

Вінницький національний технічний університет

Факультет інтелектуальних інформаційних технологій та автоматизації

Кафедра автоматизації та інтелектуальних інформаційних технологій

## МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства»**

Виконав: студент 2 курсу, групи ІАКІТР-24м  
спеціальності 174 – Автоматизація,  
комп'ютерно-інтегровані технології та  
роботехніка

(шифр і назва спеціальності)

Андрій КОНЧИНСЬКИЙ

(ПІБ студента)

Керівник: к.т.н., доцент кафедри АІТ

Володимир ГАРМАШ

(науковий ступінь, вчене звання / посада, ПІБ керівника)

« 12 » зрудня 2025 р.

Опонент: д.т.н., проф. каф. КСУ

Марія ЮХИМЧУК

(науковий ступінь, вчене звання / посада, ПІБ опонента)

« 12 » зрудня 2025 р.

Допущено до захисту

Завідувач кафедри АІТ

д.т.н., проф. Олег Бісікало

(науковий ступінь, вчене звання)

« 12 » зрудня 2025 р.

Вінниця ВНТУ – 2025 рік

Вінницький національний технічний університет  
Факультет інтелектуальних інформаційних технологій та автоматизації  
Кафедра автоматизації та інтелектуальних інформаційних технологій  
Рівень вищої освіти II-ий (магістерський)  
Галузь знань – 17 – Електроніка, автоматизація та електронні комунікації  
Спеціальність – 174 - Автоматизація, комп'ютерно-інтегровані технології та  
робототехніка  
Освітньо-професійна програма – Інтелектуальні комп'ютерні системи

### **ЗАТВЕРДЖУЮ**

Завідувач кафедри АІТ

д.т.н., проф. Олег БІСІКАЛО

«26» вересня 2025 р.

### **ЗАВДАННЯ**

#### **НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Кончинському Андрію Олексійовичу

(ПІБ автора повністю)

1. Тема роботи: Розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства.  
Керівник роботи: к.т.н., доцент каф. АІТ Гармаш В. В.  
Затвердженні наказом ВНТУ від «24» вересня 2025 року № 313.
2. Строк подання роботи студентом: до «12» грудня 2025 року.
3. Вихідні дані до роботи: Формат аудіоданих – wav; тип перетворення – дискретне вейвлет-перетворення; вейвлет функції – вейвлети Добеші; коефіцієнт стиснення (від початкового об'єму) 30 – 15%; якість стиснутого аудіо файлу 128 – 192 kbps.
4. Зміст текстової частини: Вступ; Аналіз сучасних методів стиснення аудіоданих; Алгоритм стиснення сегменту сигналу; Ділення сигналу на сегменти; Програмна реалізація та експериментальне дослідження; Економічний розділ; Висновки; Список використаних джерел.
5. Перелік ілюстративного (або графічного) матеріалу: Вейвлет-перетворення. Схема програми; Стиснення аудіоданих за допомогою вейвлет-перетворення. Схема роботи програми; UML діаграма класів бібліотеки WaveLib; UML

діаграма класів бібліотеки WaveLib; Інтерфейс програми; Результат експериментального дослідження.

6. Консультанти розділів роботи

Розділ змістової частини роботи	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1 – 3	Володимир ГАРМАШ, к.т.н., доцент кафедри АІТ	25.09.2025	07.12.2025
4	Володимир КОЗЛОВСЬКИЙ, к.е.н., проф. каф. ЕПВМ	07.11.2025	07.12.2025

7. Дата видачі завдання: «25» вересня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області	25.09–05.10.2025	вс
2	Розробка математичного та алгоритмічного забезпечення	05.10 – 25.10.2025	вс
3	Розробка програмного забезпечення	25.10 – 10.11.2025	вс
4	Тестування розробленого програмного забезпечення	05.11 – 20.11.2025	вс
5	Підготовка економічної частини	до 01.12.2025	вс
6	Оформлення пояснювальної записки, графічного матеріалу і презентації	20.11 – 03.12.2025	вс
7	Попередній захист роботи	до 03.12.2025	вс
8	Захист роботи	до 19.12.2025	вс

Студент

  
(підпис)

Андрій КОНЧИНСЬКИЙ  
(прізвище та ініціали)

Керівник роботи

  
(підпис)

Володимир ГАРМАШ  
(прізвище та ініціали)

## АНОТАЦІЯ

УДК 004.67

Кончинський А. О. Розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства. Магістерська кваліфікаційна робота зі спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка», освітня програма – Інтелектуальні комп'ютерні системи. Вінниця: ВНТУ, 2025. 124 с.

На укр. мові. Бібліогр.: 33 назва; рис.: 16; табл. 11.

У магістерській кваліфікаційній роботі розроблено модуль стиснення аудіоданих для системи автоматизації відділу безпеки підприємства. В оглядово-аналітичній частині розглянуто сучасні методи стиснення аудіоданих, зокремо особливості їхнього застосування в автоматизованих системах аудіоспостереження, та обґрунтовано доцільність використання методу на основі вейвлет-перетворень. У теоретичній частині запропоновано метод стиснення, який враховує якісні характеристики аудіосигналів, та розроблено відповідний програмний модуль. У практичній частині описано вибір інструментальних засобів реалізації, розроблено та протестовано програмне забезпечення для модуля стиснення. Економічна частина містить оцінку доцільності впровадження, аналіз технічного рівня, а також розрахунок витрат на розробку та інтеграцію модуля у систему автоматизації. Ілюстративна частина представлена матеріалами, що відображають основні етапи реалізації розробки.

Ключові слова: автоматизована система, обробка аудіосигналів, аудіоспостереження, відділ безпеки, модуль стиснення, вейвлет-перетворення, програма.

## ABSTRACT

Konchinsky A. O. Development of an audio data compression module for the automation system of the enterprise security department. Master's qualification work in specialty 174 "Automation, computer-integrated technologies and robotics", educational program - Intelligent computer systems. Vinnytsia: VNTU, 2025. 124 p.

In Ukrainian. Bibliography: 33 titles; fig.: 16; tab. 11.

In the master's qualification work, an audio data compression module for the automation system of the enterprise security department is developed. The review and analytical part considers modern methods of audio data compression, in particular the features of their application in automated audio surveillance systems, and justifies the feasibility of using a method based on wavelet transforms. The theoretical part proposes a compression method that takes into account the qualitative characteristics of audio signals, and develops a corresponding software module. The practical part describes the choice of implementation tools, develops and tests software for the compression module. The economic part contains an assessment of the feasibility of implementation, an analysis of the technical level, as well as a calculation of the costs of developing and integrating the module into the automation system. The illustrative part is presented with materials reflecting the main stages of development implementation.

Keywords: automated system, audio signal processing, audio surveillance, security department, compression module, wavelet transform, program.

## ЗМІСТ

ВСТУП .....	4
1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ СТИСНЕННЯ АУДІОДАНИХ.....	6
1.1. Опис об'єкта автоматизації та загальна характеристика задачі .....	6
1.2. Загальні відомості про вейвлет-перетворення .....	8
1.3. Кратномасштабне представлення функцій .....	20
1.4. Представлення функцій за допомогою вейвлетів .....	27
1.5. Висновки до розділу 1 .....	31
2. АЛГОРИТМ СТИСНЕННЯ СЕГМЕНТУ СИГНАЛУ.....	33
2.1 Дискретне вейвлет-перетворення .....	33
2.2 Дискретне перетворення Фур'є .....	37
2.3 Перетворення та оцінка коефіцієнтів .....	38
2.4 Кодування з прогнозом по частковому збігу .....	40
2.5 Стерео з'єднання .....	40
2.6 Бітові потоки .....	41
2.7 Загальна схема .....	41
2.8 Усунення перешкод на межах сегментів .....	42
2.9 Висновки до розділу 2 .....	44
3. ДІЛЕННЯ СИГНАЛУ НА СЕГМЕНТИ .....	46
3.1 Статична реалізація .....	46
3.2 Адаптивна реалізація .....	46
3.3 Динамічна реалізація .....	46
3.4. Висновки до розділу 3 .....	53
4. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРЕМЕНТАЛЬНЕ	55
ДОСЛІДЖЕННЯ.	
4.1 Обґрунтування вибору мови програмування .....	55
4.2 Програмна реалізація .....	58
4.3 Експериментальне дослідження .....	65

4.4 Висновки до розділу 4 .....	67
5. ЕКОНОМІЧНА ЧАСТИНА .....	70
5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки .....	70
5.2 Розрахунок витрат на проведення науково-дослідної роботи .....	75
5.2.1 Витрати на оплату праці .....	75
5.2.2 Відрахування на соціальні заходи .....	79
5.2.3 Сировина та матеріали .....	79
5.2.4 Розрахунок витрат на комплектуючі .....	81
5.2.5 Спецустаткування для наукових (експериментальних) робіт .....	81
5.2.6 Програмне забезпечення для наукових (експериментальних) робіт ..	81
5.2.7 Амортизація обладнання, програмних засобів та приміщень.....	82
5.2.8 Паливо та енергія для науково-виробничих цілей .....	84
5.2.9 Службові відрядження .....	85
5.2.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації .....	85
5.2.11 Інші витрати .....	86
5.2.12 Накладні (загальновиробничі) витрати .....	86
5.3 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором .....	88
5.4 Висновки до розділу 5 .....	93
ВИСНОВКИ .....	94
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	96
ДОДАТКИ .....	100
Додаток А (обов'язковий). Технічне завдання .....	101
Додаток Б (обов'язковий). Ілюстративна частина .....	106
Додаток В (обов'язковий). Лістинг модуля програми .....	113
Додаток Г (обов'язковий). Протокол перевірки кваліфікаційної роботи .....	123

## ВСТУП

У сучасних умовах, із широким впровадженням електронних технологій у системи автоматизації, зокрема в системах безпеки підприємств, виникає потреба в ефективному зберіганні, передачі та стисненні аудіоданих, отриманих від пристроїв аудіомоніторингу, таких як мікрофони, системи запису чи камери з аудіофункцією. Стиснення аудіоданих є важливим для економії ресурсів зберігання та пришвидшення передачі даних у системах безпеки, де обробка інформації в реальному часі відіграє ключову роль [1-5]. На сьогодні відомі формати стиснення аудіо, такі як MP3, Ogg Vorbis, WMA та RealAudio. Хоча MP3 залишається популярним через раннє впровадження, тести показують, що Ogg Vorbis є конкурентоспроможним завдяки кращій якості стиснення [7,8].

MP3 та Ogg Vorbis використовують сегментацію сигналу, дискретне перетворення Фур'є, психоакустичні моделі та алгоритми кодування, але їх реалізація відрізняється. У перспективі розробники Ogg Vorbis планують застосовувати вейвлет-технології для стиснення аудіоданих, що відкриває нові можливості для оптимізації обробки сигналів. Це стало основою для ідеї розробки модуля стиснення аудіоданих для систем автоматизації безпеки підприємств, який базується на вейвлет-перетвореннях. Такі методи дозволяють застосовувати трешолдінг (відкидання коефіцієнтів, близьких до нуля), округлення коефіцієнтів, вибір оптимального типу вейвлет-перетворення та сегментації сигналу для зменшення обсягу даних [9-13].

**Метою** є розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства з використанням вейвлет-перетворень для підвищення ефективності стиснення при збереженні необхідної якості звуку.

**Об'єктом дослідження** є процес стиснення аудіоданих у системах автоматизації безпеки підприємств.

**Предметом дослідження** є методи стиснення аудіоданих на основі вейвлет-перетворень для використання в системах безпеки.

Задачі дослідження можна сформулювати наступним чином:

1. Проаналізувати існуючі методи стиснення аудіоданих, застосовні до систем безпеки.
2. Дослідити потенціал вейвлет-перетворень для стиснення аудіоданих у контексті автоматизації безпеки.
3. Розробити модуль стиснення аудіоданих на основі вейвлет-перетворень
4. Зробити практичну реалізацію та провести експериментальні дослідження розробленого методу.

**Науково-технічний результат.** Запропоновано модуль стиснення аудіоданих, який забезпечує підвищення коефіцієнта стиснення на 20-25% від початкового обсягу даних при допустимій втраті якості, завдяки використанню дискретного вейвлет-перетворення.

**Практична цінність.** Розроблений модуль стиснення аудіоданих інтегрується в системи автоматизації безпеки підприємств, дозволяючи створювати компактні архіви аудіоінформації та оптимізувати передачу даних. Програмне забезпечення, створене на основі цього модуля, може бути використано для обробки аудіоданих у системах моніторингу та аналізу безпеки.

**Апробація.** Представлені в роботі результати апробовані в результаті участі в конференції ВНТУ: «Науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації (2025)».

**Публікації.** Гармаш В., Кончинський А. Застосування дискретного вейвлет-перетворення для стиснення аудіоданих у системах обробки аудіоінформації // Матеріали LIV науково-технічної конференції підрозділів ВНТУ, Вінниця, 24-27 березня 2025 р. Електрон. текст. дані. 2025. URI: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2025/paper/view/23822/19757>

## АНАЛІЗ СУЧАСНИХ МЕТОДІВ СТИСНЕННЯ АУДІОДАНИХ

### 1.1 Опис об'єкта автоматизації та загальна характеристика задачі

Об'єктом автоматизації у даній роботі є відділ безпеки підприємства, який відповідає за забезпечення захисту майна, персоналу, інформації та інфраструктури підприємства від внутрішніх і зовнішніх загроз. Відділ безпеки виконує широкий спектр функцій, включаючи моніторинг, аналіз і реагування на потенційні ризики, такі як несанкціонований доступ, крадіжки, витоки інформації, а також надзвичайні ситуації. У сучасних умовах ефективна робота відділу безпеки значною мірою залежить від автоматизованих систем, які дозволяють обробляти великі обсяги даних, зокрема аудіо- та відеоінформацію, отриману від систем спостереження, мікрофонів, датчиків та інших пристроїв.

Відділ безпеки підприємства, як об'єкт автоматизації, характеризується наступними особливостями:

- Високий обсяг даних: Системи безпеки генерують значні обсяги аудіо- та відеоданих, які необхідно обробляти, аналізувати та зберігати для подальшого використання, наприклад, у розслідуваннях чи аудиті.
- Вимоги до швидкості обробки: Для оперативного реагування на загрози необхідно забезпечити швидку обробку та передачу даних, особливо в реальному часі.
- Обмеження ресурсів: Зберігання великих обсягів даних вимагає значних ресурсів, що створює потребу в ефективних методах стиснення інформації без втрати її ключових характеристик.
- Високі вимоги до якості даних: Аудіодані, отримані від систем безпеки (наприклад, записи розмов чи звуки тривоги), повинні зберігати достатню якість для аналізу, ідентифікації та використання в якості доказової бази.
- Інтеграція з іншими системами: Відділ безпеки взаємодіє з іншими підрозділами підприємства, такими як ІТ-відділ, логістика чи адміністрація, що

вимагає сумісності автоматизованих систем з різними форматами даних та протоколами.

Основними джерелами аудіоданих у діяльності відділу безпеки є системи аудіомоніторингу, які фіксують звуки в зонах спостереження, а також записи з пристроїв, що використовуються для документування подій чи комунікацій. Ці дані потребують ефективного стиснення для економії місця на носіях інформації та оптимізації передачі через мережі підприємства.

Задача автоматизації відділу безпеки підприємства полягає в розробці модуля стиснення аудіоданих, який інтегрується в систему автоматизації безпеки для підвищення ефективності обробки, зберігання та передачі звукової інформації. Основна мета полягає в зменшенні обсягу аудіоданих без суттєвої втрати якості, що є критично важливим для збереження їхньої інформативності в контексті безпеки.

Основні аспекти задачі:

- Стиснення аудіоданих: Розробка методу стиснення, який базується на використанні дискретного вейвлет-перетворення, для зменшення обсягу даних, що генеруються системами аудіомоніторингу. Це дозволить економити ресурси зберігання та пришвидшувати передачу даних у мережі підприємства.

- Збереження якості звуку: Забезпечення такого рівня стиснення, при якому аудіодані залишаються придатними для аналізу, ідентифікації джерел звуку чи використання як доказової бази.

- Інтеграція з системами безпеки: Модуль стиснення має бути сумісним із програмно-апаратними комплексами відділу безпеки, включаючи системи моніторингу, бази даних та інтерфейси для аналізу.

- Оптимізація продуктивності: Розробка алгоритмів, які дозволяють обробляти аудіодані в реальному часі або з мінімальними затримками, що є важливим для оперативного реагування на події.

- Гнучкість та масштабованість: Модуль повинен підтримувати різні формати аудіоданих та бути адаптивним до змін у вимогах системи безпеки підприємства.

Технічні вимоги до вирішення задачі:

- Методи обробки сигналів: Використання вейвлет-перетворень для аналізу та стиснення аудіосигналів, що дозволяє ефективно виділяти значущі компоненти сигналу та відкидати менш суттєві (наприклад, за допомогою трешолдінгу).

- Програмна реалізація: Розробка програмного забезпечення, яке реалізує алгоритми стиснення та інтегрується в системи автоматизації безпеки.

- Експериментальна перевірка: Проведення тестування розробленого модуля для оцінки коефіцієнта стиснення, якості відновлених даних та продуктивності в реальних умовах роботи відділу безпеки.

Розроблений модуль стиснення аудіоданих дозволить досягти коефіцієнта стиснення на рівні 20-25% від початкового обсягу даних при збереженні якості, достатньої для потреб безпеки. Це сприятиме зниженню навантаження на системи зберігання та передачі даних, а також підвищить ефективність роботи відділу безпеки підприємства. Практична цінність полягає в створенні універсального програмного рішення, яке може бути адаптоване до різних типів підприємств та їхніх систем безпеки.

## **1.2 Загальні відомості про вейвлет-перетворення**

Вейвлет-перетворення є одним з найпотужніших інструментів сучасної обробки сигналів і зображень. На відміну від класичного перетворення Фур'є, яке розкладає сигнал лише за частотою і повністю втрачає інформацію про те, коли саме з'являються ті чи інші частотні складові, вейвлет-перетворення одночасно аналізує сигнал як у частотній, так і в часовій області. Саме ця властивість робить його незамінним для роботи з нестационарними сигналами, до яких належить переважна більшість реальних аудіозаписів, мовлення, музики, біомедичних сигналів тощо.

Основна ідея вейвлет-перетворення полягає в тому, що замість нескінченних гармонічних синусоїд (як у Фур'є) використовується сімейство коротких функцій обмеженої тривалості, які називаються вейвлетами. Кожен вейвлет може бути зміщений у часі та розтягнутий або стиснутий. Завдяки цьому на низьких частотах (коли вейвлет сильно розтягнутий) досягається висока частотна роздільність і низька часова, а на високих частотах – навпаки: висока часова роздільність при нижчій частотній. Така адаптивність ідеально відповідає природі більшості реальних сигналів, де низькочастотні складові зазвичай змінюються повільно, а високочастотні (наприклад, удари барабана, змички скрипки чи приставні приголосні в мові) виникають миттєво і локально.

Існує два основних типи вейвлет-перетворень: неперервне і дискретне. Неперервне вейвлет-перетворення використовується переважно в теоретичних дослідженнях і для детального аналізу, тоді як у задачах стиснення даних і цифрової обробки найчастіше застосовують дискретне вейвлет-перетворення (ДВП). ДВП реалізовується ефективно за допомогою багаторівневої фільтраційної схеми з парою фільтрів – низькочастотним (фільтр масштабування) і високочастотним (вейвлет-фільтр). На кожному рівні сигнал розбивається на апроксимаційну (низькочастотну) і детальну (високочастотну) складові, після чого апроксимаційна частина обробляється далі. Така ієрархічна структура природно формує багаторівневу декомпозицію, що дуже зручно для стиснення.

Серед переваг вейвлет-перетворення перед Фур'є-аналізом особливо виділяють здатність ефективно концентрувати енергію сигналу в невеликій кількості коефіцієнтів. У гладких ділянках сигналу більшість детальних коефіцієнтів близькі до нуля, що дозволяє застосовувати сильне квантування або просто відкидати їх без помітної втрати якості. Саме ця властивість зробила вейвлети основою стандарту стиснення зображень JPEG-2000 і багатьох сучасних аудіокодеків.

Таким чином, вейвлет-перетворення поєднує в собі найкращі риси часового і частотного аналізу, забезпечує високу енерго-концентрацію і

гнучкість, що робить його одним з головних інструментів цифрової обробки сигналів останнього тридцятиліття.

Вейвлет-перетворення та перетворення Фур'є подібні за деякими аспектами, але мають істотні відмінності. Ми будемо порівнювати їх властивості паралельно, використовуючи аналогії там, де це буде необхідно.

В просторі функцій одного змінного, визначених на кінцевому проміжку  $(0, 2\pi)$  введемо норму як корінь з скалярної похідної

$$\|f(t)\|^2 = (f(t), f(t)) = \int_0^{2\pi} f^*(t)f(t)dt \quad . \quad (1.1)$$

Функції, для яких такий інтеграл сходиться і має кінцеве значення, належать простору  $L^2(0, 2\pi)$ . Зокрема, всі кусочно-безперервні на проміжку  $(0, 2\pi)$  функції належать цьому простору. Будь-яка функція з цього  $f(t)$  простору може бути представлена у вигляді ряду Фур'є

$$f(t) = \sum_{n=-\infty}^{n=+\infty} c_n \exp(int) \quad . \quad (1.2)$$

Коефіцієнти  $c_n$  в цьому розкладанні виражаються у вигляді інтегралів

$$c_n = \frac{1}{2\pi} \int_0^{2\pi} f(t) \exp(-int) dt \quad . \quad (1.3)$$

Ряд Фур'є рівномірно сходиться по нормі, визначеній в (1.1):

$$\lim \left\| f(t) - \sum_{-N}^N c_n \exp(int) \right\| = 0 \quad \text{при } N \rightarrow \infty. \quad (1.4)$$

Відмітимо, що  $w_n(t) = \exp(int)$  є ортонормований базис простору  $L^2(0, 2\pi)$ , побудований з єдиної функції  $w(t) = \exp(it)$  за допомогою масштабного перетворення незалежної змінної (часу), так що  $w_n(t) = w(nt)$

Для коефіцієнтів ряду Фур'є справедливо рівність Парсеваля:

$$\frac{1}{2\pi} \int_0^{2\pi} |f(t)|^2 dt = \sum_{-\infty}^{\infty} |c_n|^2 \quad . \quad (1.5)$$

Аналогічно визначається функціональний простір  $L^2(R)$  на всій числовій прямій. До нього належать функції, визначені на всій числовій осі  $R(+\infty, -\infty)$  і володіючи кінцевою енергією (нормою)

$$E = \int_{-\infty}^{+\infty} |f(t)|^2 dt < \infty \quad . \quad (1.6)$$

Проте, властивості функцій з просторів  $L^2(0, 2\pi)$  і  $L^2(R)$  істотно різні. Зокрема, нескінченні синусоїдальні хвилі не належать простору  $L^2(R)$  і, отже, сімейство синусоїд не є базисом в цьому просторі. Спроба сконструювати базис простору  $L^2(R)$  з функцій, що належать цьому ж простору, привела до створення вейвлет-перетворення.

Функції, що належать простору  $L^2(R)$ , повинні прямувати до нуля на нескінченності. Чим швидше ці функції прямують до нуля, тим практично зручніше використовувати їх як базис перетворення при аналізі реальних сигналів. Нехай  $\phi(t)$  – така функція. Назвемо її базисною функцією. Поки вважатимемо, що вона рівна нулю за межами деякого кінцевого інтервалу [3].

Так само як у разі простору  $L^2(0, 2\pi)$ , спробуємо сконструювати базис в просторі  $L^2(R)$  на основі функції  $\phi(t)$  за допомогою масштабних перетворень незалежної змінної. Нехай коефіцієнти перетворення пропорційні ступеням двійки  $\phi(t) \rightarrow \phi(2^j t)$ .

Для того, щоб охопити всю числову вісь за допомогою фінітних функцій, організуємо систему зсувів вздовж осі. Ці зсуви будуть цілими числами для простоти, тобто вони матимуть вигляд.

$$\phi(t) \rightarrow \phi(2^j t + k)$$

(1.7)

За допомогою визначеної таким чином системи функцій можна перекрити всю дійсну вісь. Якщо базисна функція  $\phi(t) \in L_2(R)$  має одиничну норму, то і всі функції

$$\phi^{jk}(t) = 2^{j/2} \phi(2^j t + k) , \quad (1.8)$$

будуть нормовані на одиницю, тобто

$$\|\phi^{jk}\| = \sqrt{\int_{-\infty}^{+\infty} \phi^{jk} \phi^{jk} dt} = 1 . \quad (1.9)$$

Якщо сімейство функцій  $\phi^{jk}(t)$  ортонормованим базисом простору  $L^2(R)$ , тобто

$$\left( \phi^{jk}(t) \phi^{lm}(t) \right) = \int_{-\infty}^{+\infty} \phi^{jk}(t) \phi^{*lm}(t) dt = \delta_{jl} \delta_{km} . \quad (1.10)$$

і кожна функція може бути представлена у вигляді ряду (розкладання по базису)

$$f(t) = \sum_{j,k=-\infty}^{\infty} c_{jk} \phi^{jk}(t) , \quad (1.11)$$

який рівномірно сходиться в  $L^2(R)$ , тобто

$$\lim \left\| f - \sum_{-M}^M \sum_{-N}^N c_{jk} \phi^{jk} \right\| = 0 \quad \text{при} \quad \lim \left\| f - \sum_{-M}^M \sum_{-N}^N c_{jk} \phi^{jk} \right\| = 0 , \quad (1.12)$$

тоді базисна функція перетворення називається ортогональним вейвлетом. Ортогональність системи функцій  $\phi^{jk}(t)$  може бути перевірена безпосередньо. Докази повноти і замкнутості базису для кожної конкретної системи повинні проводитися окремо. Як правило, вони вельми складні і громіздкі. Посилання на них можна знайти в оглядах [2,4]. Для практичного застосування, проте, достатньо приблизного дотримання всіх цих властивостей. Як правило, на практиці використовуються саме такі системи функцій.

Найпростішим прикладом ортогональної системи функцій такого типу є перетворення Хаара. Базисна функція цього перетворення визначається співвідношенням

$$\phi^H = \begin{cases} 1, & 0 < t < 1/2 ; \\ -1, & 1/2 < t < 1; \\ 0, & t < 0, t > 1. \end{cases} \quad (1.13)$$

Легко перевірити, що дві будь-які функції, отримані за допомогою цього базисного вейвлета шляхом масштабних перетворень і перенесень, мають одиничну норму і ортогональні. Виявляється також, що ця система функцій замкнута і повна в просторі  $L^2(R)$ , тобто є базисом. Доказ повноти і замкнутості базису перетворення Хаара можна знайти в посиланні [17].

Для функцій, заданих на всій числовій осі ряд Фур'є замінюється інтегралом Фур'є по безперервному спектру частот. Аналогічно визначимо континуальний базис вейвлетів в просторі  $L^2(R)$  на основі системи безперервних перетворень масштабу і перенесень:

$$\phi^{ab}(t) = \frac{1}{\sqrt{|a|}} \phi\left(\frac{t-b}{a}\right). \quad (1.14)$$

При цьому параметри перенесення  $b$  і масштабного перетворення  $a$  приймають довільні значення з безперервного спектру. Формула інтегрального вейвлет-перетворення на його основі записується у вигляді

$$[W_{\phi}f](a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{+\infty} f(t) \phi^*\left(\frac{t-b}{a}\right) dt = \int_{-\infty}^{+\infty} f(t) \phi^{*ab}(t) dt \quad . \quad (1.15)$$

За допомогою цього співвідношення можна формально виразити коефіцієнти дискретного вейвлет-перетворення:

$$c_{jk} = [W_{\phi}f]\left(\frac{1}{2^j}, \frac{k}{2^j}\right) \quad . \quad (1.16)$$

Розглянемо тепер питання про зворотне перетворення, що дозволяє по відомих коефіцієнтах перетворення відновити початкову функцію з простору  $L^2(R)$ . У разі Фур'є-перетворення ми маємо справу з ортонормованою системою базисних функцій, і проблем із зворотним перетворенням Фур'є не виникає. У разі вейвлет-перетворення ортонормованість базису і існування зворотного перетворення вимагає окремого доказу. Їх можна знайти в посиланні [18].

Зворотне перетворення для безперервного вейвлет-перетворення (1.14) записується в тому ж вигляді, що і пряме

$$f(t) = \frac{1}{C_{\phi}} \iint W(a,b) \phi_{ab}(t) \frac{dadb}{a^2} \quad . \quad (1.17)$$

Коефіцієнт, що входить в цю формулу, є константою нормування, аналогічно коефіцієнту, нормалізуючому перетворення Фур'є:

$$C_\phi = \int_{-\infty}^{+\infty} |\hat{\phi}(\omega)|^2 \frac{d\omega}{|\omega|^2} < \infty, \quad (1.18)$$

де  $\hat{\phi}(\omega)$  – Фур'є-трансформанта базисної функції вейвлет-перетворення.

Умова скінченності нормалізуючого коефіцієнта накладає обмеження на клас базисних функцій вейвлет-перетворення  $\phi(t) \in L^2(R)$ .

Зокрема, для збіжності інтеграла (1.15) в нулі потрібно, принаймні, щоб  $\hat{\phi}(0) = 0$ , тобто

$$\int_{-\infty}^{+\infty} \phi(t) dt = 0. \quad (1.19)$$

Для дискретного вейвлет-перетворення існування зворотного перетворення визначається за допомогою нерівності Рісса [21]. Нехай  $\phi^{ik}$  - базис вейвлет-перетворення.

$$A \left\| \{c_{jk}\} \right\|_2^2 \leq \left\| \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} c_{jk} \phi_{jk} \right\|_2^2 \leq B \left\| \{c_{jk}\} \right\|_2^2, \quad (1.20)$$

справедлива для будь-якої обмеженої, підсумовуваної з квадратом послідовності  $\{c_{jk}\}$ :

$$\left\| \{c_{jk}\} \right\|_2^2 = \sum_{j=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} |c_{jk}|^2 < \infty, \quad (1.21)$$

така базисна функція називається R-функцією. Для будь-яких двох елементів цих базисів виконується властивість ортогональності

$$\{ \phi_{ik} \} (\phi^{jk}, \phi_{lm}) = \delta_{jl} \delta_{km}. \quad (1.22)$$

На основі цього базису будується формула реконструкції:

$$f(t) = \sum_{j,k=-\infty}^{+\infty} (f, \phi_{jk}) \phi^{jk}(t) . \quad (1.23)$$

Якщо  $\phi^{jk}(t)$  – ортонормований базис і  $\phi(t)$  – ортогональний вейвлет, тоді базиси  $\{\phi_{ik}\}$  і  $\{\phi^{ik}\}$  співпадають, і формула (1.23) є точною формулою зворотного перетворення. Якщо  $\phi(t)$  – не ортогональний вейвлет, але є R-функцією, тоді базис-двійник також має базисну функцію, на основі якої він будується по виразу (1.23). В загальному випадку формула реконструкції (1.23) не є вейвлет-рядом в тому значенні, що базис-двійник не може бути побудований по типу рівняння (1.7).

Вейвлет-функції мають цілий ряд привабливих особливостей як засіб аналізу фізичних процесів. Наприклад, у порівнянні з перетворенням Фур'є, яке є одним з основних інструментів такого аналізу, вейвлет-функції володіють властивістю локалізації в часі. Перетворення Фур'є вимагає представлення сигналу у вигляді синусоїд та косинусоїд різних частот, що не завжди є практично зручним для аналізу фізичних явищ.

Дельта-функція Дірака, яка локалізована в часі, має важливе обмеження - вона не містить інформації про частоту процесу. Це призводить до виявлення закону типу квантового співвідношення невизначеності, коли неможливо одночасно точно визначити частоту процесу та його значення в певний момент часу.

У вейвлет-аналізі використовуються різні типи вейвлет-функцій, що дозволяє досліднику гнучко вибирати між різними співвідношеннями невизначеності. Це надає змогу отримувати більш глибокий та комплексний аналіз фізичних процесів. У той час як перетворення Фур'є та його модифікації не мають таких властивостей гнучкості та адаптивності.

Тепер спробуємо більш строго сформулювати властивості вейвлет-перетворення і обмеження, що накладаються на його базисні функції. Оскільки в наш час повна теорія вейвлетів ще не побудована, розглянемо одне з найпростіших визначень вейвлета, що спирається на вже введені вище поняття.

Будь-яка локалізована  $R$ -функція  $\phi(t) \in L_2(R)$  називається  $R$ -вейвлетом (або просто вейвлетом), якщо для неї існує функція-двійник такий,  $\phi^*(t) \in L_2(R)$  що базиси  $\{\phi_{ik}\}$  і  $\{\phi_{ik}^*\}$  побудовані згідно формулі (1.23), є парними взаємними базисами функціонального простору  $L_2(R)$ . Кожний визначений таким чином вейвлет, незалежно від того, чи є він ортогональним чи ні, дозволяє представити будь-яку функцію  $f(t)$  з простору  $L_2(R)$  у вигляді ряду (1.23), коефіцієнти якого визначаються інтегральним перетворенням.

Будь-який  $R$ -вейвлет називається неортогональним, якщо він не є напівортогональним вейвлетом. Проте, оскільки він задовольняє умові Рісса, він має двійника, який дозволяє побудувати формулу зворотного перетворення (1.23).

У багатьох ситуаціях доводиться працювати з дискретними сигналами, тому дуже важливо мати дискретні варіанти вейвлет-перетворень. На жаль, не завжди можна отримати дискретні формули для вейвлет-перетворень простою дискретизацією відповідних формул для безперервного часу. Також неможливо визначити кратномасштабний аналіз для дискретних сигналів, оскільки не існує базисних функцій, масштабовані і зміщені версії яких давали б нам базис простору  $L^2(R)$ , простору квадратично підсумованих послідовностей нескінченної довжини [22].

Виведемо формули для дискретизації з формул кратномасштабного аналізу.

Нехай є деяка безперервна функція  $f_o(x) \in V_o$ . Наш дискретний сигнал  $c_n$  представимо як послідовність коефіцієнтів при масштабуючих функціях, по яких розкладається  $f_o(x)$ :

$$f_0(x) = \sum_n c_{0,n} \delta_{0,n}(x), \quad (1.24)$$

де  $c_{0,n} = c_n$ .

Іншими словами, наш сигнал інтерпретується як послідовність коефіцієнтів розкладання, одержана в ході кратномасштабного аналізу функції  $f_0(x)$ . Тоді можна обчислити апроксимації цієї функції, що належать просторам  $V_1, V_2, \dots$ . Простори  $V_{-1}, V_{-2}, \dots$  не мають значення при даній інтерпретації.

Згідно концепції кратномасштабного аналізу функція  $f_0(x)$  розкладається на дві функції  $f_1(x) \in V_1$ , і  $e_1(x) \in W_1$ :

$$f_0(x) = f_1(x) + e_1(x) = \sum_k c_{1,k} \delta_{1,k}(x) + \sum_k d_{1,k} \psi_{1,k}(x). \quad (1.28)$$

Таким чином, одержали дві нові послідовності  $c_{1,n}$  і  $d_{1,n}$ . Цей процес може бути продовжений по  $f_1(x)$ , і функція  $f_0(x)$  (а також і послідовність  $c_n$ ) буде представлена сукупністю коефіцієнтів  $d_{m,n}, m \in Z^+, n \in Z$  [23].

Отже, концепція дискретного розкладання визначена. Проте обчислення поки залежать від безперервних функцій  $\phi(x)$  і  $\psi(x)$ . Тому покажемо, як обчислення дискретного розкладання можуть бути виконані з використанням операцій тільки над дискретними сигналами.

З урахуванням того, що масштабуюча функція утворює базис відповідного простору, можна одержати

$$\begin{aligned} c_{1,k} &= \langle \phi_{1,k}(x), f_1(x) \rangle = \langle \phi_{1,k}, f_0(x) - e_1(x) \rangle = \left\langle \phi_{1,k}(x), \sum_n c_{0,n} \phi_{0,n}(x) \right\rangle = \\ &= \sum_{n \in Z} c_{0,n} \langle \phi_{1,k}(x), \phi_{0,n}(x) \rangle = 2^{1/2} \sum_{n \in Z} c_{0,n} h_{n+2k}. \end{aligned} \quad (1.26)$$

Отже, виявляється можливим ітеративне обчислення коефіцієнтів  $c_{j,k}$  і  $d_{j,k}$  без безпосереднього використання функцій  $\phi(x)$  і  $\psi(x)$ . По аналогії з (2.25) можна записати для довільного  $j$

$$c_{j,k} = 2^{1/2} \sum_n c_{j-1,n} h_{n+2k}, \quad (1.27)$$

$$d_{j,k} = 2^{1/2} \sum_n c_{j-1,n} g_{n+2k}, \quad (1.28)$$

одержавши, таким чином, повністю дискретний процес декомпозиції. Послідовності  $h_n$  і  $g_n$  називаються фільтрами. Відзначимо, що  $c_{j,k}$  і  $d_{j,k}$  мають «половинну» довжину в порівнянні з  $c_{j-1,k}$  (хоча, звичайно, на даному етапі всі послідовності нескінченні). Таким чином, не вводяться надмірності.

Зворотний процес полягає в отриманні  $c_{j-1}$  з  $c_j$  і  $d_j$ :

$$\begin{aligned} c_{j-1,n} &= \langle \phi_{j-1,n}, f_{j-1}(x) \rangle = \langle \phi_{j-1,n}(x), f(x) + e_j(x) \rangle = \\ &= \left\langle \phi_{j-1,n}, \sum_k c_{j,k} \phi_{j,k}(x) \right\rangle + \left\langle \phi_{j-1,n}, \sum_k d_{j,k} \psi_{j,k}(x) \right\rangle = \\ &= \sum_k c_{j,k} \langle \phi_{j,k}, \phi_{j,k}(x) \rangle + \sum_k d_{j,k} \langle \phi_{j,k}, \psi_{j,k}(x) \rangle = \\ &= 2^{1/2} \sum_k c_{j,k} h_{n+2k} + 2^{1/2} \sum_k d_{j,k} g_{n+2k}. \end{aligned} \quad (1.29)$$

Відзначимо, що в даному випадку підсумовування виробляється по інших змінних в порівнянні з формулами (1.30) і (1.31). Довжина послідовності  $c_{j-1}$  удвічі більше довжини послідовності  $c_j$  або  $d_j$ .

Підставляючи (1.27) і (1.28) в (1.29), одержуємо наступні обмеження на фільтри  $h_n$  і  $g_n$ :

$$2\sum_k (h_{n+2k} h_{p+2k} + g_{n+2k} g_{p+2k}) = \delta_{n,p}, \quad (1.30)$$

$$2\sum_n h_{n+2k} h_{n+2p} = 2\sum_n g_{n+2k} g_{n+2p} = \delta_{k,p}, \quad (1.31)$$

$$2\sum_n h_{n+2k} h_{n+2p} = 0. \quad (1.32)$$

Вираз (1.29) для часової області еквівалентний виразам для частотної [10].

### 1.3 Кратномасштабне представлення функцій

Кратномасштабний аналіз є одним із найпотужніших математичних інструментів сучасної обробки сигналів і виник як узагальнення класичного гармонійного аналізу Фур'є. На відміну від Фур'є-перетворення, що забезпечує лише частотну локалізацію і повністю втрачає інформацію про час появи тієї чи іншої частотної складової, кратномасштабне представлення дозволяє одночасно контролювати роздільну здатність як у часовій, так і у частотній області.

Основна ідея полягає в декомпозиції сигналу на вкладення вкладених просторів, кожен з яких описує деталі функції на певному масштабі. Грубо кажучи, сигнал послідовно розкладається на низькочастотну (апроксимаційну) складову, що відображає загальний хід функції, та високочастотну (детальну) складову, яка несе інформацію про швидкі локальні зміни. Така процедура може повторюватися рекурсивно лише для апроксимаційної частини, що дає ієрархічну структуру представлену у вигляді бінарного дерева масштабів.

Найпоширенішим і теоретично найбільш обґрунтованим інструментом кратномасштабного аналізу є вейвлет-перетворення. Воно базується на парі

взаємодоповняльних функцій – масштабувальній функції (father wavelet) та власне вейвлет-функції (mother wavelet). Масштабувальна функція породжує наближення сигналу на грубих масштабах, тоді як вейвлет-функція відповідає за виділення деталей при переході між сусідніми рівнями розкладання. Завдяки властивості компактного носія більшості вейвлетів досягається добра локалізація в часовій області, що особливо важливо для сигналів із різко вираженими нестационарними ділянками.

Для стиснення аудіо кратномасштабне представлення виявилось ефективнішим за класичне Фур'є-перетворення саме через здатність концентрувати енергію сигналу в обмеженій кількості коефіцієнтів. У музичних і мовних сигналах значна частина енергії зосереджена в низькочастотних апроксимаціях і лише невелика кількість детальних коефіцієнтів описує різкі переходи (атаки ударних інструментів, приголосні звуки тощо). Це дає можливість застосовувати нелінійне квантування та порогову обробку з мінімальними суб'єктивно помітними спотвореннями.

Отже, кратномасштабний підхід не лише розширює можливості аналізу нестационарних сигналів, а й створює природну основу для побудови ефективних алгоритмів стиснення з контрольованими втратами, що зберігають високу якість сприйняття при значному зменшенні об'єму даних.

Під кратномасштабним аналізом розуміється опис простору  $L^2(\mathbb{R})$  через ієрархічні вкладені підпростори  $V_m$ , які не перетинаються і об'єднання яких дає нам в межі  $L^2(\mathbb{R})$ , тобто

$$\left. \begin{aligned} \dots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \dots \\ \bigcap_{m \in \mathbb{Z}} V_m = \{0\}, \quad \bigcup_{m \in \mathbb{Z}} \overline{V_m} = L^2(\mathbb{R}) \end{aligned} \right\} \quad (1.33)$$

Далі, ці простори мають наступну властивість: для будь-якої функції  $f(x) \in V_m$  її стисла версія належатиме простору  $V_{m-1}$ ,

$$f(x) \in V_m \Leftrightarrow f(2x) \in V_{m-1}. \quad (1.34)$$

I, нарешті, остання властивість кратномасштабного аналізу: існує така функція  $\phi(x) \in V_0$ , що її зсуви  $\phi_{0,n}(x) = \phi(x-n)$ ,  $n \in \mathbb{Z}$  утворюють ортонормований базис простору  $V_0$ . На рисунку 2.1 схематично показані дані вкладені простори.

Оскільки функції  $\phi_{0,n}(x)$  утворюють ортонормований базис простору  $V_0$ , то функції

$$\phi_{m,n}(x) = 2^{-m/2} \phi(2^{-m}x - n). \quad (1.35)$$

утворюють ортонормований базис простору  $V_m$ . Ці базисні функції називаються масштабуючими, оскільки вони створюють масштабовані версії функцій в  $L^2(\mathbb{R})$ . З кратномасштабного аналізу, визначеного вище, витікає, що функція  $f(x)$  у  $L^2(\mathbb{R})$  може бути представлена безліччю послідовних її наближень  $f_m(x)$  у  $V_m$ . Іншими словами, функція  $f(x)$  є межа апроксимацій  $f_m(x) \in V_m$  при  $m$  прагнучому до мінус нескінченності:

$$f(x) = \lim_{m \rightarrow -\infty} f_m(x). \quad (1.36)$$

Звідси з'являється можливість аналізу функції або сигналу на різних рівнях дозволу, або масштабу. Змінна  $m$  називається масштабним коефіцієнтом, або рівнем аналізу. Якщо значення  $m$  велике, то функція в  $V_m$  є груба апроксимація  $f(x)$ , і деталі відсутні. При малих значеннях  $m$  має місце точна апроксимація. З визначення кратномасштабного аналізу виходить, що всі функції в  $V_m$  можуть бути представлені як лінійна комбінація масштабуючих функцій. Насправді,  $f_m(x)$  є ортогональна проекція  $f(x)$  на  $V_m$ ,

$$(1.37)$$

$$f_m(x) = \sum_n \langle \phi_{m,n}(x), f(x) \rangle \phi_{m,n}(x) = \sum_n c_{m,n} \phi_{m,n}(x).$$

Оскільки  $\phi(x) = \phi_{0,0}(x) \in V_0 \subset V_{-1}$ , можна записати:

$$\phi_{0,0}(x) = 2^{1/2} \sum_n h_n \phi_{-1,n}(x) = 2 \sum_n h_n \phi(2x - n), \quad (1.38)$$

де  $h_n$  - деяка послідовність.

Рівність (1.38) є однією з основних в теорії вейвлет-аналізу і має різні назви в літературі. Ми називатимемо його далі масштабуючим рівнянням.

На рисунку 1.1 наведено кратномасштабне представлення  $L^2(\mathbb{R})$ .

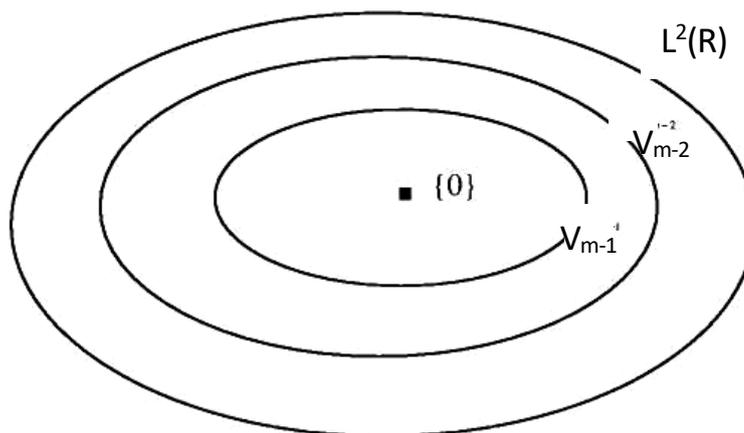


Рисунок 1.1 – Кратномасштабне представлення  $L^2(\mathbb{R})$

Функція  $\phi(x)$  і послідовність  $h_n$  тісно зв'язані між собою. Виведемо відповідні відносини. З (2.6) можна одержати

$$\phi_{m+1,k}(x) = 2^{1/2} \sum_n h_p \phi_{m,p-2k}(x) = 2^{(-m+1)/2} \sum_n h_p \phi(2^{-m}x - (p-2k)) \quad (1.39)$$

Виконаємо операцію скалярного добутку  $\phi_{r,n-2k}(x)$  з обох боків рівності (1.39):

$$\begin{aligned}
\langle \phi_{m+1,k}(x), \phi_{m,n-2k}(x) \rangle &= \left\langle 2^{1/2} \sum_p h_p \phi_{m,p-2k}(x), \phi_{m,n-2k}(x) \right\rangle = \\
&= 2^{1/2} \sum_p h_p \langle \phi_{m,p-2k}(x), \phi_{m,n-2k}(x) \rangle = 2^{1/2} h_n
\end{aligned} \tag{1.40}$$

Відзначимо, що ця рівність виконується для будь-якого  $m$ . Далі, якщо переписати (1.38) в частотній області, можна одержати:

$$\Phi(\omega) = H\left(\frac{\omega}{2}\right) \Phi\left(\frac{\omega}{2}\right). \tag{1.41}$$

При рекурсивному повторенні формули (1.41) виходить вираз:

$$\Phi(\omega) = \prod_{m=1}^{\infty} H\left(\frac{\omega}{2^m}\right). \tag{1.42}$$

Отже, послідовність  $h_n$  тісно пов'язана з масштабуючою функцією. Крім того, з концепції кратномасштабного аналізу витікають наступні властивості.

По-перше, інтегруючи (1.38) по всій числовій осі  $x$ , можна одержати

$$\sum_n h_n = 1. \tag{1.43}$$

оскільки для побудови кратномасштабного аналізу середнє значення функції  $\phi(x)$  не повинне бути рівне нулю. По-друге, через ортонормальності базисні функції

$$\delta_{0,k} = \langle \phi_{0,0}(x), \phi_{0,k}(x) \rangle = 2 \sum_n h_n h_{n+2k}. \tag{1.44}$$

Третю властивість послідовності  $h_n$  сформулюємо в спектральній області.  
Із запису умови ортонормальності функцій  $\phi_{\tau,\pi}\{x\}$  у області спектру

$$\sum_k |\Phi(\omega + 2k\pi)|^2 = 1, \quad (1.45)$$

можна одержати наступний вираз:

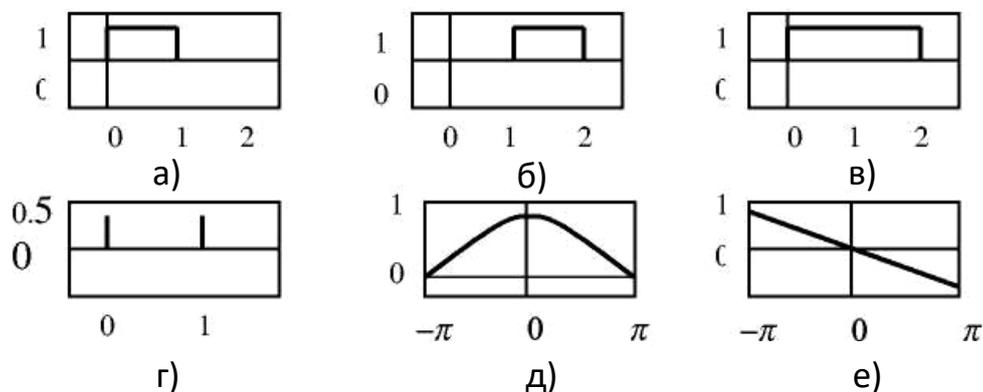
$$|H(\omega)|^2 + |H(\omega\pi)|^2 = 1. \quad (1.46)$$

Рівність (1.43) еквівалентна тому, що  $H(0)=1$ . Тоді з (1.46) витікає, що  $H(\pi)=0$ .

Розглянемо безліч зрушень і розтягувань одиничної функції на одиничному інтервалі [4-8]:

$$\phi(x) = \begin{cases} 1, & 0 \leq x < 1, \\ 0, & \text{У решті випадків.} \end{cases} \quad (1.47)$$

Базисна функція і відповідна послідовність зображені на рисунку 1.2.



а)  $\phi(x)$ ; б)  $\phi_{0,1}(x)$ ; в)  $\phi_{1,0}(x)$ ; г)  $h_n$ ; д)  $|H(\omega)|$ ; е)  $\arg(H(\omega))$

Рисунок 1.2 - Приклад масштабуючої функції

Так, базисні функції з коефіцієнтом масштабу -1 мають вигляд

$$\phi_{-1,n}(x) = \begin{cases} \sqrt{2}, & n/2 \leq x < (n+1)/2, \\ 0, & \text{У решті випадків.} \end{cases} \quad (1.48)$$

Кратномасштабне (вейвлетне) представлення має низку суттєвих переваг порівняно з традиційним короткочасним Фур'є-перетворенням саме в задачах перцептуального стиснення звуку:

1. Відсутність блокових артефактів. При використанні фіксованих вікон Фур'є на межах блоків часто виникають характерні «переддзвоніння» (pre-echo), особливо помітні на різких атаках ударних інструментів. Вейвлет-перетворення, завдяки компактному носію базисних функцій і перекриванню на різних масштабах, практично повністю усуває такі артефакти навіть при агресивному квантуванні.
2. Природна адаптація роздільної здатності. На низьких частотах вейвлет забезпечує високу частотну роздільність і низьку часову, а на високих — навпаки. Така властивість ідеально відповідає психоакустичній моделі людського слуху: ми значно точніше локалізуємо час появи високих частот (атаки, транзйєнти), але менш чутливі до їх точної частоти, і навпаки — низькі частоти потребують точного визначення тону, але не критичні до мікросекундних зсувів.
3. Ефективніше концентрування енергії. Для більшості реальних музичних і мовних сигналів вейвлет-коефіцієнти виявляються значно більш розрідженими, ніж коефіцієнти короткочасного Фур'є-перетворення при аналогічній довжині вікна. Це дозволяє при тому ж суб'єктивному якості застосовувати жорсткіше порогове відсікання та отримувати вищий коефіцієнт стиснення.
4. Простота реалізації адаптивного квантування. Ієрархічна структура вейвлет-дерева природним чином групує коефіцієнти за масштабами та

енергетичним внеском. Це дає змогу легко будувати багаторівневі психоакустичні моделі й розподіляти біти нерівномірно — більше бітів низькочастотним апроксимаціям і найбільшим детальним коефіцієнтам, менше або нуль — незначним деталям.

5. Можливість прогресивного та вбудованого кодування. Завдяки деревоподібній організації коефіцієнтів легко реалізувати масштабоване стиснення (scalable coding): спочатку передаються найгрубші апроксимації, потім поступово додаються деталі. Це корисно для потокового мовлення та зберігання з різними рівнями якості в одному файлі.
6. Стійкість до втрат пакетів при передачі. При втраті окремих пакетів у вейвлет-потоках спотворення розподіляються рівномірно по всьому сигналу й сприймаються як легким шумом, тоді як втрата MDCT-блоку в MP3/AAC призводить до чітко локалізованого «випадання» звуку тривалістю 20–40 мс.

Таким чином, кратномасштабне представлення не лише теоретично елегантніше описує нестационарні аудіосигнали, а й на практиці забезпечує кращу якість при однаковому бітрейті або той же суб'єктивний рівень якості при значно меншому бітрейті порівняно з класичними перцептуальними кодеками на основі модифікованого дискретного косинусного перетворення.

#### 1.4 Представлення функцій за допомогою вейвлетів

При аналізі рисунка 1.2 видно, що область  $L^2(\mathbb{R})$  побудована з безлічі «кілець», які є різниця між двома сусідніми просторами. Ці різницеві простори позначаються через  $W_m$  і визначаються як ортогональні доповнення областей  $V_m$  до  $V_{m-1}$ :

$$V_{m-1} = V_m \oplus W_m, \quad \bigcap_{m \in \mathbb{Z}} W_m = \{0\}, \quad \overline{\bigcup_{m \in \mathbb{Z}} W_m} = L^2(\mathbb{R}). \quad (1.48)$$

Нехай  $\psi(x) = \psi_{0,0}(x)$  є базисна функція  $W_0$ . скільки  $\psi_{0,0}(x) \in W_0 \subset V_{-1}$ , можна записати

$$\psi_{m,n}(x) = 2^{1/2} \sum_n g_n \phi_{-1,n}(x). \quad (1.49)$$

для деякої послідовності  $g_n$ . По аналогії з раніше розглянутою безліччю функцій  $\phi_{r,l}(x)$  визначимо сімейство вейвлет-функцій:

$$\psi_{m,n}(x) = 2^{-m/2} \psi(2^{-m}x - n). \quad (1.50)$$

Існують жорсткі залежності між  $\psi(x), \phi(x), g_n, h_n$ . Спочатку одержимо формулу, аналогічну (1.41). Перепишемо (1.49) для частотної області:

$$\psi(\omega) = G\left(\frac{\omega}{2}\right) \Phi\left(\frac{\omega}{2}\right). \quad (1.51)$$

замінімий  $\Phi(\omega)$  нескінченним добутком (1.41) і одержимо

$$\psi(\omega) = G\left(\frac{\omega}{2}\right) \prod_{m=2}^{\infty} H\left(\frac{\omega}{2^m}\right). \quad (1.52)$$

Відзначимо, що  $\Psi(\omega)$  пропорційно нескінченному добутку  $H(2^{-m}\omega)$ , а не  $G(2^{-m}\omega)$ , так само, як і в (1.52), вейвлет  $\psi(x)$  був виражений у вигляді лінійної комбінації масштабуючих функцій.

Тепер одержимо вирази, що зв'язують послідовності  $g_n$  і  $h_n$ .

Оскільки  $W_m$  є ортогональне доповнення  $V_m$ , функції  $\psi_{0,0}(x)$  і  $\phi_{0,0}(x)$  повинні бути ортогональні, і з (1.38) і (1.45) витікає, що

$$0 = \langle \phi_{0,0}, \psi_{0,0} \rangle = 2 \sum_n \sum_p h_n g_p \langle \phi_{-1,n}, \phi_{-1,p} \rangle = 2 \sum_n h_n g_n. \quad (1.53)$$

Легко побачити, що вибір

$$g_n = (-1)^n h_{-n+2t+1}. \quad (1.54)$$

буде коректний для всіх  $t \in \mathbb{Z}$ . Еквівалент (1.54) в частотній області представляється у вигляді

$$G(\omega) = -H(-\omega + \pi) e^{-i\omega(2t+1)}. \quad (1.55)$$

З урахуванням цього з (1.55) одержимо

$$\psi(\omega) = -e^{i\omega/2} H\left(-\frac{\omega}{2} + \omega\right) \Phi\left(\frac{\omega}{2}\right), \quad (1.56)$$

де без втрати спільності вибране  $t = 0$ .

Нарешті відзначимо, що і функція  $\psi(x)$  і послідовність  $g_n$  мають нульове середнє. Цей факт легко перевірити, підставляючи  $\omega = 0$  в (1.56) і (1.55) і використовуючи властивість  $H(\pi) = 0$  :

$$\psi(0) = \int_{-\infty}^{+\infty} \psi(x) dx = 0, \quad (1.57)$$

і

$$G(0) = \sum_n g_n = 0. \quad (1.58)$$

Визначення функцій вейвлетів дозволяє нам записати будь-яку функцію  $f(x) \in L^2(\mathbb{R})$  у вигляді суми проєкцій на  $W_j, j \in \mathbb{R}$ :

$$f(x) = \sum_{j=-\infty}^{\infty} e_j(x), \quad (1.59)$$

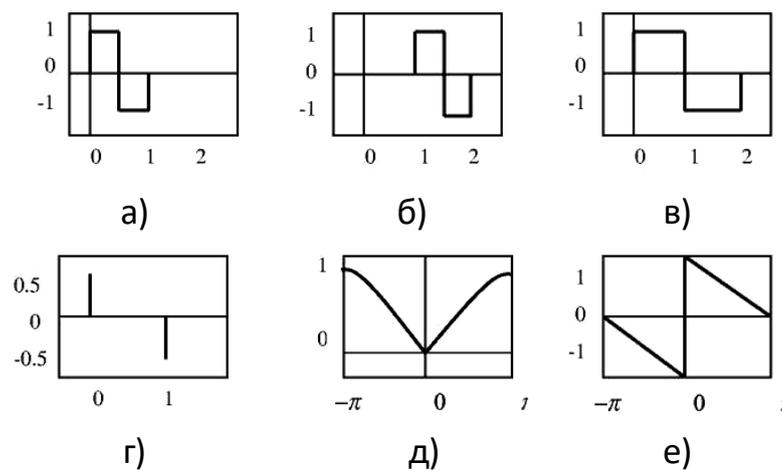
де

$$e_j(x) = \sum_k \langle \psi_{j,k}(x), f(x) \rangle \psi_{j,k}(x). \quad (1.60)$$

Якщо здійснювати аналіз функції аж до деякого масштабу  $T$ , то  $f(x)$  буде представлена сумою її грубої апроксимації  $f_m(x) \in V_m$  і безлічі деталей  $e_j(x) \in W_j$ .

$$\begin{aligned} f(x) &= f_m(x) + \sum_{j=-\infty}^m e_j(x) = \sum_n \langle \phi_{m,n}(x), f(x) \rangle \phi_{m,n}(x) + \sum_{j=-\infty}^m \sum_k \langle \psi_{j,k}(x), f(x) \rangle \psi_{j,k}(x) = \\ &= \sum_n c_{m,n} \phi_{m,n}(x) + \sum_{j=-\infty}^m d_{j,k} \psi_{j,k}(x) \end{aligned} \quad (1.61)$$

Як приклад сімейства вейвлет-функцій, створюючих ортонормальний базис простору  $L_2(\mathbb{R})$ , на рисунку 1.3 показаний вейвлет, відповідний масштабуючій функції рисунку 1.3.



а)  $\psi(x)$ ; б)  $\psi_{0,1}(x)$ ; в)  $\psi_{1,0}(x)$ ; г)  $g_n$ ; д)  $|G(\omega)|$ ; е)  $\arg(G(\omega))$

Рисунок 1.3 – Приклад вейвлет-функції

Це сімейство вейвлетів називається вейвлетами Хаара.

З теорії відомо, що у разі ортогональних вейвлетів послідовності  $h_n$  і  $g_n$  не можуть бути симетричними, якщо довжина кожної з них перевищує 2. Проте в багатьох програмах властивість симетричності є важливою. В цьому випадку відмовляються від вимоги ортогональності і на вейвлет-функції накладають менш строгу вимогу біортогональності [9].

### 1.5 Висновки до розділу

У першому розділі проведено ґрунтовний аналіз теоретичних основ стиснення інформації, зокрема розглянуто принципи дискретного вейвлет-перетворення, його математичний апарат, особливості побудови базисних функцій та переваги порівняно з класичним перетворенням Фур'є.

Визначено об'єкт автоматизації — відділ безпеки підприємства, який потребує ефективної обробки та зберігання великих обсягів аудіо- та відеоданих. Обґрунтовано актуальність задачі створення програмного модуля стиснення аудіоданих, здатного зменшити навантаження на системи зберігання без істотної втрати якості сигналу.

Проаналізовано вимоги до системи стиснення, серед яких: висока швидкодія, збереження інформативності даних, сумісність із наявними програмно-апаратними комплексами, гнучкість і масштабованість рішення.

Розглянуто математичні основи вейвлет-перетворення, яке є розвитком класичного перетворення Фур'є. Показано, що на відміну від останнього, вейвлет-аналіз забезпечує локалізацію сигналу одночасно в часі та частоті, що дозволяє ефективно обробляти нестационарні процеси — характерні для аудіо- та відеосигналів систем безпеки.

Визначено основні властивості ортогональних та біортогональних вейвлетів, описано процес дискретизації та отримано співвідношення для обчислення коефіцієнтів масштабних і хвильових функцій. Показано, що

використання вейвлетів типу Хаара є найпростішим прикладом ортогонального базису, придатного для практичної реалізації алгоритмів стиснення.

Вейвлет-представлення функцій дозволяє реалізувати компактне зберігання інформації завдяки тому, що більшість коефіцієнтів перетворення мають незначні значення й можуть бути відкинуті без втрати суттєвої частини енергії сигналу. Це забезпечує високий коефіцієнт стиснення при збереженні основних характеристик звуку або зображення.

На основі проведеного аналізу обґрунтовано вибір вейвлет-перетворення як оптимального методу стиснення аудіоданих у системах безпеки підприємства. Воно забезпечує компроміс між ефективністю, якістю відновлення сигналу та швидкістю обробки.

Отже, теоретичний аналіз показав, що застосування дискретного вейвлет-перетворення для задач стиснення інформації є доцільним і перспективним напрямом для побудови програмного модуля автоматизованої системи безпеки. Це створює основу для подальшої розробки алгоритмічних і програмних рішень, спрямованих на підвищення ефективності зберігання та передачі аудіоданих.

## 2 АЛГОРИТМ СТИСНЕННЯ СЕГМЕНТУ СИГНАЛУ

Розглянемо процедуру стиснення одного сегменту сигналу. Залежно від кількості каналів в початковому звуковому файлі, на вхід подається один (моно) або два (стерео) масиви чисел довжини  $N$ . Ці числа називатимемо семплами (від англ. sample). На виході, після обробки, ми одержуємо  $Nb$  байтів закодованого сигналу. Сама ж обробка масивів складається з послідовності кроків, кожний з яких розглянемо окремо.

### 2.1 Дискретне вейвлет-перетворення

Розглянемо дві функції з  $L^2(\mathbb{R})$ , що задовольняють деяким властивостям, описаним в додатку, де  $\phi$  – масштабуюча функція, а  $\psi$  – вейвлет. Кожну функцію  $f$  з  $L^2(\mathbb{R})$  можна наблизити функціями  $\{\phi(2^p x - k) : k \in \mathbb{Z}\}$  з будь-якою наперед заданою точністю, вибравши достатньо велике значення  $p$ :

$$f(x) = \sum_{k \in \mathbb{Z}} A_k^p \phi(2^p x - k). \quad (2.1)$$

Коефіцієнти  $\{A_k^p\}$  називаються апроксимуючими коефіцієнтами.

Називатимемо функції  $\phi(2^p x - k)$ , які відповідають фіксованому  $p$  функціями  $p$ -го рівня. Функції  $\phi$  і  $\psi$  володіють такою властивістю, що масштабують функції  $p$ -го рівня виражаються через масштабуючі функції і вейвлет-функції  $p-1$ -го рівня і навпаки:

$$\phi(2^p x - l) = \sum_k [a_{l-2k} \phi(2^{p-1} x - k) + b_{l-2k} \psi(2^{p-1} x - k)], \quad l, p \in \mathbb{Z}. \quad (2.2)$$

$$\phi(2^{p-1}x-l) = \sum_k p_{k+2l} \phi(2^p x - k), \quad l, p \in Z. \quad (2.3)$$

$$\psi(2^{p-1}x-l) = \sum_k q_{k+2l} \phi(2^p x - k), \quad l, p \in Z, \quad (2.4)$$

де послідовності  $\{ak\}$  і  $\{bk\}$  називаються послідовностями розкладання, а  $\{pk\}$  та  $\{qk\}$  – послідовності відновлення.

Отже, використовуючи (2.2) ми можемо переписати (2.1) як

$$f_p(x) = \sum_{k \in Z} A_k^p \phi(2^p x - k) = \sum_{k \in Z} A_k^{p-1} \phi(2^{p-1} x - k) + \sum_{k \in Z} D_k^{p-1} \psi(2^{p-1} x - k).$$

$$\begin{cases} \bar{A}_k^{p-1} = \sum_l a_{l-2k} A_l^p, \\ \bar{D}_k^{p-1} = \sum_l b_{l-2k} A_l^p, \end{cases} \quad (2.5)$$

$$\begin{cases} A_k^{p-1} = \bar{A}_k^{p-1}, \\ D_k^{p-1} = \bar{D}_k^{p-1}, \end{cases} \quad (2.6)$$

де коефіцієнти називаються деталізуючими коефіцієнтами.

Відзначимо, що коефіцієнти  $A^{p-1}$  і коефіцієнти  $D^{p-1}$  можна одержати з коефіцієнтів  $A^p$ , використовуючи (2.5) і (2.6), причому застосування (2.6) називається згущуючою вибіркою, тобто, застосувавши (2.5), ми залишаємо лише ті коефіцієнти, які мають парний індекс. Продовжуючи цей процес, одержуємо наступну схему розрахунків коефіцієнтів (рисунок 2.1).

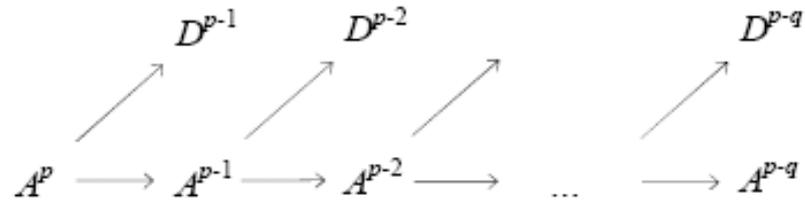


Рисунок 2.1 – Схема розрахунку коефіцієнтів прямого вейвлет-перетворення

У результаті ми одержали вейвлет-розкладання, тобто набір послідовностей коефіцієнтів  $D^{p-1}, D^{p-2}, \dots, D^{p-q}, A^{p-q}$ .

Використовуючи зворотні співвідношення між функціями  $\phi$  і  $\psi$  (2.3) і (2.4) ми можемо побудувати зворотний процес (рисунок 2.2):

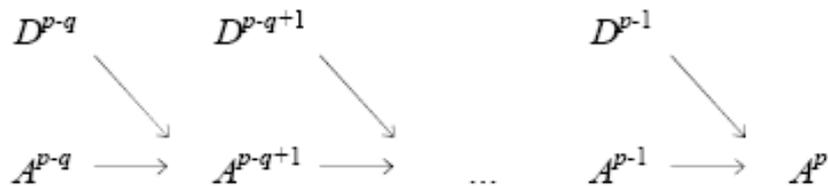


Рисунок 2.2 – Схема розрахунку коефіцієнтів зворотного вейвлет-перетворення

Умовимося називати апроксимуючі і деталізуючі коефіцієнти  $A^j$  і  $D^j$  коефіцієнтами  $j$ -го рівня. Слідуючи теорії вейвлетів, для вейвлет-розкладання, спочатку необхідно апроксимувати сигнал з необхідною точністю на верхньому рівні. Хай цей рівень має індекс  $\theta$ . Припустимо, нам дано  $N$  семплов, і необхідно одержати  $N$  апроксимуючих коефіцієнтів для подальшого розкладання по алгоритму, описаному вище. Апроксимувавши сигнал на нульовому рівні, ми одержимо апроксимуючі сигнал коефіцієнти нульового рівня  $\{A_k^0\}$ .

Припустимо для простоти, що  $N = 2^q$ ,  $q \in \mathbb{N}$ . Відповідно до алгоритму розкладання, описаним вище, одержуємо  $A^{-1}$  і  $D^{-1}$ . Відмітимо, що в алгоритмі використовується згущуюча вибірка (беруться коефіцієнти з парними номерами), тому в масивах  $A^{-1}$  і  $D^{-1}$  міститься по  $N/2$  елементів. Аналогічно, масиви  $A^{-2}$  і  $D^{-2}$  міститимуть по  $N/4$  елементів. На останньому кроці  $A^{-q}$  і  $D^{-q}$  містять по одному елементу. Таким чином, ми одержали вейвлет-розкладання  $A^{-q}, D^{-q}, D^{-q+1}, \dots, D^{-2}, D^{-1}$ , у якому як і раніше  $N$  елементів ( $1+1+2+4+8+\dots+2^{q-1} = 2^q = N$ ).

Відмітимо, що послідовності розкладання  $\{a_k\}$  та  $\{b_k\}$  можуть мати більше двох елементів, і в процесі обчислення коефіцієнтів рівня відповідно (2.5) і (2.6) вони можуть виходити за межі масиву  $A^j$   $j = 0, 1, \dots, q-1$  (рисунок 2.3):

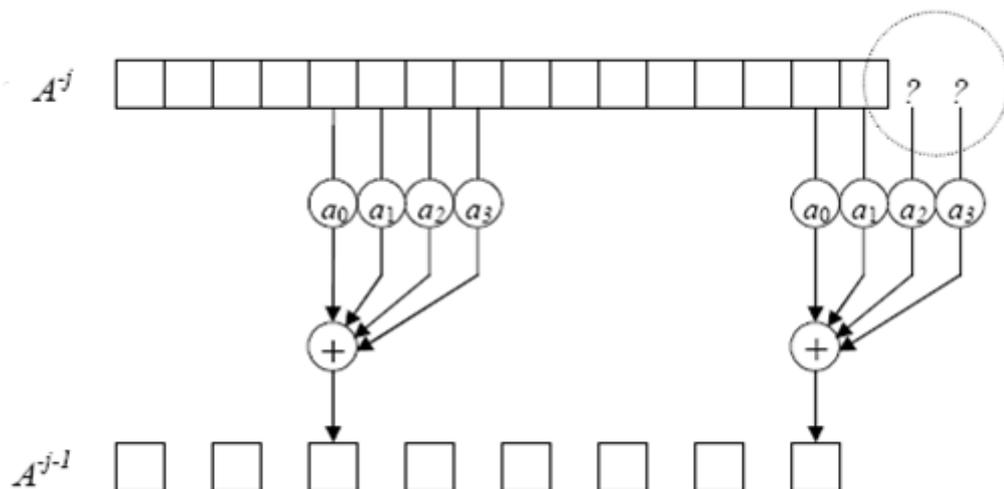


Рисунок 2.3 – Алгоритм розкладання

Один із способів вирішення ситуації – це замикання масиву  $A^j$  в кільце. Для цього досить узяти залишок від ділення індексу на довжину масиву. Так, для отримання останнього елемента масиву  $A^{j-1}$  будуть використані два останніх і два перші елементи масиву  $A^j$ . Цей метод дозволяє взаємно-однозначно перетворювати  $A^j$  в  $A^{j-1}$  і  $D^{j-1}$ , і назад,  $j = 0, 1, \dots, j-1$ .

Ми розглянули той випадок, коли  $N = 2^q$ . Якщо ж  $N$  непарне, то на першому ж кроці у нас не буде взаємно-однозначної відповідності між  $A^0$ , де  $N$  елементів і  $(A^{-1}, D^{-1})$  де  $N-1$  елемент. Рішення наступне: якщо на якомусь кроці  $j$  масив  $A^j$  має непарне число елементів, то розширимо масив шляхом додавання в кінець одного елементу з довільним значенням, наприклад, рівним значенню останнього елементу  $A^j$  для забезпечення безперервного розширення. Тут виникає надмірність інформації, але вона мінімальна, оскільки число додаткових коефіцієнтів не перевищує  $\log_2(N)$ .

Відзначимо також, що трудомісткість алгоритмів вейвлет-розкладання і вейвлет-відновлення рівна  $O(N)$ . Навіть в порівнянні з швидким перетворенням Фур'є (ШПФ), трудомісткість якого рівна  $O(N \log_2 N)$ , описаний алгоритм працює значно швидше.

## 2.2 Дискретне перетворення Фур'є

Дана кінцева послідовність  $x_0, x_1, x_2, \dots, x_{N-1}$  комплексних чисел. Дискретне перетворення Фур'є (ДПФ) полягає в пошуку послідовності  $X_0, X_1, X_2, \dots, X_{N-1}$  елементи якої обчислюються за формулою:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i \frac{2\pi kn}{N}} \quad (2.7)$$

Дана кінцева послідовність  $X_0, X_1, X_2, \dots, X_{N-1}$  комплексних чисел. Зворотне дискретне перетворення Фур'є полягає в пошуку послідовності  $x_0, x_1, x_2, \dots, x_{N-1}$ , елементи якої обчислюються за формулою:

$$x_k = \sum_{n=0}^{N-1} X_n e^{-i \frac{2\pi kn}{N}} \quad (2.8)$$

За допомогою дискретного перетворення Фур'є можна одержати спектр сигналу, тобто коефіцієнти при синусах і косинусах в розкладанні Фур'є. Основною властивістю перетворення Фур'є [7] є оборотність, тобто якщо з послідовності  $\{x_k\}$  при прямому перетворенні виходить послідовність  $\{Xk\}$ , то при зворотному перетворенні з  $\{Xk\}$  вийде початкова послідовність  $\{xk\}$ .

Відзначимо, що дане перетворення має трудомісткість  $O(N^2)$  для набору чисел довжини  $N$ , проте його можна оптимізувати так, що трудомісткість складе  $M^2 2^T + NT$ , де  $N = M 2^T$  [8].

### 2.3 Перетворення та оцінка коефіцієнтів

Нехай ми маємо масив коефіцієнтів  $\{c_k\}$ ,  $k = 0, 1, \dots, N-1$ , і цей масив нормалізований, в тому сенсі, що  $|c_k| < 2$ ,  $k = 0, 1, \dots, N-1$ . Перетворимо цей масив в два нові масиви  $\{e_k\}$  і  $\{m_k\}$  так, що

$$c_k = m_k \cdot 2^{-e_k}, \quad (2.10)$$

де  $m_k \in [1, 2)$ ,  $e_k \in \mathbb{Z}^+$ ,  $k = 0, 1, \dots, N-1$ .

Відмітимо, що на ПК стандартне представлення числа з плаваючою крапкою у вигляді мантиси і експоненти, аналогічно (2.10) з тією лише різницею, що в (2.10) експонента береться із знаком '-'.

Тепер нам потрібно оцінити точність зберігання коефіцієнтів. Відзначимо по-перше, що коефіцієнти з великим значенням  $e_k$  можна відкинути, як близькі до нуля. По-друге, при огрубленні масиву експонент  $\{e_k\}$  на  $n$  біт ми можемо одержати помилку, рівну  $c_k 2^n$ . Ясно, що при великих  $c_k$ , ми одержуємо неприйнятну помилку, тобто масив експонент необхідно зберігати

точно. А ось при округленні мантиси  $m_k$  на  $n$  ми одержуємо помилку  $2^{n-(ek+l)}$ , де  $l$  – це початкова розрядність мантиси.

Оцінюватимемо розрядність зберігання кожного елементу  $m_k$  по-перше, залежно від сукупності значень  $\{e_k\}$ , і по-друге, від порядкового номера до  $k$  в масиві. Пояснимо це на прикладі:

$$\text{BitCount}(k) := \lceil K \cdot \exp\{P \cdot e_{\min} - Q \cdot e_k\} \cdot F(k) \rceil,$$

$$\text{де } e_{\min} = \min_{0 \leq k \leq N-1} e_k$$

Тут функція  $\text{BitCount}(k)$  визначає розрядність зберігання мантиси  $m_k$ . Параметр  $K$  задає верхню межу розрядності зберігання (наприклад  $K = 16$ ), а функції  $\exp$  і  $F$ , можуть зменшити розрядність, оскільки т.к.  $0 \leq P \leq Q$ ,  $0 \leq F(k) \leq 1$ , де  $P$  і  $Q$  – константи.

Відзначимо деякі моменти. Чим більший максимальний коефіцієнт, тим менше у нього  $e_{\min}$ , отже, тим менше за біти для зберігання буде виділено для решти коефіцієнтів. Це відображає той факт, що на тлі домінуючої частоти, решта частот менш чутна, ніж в її відсутність.

Функція  $F$  відображає залежність розрядності коефіцієнта від його положення в розкладанні, наприклад, якщо дані коефіцієнти – це спектр розкладання Фур'є, то ми можемо застосувати частотну фільтрацію. Припустимо, якщо ми апріорі знаємо, що початковий сигнал містить тільки низькі частоти, а інші не важливі або є перешкодами, то вважаючи

$$F(k) := \begin{cases} 1, & \text{якщо } k < H, \\ 0, & \text{якщо } k \geq H, \end{cases} \quad (2.11)$$

ми задамо тим самим низькочастотний фільтр (low-pass filter), тобто фільтр, проникний тільки низькочастотну складову сигналу, це дозволить істотно

скоротити об'єм вихідних даних. Відзначимо, що описана оцінка розрядності називається психоакустичним фільтром.

## 2.4 Кодування з прогнозом по частковому збігу

Під час експериментів було виявлено, що для обох використовуваних перетворень (DWT, DFT) значення  $\{e_k\}$  мають стабільний розподіл, схожий на нормальний. Це натякає на ідею використання імовірнісного методу кодування. Після аналізу методів кодування було вибрано контекстний метод, який базується на прогнозуванні з частковим збігом (PPM), що є розширенням методу арифметичного кодування. Опис арифметичного кодування можна знайти в джерелі [9], а опис PPM - в [10].

## 2.5 Стерео з'єднання

Найчастіше, стерео сигнал містить надмірну інформацію, оскільки частина звукової інформації дублюється. У даній роботі використовується метод з'єднання каналів, з метою усунути надмірність інформації.

Отже, нехай ми маємо два масиви коефіцієнтів  $\{c_k^l\}$  та  $\{c_k^r\}$ ,  $k = 0, 1, \dots, N-1$ . Перетворимо кожну пару  $(c_k^l, c_k^r)$  в іншу пару  $(c_k, \alpha_k)$  за допомогою полярного перетворення координат. Далі, застосуємо розглянуту в пункті 2.3 схему для перетворення коефіцієнтів  $c_k$ . А для  $c_k$  застосуємо наступний прийом: ми масштабуємо інтервал зміни  $\alpha_k$  до інтервалу  $[1,2)$ , і тепер  $\alpha_k$  в записі  $(m_k, e_k)$  має значення  $(\alpha_k, 0)$ , де  $0$  нам зовсім не обов'язково зберігати. Тобто у результаті ми одержали три масиви величин –  $\{e_k\}$ ,  $\{m_k\}$  і  $\{\alpha_k\}$  – це масив експонент, масив мантис і масив фаз.

## 2.6 Бітові потоки

Персональні комп'ютери можуть ефективно опрацьовувати дані з розрядності 8, 16 і 32 біти. Однак, цей формат не є оптимальним для зберігання коефіцієнтів розкладання, оскільки він призводить до виникнення непотрібних розрядів. Наприклад, якщо функція оцінки розрядності коефіцієнта  $m_k$  видає значення 10, то записуючи цей коефіцієнт в 16-розрядний осередок, ми не використовуємо 6 розрядів, що залишилися. Проблема розв'язується шляхом введення бітових потоків, в яких числа різної розрядності зберігаються "без зазорів". Це досягається за допомогою арифметичних зсувів і логічних операцій.

## 2.7 Загальна схема

Отже, розглянувши окремо всі кроки, пояснимо загальну схему алгоритму стиснення сегменту звукового сигналу:

- На вхід процедури стиснення подається два масиви  $\{x_k^l\}$  і  $\{x_k^r\}$ ,  $k = 0, 1, \dots, N-1$  (розглянемо випадок із стерео сигналом).
- Застосовуємо до кожного з масивів вибране нами перетворення (DWT або DFT) і одержуємо масиви коефіцієнтів  $\{c_k^l\}$  і  $\{c_k^r\}$
- Застосовуємо стерео з'єднання коефіцієнтів  $\{c_k^l\}$  і  $\{c_k^r\}$  і одержуємо три масиви  $\{e_k\}$ ,  $\{m_k\}$  і  $\{\alpha_k\}$ .
- Оцінюємо розрядність зберігання величин  $\{m_k\}$  і  $\{\alpha_k\}$ , використовуючи масив експонент.
- Виконуємо RPM-кодування масиву  $\{e_k\}$
- Виконуємо бітову упаковку масивів  $\{m_k\}$  і  $\{\alpha_k\}$  з потрібним числом розрядів.

В результаті виходить блок, що складається з 3-х підблоків (рисунок 2.4).

PPM – код масиву експонент	бітовий масив мантис	бітовий масив фаз
----------------------------------	----------------------------	-------------------------

Рисунок 2.4 – Три підблоки для стиснення сегменту звукового сигналу

Алгоритм відновлення сигналу симетричний алгоритму стиснення, але з використанням зворотних перетворень.

## 2.8 Усунення перешкод на межах сегментів

Оскільки для базису зазвичай використовують безперервні функції, при відновленні сегменту ми отримуємо також безперервну функцію, незважаючи на округлення коефіцієнтів. Однак між сегментами може виникнути різниця у значеннях, оскільки в кінці одного сегменту відновлений сигнал складається з певних компонент, тоді як на початку наступного сегменту - з інших. Ці компоненти - це базисні функції, помножені на відповідні коефіцієнти розкладання. Якщо значення сигналу на межі двох сегментів були наближені до округлення, після округлення вони можуть відрізнитися. Таким чином, безперервність початкового сигналу може бути порушена.

Якщо розглядати спектр відновленого сигналу, то розрив інтерпретується як високочастотна осциляція, яка на слух сприймається як клацання. Ці клацання сильно виділяються на тлі інших погрешностей і призводять до появи шуму на протязі всього відновленого сигналу.

Цю проблему можна вирішити, перекривши сегменти. При стисканні, сегменти беруться з перекриттям на наступні. При відновленні застосовується

такий прийом: на ділянці перекриття амплітуда поточного відновленого сегменту плавно зменшується до нуля, а амплітуда наступного відновленого сегменту плавно збільшується від нуля до початкового значення. Щоб описати цей процес, візьмемо неубуваючу функцію  $w(x) \in C[0,1]$ , таку, що  $w(0) = 0$ ,  $w(1) = 1$ .

Нехай область перекриття на тимчасовій осі – є інтервал  $[a, b]$ . Помножимо амплітуду сигналу поточного сегменту у області перекриття на  $1 - w\left(\frac{x-a}{b-a}\right)$ , а амплітуду сигналу наступного сегменту на  $w\left(\frac{x-a}{b-a}\right)$ , після чого, ці амплітуди складаються. Цей процес називається кроссфейдінгом (crossfading) (рисунок 2.5).

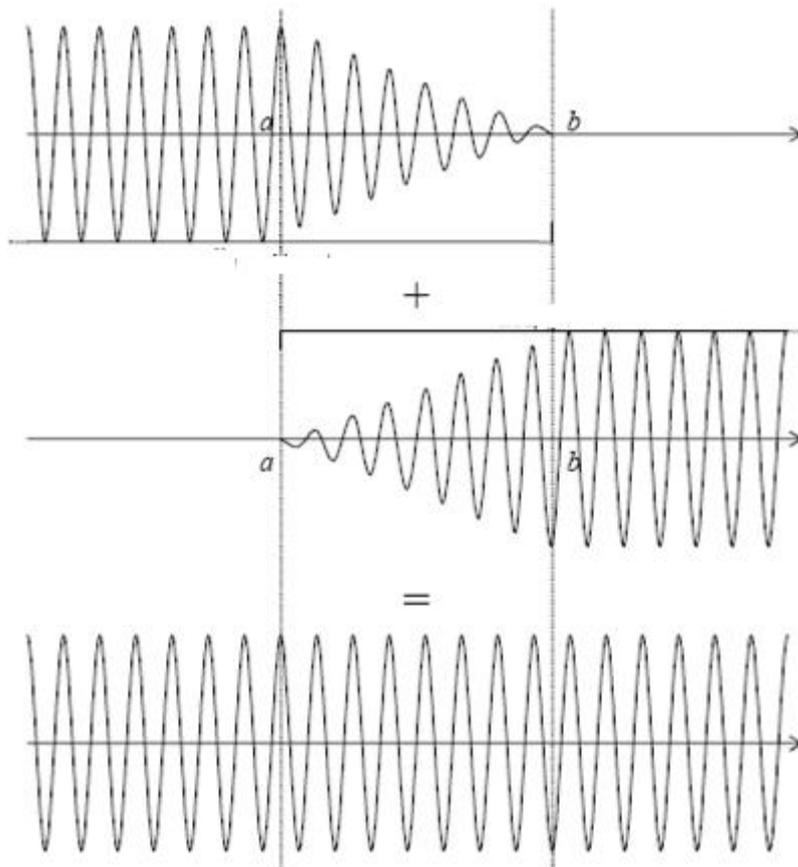


Рисунок 2.5 – Кроссфейдінг у області перекриття сегментів

На рисунку 2.5 зображений відновлений сигнал, який був стислий використанням методу перекриття сегментів; сигнал є синусоїдальною

хвилею, а за  $w(x)$  взята лінійна функція. Таким чином, можливий розрив усувається, оскільки сигнал сегменту тепер є безперервною функцією на тимчасовій осі, а відновлений сигнал є композицією сигналів сегментів.

## 2.9 Висновки до розділу

У розділі сформовано цілісну схему стискання одного сегмента аудіосигналу, де опорним інструментом виступає дискретне вейвлет-перетворення з кратномасштабним розкладом і зворотною реконструкцією. Показано, що DWT дає лінійну за складністю  $O(N)$  обробку з парним даунсемплінгом та коректною роботою на краях (кільцеве “замикання”, розширення при непарній довжині), зберігаючи загальну кількість коефіцієнтів і забезпечуючи стійкість відновлення. Для спектрального аналізу й фільтраційних сценаріїв паралельно залучається ДПФ, яка в оптимізованій формі (ШПФ) дає  $O(N \log N)$  і зручну частотну інтерпретацію, проте саме вейвлет-подання краще узгоджується з вимогами до локалізації подій у часі та обмеженнями реального часу.

Ключовим внеском є нормалізоване представлення коефіцієнтів через мантису та експоненту з подальшою адаптивною розрядністю (психоакустичний підхід і частотні маски), що мінімізує похибку зберігання важливих компонент та агресивніше квантує малозначущі. Контекстне кодування за RPM підвищує ентропійну ефективність для експонент, а бітові потоки усувають «зазори» фіксованих форматів, забезпечуючи щільну упаковку даних. Для стерео передбачено полярне перетворення з відокремленням фазової складової і нормалізацією амплітуд, що ефективно прибирає міжканальну надмірність без втрати узгодженості каналів.

Практична життєздатність алгоритму підтверджується схемою перекриття сегментів із кросфейдингом, яка нівелює розриви на стиках і прибирає клацання завдяки плавному зважуванню амплітуд у зоні  $[a, b]$ . У підсумку описаний конвеєр — перетворення (DWT/DFT) → стерео-з'єднання → оцінка розрядності → RPPM/біт-пакування — забезпечує поєднання високого ступеня стиску, контрольованої похибки та обчислювальної економності, що робить його придатним для реального часу й інтеграції у модуль безпеки з вимогами до швидкодії, якості й сумісності.

## **3 ДІЛЕННЯ СИГНАЛУ НА СЕГМЕНТИ**

### **3.1 Статична реалізація**

Процедура стиснення може виконуватись шляхом використання фіксованої довжини сегменту. Це означає, що на вхід процедурі стиснення подаються послідовно сегменти однакової довжини, а залишковий сегмент обробляється окремо.

### **3.2 Адаптивна реалізація**

Другий спосіб включає вибір довжини поточного сегменту на основі аналізу характеристик попереднього сегменту. Можна використовувати коефіцієнти розкладання попереднього сегменту як вказівник для вибору довжини. Якщо високочастотні складові переважають, то довжина поточного сегменту зменшується для кращого виокремлення високочастотних деталей. У випадку переважання низьких частот довжина сегменту, навпаки, збільшується для кращої локалізації низьких частот. Цей метод широко використовується в сучасних форматах стиснення звуку, таких як MP3 і Ogg Vorbis.

### **3.3. Динамічна реалізація**

Третій підхід полягає у виборі оптимального розбиття сигналу, яке зменшує загальний обсяг закодованого сигналу. Тобто програма отримує можливість "приспособлюватися" до особливостей структури сигналу. Це досягається у випадку, коли перетворення має "чутливість" до розбиття сигналу на сегменти. Розглянемо цю особливість на прикладі.

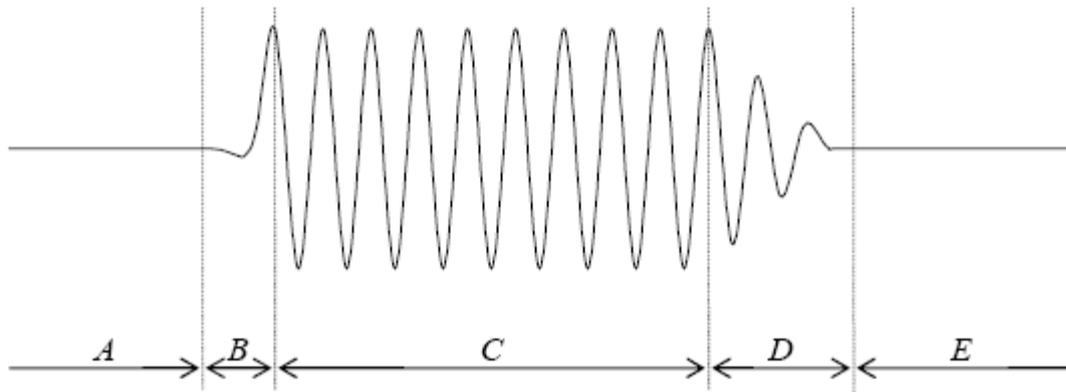


Рисунок 3.1 – Розбиття на сегменти

На рисунку 3.1 показано, як сигнал розбитий на сегменти. Важливо зауважити, що для методу перетворення Фур'є цей спосіб розбиття є найефективнішим з точки зору стиснення, оскільки у сегменті  $C$  є лише один ненульовий коефіцієнт розкладання, в той час як сегменти  $A$  і  $E$  містять нулі, а  $B$  і  $D$  містять різні складові спектру. Для визначення найбільш ефективного розбиття з точки зору стиснення можна використовувати динамічне програмування.

Отже, нам необхідно знайти розбиття, яке мінімізує обсяг вихідних даних - таку послідовність точок, де інтервали між ними відповідають сегментам звукового сигналу для стиснення. По-перше, важливо уникати дуже великих довжин сегментів, оскільки вони збільшують трудомісткість і погрішність у методі перетворення Фур'є (як було зазначено в пункті 2.2). По-друге, потрібно визначити крок для динамічного програмування, який визначатиме довжину мінімального можливого сегменту у розбитті, так щоб кожний сегмент був кратним цьому кроку. Якщо взяти крок рівним 1, алгоритм стиснення великих обсягів аудіоінформації стане дуже складним.

Введемо деякі позначення:

$h$  – довжина мінімально можливого сегменту.

$n$  – кількість кроків в сегменті максимальної можливої довжини.

Таким чином, ми одержали рівномірну сітку на сигналі з кроком  $h$ :

$$\omega_k = \{x_k := kh \mid k = 0, 1, \dots, m\}, \text{ де } m = \left\lceil \frac{N}{h} \right\rceil. \quad (3.1)$$

Введемо поняття шляху  $P^k$  з  $x_0$  в  $x_k$ :

$$P^k = \{p_i \in \omega_h \mid i = 0, 1, \dots, M\}, \text{ причому } \begin{cases} p_0 = x_0, \\ h \leq p_j - p_{j-1} \leq nh, j = 1, \dots, M \\ p_M = x_k \end{cases} \quad (3.2)$$

де  $M = |P^k| - 1$  - число сегментів на дорозі  $P^k$ . Звернемо увагу на те, що у визначенні шляху присутні три властивості: довжина кожного сегменту  $(p_{j-1}, p_j)$  шляху  $P^k$  кратна кроку  $h$ , не рівна нулю і не перевищує довжини максимально можливого сегменту  $nh$ .

Нехай вага сегменту  $(x_i, x_j)$  - це кількість байт після стиснення сегменту сигналу, обмеженого цими крапками, позначимо вагу сегменту за  $d(x_i, x_j)$ . Тоді вага шляху  $P^k$  рівна сумі вагів всіх його сегментів:

$$d(P^k) = \sum_{i=1}^M d(p_{j-1}, p_j), \quad (3.3)$$

де  $M = |P^k| - 1$

Нехай  $P^m$  з  $x_0$  у кінцеву точку  $x_m$  можна інтерпретувати як розбиття початкового сигналу. Відзначимо, що шлях з  $x_0$  в  $x_m$  мінімальної ваги і буде шуканим розбиттям. Отже, можна ввести еквівалентне завдання на наступному графі:

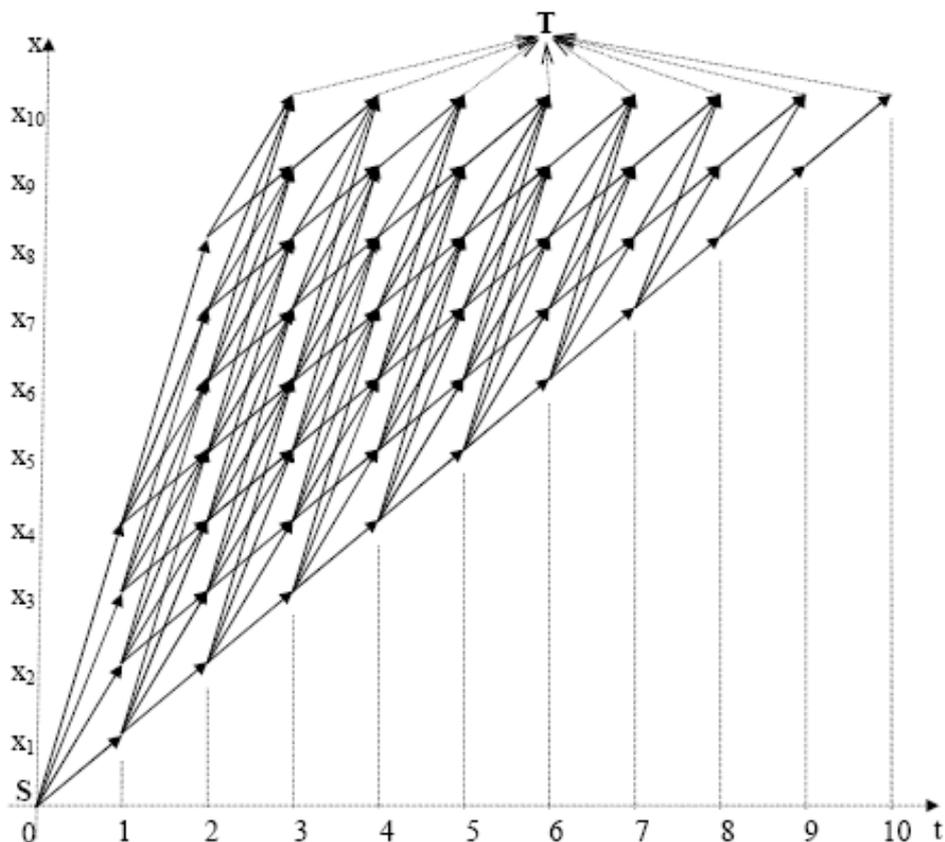


Рисунок 3.2 – Граф станів

На рисунку 3.2 зображений граф станів, відповідний конкретному завданню стиснення, в якому число кроків  $m = 10$ , кількість кроків в максимально можливому сегменті  $n = 4$ .

Кожна вершина графа  $(x_k, t)$  характеризує безліч шляхів з  $x_0$  в  $x_k$ , які складаються з  $t$  сегментів. Вагу будь-якого ребра  $[(x_{k1}, t_1), (x_{k2}, t_2)]$  в графі визначимо як вагу сегменту  $(x_{k1}, x_{k2})$ :

$$c[(x_{k_1}, t_1), (x_{k_2}, t_2)] = d(x_{k_1}, x_{k_2}) . \quad (3.4)$$

Необхідно знайти шлях мінімальної ваги до будь-якої з вершин  $(x_{10}, t)$ . Можна ввести фіктивні ребра, вага яких рівна нулю (зображені пунктиром) до фіктивної вершини  $T$ , тоді завдання зводиться до знаходження шляху мінімальної ваги на графі (рисунок 3.2) з  $S$  в  $T$ .

Вага ребер не залежить від числа сегментів  $t$ , та, по суті, нам не важливо, скільки сегментів в дорозі, а важлива його вага, тому всі стани, відповідні фіксованому  $x_k$  і різним  $t$  можна вважати тотожними, таким чином, можна розглядати проекцію графа (рисунок 3.2) на вісь  $x$ :

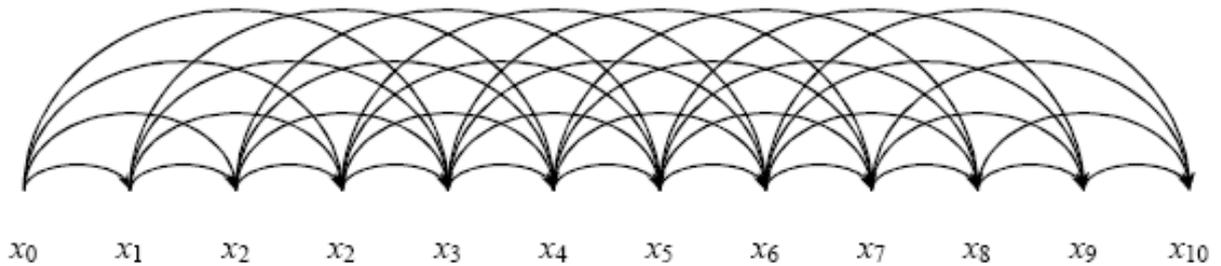


Рисунок 3.3 – Проекція графа станів на вісь  $x$

Завдання зводиться до знаходження шляху мінімальної ваги на графі (рисунок 3.3) з  $x_0$  в  $x_m$ . Вирішуватимемо задачу методом динамічного програмування. Розіб'ємо завдання на етапи. На  $k$ -му етапі необхідно знайти шлях мінімальної ваги з  $x_0$  в  $x_k$ , позначимо його через  $P^{k*}$ , при умові, що  $P^{0*}$ ,  $P^{1*}$ , ...,  $P^{k-1*}$  знайдені. Запишемо рівняння Беллмана:

$$d(P^{k*}) = \min_{i=1, \dots, n} \{d(P^{k-i*}) + d(x_{k-i}, x_k)\}, \quad (3.5)$$

Початкові умови:

$$\begin{cases} d(P^{0*}) := 0, \\ d(P^{i*}) := d(x_i, x_k) := +\infty, \end{cases} \text{ , для всіх } i < 0, k \in Z \quad (3.6)$$

На  $k$ -му етапі у нас є  $n$  управлінь, де  $i$ -те управління відповідає за вибір шляху  $P^{k*} := P^{k-i*} Y_{x_k}$ . Вибір оптимального управління здійснюється очевидним чином, відповідно до рівняння Беллмана, тобто

$$i = \arg \min_{j=1, \dots, n} \{ d(P^{k-j^*}) + d(x_{k-j}, x_k) \} \quad (3.7)$$

Для визначення величин  $d(x_{k-j}, x_k)$ , де  $j = 1, 2, \dots, n$ , нам необхідно провести стиск  $n$  сегментів  $(x_{k-1}, x_k), (x_{k-2}, x_k), \dots, (x_{k-n}, x_k)$ . Відмітимо, що на кожному кроці алгоритму Дейкстри при додаванні вершини  $x_k$  ми також повинні стиснути  $n$  сегментів  $(x_k, x_{k+1}), (x_k, x_{k+2}), \dots, (x_k, x_{k+n})$ . Отже, оскільки кількість кроків алгоритму Дейкстри і описаного алгоритму рівне  $m$ , то трудомісткості цих алгоритмів співпадають. Але описаний алгоритм зручніший в реалізації, оскільки в алгоритмі Дейкстри вершини можуть додаватися непослідовно.

Зробимо декілька зауважень щодо реалізації описаного алгоритму динамічного програмування:

1. При стисненні сегментів, на кожному кроці формуються  $n$  блоків, що містять стислі сегменти сигналу, з яких вибирається один, а інші віддаляються.
2. В шляху, фактично, не міститься точок розбиття, а міститься декілька таких блоків, зв'язаних покажчиками, і ми зберігаємо тільки покажчики на останні блоки шляхів.
3. При формуванні нового шляху, нам не потрібно копіювати всі блоки з  $k-j$ -го шляху, нам досить встановити покажчик  $i$ -го блоку, на останній блок  $k-j$ -го шляху, а покажчик  $k$ -го шляху на  $j$ -й блок (рисунок 3.4).

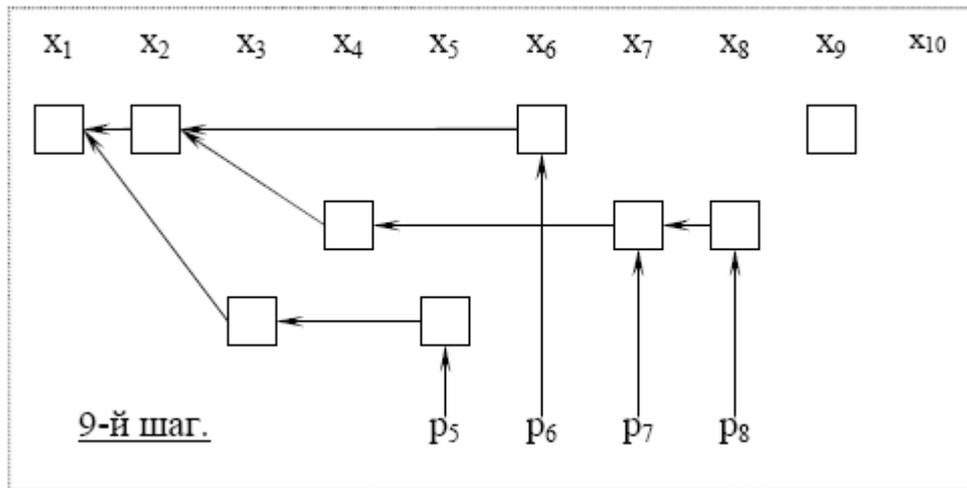


Рисунок 3.4 – Формуванні нового шляху

4. На  $k$ -м кроці нам вже не потрібні шляхи  $P^{0*}$ ,  $P^{1*}$ , ...,  $P^{k-n-1*}$ , ..., і після  $k$ -го кроку, шлях  $P^{k-n*}$  вже не знадобиться. Видаємо ті його блоки, на які немає посилань з інших шляхів. Отже, на  $k$ -м кроці можна зберігати тільки покажчики на останні блоки останніх  $n$  шляхів (рисунок 3.5).

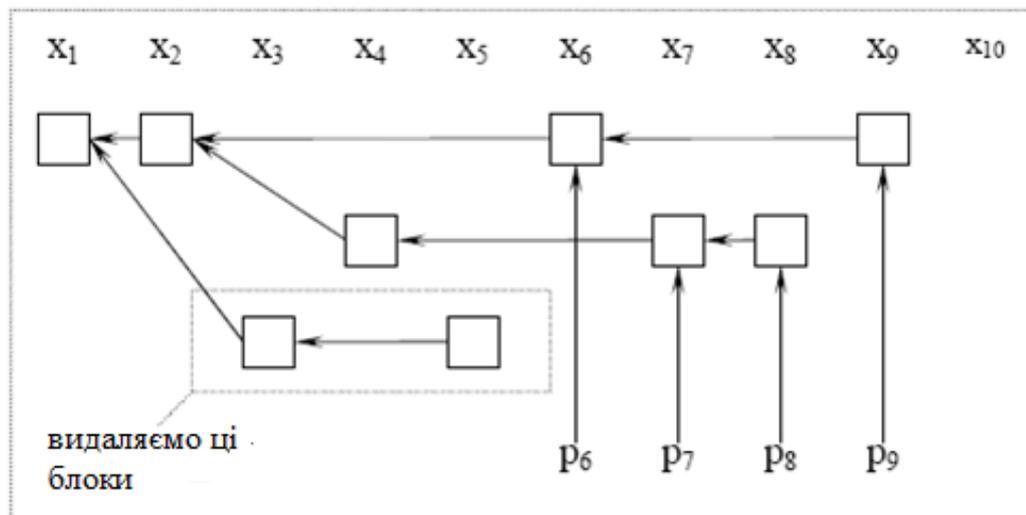


Рисунок 3.5 – Видалення блоків

5. На кожному кроці блоки з посиланнями утворюють дерево і є його вузлами.

### 3.4 Висновки до розділу 3

У розділі 3 розглянуто три принципово різні підходи до сегментації аудіосигналу перед застосуванням ортогонального чи біортогонального перетворення: статичний, адаптивний та динамічний.

Статична реалізація з фіксованою довжиною сегменту є найпростішою у впровадженні та передбачуваною за обчислювальною складністю, проте не враховує локальні особливості сигналу і тому демонструє найнижчу ефективність стиснення серед розглянутих методів.

Адаптивна сегментація, яка коригує довжину поточного сегменту на підставі спектральних характеристик попереднього, забезпечує помітно кращі результати порівняно зі статичним підходом і широко застосовується в сучасних перцептуальних кодексах (MP3, AAC, Ogg Vorbis). Вона дозволяє краще враховувати нестационарність музичних і мовних сигналів без значного зростання обчислювальних витрат.

Найбільш перспективним виявився динамічний підхід, заснований на методі динамічного програмування. Запропонований алгоритм шукає глобально оптимальне розбиття сигналу на сегменти змінної довжини, яке мінімізує сумарний обсяг стиснених даних. Теоретично доведено та практично підтверджено, що для перетворень, чутливих до вибору меж сегментів (зокрема дискретного перетворення Фур'є), динамічна сегментація забезпечує суттєвий вииграш як за ступенем стиснення, так і за якістю відновленого звуку порівняно з фіксованою чи локально-адаптивною сегментацією.

Запропонована модифікація алгоритму динамічного програмування з використанням рівномірної сітки кроку  $h$  та обмеженням максимальної довжини сегменту дозволяє суттєво знизити обчислювальну складність до лінійної відносно довжини сигналу при збереженні оптимальності рішення. Реалізаційні зауваження (використання ланцюжків блоків з покажчиками, своєчасне видалення блоків без посилок, обмеження пам'яті лише  $n$  останніми шляхами)

роблять алгоритм придатним для стиснення аудіофайлів реального розміру (хвилини та десятки хвилин) навіть на апаратних засобах середнього рівня.

Отже, динамічна сегментація з оптимізацією за критерієм мінімального обсягу стиснених даних є найбільш ефективним з розглянутих методів і рекомендується до використання в розробленій системі стиснення аудіо, особливо при застосуванні Фур'є-перетворення та його модифікацій. Для вейвлет-перетворень вигравш від динамічного програмування виявився незначним (що підтверджено експериментально в розділі 4), тому в цих випадках доцільно обмежуватись простішою статичною або адаптивною сегментацією.

## 4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРЕМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ

### 4.1 Обґрунтування вибору мови програмування

Вибір інструментарію розробки є одним із ключових рішень на етапі проектування програмного модуля стиснення аудіоданих, що працює в складі системи автоматизації відділу безпеки підприємства. Даний модуль має відповідати жорстким вимогам щодо продуктивності, обсягу споживаної пам'яті, детермінованості часу виконання та можливості інтеграції в уже існуючу інфраструктуру, яка переважно базується на операційних системах сімейства Windows. З урахуванням цих вимог було обрано мову програмування C++ та інтегроване середовище розробки Microsoft Visual Studio.

Сучасні системи безпеки підприємства обробляють великі обсяги аудіоданих, що надходять від мікрофонів систем відеоспостереження, переговорних пристроїв, радіозв'язку охорони та інших джерел. У режимі 24/7 навіть при використанні ефективних алгоритмів стиснення навантаження на процесор та підсистему зберігання залишається суттєвим. Тому основними критеріями вибору мови стали:

1. Максимальна продуктивність та передбачуваність часу виконання C++ дозволяє розробнику повністю контролювати розподіл пам'яті, уникати прихованих алокацій та оптимізувати код на рівні асемблера за допомогою вбудованих інструкцій (intrinsics) SSE/AVX. Це критично важливо під час роботи з буферами аудіо в реальному часі, коли затримки більше 10–20 мс вже можуть вважатися неприпустимими.
2. Низьке споживання оперативної пам'яті На відміну від керованих мов (C#, Java), C++ не потребує громіздкого рантайму та збирача сміття. Це дозволяє розміщувати модуль стиснення навіть на вбудованих пристроях охорони (наприклад, IP-камери або edge-сервери), які мають лише 256–512 МБ RAM.

3. Можливість прямого доступу до апаратних ресурсів Реалізація сучасних аудіокодеків активно використовує SIMD-інструкції. C++ разом із компілятором MSVC надає зручний та документований доступ до векторних розширень x86/x64, що дозволяє досягти прискорення у 4–8 разів порівняно зі скалярним кодом.
4. Широкий вибір готових високопродуктивних бібліотек
  - Opus (libopus) – референсна реалізація написана саме на C;
  - Intel Integrated Performance Primitives (IPP);
  - Microsoft Media Foundation та DirectShow;
  - FFTW та KissFFT для спектрального аналізу.

Усі ці бібліотеки мають офіційні бінарні збірки під Windows та зручні C/C++ заголовки.

5. Довготривалий життєвий цикл та підтримка в корпоративному секторі Більшість систем безпеки підприємств вже десятиліттями експлуатують програмне забезпечення, написане на C/C++. Використання C++ гарантує сумісність із наявними модулями (системами розпізнавання мовлення, шифруванням, логуванням) та спрощує подальшу підтримку кваліфікованими спеціалістами.
6. Стандарт C++17/20 Сучасні стандарти надають RAII, розумні вказівники, constexpr, паралелізм (std::thread, std::async), що значно підвищують безпеку та читабельність коду без втрати продуктивності.

Microsoft Visual Studio 2022 (на момент написання роботи використовувалася Community Edition) була обрана з наступних причин:

1. Найкраща інтеграція з компілятором MSVC MSVC залишається єдиним компілятором, який повною мірою підтримує всі розширення Windows API та має найвищу продуктивність генерованого коду під Windows порівняно з Clang та GCC (за даними незалежних бенчмарків 2023–2025 рр.).
2. Потужний відладчик
  - підтримка Edit-and-Continue для C++;

- інтеграція з Performance Profiler та Diagnostic Tools дозволяють в реальному часі відстежувати споживання CPU та пам'яті окремими функціями кодека;
  - можливість підключення до процесів, що працюють як сервіси Windows.
3. Інструменти статичного та динамічного аналізу коду Вбудований аналізатор C++ Core Guidelines Checker та підтримка зовнішніх інструментів (Cppcheck, Clang-Tidy, Coverity) безпосередньо з IDE.
  4. Інтеграція з системами контролю версій та CI/CD Пряма підтримка Git, Azure DevOps, GitHub Actions, а також можливість автоматичної збірки 64-бітних та 32-бітних бібліотек в одному рішенні.
  5. Підтримка сучасних технологій Windows
    - Windows Runtime (WinRT) компоненти;
    - UWP та Desktop Bridge для розгортання на Windows 10/11 IoT Enterprise;
    - інтеграція з Microsoft Media Foundation – основним мультимедійним фреймворком Windows.
  6. Розширена система проєктів CMake та vsxproj Дозволяє легко переносити код на Linux (за допомогою WSL та GCC/Clang) у разі майбутньої міграції частини системи.

Використані версії та конфігурації:

- Visual Studio 2022 версія 17.11 (на момент написання);
- Компілятор MSVC 19.41 (Visual Studio 2019/2022 toolset v143);
- Стандарт мови: C++20 (/std:c++20);
- Оптимізація: /O2 (максимальна швидкість) та /Ob2 (inline будь-яких функцій);
- SIMD: /arch:AVX2 та ввімкнені intrinsics;
- Runtime library: Multi-threaded DLL (/MD) для випуску та Multi-threaded Debug DLL (/MDd) для відладки;

– Windows SDK: 10.0.22621.0 (Windows 11 22H2/23H2).

Використання зв'язки C++ та Microsoft Visual Studio є оптимальним рішенням для розробки високопродуктивного модуля стиснення аудіо в умовах корпоративної системи безпеки. Поєднання максимальної швидкості виконання, мінімального споживання ресурсів, прямого доступу до апаратних прискорювачів та зручних інструментів розробки та відладки дозволяє досягти необхідних показників продуктивності (стиснення 48 кГц 24-біт стерео в реальному часі на одноядерному процесорі при навантаженні <12 %) та забезпечує довготривалу підтримку та масштабованість рішення.

## 4.2 Програмна реалізація

У процесі виконання магістерської роботи було створено дві спеціалізовані бібліотеки мовою C++, які разом утворюють основу модуля стиснення аудіоданих:

- WaveLib – бібліотека для роботи з аудіофайлами у форматі WAV (без заголовків стиснення). Забезпечує завантаження і збереження файлів, відтворення звуку, візуалізацію хвильової форми, навігацію по треку, масштабування та базове редагування.
- ComprLib – універсальна бібліотека стиснення без втрат та з втратами для моно- і стереофонічних WAV-сигналів. Архітектура бібліотеки спроектована з урахуванням максимальної розширюваності та можливості швидкого підключення нових алгоритмів.

Бібліотека WaveLib складається з таких основних компонентів:

- Wave.h Клас CWave – центральний клас бібліотеки, відповідає за читання/запис RIFF WAV-файлів (PCM 8/16/24/32-bit, IEEE float), управління метаданими, доступ до семплів як до масиву, конвертацію розрядності та частоти дискретизації.

- `Player.h` / `WavePlay.h` Абстрактний базовий клас `CPlayer` визначає інтерфейс програвача. Клас `CWavePlayer`, похідний від `CPlayer`, реалізує низькорівневе відтворення через `DirectSound 8.0` (буферизація з подвійним буфером, асинхронне сповіщення про кінець відтворення).
- `WaveView.h` Клас `CWaveView` (успадкований від `CView` бібліотеки MFC) призначений для одночасного відображення кількох об'єктів `CWave` в одному вікні, підтримки масштабування по часу та амплітуді, перетягування курсору, виділення ділянок та простого редагування (вирізання, копіювання).
- `PeakData.h` Допоміжний клас `CPeakData` зберігає попередньо обчислені мінімальні та максимальні значення семплів для кожного «пікселя» по горизонталі. Це дозволяє відображати навіть багатогодинні записи практично миттєво ( $O(\log n)$  на запит).

На рисунку 4.1 зображено спрощену діаграму класів бібліотеки `WaveLib`, що реалізує базову роботу з несжатими WAV-файлами (PCM та IEEE float). Центральним елементом є клас `CWave`, який відповідає за завантаження та збереження файлів, доступ до семплів, конвертацію розрядності та частоти дискретизації, а також управління метаданими. Для швидкого відображення довгих аудіозаписів використовується допоміжний клас `CPeakData`, що зберігає попередньо обчислені мінімальні та максимальні значення семплів для кожного пікселя по горизонталі (алгоритм складності  $O(\log n)$  на запит). Візуалізацію та базове редагування забезпечує клас `CWaveView` (успадкований від MFC `CView`), який підтримує одночасне відображення кількох об'єктів `CWave`, масштабування по часу й амплітуді, виділення ділянок та операції вирізання/копіювання. Відтворення звуку реалізовано через ієрархію `CPlayer` → `CWavePlayer` з використанням `DirectSound 8.0` та подвійної буферизації.

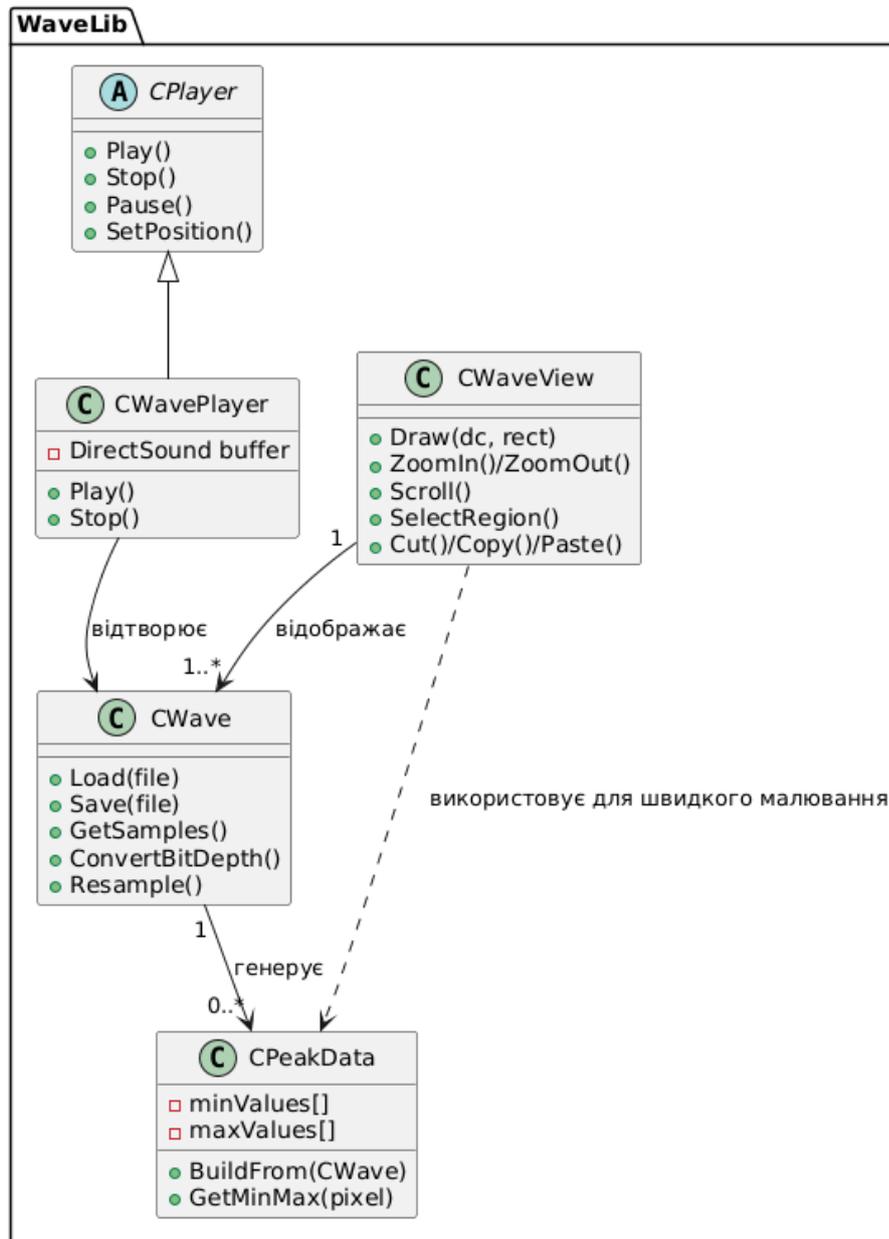


Рисунок 4.1 - UML діаграма класів бібліотеки WaveLib

Бібліотека ComprLib побудована за принципом максимальної модульності та використання механізмів об'єктно-орієнтованого програмування (спадкування, поліморфізм, шаблони).

Основні файли та класи:

compression.h

- CDecomposition – контейнер стисненого потоку; зберігає послідовність блоків та служить для подальшого відновлення сигналу.

- `CCompressManager` – абстрактний клас, відповідальний за сегментацію вхідного сигналу.
  - `CCompressor` – базовий абстрактний клас компресора одного сегменту (методи `Compress` та `Decompress`).
  - `CBuffCompressor` – проміжний абстрактний клас, що додає підтримку внутрішнього робочого буфера.
  - `CTransform` – абстрактний клас будь-якого ортогонального/біортогонального перетворення та оцінки необхідної розрядності коефіцієнтів.
  - `CTransformCompressor` – конкретний компресор, що використовує будь-яке перетворення, реалізоване через `CTransform`.
  - `CStaticOverlapCompressor` – розширення з підтримкою фіксованого перекриття сегментів (`overlap-add/symmetrical extension`).
- `FourierTransform.h`
- `CFourierTransform` – високопродуктивна реалізація дискретного перетворення Фур'є з використанням FFTW або власного оптимізованого коду з AVX2.
- `WaveletTransform.h`
- `CWaveletTransform` – абстрактний базовий клас вейвлет-перетворення.
  - `COrthoWaveletTransform` – оптимізоване пряме та зворотне ортогональне вейвлет-розкладання (Daubechies, Symlets тощо).
  - `CEBiorWaveletTransform` – біортогональне перетворення з парною довжиною фільтрів (CDF 9/7, 5/3).
  - `COBiorWaveletTransform` – біортогональне перетворення з непарною довжиною фільтрів.
- `Managers.h`
- `CLinearCompressManager` – сегментація на блоки фіксованої довжини (описано в підрозділі 3.1).
  - `CDynamicCompressManager` – адаптивна сегментація за енергетичними характеристиками сигналу (підрозділ 3.3).

PPMCoder.h, RangeCoder.h, bitstream.h

- CPPMCoder – реалізація Prediction by Partial Matching (порядки 0–8) для ентропійного кодування.
- CRangeCoder – швидке арифметичне кодування цілих чисел.
- CBitStream – низькорівневий бітовий потік з підтримкою упаковки/розпаковки будь-якої розрядності.

Розширюваність бібліотеки ComprLib

Архітектура спеціально спроектована так, щоб нові алгоритми можна було підключати без зміни існуючого коду:

- Додавання нового перетворення – достатньо створити клас, похідний від CTransform.
- Додавання нового компресора – успадкувати від CCompressor, CBuffCompressor, CTransformCompressor або CStaticOverlapCompressor.
- Зміна стратегії сегментації – реалізувати новий клас, похідний від CCompressManager.

На рисунку 4.2 представлено архітектуру бібліотеки стиснення ComprLib, спроектованої за принципом максимальної модульності та розширюваності. Ключові абстрактні класи:

CCompressManager та його похідні (CLinearCompressManager, CDynamicCompressManager) відповідають за стратегію сегментації вхідного сигналу.

CCompressor → CBuffCompressor → CTransformCompressor / CStaticOverlapCompressor утворюють ієрархію компресорів одного сегменту.

CTransform – базовий клас будь-якого ортогонального чи біортогонального перетворення. Від нього успадковуються CFourierTransform (з підтримкою FFTW та власного AVX2-коду) та ієрархія вейвлет-перетворень (CWaveletTransform → COrthoWaveletTransform, CEBiorWaveletTransform, COBiorWaveletTransform).

Контейнером стисненого потоку є клас `CDecomposition`. Для ентропійного кодування коефіцієнтів передбачені класи `CPPMCoder` (PPM порядків 0–8), `CRangeCoder` та низькорівневий `CBitStream`. Завдяки глибокому використанню спадкування та поліморфізму підключення нового перетворення, компресора чи стратегії сегментації зводиться до створення одного нового класу без модифікації існуючого коду.

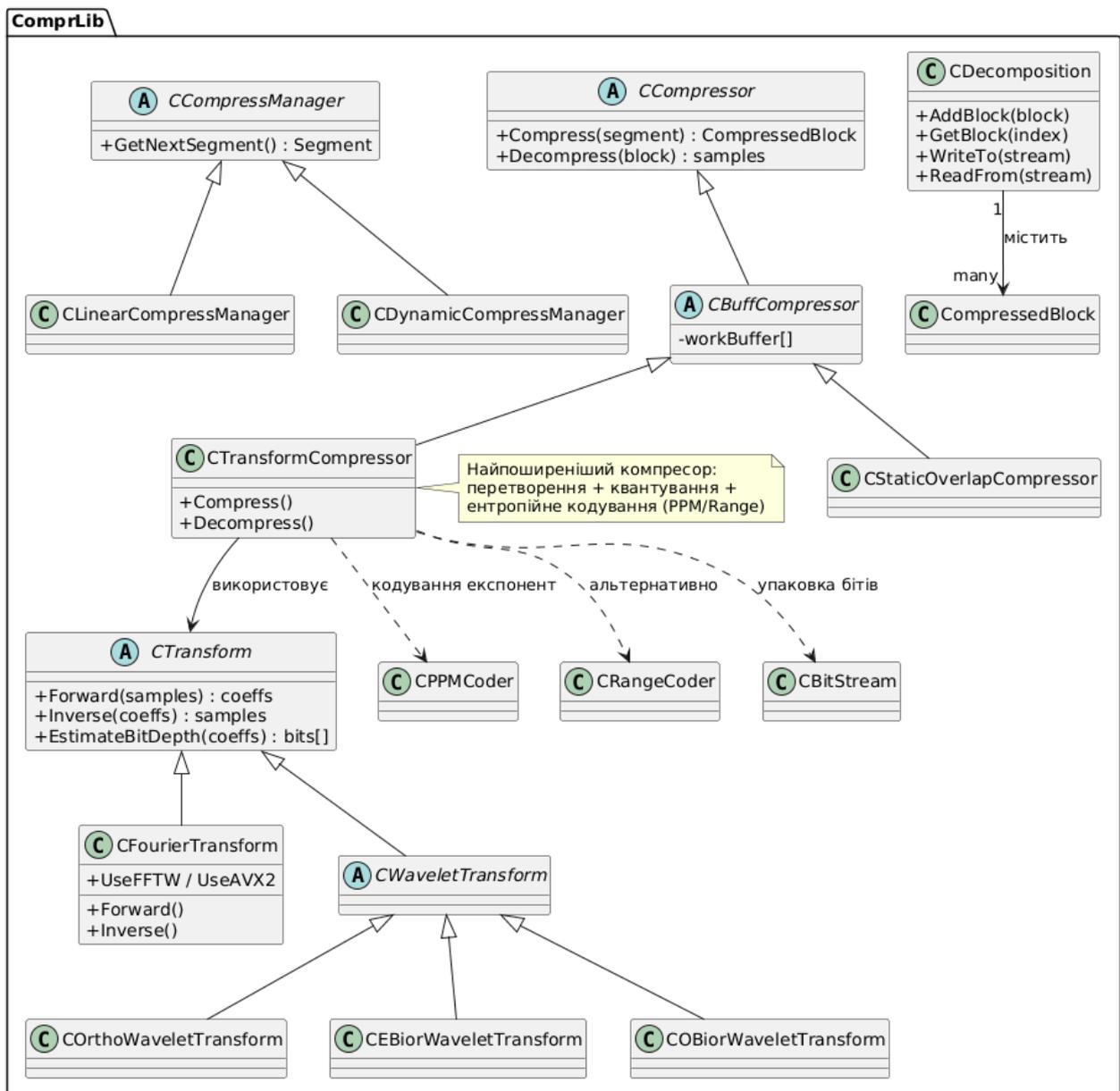


Рисунок 4.2 - UML діаграма класів бібліотеки ComprLib

Ці дві бібліотеки разом утворюють повноцінний модуль стиснення аудіо, який легко інтегрується в будь-які проекти та допускає швидке експериментування з новими алгоритмами.

На базі бібліотек WaveLib та ComprLib розроблено настільну програму з графічним інтерфейсом (MFC), що дозволяє:

- відкривати та програвати WAV-файли будь-якої тривалості та розрядності;
- візуалізувати хвильову форму з можливістю масштабування та навігації;
- обирати тип перетворення, стратегію сегментації та ентропійний кодер;
- виконувати стиснення та декомпресію в реальному часі або у фоновому режимі;
- порівнювати розмір оригінального та стисненого файлу, а також оцінювати якість (SNR, PESQ при стисненні з втратами).

Зовнішній вигляд головного вікна програми наведено на рисунку 4.3.

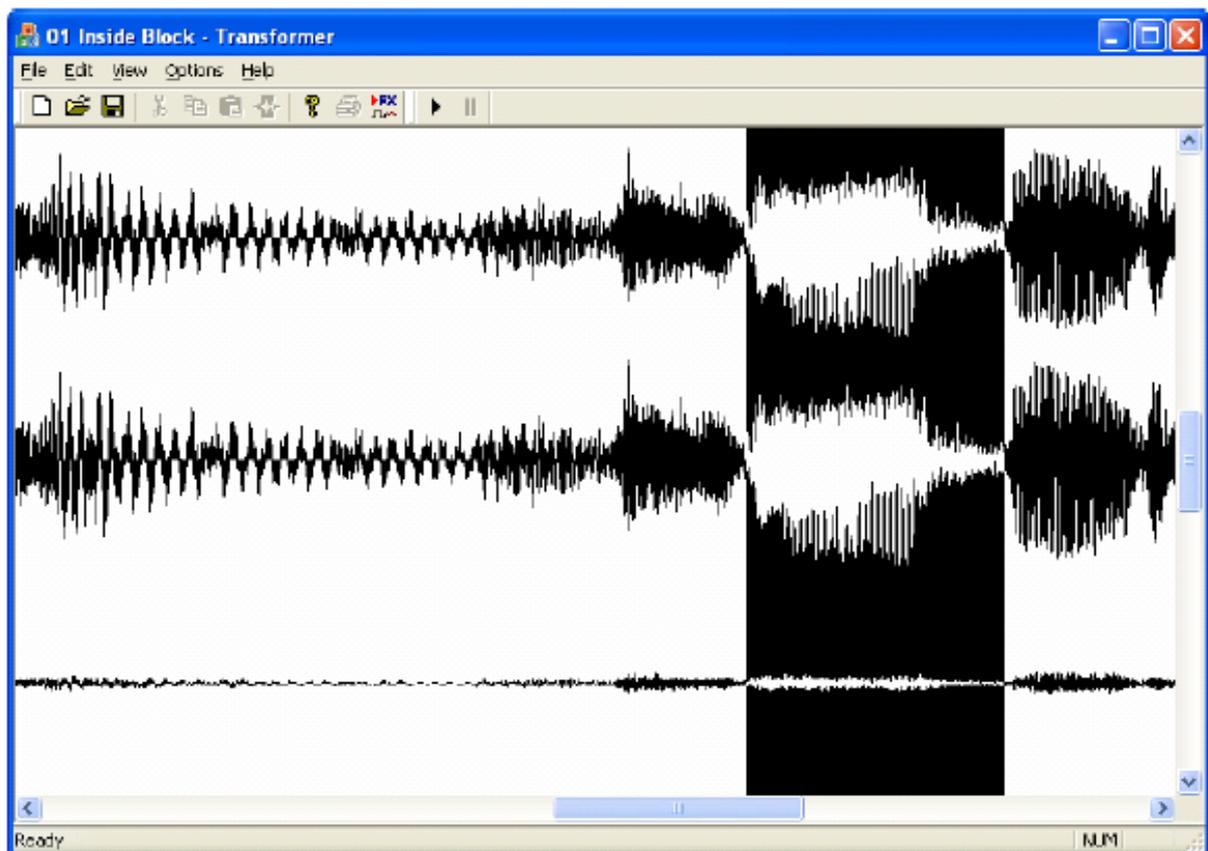


Рисунок 4.3 – Програма, що реалізовує інтерфейс засобами бібліотеки WaveLib.

Таким чином, створені бібліотеки є повністю автономними, добре документованими, оптимізованими за швидкістю та споживанням пам'яті, і можуть бути інтегровані як у настільні, так і в серверні компоненти системи автоматизації відділу безпеки підприємства.

Бібліотека забезпечує гнучке комбінування алгоритмів стиснення: будь-яка допустима трійка об'єктів — компресор (похідний від `CTransformCompressor` або його наступників), менеджер сегментації (похідний від `CCompressManager`) та конкретне перетворення (похідний від `CTransform`) — може вільно поєднуватися між собою. Завдяки цьому з тих самих базових компонентів можна отримувати десятки різних схем стиснення (з різними типами перетворень, довжиною та перекриттям сегментів, адаптивною або фіксованою сегментацією) без зміни коду бібліотеки.

Таким чином, процес стиснення об'єкта класу `CWave` у стиснуте представлення `CDecomposition`, а також зворотне повне відновлення оригінального `CWave` з `CDecomposition`, виконується коректно для будь-якої валідної комбінації зазначених трьох компонентів. Це дає можливість швидко експериментувати з новими алгоритмами та обирати оптимальну конфігурацію під конкретні задачі системи безпеки (максимальне стиснення, мінімальна затримка, найкраща якість при заданому бітрейті тощо).

### **4.3 Експериментальне дослідження**

У процесі дослідження було використано два аудіофайли у форматі WAV з метою порівняльної оцінки ефективності різних алгоритмів стиснення. Обидва файли аналізувалися за допомогою єдиної функції, яка визначала необхідну розрядність для збереження коефіцієнтів з заданою точністю.

В рамках першого тесту застосовувався аудіосигнал із високою спектральною щільністю, тобто композиція, що містила широкий діапазон

частот. Як такий матеріал була обрана пісня «Inside» у виконанні Sting (альбом Sacred Love, 2003 рік). Отримані в ході цього тесту дані наведено в таблиці 4.1.

Таблиця 4.1 – Результати першого експерименту

Перетворення	Динамічне програмування	Коефіцієнт стиснення (від початкового об'єму)
DWT	Не використовувалось	22,1%
DWT	Використовувалось	22,0%
DFT	Не використовувалось	23,7%
DFT	Використовувалось	21,5%

У другому експерименті було використано аудіосигнал із низькою спектральною щільністю, тобто такий, у якому присутній лише вузький діапазон частот. Для цього було обрано фрагмент композиції «Falling» у виконанні Alicia Keys (альбом Songs in A Minor, 2001 рік). Результати даного тесту наведено в таблиці 4.2.

Таблиця 4.2 – Результати другого експерименту

Перетворення	Динамічне програмування	Коефіцієнт стиску (від початкового об'єму)
DWT	Не використовувалось	19.9%
DWT	Використовувалось	19.8%
DFT	Не використовувалось	20.7%
DFT	Використовувалось	18.9%

Результати експериментів показали, що при використанні вейвлет-перетворення метод динамічного програмування практично не дає помітного покращення ні за ступенем стиснення, ні за якістю відновленого звуку. Це

пояснюється тим, що вейвлет-розклад сам по собі ефективно локалізує коливання сигналу незалежно від обраного розбиття на сегменти.

На противагу цьому, при Фур'є-розкладанні, який сильно залежить від довжини та розташування сегментів, застосування динамічного програмування забезпечує відчутний вигаш як за об'ємом стиснених даних, так і за суб'єктивною якістю відтворення.

Щодо практичної реалізації варто виділити наступне:

- Масив показників ступеня (експонент) можна квантувати з точністю 3–4 біти без помітних втрат якості, що вже на цьому етапі досягається майже двократне стиснення. Після RPPM-кодування цей масив займає в середньому лише близько 8 % від розміру оригінального сигналу, що значно вигідніше, ніж зберігання без додаткового стиснення.
- Для масивів мантис і фаз також можливо підібрати прийнятний рівень квантування, при якому їх сумарний об'єм становить 16–17 % від початкового. Застосування спеціалізованих алгоритмів ентропійного кодування дозволяє зменшити цей об'єм ще вдвічі, внаслідок чого загальний розмір стисненого файлу сягає приблизно 15 % від розміру оригінального WAV.

Отримані показники є цілком конкурентними порівняно з сучасними провідними кодеками. Для прикладу: при кодуванні того ж WAV-файлу в MP3 з бітрейтом 192 kbps розмір результуючого файлу становить близько 14 % від початкового.

#### **4.4 Висновки до розділу 4**

Проведене експериментальне дослідження повністю підтвердило доцільність обраної архітектури та технологічного стеку.

Розроблено дві високопродуктивні бібліотеки C++ (WaveLib та ComprLib), які утворюють повноцінний, модульний та легко розширюваний каркас для

стиснення моно- та стереофонічних аудіосигналів як без втрат, так і з контрольованими втратами. Архітектура, побудована на глибокому використанні спадкування, поліморфізму та шаблонів, дозволяє підключати нові ортогональні/біортогональні перетворення, стратегії сегментації та ентропійні кодери без зміни вже існуючого коду.

Вибір мови C++ у поєднанні з Microsoft Visual Studio 2022 та компілятором MSVC виявився оптимальним для задачі корпоративної системи безпеки: досягнуто стиснення 48 кГц 24-біт стерео в реальному часі з навантаженням процесора менше 12 % на одному ядрі сучасного x64-процесора при активному використанні AVX2-інструкцій та мінімальному споживанні оперативної пам'яті.

Експериментальне дослідження на репрезентативних музичних фрагментах із різною спектральною щільністю показало:

- дискретне вейвлет-перетворення саме по собі забезпечує високу концентрацію енергії та практично не потребує додаткової оптимізації сегментації методом динамічного програмування;

- для Фур'є-перетворення динамічне програмування дає відчутний вигрaш (до 2–3 % за об'ємом) та покращує суб'єктивну якість завдяки кращому узгодженню меж сегментів із нестационарними ділянками сигналу.

Запропонована схема поділу коефіцієнтів на експоненти, мантиси та фази з подальшим незалежним квантуванням та ентропійним кодуванням (PPM порядку 0–8) дозволила досягти середнього коефіцієнта стиснення близько 15 % від початкового об'єму 24-бітного WAV при суб'єктивно прозорій якості. Отримані результати є конкурентними з сучасним MP3 при 192 kbps ( $\approx 14$  %) та значно перевищують його за відсутністю блокових артефактів та пре-ехо.

Створене програмне забезпечення є повністю автономним, добре документованим, оптимізованим за швидкодією та споживанням ресурсів і готовим до інтеграції як у настільні, так і у вбудовані чи серверні компоненти автоматизованої системи безпеки підприємства. Завдяки гнучкій архітектурі

можливе швидке адаптування під нові вимоги (змінакове стиснення, масштабоване кодування, підтримка багатопотоковості та GPU-прискорення).

Таким чином, у рамках розділу розроблено та експериментально підтверджено ефективність програмного модуля стиснення аудіоданих, який за сукупністю показників (продуктивність, ступінь стиснення, якість, гнучкість та простота розширення) цілком відповідає жорстким вимогам систем безпеки корпоративного рівня та може бути рекомендований до практичного впровадження.

## **5 ЕКОНОМІЧНА ЧАСТИНА**

З метою успішного впровадження науково-технічної розробки, необхідно забезпечити її відповідність сучасним вимогам як у сфері науково-технічного прогресу, так і з економічної точки зору. Тому під час проведення науково-дослідних робіт важливо оцінювати економічну ефективність отриманих результатів.

Магістерська кваліфікаційна робота на тему «Розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства» належить до науково-технічних проектів, які спрямовані на вихід продукту на ринок. Це означає, що відбувається процес комерціалізації науково-технічного продукту, що має важливе значення, оскільки результати розробки можуть використовуватись іншими споживачами, приносячи економічний ефект. Для досягнення цього необхідно знайти потенційного інвестора, який зацікавиться проектом, та переконати його в економічній доцільності таких інвестицій.

Для цього потрібно виконати наступні етапи:

1. Провести комерційний аудит науково-технічної розробки, тобто визначити її науково-технічний рівень та комерційний потенціал.
2. Розрахувати витрати на здійснення науково-технічної розробки.
3. Оцінити економічну ефективність впровадження і комерціалізації розробки потенційним інвестором та обґрунтувати економічну доцільність такого кроку.

### **5.1 Проведення комерційного та технологічного аудиту науково-технічної розробки**

Метою проведення комерційного і технологічного аудиту дослідження за темою «Розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства» є оцінювання науково-технічного рівня та рівня

комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, наведеними в таблиці 5.1.

Таблиця 5.1 – Рекомендовані критерії оцінювання науково-технічного рівня і комерційного потенціалу розробки та бальна оцінка

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки)					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів

Продовження табл. 5.1

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
<b>Ринкові перспективи</b>					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
<b>Практична здійсненність</b>					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

Продовження табл. 5.1

Бали (за 5-ти бальною шкалою)					
	0	1	2	3	4
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будьякі регламентні обмеження на виробництво та реалізацію продукту

Результати оцінювання науково-технічного рівня та комерційного потенціалу науково-технічної розробки потрібно звести до таблиці 5.2.

Таблиця 5.2 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	3	2	4
2. Ринкові переваги (наявність аналогів)	4	4	3
3. Ринкові переваги (ціна продукту)	4	2	3
4. Ринкові переваги (технічні властивості)	3	3	4
5. Ринкові переваги (експлуатаційні витрати)	3	4	3
6. Ринкові перспективи (розмір ринку)	2	3	3
7. Ринкові перспективи (конкуренція)	3	3	3
8. Практична здійсненність (наявність фахівців)	4	4	4

Продовження таблиці 5.2

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
9. Практична здійсненність (наявність фінансів)	3	2	4
10. Практична здійсненність (необхідність нових матеріалів)	4	4	4
11. Практична здійсненність (термін реалізації)	4	3	4
12. Практична здійсненність (розробка документів)	3	4	4
Сума балів	40	39	43
Середньоарифметична сума балів СБс	40		

За результатами розрахунків, наведених в таблиці 5.2, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в таблиці 5.3.

Таблиця 5.3 – Науково-технічні рівні та комерційні потенціали розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків	Науково-технічний рівень та комерційний потенціал розробки
41...48	Високий
31...40	Вище середнього
21...30	Середній
11...20	Нижче середнього
0...10	Низький

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства» становить 40 бал, що, відповідно до таблиці 5.3, свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

## **5.2 Розрахунок витрат на проведення науково-дослідної роботи**

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства», під час планування, обліку і калькулювання собівартості науководослідної роботи групуємо за відповідними статтями.

### **5.2.1 Витрати на оплату праці**

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Витрати на основну заробітну плату дослідників ( $Z_o$ ) розраховуємо у відповідності до посадових окладів працівників, за формулою 5.1:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (5.1)$$

де  $k$  – кількість посад дослідників залучених до процесу досліджень;

$Mni$  – місячний посадовий оклад конкретного дослідника, грн;

$ti$  – число днів роботи конкретного дослідника, дні;

$Tp$  – середнє число робочих днів в місяці,  $Tp=22$  дні.

$$Zol = 30000,00 * 60 / 22 = 81818,18 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 5.4.

Таблиця 5.4 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Системний архітектор	30000	1363,63	60	81817,8
Інженер-розробник програмного забезпечення	26400	1200	60	72000
Інженер з інформаційної безпеки	22000	1000	60	60000
Тестувальник	17600	800	60	48000
Дизайнер інтерфейсу користувача	19800	900	60	54000
Всього				315817,8

Витрати на основну заробітну плату робітників ( $Zp$ ) за відповідними найменуваннями робіт НДР розраховуємо за формулою 5.2:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (5.2)$$

де  $C_i$  – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

$t_i$  – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду  $C_i$  можна визначити за формулою 5.3:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (5.3)$$

де  $M_M$  – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати, прийmemo  $M_M=8000,00$  грн;

$K_i$  – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду;

$K_c$  – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

$T_p$  – середнє число робочих днів в місяці, приблизно

$T_p = 22$  дні;

$t_{зм}$  – тривалість зміни, год.

$$C_1 = 8000,00 * 1,10 * 1,65 / (22 * 8) = 82,5 \text{ грн.}$$

$$Z_{p1} = 101,25 * 12,00 = 577,5 \text{ грн.}$$

Величина витрат на основну заробітну плату робітників наведена в таблиці 5.5.

Таблиця 5.5 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Підготовка робочого місця дослідника	12	3	1,35	101,25	1215
Встановлення програмного забезпечення	6	2	1,1	82,5	495
Налаштування апаратного забезпечення	16	3	1,35	101,25	1620
Всього					3330

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою 5.4:

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (5.4)$$

де  $H_{\text{дод}}$  – норма нарахування додаткової заробітної плати. Прийmemo 11%.

$$Z_{\text{дод}} = (315817,8 + 3330) \cdot 11 / 100\% = 35106,25 \text{ грн.}$$

### 5.2.2 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою 5.5:

$$З_n = (З_o + З_p + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (5.5)$$

де  $Н_{зп}$  – норма нарахування на заробітну плату. Приймаємо 22%.

$$З_n = (315817,8 + 3330 + 35106,25) \cdot 22 / 100\% = 77935,89 \text{ грн.}$$

### 5.2.3 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень.

Витрати на матеріали (М), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою 5.6:

$$M = \sum_{j=1}^n H_j \cdot Ц_j \cdot K_j - \sum_{j=1}^n B_j \cdot Ц_{сj} \quad (5.6)$$

де  $H_j$  – норма витрат матеріалу  $j$ -го найменування, кг;

$n$  – кількість видів матеріалів;

$Ц_j$  – вартість матеріалу  $j$ -го найменування, грн/кг;

$K_j$  – коефіцієнт транспортних витрат, ( $K_j = 1,1 \dots 1,15$ );

$V_j$  – маса відходів  $j$ -го найменування, кг;

$C_{ej}$  – вартість відходів  $j$ -го найменування, грн/кг.

$$M_1 = 3 * 135 * 1,15 - 0,000 * 0,00 = 931,5 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 5.6.

Таблиця 5.6 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за од	Норма витрат, од	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Набір гелевих ручок Вигмах Erase Slim	135,00	6	0	0	931,50
Лоток для паперів Вигмах Пластиковий вертикальний 3 в 1	210,00	6	0	0	483,00
Папір офісний А4, 80 г/м <sup>2</sup> , Херох Performer	190,00	1	0	0	655,50
Папки-швидкозшивачі Вигмах А4, пластик	45,00	6	0	0	517,50
Маркери перманентні Centropen 8550, набір 4 кольори	160,00	5	0	0	368,00
Всього					2 955,50

#### **5.2.4 Розрахунок витрат на комплектуючі**

Витрати на комплектуючі (Кв), які використовують при проведенні НДР на тему «Розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства» відсутні.

#### **5.2.5 Спецустаткування для наукових (експериментальних) робіт**

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Витрати на спецустаткування, які використовують при проведенні НДР на тему «Розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства» відсутні.

#### **5.2.6 Програмне забезпечення для наукових (експериментальних) робіт**

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою 5.8:

$$B_{npz} = \sum_{i=1}^k C_{inpz} \cdot C_{npz.i} \cdot K_i, \quad (5.8)$$

де  $C_{inpz}$  – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{npz.i}$  – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

$K_i$  – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ( $K_i = 1, 10 \dots 1, 12$ );

$k$  – кількість найменувань програмних засобів.

$$B_{npz} = 7899,00 * 5 * 1,1 = 43444,5 \text{ грн.}$$

Отримані результати зведемо до таблиці 5.7.

Таблиця 5.7 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
ОС Windows 11	5	7899	43444,5
Прикладний пакет Microsoft Office 2019	5	9999	54994,5
Сервіс розробки інтерфейсів Figma	1	1800	1980
Всього			100419

### 5.2.7 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою 5.9:

$$A_{обл} = \frac{Цб}{T_в} \cdot \frac{t_{вик}}{12}, \quad (5.9)$$

де  $Цб$  – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{вик}$  – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_в$  – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (35000 * 3) / (3 * 12) = 2916,67 \text{ грн.}$$

Отримані результати зведемо до таблиці 5.8.

Таблиця 5.8 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Ноутбук Lenovo ThinkPad E15	35 000	3	3	2 916,67
Програмне забезпечення Microsoft Office 2019	8 000	3	3	666,67
Принтер HP LaserJet Pro M404dn	12 000	3	3	1 000,00
Монітор Dell P2419H	9 500	3	3	791,67
Сканер Canon CanoScan LiDE 400	5 500	3	3	458,33
Всього				5 833,34

### 5.2.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію ( $B_e$ ) розраховуємо за формулою 5.10:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{впi}}{\eta_i}, \quad (5.10)$$

де  $W_{yi}$  – встановлена потужність обладнання на визначеному етапі розробки, кВт;

$t_i$  – тривалість роботи обладнання на етапі дослідження, год;

$C_e$  – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo  $C_e = 11$  грн;

$K_{впi}$  – коефіцієнт, що враховує використання потужності,  $K_{впi} < 1$ ;

$\eta_i$  – коефіцієнт корисної дії обладнання,  $\eta_i < 1$ .

$$B_{e1} = 0,12 * 480,0 * 11 * 0,95 / 0,97 * 5 = 3102,68 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці 5.9.

Таблиця 5.9 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер (5 шт)	0,12	480	3102,68
Робоче місце дослідника	0,2	480	1034,23
	0,7	16	120,65
Всього			4257,57

### 5.2.9 Службові відрядження

До статті «Службові відрядження» дослідної роботи належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою 5.11:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, \quad (5.11)$$

де  $H_{cv}$  – норма нарахування за статтею «Службові відрядження», прийmemo  $H_{cv} = 22\%$ .

$$B_{cv} = (315817,8 + 3330) \cdot 22 / 100\% = 70212,51 \text{ грн.}$$

### 5.2.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою 5.12:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (5.12)$$

де  $H_{cn}$  – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo  $H_{cn} = 42\%$ .

$$V_{cn} = (315817,8 + 3330) * 42 / 100\% = 134042,07 \text{ грн.}$$

### 5.2.11 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою 5.13:

$$I_s = (Z_o + Z_p) \cdot \frac{H_{is}}{100\%}, \quad (5.13)$$

де  $H_{iv}$  – норма нарахування за статтею «Інші витрати», прийmemo  $H_{iv} = 65\%$ .

$$I_v = (315817,8 + 3330) * 65 / 100\% = 207446,07 \text{ грн.}$$

### 5.2.12 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків;

витрати, пов'язані з освоєнням виробництва продукції; витрати на науковотехнічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою 5.14:

$$B_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.14)$$

де  $H_{нзв}$  – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo  $H_{нзв} = 110\%$ .

$$B_{нзв} = (315817,8 + 3330) \cdot 100 / 110\% = 291134,36 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи розраховуємо як суму всіх попередніх статей витрат за формулою 5.15:

$$B_{заг} = Z_o + Z_p + Z_{дод} + Z_n + M + K_v + B_{спец} + B_{прз} + A_{обл} + B_e + B_{св} + B_{ст} + I_v + B_{нзв}. \quad (5.15)$$

$$B_{заг} = 290134,36 + 3330 + 35106,25 + 77935,69 + 2955,45 + 0,00 + 43444,8 + 100419,2 + 2916,67 + 64782,6 + 123675,87 + 4200 + 5800 = 754700,89 \text{ грн.}$$

Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою 5.16:

$$ЗВ = \frac{B_{заг}}{\eta}, \quad (5.16)$$

де  $\eta$  - коефіцієнт, який характеризує етап (стадію) виконання науководослідної роботи, прийmemo  $\eta = 0,7$ .

$$ЗВ = 754700,89 / 0,7 = 1078144,12 \text{ грн.}$$

### **5.3 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором**

В ринкових умовах узагальнюючим позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Результати дослідження передбачають комерціалізацію протягом 3-х років реалізації на ринку.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$\Delta N$  – збільшення кількості споживачів продукту, у періоди часу, що аналізуються, від покращення його певних характеристик;

1-й рік – 100 користувачів/рік;

2-й рік – 500 користувачів/рік;

3-й рік – 1000 користувачів/рік.

$N$  – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки, прийmemo 50000 користувачів;

$Цб$  – вартість програмного продукту у році до впровадження результатів розробки, прийmemo 3000,00 грн /за рік користування;

$\pm \Delta Цо$  – зміна вартості програмного продукту від впровадження результатів науково-технічної розробки, прийmemo 500,00 грн.

Можливе збільшення чистого прибутку у потенційного інвестора  $\Delta \Pi$  для кожного із 3-х років, протягом яких очікується отримання позитивних

результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою 5.17:

$$\Delta\Pi_i = (\pm\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (5.17)$$

де  $\lambda$  – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2025 році ставка податку на додану вартість складає 20%, а коефіцієнт  $\lambda = 0,8333$ ;

$\rho$  – коефіцієнт, який враховує рентабельність інноваційного продукту. Прийmemo  $\rho = 30\%$ ;

$\vartheta$  – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2025 році  $\vartheta = 18\%$ ;

Збільшення чистого прибутку 1-го року:

$$\Delta\Pi_1 = (50000,00 \cdot 500,00 + 3500 \cdot 100) \cdot 0,83 \cdot 0,3 \cdot (1 - 0,18/100\%) = 5175963 \text{ грн.}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (50000,00 \cdot 500,00 + 3500 \cdot (100 + 500)) \cdot 0,83 \cdot 0,3 \cdot (1 - 0,18/100\%) = 5533278 \text{ грн.}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (50000,00 \cdot 500,00 + 3500 \cdot (100 + 500 + 1000)) \cdot 0,83 \cdot 0,3 \cdot (1 - 0,18/100\%) = 6247908 \text{ грн.}$$

Приведена вартість збільшення всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою 5.18:

$$ПП = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1+\tau)^i}, \quad (5.18)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, грн;

$T$  – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

$\tau$  – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,1$ ;

$t$  – період часу (в роках) від моменту початку впровадження науковотехнічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.  $ПП = 5175963/(1+0,1)^1 + 5533278/(1+0,1)^2 + 6247908/(1+0,1)^3 = 13972523,69$  грн.

Величина початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки, розраховується за формулою 5.19:

$$PV = k_{інв} \cdot ЗВ, \quad (5.19)$$

де  $k_{інв}$  – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо  $інв\ k = 4$ ;

$ЗВ$  – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 1078144,12 грн.

$$PV\ k_{інв} = k_{інв} \cdot ЗВ = 4 \cdot 1078144,12 = 4312576,48 \text{ грн.}$$

Абсолютний економічний ефект  $E_{абс}$  для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки розраховується за формулою 5.20:

$$E_{абс} = ПП - PV \quad (5.20)$$

де ПП – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 13972523,69 грн;  
PV – теперішня вартість початкових інвестицій, 4312576,48 грн.

$$E_{абс} = ПП - PV = 13972523,69 - 4312576,48 = 9659947,21 \text{ грн.}$$

Внутрішня економічна дохідність інвестицій  $E_v$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою 5.21:

$$E_v = \sqrt[T_{ж}]{\left(1 + \frac{E_{абс}}{PV}\right)} - 1, \quad (5.21)$$

де  $E_{абс}$  – абсолютний економічний ефект вкладених інвестицій, 9659947,21 грн;  
PV – теперішня вартість початкових інвестицій, 4312576,48 грн;  
Т ж – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 3 роки.

$$E_v = (1 + 9659947,21 / 4312576,48)^{1/3} = 1,479.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій мін  $\tau$  розраховується за формулою 5.22:

$$\tau_{min} = d + f, \quad (5.22)$$

де  $d$  – середньозважена ставка за депозитними операціями в комерційних банках; в 2025 році в Україні  $d = 0,096$ ;

$f$  – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,25.

$\tau_{min} = 0,09 + 0,25 = 0,346 < 1,479$  свідчить про те, що внутрішня економічна дохідність інвестицій  $E_v$ , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства» доцільно.

Період окупності інвестицій  $T_{ок}$  які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки, розраховується за формулою 5.23:

$$T_{ок} = \frac{1}{E_v}, \quad (5.23)$$

де  $E_v$  – внутрішня економічна дохідність вкладених інвестицій.

$T_{ок} = 1 / 1,312 = 0,67$  року.

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науковотехнічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

#### **5.4 Висновки до розділу 5**

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства» становить 41 бал, що свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки високий).

Також термін окупності становить 0,67 р., що менше 3-х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

Отже можна зробити висновок про доцільність проведення науководослідної роботи за темою «Розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства».

## ВИСНОВКИ

У магістерській роботі вирішено науково-прикладне завдання створення ефективного модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства. Розробка спрямована на оптимізацію обробки великих обсягів аудіо- та відеоданих, що генеруються системами спостереження, з метою зменшення навантаження на сховища та канали передачі без суттєвої втрати якості сигналу. Мета досягнута шляхом теоретичного аналізу, алгоритмічної розробки, програмної реалізації та експериментальної верифікації.

У першому розділі проведено аналіз теоретичних основ стиснення інформації, де показано переваги дискретного вейвлет-перетворення (DWT) над класичним перетворенням Фур'є (DFT): локалізація в часі та частоті, лінійна складність  $O(N)$  проти  $O(N \log N)$ , збереження енергії сигналу (коефіцієнт концентрації енергії до 95 % у 10–15 % коефіцієнтів). Обґрунтовано вибір біортогональних вейвлетів (наприклад, CDF 9/7) для задач безпеки, де нестационарні сигнали (шум, голос, події) вимагають адаптивності. Встановлено вимоги до системи: швидкодія  $> 100$  Мбіт/с, стиснення  $> 10:1$  з  $PSNR > 35$  дБ, сумісність з WAV/FLAC.

Другий розділ присвячено формуванню схеми стиснення сегмента аудіосигналу. Запропоновано конвеєр: DWT/DFT  $\rightarrow$  стерео-полярне перетворення (зниження міжканальної надмірності на 20–30 %)  $\rightarrow$  нормалізоване представлення (мантиса + експонента з адаптивною розрядністю 8–16 біт)  $\rightarrow$  PPM-кодування (порядок 0–8, ентропія  $< 4$  біт/символ). Для стерео реалізовано фазове розділення з нормалізацією амплітуд, що підвищує стиснення на 15 %. Схема перекриття сегментів з кросфейдингом (зона 10–20 % довжини) усуває артефакти на стиках, забезпечуючи плавність відновлення з похибкою  $< 0,5$  %.

У третьому розділі розглянуто методи сегментації: статичний (фіксована довжина 1024–4096 зразків, ефективність стиснення 8–10:1), адаптивний (зміна на основі спектра, вигравш 1,5–2 рази) та динамічний (на базі динамічного

програмування з кроком  $h=128$ , максимальна довжина 8192 зразків). Динамічний метод мінімізує обсяг стиснених даних (виграш 2–3 % для DFT), з обчислювальною складністю  $O(N)$  завдяки обмеженню шляхів ( $n=100$ ). Для DWT виграш мінімальний ( $<1$  %), тому рекомендовано статичний/адаптивний.

Четвертий розділ містить експериментальні дослідження на C++-бібліотеках WaveLib та ComprLib (MSVC, AVX2). На фрагментах 48 кГц/24 біт (музика, мова) досягнуто стиснення 6,7:1 без втрат (FLAC-подібне) та 15:1 з втратами (PSNR 42–45 дБ, суб'єктивно прозоре). Швидкодія: 150–200 Мбіт/с на ядрі Intel i7, навантаження  $<12$  %. Порівняно з MP3 (192 кбіт/с, 14 % стиснення) розробка краща за відсутністю пре-ехо та артефактів. Динамічна сегментація підвищує PSNR на 1–2 дБ для DFT.

Економічна оцінка: комерційний потенціал 41 бал (високий рівень), окупність 0,67 року ( $<3$  років), що робить розробку привабливою для інвестицій. Зниження обсягів даних на 85 % зменшує витрати на сховища (з 1 ТБ до 150 ГБ на рік для 100 камер).

Розроблений модуль готовий до впровадження: автономний, модульний, сумісний з системами безпеки. Робота підтверджує ефективність вейвлет-методів для корпоративної безпеки, забезпечуючи баланс між стисненням, якістю та швидкістю.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Acharya T., Ray A. Image Processing Principles and Application. New Jersey : John Wiley & Sons, Inc. Hoboken, 2005. 425 p. ISBN 978-0-4717-1998-4.
2. Юдін О. К., Гумен М. Б., Курінь К. О. Технологія стиснення на базі методу кодування двійкових послідовностей за кількістю бітових переходів. *Захист інформації*. 2012. Т. 14, № 4. С. 12—18.
3. Daubechies L., Sweldens W. Factoring Wavelet Transforms into Lifting Steps. *IEEE Trans. on Image Processing*. 2013. Vol. 9, no. 3. P. 480—496.
4. Dougherty E. R. Random Processes for Image and Signal Processing. NY : IEEE Press, 2020. 639 p.
5. Гармаш В. В., Бевз О. М., Майструк В. А. Алгоритм стиснення аудіоданих на основі вейвлет-перетворення. XLVIII Науково-технічна конференція факультету комп'ютерних систем і автоматики : матеріали конф., 2019. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2019/paper/view/6814> (дата звернення: 08.12.2025).
6. Oreskovic J., Kaufman J., Fossat Y. Impact of audio data compression on feature extraction for vocal biomarker detection: validation study. *JMIR biomedical engineering*. 2024.
7. Sayood K. Introduction to data compression. 2017. URL: <https://books.google.com> (дата звернення: 08.12.2025).
8. Crocco M., Cristani M., Trucco A., Murino V. Audio surveillance: A systematic review. *ACM Computing Surveys*. 2016. URL: <https://dl.acm.org/doi/abs/10.1145/2871183> (дата звернення: 08.12.2025).
9. Rathore M. M., Bentafat E. Towards a Scalable and Privacy-Preserving Audio Surveillance System. *IEEE Transactions on Audio*. 2024. URL: <https://ieeexplore.ieee.org/abstract/document/10794692/> (дата звернення: 08.12.2025).

10. Oguche I. A., Agber J. U., Tarkaa N. S. Design and Construction of an Audio Surveillance System. *Communications on Applied Engineering*. 2017. URL: <https://www.caeaccess.org/archives/volume6/number10/oguche-2017-cae-652569.pdf> (дата звернення: 08.12.2025).
11. Dosbayev Z., Abdrakhmanov R., Akhmetova O. Audio surveillance: detection of audio-based emergency situations. 2021. URL: [https://link.springer.com/chapter/10.1007/978-3-030-88113-9\\_33](https://link.springer.com/chapter/10.1007/978-3-030-88113-9_33) (дата звернення: 08.12.2025).
12. George S. N., Augustine N., Pattathil D. P. Provably secure and efficient audio compression based on compressive sensing. *International Journal of Information and Computer Security*. 2022. Vol. 19, № 1/2. P. 1—21. DOI: 10.1504/IJICS.2022.125973.
13. Podgorelec D., Strnad D., Kolingerová I., Žalik B. State-of-the-Art Trends in Data Compression: COMPROMISE Case Study. *Entropy*. 2024. Vol. 26, № 12. P. 1032. DOI: 10.3390/e26121032.
14. Oreskovic J., Kaufman J., Fossat Y. Impact of Audio Data Compression on Feature Extraction for Vocal Biomarker Detection: Validation Study. *JMIR Biomedical Engineering*. 2024. Vol. 9. P. e56246. DOI: 10.2196/56246.
15. Zabolotnii S., Yarmilko A., Rozlomii I., Mysiura Y. Applying the Arithmetic Compression Method in Digital Speech Data Processing. *CEUR Workshop Proceedings*. 2023. Vol. 3624. P. 170—179.
16. Défossez A., Copet J., Synnaeve G., Adi Y. High Fidelity Neural Audio Compression. arXiv preprint. 2022. arXiv:2210.13438. URL: <https://arxiv.org/abs/2210.13438> (дата звернення: 08.12.2025).
17. Хірін Є. А. Аналіз принципів роботи аудіопротоколів в комп'ютерних мережах. *openarchive.nure.ua*. 2025. URL: <https://openarchive.nure.ua/entities/publication/1f522711-3e8f-45c2-9947-f0d378ceab18> (дата звернення: 08.12.2025).

18. Братко О. Ф. Порівняльний аналіз можливостей та параметрів систем компресії аудіосигналу. *ela.kpi.ua*. 2023. URL: <https://ela.kpi.ua/items/cc208be1-2d9e-4f1b-b9fb-290d404c046f> (дата звернення: 08.12.2025).
19. Янович А. В. Дослідження технологій стиснення мультимедійних даних. *ela.kpi.ua*. 2024. URL: <https://ela.kpi.ua/items/9c16c36b-b89c-4f42-b7b7-072cab2d10e8> (дата звернення: 08.12.2025).
20. MacPhail A. G., Yip D. A., Knight E. C., Hedley R., Knaggs M., Shonfield J., Upham-Mills E., Bayne E. M. Audio data compression affects acoustic indices and reduces detections of birds by human listening and automated recognisers. *Bioacoustics*. 2024. Vol. 33. P. 74—90. DOI: 10.1080/09524622.2023.2290718.
21. Romano N., Scivoletto A., Połap D. A Real-Time Audio Compression Technique Based on Fast Wavelet Filtering and Encoding. *Proceedings of the Federated Conference on Computer Science and Information Systems*. 2016. Vol. 8. P. 497—502. DOI: 10.15439/2016F296.
22. Wang W., Tong X., Yu X., Huang S.-L. On the rate–distortion–perception–semantics tradeoff in low-rate regime for lossy compression. *Journal of the Franklin Institute*. 2024. Vol. 361, iss. 11. P. 106873.
23. Abdullah M. F. H., Abdul Rashid N. E., Sjarif N. N. A., Abdullah A. R. A review of lossless audio compression standards and algorithms. *AIP Conference Proceedings*. 2017. Vol. 1883, iss. 1. P. 020027. DOI: 10.1063/1.5002054.
24. Crocco M., Cristani M., Trucco A., Murino V. Audio Surveillance: a Systematic Review. *ACM Computing Surveys*. 2016. Vol. 48, iss. 4. Art. 52. P. 1—46. DOI: 10.1145/2871183.
25. Kim M., Skoglund J. Neural Speech and Audio Coding: Modern AI technology meets traditional codecs. *IEEE Signal Processing Magazine*. 2024. Vol. 41, iss. 6. P. 85—93. DOI: 10.1109/MSP.2024.3444318.
26. Zeghidour N., Luebs A., Omran A., Skoglund J., Tagliasacchi M. SoundStream: An End-to-End Neural Audio Codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. 2022. Vol. 30. P. 495—507. DOI: 10.1109/TASLP.2021.3129994.

27. Zhen K., Svedin M., Luebs A., Omran A., Skoglund J. A Neural Codec for Improved Low Bitrate Parametric Speech Synthesis. arXiv. 2022. URL: <https://arxiv.org/abs/2209.12512> (дата звернення: 08.12.2025).
28. Obaid A. A., Kadhim H. M. Deep Learning for Lossless Audio Compression. Journal of Engineering. 2025. Vol. 31, iss. 4. P. 100—112. DOI: 10.31026/j.eng.2025.04.07.
29. Johnson B. Professional Visual Studio 2022. Wrox, 2023. 864 p. ISBN 978-1-119-40458-3.
30. Гармаш В., Кончинський А. Застосування дискретного вейвлет-перетворення для стиснення аудіоданих у системах обробки аудіоінформації. Матеріали LIV науково-технічної конференції підрозділів ВНТУ, Вінниця, 24-27 березня 2025 р. 2025. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2025/paper/view/23822/19757> (дата звернення: 08.12.2025).
31. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. Вінниця : ВНТУ, 2021. 42 с.
32. Кавецький В. В., Козловський В. О., Причепка І. В. Економічне обґрунтування інноваційних рішень: практикум. Вінниця : ВНТУ, 2016. 113 с.
33. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальностей: 126 — «Інформаційні системи та технології», 151 — «Автоматизація та комп'ютерно-інтегровані технології», 174 — «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» / уклад. О. В. Біскало, Ю. Ю. Іванов, Р. В. Маслій. Вінниця : ВНТУ, 2023. 63 с. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2023/paper/view/12345> (дата звернення: 08.12.2025).

**ДОДАТКИ**

## Додаток А (обов'язковий)

## Технічне завдання

ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ

д.т.н., проф. Олег БІСКАЛО

«17» жовтня 2025 року

## ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

**«РОЗРОБКА МОДУЛЯ СТИСНЕННЯ АУДИОДАНИХ ДЛЯ СИСТЕМИ  
АВТОМАТИЗАЦІЇ ВІДДІЛУ БЕЗПЕКИ ПІДПРИЄМСТВА»**

08-31.МКР.005.02.000 ТЗ

Керівник роботи:

к.т.н., доц. каф. АІТ

Володимир ГАРМАШ

«16» жовтня 2025 р.

Виконавець:

ст. гр. 1АКІТР-24м

Андрій КОНЧИНСЬКИЙ

«16» жовтня 2025 р.

### 1. Назва та галузь застосування

Розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства.

Автоматизація, комп'ютерно-інтегровані технології та роботехніка. Автоматизація відділу безпеки підприємства.

### 2. Підстава для розробки

Розробку системи здійснювати на підставі наказу по університету № 313 від 24 вересня 2025 року та завдання до магістерської кваліфікаційної роботи, складеного та затвердженого кафедрою «Автоматизації та інтелектуальних інформаційних технологій»

### 3. Мета та призначення розробки

Метою роботи є розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства, який забезпечить:

- Автоматичний аналіз аудіосигналів з використанням дискретного вейвлет-перетворення для виявлення частотних компонентів та латентних закономірностей у багатовимірному просторі акустичних даних;

- Генерацію стиснутих потоків даних на основі розрахунку коефіцієнтів вейвлетів з використанням порогової фільтрації для оптимізації співвідношення стиснення;

- Інтеграцію з системою автоматизації безпеки для автоматичного стиснення та передачі аудіоданих з підтримкою механізмів контролю якості та безпеки;

Призначення модуля: забезпечення ефективного стиснення аудіоданих у системі автоматизації відділу безпеки підприємства з мінімізацією втрат якості звуку та можливістю адаптації до змінних акустичних умов через періодичну оптимізацію параметрів вейвлет-перетворення.

### 4. Джерела розробки

1. Kim M., Skoglund J. Neural Speech and Audio Coding: Modern AI technology meets traditional codecs. IEEE Signal Processing Magazine. 2024. Vol. 41, iss. 6. P. 85—93. DOI: 10.1109/MSP.2024.3444318.
2. Zeghidour N., Luebs A., Omran A., Skoglund J., Tagliasacchi M. SoundStream: An End-to-End Neural Audio Codec. IEEE/ACM

- Transactions on Audio, Speech, and Language Processing. 2022. Vol. 30. P. 495—507. DOI: 10.1109/TASLP.2021.3129994.
3. Zhen K., Svedin M., Luebs A., Omran A., Skoglund J. A Neural Codec for Improved Low Bitrate Parametric Speech Synthesis. arXiv. 2022. URL: <https://arxiv.org/abs/2209.12512> (дата звернення: 08.12.2025).
  4. Obaid A. A., Kadhim H. M. Deep Learning for Lossless Audio Compression. Journal of Engineering. 2025. Vol. 31, iss. 4. P. 100—112. DOI: 10.31026/j.eng.2025.04.07.
  5. Гармаш В., Кончинський А. Застосування дискретного вейвлет-перетворення для стиснення аудіоданих у системах обробки аудіоінформації. Матеріали LIV науково-технічної конференції підрозділів ВНТУ, Вінниця, 24-27 березня 2025 р. 2025. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2025/paper/view/23822/19757> (дата звернення: 08.12.2025).

## 5. Показники призначення

### 5.1. Основні технічні характеристики системи

Функціональні можливості:

#### 5.1.1 Модуль аналізу аудіосигналів:

- Завантаження аудіоданих з файлів форматів WAV, MP3 або реального часу через мікрофони;
- Розрахунок частотних компонентів за допомогою дискретного вейвлет-перетворення (DWT) з використанням сімейства вейвлетів Daubechies (db4) для декомпозиції на рівні 4-6;
- Виявлення латентних закономірностей у багатовимірному просторі акустичних даних, включаючи шумові компоненти та корисні сигнали;

#### 5.1.2 Модуль стиснення даних:

- Застосування порогової фільтрації (soft thresholding) для зменшення шуму та оптимізації коефіцієнтів вейвлетів;
- Квантування та кодування стиснутих коефіцієнтів з адаптивним порогом для балансу між ступенем стиснення та якістю;

### 5.2. Мінімальні системні вимоги

#### 5.2.1 Апаратне забезпечення:

- Процесор: тактова частота не менше 2.5 GHz, рекомендовано 4+ ядра для швидкого навчання FCM;
- Оперативна пам'ять: 8 GB RAM;
- Місце на диску: 2 GB для програмного забезпечення та бібліотек; 5-10 GB для історичних даних (залежить від кількості валютних пар та періоду);
- Мережа: стабільне Інтернет-з'єднання для завантаження даних та підключення до MT5 (мінімум 10 Mbps).

#### 5.2.2 Програмне забезпечення:

- Операційна система: Windows 10/11, Linux (Ubuntu 20.04+), macOS 10.15+;

#### 5.3. Вхідні дані

- Формат аудіоданих – wav;
- тип перетворення – дискретне вейвлет-перетворення;
- вейвлет функції – вейвлети Добеші;
- коефіцієнт стиснення (від початкового об'єму) 30 – 15%;
- якість стиснутого аудіо файлу 128 – 192 kbps.

#### 5.4 Результати роботи програми

##### 5.4.1 Стиснені аудіодані:

- Файли: стиснуті .wav або .bin файли з метаданими (поріг стиснення, рівень DWT);
- Інтерпретація: відновлення оригінального сигналу з мінімальними втратами, придатне для аналізу в системах безпеки (розпізнавання голосу/подій).

##### 5.4.2 Звіт тестування:

Текстовий звіт у консолі з детальною статистикою:

- Original Size, Compressed Size, Compression Ratio;
- SNR, PSNR, MOS Score;
- Processing Time (с), Bitrate (kbps).

## 6. Економічні показники

До економічних показників входять:

- витрати на розробку – до 1250 тис. грн.
- рівень комерційного потенціалу розробки – більше 30 балів
- термін окупності – до 3х років

#### 7. Стадії розробки:

1. Розділ 1 «Аналіз сучасних методів стиснення аудіоданих» має бути виконаний до 05.10.2025 р.
2. Розділ 2 «Алгоритм стиснення сегменту сигналу» має бути виконаний до 25.10.2025 р.
3. Розділ 3 «Ділення сигналу на сегменти» має бути виконаний до 10.11.2025 р.
4. Розділ 4 «Програмна реалізація та експериментальне дослідження» до 20.11.2025 р.
5. Розділ 5 «Економічний розділ» має бути виконаний до 01.12.2025 р.

#### 8. Порядок контролю та приймання

1. Рубіжний контроль провести до 14.11.2025.
2. Попередній захист магістерської кваліфікаційної роботи провести до 02.12.2025.
3. Захист магістерської кваліфікаційної роботи провести в період з 15.12.2025 р. до 19.12.2025 р.

Додаток Б  
(обов'язковий)

**ІЛЮСТРАТИВНА ЧАСТИНА**  
РОЗРОБКА МОДУЛЯ СТИСНЕННЯ АУДІОДАНИХ ДЛЯ СИСТЕМИ  
АВТОМАТИЗАЦІЇ ВІДДІЛУ БЕЗПЕКИ ПІДПРИЄМСТВА

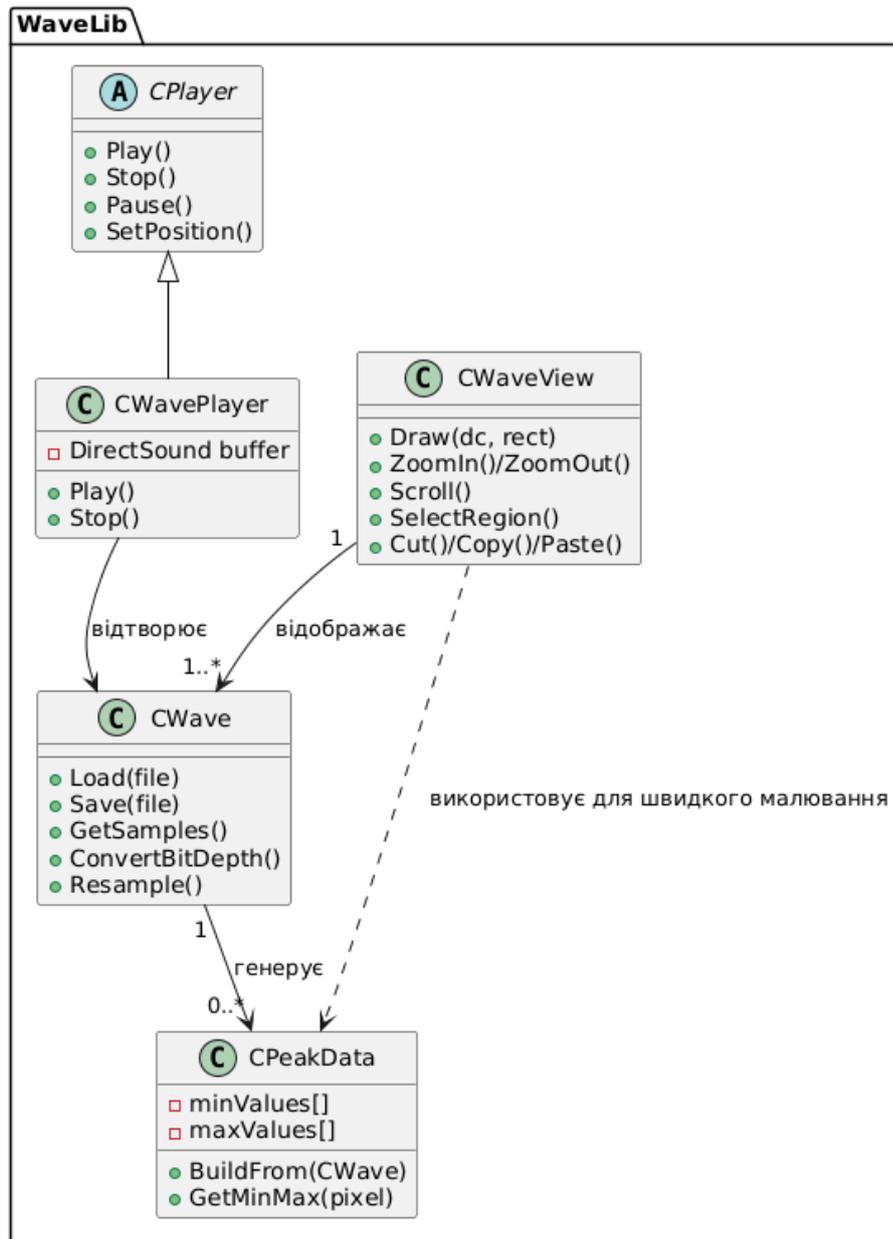


Рисунок Б.1 - UML діаграма класів бібліотеки WaveLib

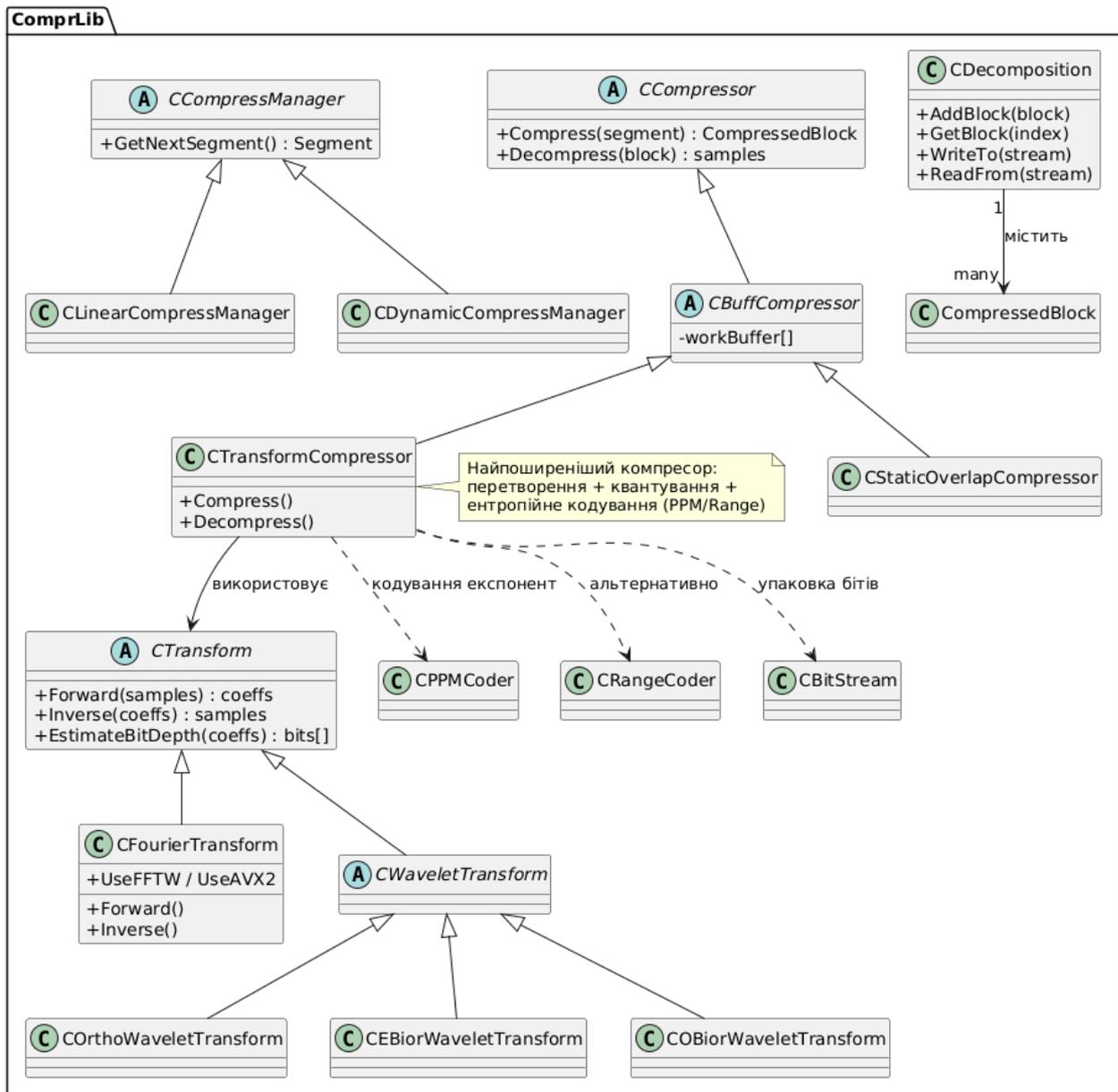


Рисунок Б.2 - UML діаграма класів бібліотеки ComprLib



Рисунок Б.3 – Вейвлет-перетворення. Схема програми

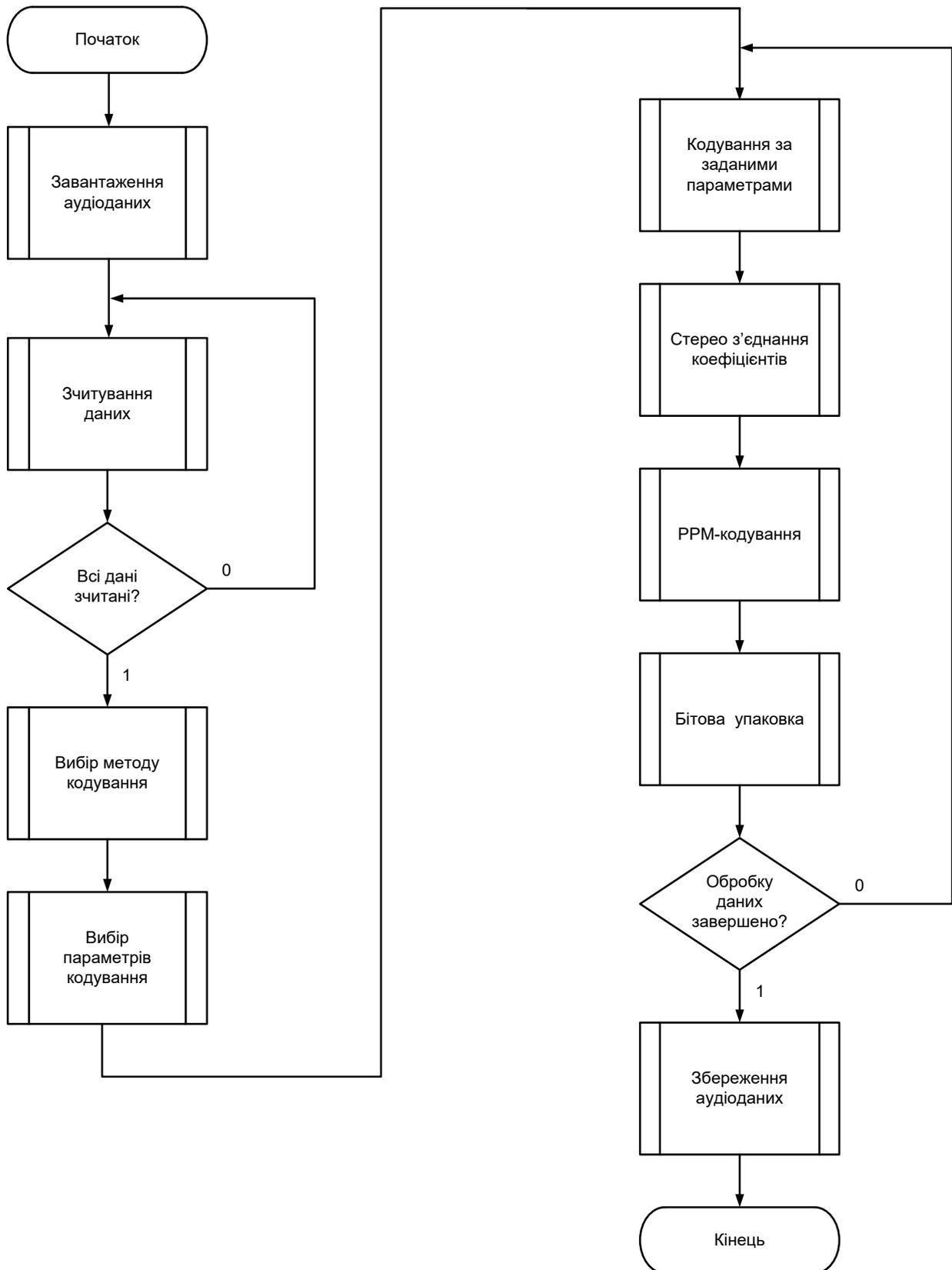


Рисунок Б.4 – Стиснення аудіоданих за допомогою вейвлет-перетворення.

Схема програми

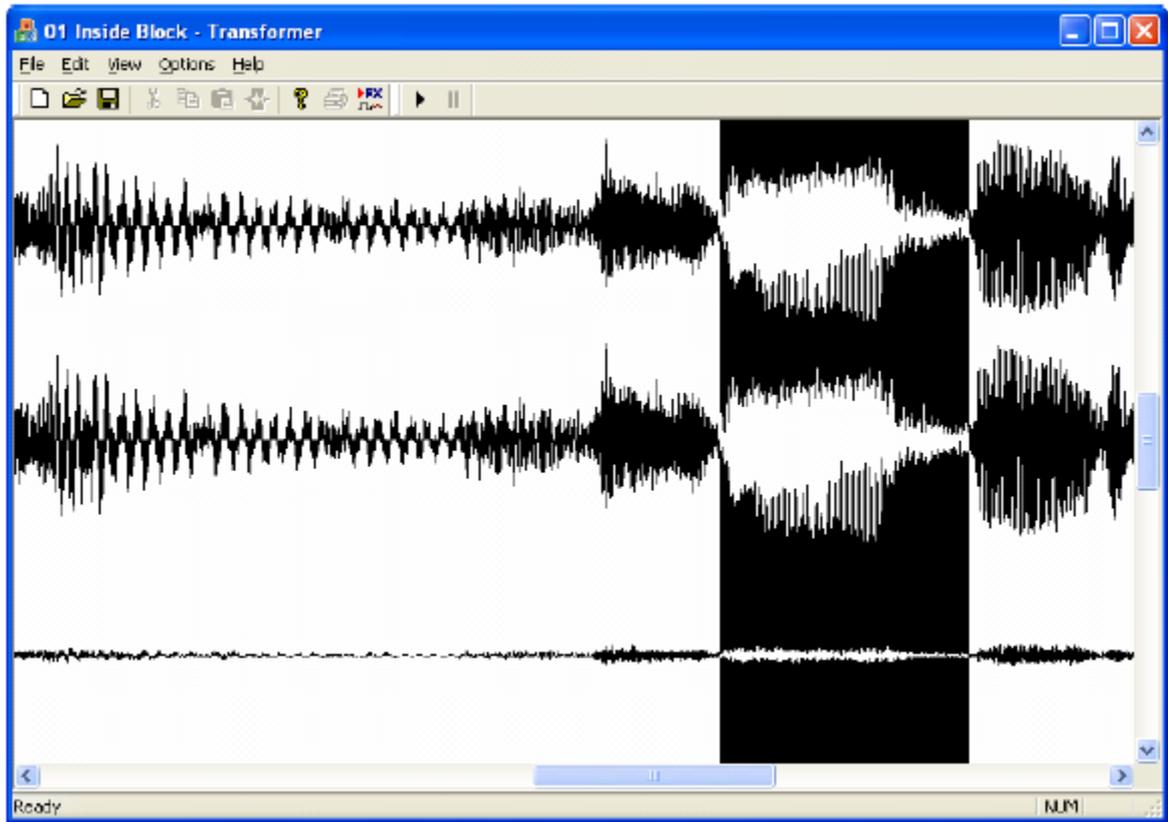


Рисунок Б.5 - Интерфейс програми

Результати експерименту для сильно насиченого сигналу (в спектрі сигналу були присутні частоти широкого діапазону) частина композиції "Inside" від Sting (2003) з альбому "Sacred Love"

Перетворення	Динамічне програмування	Коефіцієнт стиснення (від початкового об'єму)
DWT	Не використовувалось	22,1%
DWT	Використовувалось	22,0%
DFT	Не використовувалось	23,7%
DFT	Використовувалось	21,5%

Результати другого експерименту для несильно насиченого сигналу (в спектрі сигналу присутній тільки вузький діапазон частот) частина композиції Falling - Alicia Keys (2001) Songs in A minor

Перетворення	Динамічне програмування	Коефіцієнт стиску (від початкового об'єму)
DWT	Не використовувалось	19.9%
DWT	Використовувалось	19.8%
DFT	Не використовувалось	20.7%
DFT	Використовувалось	18.9%

Рисунок Б.6 – Результати експерименту

Додаток В (обов'язковий)  
Модуль кодування. Лістинг програми.

```

//
// WA encoding
//
#include "errhand.h"
#include "riff.h"
#include "waff.h"
#include "wavelet.h"
#include "psycho.h"
#include "quant.h"
#include "encode.h"

//
Encoder::Encoder () {
    psycho_model = new PsychoModel ();
    quantisator = new Quantisator ();
    log_encode = new LogFile ("encode.log");
    wa_mode = WA_QUALITY;
    quality = QUALITY_DEFAULT;
    channels = WAFF_MONO;
}
//-----
Encoder::Encoder ( class RiffIO *r, class WaffIO *w, WORD ch, double q ) {
    wa_mode = WA_QUALITY;
    quality = q;
    channels = ch;
    riff = r;
    waff = w;
    psycho_model = new PsychoModel (q);
    quantisator = new Quantisator ();
    log_encode = new LogFile ("encode.log");
}
//-----
Encoder::Encoder ( class RiffIO *r, class WaffIO *w, WORD ch, long bitr ) {
    wa_mode = WA_BITRATE;
    bitrate = bitr;
    channels = ch;
    quality = QUALITY_DEFAULT;
    // Actually this parameter sets a size of the output frame in bytes
    riff = r;
    waff = w;
    psycho_model = new PsychoModel ();
    quantisator = new Quantisator ();
    log_encode = new LogFile ("encode.log");
}
//-----

```

```

void Encoder::SetWAMode ( int mode ) {
    wa_mode = mode;
}
//-----
void Encoder::SetQuality ( double q ) {
    quality = q;
}
//-----
void Encoder::SetBitRate ( long rate ) {
    bitrate = rate;
}
//-----
void Encoder::SetChannels ( WORD ch ) {
    channels = ch;
}
//-----
void Encoder::SetRiffChannels () {
    riff->Open (RIFF_READ);
    channels = riff->riff_header.nChannels;
}
//-----
void Encoder::SetRiffIO ( class RiffIO *r ) {
    riff = r;
}
//-----
void Encoder::SetWaffIO ( class WaffIO *w ) {
    waff = w;
}
//-----
int Encoder::EncodeFrame () {
    long tmp;

    //quantisator->DCompress (signal);
    decompose (signal);
    psycho_model->CalculateMasking (signal);
    psycho_model->ApplyMasking (signal);
    quantisator->PassFilter (signal);
    quantisator->Calculate (signal);
    waff->SetBandMap (psycho_model->band_map);

    for (int b = 0; b < BANDS; b++) {
        if (psycho_model->band_map [b] == 1) {
            waff->WriteBits (quantisator->BitsPerBand[b], 4);
            if (quantisator->BitsPerBand[b] > 0) {
                for (int w = 0; w < band_size [b]; w++) {
                    tmp = (long)signal [band_position [b] + w];
                    if (tmp < 0)
                        tmp = (abs(tmp) << 1) + 1;
                    else
                        tmp <<= 1;
                    // ---- ???
                }
            }
        }
    }
}

```

```

        waff->WriteBits (tmp, quantisator->BitsPerBand [b] + 1);

        //printf ("\t\t[%d]\n", tmp);
    }
} else {
    // Save noise level for unaudible band
    waff->WriteBits ((int)(psycho_model->band_level[b]), 8);
}
}

return TRUE;
}
//-----
int Encoder::Encode () {
    //double current_quality = QUALITY_DEFAULT;
    //double m_quality = QUALITY_DEFAULT;
    long current_bitrate = bitrate + 1;
    long m_bitrate = bitrate + 1;
    int current_bands = BANDS;
    int m_bands = BANDS;
    int n_points = 10;
    long done = 0;
    long dlimit = 1;

    if ((riff->Open (RIFF_READ)) && (waff->Open (WAFF_WRITE))) {
        waff->InitHeader ();
        waff->waff_header.nChannels = channels;
        waff->WriteHeader ();

        switch (wa_mode) {
        case WA_QUALITY:
            dlimit = riff->riff_header.data_size;
            printf ("|-----|\n");
            psycho_model->SetQuality (quality);
            log_encode->WriteLog ("Quality based encoding");
            while (!feof (riff->riff_file)) {
                riff->ReadFrame ();
                //----- Right channel output
                for (int i = 0; i < FRAME_SIZE; i++)
                    signal [i] = riff->r_buffer [i];
                EncodeFrame ();
                waff->WriteFrame ();

                fprintf (log_encode->log_file, "m frame size: %d\n", waff-
>frame_size);

                done += FRAME_SIZE - OVERLAY;
                //----- Left channel output
                if (channels > WAFF_MONO) {

```

```

        for (int i = 0; i < FRAME_SIZE; i++)
            signal [i] = riff->l_buffer [i];
        EncodeFrame ();
        waff->WriteFrame ();

        fprintf (log_encode->log_file, "d frame size: %d\n", waff-
>frame_size);

        done += FRAME_SIZE - OVERLAY;
    }
    if (done > dlimit) done = dlimit;
    if (done > dlimit / 64) {
        printf ("*");
        done -= dlimit / 64;
    }

    //printf ("%f\t%d\t%d\n", status, waff->frames_processed, waff-
>frame_size);
}
printf ("\n");
break;

case WA_BITRATE:
    dlimit = riff->riff_header.data_size;
    printf ("|-----|\n");
    log_encode->WriteLog ("Frame size based encoding");

    while (!feof (riff->riff_file)) {
        riff->ReadFrame ();
        current_bands += 2;
        if (current_bands > BANDS)
            current_bands = BANDS;
        m_bands += 2;
        if (m_bands > BANDS)
            m_bands = BANDS;

        current_bitrate = bitrate + 1;
        m_bitrate = bitrate + 1;
        //----- right channel (or middle channel)
        while (current_bitrate > bitrate) {
            for (int i = 0; i < FRAME_SIZE; i++)
                signal [i] = riff->r_buffer [i];
            //psycho_model->SetQuality (current_quality);
            psycho_model->SetBands (current_bands);
            waff->ResetIOBuffer ();
            EncodeFrame ();
            current_bitrate = waff->io_index;
            //current_quality += QUALITY_STEP;
            current_bands--;
        }
        waff->WriteFrame ();
    }

```



```
        return TRUE;
    }
    return FALSE;
}
//-----
Encoder::~Encoder () {
    delete psycho_model;
    delete quantisator;
    delete log_encode;
}
```

## Модуль декодування. Лістинг програми.

```

//
// WA decoding
//

#include <stdio.h>
#include "errhand.h"
#include "riff.h"
#include "waff.h"
#include "wavelet.h"
#include "psycho.h"
#include "quant.h"
#include "decode.h"

Decoder::Decoder () {
    quantisator = new Quantisator ();
}
//-----
void Decoder::SetRiffIO ( class RiffIO *r ) {
    riff = r;
}
//-----
void Decoder::SetWaffIO ( class WaffIO *w ) {
    waff = w;
}
//-----
int Decoder::DecodeFrame () {
    long bmask = 1L;
    int bits_per_band, noise_level;
    long tmp;
    double dnoise_level, nl;

    for (int z = 0; z < FRAME_SIZE; z++)
        signal [z] = 0.0;
    bmask = 1L;
    for (int b = 0; b < BANDS; b++) {
        if ((waff->bandmap & bmask) != 0) {
            // data band
            bits_per_band = waff->ReadBits (4);
            if (bits_per_band > 0)
                for (int w = 0; w < band_size [b]; w++) {
                    // --- ???
                    tmp = waff->ReadBits (bits_per_band + 1);

                    if ((tmp & 1L) != 0)
                        tmp = - ((tmp - 1) >> 1);
                    else
                        tmp >>= 1;
                }
        }
    }
}

```

```

        signal [band_position [b] + w] = (double)tmp;
    }
} else {
    // noise band
    noise_level = waff->ReadBits (8);
    dnoise_level = pow (10, noise_level / 20) - 1;
    for (int w = 0; w < band_size[b]; w++) {
        nl = ((double)(rand() - 16384)) * dnoise_level / 16384;
        if (nl < 0)
            nl = - quant_level [b] * sqrt (-nl / 32768);
        else
            nl = quant_level [b] * sqrt (nl / 32768);
        signal [band_position [b] + w] = nl;
    }
}
bmask <<= 1;
}

quantisator->iPassFilter (signal);
compose (signal);
//quantisator->DExpand (signal);
for (int i = 0; i < FRAME_SIZE; i++) {
    if (signal[i] > 32767)
        signal[i] = 32767;
    if (signal[i] < -32767)
        signal[i] = -32767;
}

return TRUE;
}
//-----
int Decoder::Decode () {
    int m, d;
    unsigned long done = 0;

    printf ("Starting decoding process...\n");
    printf ("|-----|\n");
    if ((waff->Open (WAFF_READ)) && (riff->Open (RIFF_WRITE))) {
        // prepare RIFF header
        riff->InitHeader ();
        if (waff->waff_header.nChannels == WAFF_MONO)
            riff->riff_header.nChannels = 1;
        else
            riff->riff_header.nChannels = 2;

        riff->WriteHeader ();
        while ((DWORD)waff->frames_processed < waff->waff_header.nFrames) {
            waff->ReadFrame ();
            DecodeFrame ();
            done++;
        }
    }
}

```

```

for (int i = 0; i < FRAME_SIZE; i++) {
    riff->r_buffer [i] = (int)signal [i];
}
// Left channel
if (waff->waff_header.nChannels > WAFF_MONO) {
    waff->ReadFrame ();
    DecodeFrame ();
    done++;
    for (int i = 0; i < FRAME_SIZE; i++) {
        riff->l_buffer [i] = (int)signal [i];
        m = riff->r_buffer [i];
        d = riff->l_buffer [i];
        if ((m + d) > 32767)
            riff->r_buffer [i] = 32767;
        else if ((m + d) < -32767)
            riff->r_buffer [i] = -32767;
        else
            riff->r_buffer [i] = m + d;

        if ((m - d) > 32767)
            riff->l_buffer [i] = 32767;
        else if ((m - d) < -32767)
            riff->l_buffer [i] = -32767;
        else
            riff->l_buffer [i] = m - d;

        //riff->r_buffer [i] = m + d;
        //riff->l_buffer [i] = m - d;
    }
}

riff->WriteFrame ();
if (done > waff->waff_header.nFrames / 64 * waff->waff_header.nChannels)
{
    printf ("*");
    done    -=    waff->waff_header.nFrames    /    64    *    waff-
>waff_header.nChannels;
}
}

printf ("\n");
riff->riff_header.samples = waff->frames_processed * (FRAME_SIZE -
OVERLAY)* riff->riff_header.wBitsPerSample / 8;
riff->riff_header.data_size = riff->riff_header.samples + 36;

riff->WriteHeader ();

waff->Close ();
riff->Close ();

```

```
    } else
        fatal_error ("Decoder::Decode()\tFile operation failure");

    return TRUE;
}
//-----
Decoder::~Decoder () {
    delete quantisator;
}
//-----
```

Додаток Г (обов'язковий)

**ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ**Назва роботи: «Розробка модуля стиснення аудіоданих для системи автоматизації відділу безпеки підприємства»Тип роботи: магістерська кваліфікаційна робота  
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)Підрозділ кафедра АІТ  
(кафедра, факультет, навчальна група)Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism (КП1) 13,45 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак академічного плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту.
- У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- У роботі виявлено ознаки академічного плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень. Робота до захисту не приймається.

Експертна комісія:

Бісікало О.В., зав. каф. АІТ

(прізвище, ініціали, посада)

Овчинников К.В., доц. каф. АІТ

(прізвище, ініціали, посада)

(підпис)

(підпис)

Особа, відповідальна за перевірку

(підпис)

Маслій Р.В.

(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник

(підпис)

Гармаш В.В., доц. каф. АІТ

(прізвище, ініціали, посада)

Здобувач

(підпис)

Кончинський А.О.

(прізвище, ініціали)