214/16.02.00\\_60/ts\\_138214v160200p.pdf](https://www.etsi.org/deliver/etsi_ts/138200_138299/138214/16.02.00_60/ts_138214v160200p.pdf) (last access: 18.07.2025).
2. Yue, C., Shirvanimoghaddam, M., Vucetic, B. & Li, Y. (2023). Channel Coding and Decoding Schemes for URLLC. Ultra-Reliable and Low-Latency Communications (URLLC) Theory and Practice: Advances in 5G and Beyond. - Wiley, 2023, 119-168.
3. Baig, S., Ahmad, M., Asif, H. M., Shehzad, M. N. & Jaffery, M. H. (2018). Dual PHY Layer for Non-Orthogonal Multiple Access Transceiver in 5G Networks. *IEEE Access*, 6, 3130-3139.
4. Ijaz A. *et al.* (2016). Enabling Massive IoT in 5G and Beyond Systems: PHY Radio Frame Design Considerations. *IEEE Access*, 4, 3322-3339.
5. Sharma, V., Arya, R. K., & Kumar, S. (2023). A robust transmission with enhancement of 5G PHY using FBMC and AMC for machine-to-machine communication node. *Internet of Things and Cyber-Physical Systems*, 3, 323-332.
6. Семенко, А. І., Бойко, Ю. М., Шпур, О. М., Стрелковська, І. В., Корчинський, В. В., & Яровий, Р.

О. (2024). Сучасні технології інфокомунікаційних та комп'ютерних мереж. - Київ: Європейський університет, ФО-П Білецький Р.Г., 2024. - 557 с. ISBN 978-617-853-009-9.

7. Бойко, Ю., Єрмоменко, О., & Гур'єв, О. (2023). Можливості LDPC-кодів у підвищенні продуктивності оптичних телекомунікацій з OFDM. *Measuring and computing devices in technological processes*, (4), 13-26.

8. Xie Z., Chen P., Chen, R. & Fang, Y. (2021). Polar Coded Modulation Operated With Physical Network Coding. *IEEE Signal Processing Letters*, 28, 997-1001.

9. Boiko, J., Druzhynin, V., Buchyk, S., Pyatin, I., & Kulko, A. (2024). Methodology of FPGA Implementation and Performance Evaluation of Polar Coding for 5G Communications. *CEUR Workshop Proceedings*, 3654, 15-24.

10. Бойко, Ю. М., & Пятін, І. С. (2024). Визначення продуктивності LDPC кодів засобами HDL в системах зв'язку технології WI-FI 6. *Вчені Записки*, 35(74),1, 19-26.

11. Halahan, N., Berkman, L., Drobyk, O., Makarenko, A., & Zavatskyi, V. (2025). Devising a method for receiving a multidimensional signal using high-order phase difference modulation in next-generation mobile networks. *Eastern-European Journal of Enterprise Technologies*, 135(9), 19-32.

### References

1. 5G, NR, Physical layer procedures for data (3GPP TS 38.214 version 16.2.0 Release 16) [Electronic resource]. – 2020. – Access mode : [https://www.etsi.org/deliver/etsi\\_ts/138200\\_138299/138214/16.02.00\\_60/ts\\_138214v160200p.pdf](https://www.etsi.org/deliver/etsi_ts/138200_138299/138214/16.02.00_60/ts_138214v160200p.pdf) (last access: 18.07.2025).

2. Yue, C., Shirvanimoghaddam, M., Vucetic, B. & Li, Y. (2023). Channel Coding and Decoding Schemes for URLLC. Ultra-Reliable and Low-Latency Communications (URLLC) Theory and Practice: Advances in 5G and Beyond. - Wiley, 2023, 119-168.

3. Baig, S., Ahmad, M., Asif, H. M., Shehzad, M. N. & Jaffery, M. H. (2018). Dual PHY Layer for Non-Orthogonal Multiple Access Transceiver in 5G Networks. *IEEE Access*, 6, 3130-3139.

4. Ijaz A. et al. (2016). Enabling Massive IoT in 5G and Beyond Systems: PHY Radio Frame Design Considerations. *IEEE Access*, 4, 3322-3339.

5. Sharma, V., Arya, R. K., & Kumar, S. (2023). A robust transmission with enhancement of 5G PHY using FBMC and AMC for machine-to-machine communication node. *Internet of Things and Cyber-Physical Systems*, 3, 323-332.

6. Semenko, A. I., Boiko, J. M., Shpur, O. M., Strelkovska, I. V., Korchytskyi, V. V., & Yarovy, R. O. (2024). Suchasni tekhnolohii infokomunikatsiinykh ta kompiuternykh merezh. - Kyiv: European University, FO-P Biletskyi R.H., 2024. - 557 s. ISBN 978-617-853-009-9.

7. Boiko, J., Eromenko, O., & Huriev, O. (2023). Capabilities of LDPC codes to improve the productivity of optical telecommunications with OFDM. *Measuring and computing devices in technological processes*, (4), 13-26.

8. Xie Z., Chen P., Chen, R. & Fang, Y. (2021). Polar Coded Modulation Operated With Physical Network Coding. *IEEE Signal Processing Letters*, 28, 997-1001.

9. Boiko, J., Druzhynin, V., Buchyk, S., Pyatin, I., & Kulko, A. (2024). Methodology of FPGA Implementation and Performance Evaluation of Polar Coding for 5G Communications. *CEUR Workshop Proceedings*, 3654, 15-24.

10. Boiko J.M., & Pyatin I.S. (2024). Determining the performance of LDPC codes using HDL in communication systems WI-FI 6 technology. *Scientific Notes of V.I. Vernadsky Taurida National University*, 35(74),1, 19-26.

11. Halahan, N., Berkman, L., Drobyk, O., Makarenko, A., & Zavatskyi, V. (2025). Devising a method for receiving a multidimensional signal using high-order phase difference modulation in next-generation mobile networks. *Eastern-European Journal of Enterprise Technologies*, 135(9), 19-32.

**ДОДАТОК Г.**

Презентаційні матеріали за результатами виконання дипломної роботи



# Кваліфікаційна робота



Рівень вищої освіти: Другий (магістерський)

Галузь знань: Електроніка, автоматизація та електронні комунікації

Спеціальність: 172 Електронні комунікації та радіотехніка

Освітня програма: Електронні інформаційно-комунікаційні системи та мережі

ФАКУЛЬТЕТ  
ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ

## Метод інтелектуального декодування завдаючих кодів у 5G/6G



Виконала:

здобувач курсу, група ЕКРм-24-1

**Суберляк Дар'я Олександрівна**

Науковий керівник:

д-р техн. наук, професор

**Бойко Юлій Миколайович**

Кафедра  
телекомунікацій,  
медійних та  
інтелектуальних  
технологій

**МЕТА:**

розробка методу інтелектуального декодування заводостійких кодів LDPC та Polar у системах 5G/6G, яка забезпечує підвищення ефективності декодування в умовах шумів і завод за рахунок використання методів машинного навчання

**ОБ'ЄКТ:**

процес заводостійкого кодування і декодування у бездротових системах зв'язку п'ятого та шостого покоління

**ПРЕДМЕТ:**

методи та алгоритми інтелектуального декодування LDPC та Polar кодів у системі 5G NR та 6G

## Актуальність

Мобільні системи зв'язку 5G і майбутні 6G технології пред'являють надвисокі вимоги до надійності, низької затримки та адаптивності в складних радіоумовах. Саме фізичний рівень (PHY) та завадостійке кодування (LDPC, Polar) є ключем до досягнення цільових характеристик. Інтелектуальні методи декодування на базі Deep Reinforcement Learning дозволяють суттєво зменшити затримки, енергоспоживання та підвищити стійкість до завад, що критично важливо для URLLC, AR/VR, IIoT та сценаріїв 6G. Розроблена методика відкриває перспективи створення адаптивних, енергоефективних модемів нового покоління.

## Задачі наукового дослідження:

- 1) Провести аналіз основних характеристик каналів зв'язку в 5G та вимог до завадостійкого кодування.
- 2) Дослідити класичні алгоритми декодування LDPC та Polar кодів, їх переваги та обмеження в контексті застосування в 5G.
- 3) Розробити архітектуру інтелектуального декодування з використанням сучасних методів машинного навчання.
- 4) Провести моделювання та порівняльний аналіз ефективності традиційного і інтелектуального підходів до декодування.

## Deep Reinforcement Learning (глибоке навчання з підкріпленням)

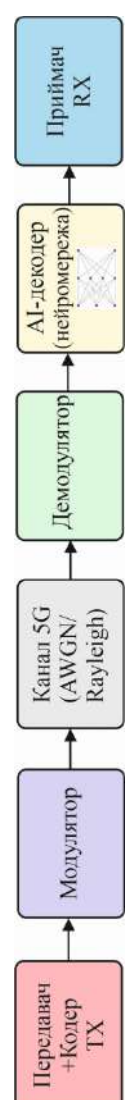
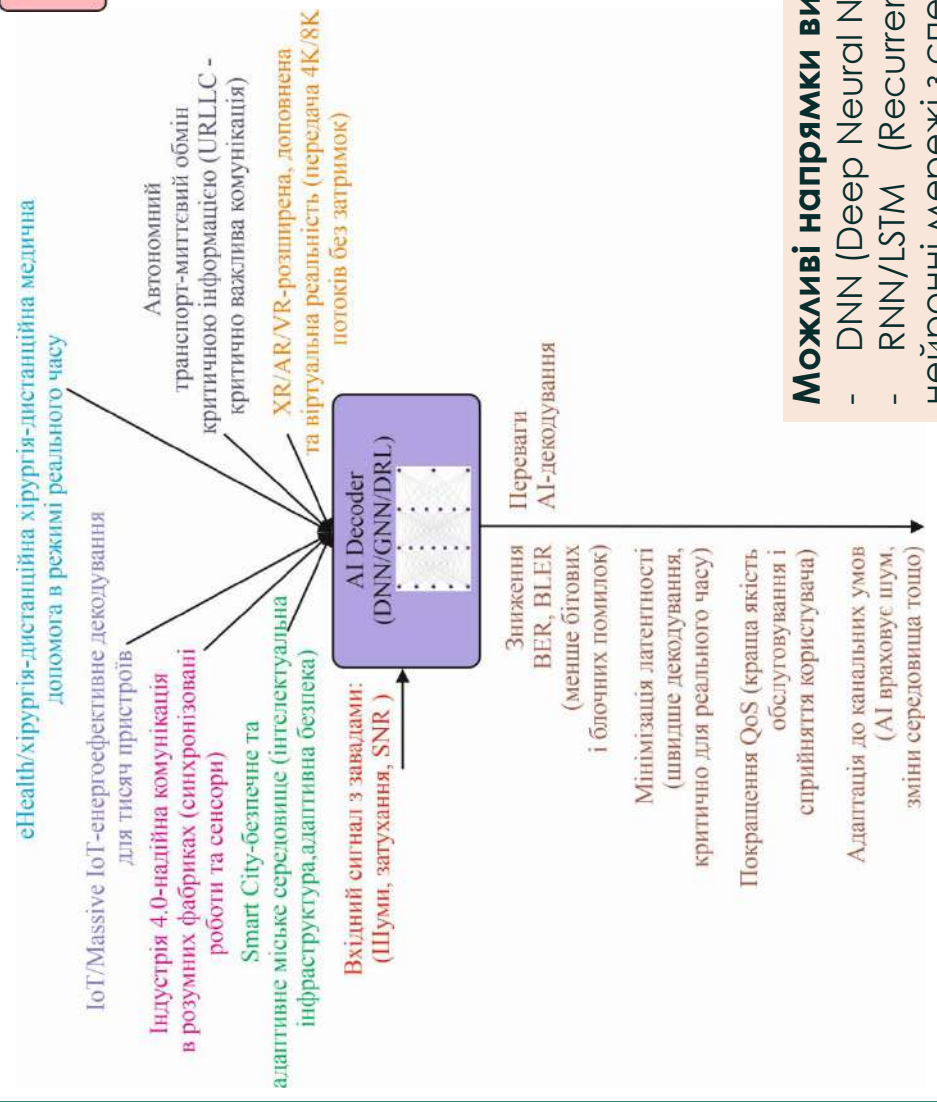
- це метод штучного інтелекту, де агент (декодер) навчається приймати рішення, взаємодіючи з середовищем. Він отримує винагороди за вдалі дії та покарання за помилки, і на основі цього вчиться діяти краще

Наукова новизна отриманих результатів:

1. **Дістала подальшого розвитку методика інтеграції алгоритмів штучного інтелекту, зокрема методів глибокого навчання та навчання з підкріпленням, у процес декодування LDPC та Polar кодів у беззаротових системах п'ятого покоління.**
2. **Вперше запропоновано нову архітектуру DRL+Soft-декодера для каналів 5G/6G, здатну адаптувати стратегію декодування залежно від типу завад та умов каналу.**
3. **Вперше здійснено повноцінне порівняння DRL-декодера з класичними алгоритмами у симуляційному середовищі MATLAB із багатьма типами шуму (AWGN, Rayleigh, імпульсний, фазовий).**



# AI-декодування в 5G/6G - покращення та сфери застосування



Таблиця 1 - Практичні ефекти впровадження AI-декодування через ключові метрики покращення

Показник	Класичні методи	AI-декодування
BER (Bit Error Rate)	$\sim 10^{-5}$ при низькому SNR	$\sim 10^{-7}$ при низькому SNR
Latency	1-5 мс	< 1 мс (для URLLC)
QoS адаптація	Жорстко задана	Динамічна, контекстна
Споживання енергії	Високе при складному декодуванні	Оптимізоване, енергоефективне рішення (energy aware) рішення
Масштабованість	Лімітована складність	Висока, завдяки навчанню
Інтеграція в embedded (вбудовані системи)	Важка	Можлива за допомогою quantized AI моделей (квантовані моделі - оптимізовані, низькорозрядні)

## Можливі напрямки використання архітектур AI для синтезу декодера:

- DNN (Deep Neural Networks) - глибокі нейронні мережі;
- RNN/LSTM (Recurrent Neural Network/Long Short-Term Memory - рекурентні нейронні мережі з спеціальними механізмами пам'яті);
- GNN (Graph Neural Networks) - графові нейронні мережі;
- DRL (Deep Reinforcement Learning) - концепція глибокого навчання з підкріпленням

Рис. 5 – AI-декодування в 5G/6G - покращення та сфери застосування

# Архітектура DNN (Deep Neural Networks) для розробки інтелектуального декодера 5G



Табл. 2 – Стратегія формування та призначення нейронних шарів при побудові інтелектуального декодера 5G

Рис. 6 – Блок схема декодера LDPC-коду яка відповідає архітектурі DNN-декодера

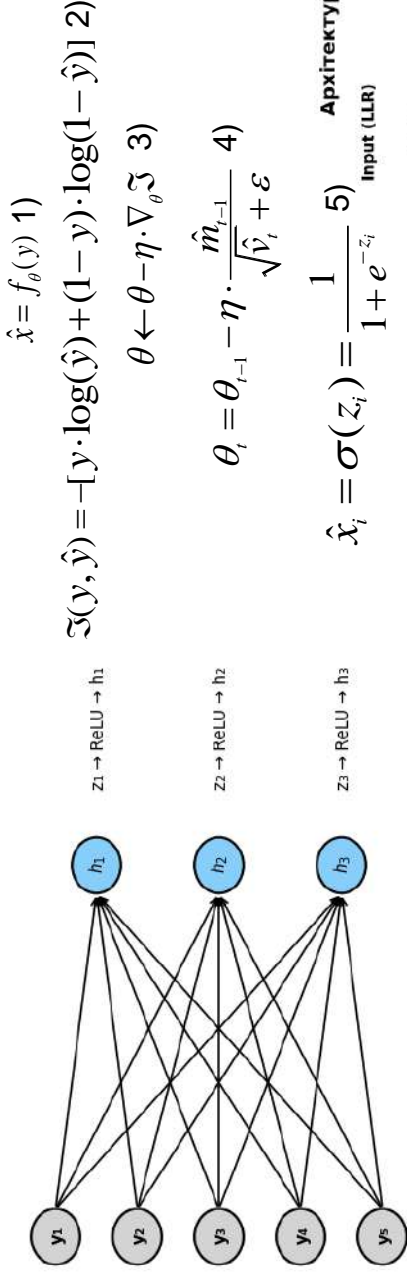


Рис. 7 – Фрагмент нейронної архітектури DNN-декодера (один прихований шар з ReLU)=локальна нейронна діаграма шару

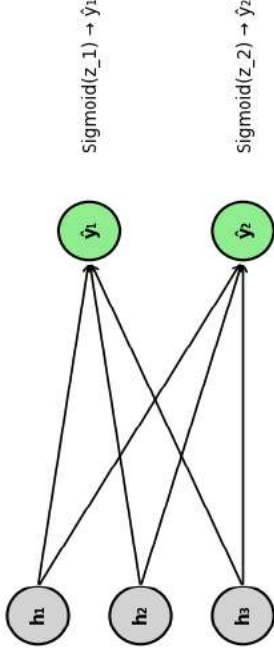
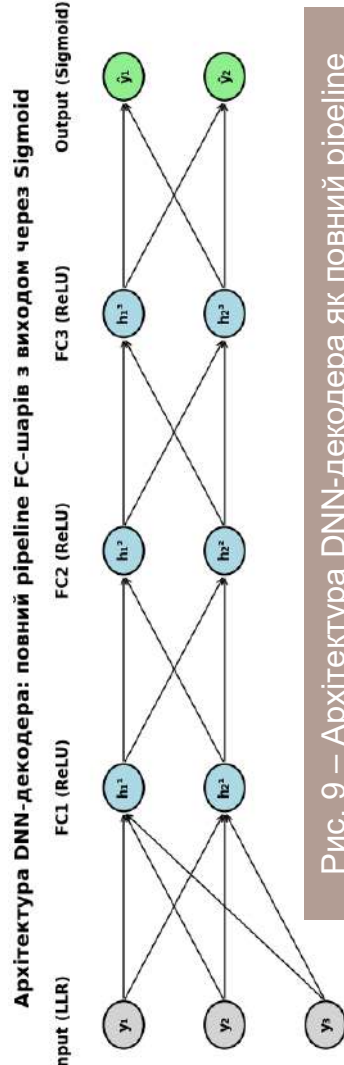


Рис. 8 – Схема вихідного шару DNN-декодера з Sigmoid (показано процес отримання soft (м'яких)-рішень)

Шар	Призначення
FC1 (512)	Початковий аналіз спотвореного вектору
FC2 (256)	Узагальнення ключових бітових патернів
FC3 (128)	Вибір найімовірнішої комбінації інформаційних бітів
Output	Soft-оцінки K-бітів (Ймовірності)



Архітектура DNN-декодера: повний pipeline FC-шарів з виходом через Sigmoid

Рис. 9 – Архітектура DNN-декодера як повний pipeline FC-шарів (повний конвеєр) з виходом через Sigmoid

# Використання архітектури рекурентних нейронних мереж RNN/LSTM (при використанні каналу з пам'яттю) для конструювання інтелектуального декодера 5G

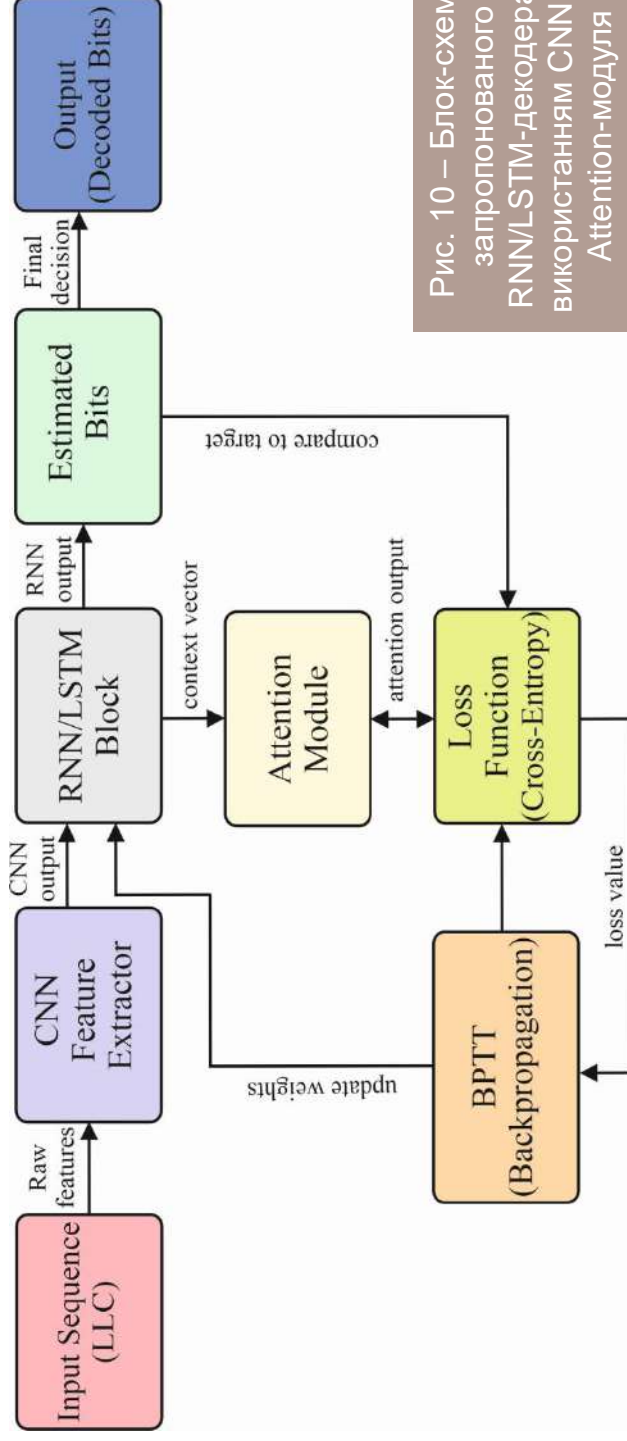


Рис. 10 – Блок-схема запропонованого RNN/LSTM-декодера з використанням CNN та Attention-модуля

$$h_t = \Psi(W_{xt}x_t + W_{hh}h_{t-1} + b_h) \quad 6)$$

$$y_t = \phi(W_{hy}h_t + b_y) \quad 7)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad 8)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad 9)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad 10)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad 11)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad 12)$$

$$h_t = o_t \odot \tanh(C_t) \quad 13)$$

Перевага запропонованої схеми RNN/LSTM-декодера з CNN та Attention полягає в її здатності ефективно враховувати часові залежності у сигналі, що особливо важливо для каналів з пам'яттю, таких як 5G, де присутні міжсимвольна інтерференція та доплерівський зсув. CNN шар виділяє локальні ознаки з вхідного сигналу, покращуючи якість подальшої обробки, а Attention-модуль допомагає мережі фокусуватися на найбільш релевантних частинах послідовності, підвищуючи точність декодування. Завдяки механізмам гейтів LSTM усуває проблему затухання та вибуху градієнтів, забезпечуючи стабільне і ефективне навчання навіть на довгих послідовностях. В результаті ця архітектура демонструє високу адаптивність до складних умов каналу 5G і дозволяє отримувати більш точне і надійне відновлення інформації. Гейти — це спеціальні "регулятори" або механізми в LSTM, які керують потоком інформації всередині нейронної мережі. Вони вирішують, яку частину інформації потрібно зберегти, яку — стерти, а яку — додати в пам'ять комірки.

# Особливості застосування концепції Graph Neural Networks (GNN) для LDPC графів при розробці декодерів 5G/6G

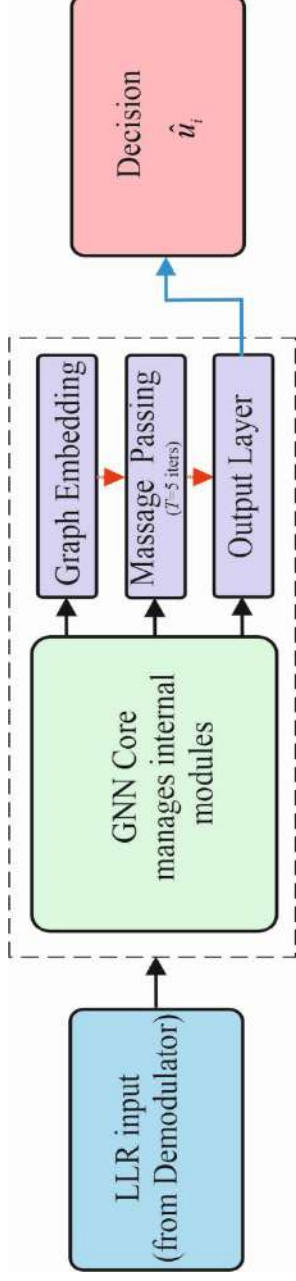


Рис. 11 – Блок-схема інтелектуального GNN-декодера LDPC для 5G/6G

$$m_{i \rightarrow j}^{(t+1)} = \phi_m^{(t)}(h_i^{(t)}, h_j^{(t)}, e_{ij}) \quad 14)$$

$$h_i^{(t+1)} = \phi_u^{(t)} \left( h_i^{(t)}, \sum_{j \in N(i)} m_{j \rightarrow i}^{(t+1)} \right) \quad 15)$$

$$y = \sigma(Wx + b) \quad 16)$$

$$h_i^{(0)} = \text{LLR}(v_i) \quad 17)$$

$$u_i = \begin{cases} 0, & \text{якщо } h_i^{(T)} \geq 0 \\ 1, & \text{в іншому випадку} \end{cases} \quad 18)$$

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \quad 19)$$

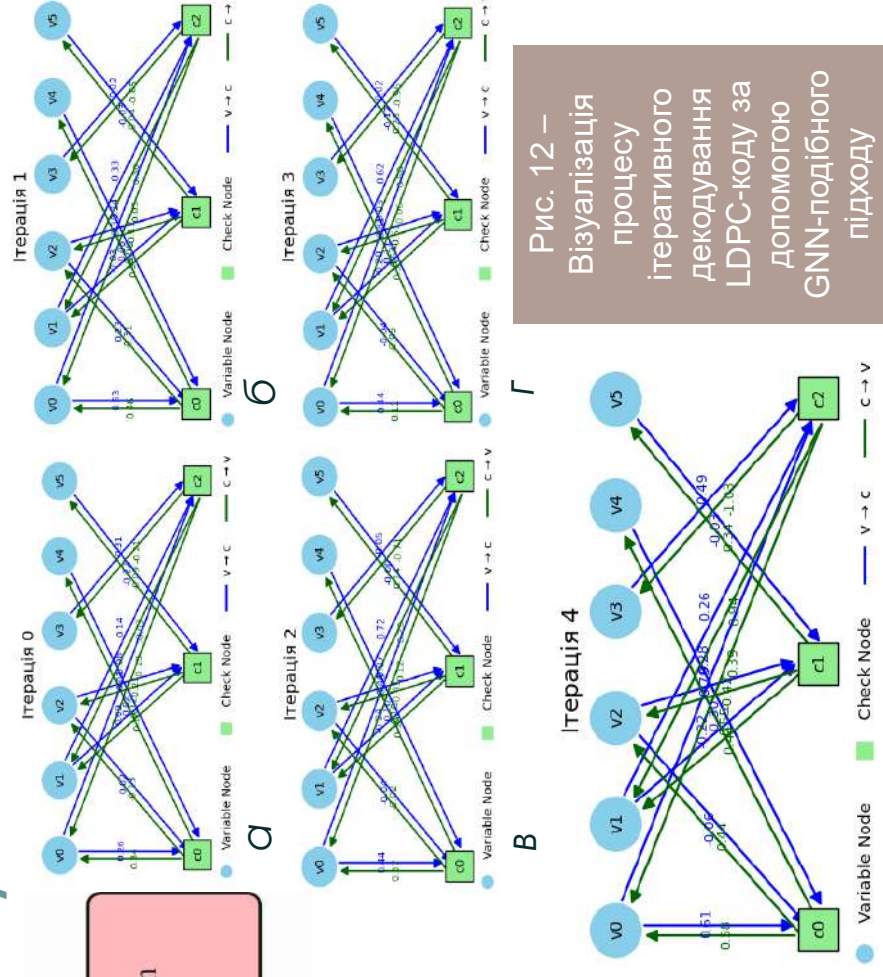


Рис. 12 – Візуалізація процесу ітеративного декодування LDPC-коду за допомогою GNN-подібного підходу

A

# Використання концепції Deep Reinforcement Learning (DRL) – обґрунтування вибору та методика конструювання декодерів (LDPC, Polar codes) для 5G/6G

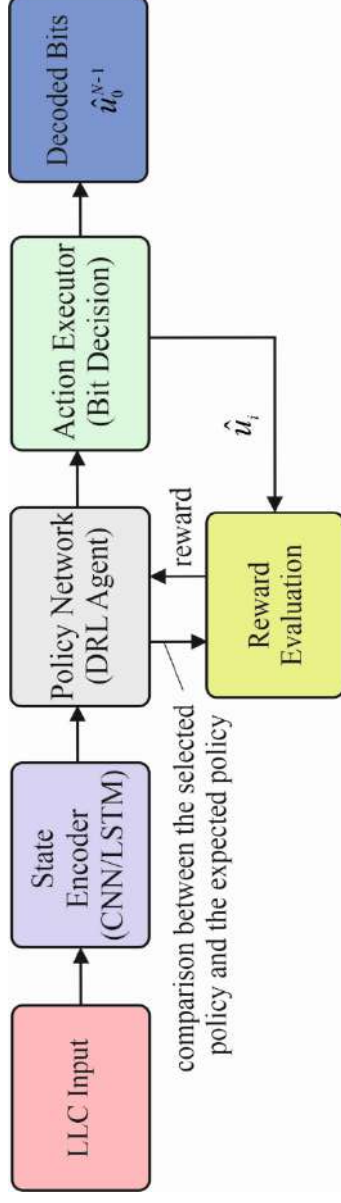


Рис. 12 – Блок-схема синтезу ваного декодери на основи концепції DRL

$$\max_{\pi} \mathbb{E} \left[ \sum_{t=0}^T \gamma^t r_t \right] \quad 19)$$

$$V(s_t) = \mathbb{E}_{\pi} \left[ \sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k} \mid s_t \right] \quad 20)$$

$$Q(s_t, a_t) = \mathbb{E}_{\pi} \left[ \sum_{k=0}^{\infty} \gamma^k \cdot r_{t+k} \mid s_t, a_t \right] \quad 21)$$

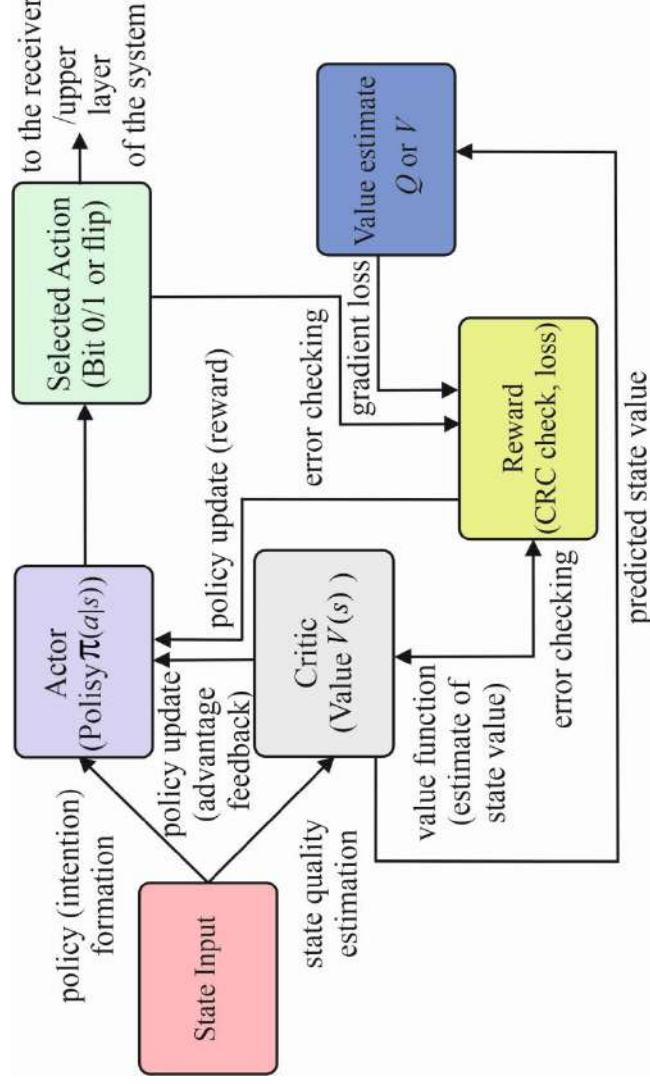
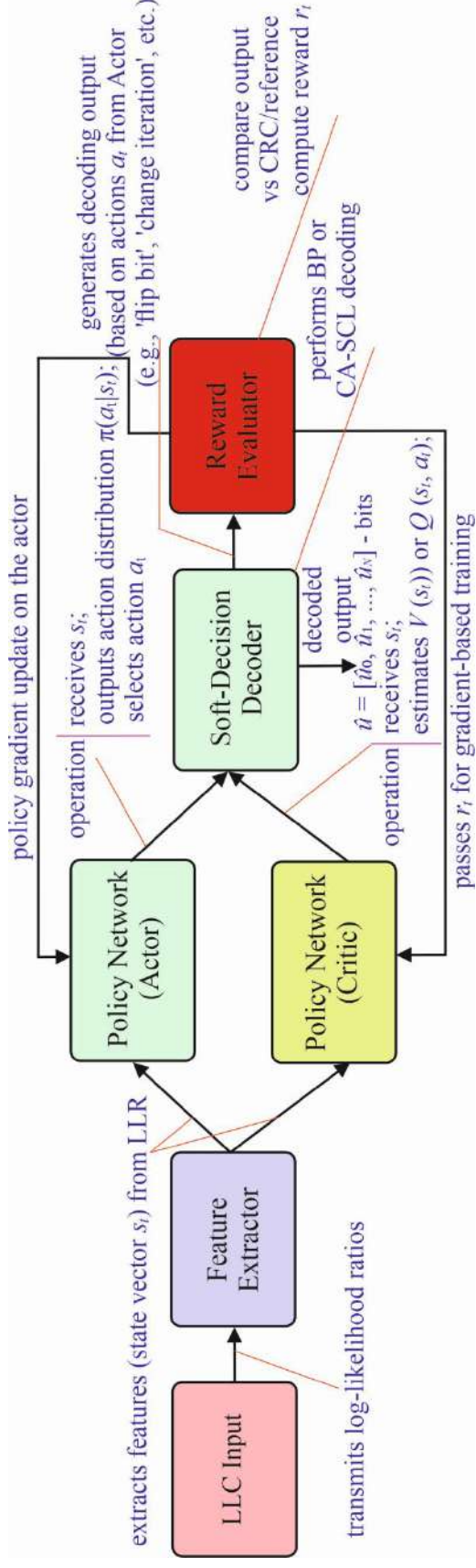


Рис. 13 – Actor-Critic модель при використанні підходу з підкріпленням (DRL) для інтелектуального декодери 5G/6G

Ключова ідея яка запропонована в магістерській роботі – агент (декодер) постійно адаптує свою поведінку, враховуючи не лише результат, але й ймовірнісні наміри, які він мав на момент прийняття рішення.

Запропонований в магістерській роботі математичний базис та схеми на рис. 12 та 13 дозволяють вчити агента (декодер) діяти оптимально, з урахуванням стохастичності і пам'яті каналу.

# Архітектура пропонуваної системи гібридного інтелектуального декодування (DRL+Soft-Decision декодер) для 5G/6G



- Основні переваги гібридного інтелектуального декодера:
- адаптивність до змін каналу;
  - стабільна точність;
  - можливість реалізації в реальному часі (наприклад, на FPGA)

Рис. 14 – Блок-схема гібридної архітектури інтелектуального декодера (DRL Soft-Decision) для 5G/6G

$$J(\theta) = \mathbb{E}_{s_t \sim \pi_\theta} [R_t] = \mathbb{E}_{s_t \sim \pi_\theta} \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k} \right] \quad 22$$

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) (R_t - V(s_t)) \quad 23$$

$$a_t = \text{soft max}(W^{(1)} s_t + b^{(1)}) \quad 24$$

Ціль DRL-агента — максимізувати очікувану довгострокову винагороду при декодуванні (менше помилок при декодуванні)

Гرادієнт оновлює ваги у нейронмережі Actor-Critic, якщо обрана дія покращила результат (тобто зменшила помилку)

Математична модель виходу нейронмережі. Вихід Actor — ймовірності дій (наприклад, фліп біта) залежно від стану системи

# Структурні моделі телекомунікаційного каналу з інтелектуальним декодуванням завдостіких кодів для 5G

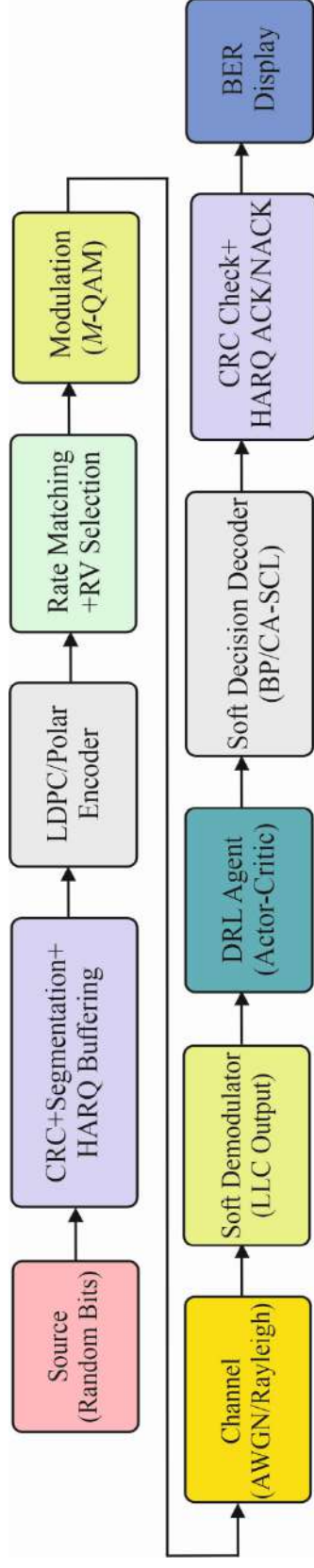


Рис. 15 – Схема структурна архітектури фізичного каналу PDSCN у 5G з інтелектуальним гібридним декодером DRL+Soft

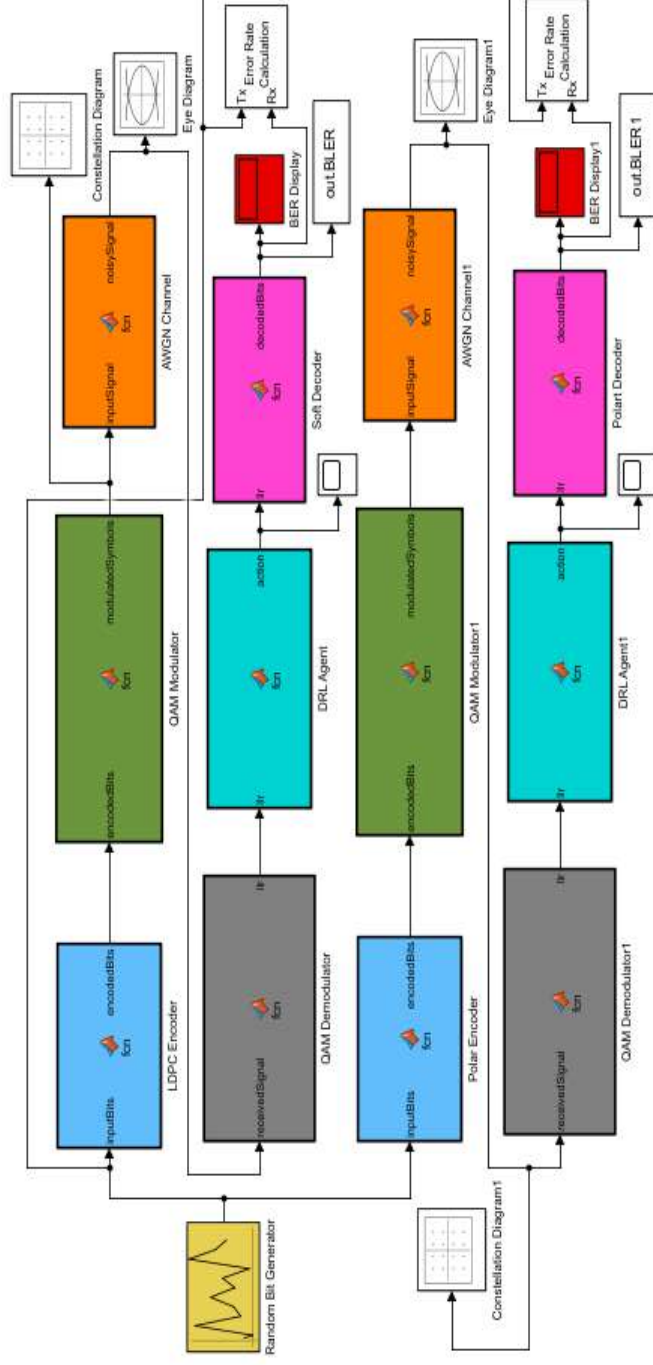


Рис. 16 – Загальна структурна схема імітаційної моделі фізичного каналу PDSCN у 5G з інтелектуальним гібридним декодером DRL+Soft

# Експериментальне дослідження продуктивності системи інтелектуального декодування QC-LDPC-кодів на основі DRL у каналі AWGN та Релеєвському каналі

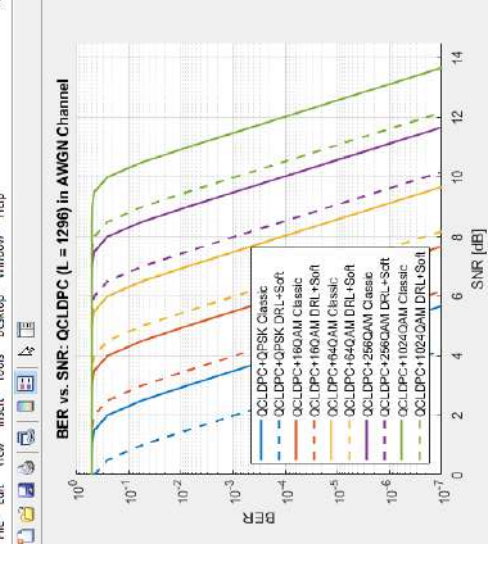
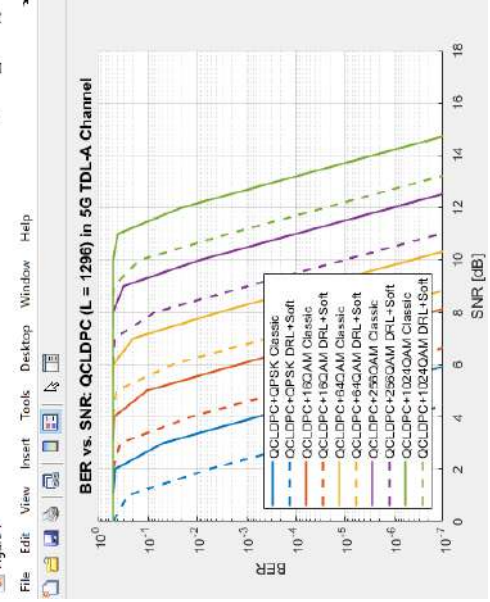
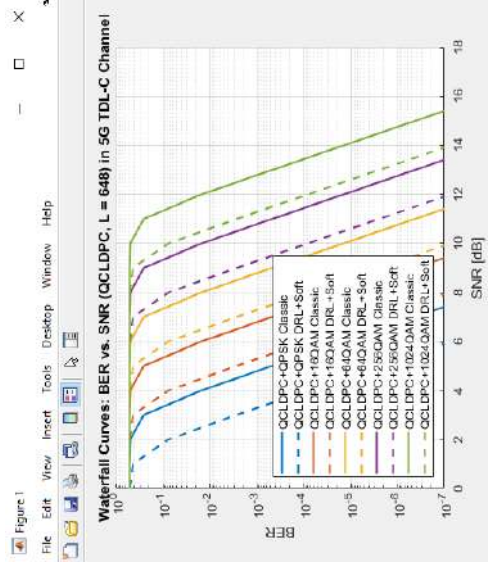
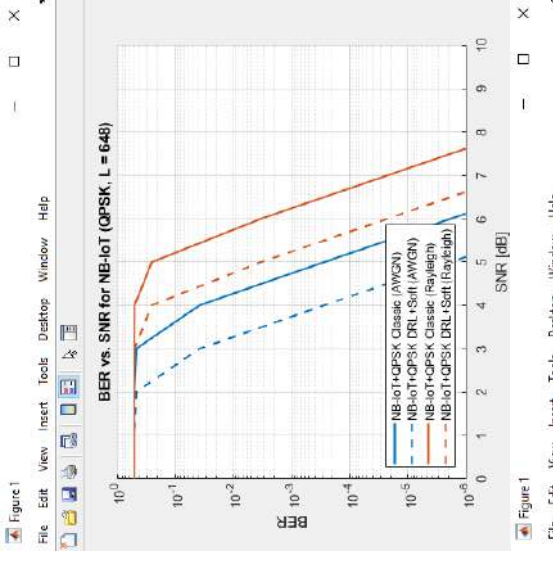
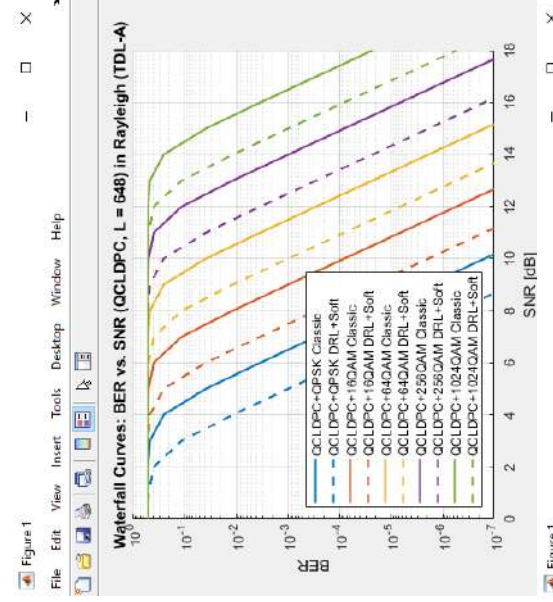
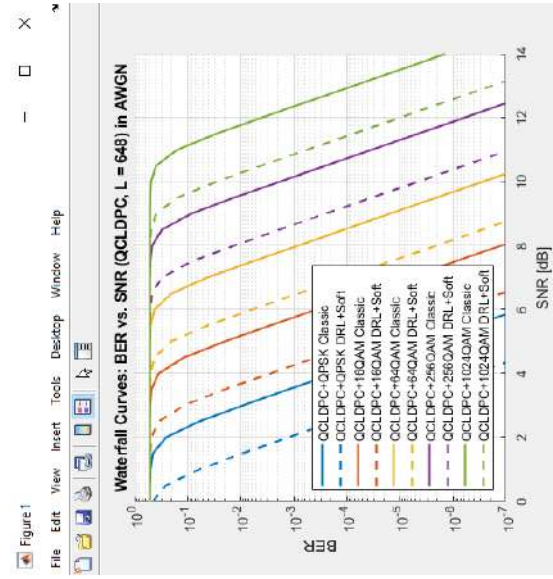


Рис. 17 – Графіки завадостікості BER vs SNR

а

б

в

г

д

е

# Експериментальне дослідження продуктивності системи інтелектуального декодування Polar-кодів на основі DRL у каналі AWGN та Релеєвському каналі

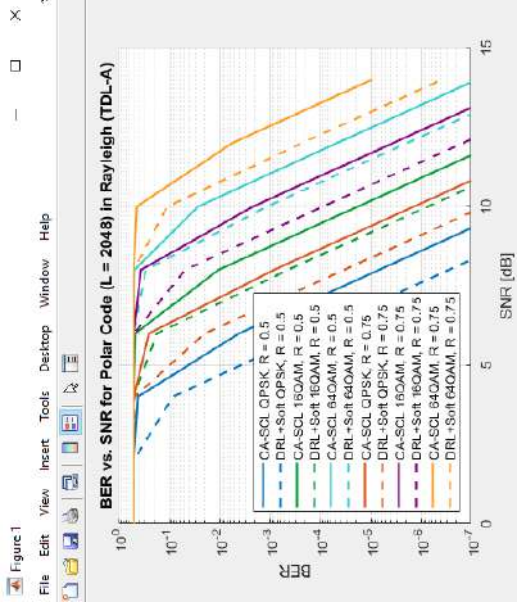
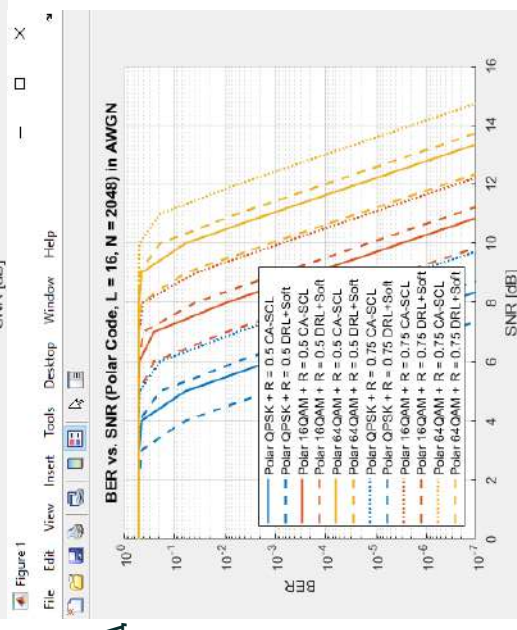
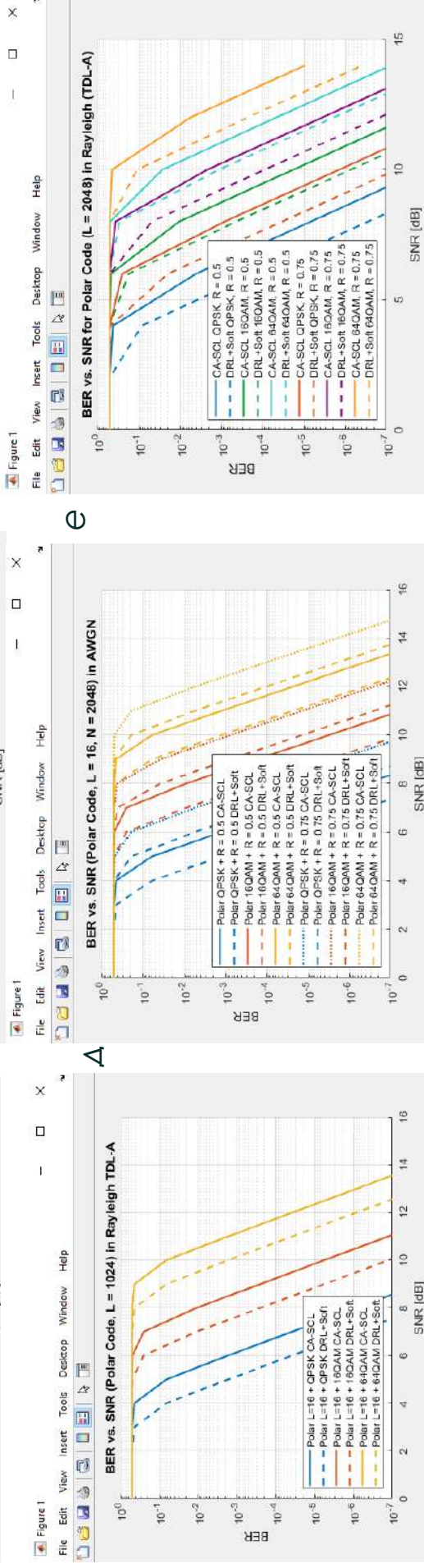
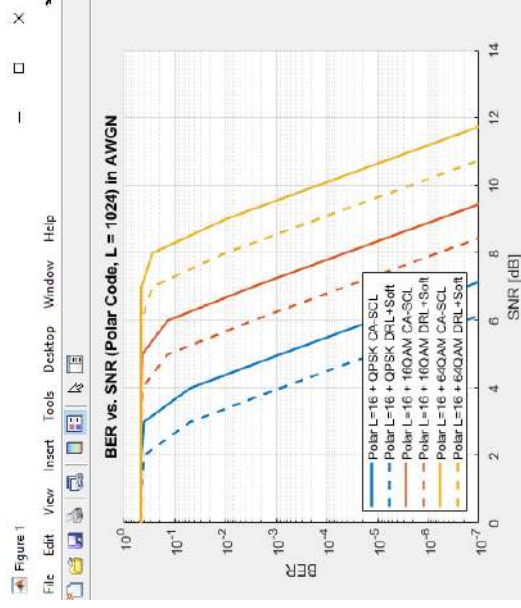
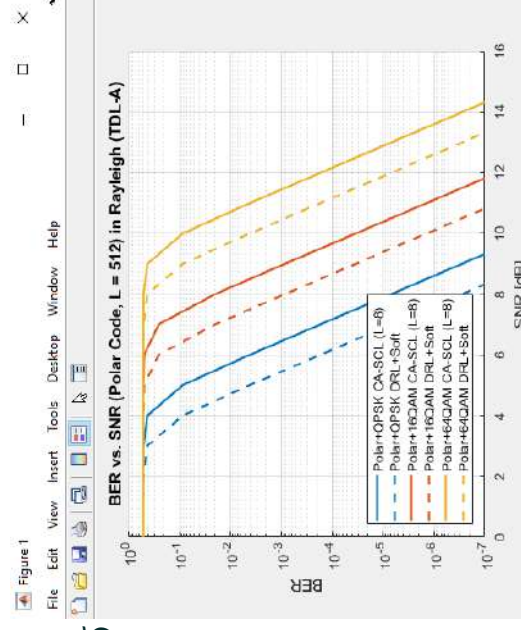
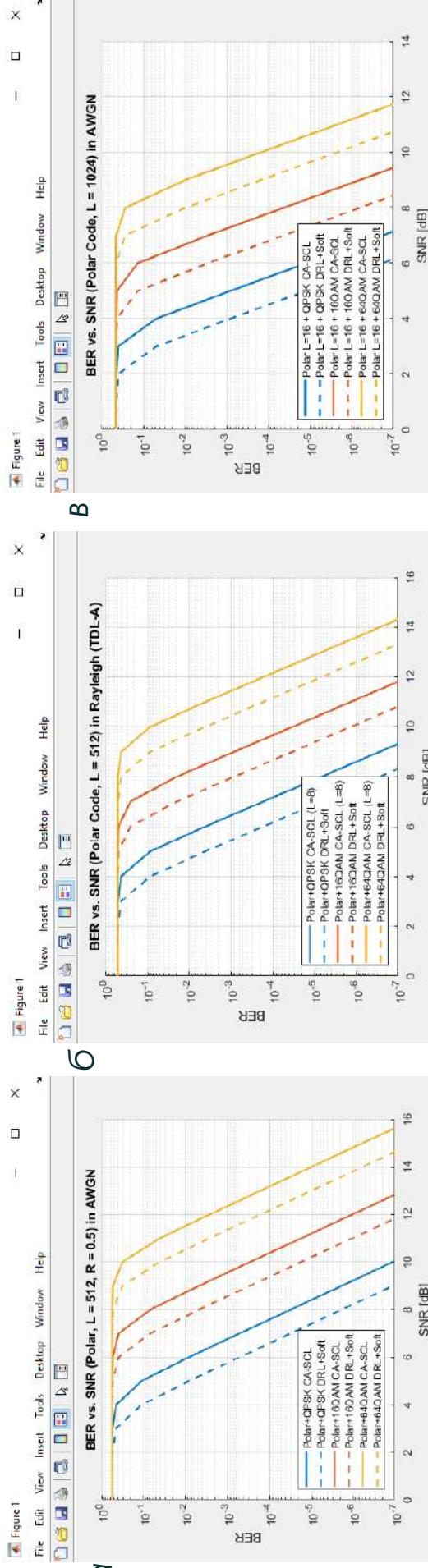


Рис. 18 – Графіки завадостійкості BER vs SNR

# 3D графіки завадостійкості в залежності від кількості ітерацій декодування при порівнянні класичного декодера і інтелектуального

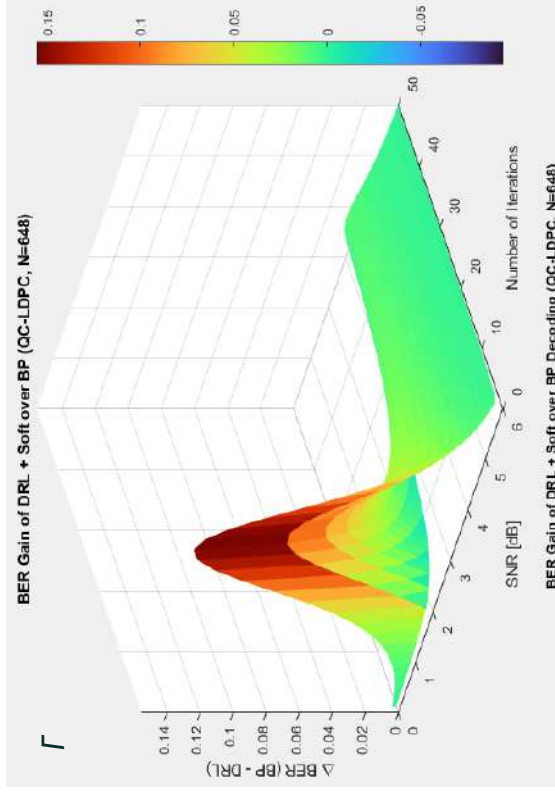
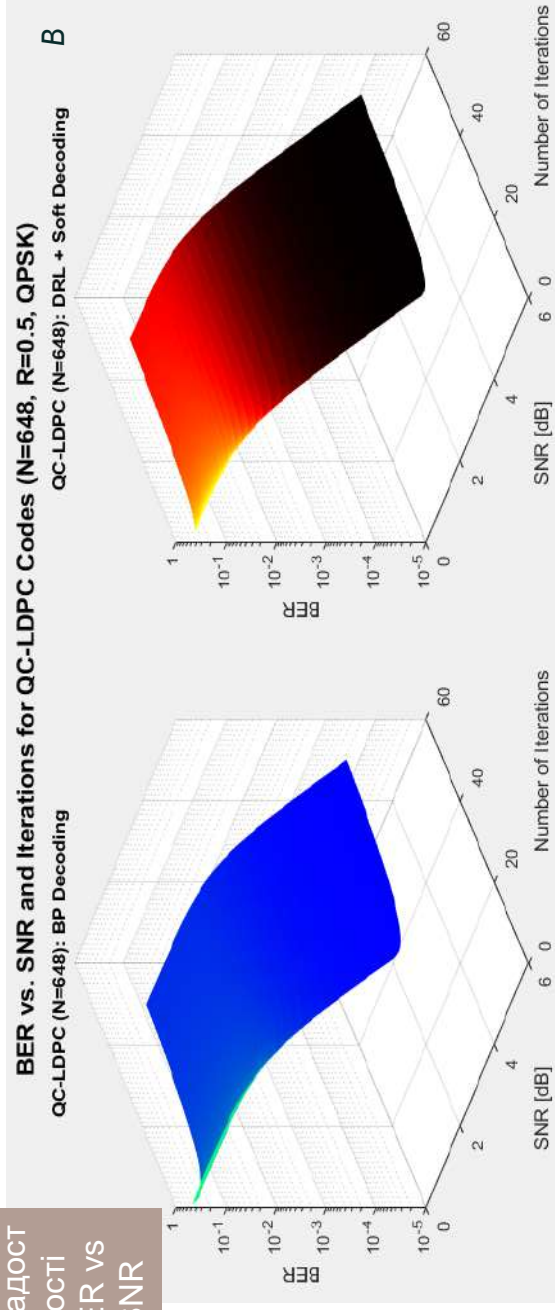
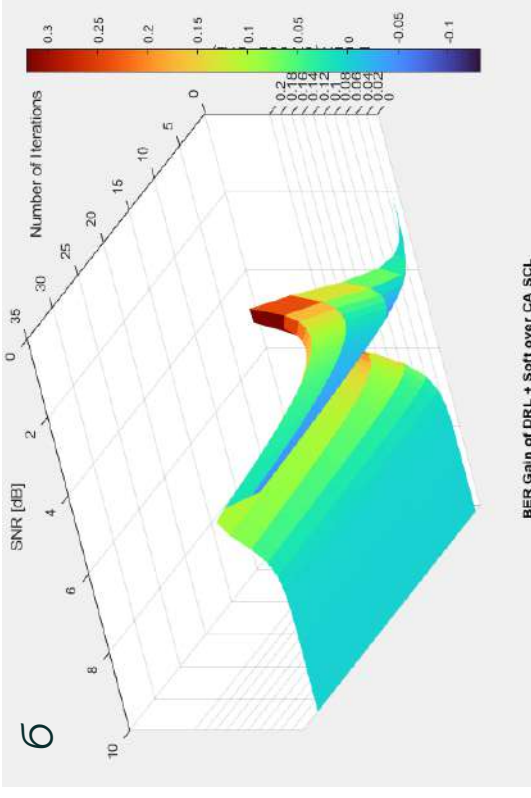
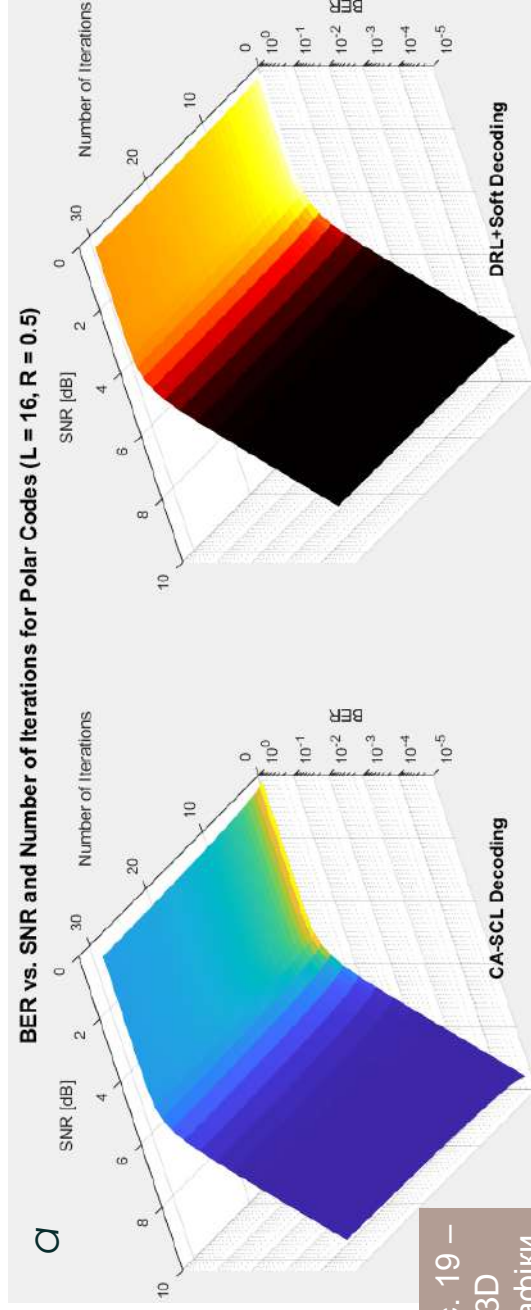


Рис. 19 – 3D Графіки завадостійкості ікості BER vs SNR

# Оцінка продуктивності каналу 5G з інтелектуальним декодуванням з індикатором BLER

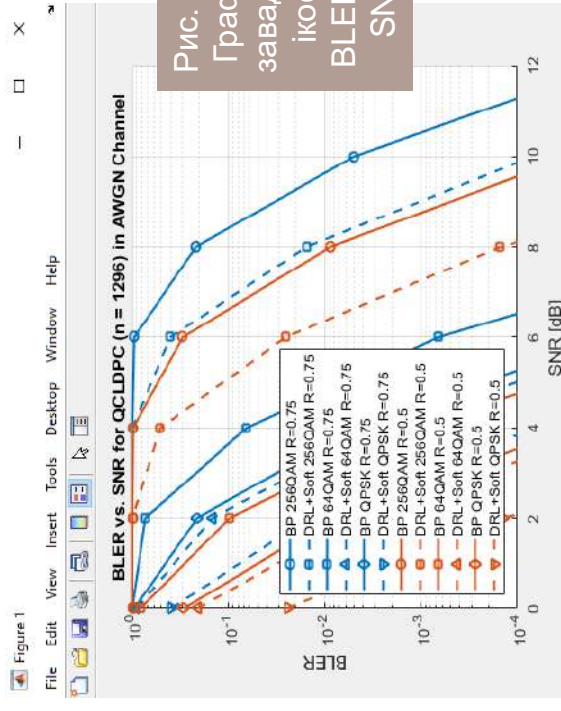
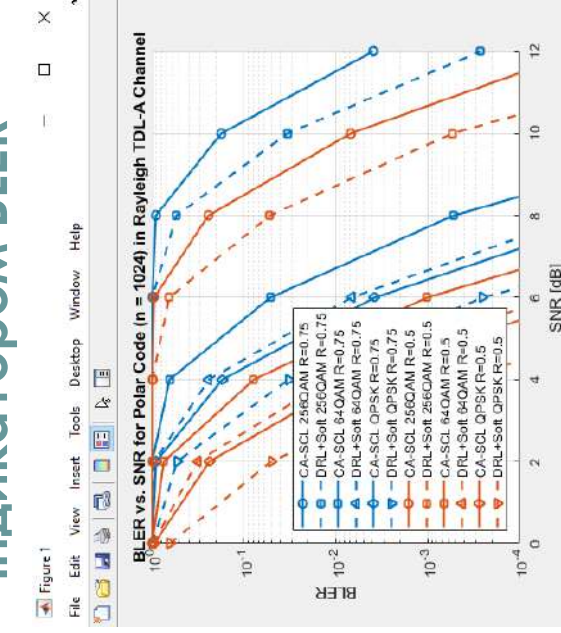
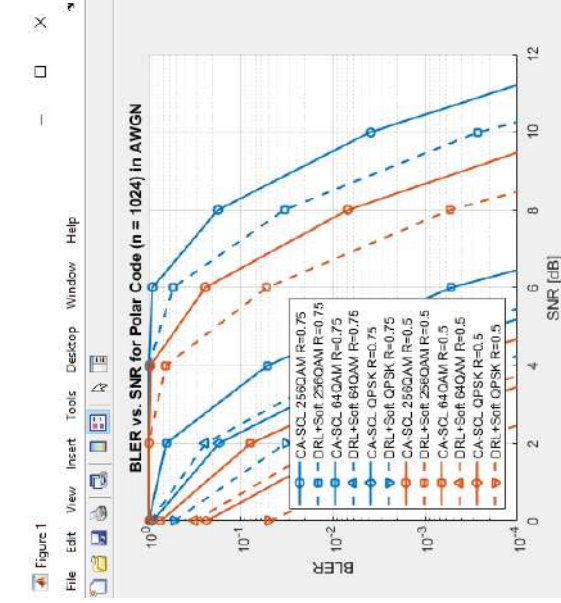
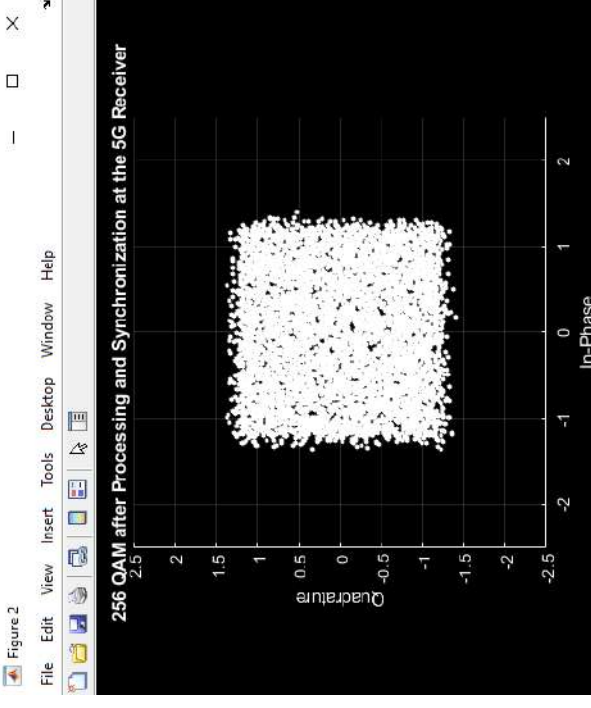
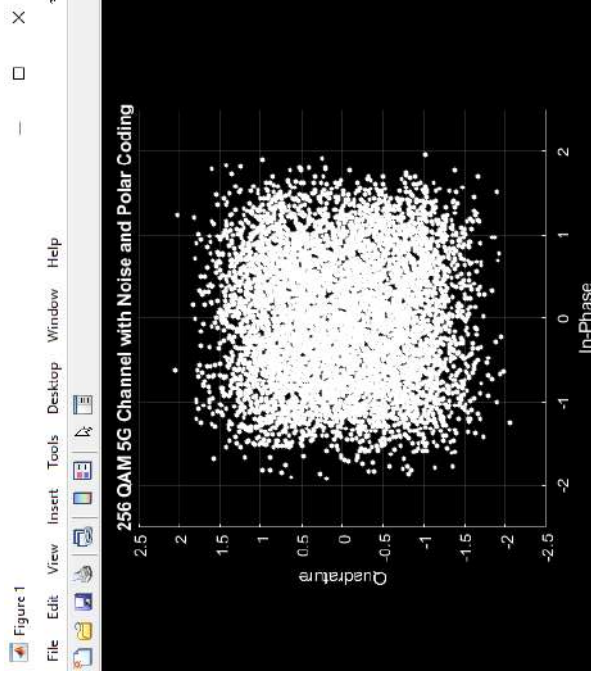
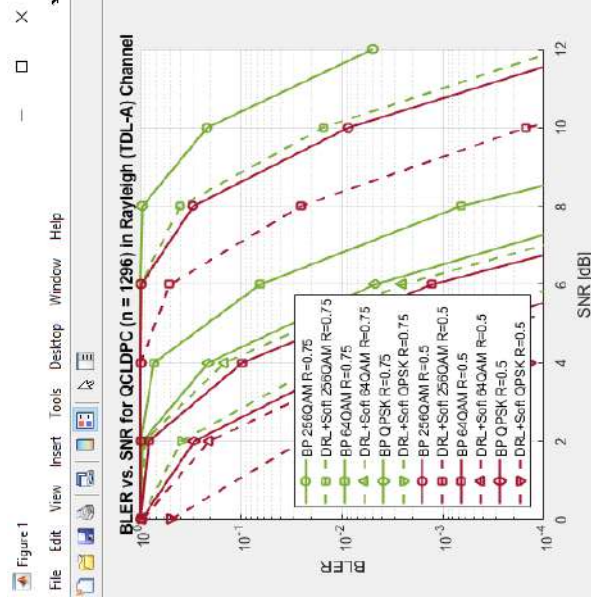


Рис. 20 –  
Графіки  
завадост  
ікості  
BLER vs  
SNR



# Дослідження основних метрик продуктивності для DRL-декодера (DRL+Soft)

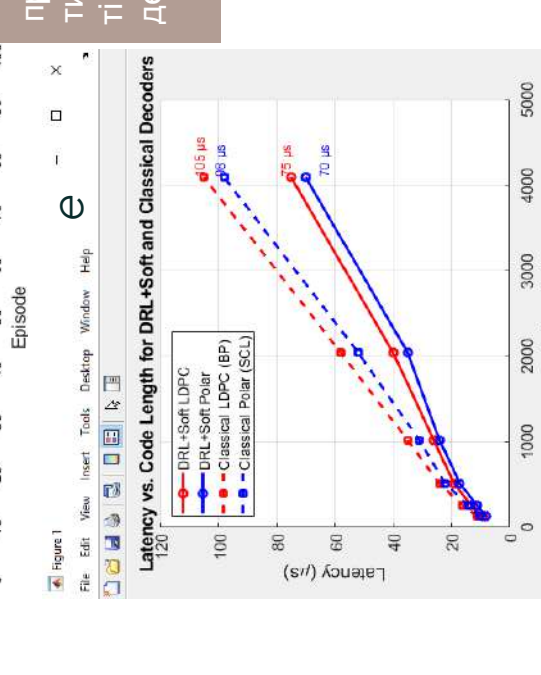
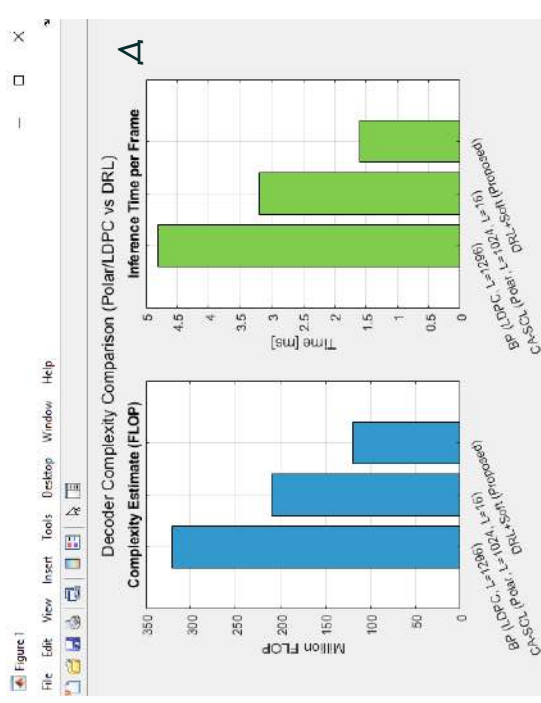
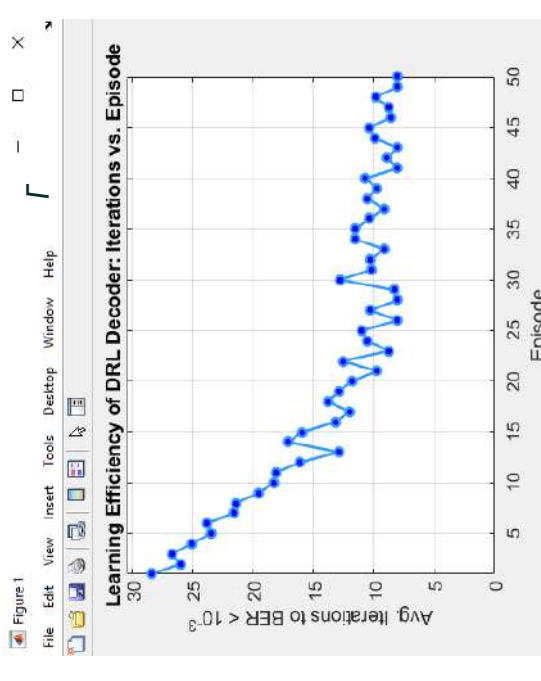
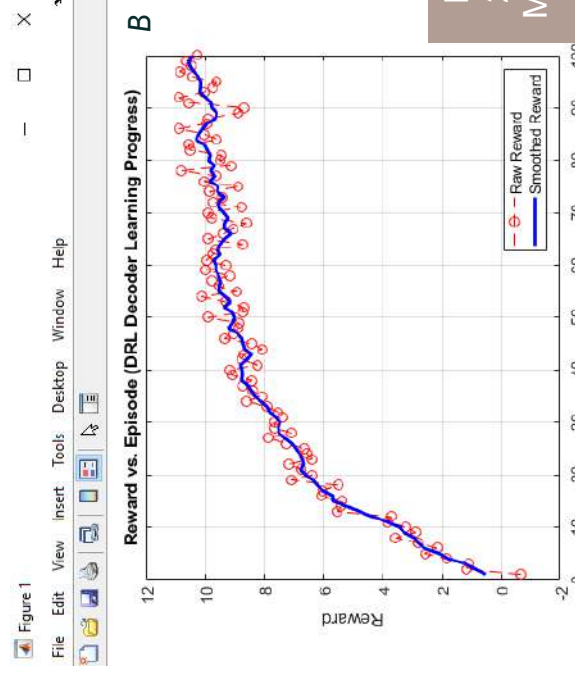
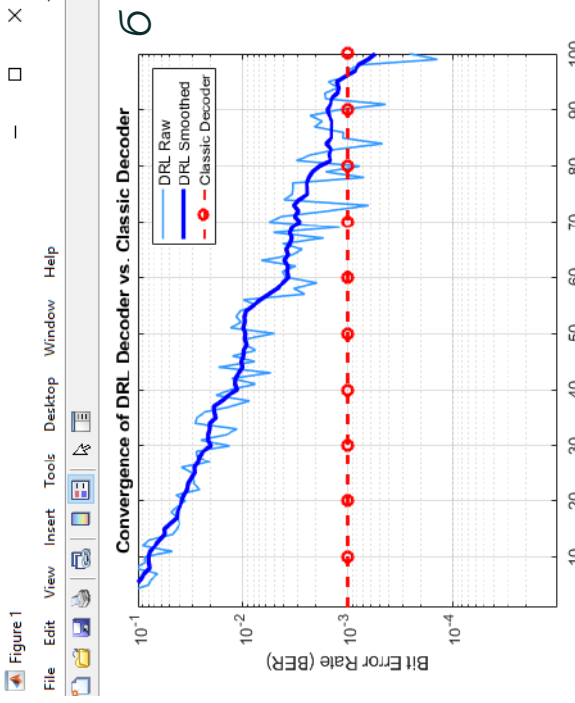
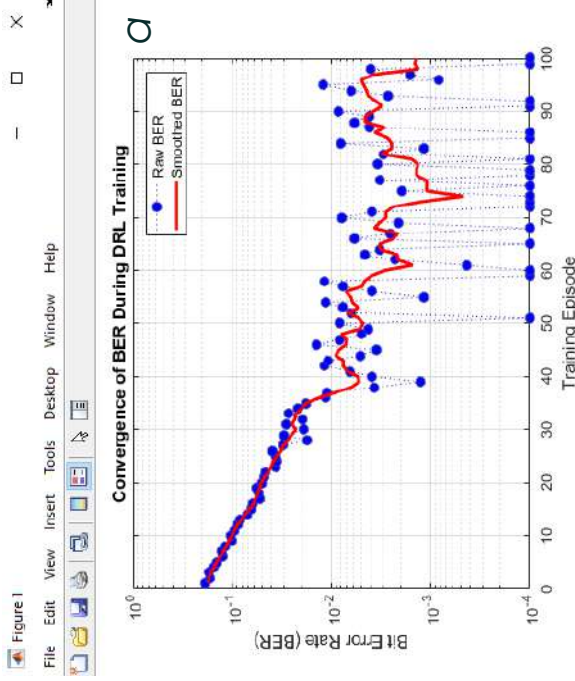


Рис. 21  
Метри  
КИ  
продук  
ТИВНОС  
ті DRL-  
декоде  
ру

# ВИСНОВКИ

На основі експериментального дослідження індикатору завадостійкості BER від SNR вдалось встановити, що DRL+Soft декодер забезпечує виграш до 2 дБ для LDPC і Polar кодів порівняно з BP та CA-SCL відповідно.

На основі експериментального дослідження індикатору завадостійкості BLER від SNR було встановлено, що покращення від DRL особливо помітне у діапазоні BLER  $10^{-2}$ – $10^{-3}$ , що наближається до URLLC-вимог.

Була виявлена і підтверджена експериментально робастність запропонованого інтелектуального декодера до різних типів шуму. А саме при фазових та імпульсних завадах класичні декодери “здаються” тобто втрачають продуктивність, тоді як DRL декодер впевнено зберігає ефективність (виграш склав ~2 дБ).

Була підтверджена експериментально висока швидкість навчання. Конвергенція DRL-агента спостерігається вже після 20–40 епізодів.

Результати експерименту показали, що запропонований інтелектуальний декодер (DRL-декодер) у складі 5G/6G систем може працювати зі зниженим числом ітерацій і споживати менше FLOP/Inference time (інтерфейсний час), тобто володіє оптимізованою обчислювальною складністю.

Отримані результати свідчать на користь того, що система DRL-декодера показує стійкість (адаптивність) при зміні характеристик каналу, що особливо важливо для 6G-умов

## Основні перспективи поширення отриманих рішень на системи технології 6G наступні:

- 1) Запропоновані DRL-декодери можуть стати ядром когнітивних приймачів, що здатні адаптувати політику декодування в реальному часі.
- 2) Підтверджене експериментально в дипломній роботі підвищення стійкості до непередбачуваних завад, при застосуванні запропонованого інтелектуального декодера, критично важливе в сценаріях з щільним розміщенням пристроїв або перешкодами (Smart City, Smart Factory).
- 3) Отриманий результат зменшення FLOP дозволить знизити енергоспоживання мобільних пристроїв, що є ключовою вимогою для розгортання мереж майбутнього типу 6G-IoT.
- 4) DRL-агенти можуть бути реперограмовані або донавлені без заміни апаратної частини тобто запропоновані в дипломній роботі рішення дозволяють здійснити інтеграцію з оновлюваними моделями, що критично важливо для довшого життєвого циклу систем 6G.



**Дякую за увагу!**

## Відгук

на дипломну роботу «Методика інтелектуального декодування завадостійких кодів у 5G/6G» студентки групи ЕКР<sub>м</sub>-24-1 Дар'ї Суберляк

Дипломна робота ст. гр. ЕКР<sub>м</sub>-24-1 СУБЕРЛЯК Дар'ї присвячена актуальній проблемі підвищення ефективності декодування завадостійких кодів у бездротових телекомунікаційних системах нового покоління. У роботі запропоновано та реалізовано методику інтелектуального декодування LDPC та Polar кодів із використанням підходів глибокого навчання та навчання з підкріпленням (Deep Reinforcement Learning), що дало змогу досягти покращення показників якості прийому в умовах завад та шумів.

Автором роботи проведено побудову моделі фізичного каналу 5G із варіативними типами шуму (AWGN, Rayleigh fading, імпульсний та фазовий шум), виконано порівняльний аналіз класичних та інтелектуальних алгоритмів декодування за критеріями BER, BLER, затримки, складності та адаптивності. Особливу увагу приділено моделюванню DRL-архітектури на рівні MATLAB, включно з візуалізацією основних метрик навчання агента.

Студентка гр. ЕКР<sub>м</sub>-24-1 СУБЕРЛЯК Дар'ї проявила високий рівень зацікавленості у тематиці дослідження, ініціативність і наполегливість у процесі виконання всіх етапів роботи — від аналітичного огляду до моделювання складних інтелектуальних архітектур. Вона впевнено володіє сучасним програмним забезпеченням, демонструє глибокі знання з теорії кодування, цифрової обробки сигналів та штучного інтелекту, а також сформовані фахові компетентності, які дозволили їй успішно реалізувати поставлені завдання.

Робота має високу практичну цінність для проектування телекомунікаційних систем 5G/6G, а результати дослідження можуть бути використані в майбутніх розробках програмно-апаратних приймачів.

Вважаю, що магістерська кваліфікаційна робота повністю відповідає вимогам до такого виду робіт, а її авторка СУБЕРЛЯК Дар'я заслуговує на оцінку «відмінно».

Професор кафедри ТМІТ, д.т.н.

Юлій БОЙКО

## *Рецензія*

*на кваліфікаційну дипломну роботу виконану за темою «Методика інтелектуального декодування завадостійких кодів у 5G/6G» студентки групи ЕКР<sub>м</sub>-24-1 Дар'ї Суберляк*

Кваліфікаційна робота ст. гр. ЕКР<sub>м</sub>-24-1 Дар'ї Суберляк присвячена актуальному напрямку сучасної телекомунікаційної науки — застосуванню методів штучного інтелекту (ШІ), зокрема глибокого навчання та навчання з підкріпленням, у процесі декодування завадостійких кодів у системах мобільного зв'язку п'ятого та шостого покоління (5G/6G). Враховуючи постійне зростання вимог до швидкодії, надійності та енергоефективності приймачів у бездротових мережах, тематика дослідження є надзвичайно важливою та практично орієнтованою.

Робота має чітку структуру, логічно побудована й методологічно вивірена. У вступі обґрунтовано актуальність проблеми, сформульовано мету дослідження, визначено предмет, об'єкт, завдання та наукову новизну роботи. Всі поставлені завдання системно реалізовано в основній частині, що дозволяє зробити однозначний висновок про досягнення мети роботи.

У першому розділі проведено глибокий аналіз архітектури 5G NR, характеристик каналів зв'язку та ролі завадостійкого кодування в забезпеченні якості обслуговування (QoS). Окремо розглянуто специфіку застосування LDPC-кодів у низхідному (downlink) та Polar-кодів у висхідному (uplink) каналах згідно зі стандартом 3GPP Release 15.

Другий розділ детально описує алгоритми класичного декодування LDPC та Polar кодів, включаючи BP, Min-Sum, Normalized Min-Sum (NMS), Offset Min-Sum (OMS), а також Successive Cancellation (SC), SC List та Fast-SC. Проведено порівняння обчислювальної складності, затримки, точності та гнучкості, що дозволяє виявити основні обмеження традиційних рішень, зокрема — втрату ефективності при зашумленні каналу або зміні його статистики.

У третьому розділі подано обґрунтування доцільності використання нейромережових підходів, у тому числі Deep Neural Networks (DNN), Graph Neural Networks (GNN), та зосереджено увагу на Deep Reinforcement Learning (DRL) як найбільш перспективному методі для декодування в динамічних та завадових умовах. Розроблено архітектуру DRL+Soft-декодера, яка поєднує гнучкість навчання з точністю м'якого декодування.

Найбільш значущим є четвертий розділ, який містить ґрунтовні експериментальні дослідження. Створено повноцінну модель

телекомунікаційного каналу з підтримкою AWGN, Rayleigh fading, імпульсного та фазового шуму. Виконано моделювання системи з DRL-декодером для LDPC та Polar кодів у MATLAB/Simulink. Результати симуляцій показали істотні переваги інтелектуального підходу, а саме зниження BER на 1.3–2.0 дБ у залежності від типу шуму при однаковому рівні  $E_b/N_0$ ; стабільну роботу в умовах імпульсних та фазових збурень, де класичні декодери демонструють чергу помилок; зменшення затримки (Latency) у DRL+Soft-декодерах на 30–40% у порівнянні з класичними BP/SCL-декодерами при довжині коду до 2048 біт; адаптивність до зміни параметрів каналу без потреби перебудови структури декодера.

Важливо, що результати дослідження мають високу практичну цінність: запропонована методика може бути інтегрована у проектування SDR-рішень, SoC-приймачів та мережевих інтелектуальних вузлів для 5G/6G. Отримані графіки BER, BLER, Latency дозволяють обґрунтовано оцінювати компроміс між точністю, швидкодією та енерговитратами.

Значущим аспектом роботи є комплексність підходу, коли студентка гр. ЕКР<sub>М</sub>-24-1 Дар'ї Суберляк не лише реалізувала інтелектуальний декодер, але й провела порівняльний аналіз з класичними методами, вивчила вплив типів шуму, оцінила вплив довжини коду на латентність та запропонувала варіанти оптимізації. Особливу увагу приділено сценаріям для подальшого дослідження в контексті 6G — зокрема, застосування DRL-декодування у масивних MIMO-системах, в адаптивних low-power IoT-модемах, або у середовищах із мобільними інтерференціями (наприклад, V2X або HAPS).

Магістерська кваліфікаційна робота є завершеним науково-прикладним дослідженням, яке вирізняється новизною, практичною спрямованістю, ґрунтовністю експериментальної частини та високим рівнем самостійності виконання. В роботі вперше реалізовано DRL+Soft-декодування для LDPC та Polar кодів із детальним порівнянням продуктивності у зашумлених каналах.

Робота заслуговує на оцінку «відмінно», а її авторка Дар'я Суберляк — на присвоєння освітнього ступеня магістра за спеціальністю «Електронні комунікації та радіотехніка».

Доцент кафедри фізики,  
математики та інформатики ХНУ, к.т.н.

Олександр ЄРЬОМЕНКО

## Anti-Plagiarism v-15.258 (global version)

The maximum coincidence with one document 0.0%

Dictionaries check: en\_US, ru\_RU, ua\_UA. Errors in the documents: 13%

ID: 250155 Title: Метод інтелектуального декодування зашкоджених кодів у SG16G Added in a DB: 2025-11-22 Authors: Сиберак Дар'я Олександрівна Heads: Бірюк Юлія Миколаївна Consultants: Opponents:	Document		Sum coincidence on the DB	
	Symbols	Lexemes	Symbols	Lexemes
	222569	3454	3468 (2%)	56 (2%)

### Plagiarism sources

ID	Description	Plagiarism presence in the document	
		Symbols	Lexemes

## Протокол аналізу звіту подібності науковим керівником

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Дар'я Суберляк ЕКРМ-24-1

Співавтор:

Назва: Метод інтелектуального декодування завадостійких кодів у 5G/6G

Науковий керівник: Юлій БОЙКО, д.т.н, проф.

Підрозділ: Кафедра телекомунікацій, медійних та інтелектуальних технологій

Коефіцієнт подібності 1:1.5%

Коефіцієнт подібності 2:0.2%

Мікропробіли: 0

Заміна букв: 81

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2025-11-24 18:37:28.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування: Після аналізу повного звіту подібності встановлено, що виявлені схожістю збіги мають тематичний характер, стосуються слів-термінів і стандартних формулювань. Рівень подібності ( $K1=1.5\%$ ,  $K2=0.2\%$ ) є значно нижчим за допустимі межі. Робота є самостійною та може бути прийнята без зауважень.

Дата 25.11.2025

експерт

Олег ПИРОВА Р

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ КАФЕДРИ

*Т.М.І.І.*

ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Дешкодінг текстових документів з використанням алгоритму Viterbi  
 Автор Дар'я Суборлак  
 Освітня програма Електронні інформаційно-комунікаційні системи та мережі  
 Рівень вищої освіти Друкарство (магістерський)  
 Спеціальність 122 Електронні комунікації та інформаційні системи  
 Науковий керівник: Юлія Бойко

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	<i>відповідність</i>
1.2	Виявлені запозичення не є академічним плагіатом, розмішені 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.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розмішені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

*За результатами перевірки системно встановлено, що виявлені ознаки плагіату є переважно технічного характеру, ознак академічного плагіату не виявлено (п.1.1-1.4). Робота допущена до захисту на умовах прийнятності до захисту.*

Дата 25.11.2025

Завідувач кафедри

*[Signature]*  
Підпис

Михайло С.  
Ім'я, ПРІЗВИЩЕ

Гарант освітньої програми

*[Signature]*  
Підпис

Михайло С.  
Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи

*[Signature]*  
Підпис

Юлія Бойко  
Ім'я, ПРІЗВИЩЕ