

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

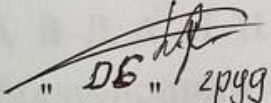
Кафедра обчислювальної техніки

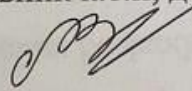
**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

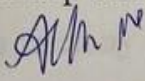
на тему:

**МЕТОД ПРИСКОРЕННЯ РОЗРАХУНКІВ АКУСТИЧНИХ ПОВЕРХОНЬ В БАГАТОКАНАЛЬНІЙ АНАЛОГО-ЦИФРОВІЙ СИСТЕМІ**

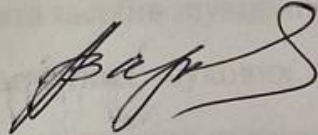
Виконав студент 2 курсу, групи 2КІ-22м  
спеціальності 123 – Комп'ютерна інженерія

  
" 06 " грудня 2023 р. Савчук Д.А.

Керівник к.т.н, доц.каф ОТ  
  
" 07 " 12 2023 р. Крупельницький Л.В.

Опонент к.ф.-м.н доц.каф МБІС  
  
" 08 " 12 2023 р. Шиян А.А.

Допущено зав.каф ОТ  
д.т.н., проф. Азаров О.Д.

  
" 11 " грудня 2023 р.

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра обчислювальної техніки

Рівень вищої освіти II-й (магістерський)

Галузь знань 12 — Інформаційні технології

Спеціальність 123 — «Комп'ютерна інженерія»

Освітня програма — «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

обчислювальної техніки

проф., д.т.н. О.Д. Азаров

«26» вересня 2023 р.

## ЗАВДАННЯ

### НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

студенту Савчуку Дмитру Анатолійовичу

1 Тема роботи «Метод прискорення розрахунків акустичних поверхонь в багатоканальній аналого-цифровій системі» керівник роботи **Крупельницький Леонід Віталійович**, к.т.н., доцент,

затверджені наказом вищого навчального закладу від 18 вересня 2023р. № 247

2 Строк подання студентом роботи 8 грудня 2023 року

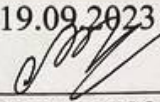
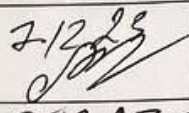
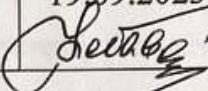
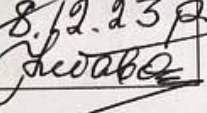
3 Вихідні дані до роботи: дані, що отримуються в результаті аудіо локації.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналіз методів та засобів звукової локації та ідентифікації, аналіз технології обробки низькочастотних звукових сигналів, розробка програмного модуля, експериментальні дослідження та тестування, розрахунок економічної доцільності створення програмного модуля для обрахунку акустичних поверхонь в багатоканальній аналого-цифровій системі.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): план розподілу полів головної сторінки, загальний вигляд головної сторінки додатка.

6 Консультанти розділів роботи представлені в таблиці 1.







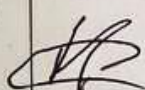
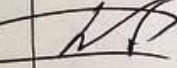
Таблиця 1— Консультанти розділів роботи

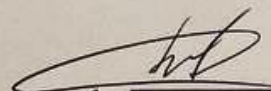
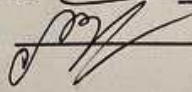
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Крупельницький Л.В., к.т.н., доц. каф. ОТ	19.09.2023 	2.12.23 
5	Небава М.І., к.е.н., професор кафе- дри ЕПВМ	19.09.2023 	8.12.23 р. 

7 Дата видачі завдання 19.09.2023 р.

8 Календарний план наведено в таблиці 2.

Таблиця 2—Календарний план

№	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Підпис
1	Аналіз методів та засобів звукової локації та ідентифікації	04.10.2023	
2	Аналіз технології обробки низькочастотних звукових сигналів	18.10.2023	
3	Розробка програмного модуля	01.11.2023	
4	Експериментальні дослідження та тестування	15.11.2023	
5	Економічна частина	25.11.2023	
6	Попередній захист	16.10.2023	
7	Оформлення пояснювальної записки і презентації	02.12.2023	
8	Перевірка якості виконання магістерської кваліфікаційної роботи та усунення недоліків	07.12.2023	

Студент  Савчук Д. А.  
Керівник роботи  Крупельницький Л. В.

## АНОТАЦІЯ

УДК 004

Савчук Д.А. Метод прискорення розрахунків акустичних поверхонь в багатоканальній аналого-цифровій системі. Магістерська кваліфікаційна робота зі спеціальності 123 – Комп'ютерна інженерія, Вінниця: ВНТУ, 2023 – 116 с. На укр. Мові. Бібліогр. 21 назв; рис.: 27; табл. 8.

У роботі розглянуто існуючий метод прискорення обрахунку акустичних поверхонь в багатоканальній аналого-цифровій системі. Аналізуються метод і засоби акустичної локації та ідентифікації об'єктів. Досліджено метод та системи багатоканального сканування за допомогою мікрофонних решіток. Розглянуто технологію CUDA-ядер в мові програмування C++. Було розроблено модуль програмного забезпечення для розрахунку акустичних поверхонь в багатоканальній аналого-цифровій системі.

Ключові слова: акустичні поверхні, багатоканальні аналого-цифрові системи, мікрофонна решітка, низькочастотний сигнал, CUDA-технології, CUDA-ядра.

## **ABSTRACT**

УДК 004

Savchuk D.A. Methods for accelerating the calculation of acoustic surfaces in a multichannel analogue-digital system. Master's thesis in speciality 123 - Computer Engineering, Vinnytsia: VNTU, 2023 - 116 p. In Ukrainian. Language. Bibliography: 21 titles; Figures: 27; Table 8.

The paper considers the existing methods for accelerating the calculation of acoustic surfaces in a multichannel analogue-digital system. The methods and means of acoustic location and identification of objects are analysed. Methods and systems for multichannel scanning using microphone arrays are investigated. The technology of CUDA cores in the C++ programming language is considered. A software module for the calculation of acoustic surfaces in a multichannel analogue-digital system was developed.

**Keywords:** acoustic surfaces, multichannel analogue-digital systems, microphone array, low-frequency signal, CUDA technologies, CUDA kernels.

## ЗМІСТ

<b>ВСТУП</b> .....	<b>8</b>
<b>1 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ І ЗАСОБІВ АКУСТИЧНОЇ ЛОКАЦІЇ ТА ІДЕНТИФІКАЦІЇ ОБ’ЄКТОВ НА МІСЦЕВОСТІ</b> .....	<b>11</b>
1.1 Аналітичний огляд структурних і обчислювальних методів, апаратних і програмних засобів акустичної локації та ідентифікації об’єктів .....	11
1.2 Загальна структурна схема, склад та принцип роботи АЦ-системи аудіолокації й ідентифікації об’єктів на місцевості .....	13
1.3 Розробка технічних вимог до параметрів апаратних пристроїв і функцій програмного забезпечення АЦ-систем .....	16
1.3.1 Вимоги до складу, структури та технічних параметрів АЦ-систем на основі самокаліброваних АЦП і ЦАП .....	16
1.3.2 Вимоги до складу, структури та технічних параметрів АЦ-систем на основі самокаліброваних АЦП і ЦАП .....	26
<b>2 АНАЛІЗ ТЕХНОЛОГІЇ ОБРОБКИ НИЗЬКОЧАСТОТНИХ ЗВУКОВИХ СИГНАЛІВ ПРИ БАГАТОКАНАЛЬНОМУ СКАНУВАННІ В СИСТЕМІ МІКРОФОННОЇ РЕШІТКИ</b> .....	<b>30</b>
2.1 Аналіз методів побудови і принципів функціонування мікрофонних решіток .....	30
2.2 Конструктивно-схемотехнічні рішення мікрофонних решіток .....	36
<b>3 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ДЛЯ РОЗРАХУНКУ АКУСТИЧНИХ ПОВЕРХОНЬ В БАГАТОКАНАЛЬНІЙ АНАЛОГО-ЦИФРОВІЙ СИСТЕМІ</b> .....	<b>44</b>
3.1 Огляд технології CUDA .....	44
3.2 Програмування з використанням CUDA технологій .....	49
3.3 Огляд компілятора для використання CUDA в коді C+ .....	52

						<b>08-54.МКР.038.00.000 ПЗ</b>		
<i>Змн.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>		<i>Савчук Д.А.</i>			<i>Метод прискорення розрахунків акустичних поверхонь в багатоканальній аналого-цифровій системі</i>	<i>Літ.</i>	<i>Арк.</i>	<i>Аркушів</i>
<i>Керівник</i>		<i>Крупельницький Л.В</i>					6	116
<i>Рецензент</i>		<i>Шиян А.А.</i>				<b>ВНТУ, гр.2КІ-22м</b>		
<i>Н. Контроль</i>		<i>Швець С.І.</i>						
<i>Затверджую</i>		<i>Азаров О.Д.</i>						



## ВСТУП

**Актуальність дослідження** полягає у застосуванні сучасних технологій у прискоренні розрахунків акустичних поверхонь в багатоканальній аналого-цифровій системі. У сучасному світі, де технології звукової локації стрімко розвиваються, виникає необхідність в оптимізації розрахунків у звуковому спектрі з використанням передових методів та засобів.

Наразі вітчизняні аналоги розробок в даній області відсутні, а інформація про подібні зарубіжні системи залишається недоступною чи недостатньо конкретизованою. Інноваційний підхід до прискорення розрахунків акустичних поверхонь через використання покращеного методу є важливим в контексті відсутності відомих ефективних рішень у цьому напрямку.

Запропонований метод використовує технологію CUDA для паралельних обчислень, що забезпечує значне прискорення розрахунків за рахунок використання графічних процесорів від компанії Nvidia.

З урахуванням вищезазначеного, запропонований метод для прискорення розрахунків акустичних поверхонь стає актуальним завданням. Широка сфера використання подібних систем у звуковій області підкреслює важливість вирішення цієї проблеми та вимагає подальших наукових досліджень.

**Метою дослідження** є пришвидшення розрахунку акустичних поверхонь в багатоканальній аналого-цифровій системі за рахунок застосування методів CUDA – технологій, для використання в статичних і портативних комп'ютерних системах.

Для досягнення цієї мети необхідно вирішити такі задачі:

- здійснити аналіз галузі застосування системи;
- класифікувати методи розпізнавання звукових сигналів;
- розробити систему розпізнавання низькочастотних сигналів;
- розробити програмне забезпечення для обробки потокового низькочастотного сигналу;
- провести діагностику роботи системи та програмного забезпечення;



— провести тестування розробленої системи та програмного забезпечення.

**Методи дослідження** дипломної роботи: використовувались та досліджувався метод прискорення розрахунків акустичних поверхонь в багатоканальних аналого-цифрових системах, та наведенні висновки про їх потокову обробку.

**Предметом дослідження** є процеси, обробки низькочастотних сигналів в потоці часу.

**Об'єктом дослідження** є метод прискорення розрахунку акустичних поверхонь за допомогою використанням технології CUDA.

**Наукова новизна** даного дослідження полягає в покращенні методу прискореного розрахунку акустичних поверхонь в багатоканальній аналого-цифровій системі за допомогою передових CUDA-технологій. Цей підхід визначається як новаторський у контексті оптимізації обчислень у звуковому спектрі, що відкриває нові можливості для ефективного використання акустичних систем в різноманітних сферах застосування.

**Практичне значення** розробленого програмного забезпечення в тому, що його функціональність дозволяє створювати динамічні акустичні поверхні у режимі реального часу. Це відкриває широкі можливості для миттєвого адаптування акустичного середовища під конкретні вимоги та умови, що є важливим у сферах, де необхідна негайна реакція на змінні акустичні умови.

**Апробації** результатів роботи здійснена в доповіді на Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії.

Матеріали роботи доповідались та опубліковувались:

Д. А. Савчук, Л. В. Крупельницький / Прискорення цифрової обробки низькочастотних сигналів в аналого-цифрових системах реального часу за допомогою технології CUDA // Тези доповіді. LII Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2023). Вінниця 2023 р. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2023/paper/view/18399>

Звіт про науково-дослідну роботу «Високопродуктивні багатоканальні аналого-цифрові самокалібровані системи моніторингу й синхронного опрацювання низькочастотних сигналів» / ВНТУ, Вінниця, 2022 р., N держ. Реєстрації 0120U002205 – 308 с.

## **1 ОГЛЯД ІСНУЮЧИХ МЕТОДІВ І ЗАСОБІВ АКУСТИЧНОЇ ЛОКАЦІЇ ТА ІДЕНТИФІКАЦІЇ ОБ'ЄКТОВ НА МІСЦЕВОСТІ.**

### **1.1 Аналітичний огляд структурних і обчислювальних методів, апаратних і програмних засобів акустичної локації та ідентифікації об'єктів**

Технології акустичної локації та розпізнавання об'єктів, такі як співвіднесення та електронне сканування каналів мікрофонних решіток, зараз широко використовуються в сучасному світі. Вони частково походять від таких технологій, як радар, гідролокатор і сейсміка. Ефективність аудіолокації визначається не тільки якістю і точністю каналів вимірювання, але й швидкістю обробки сигналу. На вході вимірювальних каналів ефективніше використовувати аналогові сигнали, оскільки вони менш чутливі до шумів і перешкод.

Для забезпечення точності та ідентифікації необхідно використовувати, поряд із самокалібруванням підсилювачів, як динамічні, так і акустичні властивості електричних вимірювальних каналів, а також характеристики мікроскопічної акустики [3]. Самокалібрування припускає лише те, що шум і швидкість оптимально оптимізовані у вимірювальних каналах. Деякі інші параметри (зсув, коефіцієнт підсилення, амплітудно-фазо-частотна характеристика, різниця між каналами) можуть бути визначені і відрегульовані чисельно. Поширюючи принцип самоналаштування, калібрування та вимірювання на вимірювальні канали систем змінного струму, можна підвищити їх продуктивність порівняно з відомими рішеннями.

Як наслідок, традиційні багатоканальні алгоритми електронної розгортки обмежені у своїй продуктивності кількістю послідовних зсувів. Вони також не пропонують режимів реального часу і часто не обчислюють кореляційні суми сигналів. Методи розпаралелювання не беруть до уваги сигнали.

Для ефективного виконання багатоканальної цифрової обробки для системи змінного струму можна використовувати нейронні ієрархічні структури на основі паралелізму та графічні процесори GPU, що використовують технологію багатоканальної обробки CUDA.

Координати джерел сигналів можуть бути спотворені через зовнішні впливи та рух об'єктів, а також апаратні помилки. Для уточнення дискретних координат об'єктів у полі можна застосувати виведення на основі зворотної логіки, яке потім зводиться до розв'язання системи рівнянь із застосуванням нечіткої логіки. Але більшість таких систем можна розв'язати аналітично. Але визначення множини мінімальних розв'язків залишалось недосяжним.

Вирішуючи проблему зворотної ідентифікації, ідея використання нечіткої логіки полягає в тому, щоб підвищити роздільну здатність реальних координат об'єктів. Можна використовувати принцип інкрементної декомпозиції при спробі розв'язати багатовимірні рівняння.

Коли аудіосигнали розпізнаються, вони можуть бути представлені в частотній області лінійними коефіцієнтами передбачення і захищеними коефіцієнтами. Вони також мають спектральні частотні піки. У процесах прийняття рішень найчастіше використовуються змішані гауссівські або приховані марковські моделі. Надійність розпізнавання сигналів також значно падає зі збільшенням співвідношення між шумом і сигналом [5].

В результаті з'явилася можливість реалізувати метод, який спочатку був розроблений для швидкого пошуку по дереву, в  $K$ -вимірному графі, який представляє параметри звуку у вигляді набору параметрів. Використання піків спектральних частот для представлення звукових сигналів допоможе підвищити точність розпізнавання джерел звуку з низьким відношенням сигнал/шум.

Для забезпечення пірингової безпеки необхідні короткі приховані пакети з криптокодуванням. Найпростіший метод криптографічного захисту — гейміфікація — може бути використаний для поєднання завадостійкого кодування відео-сигналу БПЛА та потокового шифрування. Цей та інші методи потребують значних додаткових апаратних та програмних засобів шифрування. Крім того, відкриті вітчизняні та міжнародні публікації не містять конкретних деталей щодо побудови систем криптографічного захисту у складі спеціалізованої системи ідентифікації об'єктів.

Підсистема може ідентифікувати звуковий об'єкт. Для досягнення поставленої мети важливо, щоб сигнал, який використовується для розпізнавання, мав невелику амплітуду. Тим не менш, система повинна бути достатньо захищеною.

## 1.2 Загальна структурна схема, склад та принцип роботи АЦ-системи аудіолокації й ідентифікації об'єктів на місцевості

Багатоканальна обробка акустичних мікрофонних решіток використовується для розробки спеціалізованих АС для аудіолокації та розпізнавання об'єктів. У контексті автоматичного контролю доступу, безпеки, комп'ютерних систем, що ідентифікують джерела звуку, мультимедійних систем на стадіонах, концертних і виставкових залах тощо, акустична локалізація об'єктів і дистанційний збір інформації мають велике значення.

На початку ХХ століття для визначення місцезнаходження артилерії та виявлення літаків використовували спрямовані звукові трубки як рефлектори.

Аналоговий етап обробки сигналів збігся з розвитком електроніки та використанням перших мікрофонних решіток для збору акустичної інформації;

Етап цифрової обробки потокового сигналу. Інструменти цифрової обробки широко використовуються з 1990-х років для вирішення багатьох завдань. До них відносяться визначення місцезнаходження джерела звуку, аналіз звуку, виявлення мовців, слухові апарати та відокремлення звуку від фонового шуму.

Мікрофонний масив може бути різновидом спрямованих мікрофонів, які розташовані як приймачі звуку. Вони працюють разом (синхронно або з фазовою затримкою) всередині або зовні.

Багатоканальні масиви працюють шляхом цифрового об'єднання сигналів з часовими затримками для створення графіка. Багатоканальні перетворювачі перетворюють акустичні сигнали, показані на рисунку 1.1, в цифровий формат. Потім сигнал передається. Персональний комп'ютер виконує всю необхідну обробку цифрових аудіосигналів, включаючи перетворення їх у бітові масиви для подальшої обробки.

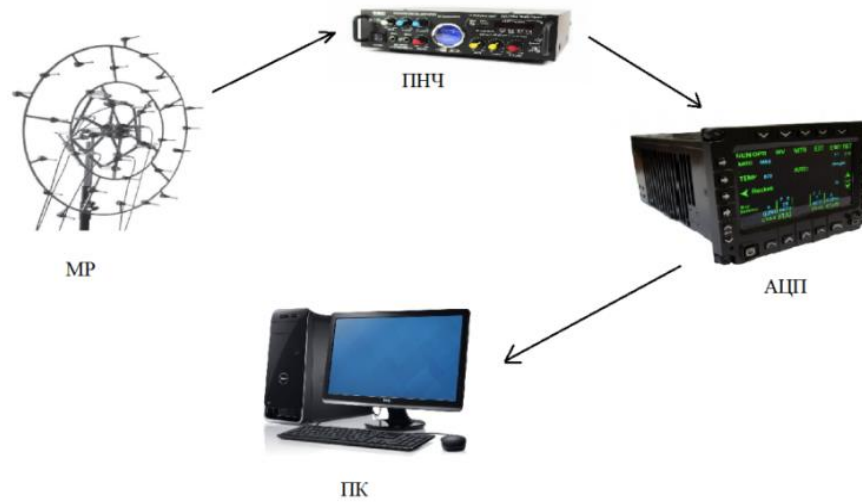


Рисунок 1.1 — Проходження акустичних сигналів у комп'ютерній системі з «цифровою» мікрофонною решіткою

Коли акустичні сигнали досягають масиву мікрофонів, виникають затримки, зумовлені джерелом сигналу і швидкістю, з якою поширюється хвиля. Як показано на рисунку 1.2, ви можете вибрати сигнал у зворотному напрямку, встановивши необхідну затримку в кожному каналі.

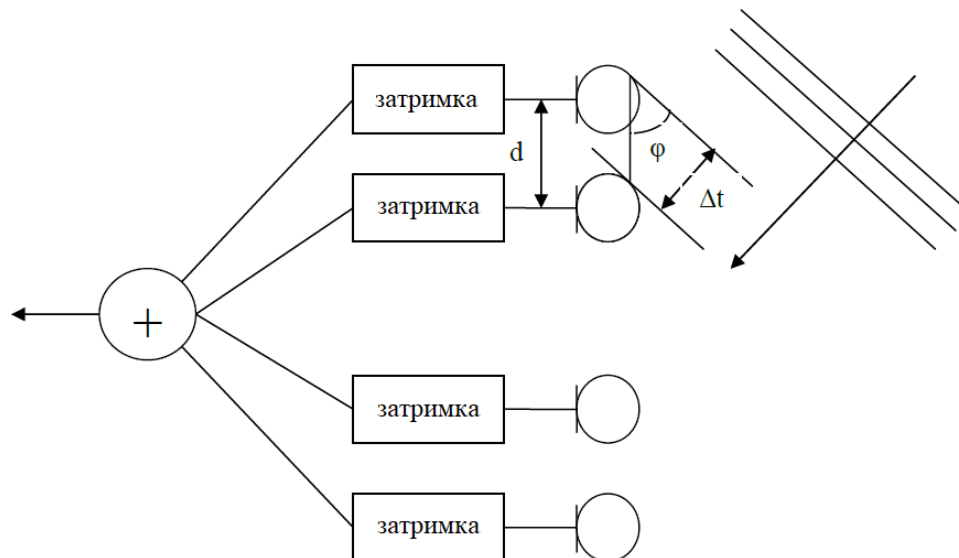


Рисунок 1.2 — Затримки сигналів для виділення напрямку на джерело звуку

Нехай мікрофонна решітка складається з  $N$  приймачів,  $d$  — відстань між сусідніми приймачами. Потім, щоб задати напрямок основної пелюстки візерунка під кутом  $\varphi$ , зсув часу між сусідніми мікрофонами має дорівнювати:

$$\Delta t = d \cdot \sin(\varphi) / v, \quad (1.1)$$

де  $v$  — швидкість розповсюдження звуку в повітрі.

Щоб розрізнити різні кути напрямків,  $\rho_h$  в сигналах, отриманих мережею мікрофонів, необхідно об'єднати дискретизовані сигнали з часовими затримками, відповідними для кожного напрямку. Легко ввести затримки між сигналами, зсунувши їх на однакову кількість  $m$  відліків, що безпосередньо пов'язано з частотою дискретизації:

$$\Delta t = m / Fd. \quad (1.2)$$

Кут  $\varphi$  діаграми направленості мікрофонної решітки при однаковій відстані між мікрофонами  $d$  залежить від кількості зсувів між дискретними відліками сусідніх сигналів перед підсумовуванням:

$$\varphi = \arcsin(m \cdot v / (d \cdot Fd)). \quad (1.3)$$

При малих значеннях  $M$  послідовні значення кутів  $\rho_h$  для  $m = \dots, 3, 2, 1, 0, +1, \dots$ , плюс, два, три і т.д. приблизно рівні. Але ця характеристика не є принциповою для електронних сканерів місцевості. Роздільна здатність буде знижуватися під великими кутами, але точні значення все одно будуть відповідати співвідношенню.

Тепер розглянемо деякі конструктивні параметри мікрофонної решітки. Слід зазначити, що відстань  $d$  між мікрофонами не може бути довільною, оскільки необхідно враховувати час, за який звукова хвиля досягає потрібної частоти сигналу. Це визначається швидкістю звуку:

$$Ds = v / Fs. \quad (1.4)$$

Так, при  $v = 330$  м/с і частоті звукового сигналу  $Fs = 3300$  Гц довжина періоду сигналу складе 0.1 м (100 мм), що означатиме періодичне повторення сигналу, який приходиме з кутів  $\varphi = 180^\circ$  та  $\varphi = -180^\circ$ , якщо відстань між мікрофонами складатиме 100 мм. Це означатиме, що на частоті  $Fs = 3300$  Гц утвориться дві бокових пелюстки діаграми направленості, рівні центральній на куті  $\varphi = 0^\circ$ .

З огляду на вищевказану інформацію та з урахуванням мети і основних бакалаврської роботи, було вирішено використати спосіб роботи системи, який ілюструє структурна система, що наведена на рисунку 1.3.

Аудіосигнали приймає одна мікрофонна решітка. Попередньо посилені аналогові сигнали в кожному з вимірювальних каналів перетворюються в цифрову форму за допомогою високоточного АЦП [7].

### 1.3 Розробка технічних вимог до параметрів апаратних пристроїв і функцій програмного забезпечення АЦ-систем

#### 1.3.1 Вимоги до складу, структури та технічних параметрів АЦ-систем на основі самокаліброваних АЦП і ЦАП

За останні кілька десятиліть значно зросли вимоги до статичних і динамічних властивостей вимірювально-інформаційних систем для перетворення та обробки низькочастотних сигналів, спектр яких знаходиться в межах звукового діапазону частот. Ці системи змінного струму можуть бути використані для сейсмічної розвідки, тензометрії, гідрографії та локалізації звуку, а також у біомедицині. Здатність системи АЦП досягати бажаних властивостей залежить від АЦП і ЦАП, а також інших аналогових пристроїв, таких як підсилювачі, перемикачі та фільтри.



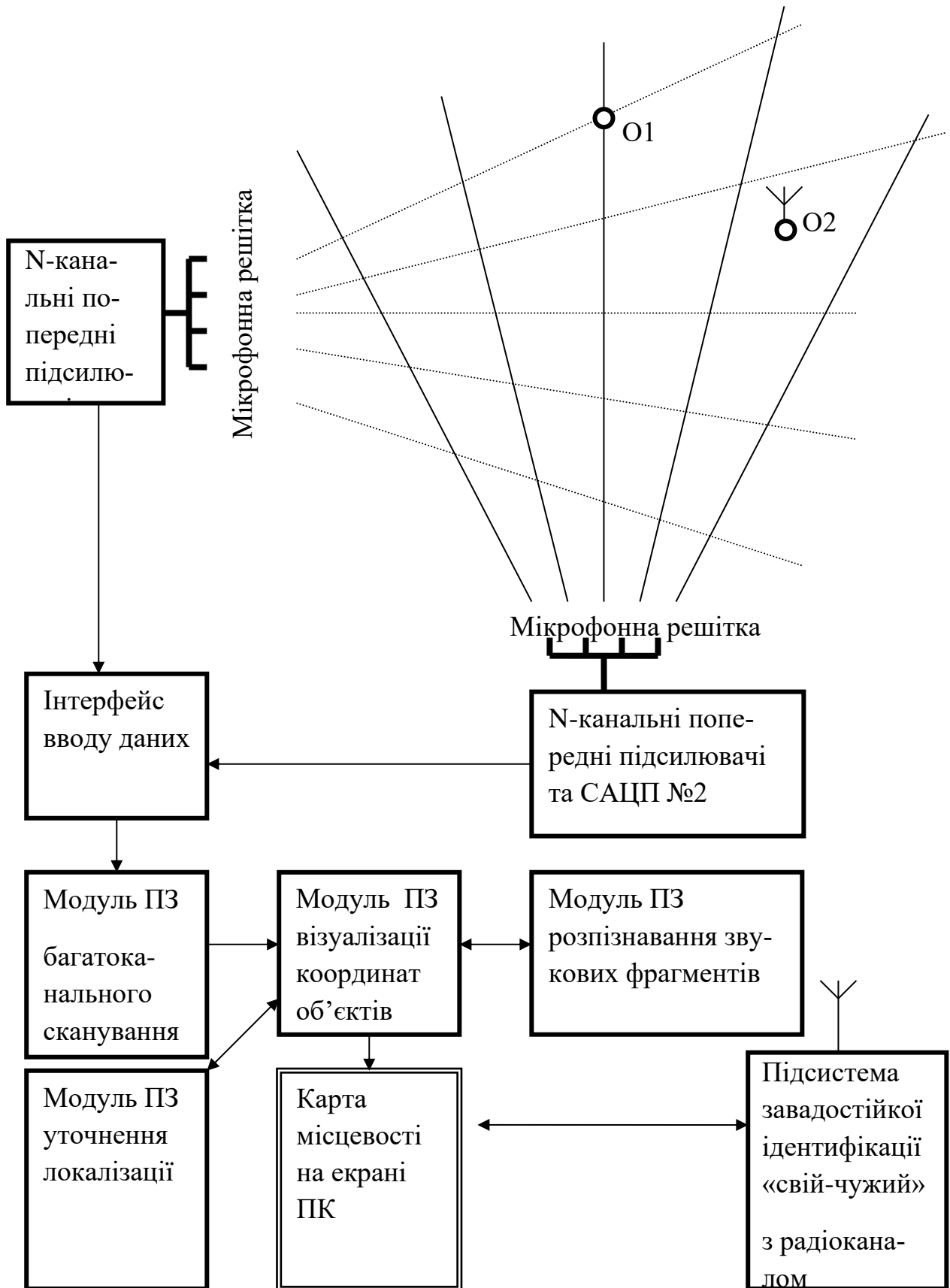


Рисунок 1.3 -- Структурна схема АЦ-системи аудіолокації та ідентифікації об'єктів на місцевості

Вибір елементної бази визначається діапазонами значень рівнів і частот досліджуваних сигналів, а також вимогами до статичних і динамічних властивостей перетворення. Тому, залежно від призначення АЦ-системи, статичні характеристики повинні забезпечувати такий комплекс показників:

- широкий динамічний діапазон сигналів, що перетворюються та обробляються (від 100 дБ до 140 дБ);
- висока роздільна здатність (від 16 двійкових цифр до 24 двійкових цифр);
- низька нелінійність аналого-цифрового перетворення (від 0,001% до 0,01%).

Динамічні властивості АЦ-системи повинні враховувати такі основні показники:

- широкий діапазон частот сигналів (від 1 Гц до 10000 Гц);
- незначний рівень шумів в спектрі сигналу (від мінус 90 дБ до 120 дБ);
- низький коефіцієнт нелінійних спотворень (від 0,001 % до 0,01%);
- низька нерівномірність амплітудно-частотної характеристики ( $\pm 0,03$  дБ) та нелінійність фазочастотної характеристики ( $\pm 0,1$ ) градусів у смузі корисного сигналу.

Для забезпечення виконання цих вимог можуть бути використані методи створення самокорегуючих АЦП, ЦАП та аналогових пристроїв з використанням двокаскадних підсилювачів постійного струму зі збалансованим зворотним зв'язком, а також методи побудови аналогових пристроїв. Незважаючи на успіх досліджень, важливо зазначити, що використання методів корекції в системах АЦП загалом було недостатнім.

Метою роботи є покращення динамічних та статичних властивостей АЦП для обробки низькочастотних сигналів за рахунок використання АЦП, що самокалібруються, та ЦАП на вимірювальних каналах. Поставлена мета має наступні задачі дослідження:

— проаналізувати можливість використання АЦП на основі НПСЧ з метою покращення динамічних та статичних властивостей каналів перетворення для багатоканальної системи АЦП;

— розглянути шляхи підвищення швидкодії та точності АЦП шляхом поширення принципів корекції та компенсації динамічних похибок в надлишкових АЦП, ЦАП, на функціональні блоки вимірювальних каналів;

— запропонувати структурні схеми АЦП та ЦАП на основі ШІМ для вимірювання та аналізу низькочастотних сигналів, а також показати отримані значення статичних та динамічних параметрів.

Розглянути важливі аспекти АЦП, ЦАП та ФНЧ на основі системи числення з ваговою надлишковістю [1, 16].

В НПСЧ використовується недвійкова основа системи числення  $1 < \alpha < 2$ , а ваги розрядів представляються у вигляді:

$$Q_i = q \cdot \alpha^i, \quad (1.4)$$

де  $q$  – вага молодшого розряду.

Будь-яке дійсне число може бути представлене в НПСЧ у вигляді суми ваг розрядів:

$$X = \sum_{i=0}^N a_i \cdot q \cdot \alpha^i, \quad (1.5)$$

де  $a_i = \overline{0, 1}$  – двійковий  $i$ -тий біт  $N$ - розрядного результату перетворення.

До НПСЧ відноситься, зокрема, так званий код "золотої пропорції" з новою

$$\alpha = (\sqrt{5} + 1)/2 \approx 1,618 \quad (1.6)$$

Застосування ШПФ в аналого-цифровому перетворенні має основну перевагу — усунення "пропусків", спричинених відхиленнями між розрахунковими та фактичними значеннями для бітів. Це відхилення не повинно перевищувати половини значення молодшого значущого біта. АЦП, що використовують "золоту середину", можуть мати відносні похибки у вазі розрядів, спричинені технологічними, температурними та часовими факторами, до 23,6%. Вони не призводять до пропусків коду. Таким чином, можна отримати аналогове вхідне значення, знаючи значення фактичних ваг бітів, які беруть участь у перетворенні.

Самокалібрування — режим роботи АЦП, який дозволяє вимірювати фактичні ваги. Бітовий збалансований АЦП, який використовує ШПФ, має додаткову функцію, яка називається прискореним цифро-аналоговим перетворенням. Самокорекція зменшує час, необхідний для кожного порівняння, оскільки помилка неправильного відкриття старших бітів виправляється активацією молодших бітів.

Теоретично та експериментально доведено [1, 2], що застосування вагової надлишковості в техніці АЦ-перетворення дозволяє істотно (на 1-2 порядки) підвищити лінійність характеристики перетворення пристроїв, побудованих на неточних елементах, наприклад, за умови побудови ЦАП без технологічного припасування ваг розрядів. Крім цього, є можливість істотно (у 5-8 разів) збільшити швидкість порозрядного аналого-цифрового перетворення за рахунок автоматичної компенсації динамічних похибок врівноваження [2]. Вказані переваги НПСЧ для побудови точних та швидкодіючих АЦП і ЦАП реалізуються як в симетричному базисі числення  $(\bar{1}, 1)$ , та і у більш поширеному – несиметричному  $(0,1)$  базисі.

Схема базової самокаліброваної АЦ-системи на рисунку 1.4 містить:

С — сенсори вхідних аналогових сигналів ( $m$  в кожному каналі);

ВК — вхідні комутатори;

ПП — програмовані підсилювачі;

ФНЧ — антиаліасінгові фільтри низьких частот;

КП — компаратори перевантаження;

САЦП — сигнальний АЦП, що самокоригується;

ДОН — джерело опорної напруги для перемножуючого ЦАП опорних сигналів.

Послідовне використання декількох сенсорів вхідного сигналу  $C(1..m)$  необхідно для економії кількості каналів системи, на випадок якщо є можливість змінити групи або типи сенсорів або змінити напрямок прийому акустичного сигналу. Вибір оптимального коефіцієнта підсилення можливий завдяки використанню компараторів КП, які контролюють наявність перевантажень вхідними сигналами.

Контролери інтерфейсу даних на мікропроцесорах або спеціалізованих сигнальних процесорах отримують дані з каналів АЦП, а потім передають їх через обраний високошвидкісний інтерфейс (USB/S/PDIF/Bluetooth/Ethernet/Wi-Fi). На ПК запускається спеціалізована програма для аналізу сигналу, яка потім керує системою корекції АЦП за допомогою відповідного контролера.

До системи самокалібрування АЦП додається мультиплікаційний ЦАП. Він має максимальну амплітуду сигналу, яку можна чітко визначити, що дорівнює напрузі DON. ЦАП відтворюють тестові сигнали на основі алгоритму або калібрування. До них відносяться імпульси з фіксованим періодом, амплітудою і частотою, а також гармоніки і полігармоніки з набору заданих частот. Тестові сигнали з ВК подаються одночасно на вимірювальні канали АЦП через комутатор ВК на програмовані підсилювачі в ПП. Таке калібрування виконується "зверху вниз" послідовно для всіх значень коефіцієнта підсилення ПП, охоплюючи параметри АЦП, комутаторів, підсилювачів і фільтрів. Схемотехніка аналогових АЦП має свої особливості і може бути побудована на основі принципів підсилення за струмом. АЦП, побудований з використанням НПФК, має можливість самокорекції методами, які вже добре відомі [1,2,6] або нові.

У структурі рис. 1.10 кожен канал оснащений окремим АЦП. Це можуть бути сучасні дельта-сигма АЦП. Надмірна частота дискретизації, децимація та цифровий фільтр дозволяють отримати вихідні дані на рівні від 20 до 24 біт у

низькочастотних діапазонах та для аудіосигналів. Ці дельта-сигма АЦП не можна використовувати в системах з швидким перемиканням датчиків або імпульсними сигналами. Вони також не можуть впоратися з короткочасними перевантаженнями в підсилювачах. Для таких застосувань підходить схема на рисунку 1.5 з послідовним перетворенням змінного струму в кожному каналі є економічно ефективним і оптимальним рішенням завдяки побітовій компенсації та ваговій надлишковості АЦП. АЦП, що самокорегується, доповнює структуру багатоканального АЦП для аналізу сигналів на низьких частотах.

Датчики аналогових входів здатні перетворювати різні фізичні сигнали в напругу або струм. Однак їхні принципи роботи суттєво відрізняються. Пасивні аналогові датчики безпосередньо генерують вихідний струм, пропорційний зовнішньому фізичному впливу. До них відносяться динамічні мікрофони, такі як акустичні або гідроакустичні, фотоелектричні, сегнетоелектричні, термопари та інші.

Напруга, яку створюють ці датчики, може бути досить низькою - від десятків до тисяч мілівольт. Потенціометри (активні датчики), термістори (термодатчики), тензорезистори (гальваномагнітні датчики) тощо - все це активні датчики. Такі датчики зазвичай живляться від мостових блоків живлення, які генерують низькі вихідні диференціальні напруги або струми. Використовуйте перемикання полярності або мости змінного струму, щоб відокремити корисні сигнали від фонових перешкод.

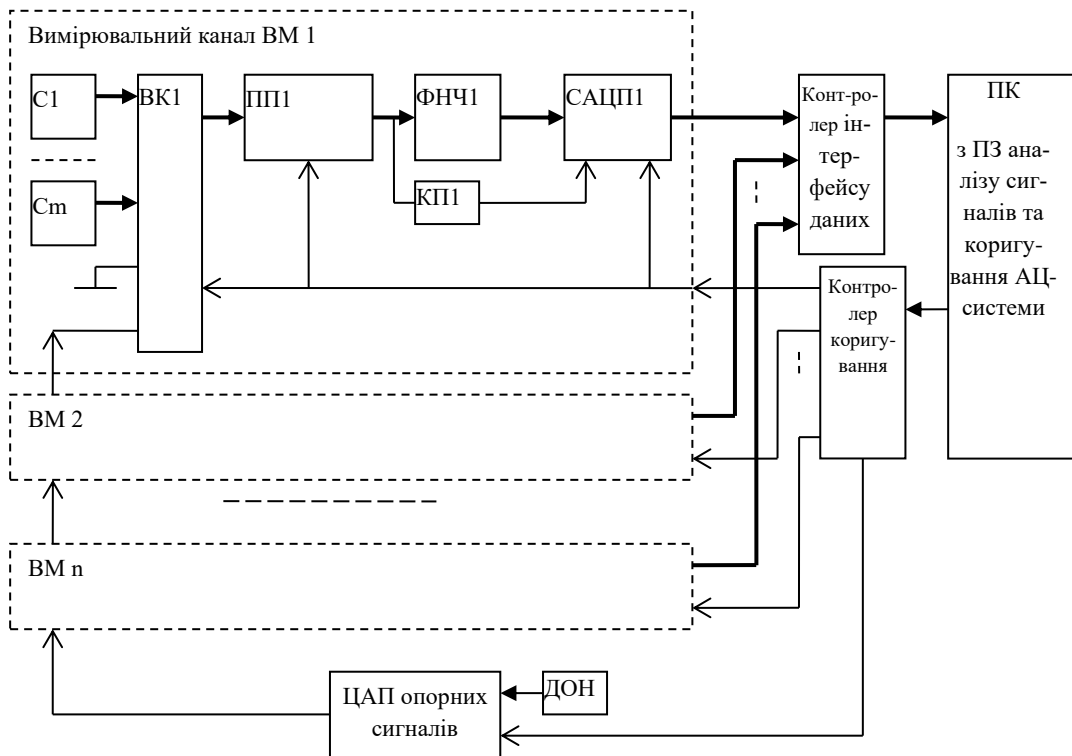


Рисунок 1.4 — Структура багатоканальної АЦ-системи для аналізу сигналів з коригуванням характеристик вимірювальних каналів

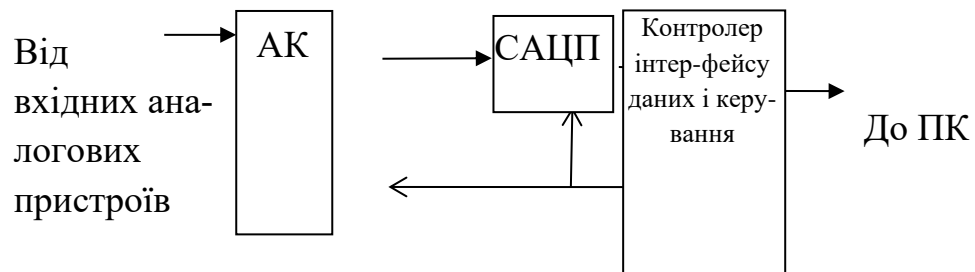


Рисунок 1.5 — Варіант схеми з послідовним АЦ-перетворенням сигналів

Активні датчики також об'єднані в окрему групу. В цих активних датчиках в якості інформаційного параметра використовується амплітуда вихідної напруги змінного струму, його частота або фаза. Ці датчики вимірюють фізичні величини, які впливають на ємність, взаємну індуктивність та індуктивність. До них відносяться ємнісні датчики, індуктивні датчики, трансформатори, магнітострикційні датчики, струнні та інші. Вони можуть подавати свій змінний вихід

безпосередньо на АЦП або перетворювати його в пропорційний сигнал. Параметри властивостей перетворювача визначаються під час виготовлення пристрою, а також під час його експлуатації. Для визначення реальних значень цих параметрів використовується окрема процедура калібрування, а потім коригування передавальних характеристик. Можна, наприклад, визначити і виправити нелінійність, пов'язану зі статичною передавальною характеристикою вимірювального датчика в кожному каналі [2,4]. Також можна визначити і виправити неідентичність і нерівномірність частотних характеристик у каналах вимірювання [4].

Вхідні кола в комутаторах аналогових сигналів безпосередньо взаємодіють з датчиками фізичних величин. Тому вони повинні бути спроектовані відповідно до типу вхідної лінії та комутаційного з'єднання. Вхідні обмежувальні резистори або захисні діоди використовуються для забезпечення необхідного захисту від перенапруги в симетричних колах. У деяких випадках для гальванічної розв'язки входів використовуються оптоелектронні та трансформаторні схеми. Ці додаткові захисні елементи можуть мати значний вплив на параметр точності вимірювання. Однак, при підключенні схем захисту до вхідного комутатора, їх параметри міжканальності та неідентичності також враховуються і розглядаються нарівні з параметрами вхідних датчиків [2,5].

Основною функцією програмованого підсилювача в системі змінного струму є перетворення диференціального вхідного сигналу з видаленням синфазної складової, а також узгодження динаміки діапазону вхідного сигналу з діапазоном напруги як ФНЧ, так і DSP. Програмовані підсилювачі зазвичай базуються на диференціальній схемі, що складається з трьох операційних каскадів. Однак вони також можуть базуватися на підсилювачах струму, які мають симетричні проміжні та буферні каскади. Такі підсилювачі струму є найефективнішими і мають найкращу частотну характеристику. Вони також можуть бути використані для високошвидкісних вимірювань в АЦП, що самокорегуються.

У процедурі калібрування необхідно визначити і врахувати для кожного коефіцієнта підсилення програмованого підсилювача як його реальний коефіцієнт підсилення, так і зсув нуля. Це можна зробити, підключивши роз'єм аналога



до нульової шини і виходу з еталонного ЦАП. Для калібрування різних коефіцієнтів підсилення використовується процедура "зверху-вниз". Спочатку підсилювач має бути відкалібрований для більшого вхідного діапазону - помножувач опорного сигналу ЦАП створює напругу, амплітуда якої може бути встановлена за допомогою DON. ЦАП виробляє менший сигнал опорної напруги, значення якого можна точно виміряти спочатку за допомогою каліброваного коефіцієнта підсилення підсилювача. Потім програмований підсилювач перемикається на менший діапазон вхідного сигналу, де знову вимірюється опорна напруга.

Під час перетворення та обробки результати налаштування використовуються для корекції кожного каналу в оксимо АЦП. При цьому враховуються параметри визначеної реальної статичної характеристики передачі та реальної амплітудно-фазо-частотної характеристики. Система АЦП може бути адаптована до спектру, рівня та розмаху вхідних сигналів у кожному каналі на основі як апаратних компонентів (виходи з НП та АЦП), так і програмного аналізу (в ПК та сигнальному процесорі).

Застосування методів калібрування статичних та динамічних характеристик у багатоканальній АЦ-системі обробки низькочастотних сигналів повинно забезпечити такі основні технічні параметри:

- кількість вимірювальних каналів - 4;
- кількість каналів з регулюванням напруг і струмів для живлення активних сенсорів - 4;
- тип аналогових вхідних сигналів – симетричний;
- діапазон напруг аналогових вхідних сигналів – від 0,1 В до 1 В;
- діапазон частот вхідних сигналів – від 1 Гц до 10000 Гц;
- динамічний діапазон, обмежений відношенням сигнал / шум – не менше 126 дБ;
- спектральна роздільна здатність – не менше 140 дБ;
- нелінійні спотворення – не більше 0,003%;
- нерівномірність АЧХ та не лінійність ФЧХ каналів – не більше 0,03 дБ 0,1°;

- інтерфейс даних – USB 2.0 та Bluetooth;
- напруга живлення  $+5\text{ В} \pm 10\%$  від порту USB або від акумулятора;
- конструктивне виконання – на основі друкованої плати розміром не більше  $125\text{ мм} * 125\text{ мм} * 30\text{ мм}$ .

Отже, в результаті структурно-параметричного аналізу, проведеного в цьому підрозділі, було отримано наступне:

- показана можливість поліпшення статичних та динамічних властивостей АЦ-систем для обробки низькочастотних сигналів за допомогою АЦП і ЦАП на основі НПСЧ;

- вказані шляхи підвищення точності та швидкості АЦ-систем шляхом розширення принципів корекції статичних та автокомпенсації динамічних помилок АЦП і ЦАП на інші функціональні блоки вимірювальних каналів;

- запропоновано узагальнену структурну схему АЦ-системи для вимірювання та аналізу низькочастотних сигналів на основі самокаліброваних АЦП та ЦАП;

- сформульовано вимоги до найважливіших технічних параметрів самокаліброваної 4-канальної АЦ-системи для обробки низькочастотних сигналів.

Досягнення вказаних технічних параметрів підтверджується результатами попередніх експериментальних досліджень та параметрами подібних виробів та систем, розроблених авторами [18].

### 1.3.2 Вимоги до складу, структури та технічних параметрів АЦ-систем на основі самокаліброваних АЦП і ЦАП

Використання мікрофонних решіток є перевіреним методом візуалізації акустичного поля в системах моніторингу місцевості. Фізичний принцип використання невеликої кількості мікрофонів для побудови масиву через нерівномірний розподіл фокальних точок може призвести до розрідженості даних. Реконструкція звукового поля за неповними даними може бути виконана шляхом ретроспективного поширення звукового тиску шляхом розв'язання оберненої задачі [9]. Для підвищення роздільної здатності масиву необхідно знайти потужність і

координати всіх джерел звуку, якщо їх розташування і кількість невідомі. Модель поля базується на фундаментальних законах передачі звукової енергії. Для розв'язання оберненої задачі реконструкції звукового поля класичні методи регуляризації застосовні лише тоді, коли кількість джерел невелика. Коли дані є невизначеними, для оцінки параметрів джерел можна використовувати статистичні методи. Ці методи є обчислювально дорогими і вимагають багато дорогих експериментів для визначення потужності джерел.

Збільшенням кількості вимірювань вирішується проблема неповноти даних. Оцінки параметрів джерел потім визначаються методом формування променя. Експерименти відрізняються від класичного формування променя тим, що вони проводяться при різних розташуваннях решітки з нерівномірним розподілом мікрофонів. Це робиться для того, щоб знайти ідеальний режим і положення решітки. У роботі послідовності розташування решітки навколо джерел вибираються випадковим чином, або проводяться експерименти з детермінованими шаблонами для позицій вимірювань.

Введено поняття вибору стаціонарної області на основі результатів з декількох позицій вимірювання. На основі узгодженості мод з найбільш часто повторюваних спостережень можна вивести процедуру формування променя з діаграмою спрямованості, яка залежить від відстані від центру решітки. Збільшуючи поріг звуку, концентруючись на певному класі джерел шуму [2,20], можна зменшити кількість експериментів, одночасно збільшуючи частоти реконструкції. Методи Монте-Карло використовуються для пошуку оптимального розташування сенсорів у роботі [20].

Статистичні методи виведення використовуються для обробки експериментальних даних з метою визначення положення джерел та їхнього внеску в розподіл загальної звукової енергії. Рівні звукової енергії, що генеруються джерелами звуку, представлені у вимірах в кожній точці. Метод ґрунтується на максимізації функції правдоподібності, яку можна перекласти як ступінь узгодженості статистичних моделей і даних вимірювань. Для оцінки параметрів джерел шуму в використовується алгоритм максимізації математичного сподівання,

який ітеративно максимізує критерій шляхом послідовного оцінювання внеску кожного джерела в загальний розподіл звуку в полі. Для визначення точності оцінки параметрів важливо, щоб дані були розрідженими. Загалом, внесок кожного джерела визначається за допомогою імітаційного моделювання. Це передбачає видалення поля, з якого генеруються всі джерела, доки воно не перестане бути зашумленим .

Байєсівське фокусування - це уточнення акустичних зображень за допомогою байєсівських міркувань [2, 21]. Байєсівська реконструкція є більш точною, якщо розподіл джерел відомий апріорі. Шляхом підбору даних до моделі поширення енергії можна визначити місцезнаходження джерела звуку. Розв'язання обернених задач дозволяє оцінити параметри сканування. Потім функції налаштування системи оновлюються з використанням оцінених параметрів. Дорога статистично оптимізована реконструкція поля. Вартість експериментів зростає зі збільшенням апаратних потреб, що дозволяє автоматично підбирати параметри формування променя емпіричним шляхом відповідно до розміру та форми масиву мікрофонів.

Багаторазова ініціалізація джерел звуку неможлива при обробці потоків звукових даних. У цьому випадку для отримання тривимірної моделі використовується альтернативна технологія. Відокремлення джерел, що знаходяться близько одне від одного, з точки зору обчислювальних витрат не є найкращою ідеєю, оскільки надмірна деталізація не покращить зображення реального явища в екстатичній області. Для досягнення тривимірної візуалізації складних акустичних ситуацій ми використовуємо метод еквівалентних джерел. Цей метод відтворює звуковий тиск у віртуальних об'ємах за допомогою методу найменших квадратів. Можна спростити просторову реконструкцію, щоб краще відповідати реальним сценаріям вимірювань, без необхідності знати всі параметри джерела.

Метод еквівалентного джерела створює моделі звукового поля з інтервальними або нечіткими значеннями параметрів за наявності неповних даних. Методи нечіткої кластеризації, засновані на аналізі щільності піків у звуковому

полі, використовуються для оцінки розташування джерел [21]. Для моделювання внеску одного або декількох джерел можна використовувати кореляційну матрицю. Після цього дані коригуються.

Для відновлення параметрів в області групових ривків використовуються методи ре-регуляризації [2, 10]. Кластерний аналіз використовується для генерації можливих сценаріїв, які можуть відтворювати реалістичні акустичні ситуації. Потім здійснюється підбір генетичних джерел, які забезпечать мінімальну відстань між модельованими і спостережуваними рівнями звуку. Обчислювальна робота, необхідна для реконструкції звукового поля, згенерованого декількома джерелами, є значною, оскільки вона зводиться до вибору та створення матриць взаємозв'язків, які визначають розподіл рівнів звуку на полі.

## **2 АНАЛІЗ ТЕХНОЛОГІЇ ОБРОБКИ НИЗЬКОЧАСТОТНИХ ЗВУКОВИХ СИГНАЛІВ ПРИ БАГАТОКАНАЛЬНОМУ СКАНУВАННІ В СИСТЕМІ МІКРОФОННОЇ РЕШІТКИ**

### **2.1 Аналіз методів побудови і принципів функціонування мікрофонних решіток**

Сучасні системи збору та обробки акустичних сигналів покладаються на мікрофонні решітки. Ці мікрофонні решітки застосовуються у багатьох сферах, включаючи розпізнавання мови, слухові апарати та військову техніку. Основна функція мікрофонних решіток - поліпшення спрямованості і зменшення шуму.

Масив складається з декількох мікрофонів, розташованих у певному порядку. Фазові затримки між звуковими хвилями, що надходять до мікрофонів, використовуються для досягнення просторової селекції. Потім комбінатор може об'єднати всі звукові сигнали і створити спрямовану діаграму спрямованості.

Зважування відіграє ключову роль у зміні фази та амплітуди сигналів, що надходять з різних мікрофонів. Програмне забезпечення виконує це регулювання, дозволяючи отримувати різні діаграми спрямованості без фізичної зміни розташування або конструкції масиву.

Зауважте, що електронні сигнали, які генеруються мікрофонами, не просто ізолюють звуки в одному напрямку, а надають дані про всі напрямки звуку. Подальша обробка електронних сигналів дозволяє ізолювати бажані звуки і зберігає ефективність системи в різних умовах.

Broadside, або Endfire - дві основні структури для мікрофонних решіток. Вони дуже схожі на параболічні рефлектори, що використовуються в мікрофонах або трубчастих спрямованих мікрофонах.

У таблиці 2.1 описано структуру та функції мікрофонних решіток. Вони мають велике значення для високотехнологічних застосувань, які вимагають найкращого збору та обробки акустичних даних.

Таблиця 2.1 — Порівняння видів мікрофонних решіток

Вид решітки	Переваги	Недоліки
Broadside	Плоска геометрія. Проста реалізація обробки. Можливість управління напрямком променя.	Менше придавлення поза віссю МР. Мінімальна відстань між мікрофонами та їх велика кількість, що необхідно для запобігання просторовому витоку.
Endfire	Краще придавлення поза віссю. Менший загальний розмір.	Неплоска (об'ємна) геометрія. Більш складна обробка. Придавлення корисного сигналу в діапазоні низьких частот. Для двовимірних решіток формування променя можливо тільки в горизонтальному напрямку (площині решітки).

Основний алгоритм, який використовується для обробки сигналів у мікрофонній решітці (МР) для вилучення прямих звуків об'єктів, відомий як алгоритм прямого звуку (Direct Sound). Цей алгоритм призначений для посилення сигналів, що відповідають цільовому об'єкту, шляхом додавання мікрофонних сигналів з різними фазовими зсувами (затримками) і вирівнювання фаз для частот, що надходять з обраного напрямку, який є напрямком руху цільового об'єкта.

Алгоритм прямого звуку по суті фокусується на посиленні сигналів, згенерованих звуковими хвилями, що надходять з певного напрямку, який нас цікавить. Цей процес схожий на "фокусування" на потрібному джерелі серед різних фонових звуків і шумів. Фазові зсуви, що застосовуються під час підсумовування сигналів, мають вирішальне значення для досягнення такого спрямованого посилення.

Визначення відповідного фазового зсуву може базуватися на наближенні "далекого поля" або наближенні "ближнього поля", залежно від відстані між мікрофонною решіткою і джерелом звуку.[8]

$$L > 2D^2 / \lambda, \quad (2.1)$$

де  $L$  — відстань від джерела до МР;

$D$  – лінійний розмір (апертура) МР;

$\lambda$  – довжина звукової хвилі.

Мікрофонний масив - це система з декількох мікрофонів, які розташовані так, щоб захоплювати акустичний сигнал з різних напрямків. Діаграму спрямованості можна використовувати для характеристики функціональності масиву, оскільки вона графічно показує, як вони реагують на різні сигнали.

Діаграма спрямованості: Це графічне відображення передавальної функції МР. Ця функція визначає, як антени (в даному випадку мікрофони) реагують на різні кути акустичного сигналу. Ця функція має вирішальне значення для розділення сигналів і придушення небажаних перешкод.

Геометрія МР - це те, що визначає форму і характеристики діаграми спрямованості. Вона залежить від розташування і кількості мікрофонів. Ви розглядаєте восьмиелементну решітку мікрофонів з рівною відстанню між кожним елементом. Апертура становить 35 см, а відстань між мікрофонами - 5 см. Конфігурація, що використовується тут, призначена для аналізу сигналів у діапазоні (1, 2, 4 кГц).

Діаграми спрямованості показано на рисунку 2.1. Горизонтальна вісь представляє кут, під яким приходить звук, тоді як вертикальна вісь вказує величину вихідного сигналу на лінійній основі. Ці діаграми можна використовувати для визначення того, наскільки добре МР реагує на різні сигнали в обраних частотах.



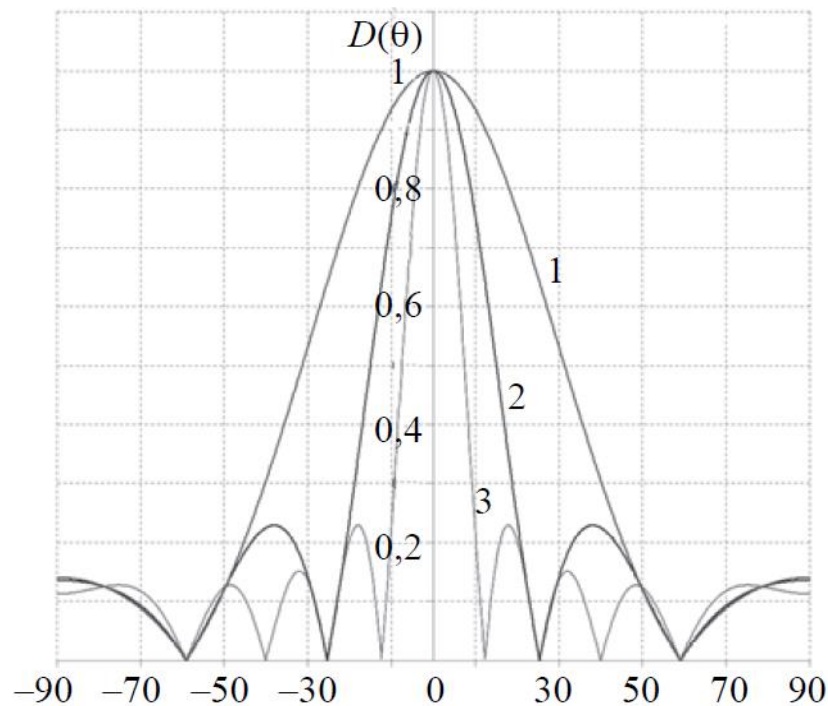


Рисунок 2.1 — Залежність діаграми спрямованості еквідистантної восьмиелементної мікрофонної решітки: 1 – 1 кГц; 2 – 2 кГц; 3 – 4 кГц

У контексті алгоритму прямого звуку і мікрофонних решіток кілька важливих параметрів відіграють значну роль у визначенні продуктивності системи. До цих параметрів відносяться:

- ширина головної пелюстки на рівні 3 дБ — центральна, найбільш чутлива область діаграми спрямованості на діаграмі спрямованості, ширина головної пелюстки на рівні 3 дБ відноситься до кутового проміжку центральної області, де рівень сигналу знаходиться в межах 3 дБ від максимального, вужча головна пелюстка вказує на краще спрямоване фокусування;

- рівень бічних пелюсток — вторинні пелюстки діаграми спрямованості, що знаходяться на відстані від головної пелюстки, рівень бічних пелюсток має вирішальне значення, оскільки високий рівень бічних пелюсток може призвести до чутливості до сигналів з небажаних напрямків, зменшуючи ефективність шумозаглушення, низький рівень бічних пелюсток сприяє кращій просторовій селективності;

— індекс спрямованості — міра здатності мікрофонної решітки придушувати шум з усіх напрямків, крім бажаного, вищий індекс спрямованості свідчить про краще придушення шуму і краще фокусування на цільовому джерелі;

— частотна залежність — властивості мікрофонної решітки, включаючи розмір головної пелюстки, впливає частота оброблюваних сигналів, різні частоти можуть призвести до зміни характеристик спрямованості решітки.

У таблиці 2.2 наведено залежність розміру головної пелюстки від частоти сигналу для 8-елементного лінійного мікрофона з розмірами 8 см x 5 см.

Таблиця 2.2 — Ширина основної пелюстки діаграми направленості залежно від частоти

Частота, кГц	1 кГц	2 кГц	3 кГц	4 кГц	5 кГц	6 кГц	8 кГц
Ширина променю	±30	° ±15°	±10°	±7,5°	±6	° ±5°	±3,75°

Кутова роздільна здатність МР залежить від апертури (лінійного розміру МР), довжини хвилі (частоти сигналу), напрямку променя і методу обробки сигналу [8]:

$$\Delta\alpha = 1,22 / \cos(\alpha) \times \lambda / D, \quad (2.1)$$

де  $\Delta\alpha$  – кутова роздільна здатність (радіан);  $D$  – апертура;  $\lambda$  – довжина хвилі;

$\alpha$  – кут напрямку променю.

Просторова роздільна здатність в площині, що знаходиться на дистанції  $L$ , залежить від кута напрямку променю наступним чином [9]:

$$X(\Delta\alpha) = 1,22 / \cos^3(\alpha) \times \lambda L / D. \quad (2.2)$$

У таблиці 2.3 наведено інформацію про просторову роздільну здатність як функцію кута променя, частоти сигналу та 8-елементної МА з розмірами 8 см x 5 см.

Аналіз проводився в площині на відстані 10 метрів. Цей параметр є важливим з точки зору масивів мікрофонів, оскільки він стосується саме здатності системи розділяти звуки, що надходять під різними кутами. Для того, щоб визначити, як мікрофон може розрізнити просторові деталі, необхідно враховувати кут нахилу променя, а також частоту сигналу.

Ключовий вплив на вимірювану просторову роздільну здатність має відстань вимірювання і розмір мікрофонів (8 см x 5 см), а також місце, де вони розміщені (10 метрів від лінійки).

Таблиця 2.3 — Просторова роздільна здатність (м) в площині, розташованій на відстані 10 м від мікрофонної решітки

Частота, кГц	Кут направлення променя						
	0°	10°	20°	30°	40°	50°	60°
1	11,86	12,10	14,23	18,26	26,33	44,59	94,88
2	5,93	6,05	7,11	9,13	13,16	22,39	47,44
4	2,97	3,02	3,56	4,56	6,58	11,15	23,72
6	1,98	2,02	2,37	3,04	4,39	7,43	15,81

Задача оптимізації форми мікроскопічної решітки є складною і зазвичай вирішується за допомогою статистичного моделювання. На відміну від деяких інших проблем з аналітичними рішеннями, оптимізація геометрії мікрофонної решітки передбачає використання статистичних методів для визначення оптимальної конфігурації. Процес є ітеративним і часто включає в себе моделювання та коригування.

Прямого аналітичного рішення не існує. Оптимізація геометрії МР за різними критеріями може призвести до структур з нерівномірним розміщенням мікрофонів. Ця гнучкість дозволяє розробникам налаштовувати масив мікрофонів

для задоволення конкретних потреб і застосувань, включаючи підвищення чутливості до напрямку, зменшення шуму або поліпшення просторової чіткості.

## 2.2 Конструктивно-схемотехнічні рішення мікрофонних решіток

Мікрофонні решітки можна будувати як за рівномірними, так і нерівномірними схемами. Алгоритм отримання заданих затримок у каналах спрощується при використанні неоднорідної решітки. На рис. 2.1 показано практичну реалізацію мікрофонної решітки за системою 1-2-1. Відсні між мікрофонами в даній конфігурації становлять 100 мм, 200 мм і 100 мм. Конкретна конфігурація вибрана для спрощення алгоритмічних розрахунків затримок між сигналами в різних каналах.

Значення модуля 100 мм відповідає періоду довжини звукової хвилі при швидкості звуку 340 м/с і частоті 3300 Гц. На всіх частотах нижче 3300 Гц помилкові діаграми спрямованості не з'являться. Важливо підтримувати точність і уникати інтерференції діаграми спрямованості.

Включення нерекурсивного цифрового фільтра в усі канали може обмежити спектр вхідного сигналу частотами до 3300 Гц. Метод фільтрації гарантує, що низькочастотні сигнали, які можуть не мати суттєвого впливу на якість розпізнавання, будуть ефективно зменшені.

Неоднорідний масив мікрофонів з діаграмою направленості [9] 1-2-1, як показано на відстанях від 100 мм до 200 мм на 100 мм, пропонує переваги з точки зору алгоритмічної складності та уникнення помилкових променів. Цифрові фільтри допомагають оптимізувати вихідний сигнал масиву, керуючи спектром вхідних сигналів.



Рисунок 2.1 — Конструкція 8-канальної мікрофонної решітки з центральними проміжками

Після фільтрації можливе зменшення частоти дискретизації до частоти 8000 Гц, що значно зменшить об'єм обчислень при визначенні амплітуд сигналів в координатних точках контрольованої території та пришвидшить виконання аудіодетекції сигналів[10].

### 2.3 Розрахункові співвідношення для сканування й обробки сигналів

Для розробки модулів сканування необхідно вирішити ряд геометричних задач, пов'язаних з розміщенням мікрофонних решіток А і В та усіх точок С (рис. 2.5) перетину променів їх діаграм направленості на контрольованій місцевості.

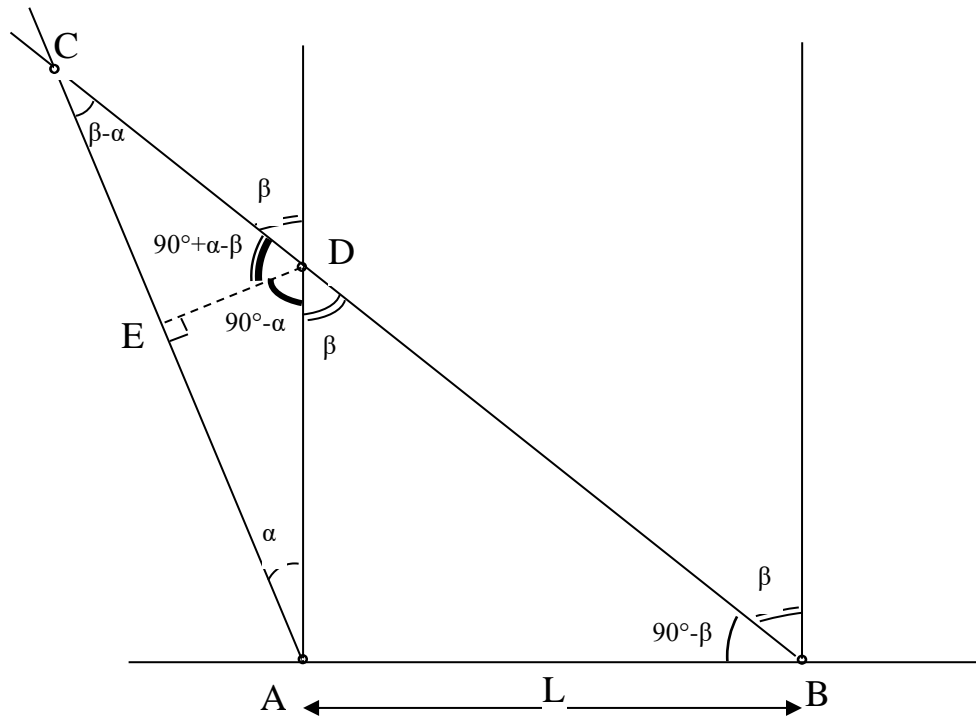


Рисунок 2.5 — Знаходження відстаней до точки С для «лівої» частини поля

Однією із задач які потрібно вирішити являється знаходження відстаней до точки перетину промінів.

Дано:  $|AB| = L$  – відстань між мікрофонами, для прикладу,  $L=20$  м;

$\alpha$  – кут напрямку променю з решітки в точці А,  $\alpha \in [0^\circ; 75^\circ]$ ;

$\beta$  – кут з точки В,  $\beta \in [0^\circ; 75^\circ]$ .

Знайти: відстані від решіток до точки С перетину променів  $|AC|$ ,  $|BC|$ ;

різницю відстаней  $\Delta = |BC| - |AC|$ .

Рішення задачі проводим наступним чином.

Знаходимо суміжні і різницеві кути в трикутниках ABC, ADB, ADC на основі заданих кутів  $\alpha$  і  $\beta$  (рис. 2.6);

$$|BD| = L / \sin(\beta);$$

$$|AD| = |BD| * \cos(\beta) = L / \tan(\beta);$$

$$|AE| = |AD| * \cos(\alpha);$$

$$|ED| = |AD| * \sin(\alpha);$$

$$|CD| = |ED| / \sin(\beta - \alpha);$$

$$\begin{aligned}
|CE| &= |CD| * \cos(\beta - \alpha) = |ED| / \tan(\beta - \alpha); \\
|AC| &= |AE| + |CE| = |AD| * \cos(\alpha) + |ED| / \tan(\beta - \alpha) = \\
&= |AD| * \cos(\alpha) + |AD| * \sin(\alpha) / \tan(\beta - \alpha) = |AD| * (\cos(\alpha) + \sin(\alpha) / \tan(\beta - \alpha)) = L / \tan(\beta) * (\cos(\alpha) + \sin(\alpha) / \tan(\beta - \alpha)),
\end{aligned}$$

отже, відстань від решітки А до точки С є рівною  $L / \tan(\beta) * (\cos(\alpha) + \sin(\alpha) / \tan(\beta - \alpha))$  [5];

$$\begin{aligned}
|BC| &= |BD| + |CD| = L / \sin(\beta) + |ED| / \sin(\beta - \alpha) = L / \sin(\beta) + |AD| * \sin(\alpha) / \sin(\beta - \alpha) = L / \sin(\beta) + L / \tan(\beta) * \sin(\alpha) / \sin(\beta - \alpha) = L / \sin(\beta) * (1 + \cos(\beta) * \sin(\alpha) / \sin(\beta - \alpha)),
\end{aligned}$$

отже, відстань від решітки А до точки С розраховується за формулою (2.2);

$$|BC| = L / \sin(\beta) * (1 + \cos(\beta) * \sin(\alpha) / \sin(\beta - \alpha));$$

Різниця відстаней від точки С до решіток А і В для врахування потрібної затримки звукового сигналу [5]

$$\Delta = |BC| - |AC|$$

Ці завдання вимагають обчислення відстаней і координат між точками перетину променів в межах певної області.

Калькулятор відстаней:

— відстань залежить від відстані (L) між решітками А і В, а також від кутів нахилу променів відносно прямої лінії, відстань до точки С залежить як від відстані між двома решітками (L), так і від кута ( $\alpha$  або  $\beta$ ), який промені утворюють з прямою лінією;

— різниця буде використана для визначення часової затримки між решітками, затримка між двома решітками є важливим параметром для розуміння часового аспекту акустичного середовища.

Обчислення координат (X та Y $\alpha$ ):

— визначте координати всіх точок (X і Y) перетину променів під кутом  $\alpha$  або  $\beta$  зміни;

— координати будуть використані для створення акустичної карти, яка відображає рівні шуму на даній території;

— розрахунки проводяться для точок С, розташованих у правій частині контрольованого поля, це базується на конкретній конфігурації, показаній на рис. 2.6, ця конфігурація є "симетричною" до конфігурації на рис. 2.5, кути  $\alpha$  і  $\beta$  можуть бути "замінені" один на одного;

— у цьому симетричному розташуванні зміна знаку запізнення позначається точками, розташованими з правого боку.

Ці завдання вимагають як просторових, так і часових розрахунків. Для визначення часової затримки між решітками використовуються кути і геометрія променів. Для того, щоб побудувати карту акустичних полів в контрольованій зоні, просторова складова передбачає розрахунок координат точок перетину. Розрахунки допоможуть зрозуміти і змодельовати поведінку звукової хвилі в заданому середовищі.

Друга задача полягає в розрахунку координат  $X$  і  $Y$  для усіх точок С перетину променів за заданим кроком зміни кутів  $\alpha$  і  $\beta$ . Це необхідно для побудови акустичного поля звукових рівнів на контрольованій місцевості.

Для точок С, розміщених на «правій» частині контрольованого поля, розрахунки здійснюються згідно з рис. 2.6, що «симетричний» до попередньо розглянутого рис. 2.5. Кути  $\alpha$  і  $\beta$  «замінюють» один одного. Знак затримки змінюється.

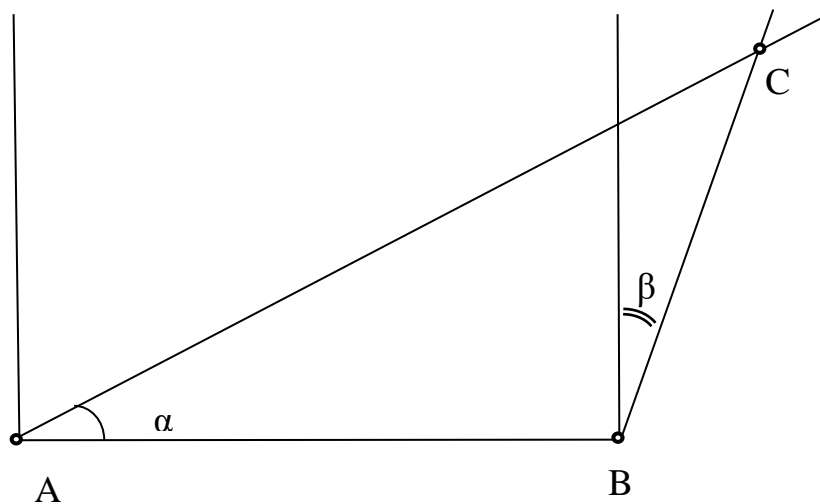


Рисунок 2.6 — Знаходження відстаней до точки С для «правій» частини поля



У випадку повороту променів решіток А і В на однакові кути, вони не перетинаються, а рішення згідно з рис. 2.6 спрощується.

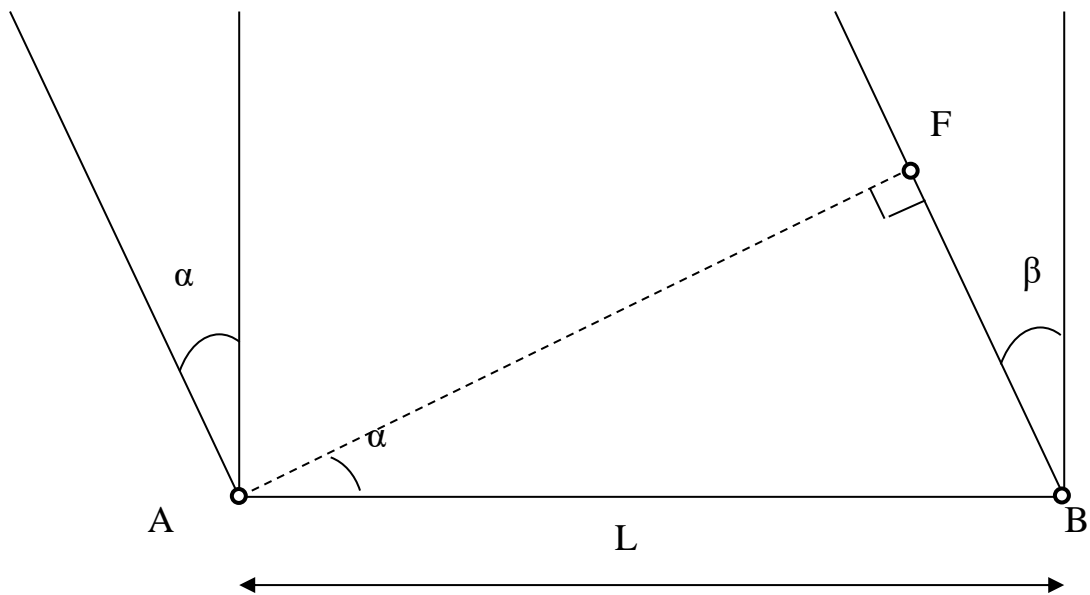


Рисунок 2.7 — Знаходження різниці відстаней при повороті на однакові кути

Якщо кути однакові  $\alpha = \beta$ , то обчислюється тільки різниця відстаней – відрізок  $|BF| = L \cdot \sin(\alpha)$

На основі розрахованих співвідношень, для  $L = 20$  м, і кроків кутів  $\alpha$  і  $\beta$  по  $5^\circ$  в рамках від  $+75^\circ$  («ліва» частина) до мінус  $75^\circ$  («права» частина) розраховано відстані  $|AC|$ ,  $|BC|$  і їх різницю  $\Delta$ .

Другою задачею являє собою розрахунок координатної матриці в значеннях  $X$  і  $Y$  для кожної точки перетину променів.

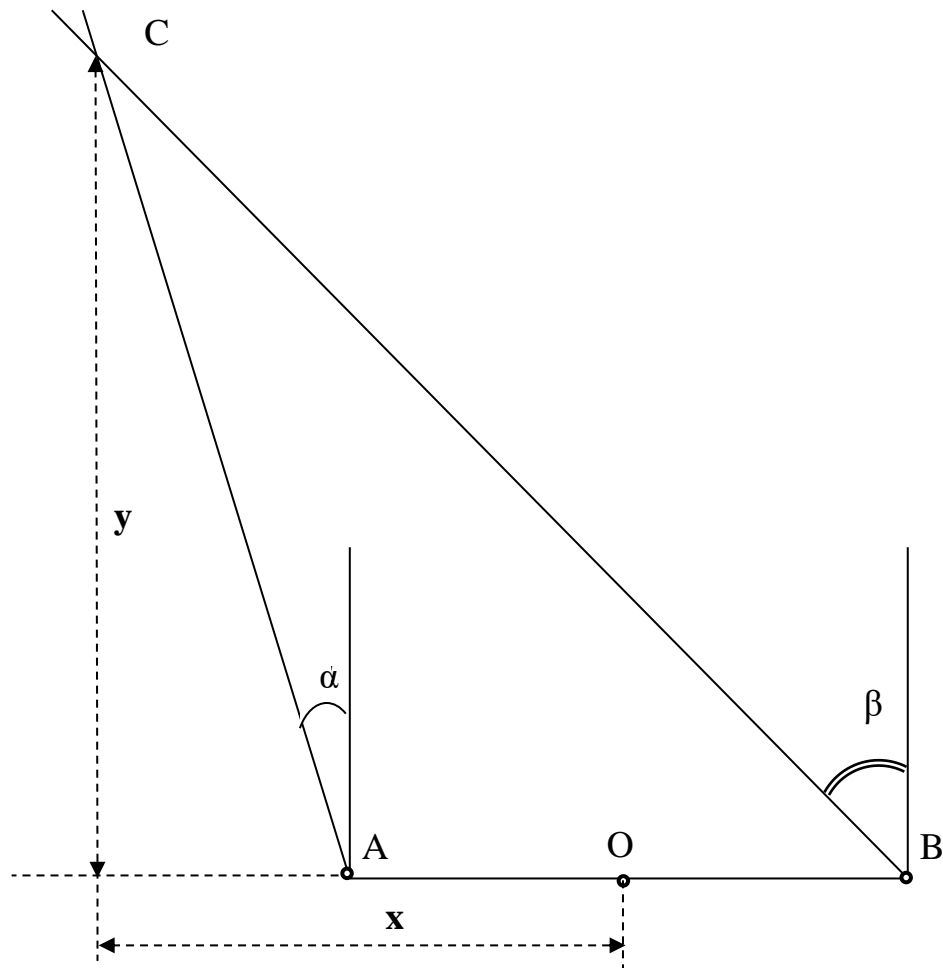


Рисунок 2.8 — Знаходження координат X і Y точки C

За умовою задачі дано  $|AB| = L = 20$  м,  $|AO| = |OB| = 10$  м,  
 $\alpha, \beta, |AC|, |BC|$

Потрібно знайти координати  $x, y$  кожної значущої точки на перехресті ліній з кутами  $\alpha$  і  $\beta$ , в рішенні використовуються формули (2.5, 2.6);

Рішення задачі проводим наступним чином.

$$|AG| = |AC| \cdot \sin(\alpha), \text{ де } X = |GO| = |AG| + L/2 = |AC| \cdot \sin(\alpha) + L/2;$$

$$Y = |GC| = |AC| \cdot \cos(\alpha);$$

Виконати п.1 і п.2 для всіх  $\beta [5^\circ - 75^\circ]$  з кроком  $5^\circ$ ;

Для  $\alpha = \beta$  точки повинні знаходитись на сторонах квадрата  $250\text{м} \times 250\text{м}$ , що відповідає розмірам контрольованої території місцевості, як показано на рис. 2.9:

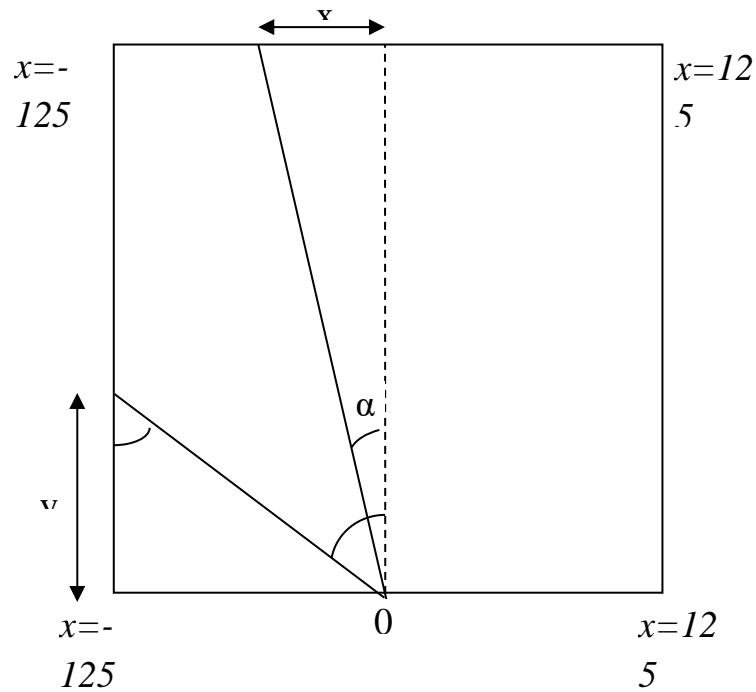


Рисунок 2.9 — Знаходження координат  $X$  і  $Y$  точок, що знаходяться на сторонах квадрату контрольованої території

На основі розрахованих співвідношень, для  $L = 20$  м і для кожної точки перетину променів  $\alpha$  і  $\beta$  по  $5^\circ$  в рамках від плюс  $75^\circ$  («ліва» частина) до мінус  $75^\circ$  («права» частина) розраховано координати  $X$  і  $Y$  усіх точок  $C$  [11].

### **3 РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ДЛЯ РОЗРАХУНКУ АКУСТИЧНИХ ПОВЕРХОНЬ В БАГАТОКАНАЛЬНИЙ АНАЛОГО-ЦИФРОВІЙ СИСТЕМІ**

#### **3.1 Огляд технології CUDA**

CUDA - це платформа паралельних обчислень і модель програмування, розроблена компанією Nvidia для полегшення обчислень загального призначення на власних графічних процесорах (GPU). Ця платформа дозволяє розробникам прискорювати обчислювальні програми, використовуючи можливості паралельних обчислень на графічних процесорах. Незважаючи на існування альтернативних API GPU, таких як OpenCL, та конкуренцію з боку інших компаній, таких як AMD, поєднання CUDA та графічних процесорів Nvidia займає провідні позиції в різних сферах застосування, зокрема в глибокому навчанні, і лежить в основі одних з найшвидших комп'ютерів у світі. Історична еволюція відеокарт бере свій початок з перших днів появи персональних комп'ютерів, а першим прикладом є монохромний адаптер дисплея від IBM 1981 року. У 1988 році ATI запропонувала 16-бітну 2D карту VGA Wonder, а в 1996 році 3dfx Interactive представила 3D графічний прискорювач для більш плавної роботи в іграх, в той час як Nvidia відзначилася з GeForce 256 в 1999 році, який визнаний першим графічним процесором. Спочатку графічні процесори асоціювалися в першу чергу з іграми, і лише згодом вони знайшли застосування в математиці, науці та інженерії.

У 2003 році відбувся прорив, коли група дослідників під керівництвом Яна Бака (Jan Buck) представила Brook - новаторську модель програмування, яка розширила мову C паралельними структурами даних. Згодом Бак приєднався до Nvidia і очолив запуск CUDA у 2006 році, що стало першим комерційним рішенням для обчислень на GPU загального призначення.

Конкурент CUDA, OpenCL, з'явився у 2009 році завдяки співпраці Apple та Khronos Group, яка мала на меті створити стандарт для гетерогенних обчислень, що виходив би за межі процесорів Intel/AMD з графічними процесорами Nvidia.

Незважаючи на свою загальну привабливість, OpenCL не досягла такого ж рівня успіху, як CUDA на графічних процесорах Nvidia. Багато фреймворків для глибокого навчання або не мають підтримки OpenCL, або надають її лише в якості другорядного фактору після реалізації підтримки CUDA. Протягом багатьох років CUDA постійно вдосконалювала та розширювала свої можливості, розвиваючись разом з розвитком графічних процесорів Nvidia. Починаючи з версії CUDA 9.2, використання декількох серверних графічних процесорів P100 може забезпечити приріст продуктивності до 50 разів у порівнянні з CPU [12].

### NVIDIA Tesla P100 Performance

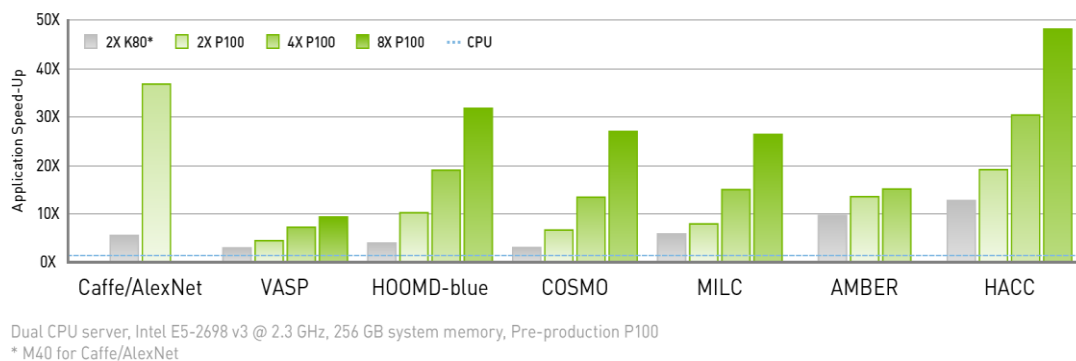


Рис.3.1 — Графік продуктивності процесора NVIDIA Tesla P100

Історична траєкторія зростання продуктивності однопотоківих процесорів, яка за законом Мура повинна була подвоюватися кожні 18 місяців, зіткнулася з уповільненням, і зараз зупинилася на рівні приблизно 10 відсотків на рік (як показано на рисунку 3.2). Це уповільнення можна пояснити фізичними обмеженнями, з якими стикаються виробники мікросхем, включаючи обмеження роздільної здатності маски мікросхеми, вихід мікросхем під час виробничого процесу та обмеження тактової частоти під час роботи, пов'язані з тепловим впливом.

Графічні процесори CUDA та Nvidia знайшли застосування в різних галузях, які вимагають високої продуктивності обчислень з плаваючою комою, як показано на рисунку вище. Розширений список цих додатків включає:

- обчислювальні фінанси;
- моделювання клімату, погоди та океану;
- наука про дані та аналітика;
- глибоке навчання і машинне навчання;
- оборона та розвідка;
- виробництво / АЕС (архітектура, інженерія та будівництво);
- медіа та розваги.

Використання CUDA та графічних процесорів Nvidia довело свою ефективність у розширенні обчислювальних можливостей у цих галузях, сприяючи про- ривам та інноваціям у сферах від фінансів до кліматичного моделювання та від глибокого навчання до медіа та розваг.

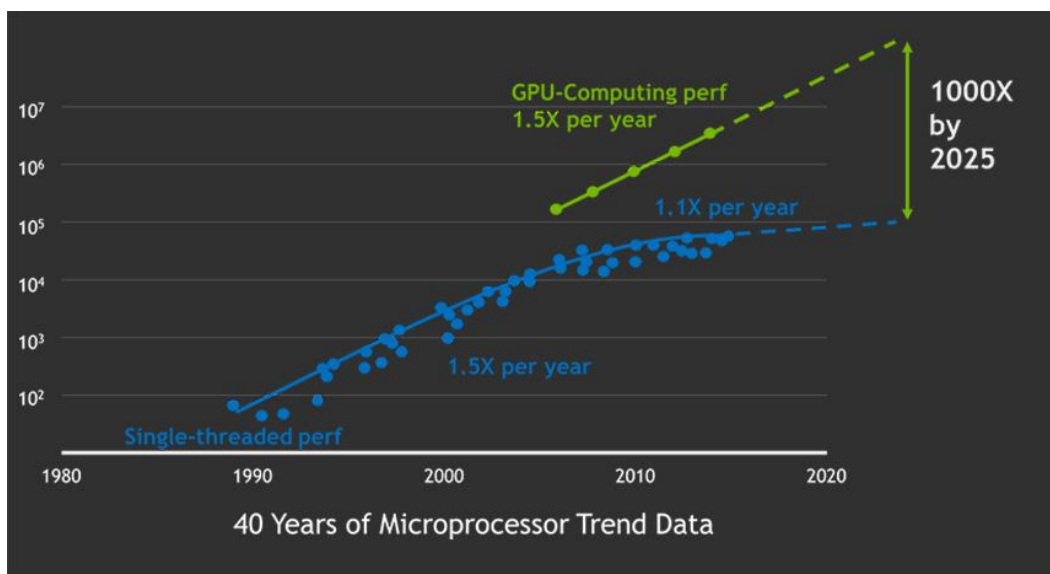


Рис.3.2 — Дані мікропроцесорів за 40 років

Глибоке навчання висуває значні вимоги до швидкості обчислень. Показовим прикладом є навчання моделей для Google Translate у 2016 році, де команди Google Brain і Google Translate виконували сотні щотижневих запусків TensorFlow на графічних процесорах. Щоб пришвидшити ці тренувальні про-

гони, вони придбали 2 000 серверних графічних процесорів від Nvidia. Важливість графічних процесорів у цьому контексті очевидна, оскільки без них тренувальні прогони розтягнулися б з тижнів або днів до місяців, щоб збігтися. Для розгортання цих моделей перекладу TensorFlow компанія Google представила спеціальний процесорний чіп, відомий як Tensor Processing Unit (TPU). Окрім TensorFlow, численні інші фреймворки для глибокого навчання (DL) значною мірою залежать від CUDA завдяки підтримці GPU. Серед них Caffe2, CNTK, Databricks, H2O.ai, Keras, MXNet, PyTorch, Theano та Torch. Більшість з цих фреймворків використовують бібліотеку cuDNN для глибоких нейромережових обчислень. Бібліотека cuDNN відіграє ключову роль у визначенні продуктивності фреймворків глибокого навчання, оскільки всі фреймворки, що використовують певну версію cuDNN, по суті, демонструють однакову продуктивність для еквівалентних сценаріїв використання. Оскільки CUDA і cuDNN розвиваються з кожною версією, будь-яка система глибокого навчання, що переходить на нову версію, відчуває відповідний приріст продуктивності. Різниця в продуктивності часто пов'язана з тим, наскільки добре фреймворки масштабуються до декількох GPU і декількох вузлів, не кажучи вже про відмінності між кадрами.

Інструментарій CUDA включає бібліотеки, засоби налагодження та оптимізації, компілятор, документацію та бібліотеку часу виконання для розгортання додатків. Ці компоненти підтримують різні області, включаючи глибоке навчання, лінійну алгебру, обробку сигналів та паралельні алгоритми. Хоча бібліотеки CUDA зазвичай розширюють підтримку всіх сімейств графічних процесорів Nvidia, оптимальна продуктивність спостерігається на останніх поколіннях, таких як V100, які можуть запропонувати до трьох разів більшу швидкість, ніж P100 для навчальних робочих навантажень (як показано на рисунку 3.3). Використання однієї або декількох бібліотек є найбільш простим методом використання потужності графічних процесорів, особливо коли необхідні алгоритми реалізовані у сумісній бібліотеці.

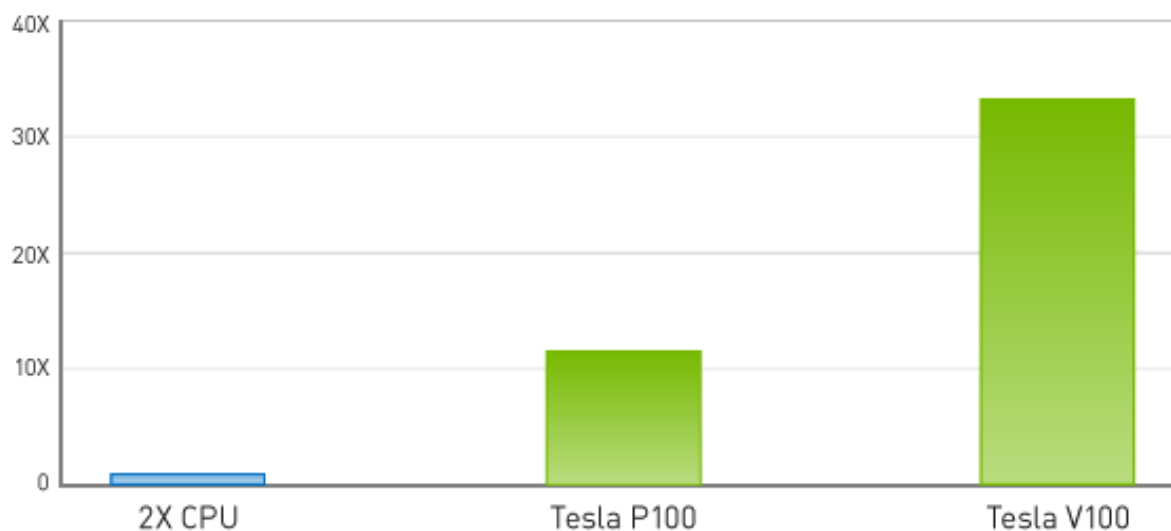


Рис.3.3 — Глибоке навчання нейронної мережі за декілька годин

У сфері глибокого навчання три видатні бібліотеки з GPU-прискоренням відіграють ключову роль:

— як вже згадувалося раніше, cuDNN слугує GPU-компонентом для численних платформ глибокого навчання з відкритим вихідним кодом. Він відіграє вирішальну роль у підвищенні продуктивності завдань глибокого навчання;

— TensorRT виділяється як високопродуктивний оптимізатор вихідних даних для глибокого навчання та середовище виконання. Воно дозволяє користувачам оптимізувати моделі нейронних мереж, калібрувати для низької точності з високою точністю і безперешкодно розгортати навчені моделі в різних середовищах, включаючи хмару, центри обробки даних, вбудовані системи та автомобільні платформи;

— DeepStream позиціонується як бібліотека відео-висновків, DeepStream фокусується на полегшенні ефективної обробки висновків для відео-додатків.

Крім глибокого навчання, кілька бібліотек, присвячених паралельним алгоритмам, слугують різним цілям:



— NCCL (Nvidia Collective Communications Library) бібліотека призначена для масштабування додатків на декілька графічних процесорів та вузлів, забезпечуючи ефективну комунікацію та координацію в паралельних обчислювальних середовищах;

— nvGRAPH призначена для аналізу паралельних графів, nvGRAPH надає основні функціональні можливості для роботи з графовими структурами даних у паралельних обчисленнях;

— Thrust являє собою бібліотеку шаблонів CUDA, засновану на стандартній бібліотеці шаблонів C++. Пропонуючи багату колекцію паралельних примітивів, таких як сканування, сортування та зведення, Thrust спрощує реалізацію паралельних алгоритмів.

У певних сценаріях випадаючі функції CUDA можна легко замінити еквівалентними функціями CPU. Наприклад, підпрограми GEMM (General Matrix Multiply) в BLAS (Basic Linear Algebra Subprograms) можна замінити на GPU-прискорені версії, звернувшись до бібліотеки NVBLAS.

Цей різноманітний набір бібліотек з GPU-прискоренням надає розробникам і дослідникам комплексний інструментарій для оптимізації, розгортання та масштабування своїх додатків у різних галузях паралельного та глибокого навчання.

## 3.2 Програмування з використанням CUDA технологій

При використанні звичайних API, незалежно від складності алгоритму, програмістам незмінно доводиться налаштовувати кожен складову графічного конвеєра. Ця реальність значно ускладнює використання GPU для задач загального призначення, оскільки навіть така, здавалося б, проста операція, як додавання двох матриць, вимагає виконання безлічі підготовчих команд. Отже, кілька рядків шейдерної програми можуть швидко перетворитися на сотні рядків додаткового коду.

Парадигма програмування, що використовується в CUDA, відрізняється від традиційних API тим, що повністю приховує тонкощі графічного конвеєра

від програміста. Ця абстракція дозволяє програмістам створювати програми у більш традиційний спосіб в рамках розширеної варіації мови C.

Крім того, CUDA надає програмісту зручну модель керування пам'яттю. Минули часи зберігання даних у 128-бітних текстурах, оскільки CUDA забезпечує прямий доступ до даних у пам'яті відеокарти.

NVIDIA CUDA включає в себе два різних API: високорівневий API, відомий як CUDA Runtime API, та низькорівневий API, який називається CUDA Driver API, як показано на рисунку 3.4. У випадках, коли використання низькорівневих функцій GPU є вкрай необхідним, програмісти можуть використовувати API драйверів, відмовляючись від API виконання.

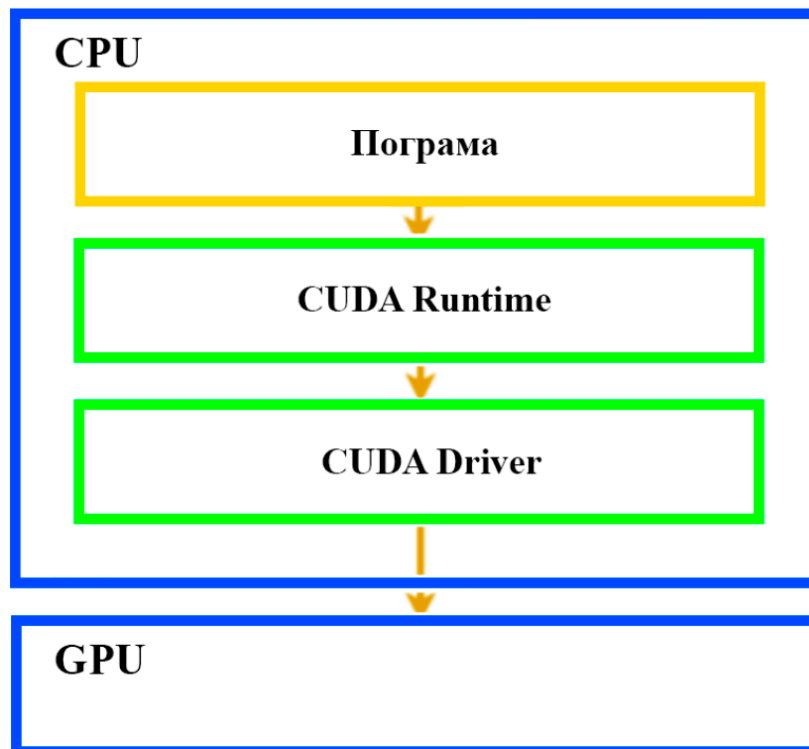


Рисунок 3.4 — Схема роботи CUDA API

Початковий етап адаптації існуючого алгоритму до CUDA включає ретельний аналіз для виявлення ділянок, придатних для розпаралелювання. Як правило, ці ділянки в алгоритмі CPU інкапсульовані в циклах або рекурсіях.

Повне перенесення алгоритму на GPU неможливе, оскільки GPU не має доступу до пам'яті комп'ютера та пристроїв вводу/виводу. Під час виконання

програми центральний процесор зберігає відповідальність за підготовку та постобробку даних, в той час як графічний процесор виконує найбільш обчислювально інтенсивні завдання. Графічний процесор виконує набір інструкцій, відомий як ядро.

Центральний процесор відповідає за створення та компіляцію ядер. Відеочіп, в свою чергу, бере скомпільоване ядро і генерує копії для кожного елемента даних. Ключова перевага технології CUDA в розробці програмного забезпечення полягає в тому, що кожне сформоване ядро виконує своє завдання у виділеному потоці.

Потоки GPU працюють групами по 32 екземпляри, а максимальна кількість екземплярів в одній задачі дорівнює кількості ядер CUDA у відеокарті. Отже, апаратне забезпечення визначає, які групи потоків будуть виконуватися кожен такт. Цей процес відбувається майже миттєво, на відміну від потенційно тривалих циклів перемикання на центральному процесорі.

На відміну від шейдерів, де дані представлені у вигляді чотирикомпонентних векторів, дані ядра є скалярними. Таке представлення є більш інтуїтивно зрозумілим для більшості неграфічних задач. Кожне ядро виконується на сітці блоків, і лише одна сітка може виконуватися на GPU в будь-який момент часу. Таке групування забезпечує оптимальну масштабованість. Якщо графічному процесору не вистачає ресурсів для одночасного виконання всіх блоків, вони виконуються послідовно. Такий підхід дозволяє розробникам абстрагуватися від занепокоєння щодо потужності пристрою, на якому буде працювати додаток.

У кожному додатку, розробленому з використанням NVIDIA CUDA, незалежно від його конкретного призначення, можна виділити кілька загальних етапів:

- підготовка пам'яті;
- конфігурація сітки та блоків;
- запуск ядра;
- отримання результатів і звільнення пам'яті.

### 3.3 Огляд компілятора ILGPU для використання CUDA в кодї C+

Щоб розпочати розробку програмного забезпечення на C+, використання сторонніх зовнішніх бібліотек стає необхідним для взаємодії з ядрами CUDA GPU. Видатною бібліотекою в цьому контексті є ILGPU.

ILGPU виділяється як компілятор just-in-time (JIT), створений для високопродуктивних додатків, що використовують обчислювальну потужність GPU та реалізовані на мовах на базі .Net. Примітно, що ILGPU повністю побудований на C# без залежностей від зовнішніх бібліотек, забезпечуючи основу для розробки GPU-орієнтованих додатків. Основна мета цього компілятора - об'єднати зручні для користувача аспекти C++ AMP з надійними характеристиками продуктивності CUDA. У межах ядра функції не мають анотацій (нагадуючи функції за замовчуванням у C#) і здатні працювати з різними типами даних. Процес компіляції дотримується принципів, показаних на рисунку 3.5.

Використання сторонніх бібліотек, таких як ILGPU, полегшує інтеграцію можливостей GPU у додатки на C#, долаючи розрив між синтаксисом мови та потужністю ядер графічного процесора CUDA. Розробники можуть використовувати зручність C#, одночасно розкриваючи потенціал високопродуктивних обчислень на GPU за допомогою ILGPU.

Даний компілятор дозволяє виконання коду на графічному процесорі за допомогою NVIDIA Nsight Visual Studio Edition. Також реалізує розширенні функції C#, включаючи віртуальні функції та загальні засоби паралельного багатопоточного обчислення[15].

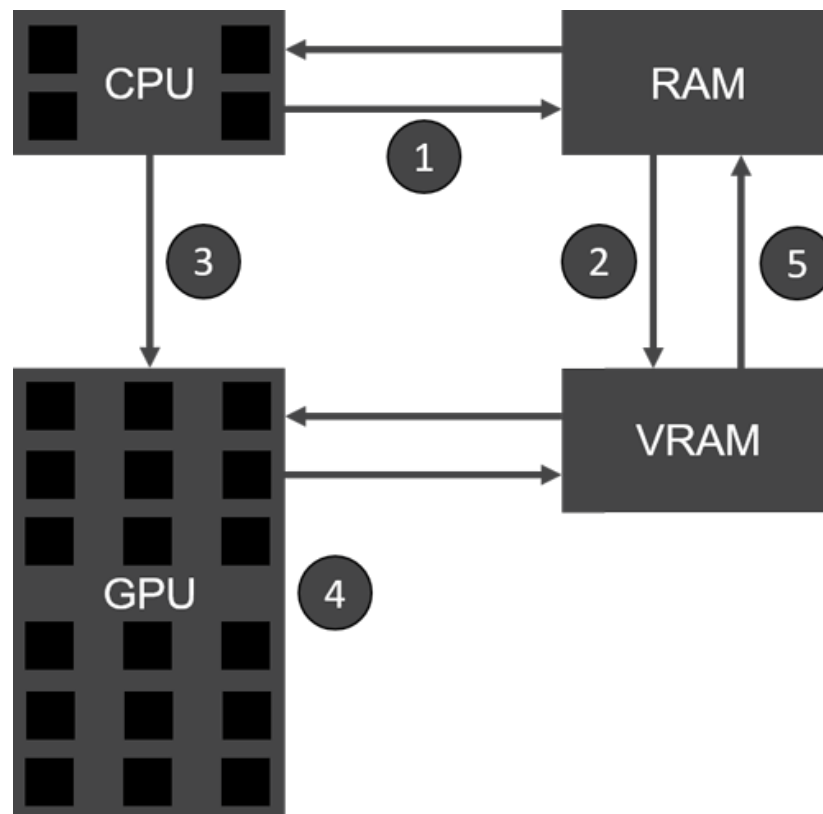


Рис.3.5 — Процес компіляції коду за допомогою ILGPU

### 3.4 Огляд програмного середовища та мови програмування C++.

Для розробки програмного забезпечення системи швидкої обробки сигналів ми обрали програмне середовище Visual Studio 2022, відоме своєю зручністю та інтуїтивно зрозумілим інтерфейсом, що забезпечує максимальну зручність та продуктивність. Ця платформа підтримує розробку програмного забезпечення на різних платформах, включаючи персональні комп'ютери, портативні пристрої та мобільні пристрої. Visual Studio полегшує розробку програмного забезпечення на одній з найпоширеніших мов програмування, а саме C++.

C++, об'єктно-орієнтована мова програмування, була представлена у 1979 році Бьярном Страуструпом [11]. Вона була розроблена як розширення мови програмування C з можливістю об'єктно-орієнтованого програмування. C++ є стандартом ISO і широко використовується при розробці різних систем, від вбудованих пристроїв до великих корпоративних додатків.

Разом з Visual Studio, C++ підтримує кросплатформенну розробку та надає розширені можливості для оптимізації продуктивності програмного забезпечення. Мова дозволяє прямий доступ до пам'яті, що робить її ефективною для розробки високопродуктивних додатків.

C++ підтримує широкий спектр типів, від примітивів до складних, включно з визначеними користувачем структурами та класами. Це дає розробникам високий рівень контролю над типами даних та їх обробкою.

Однією з важливих функцій безпеки C++ є заборона доступу до неініціалізованих змінних, що допомагає уникнути ризиків неконтрольованого приведення типів або виходу за межі пам'яті.

Працюючи в середовищі Visual Studio, розробник може створити новий проект або відкрити вже існуючий. Вибір опції "Створити новий проект" відкриває вікно налаштувань проекту, де можна вибрати тип програми, налаштувати параметри компіляції та інші параметри, необхідні для розробки C++ додатку, як показано на рисунку 3.6.

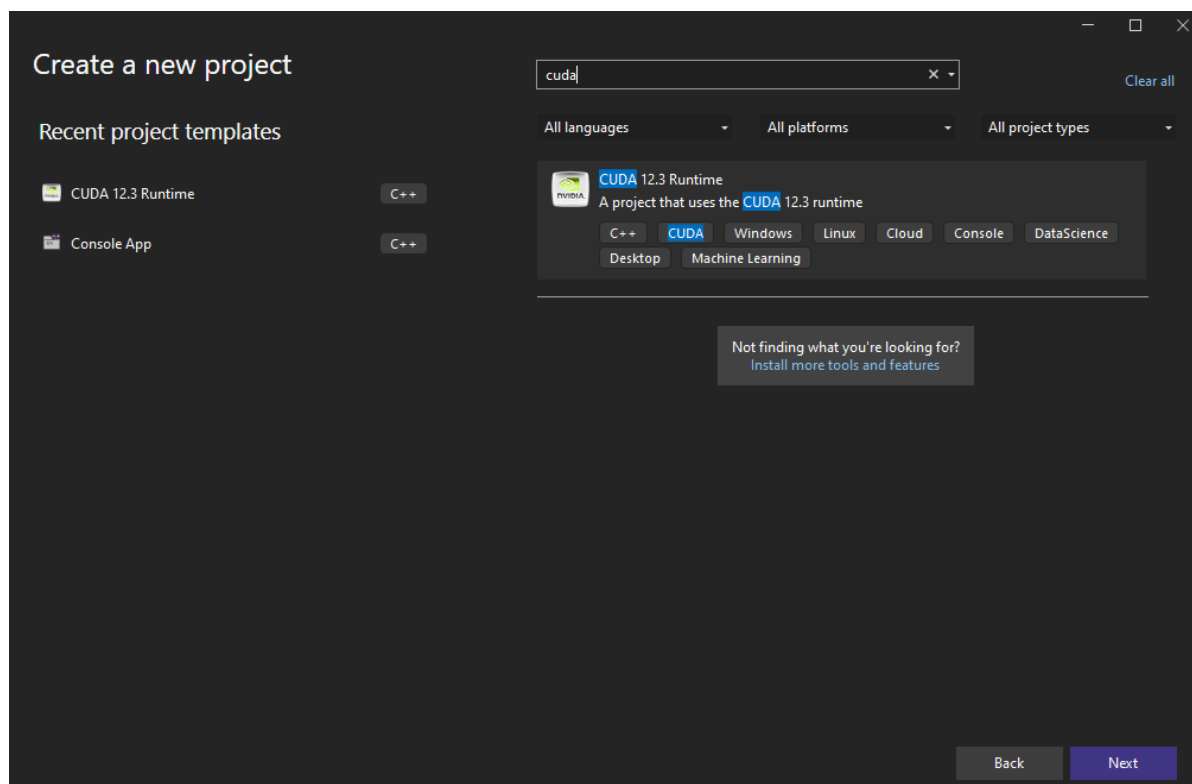


Рисунок 3.6 — Створення нового проекту

Нам буде доступний вибір між мовами програмування, типом проекту тощо. Але так як розробляється конвольне програмне забезпечення на мові C++, потрібно вибрати «Console App (CUDA 12.3 Runtime)». Після вибору та натиснення клавіші «Ok», відкривається середовище розробки із базовим програмним кодом, приклад зображено на рисунку 3.7.

```

1
2 #include "cuda_runtime.h"
3 #include "device_launch_parameters.h"
4
5 #include <stdio.h>
6
7 cudaError_t addWithCuda(int *c, const int *a, const int *b, unsigned int size);
8
9 __global__ void addKernel(int *c, const int *a, const int *b)
10 {
11     int i = threadIdx.x;
12     c[i] = a[i] + b[i];
13 }
14
15 int main()
16 {
17     const int arraySize = 5;
18     const int a[arraySize] = { 1, 2, 3, 4, 5 };
19     const int b[arraySize] = { 10, 20, 30, 40, 50 };
20     int c[arraySize] = { 0 };
21
22     // Add vectors in parallel.
23     cudaError_t cudaStatus = addWithCuda(c, a, b, arraySize);
24     if (cudaStatus != cudaSuccess) {
25         fprintf(stderr, "addWithCuda failed!");
26         return 1;
27     }
28
29     printf("{1,2,3,4,5} + {10,20,30,40,50} = {%d,%d,%d,%d,%d}\n",
30           c[0], c[1], c[2], c[3], c[4]);
31
32     // cudaDeviceReset must be called before exiting in order for profiling and
33     // tracing tools such as Nsight and Visual Profiler to show complete traces.
34     cudaStatus = cudaDeviceReset();
35     if (cudaStatus != cudaSuccess) {
36         fprintf(stderr, "cudaDeviceReset failed!");
37         return 1;
38     }
39
40     return 0;
41 }

```

Рисунок 3.7 — Приклад створення нового проекту

### 3.5 Розробка програмного забезпечення для обчислення низько-частотних сигналів з використанням технології CUDA.

На початкових етапах розробки програмного забезпечення процес починається зі створення комплексної блок-схеми, що ілюструє алгоритм роботи програми. Блок-схеми слугують графічним представленням даних або алгоритмів, покращуючи візуальне розуміння коду. Вони надають покрокове зображення рі-

шень проблем, алгоритмів або процесів, пропонуючи візуальну допомогу програмістам для розуміння та вдосконалення алгоритмів їхніх програм. Блок-схеми допомагають продемонструвати послідовний перебіг процесу, що сприяє виявленню та усуненню потенційних проблем в алгоритмі.

Блок-схема зазвичай складається з взаємопов'язаних блоків, які відображають логічний хід процесу. Таке візуальне представлення спрощує інтерпретацію та розуміння алгоритму для розробників. Візуальна природа блок-схем має перевагу в забезпеченні правильної побудови алгоритму програми і полегшує вирішення проблем на етапі розробки.

На рисунку 3.8 показано блок-схему програмного забезпечення, яке зараз розробляється, що є наочним посібником для послідовних кроків, пов'язаних з цим процесом. Це графічне представлення слугує цінним інструментом для розробників, що дозволяє впорядкувати процес розробки та підвищити загальну ясність і узгодженість алгоритму.

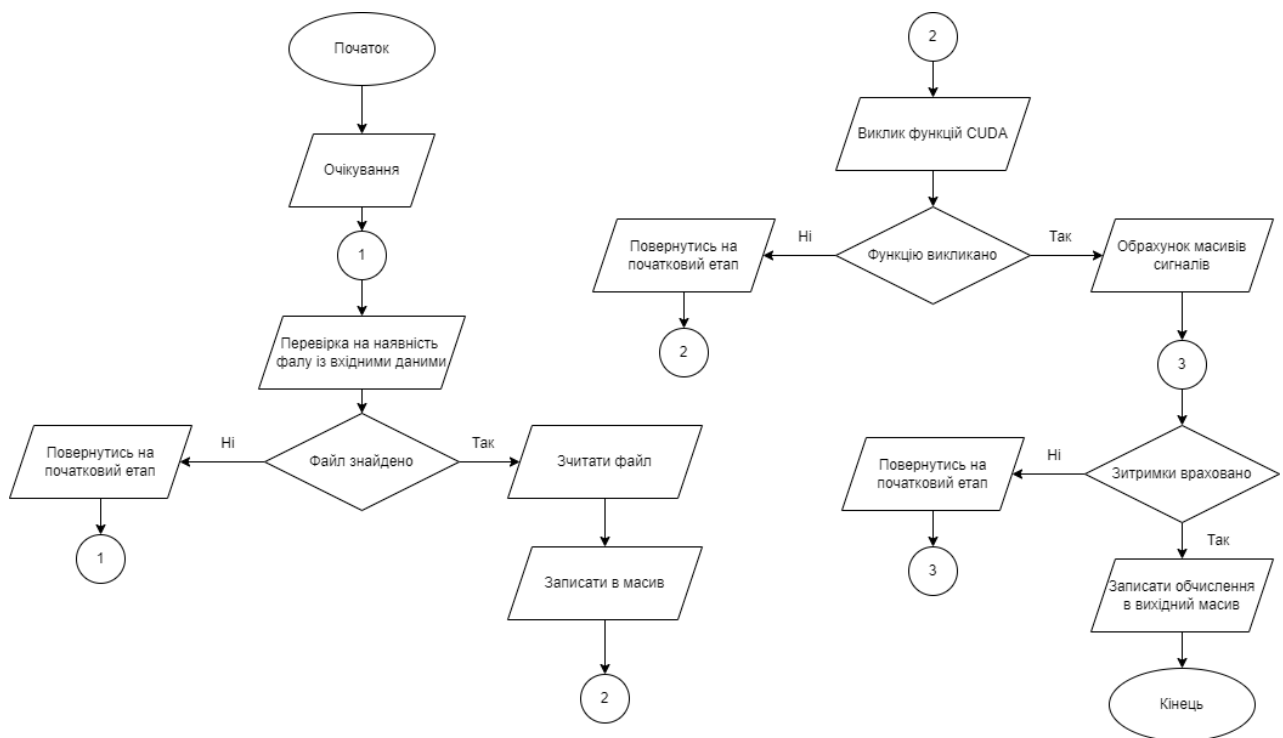


Рисунок 3.8 — Блок-схема роботи алгоритму



При поставленій задачі можна приступати до безпосередньої розробки програми. Основною проблемою при розробки програмного забезпечення в якому використовуються технології CUDA, являє собою відсутність потрібного мінімального GPU. Для цього першим кроком при написанні програми реалізовано перевірку наявних графічних прискорювачів (Лістинг 1).

Лістинг 1— Виведення інформації про GPU із підтримкою CUDA-ядер

```
int main() {  
  
    cudaError_t cudaStatus = cudaSetDevice(0);  
  
    if (cudaStatus != cudaSuccess) {  
  
        std::cerr << "cudaSetDevice failed! Error: " <<  
cudaGetErrorString(cudaStatus) << std::endl;  
  
        return 1;  
  
    }  
  
    cudaDeviceProp deviceProp;  
  
    cudaStatus = cudaGetDeviceProperties(&deviceProp, 0);  
  
    if (cudaStatus != cudaSuccess) {  
  
        std::cerr << "cudaGetDeviceProperties failed! Error: " <<  
cudaGetErrorString(cudaStatus) << std::endl;  
  
        return 1;  
  
    }  
}
```

Як результат роботи ми отримаємо виведення інформації в консоль про всі наявні прискорювачі, також кількістю доступних CUDA-ядер, розмір пам'яті графічного процесора, та версію драйвера, демонстрація роботи наведено на рисунку 3.10.

```

Microsoft Visual Studio Debug Console
Device Name: NVIDIA GeForce RTX 2060
Compute Capability: 7.5
Total Global Memory: 6442123264 bytes
Multiprocessors: 30
CUDA Cores per Multiprocessor: 64
Total CUDA Cores: 1920
Clock Rate: 1200 MHz
Memory Clock Rate: 5480 MHz
Memory Bus Width: 192 bits

C:\Users\dimas\source\repos\CudaSignals\x64\Debug\CudaSignals.exe (process 21648) e
xited with code 0.
Press any key to close this window . . .|

```

Рисунок 3.10 — Виведення інформації про наявний графічний процесор з підтримкою CUDA-ядер

Далі для перетворення звукового файлу в масив даних потрібно реалізувати метод `ReadToByte`. Даний метод було реалізовано за допомогою мови програмування C#, та використано як модуль. Цей метод призначений для відкриття .wav-файлу і подальшого зчитування його значень у масив байт. Отримані масиви зберігаються в окремі текстові файли для подальшої обробки, відповідний код представлено у Додатку Г.

Після цього розроблено метод, який відповідає за читання конвертованого аудіофайлу та подальший запис кожного каналу в одновимірний масив даних. Фрагмент його вмісту наведено у лістинг 2, а повний лістинг наведено у Додатку Д.

Для ефективного читання та запису даних з файлу до масиву необхідно використовувати простір імен `fstream` та `sstream`. Ці простори імен охоплюють пов'язані з вводом-виводом методи, структури та числення. Ці методи полегшують читання і запис даних у файли або потоки даних, а також включають функціональність для керування файлами і каталогами.

На додаток до цього, процес включає використання можливостей простору імен `sstream` та `fstream` для безперешкодної обробки файлових операцій, забезпечуючи плавне та ефективно читання даних із файлів у доступні структури із даними для подальшого використання (Лістинг 2).

Лістинг 2 — Ітерація по доступним файлам із даними

```
std::vector<std::vector<int>> inputData;
int maxRows = 0;
for (int i = 0; i < 4; ++i) {
    std::string filename = "C:/Users/dimas/Desktop/Data/chanel_" +
std::to_string(i + 1) + ".txt";
    std::ifstream inputFile(filename);
    if (!inputFile.is_open()) {
        std::cerr << "Unable to open file: " << filename << std::endl;
        return 1;
    }
    ...
}
```

Важливо зауважити, що метод `fileData` має важливу роль у забезпеченні ефективного читання даних з файлу. Під час свого виконання він не лише зчитує символи, але й може виконувати різноманітні операції обробки та фільтрації отриманих даних. Це дозволяє підготувати інформацію для подальшого використання в програмі.

Крім того, важливо передбачити в обробці помилок, які можуть виникнути при зчитуванні файлу. Наприклад, якщо файл не існує або відсутній доступ до читання, необхідно надати користувачеві інформативне повідомлення та вжити відповідних заходів для вирішення проблеми. Ключовим методом після читання даних з файлу є `fileData`. Цей метод забезпечує читання символів у потік. Метод метод циклічно зчитує наступний символ або набір символів з вхідного потоку, після чого записує дані до вектора (Лістинг 3).

Лістинг 3 — Оновлення максимальної кількості рядків та додавання даних до вектора

```
std::vector<std::vector<int>> inputData;
int maxRows = 0;
{
...
std::vector<int> fileData;
    int value;
    while (inputFile >> value) {
        fileData.push_back(value);
maxRows = std::max(maxRows, static_cast<int>(fileData.size()));
inputFile.close();
fileData.resize(maxRows, 0);
inputData.push_back(fileData);
    }
```

У процесі роботи з файлами дуже важливо розпізнавати і використовувати можливості простору імен `sstream` та `fstream`, що дозволяє безперешкодно виконувати операції читання і запису, а також ефективно керувати файлами і каталогами.

Далі, завантаження створених масивів у пам'ять графічного процесора. Цей процес передбачає виділення області пам'яті для конкретних типів значень на визначених прискорювачах. Передача даних на будь-який прискорювач та з нього здійснюється за допомогою методу `cudaMalloc`, що дозволяє виконувати як синхронне, так і асинхронне копіювання. `CudaMalloc` безпосередньо взаємодіє з ядром графічного процесора, забезпечуючи безперебійну взаємодію (Лістинг 4).

Такий підхід забезпечує ефективну обробку даних між CPU та GPU, оптимізуючи управління пам'яттю для підвищення продуктивності в задачах обробки графіки.

Лістинг 4 — Алокація, виділення пам'яті та копіювання даних до відеокарти

```
int* d_inputData, * d_matrix, * d_result;
cudaMalloc(&d_inputData, maxRows * 4 * sizeof(int));
cudaMalloc(&d_matrix, 31 * 4 * sizeof(int));
cudaMalloc(&d_result, maxRows * 31 * sizeof(int));
for (int i = 0; i < 4; ++i) {
    cudaMemcpy(d_inputData + i * maxRows, inputData[i].data(), maxRows *
sizeof(int), cudaMemcpyHostToDevice);
}
cudaMemcpy(d_matrix, matrix.data(), 31 * 4 * sizeof(int),
cudaMemcpyHostToDevice);
```

У цій секції відбувається первинний виклик об'єкта `cudaMemcpyHostToDevice`. Усі класи та функції компілятора залежать від екземпляра. `CudaMemcpyHostToDevice`, який слугує інтерфейсом для компілятора. Важливо зазначити, що об'єкт `cudaMemcpyHostToDevice`, як і більшість екземплярів класів, вимагає виклику методу `dispose` для запобігання витoku пам'яті. Прискорювач (Accelerator) у цьому коді означає апаратний або програмний графічний процесор. Зокрема, представлено прискорювач типу CUDA, що вказує на використання системою CUDA-сумісного графічного процесора. Однак, важливо підкреслити недолік - вимога до графічного процесора з серії Nvidia GTX 980 або вище, а також встановлення програмного забезпечення CUDA, не нижче 10-ї версії.

Наступним етапом розробки програмного забезпечення є визначення функцій ядра для використання в методі `calculateResult`. Функції ядра є основними функціями, які виконують паралельні обчислення на відеокарті. У відображеному лістингу коду, функція `calculateResult` викликається в паралельному режимі для кожного рядка вхідних даних. (Лістинг 5).

## Лістинг 5 —Оголошення ядра

```

__global__ void calculateResult(const int* inputData, const int* matrix, int*
result, int rows, int cols, int channels) {
    int tid = blockIdx.x * blockDim.x + threadIdx.x;
    while (tid < rows) {
        for (int i = 0; i < cols; ++i) {
            int sum = 0;
            for (int j = 0; j < channels; ++j) {
                sum += matrix[i * channels + j] * inputData[tid * channels + j];
            }
            result[tid * cols + i] = sum / channels;
        }
        tid += blockDim.x * gridDim.x;
    }
}

```

Осново́вний алгоритм обчислення поставленої задачі задається за формулою (3.1)

$$\begin{aligned}
 & result[tid \times cols + i] \\
 = & \frac{\sum_{j=0}^{channels-1} matrix[i \times channels + j] \times inputData[tid \times channels + j]}{channels}
 \end{aligned}
 \tag{3.1}$$

де:  $tid$  — ідентифікатор потоку, який визначається як  $blockIdx.x \times blockDim.x + threadIdx.x$ ;

$cols$  — кількість стовпців у вихідній матриці результату;

$i$  — індекс стовпця у вихідній матриці (від 0 до  $cols - 1$ );

$channels$  — кількість каналів у вхідних даних та матриці заримок;

$matrix$  — елемент матриці заримок у  $i$  — му стовпці та  $j$  — му каналі;

$inputData$  — елемент вхідних даних у  $j$  — му каналі для  $tid$  — го потоку;

Наступним кроком після визначення конфігурації блоків та ниток для виклику CUDA kernel та виклику самого ядра є копіювання результатів з відеокарти (девайса) назад на хост (центральний процесор). Цей етап важливий для того, щоб отримати результати обчислень із відеокарти та продовжити їхню обробку на центральному процесорі або для подальших операцій.

Для цього використовуються функції копіювання пам'яті `cudaMemcpy`, які забезпечують обмін даними між хостом і девайсом (Лістинг 6).

Лістинг 6 — Конфігурація графічних блоків з викликом CUDA kernel та копіюванням результатів до хосту

```
int threadsPerBlock = 256;
int blocksPerGrid = (maxRows + threadsPerBlock - 1) / threadsPerBlock;
calculateResult << <blocksPerGrid, threadsPerBlock >> > (d_inputData, d_ma-
trix, d_result, maxRows, 31, 4);
std::vector<int> result(maxRows * 31);
cudaMemcpy(result.data(), d_result, maxRows * 31 * sizeof(int), cudaMem-
cpyDeviceToHost);
```

Після виконання CUDA-коду та отримання результатів на хості важливо вчасно вивільнити використану пам'ять на відеокарті для ефективного управління ресурсами. Для цього використовується функція `cudaFree`, щоб звільнити пам'ять, яку було виділено на відеокарті за допомогою `cudaMalloc`. Ця операція є важливим етапом управління ресурсами, особливо в разі великих обчислювальних завдань чи в багатоетапних програмах.

В результаті виконання цього модулю отримуємо кількість масивів, рівну кількості затримок, а кількість елементів у кожному масиві відповідає кількості елементів у вхідних даних, що є продемонстровано на рис. 3.16.





## 4 ЕКСПЕРЕМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА ТЕСТУВАННЯ

### 4.1 Деталізація акустичної поверхні за умов неповних даних шляхом оптимізації нечіткої бази знань II типу

У сфері моніторингу місцевості мікрофонні решітки стали поширеною технологією для візуалізації акустичного поля. Розв'язання оберненої задачі дозволяє реконструювати акустичне поле. За наявності відомої кількості розріджених джерел і достатньої кількості вимірювань застосовують класичні методи регуляризації. Однак побудова мікрофонних решіток з обмеженою кількістю мікрофонів вносить невизначеність у дані через нерівномірний розподіл фокальних точок як у ближньому, так і в дальньому полі.[18] Отже, за наявності невизначеності даних необхідні додаткові методи для уточнення акустичних зображень.

У ближньому полі методи деконволюції використовуються для підвищення роздільної здатності точкових джерел, розташованих поблизу центру масиву. Однак, ці методи можуть бути обчислювально дорогими, особливо при роботі зі складними розподілами джерел. Навпаки, в дальньому полі дослідники проводять експерименти, щоб визначити оптимальне положення мікрофонної решітки. Згодом для оцінки параметрів джерела застосовують статистичні методи. Фокусування акустичного зображення, засноване на байєсівському висновку, також використовується для зменшення кількості вимірювань, використовуючи попередню інформацію про розподіл джерел.

Внутрішня невизначеність даних акустичних масивів ефективно описується моделями, що включають нечіткі або інтервальні параметри.[19] Ця невизначеність стимулювала розвиток систем з гібридними невизначеними параметрами у вигляді нечітких інтервалів. Проектування таких систем передбачає мінімізацію ентропії відгуку гучномовця за допомогою моделі з екстремальними значеннями параметрів формування променя. Через різноманітні види невизначеностей, що проявляються на різних рівнях, параметри джерела визначаються з високим ступенем нечіткості. Для прогнозування невизначеності поширення по-

току енергії використовуються інтервальні оцінки, які представляють відгук акустичної системи у вигляді нижніх і верхніх акустичних рівнів для кожного вузла решітки. Матриця зв'язку або матриця змішування використовується для моделювання внеску джерел або груп джерел у загальний рівень звукової енергії поля. Реконструкція поля передбачає генерування та вибір варіантів, що найбільш точно відповідають розподілу енергії, який спостерігається під час вимірювань. Для вибору параметрів джерела з набору реконструкцій застосовуються генетичні алгоритми.

Збільшення кількості мікрофонів покращує якість зображення, але також збільшує вартість системи і час обробки. Однак при обробці потоків акустичних даних розширення часового вікна масиву вважається неприйнятним. Отже, необхідний компроміс між вартістю обладнання, точністю та складністю реконструкції. У цьому дослідженні акустична поверхня моделюється з використанням нечіткої бази знань типу II.[20] Розподіл звукової енергії в полі визначається інтервальною матрицею нечітких відношень. Отже, множина розв'язків оберненої задачі відповідає множині розв'язків системи рівнянь нечіткої логіки (СНЛ) для матриці відношень. При роботі з неповними даними множина розв'язків СФЛЗ інтерпретується як множина реконструкцій потенціального поля у вигляді нижньої та верхньої акустичних поверхонь. Таким чином, задача уточнення акустичних зображень спрощується за рахунок оптимізації нечіткої бази знань, що включає пошук розташування джерел, акустичних рівнів і нечіткої матриці розподілу звукової енергії, що узгоджується з бажаним або екстремальним рівнем точності і часової складності реконструкції поля. Цей підхід спрощує процес реконструкції для економічно ефективних акустичних систем, усуваючи необхідність проведення серії дорогих експериментів для позиціонування мікрофонної сітки навколо рухомих джерел звуку.

Шляхом акустичного бачення контролюється відкрита ділянка місцевості  $250 \times 250$  м. Акустичне зображення  $u(x_1, x_2)$  на вході мікрофонної решітки (рис. 4.1) згенероване  $n=270$  джерелами, діапазон сили звуку яких складає  $I_p \in [10^{-8}, 10^{-1}]$  Вт/м<sup>2</sup>,  $p = \overline{1, n}$ , що відповідає акустичним рівням 40–110 дБ.

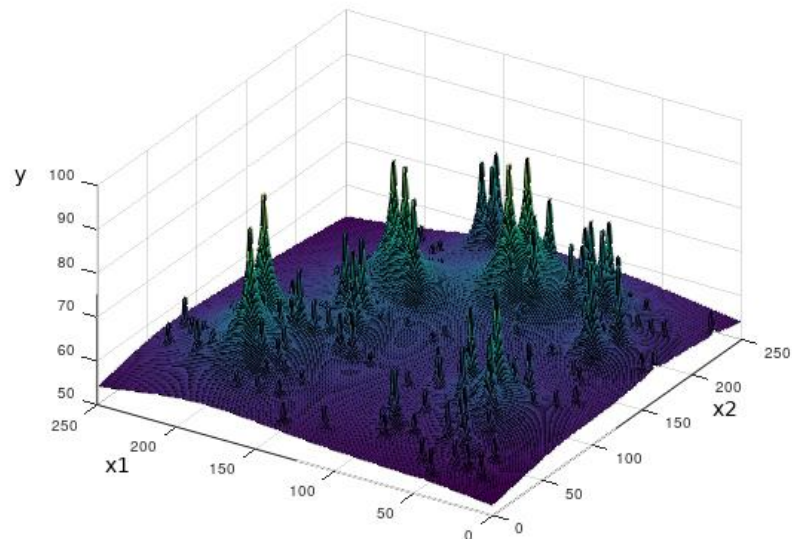


Рисунок 4.1 – Акустичне зображення на вході мікрофонної решітки

Акустичне зображення на виході мікрофонної решітки (рис. 4.2) отримано у вигляді пар результатів вимірювань «координати  $(\hat{x}_1^s, \hat{x}_2^s)$  – рівень акустичної енергії  $(\hat{y}_s)$ »,  $s = \overline{1, Q}$ . Мікрофонна решітка сформована матрицею 32\*32 мікрофонів із відстанню 25 м і кроком сканування  $1^0$ . [19] Роздільна здатність зображення становить  $Q = 3615$  точок, зменшуючись в міру збільшення відстані від центру решітки.

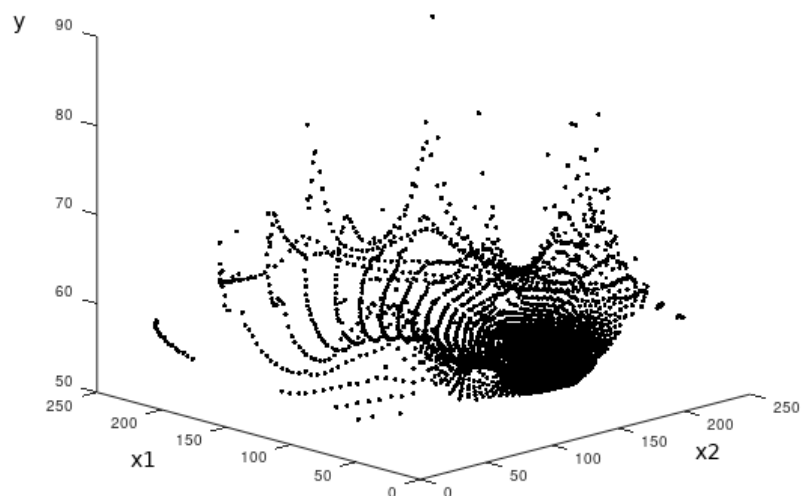


Рисунок 4.2 – Акустичне зображення на виході мікрофонної решітки

Нечітка модель працює на принципах нечітких множин і правил, де кожен сегмент у просторі аналізу представлений у вигляді нечіткої множини з елементами від 0 до 1, що позначають рівні активності. Включення функцій належності дозволяє точно оцінити, наскільки добре даний сигнал або шум узгоджується з кожною областю. Нечітка база знань моделюється за допомогою матриці нечітких відношень

$$R \subseteq A_K \times E_J = [r_{KJ}, K = \overline{1, Z}, J = \overline{1, M}]. \quad (4.1)$$

Матриці нечітких відношень слугують фундаментальним інструментом в теорії нечіткої логіки, який використовується для моделювання нечітких взаємозв'язків між елементами. Ці матриці виявляються цінними для представлення ступенів приналежності елементів до нечітких множин або для вимірювання ступеня подібності між ними. Зазвичай матриця нечітких відношень має розмірність, яка відповідає кількості елементів, що розглядаються. Для кожної пари елементів у матриці точно визначається ступінь їхньої нечіткої взаємодії або зв'язку. Це дозволяє комплексно представити складні відносини і ступені зв'язку між різними елементами в нечіткій системі.

Центральним елементом нечіткої логіки є встановлення функцій належності для кожної нечіткої множини, що відповідає сегменту акустичного простору. Ці функції кількісно визначають ступінь відповідності акустичного сигналу або шуму певній просторовій області. Найпоширеніші типи функцій належності включають трикутні та трапецієподібні. Трикутні функції демонструють лінійне зростання приналежності від нуля до одиниці, тоді як трапецієподібні функції можуть мати лінійне або нелінійне зростання або спадання. Гаусові функції, що нагадують гаусові розподіли, використовуються для моделювання невизначеності або нормальних розподілів. Кембрійські функції, що є різновидом гауссових, можуть краще відповідати реальним даним. Сигмоїдальні функції, що нагадують S-подібні криві, моделюють екстремальні значення або різкі переходи між нечіткими областями. Профільні функції враховують складні форми належності, такі

як піки або локальні максимуми. Лінійні функції використовують лінійні залежності для визначення ступеня приналежності. Множинні функції можуть представляти акустичний сигнал або шум відносно окремих просторових областей або аспектів активності. Деякі функції належності можуть бути адаптивними, змінюючись залежно від зовнішніх факторів або динаміки сигналу з часом.

Всесвіт охоплює повний спектр потенційних значень акустичних параметрів. Поділ його на нечіткі множини окреслює області для подальшого аналізу. Наприклад, у частотному аналізі всесвіт можна розділити на низько-, середньо- та високочастотні області. В контексті акустичної моделі всесвіт може охоплювати такі параметри, як частота, амплітуда та час. Розбиття передбачає поділ діапазону на нечіткі множини, визначаючи різні ступені приналежності для конкретних значень. Такий детальний підхід враховує нюанси у визначенні активності у всьому всесвіті. Кожна нечітка множина має функцію приналежності, яка визначає ступінь зв'язку кожного значення в універсумі. Експертний аналіз або алгоритми кластеризації визначають кількість і розміщення нечітких множин, які можуть бути рівномірно розподілені або адаптовані до характеристик акустичного середовища. Нечіткі множини, що перетинаються, вказують на одночасну приналежність значення до кількох областей, що впливає на визначення активності. Динамічне оновлення розбиття простору може мати вирішальне значення, особливо при зміні умов навколишнього середовища. Адаптивні алгоритми можуть автономно або напівавтономно коригувати розбиття для ефективного відображення змін даних.

Нечіткі моделі використовують логічні правила для обчислення результатів, використовуючи функції належності. Ці правила виражаються в операторах "ЯКЩО-ТОДІ", де "ТОДІ" [19] визначає вихід для кожного правила на основі ступеня приналежності. Наприклад, якщо "Зона 1" має високий ступінь належності "низька активність", а "Зона 2" має високий ступінь належності "висока активність", результатом може бути "позначити Зону 1 як менш активну, ніж Зону 2".

Вхідні змінні, такі як інтенсивність звуку, частотний діапазон і часові характеристики, є параметрами або характеристиками. Кожна зона може мати свій набір вхідних змінних, кожній з яких присвоюється функція належності, що вказує на її відповідність певній зоні або рівню активності. Функції належності, трикутної, трапецієподібної або іншої форми, забезпечують цю вказівку. Логічні оператори, такі як "І", "АБО" та "НЕ", визначають комбінації умов для кожного правила. Вихідні змінні визначають ступінь активності для кожної зони або виходять за межі зон, що є результатом обчислень, застосованих до функцій належності та логічних правил. Кожне логічне правило оцінює, наскільки добре вхідні параметри відповідають кожній зоні, а рівні активності обчислюються шляхом агрегування відповідних функцій належності.

Агрегування вхідних сигналів (акустичних даних) для кожної зони передбачає використання функцій належності, що потенційно може включати такі операції, як максимізація, мінімізація або усереднення для визначення ступеня активності в кожній зоні.

На початковому етапі агрегації для кожної зони виявлення збираються нечіткі дані, що включають вимірювання акустичних параметрів, таких як інтенсивність і частота звуку. Потім для кожного параметра визначаються відповідні функції належності, що відображають ступінь належності значень до певних нечітких множин. Ці функції належності можуть бути визначені експертами або адаптуватися автоматично на основі даних. Логічні правила, що враховують ці функції належності, керують агрегацією для кожної області, використовуючи такі операції, як максимізація, мінімізація або усереднення ступенів належності. Після агрегування визначаються рівні активності для кожної області, представлені числовими значеннями або діапазоном, що вказує на ступінь інтенсивності акустичної активності.[20]

Вирішення складних завдань моделювання та ідентифікації зон активності в акустичній аналітиці вимагає точності та ефективності. Ця технічна записка висвітлює процес застосування нечітких підходів для цієї мети.

Під зоною активності в цьому контексті мається на увазі просторова область, в якій відбувається помітна акустична подія. Кожна така зона характеризується нечіткою множиною, функції належності якої відіграють ключову роль у визначенні ступеня відповідності конкретної точки характеристикам зони.

Зона активності окреслюється як просторова область, де відбувається значна акустична подія. Представлення кожної зони включає в себе нечітку множину, де функції належності визначають ступінь, до якого певна точка асоціюється з цією конкретною зоною. Формально, нехай  $A_i$  -  $i$ -та зона, а  $\mu A_i(x)$  - функція приналежності точки  $x$  до цієї зони.

Функції належності ( $x$ ) визначаються на основі акустичних параметрів, таких як інтенсивність звуку, частотний спектр і часові характеристики. Для кожної зони застосовуються відповідні методи оцінки цих параметрів, причому функції належності можуть бути адаптивними з часом. Використання динамічних нечітких систем полегшує відстеження змін в акустичному середовищі в режимі реального часу, що дозволяє змінювати межі зон активності. Операції нечіткої логіки, в тому числі об'єднання нечітких множин, використовуються для виявлення ділянок, де зони активності можуть перекриватися, запобігаючи прогалинам в інформації і враховуючи потенційну можливість виникнення декількох подій в даній області.

Процес виведення в нечіткій моделі, заснований на нечітких правилах, визначає нечітке значення для конкретного результату шляхом розгляду набору правил, що визначають взаємозв'язок між вхідними і вихідними змінними. Цей процес відбувається наступним чином:

— початкові значення вхідних змінних, що представляють нечіткі поняття, перетворюються в нечіткі множини за допомогою функцій приналежності, кожній множині присвоюється ступінь приналежності в діапазоні від 0 до 1, що відображає, наскільки добре значення відповідає концепції;

— визначаються нечіткі правила, що виражають зв'язок між вхідними і вихідними нечіткими змінними, кожне з яких формулюється у вигляді "ЯКЩО ... ТОДІ", із зазначенням умов і висновків;

- значення правил агрегуються для кожної вихідної нечіткої змінної за допомогою оператора "Максимум" (у випадку оператора "АБО"), це визначає ступінь належності для кожного потенційного варіанту вихідного значення;
- метод дефазифікації застосовується для перетворення сукупних значень вихідних змінних у конкретні числові значення, можуть бути використані різні методи, такі як центр ваги або максимальна важливість;
- отримані числові значення слугують виходом нечіткої моделі, вказуючи на рекомендовані дії, рівні впливу або інші числові значення, засновані на базових концепціях і логіці правил.

Зони акустичної активності позначають просторові області, де звукові сигнали виявляють активність. Ці зони можна охарактеризувати як ділянки, де виявлено звукову активність, що охоплюють простори з чіткими звуковими сигналами, галасливе середовище або місця з виразними акустичними характеристиками. Процес виявлення зон акустичної активності починається зі збору даних з акустичних датчиків, таких як мікрофони або інші звукозаписуючі пристрої, що реєструють звукові хвилі у визначеному просторі. Аналого-цифрове перетворення отриманих акустичних сигналів дає числові дані, які обробляються за допомогою обчислювальних алгоритмів.

Функції приналежності, на які впливають числові дані, вводяться для оцінки ступеня відповідності кожної просторової точки певній зоні акустичної активності, що відображає такі параметри, як інтенсивність звуку та частотний спектр. Потім застосовуються алгоритми кластеризації та класифікації для групування просторових ділянок у зони на основі спільних характеристик акустичної активності. Для кожної зони акустичної активності обчислюються сукупні характеристики, такі як середні значення і ступінь приналежності, а результати для наочності представляються у вигляді карт або графіків. Систематичний моніторинг цих зон дозволяє адаптуватися до змін у звуковому середовищі за допомогою механізмів автоматичного оновлення на основі нових акустичних даних.



Для визначення зон акустичної активності застосовуються різні підходи та методи, зокрема:

— спектральний аналіз використовується для визначення частотних характеристик аудіосигналів у реальному часі, спектральний аналіз допомагає виявити зони з підвищеною активністю шляхом виявлення первинних і вторинних частотних піків.

— часовий аналіз дозволяє визначити, коли і як відбувається активність у просторі, методи визначення зон передбачають виявлення інтервалів, коли звукова активність перевищує попередньо визначений поріг.

— кореляційний аналіз досліджує взаємозв'язок між різними просторовими точками на основі схожості акустичних сигналів, зони активності можуть бути окреслені як області, що демонструють високу кореляцію між звуковими сигналами.

Для реконструкції акустичної поверхні кількість нечітких локацій і класів потужності звукових джерел обмежувалась так:  $\bar{k}_1 \leq 25$ ,  $\bar{k}_2 \leq 25$ ,  $\bar{M} \leq 7$ . Задача полягала у знаходженні варіанта реконструкції поля, який забезпечує: мінімальну кількість зон акустичної активності для заданої похибки реконструкції  $Z \rightarrow \min$  і  $\bar{\varepsilon} \leq 7.5$  у прямій постановці; мінімальну похибку реконструкції для заданої кількості зон акустичної активності  $\varepsilon \rightarrow \min$  і  $\bar{Z} \leq 100$  у двоїстій постановці. Програмна модель реалізована в системі MATLAB (додаток Е).

У результаті розв'язання задач оптимізації (4.3) і (4.4) встановлено кількість і границі класів потужності звукових джерел:

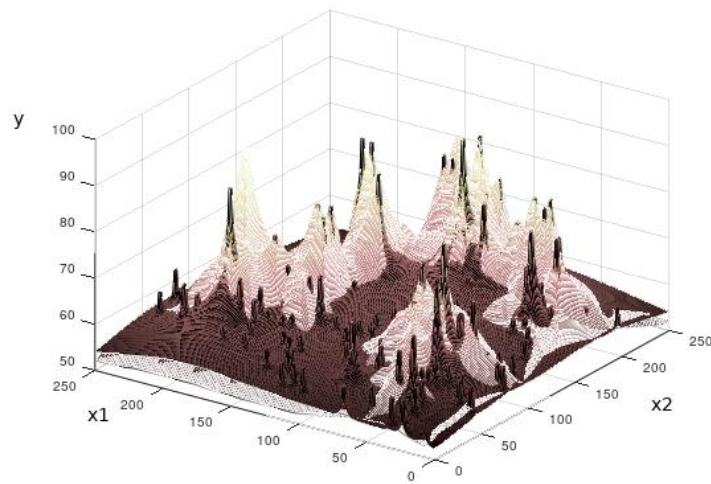
$M=6$  ( $M=7$ ) для прямої (двоїстої) задачі;

$D_1=[46, 55]$  dB,  $D_2=[55, 65]$  dB,  $D_3=[65, 74]$  dB,  $D_4=[74, 84]$  dB,  $D_5=[84, 92]$  dB,  $D_6=[92, 100]$  dB для прямої задачі;

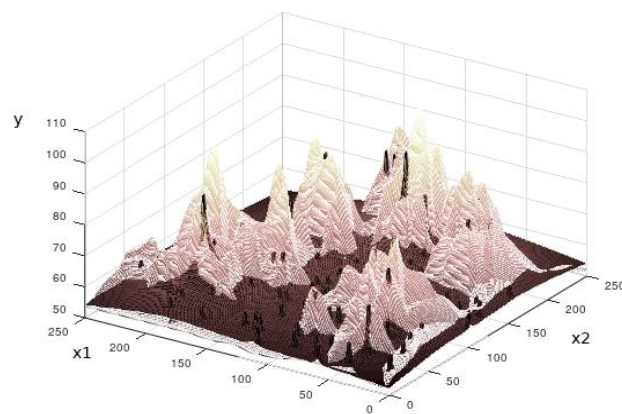
$E_1=D_1=[46, 54]$  dB,  $E_2=D_2=[54, 64]$  dB,  $E_3=[64, 70]$  dB;  $E_4=[70, 78]$  dB,  $E_5=[78, 85]$  dB,  $E_6=D_5=[85, 92]$  dB,  $E_7=D_6=[92, 100]$  dB для двоїстої задачі;

Тобто, при розв'язанні прямої і двоїстої задачі два класи малої ( $D_1, D_2$  і  $E_1, E_2$ ) і великої ( $D_5, D_6$  і  $E_6, E_7$ ) потужності залишились без змін, а два класи середньої потужності ( $D_3, D_4$ ) було розділено на три класи ( $E_3, E_4, E_5$ ) [2, 19].

Результати деталізації акустичної поверхні для розв'язку прямої і двоїстої задачі (рис. 4.3) демонструють значне покращення точності реконструкції порівняно з для ближнього і дальнього поля. Порівняння ліній рівня для реального зображення, акустичної поверхні для розв'язку прямої і двоїстої задачі (рис. 4.4) свідчить про досягнуту точність реконструкції у дальньому полі.

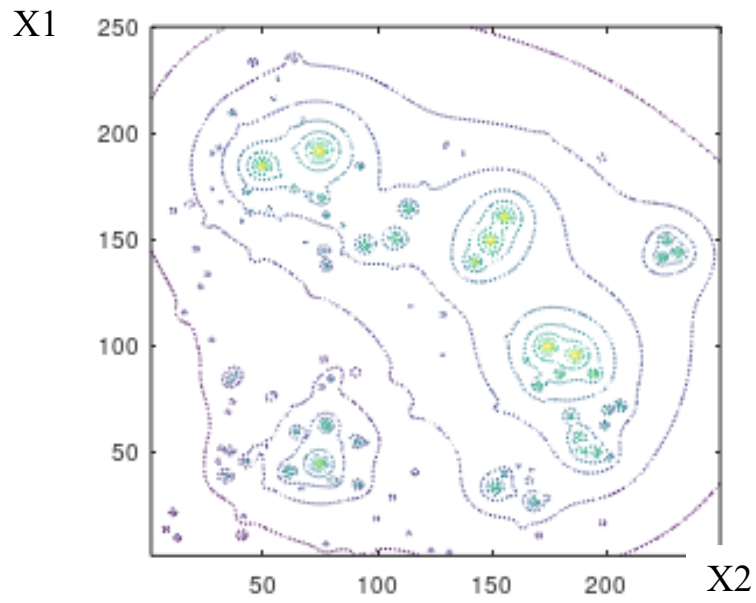


а)

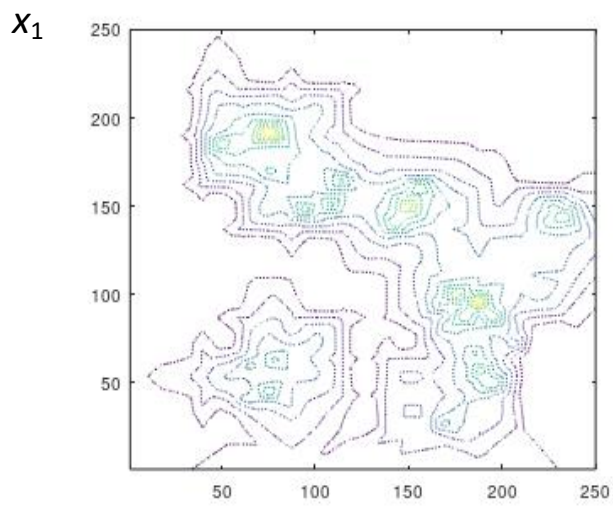


б)

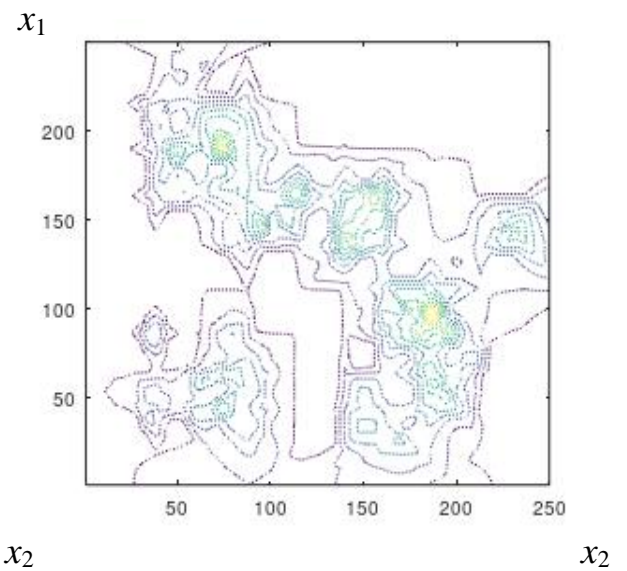
Рисунок 4.3– Результати деталізації акустичної поверхні для розв'язку прямої (а) і двоїстої (б) задачі



а)



б)



в)

Рисунок 4.4 – Лінії рівня для реального зображення (а), акустичної поверхні для розв'язку прямої (б) і двоїстої (в) задачі

4.2 Чисельні результати експериментальних досліджень, натурних випробувань, приклади функціонування системи, пристрою, програми в різних режимах.

В експерименті використовувалося обладнання з класичними методами формування пучка, що виключає можливість тонкої деталізації акустичної поверхні. Розроблений метод порівнювався з методами реконструкції акустичного поля, які оцінюють внесок зон акустичної активності в загальну енергію поля за результатами вимірювань, використовуючи генетичний відбір реляційних моделей даних. У цих дослідженнях набір варіантів реконструкції поля характеризувався інтервальною матрицею зв'язків (змішуванням), а досягнення більш тонкої деталізації вимагало багаторазового виконання генетичного алгоритму. Розмірність матриці відношень зростала пропорційно кількості зон акустичної активності.

Відмінною особливістю цього методу є можливість одночасного пошуку нижньої та верхньої меж нечітких відношень для кожного класу потужності джерел звуку, що призводить до скорочення часу реконструкції. Реалізація моделей з детальними матрицями відношень передбачає розв'язання послідовності  $V$  задач з параметрами  $3M$ , де  $V$  - кількість варіантів реконструкції поля. Реконструкція акустичної поверхні у вигляді розв'язку зв'язаних диференціальних рівнянь у частинних похідних (PDE) вимагає розв'язання послідовності  $VM$  задач з  $Z$  параметрами для нижньої та верхньої границь нечітких відношень. Крім того, генерація нульового варіанту реконструкції вимагає розв'язання оптимізаційної задачі з  $2(k_1+k_2)M$  змінними для нижньої та верхньої функцій приналежності нечітких локалізацій джерел звуку. Розв'язок задачі послідовного нелінійного програмування (SNLP) було реалізовано за допомогою MATLAB (Додаток Е).

Зменшення обчислювальної складності призвело до наступних оцінок часу реконструкції для максимальної кількості зон акустичної активності ( $Z = 100$ ). Використовуючи принципи паралельних обчислень, час генерації нижньої та

верхньої акустичних поверхонь для прямої задачі не перевищує 15 секунд. Деталізація акустичної поверхні для двоїстої задачі потребує 3 секунди. При збільшенні кількості зон акустичної активності зі складним розподілом джерел, що мають як великі, так і малі перепади потужностей, деталізація акустичної поверхні традиційними методами призводить до затримки до 0,5 хвилини. На відміну від цього, часові оцінки, отримані за допомогою запропонованого методу, не перевищують час оновлення акустичної сцени для рухомих джерел звуку. Отже, запропонований метод майже вдвічі збільшує частоту реконструкції, досягаючи двократного прискорення деталізації акустичної поверхні для реальних сценаріїв в акустичному полі. При цьому вартість реконструкції залишається прийнятною завдяки використанню мікрофонних решіток з класичним формуванням діаграми спрямованості.

#### 4.3 Результати впровадження і експлуатації, рекомендації щодо вдосконалення і коригування, перспективи досліджень і озробкир.

У вимірювальних каналах АЦП системи суттєво підвищено точність підсилення та АЦП низькочастотних сигналів за рахунок самокалібрування статичних і динамічних характеристик та цифрової корекції оброблюваних сигналів. Натурні випробування експериментальних зразків розроблених АЦП систем моніторингу та сканування сигналів сприяли визначенню досягнутих технічних параметрів. Це, в свою чергу, дозволило внести необхідні корективи в методи обробки сигналів, апаратне та програмне забезпечення. Результати цих випробувань мали вирішальне значення для успішного виконання госпдоговірних робіт за темою дослідження.

Реалізація АЦП з ваговою надлишковістю можлива на основі конденсаторно-цифрових перетворювачів з недвійковим співвідношенням ваг розрядів. Такий підхід забезпечує безперервність характеристики кодування навіть при зміні температури та інших параметрів навколишнього середовища, дозволяючи проводити калібрування виключно в цифровому вигляді без необхідності вико-

ристання додаткових калібрувальних ЦАП. Впровадження вагової надлишковості обмежує кількість кодових комбінацій, що дозволяє фіксувати відхилення ваги бітів шляхом контролю їх набору. Надано наукове обґрунтування, яке демонструє, що використання вагової надлишковості значно розширює допустимий діапазон відхилень вагових коефіцієнтів АЦП. Крім того, це дозволяє фіксувати відхилення шляхом аналізу вихідного коду.

Підвищення точності локалізації джерела звуку досягається шляхом генерації розв'язків оберненої задачі у вигляді нечітких правил І-НІ. Методологія передбачає побудову та налаштування баз знань, зокрема "джерело сигналу, що відновлюється, - енергетичний розподіл спостережуваного сигналу", з використанням генетично-нейронних технологій. Генетичний алгоритм сприяє швидкому, але наближеному відновленню подій в акустичному полі, тоді як нейронечітка технологія адаптує рішення і уточнює координати в міру надходження нових експериментальних даних. Такий динамічний підхід дозволяє відновлювати події в акустичному полі в реальному часі без шкоди для точності. Науково обґрунтовано та доведено існування повного набору розв'язків оберненої задачі нечіткого логічного виводу для покращення роздільної здатності локалізації сигналів.

Підвищення вірогідності правильного розпізнавання акустичних сцен і подій досягається за рахунок використання нових і вдосконалених методів обробки аудіосигналів. Розроблено новий метод адаптивного формування словника ознак, що містить інформативні ознаки (параметри) звукових подій, зафіксованих в атмосфері звукової сцени. Це підвищує ймовірність прийняття правильних рішень про характер звуку в умовах погіршення співвідношення сигнал/шум. Удосконалено існуючий метод кластеризації на основі EM-алгоритму з використанням удосконаленого методу k-середніх для формування початкової конфігурації центроїдів. Результати експериментальної перевірки підтверджують ефективність використання просторово-часового представлення параметрів та удосконаленого методу кластеризації k-середніх для створення словника ознак аудіосигналів.

Очікувані науково-практичні результати дослідження, що проявляються в експериментальних зразках АД систем моніторингу та синхронної обробки низькочастотних сигналів, можуть бути ефективно застосовані в додатках, пов'язаних з безпекою, контролем доступу та моніторингом на територіях поблизу промислових та військових об'єктів підвищеної важливості. Об'єктами моніторингу можуть бути державні кордони, лінії адміністративних кордонів, території поблизу аеродромів, атомних електростанцій, небезпечних виробництв, центрів енерго- та водопостачання тощо. Досліджені методи та засоби багатоканального моніторингу мають перспективні застосування в найрізноманітніших галузях, включаючи телерадіомовлення, промисловість, екологію, медицину тощо. Ці сфери застосування мають першочергове значення для України та відкривають багатообіцяючі перспективи для виходу на зовнішній ринок.

## 5 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМНОГО МОДУЛЯ ДЛЯ ОБРАХУНКУ АКУСТИЧНИХ ПОВЕРХОНЬ В БАГАТОКАНАЛЬНІЙ АНАЛОГО-ЦИФРОВІЙ СИСТЕМІ

### 5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку нового програмного модулю, призначеного для потокового обчислення акустичних поверхонь в аналого-цифровій системі. Особливістю модулю є те, що дана методика використовує технології багатоканальних обчислень за допомогою технолої CUDA-ядер, що дозволяє нам виконувати обчислення в режимі реального часу. Крім того програмний модуль використовує нову методику обрахунків, тому він не має відчизняних аналогів, а відомі закордонні, не є загальнодоступними для звичайного використання.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 5.1.

Таблиця 5.1 — Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах



Продовження табл. 5.1

Ринкові переваги					
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
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
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

Продовження табл. 5.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 — Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	3	4
Наявність аналогів на ринку	3	3	4
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4

Продовження табл. 5.2

Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	4
Супровідна документація	3	3	4
Сума	42	41	44
Середньоарифметична сума балів	$(42+41+44) / 3 = 42,33$		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 — Рівні комерційного потенціалу розробки

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

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного модулю є високим, що досягається за рахунок того, що програмний модуль не має відчизняних аналогів, а зарубіжні системи є дорогі та важкодоступні.

## 5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (5.1)$$

де  $M$  — місячний посадовий оклад конкретного розробника (дослідника), грн.;

$T_p$  — число робочих днів в місяці, 22 днів;

$t$  — число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 5.3.

Таблиця 5.3 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	52000	2363,64	43	101636,364
Програміст	45000	2045,45	43	87954,545
Всього				189590,91

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

Додаткова заробітна плата розробників, які приймали участь в розробці обладнання.

Додаткова заробітна плата прийнято розраховувати як 15 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o \cdot 15 \% / 100 \% , \quad (5.2)$$

$$Z_d = (189590,91 \cdot 15 \% / 100 \% ) = 28438,64 \text{ (грн.)}$$

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$H_3 = (Z_o + Z_d) \cdot 22 \% / 100\% \quad (5.3)$$

$$H_3 = (189590,91 + 28438,64) \cdot 22 \% / 100 \% = 47966,50 \text{ (грн.)}$$

Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

$$A = \frac{Ц}{T_в} \cdot \frac{t_{вик}}{12} [\text{Грн.}]. \quad (5.4)$$

де Ц — балансова вартість обладнання, грн.;

T — термін корисного використання обладнання згідно податкового законодавства, років

$t_{вик}$  — термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 32000 грн., термін його корисного використання згідно податкового законодавства — 2 роки, а термін його використання — 1,95 міс.

$$A_{обл} = \frac{32000}{2} \times \frac{1,95}{12} = 2660,06 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 5.4.

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних нема-

теріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю,  $V_{нем.ак.} = 600$  грн.

Таблиця 5.4 — Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія (Lenovo Legion 5)	32000	2	1,95	2606,061
Офісне обладнання (меблі)	25000	4	1,95	1017,992
Приміщення	1200000	20	1,95	9772,727
Всього				13396,78

Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_{\Pi}, \quad (5.5)$$

де  $V$  — вартість 1 кВт-години електроенергії для 1 класу підприємства,  $V = 31.6$  грн./кВт;

$P$  — встановлена потужність обладнання, кВт.  $P = 0,3$  кВт;

$\Phi$  — фактична кількість годин роботи обладнання, годин.

$K_{\Pi}$  — коефіцієнт використання потужності,  $K_{\Pi} = 0,9$ .

$$V_e = 0,9 \cdot 0,3 \cdot 8 \cdot 43 \cdot 31,6 = 8805,024 \text{ (грн.)}$$

Інші витрати та загальновиробничі 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» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{iv}}{100\%}, \quad (5.6)$$

де  $H_{iv}$  — норма нарахування за статтею «Інші витрати».

$$I_e = 189590,91 \cdot 95\% / 100\% = 180111,4 \text{ (грн.)}$$

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 науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (Z_o + Z_p) \cdot \frac{H_{нзв}}{100\%}, \quad (5.7)$$

де  $H_{нзв}$  — норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 189590,91 \cdot 145\% / 100\% = 274907 \text{ (грн.)}$$

Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 189590,91 + 28438,64 + 47966,50 + 13396,78 + 600 + 8805,024 + 180111,4 + 274907 = 743816,028 \text{ грн.}$$

Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються  $ZB$ , визначається за формулою:

$$ZB = \frac{B_{заг}}{\eta} \text{ (грн)}, \quad (5.8)$$

де  $\eta$  — коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то  $\eta=0,1$ ; технічного проектування, то  $\eta=0,2$ ; розробки конструкторської документації, то  $\eta=0,3$ ; розробки технологій, то  $\eta=0,4$ ; розробки дослідного зразка, то  $\eta=0,5$ ; розробки промислового зразка, то  $\eta=0,7$ ; впровадження, то  $\eta=0,9$ . Оберемо  $\eta = 0,5$ , так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ZB = 743816,028 / 0,5 = 1487632,06 \text{ грн.}$$



### 5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

- вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;
- зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);
- кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;
- визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунків економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

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

$$\Delta\Pi_i = (\pm\Delta C_0 \cdot N + C_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (5.10)$$

де  $\pm\Delta C_0$  — зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

$N$  — кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$C_0$  — основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки,  $C_0 = C_b \pm \Delta C_0$ ;

$C_b$  — вартість програмного продукту у році до впровадження результатів розробки;

$\Delta N$  — збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

$\lambda$  — коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт  $\lambda = 0,8333$ .

$\rho$  — коефіцієнт, який враховує рентабельність продукту;

$\vartheta$  — ставка податку на прибуток, у 2023 році  $\vartheta = 18\%$ .

Припустимо, що при прогнозованій ціні 350 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 50 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 50000 шт., протягом другого року — на 75000 шт., протягом третього року на 100000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*50 + (350 + 50)*50000)*0,8333*0,33*(1 - 0,18) = 3946249,842 \text{ грн.}$$

$$\begin{aligned} \Delta\Pi_2 &= (0*50 + (350 + 50)*(50000+75000))*0,8333*0,33*(1 - 0,18) = \\ &11274999,549 \text{ грн.} \end{aligned}$$

$$\begin{aligned} \Delta\Pi_3 &= (0*50 + (350 + 50)*(50000+75000+100000))*0,8333*0,33*(1 - 0,18) \\ &= 20294999,18 \text{ грн.} \end{aligned}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 35516248,58 грн.

Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків  $\Pi\Pi$ , що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$\Pi\Pi = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (5.11)$$

де  $\Delta\Pi_i$  – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

$T$  – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

$\tau$  — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні,  $\tau = 0,05 \dots 0,15$ ;

$t$  — період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$\begin{aligned} \text{ПП} = & (3946249,842/(1+0,1)^1) + (11274999,549/(1+0,1)^2) + (20294999,188/ \\ & / (1+0,1)^3) = 3587499,86 + 9318181,445 + 15247933,27 = 28153614,58 \text{ грн.} \end{aligned}$$

Далі розраховують величину початкових інвестицій  $PV$ , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{инв} * ZB, \quad (5.12)$$

де  $k_{инв}$  — коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай  $k_{инв} = 2 \dots 5$ , але може бути і більшим;

$ZB$  — загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 1487632,06 = 2975264,11 \text{ грн.}$$

Тоді абсолютний економічний ефект  $E_{abc}$  або чистий приведений дохід (NPV, Net Present Value) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = \text{ПП} - PV, \quad (5.13)$$

$$E_{abc} = 28153614,58 - 2975264,11 = 25178450,4 \text{ грн.}$$

Оскільки  $E_{abc} > 0$  то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (IRR, Internal Rate of Return) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій  $E_e$ . Для цього використаємо формулу:

$$E_e = \sqrt[T_{жс}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (5.14)$$

$T_{жс}$  — життєвий цикл наукової розробки, роки.

$$E_e = \sqrt[3]{(1 + 25178450,4/2975264,11) - 1} = 1,115$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (5.15)$$

де  $d$  — середньозважена ставка за депозитними операціями в комерційних банках; в 2023 році в Україні  $d = (0,09...0,14)$ ;

$f$  — показник, що характеризує ризикованість вкладень; зазвичай, величина  $f = (0,05...0,5)$ .

$\tau_{min}$ .

Так як  $E_b > \tau_{min}$ , то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_b}, \quad (5.16)$$

$$T_{ок} = 1 / 1,115 = 0,89 \text{ р.}$$

Оскільки  $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,89 роки, то фінансування даної наукової розробки є доцільним.

Розрахунком витрат на розробку нового програмного модулю є сума, яка складає 1487632 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний модуль є висококонкурентоспроможним. Період окупності складе близько 0,89 роки.

## ВИСНОВКИ

Оглядаючись на сучасний розвиток інформаційних технологій та обчислювальної техніки існує актуальна проблема прискорення розрахунків акустичних поверхонь у багатоканальних аналого-цифрових системах. Запропонований метод, що використовує технологію CUDA для паралельних обчислень, покликаний значно полегшити цей процес за рахунок використання графічних процесорів від Nvidia.

При виконанні даної роботи було розроблено модуль програмного забезпечення який закликаний вирішити проблему розрахунку акустичних поверхонь в багатоканальних аналого-цифрових система.

У першому розділі магістерської роботи проведений огляд існуючих методів і засобів акустичної локації та ідентифікації об'єктів на місцевості, також розглянуто загальну структурну схему, склад та принципи роботи АЦ-системи аудіо-локації й ідентифікації об'єктів на місцевості. Розроблено технічні вимоги до параметрів апаратних пристроїв і функцій програмного забезпечення АЦ-систем.

У другому розділі магістерської роботи було проаналізовано технології обробки звукових сигналів при багатоканальному скануванні в системі мікрофонних решіток. Розглянуто конструктивно-схемотехнічні рішення мікрофонних решіток

У третьому розділі магістерської роботи описаний процес розробки програмного модуля, для формування якого було використано такі технології: C++, C# та CUDA – технології для можливості багатоканальних обчислень за допомогою графічного процесора. Розробка відбувалась у інтегрованому середовищі розробки Visual Studio.

У четвертому розділі магістерської роботи описано перевірку працездатності розробленого програмного продукту та проведено тестування й експериментальне дослідження його роботи.

У п'ятому розділі магістерської роботи виконані економічні розрахунки із обґрунтування доцільності виконання нового модуля для обчислення акустичних поверхонь в багатоканальній аналого-цифровій, обчислені фінансові затрати

на створення програмного модуля та визначено економічні переваги від впровадження запропонованого наукового рішення у вигляді завершеного програмного модуля.

Розроблений модуль відповідає всім вимогам, що були поставлені при його проектуванні. Дане програмне рішення було протестоване, програмних збоїв та помилок зафіксовано не було.

Даний програмний модуль є актуальним на ринку та повністю готовий до запуску у відкрите користування.



## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Д. А. Савчук, Л. В. Крупельницький / Прискорення цифрової обробки низькочастотних сигналів в аналого-цифрових системах реального часу за допомогою технології CUDA // Тези доповіді. ЛІІ Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2023). Вінниця 2023 р. Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2023/paper/view/18399>
2. Звіт про науково-дослідну роботу «Високопродуктивні багатоканальні аналого-цифрові самокалібровані системи моніторингу й синхронного опрацювання низькочастотних сигналів» / ВНТУ, Вінниця, 2022 р., N держ. Реєстрації 0120U002205 – 308 с.
3. Арнаутов О.А., Момот Р.В., Худов Г.В. Методи визначення координат об'єктів в системах пасивної локації // Системи озброєння і військова техніка. – 2012. – № 3 (31). – С. 111–113.
4. Азаров О.Д. Аналого-цифрове порозрядне перетворення на основі надлишкових позиційних систем числення з ваговою надлишковістю: Монографія / Азаров О.Д. – Вінниця.– ВНТУ, 2010. – 232 с.
5. Крупельницький Л.В., Азаров О.Д. Аналого-цифрові пристрої систем, що самокоригуються, для вимірювань і оброблення низькочастотних сигналів: Монографія / Під ред. О.Д. Азарова.– УНІВЕРСУМ-Вінниця, 2005.– 167 с.
6. Ткаченко О.М. Метод швидкого пошуку найближчого сусіда з обчисленням відстані за зваженою евклідовою метрикою / О.М. Ткаченко, О.Ф. Грійо Тукало // Вісник Вінницького політехнічного інституту. – Вінниця: ВНТУ.– 2013.– №1.– С.116–122.
7. Методи та засоби для визначення напрямку та для ідентифікації джерел звуків на місцевості / Ткаченко О. М.; Крупельницький Л. В.; Дерев'яга Б.С.; Зінчук Р. С. // Матеріали XLV Науково-технічної конференції ВНТУ, Вінниця, 23-24 березня 2016 р.
8. Л.В. Крупельницький Характеристики і структури багатоканальних АЦ-систем, що самокоригуються, для аналізу аудіо сигналів // Тези доповідей

П'ятої Міжнародної науково-практичної конференції "Методи та засоби кодування, захисту й ущільнення інформації". –Україна.– Вінниця, 19-21 квітня 2016 р. – Вінниця: ВНТУ, 2016. – С. 129–133.

9. Ткаченко О.М. Пошук найближчого вектора у кодових книгах на основі бінарного дерева / О.М. Ткаченко, О.Ф. Грійо Тукало // Інформаційні технології та комп'ютерна інженерія: Міжнародний науково-технічний журнал.– Вінниця, 2014.– №3(31).– С. 67–55.

10. Степанова Т.М. Етапи розвитку мікрофонних решіток для комп'ютерних систем акустичної локації // Матеріали XLVII Науково-технічної конференція ВНТУ , Вінниця, 21-23 березня 2018 р.

11. Гарнага В.А. Параметри мікрофонних акустичних решіток і попередніх підсилювачів аудіосигналів // Матеріали XLVII Науково-технічної конференція ВНТУ , Вінниця, 21-23 березня 2018 р

12. Крупельницький Л.В., Азаров О.Д. Аналого-цифрові пристрої систем, що самокоригуються, для вимірювань і оброблення низькочастотних сигналів: Монографія / Під ред. О.Д. Азарова.– УНІВЕРСУМ-Вінниця, 2005.– 167 с.

13. NVIDIA DOCUMENTATION [Електронний ресурс] — Режим доступу до ресурсу: <https://docs.nvidia.com/cuda/>

14. Мова програмування C# і платформа .NET [Електронний ресурс] — Режим доступу до ресурсу: <https://metanit.com/sharp/tutorial/1.1.php>

15. CUDA Programming Guide [Електронний ресурс] — Режим доступу до ресурсу: <https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html>

16. ILGPU Documentation [Електронний ресурс] — Режим доступу до ресурсу: <https://www.ilgpu.net/docs/ReadMe>

17. Азаров О.Д. Аналого-цифрове порозрядне перетворення на основі надлишкових позиційних систем числення з ваговою надлишковістю: Монографія / Азаров О.Д. – Вінниця. – ВНТУ, 2010. – 232 с.

18. Спеціалізоване і вимірювальне обладнання власної розробки і виробництва для телерадіомовлення: Каталог НТЦ "Аналого-цифрові системи"

ВНТУ / Азаров О.Д., Крупельницький Л.В., Стейскал В.Я., Білоконь О.А. – Вінниця, 2015. – 40 с.

19. . Ракитянська Г.Б. Розв'язання систем нечітких логічних рівнянь у задачах оберненого виведення / Ракитянська Г.Б. // Вісник Національного університету "Львівська політехніка". Серія: Комп'ютерні науки та інформаційні технології: Збірник наукових праць. – 2015. – № 826. – с.259.

20. Ротштейн О.П., Ракитянська Г.Б. Діагностика на базі нечітких відношень в умовах невизначеності. – Вінниця: УНІВЕРСУМ, 2006. – 275 с.

21. Wang C. Hybrid uncertainty propagation of coupled structural–acoustic system with large fuzzy and interval parameters / Wang C., Qiu Z., Li Y. // Applied Acoustics. – 2016. – Vol. 102. – P. 62–70

**ДОДАТОК А**

Технічне завдання

Міністерство освіти та науки України

Вінницький національний технічний університет

Інститут інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТд.т.н., професор Азаров О. Д.

(наук. ст., вч. зв., ініц. та прізви.)

(підпис)

“29” Вересня 2023р.**ТЕХНІЧНЕ ЗАВДАННЯ**

на виконання магістерської кваліфікаційної роботи

Метод прискорення розрахунків акустичних поверхонь в багатоканальній аналого-цифровій системі

08-54.МКР.038.00.000 ТЗ

Науковий керівник: к.т.н., доц.каф. ОТ

Крупельницький Л.В.студент групи 2Кі-22мСавчук Д.А.

Вінниця 2023

## 1 Підстава для виконання магістерської кваліфікаційної роботи (МКР)

Підставою для розробки даної магістерської кваліфікаційної роботи є наказ ВНТУ №\_\_\_\_ від «\_\_» \_\_\_\_ 2023 року та рішення засідання кафедри обчислювальної техніки (протокол №\_\_ від «\_\_\_\_\_»\_\_ 2023 року).

## 2 Мета і призначення МКР

2.1 Мета полягає у розробці модуля програмного забезпечення для пришвидшення розрахунку акустичних поверхонь в багатоканальній аналого-цифровій системі за рахунок застосування методів CUDA- технологій, для ви використання в статичних і портативних системах.

2.2 Призначення роботи полягає у створенні програмного модуля для пришвидшення розрахунку акустичних сигналів в багатоканальній аналого-цифровій системі

## 3 Вихідні дані для виконання МКР

Тридцять два масива даних опрацьованого звукового сигналу із урахуванням затримок з частотою дискритизації 48 кГц, розрядністю 16-24 двійкових розряди.

## 4 Вимоги до виконання МКР

4.1 Провести аналіз методів та засобів звукової локації та ідентифікації.

4.2 Провести аналіз технології обробки низькочастотних звукових сигналів;

4.3 Запропонувати власне рішення програмного модуля для обробки акустичних поверхонь;

4.4 Розробити прототип розроблюваного рішення.

4.5 Провести експериментальні дослідження та тестування програмного модуля

5 Етапи МКР та очікувані результати приведені в табл. А.1.

Таблиця А.1 — Етапи МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз методів та засобів звукової локації та ідентифікації	20.09.2023р.	04.10.2023р.	Розділ 1
2	Аналіз технології обробки низькочастотних звукових сигналів	05.10.2023р.	18.10.2023р.	Розділ 2
3	Розробка програмного модуля	19.10.2023р.	01.11.2023р.	Розділ 3
4	Експериментальні дослідження та тестування	02.11.2023р.	15.11.2023р.	Розділ 4
5	Економічна частина	17.11.2023р.	25.11.2023р.	Розділ 5

### 6 Матеріали, що подаються до захисту МКР

Пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відзив наукового керівника, відзив опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, нормоконтроль про відповідність оформлення МКР діючим вимогам.

### 7 Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

## 8 Вимоги до оформлювання та порядок виконання МКР

### 8.1 При оформлювання МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— міждержавний ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;

— Методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 — «Комп'ютерна інженерія». Кафедра обчислювальної техніки ВНТУ 2022;

— документами на які посилаються у вище вказаних.

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21».

## ДОДАТОК Б

Структурна схема АЦ-системи аудіолокації та ідентифікації об'єктів на місцевості

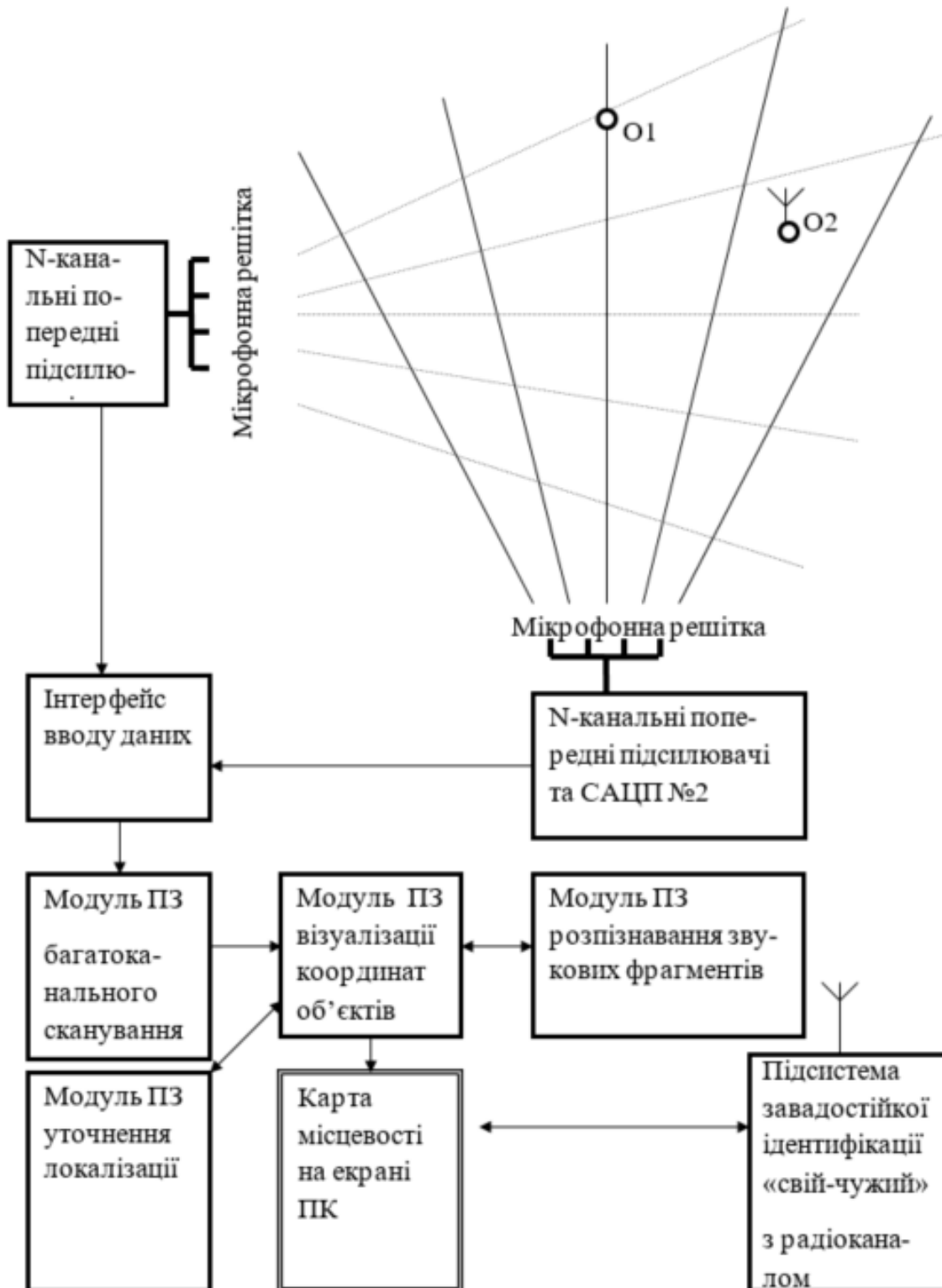


Рисунок Б — Структурна схема АЦ-системи аудіолокації та ідентифікації об'єктів на місцевості



## ДОДАТОК В

### Конструкція мікрофонних решіток двох типів



Рисунок В.1 — Зовнішній вигляд АЦ- системи портативного варіанту



Рисунок В.2 — Зовнішній вигляд АЦ- системи стаціонарного варіанту

## ДОДАТОК Г

Лістинг класу для конвертації wav-файлу

```
using System;
using NAudio.Wave;
class Program {
    static void Main() {
        string audioFilePath = "шлях до аудіофайлу.wav";
        string outputTextFilePath = "шлях до текстового файлу.txt";
        using (WaveFileReader waveFileReader = new WaveFileReader(audioFilePath)) {
            int sampleRate = waveFileReader.WaveFormat.SampleRate;
            int bitsPerSample = waveFileReader.WaveFormat.BitsPerSample;
            byte[] buffer = new byte[1024];
            using (var writer = new System.IO.StreamWriter(outputTextFilePath))
            {
                while (waveFileReader.Position < waveFileReader.Length) {
                    int bytesRead = waveFileReader.Read(buffer, 0, buffer.Length);
                    for (int i = 0; i < bytesRead; i += bitsPerSample / 8) {
                        short sampleValue = BitConverter.ToInt16(buffer, i);
                        writer.WriteLine(sampleValue);
                    }
                }
            }
        }

        Console.WriteLine("Конвертація завершена.");
    }
}
```

## ДОДАТОК Д

Лістинг основного коду програмного модуля

```
#include <stdio.h>
#include <iostream>
#include "cuda_runtime.h"
#include "device_launch_parameters.h"
#include <vector>
#include <fstream>
#include <sstream>

__global__ void calculateResult(const int* inputData, const int* matrix, int* result,
int rows, int cols, int channels) {
    int tid = blockIdx.x * blockDim.x + threadIdx.x;
    while (tid < rows) {
        for (int i = 0; i < cols; ++i) {
            int sum = 0;
            for (int j = 0; j < channels; ++j) {
                sum += matrix[i * channels + j] * inputData[tid * channels + j];
            }
            result[tid * cols + i] = sum / channels;
        }
        tid += blockDim.x * gridDim.x;
    }
}

int main() {
    std::vector<int> matrix = {
        -164, -82, 82, 164,
```

- 160, -80, 80, 160,  
- 154, -77, 77, 154,  
- 146, -73, 73, 146,  
- 138, -69, 69, 138,  
- 130, -65, 65, 130,  
- 120, -60, 60, 120,  
- 108, -54, 54, 108,  
- 98, -49, 49, 98,  
- 84, -42, 42, 84,  
- 72, -36, 36, 72,  
- 58, -29, 29, 58,  
- 44, -22, 22, 44,  
- 30, -15, 15, 30,  
- 14, -7, 7, 14,  
0, 0, 0, 0,  
14, 7, -7, -14,  
30, 15, -15, -30,  
44, 22, -22, -44,  
58, 29, -29, -58,  
72, 36, -36, -72,  
84, 42, -42, -84,  
98, 49, -49, -98,  
108, 54, -54, -108,  
120, 60, -60, -120,  
130, 65, -65, -130,  
138, 69, -69, -138,  
146, 73, -73, -146,  
154, 77, -77, -154,

```

160, 80, -80, -160,
164, 82, -82, -164,
};
std::vector<std::vector<int>> inputData; // Дані з усіх файлів
int maxRows = 0;
for (int i = 0; i < 4; ++i) {
    std::string filename = "C:/Users/dimas/Desktop/Data/chanel_" + std::to_string(i
+ 1) + ".txt";
    std::ifstream inputFile(filename);
    if (!inputFile.is_open()) {
        std::cerr << "Unable to open file: " << filename << std::endl;
        return 1; // ПОМИЛКА
    }
    std::vector<int> fileData;
    int value;
    while (inputFile >> value) {
        fileData.push_back(value);
    }
    maxRows = std::max(maxRows, static_cast<int>(fileData.size()));
    inputFile.close();
    fileData.resize(maxRows, 0);
    inputData.push_back(fileData);
}
int* d_inputData, * d_matrix, * d_result;
cudaMalloc(&d_inputData, maxRows * 4 * sizeof(int));
cudaMalloc(&d_matrix, 31 * 4 * sizeof(int));
cudaMalloc(&d_result, maxRows * 31 * sizeof(int));

```

```

for (int i = 0; i < 4; ++i) {
    cudaMemcpy(d_inputData + i * maxRows, inputData[i].data(), maxRows *
sizeof(int), cudaMemcpyHostToDevice);
}

cudaMemcpy(d_matrix, matrix.data(), 31 * 4 * sizeof(int),
cudaMemcpyHostToDevice);

int threadsPerBlock = 256;

int blocksPerGrid = (maxRows + threadsPerBlock - 1) / threadsPerBlock;

cudaEvent_t start, stop;
cudaEventCreate(&start);
cudaEventCreate(&stop);
cudaEventRecord(start);

calculateResult << <blocksPerGrid, threadsPerBlock >> > (d_inputData, d_matrix,
d_result, maxRows, 31, 4);

cudaEventRecord(stop);
cudaEventSynchronize(stop);

std::vector<int> result(maxRows * 31);

cudaMemcpy(result.data(), d_result, maxRows * 31 * sizeof(int),
cudaMemcpyDeviceToHost);

std::ofstream outputFile("C:/Users/dimas/Desktop/Data/resaltMatrix.txt");

if (outputFile.is_open()) {
    for (int i = 0; i < maxRows; ++i) {
        for (int j = 0; j < 31; ++j) {
            outputFile << result[i * 31 + j];
            if (j < 30) {
                outputFile << " ";
            }
        }
    }

    outputFile << std::endl;
}

```

```
    }  
    outputFile.close();  
    std::cout << "The results are saved to the file resultMatrix.txt" << std::endl;  
}  
else {  
    std::cerr << "Could not open file to write result" << std::endl;  
    return 1; // Помилка  
}  
float milliseconds = 0;  
cudaEventElapsedTime(&milliseconds, start, stop);  
std::cout << "Execution time: " << milliseconds << " milliseconds" << std::endl;  
cudaDeviceProp prop;  
cudaGetDeviceProperties(&prop, 0);  
std::cout << "Video card information:" << std::endl;  
std::cout << "Product name: " << prop.name << std::endl;  
std::cout << "Memory capacity: " << prop.totalGlobalMem / (1024 * 1024) << "  
MB" << std::endl;  
std::cout << "Number of multiprocessors: " << prop.multiProcessorCount <<  
std::endl;  
cudaFree(d_inputData);  
cudaFree(d_matrix);  
cudaFree(d_result);  
cudaEventDestroy(start);  
cudaEventDestroy(stop);  
return 0;  
}
```

## ДОДАТОК Е

Програмна модель реконструкції акустичних поверхонь в системі MATLAB

```

function ff_int
ff_int=ff_point(FuzzInpLow,FuzzOutLow)+ff_point(FuzzInpUp,FuzzOutUp);
function z = ff_point(A_exp,B_exp)
sum = 0;
min1 = zeros(1, NumRow);
max1 = zeros(1, DataSize);
for j = 1:DataSize,
    for i = 1:NumRow, min1(i) = min([A_exp(j,i), x(i)]);    end;
    max_min=0;
    for s= 1:NumRow,
        if min1(s)>max_min, max_min=min1(s); end;
    end
    max1(j) = max_min;
    sum = sum + (B_exp(j) -max1(j))^2;
end;

z = sum;

function InvMatrix()

NumRow=100;
NumCol=7;
DataSize=3665;

load FuzzInpLow.txt
load FuzzOutLow.txt
load FuzzInpUp.txt
load FuzzOutUp.txt

global VectorLow VectorUp FuzzInpLow FuzzInpUp;

for j = 1:NumCol,
    for i = 1:DataSize,
        VectorLow(j) = FuzzOutLow(j, i);
        VectorUp(j) = FuzzOutUp(j, i);

    end;
    IntSearch_U1LS;
end;

function [x,fval] = IntSearch_U1LS

```



```

% Fitness function and number of variables
fitnessFcn = @ff_int;
numberOfVariables = NumRow;

% Bound = [LB;UB];
LB = zeros(1,numberOfVariables);
UB = ones(1,numberOfVariables);

Bound = [LB;UB];
Generation = 20; Population = 25;

priv_fval = NumRow;

% Кількість запусків ГА для пошуку максимальної границі
numOfGA = 10;
for iterator_j = 1:numOfGA,
    options = gaoptimset('CreationFcn',@int_pop,'MutationFcn',@int_mutation, ...
        'PopInitRange',Bound,'Display','final', 'StallGenL',40,'Generations',Generation, ...
        'PopulationSize',Population,'PlotFcns',{ @gaplotbestf,@gaplotbestindiv });
    [x,fval] = ga(fitnessFcn,numberOfVariables,options);

    if fval <= priv_fval,
        priv_fval = fval;
    else
        continue;
    end;
    LB = x;
    Bound = [LB;UB];
end;

MAX = LB;
UB = ones(1,numberOfVariables);
LB = zeros(1,numberOfVariables);
Bound = [LB;UB];
Generation = 25;
Population = 25;

% Достатня різниця між знайденою мінімальною границею і попередніми гра-
ницями
LIMIT_MIN = 0.05;
% Достатня кількість мінімальних розв'язків
sizeOfMins = 100;

% Кількість знайдених мінімальних рішень
iterator_mins = 0;

```

```

iterator_var = 0;
mins = zeros(sizeOfMins, numberOfVariables);

% Кількість запусків ГА для пошуку мінімальних рішень
numOfSearch = 1000;
% Кількість запусків ГА для пошуку однієї границі
numOfGA = 5;

for iterator_k = 1:numOfSearch,
    UB=MAX;
    Bound = [LB;UB];

    for iterator_j = 1:numOfGA,
        options = gaoptimset('CreationFcn',@int_pop,'MutationFcn',@int_mutation, ...
            'PopInitRange',Bound,'Display','final', 'StallGenL',40,'Generations',Generation,
            ...
            'PopulationSize',Population,'PlotFcns',{ @gaplotbestf,@gaplotbestindiv });

        [x,fval] = ga(fitnessFcn,numberOfVariables,options);

        if fval <= priv_fval, priv_fval = fval; else continue; end;

        UB = x;
        Bound = [LB;UB];
    end;

    for l = 1:iterator_mins,
        iterator_var = 0;
        for g = 1:numberOfVariables,
            if mins(l,g) - UB(g) < LIMIT_MIN,
                if mins(l,g) - UB(g) > (-1)*LIMIT_MIN,
                    iterator_var = iterator_var + 1;
                end;
            end;
        end;
        if iterator_var == numberOfVariables,
            break;
        end;
    end;

    if iterator_var ~= numberOfVariables,
        iterator_mins = iterator_mins + 1;
        for g = 1:numberOfVariables,
            mins(iterator_mins,g) = UB(g);
        end;
    end;
end;

```

```

    end;

    if iterator_mins > sizeOfMins,
        break;
    end;
end;
mins
MAX
% Mutation function to generate childrens satisfying the range
% constraints on decision variables.
function mutationChildren = int_mutation(parents,options,GenomeLength, ...
    FitnessFcn,state,thisScore,thisPopulation)
shrink = .01;
scale = 1;
scale = scale - shrink * scale * state.Generation/options.Generations;
range = options.PopInitRange;
lower = range(1,:);
upper = range(2,:);
scale = scale * (upper - lower);
mutationPop = length(parents);
mutationChildren = repmat(lower,mutationPop,1) + ...
    repmat(scale,mutationPop,1) .* rand(mutationPop,GenomeLength);
% End of mutation function
%-----
function Population = int_pop(GenomeLength,FitnessFcn,options)
totalpopulation = sum(options.PopulationSize);
range = options.PopInitRange;
lower= range(1,:);
span = range(2,:) - lower;
Population = repmat(lower,totalpopulation,1) + ...
    repmat(span,totalpopulation,1) .* rand(totalpopulation,GenomeLength);
% End of creation function

```

**ДОДАТОК Ж**

Протокол перевірки кваліфікаційної роботи

**ПРОТОКОЛ ПЕРЕВІРКИ НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ)  
РОБОТИ**Назва роботи: Метод прискорення розрахунків акустичних поверхонь в бака-  
токанальній аналого-цифровій системіТип роботи: \_\_\_\_\_ магістерська кваліфікаційна робота \_\_\_\_\_  
(БДР, МКР)Підрозділ \_\_\_\_\_ кафедра обчислювальної техніки \_\_\_\_\_  
(кафедра, факультет)**Показники звіту подібності  
Unicheck**

Оригінальність \_\_\_\_\_ 81,7% \_\_\_\_\_ Схожість \_\_\_\_\_ 18,3% \_\_\_\_\_

Аналіз звіту подібності (відмітити потрібне):

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

Особа, відповідальна за перевірку \_\_\_\_\_  
(підпис)Захарченко С.М.  
(прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи \_\_\_\_\_  
(підпис)

\_\_\_\_\_ (прізвище, ініціали)

Керівник роботи \_\_\_\_\_

\_\_\_\_\_