КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ВИСОКИХ ТЕХНОЛОГІЙ

Завідувач кафедри нанофізики конденсованих середовищ
проф. Валерій Антонович Скришевський
Протокол №____ засідання кафедри
від “____” _____20___ р.

ПОРТАТИВНИЙ АКУСТИЧНИЙ СПЕКТРОМЕТР

Випускна кваліфікаційна робота бакалавра
студента спеціальності 105
Прикладна фізика та наноматеріали
ОП «Нанофізика та комп’ютерні технології»
Щура Максима Володимировича

Науковий керівник
доцент кафедри нанофізики
конденсованих середовищ
к.ф.-м.н **Іванов Іван Іванович**

Оцінка захисту роботи
_____

Київ – 2024 р.

АНОТАЦІЯ

Щур М. В. Портативний акустичний спектрометр – Випускна кваліфікаційна робота бакалавра за спеціальністю 105 Прикладна фізика та наноматеріали ОП «Нанофізика та комп’ютерні технології».

У ході роботи проведено огляд літератури та інших матеріалів на тему акустичного спектрометра, методи його реалізації. Було розглянуто різні підходи виконання проекту, порівняно переваги та недоліки методик, визначено в яких випадках краще застосовуються, обмірковані варіації подальших потенціальних покращень чи змін для іншого використання. Зроблено портативний акустичний спектрометр. Отримані результати можуть бути використані для дослідження та порівняння звукових характеристик, також можуть бути використані в подальшій роботі для розпізнавання звуку, роботі зі звуковими командами тощо.

Ключові слова: звук; перетворення Фур’є; вейвлет-перетворення.

ЗМІСТ

Перелік умовних позначень

ПФ (FT, Fourier Transformation) – перетворення Фур’є

ДПФ (DFT, Discrete Fourier Transform) – дискретне перетворення Фур’є

ШПФ (FFT, Fast Fourier Transformation) – швидке перетворення Фур’є

ВПФ (WFT, Windowed Fourier Transformation) – віконне перетворення Фур’є

ВП (WT, Wavelet Transformation) – вейвлет-перетворення

TinyML (Tiny Machine Learning) – дрібне машинне навчання

IP (IoT, Internet of Things) – Інтернет речей – мережа взаємозв’язаних фізичних пристроїв, що обмінюються даними

РНМ (RNN, Recurrent Neural Networks) – рекурентна нейронна мережа

ДКП (LSTM, Long Short-Term Memory) – довга короткочасна пам’ять

СРМ (SVD, Singular-Value Decomposition) – сингулярний розклад матриці

ЗПКС (WFST, Weighted Finite State Transducers) – зважені перетворювачі кінцевих станів

БПЛА – безпілотний літальний апарат

ППІ (SPI, Serial Peripheral Interface) – послідовний периферійний інтерфейс – стандарт передачі даних для сполучення мікроконтролера з приладами

RP2040w – Raspberry Pi Pico W

Вступ

Слух – один з ключових органів чуття для розпізнавання інформації про зовнішній світ для людини, другий по важливості після зору. З машинного погляду ідентифікація зображення потребує більше ресурсів, баз даних і складніше порівняно з розпізнаванням звуку, до того ж що нинішні технології ще досі намагаються наздогнати людські можливості в плані зору в той час, як можливості звукового аналізу машини давно перегнали діапазон сприйнятливий для людини. Для створення акустичного сенсора достатньо лише мікрофона та мікроконтролера для аналізу звуку. Відомо, що приблизні межі слуху це 20-20000 Гц, нижні й верхні межі індивідуальні, тобто варіюється між людьми й мають нижчий порок з віком. Межі для розпізнавання комп'ютером залежать лише від якості мікрофона чи іншого звукового сенсора. Розпізнавання голосу, голосових команд та дослідження звуку стає все більш повсякденним, знаходить більше застосувань в житті в найрізноманітніших сферах та має великий потенціал.

Один з нових цікавих екземплярів використання портативного акустичного спектрометра був представлений на виставці Brave1: акустичні сенсори для детектування повітряних цілей Zvook [Додаток 1]. Концепт мав операторів, згодом був покращений штучним інтелектом для якіснішої ідентифікації та продовжує удосконалюватись. Дешевий, доступний, легко масштабований, має перевагу при цілях на низькій висоті, що містять менше металевих компонентів, та більш радіопроникних загроз (як, наприклад, шахіди погано помітні в радіодіапазоні й гучні) порівняно з традиційними радіолокаційними системами, метод здатний доповнювати традиційну систему: ці 2 способи здатні перекривати мінуси один одного для більшої надійності, швидшої адаптації до нових загроз, зменшення кількості неправдивих спрацювань систем, збагачення інформації загальної інтегрованої спільної системи. Крім того, ці сенсори непомітні для радіоелектронних засобів розвідки та економічно не вигідні для ураження

при своїй вартості в 500 доларів. Наразі наявна децентралізована мережа з понад 180 акустичних сенсорів, кожний з яких може виявляти повітряні цілі на відстані до 10 км. Сенсор довго працює, мало споживає: 10 Вт живлення при 5 В.

Як покращення можна використати технологію мікрофонної решітки для точнішого позиціювання, азимутна інформація дасть якісніше прицілювання для мобільних вогневих груп, прожекторам протиповітряної оборони чи перехоплювачам дронів. Параболічне дзеркало з мікрофоном у фокусі здатне краще зосереджувати звук на датчику. Дослідження звуку є корисним процесом для багатьох сфер: від музики й комунікації до голосових команд та військової справи.

Актуальність роботи: звукові розпізнавальні системи та пристрої акустичної спектрометрії активно розвиваються та знаходять нові місця застосування, лише недавно були створені такі корисні голосові асистенти як Гугл помічник, Алекса, перекладачі мов тощо; якісний звуковий аналіз допомагає краще зрозуміти шумові характеристики об’єктів, що в деяких сферах та застосуваннях має значну роль.

Мета роботи: розглянути теорію обробки звукового сигналу і зробити портативний акустичний спектрометр.

Розділ 1.

Теоретична частина

1.1. Перетворення Фур’є

В часи активних ядерних випробувань вибухи в на землі чи під водою не було складно виявити, проте підземні тести виявилися значно складнішими для детектування. Сейсмографи здатні зафіксовувати слабкі поштовхи від підземних вибухів, проте було незрозуміло як відрізнити вібрації від ядерних випробувань від звичайних підземних поштовхів, які часто відбуваються. Також це дозволяло стежити за успіхами супротивників визначаючи потужність вибухів. Проте чим глибше відбувалось випробування, тим складніше було зафіксувати. Це була надзвичайно важлива задача для вчених і необхідна інформація лежала в сейсмографі, але була складність зчитати її, для ідентифікації треба було знати скільки і які частоти зійшлись. Це і є задача для перетворення Фур’є, яке необхідно було застосувати. Якби вчені на початку 60-их років були впевнені що зможуть дистанційно виявляти підземні випробування ядерної зброї й відкрили ШПФ раніше, можна було б досягти детектування цих вибухів будь-де і відповідно повної заборони ядерних випробувань, зупинивши ядерні перегони до їх початку.

ПФ також застосовується для стиснення даних. Стиснення даних про звук має кодування, яке для пропускної смуги потокового аудіо зменшує обсяг файлів звуку, називають аудіокодеками (найпопулярніший приклад такої програми це Windows Media Player). Існує 2 види стиснення звуку: без втрат (надає можливість відновити вихідні дані без спотворень, як вільний аудіокодек без втрат) і з втратами (не можливо повністю відновити без спотворень, як ПФ). При використанні стиснення з втратами це дає більше стиснення і більше простору для запису інформації: якщо взяти аудіо компакт-диск з записаною не стисненою музикою, довжиною в годину, тоді після стиснення без втрат цей же компакт-диск зможе містити до 2 годин

музики й після стиснення з втратами з середнім бітрейтом 7-10 годин. Бітрейт – головний параметр для стиснень з втратами, за його показником визначається наскільки стиснений файл та його якість, бітрейти поділяють на постійний, змінний та усереднений бітрейти.

Стиснення з втратами широко застосовується для аудіо, цифрового телебачення, радіо та потокового медіа в Інтернеті, цей метод може використовувати психоакустику, щоб виявити частоти, що не сприймає людина, тому вони передаються з меншою точністю чи взагалі не передаються. Для маскування сигналу з послідовністю відліків амплітуди на часу здійснюється перетворення на послідовність спектрів звуків, де кожна частина спектра кодується відокремлено [1].

ПФ – спосіб розкласти сигнал на його чіткі синусоїди з певною амплітудою і частотою, перетворення є зворотним. Щоб дізнатися скільки певної синусоїди в деякому сигналі (тобто знайти схожість синусоїди) необхідно помножити цю синусоїду на вибраний сигнал у кожній точці, після чого додати площі утворені під кривою (рисунок 1.1). Чим менша площа знизу, тим більша схожість синусоїди з сигналом, якщо площі знизу немає (добуток функцій завжди додатній), то хвиля відповідає частоті сигналу та вони корельовані (рисунок 1.2). Це вказує чи є хвиля компонентом сигналу, частота підібраної корельованої хвилі також характерна досліджуваному сигналу [2].

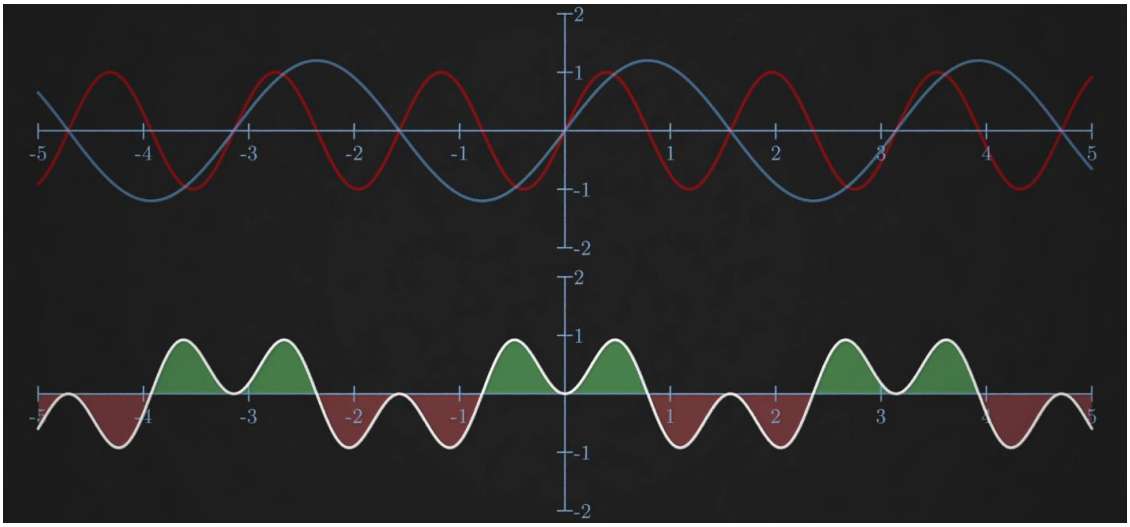


Рисунок 1.1. Приклад некорельованих хвиль.

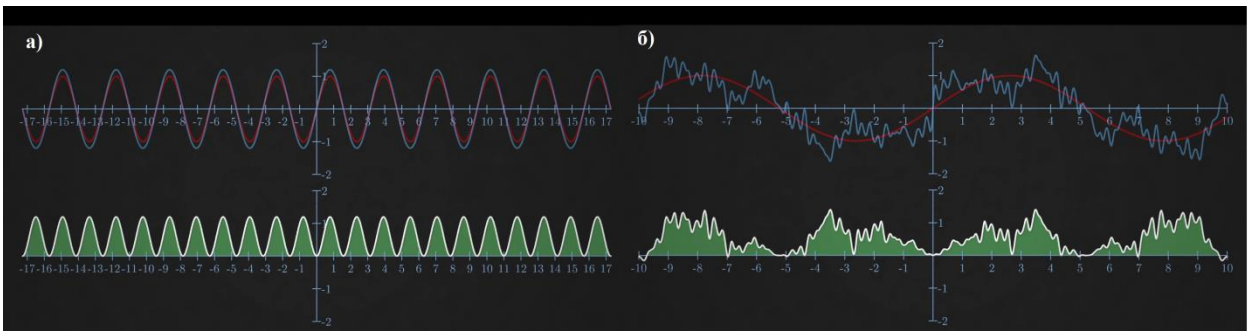


Рисунок 1.2. Приклад корельованих хвиль з однією частотою (а) та з різними частотами (б).

Розмір площі вказує на відносну амплітуду синусоїдальної хвилі у складі сигналу (тобто інтенсивність певної хвилі в досліджуваному сигналі). Повторюючи для кожної частоти синусоїди буде результат інтенсивності кожної з них в досліджуваному сигналі, тобто в результаті буде спектр частот (рисунок 1.3). Це показує які частоти є в сигналі та їхню пропорцію.

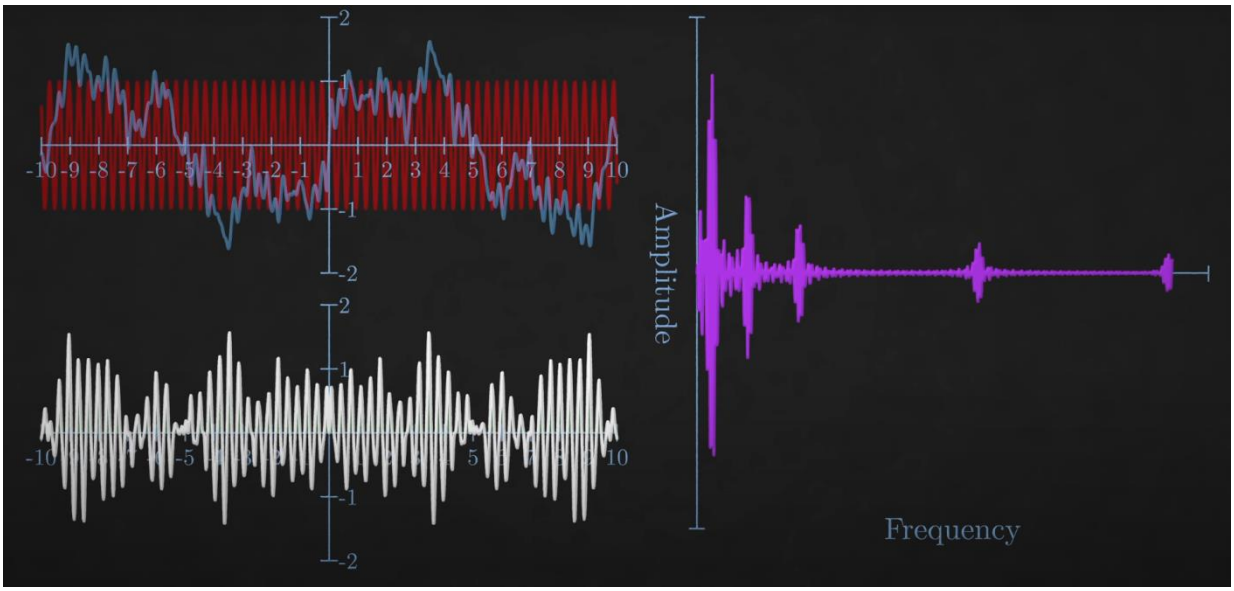


Рисунок 1.3. Спектр частот сигналу.

Проте якщо в сигналі будуть хвилі з функцією косинуса, то таким способом не можна представити лише через зміну частоти та амплітуди функції синуса. Якщо для базової функції косинуса взяти функцію синуса з тією ж амплітудою та частотою, то хвилі будуть не корельовані й площа знизу буде рівною площі зверху графіка, це буде вказувати що дана частота не належить сигналу. В такому випадку необхідно для кожної частоти виконувати множення на синус, на косинус, потім шукати амплітуду 2

випадків, їхнє співвідношення вказує фазу сигналу. Варто зазначити, що ПФ – інтегральне перетворення, яке приймає комплексну функцію як вхідні дані та виводить іншу комплексну функцію, що описує наявність різних частот початкового сигналу. Амплітуду синусів та косинусів можна обчислювати окремо або за формулою Ейлера (1), що використовує єдиний експоненціальний множник, дійсна частина рівняння – амплітуда косинуса, а уявна – синуса. Математично перетворення Фур'є є комплексно значною функцією F(ω) задається інтегралом (2) [3].

$$\cos(2\pi \mathrm{ft}) + \mathrm{i}\sin(2\pi \mathrm{ft}) = e^{i2\pi \mathrm{ft}} \quad (1)$$

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt \quad (2)$$

Реальні сигнали не є неперервними нескінченними хвилями та після ПФ не отримується нескінченний неперервний спектр частот. Реальні сигнали обмежені та складаються з окремих фрагментів чи точок даних. Прилад не має нескінченну точність аналізу даних, завжди є фрагментарність, тому для отриманого набору точок даних не можливо застосувати ПФ в ідеальному варіанті. Для подібних випадків застосовується дискретне перетворення Фур'є.

1.2. Дискретне перетворення Фур'є

ДПФ – розповсюджена варіація ПФ цифрової обробки дискретного сигналу задля дослідження його гармонічних або частотних характеристик. Процедура надає можливості для аналізу, перетворення й створення сигналів способами, що не доступні при аналоговій (неперервній) обробці. Одна з особливостей ДПФ: спектр частот дискретний і кінцевий (рисунок 1.4). Математично ДПФ перетворює набір дискретних значень $\{x_n\} := x_0, x_1, \ldots, x_{N-1}$ в набір комплексних чисел $\{X_k\} := X_0, X_1, \ldots, X_{N-1}$, що представлено формулою (3).

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi \frac{k}{N} n} \quad (3)$$

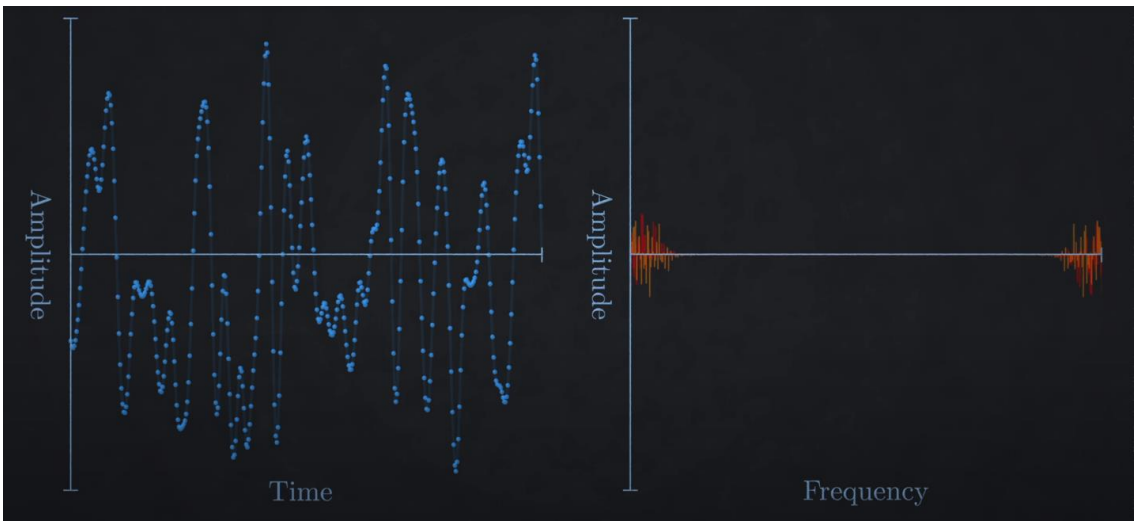


Рисунок 1.4. Приклад графіку сигналу сейсмометра з точок даних і його ПФ.

Частоти потрапляють у певне число інтервалів, їхня кількість і розмір відповідає кількості та щільності точок вихідного сигналу. Чим далі точки віддалені одне від одного, тим нижча максимальна вимірювальна частота, бо дані розташовані не достатньо близько щоб зафіксувати коливання високої частоти (рисунок 1.5).

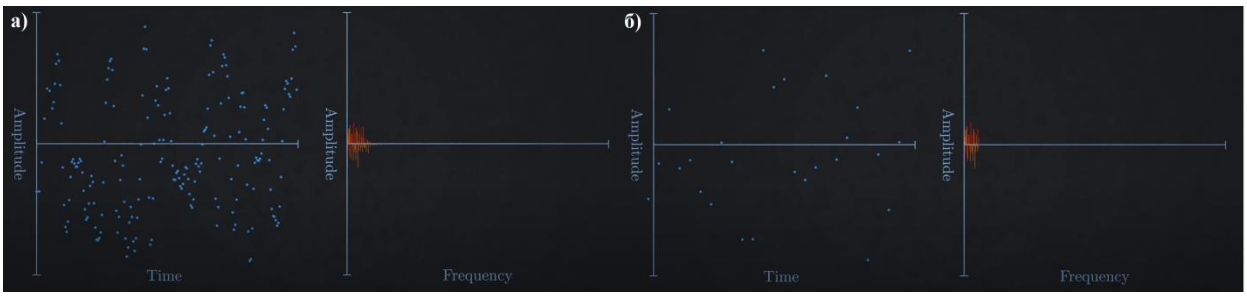


Рисунок 1.5. Порівняння і вплив зменшення щільності точок даних на отриманий спектр від початкового сигналу з рисунка 1.4.

Чим коротший сигнал, тим важче відокремити й відрізнити схожі частоти, що знижує частотну роздільну здатність. Тобто коротший сигнал спричиняє ширші інтервали частот (рисунок 1.6). Найнижча ненульова частота, яку можна виміряти, має період, що дорівнює тривалості сигналу, а інтервали з вищими частотами кратні найнижчій частоті (вони вміщаються у сигнал 2, 3, 4 тощо разів). Загальна кількість інтервалів рівна числу фрагментів сигналу. Наприклад, для 8 точок при ДПФ буде 8 інтервалів

(стовпчиків) частоти на спектрі (рисунок 1.6, б). Спочатку кожна точка множиться на 1 і додається разом. Наступний стовпчик (F_1) відповідає частоті, що вміщується лише 1 раз у сигнал, тобто її період збігається з довжиною сигналу. Кожна точка множиться на синус і косинус цієї частоти, потім окремо додаються. Процес продовжується аналогічно для третього стовпчика (F_2), синус частоти якого вміщується двічі в сигналі (період синуса рівний половині довжині сигналу), для четвертого стовпчика, синус частоти якого вміщується тричі в сигналі й так до останнього стовпчика [4].

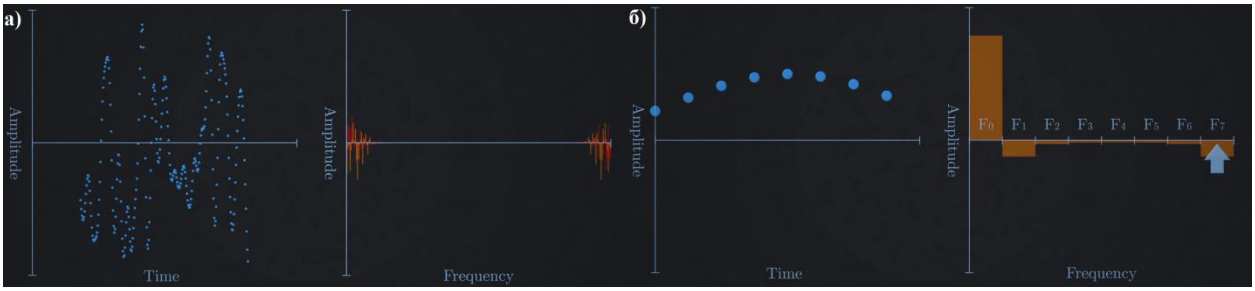


Рисунок 1.6. Порівняння і вплив зменшення тривалості сигналу на отриманий спектр від початкового сигналу з рисунка 1.4.

Метод дуже корисний і практичний для дискретних випадків, чудово працює, коли треба застосувати ДПФ, але процес потребує багато обчислень. Для N точок даних треба провести множення з N хвиль різних частот, отримаємо N^2 комплексних множень. Якщо для 8 точок необхідно отримати помірно 64 результатів, то для більшого числа точок кількість результатів обчислень продовжує рости квадратичною залежністю. Історично цей головний недолік ДПФ став поштовхом для пошуку інших варіантів виконання цього завдання при меншій кількості необхідних розрахунків та відповідно часу. Для ДПФ при мільйоні точок необхідно було виконати трильйон множень, що для потужностей комп’ютерів 60-их років знадобилось би понад 3 років для одного такого перетворення, для порівняння новий запропонований метод пришвидшеної дії з набагато меншою кількістю обчислень можна було виконати це ж перетворення за 35 хвилин. Цей спосіб отримав назву швидке перетворення Фур’є [5].

1.3. Швидке перетворення Фур'є

ШПФ – алгоритм швидкого обчислення дискретного перетворення Фур'є, що часто використовується для обробки цифрових сигналів для перетворення часових даних дискретного формату у частотні дані. Для ДПФ при 8 точках необхідно було помножити кожну з них на 8 хвиль різної частоти, але синусоїди періодичні й хвилі різних частот будуть накладатися передбачуваним чином. Наприклад посередині сигналу усі непарні частоти матимуть одне і те ж саме значення (точно так же для усіх парних частот), тому посередині замість 8 разів можна помножити лише двічі. Така ситуація не унікальна лише для середини сигналу, це повторюється також для інших точок. Загалом для 8 точок ДПФ передбачає 64 обчислення, а ШПФ, враховуючи специфіку точок перетину синусоїд, потребує лише 24. Тобто нова методика надає значне скорочення кількості обрахунків: для дослідження сигналу при прямому обчисленні за допомогою ДПФ для N точок даних необхідно N^2 математичних операцій, а для ШПФ – $N*\log_2 N$ операцій. ШПФ застосовується для розкладання сигналу на набір гармонічних коливань, в результаті трансформації функція розкладається на осциляторні функції. Осциляторні (або гармонічні) функції – функції періодичних коливань, що показують залежність певної фізичної величини від часу та має синусоїдальний вигляд графіку [6].

При скороченні кількості множення постає питання які саме обчислення можна пропустити. Для прикладу і порівняння розглянуті знову 8 точок. Спочатку розглядаються окремо точки з парними та непарними номерами, усі вони множаться на кожну з 8 частот. Розглянувши лише парні точки й порівнявши перші 4 частоти з останніми 4 частотами (1-шу з 5-ю, 2-гу з 6-ю, 3-ю з 7-ю і 4-ту з 8-ю), на всіх цих точках значення синусоїд збігаються, те ж саме спостерігається з непарними індексами точок, але значення однієї синусоїди є протилежним до іншої, вони пов'язані

комплексним числом, тобто так нема потреби проводити обчислення для останніх 4 частот (рисунок 1.7). Якщо порахувати суми для парних і непарних низьких частот, ті ж значення можна використовувати для вищих частот, тому кількість обчислень скорочено вдвічі. Це перша ітерація, щоб дійти з N^2 до $N*\log_2 N$ повторюємо знову, розглядаючи окремо першу і другу половину значень, та знову ділимо точки даних на парні та непарні. Це повторюється доки не буде виділено окремо кожну точку даних, щоразу покладаючись на симетрію синусоїдальних функцій. Завдяки цьому скорочується кількість необхідних обчислень вдвічі, кількість таких етапів рівна $\log_2 N$, тому методика дозволяє дійти до $N*\log_2 N$ множень. Чим більше вхідних даних і відповідно чим більша роздільна здатність кожного стовпчика частоти (діапазон частот для кожного стовпчика спектра зменшується, тим самим краще відображає конкретну частоту сигналу), тим більше необхідно обрахунків, що збільшує навантаження на машину і потребує більше часу. Час виконання перетворення залежить також від потужностей обробки даних машини.

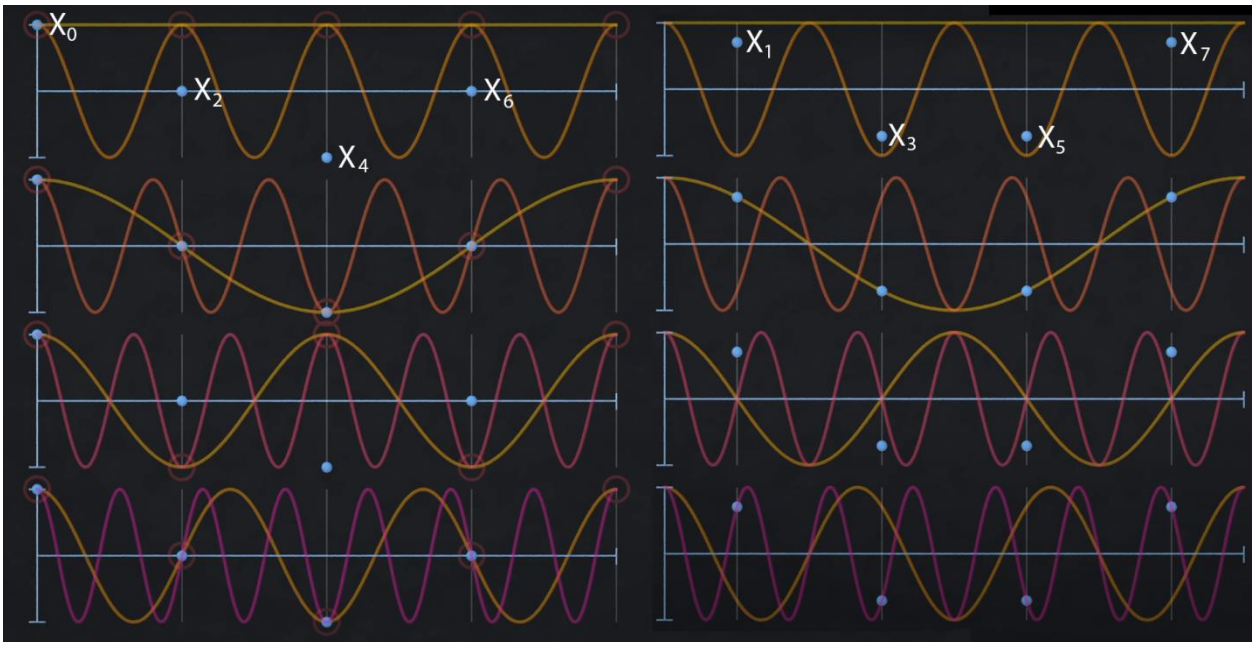


Рисунок 1.7. Процес ШПФ для 8 точок даних з 8 синусоїдами різної частоти.

Дана методика скорочень обрахувань називається “метелик”, алгоритм 24 комплексних множень представлений показаний на схемі (рисунок 1.8).

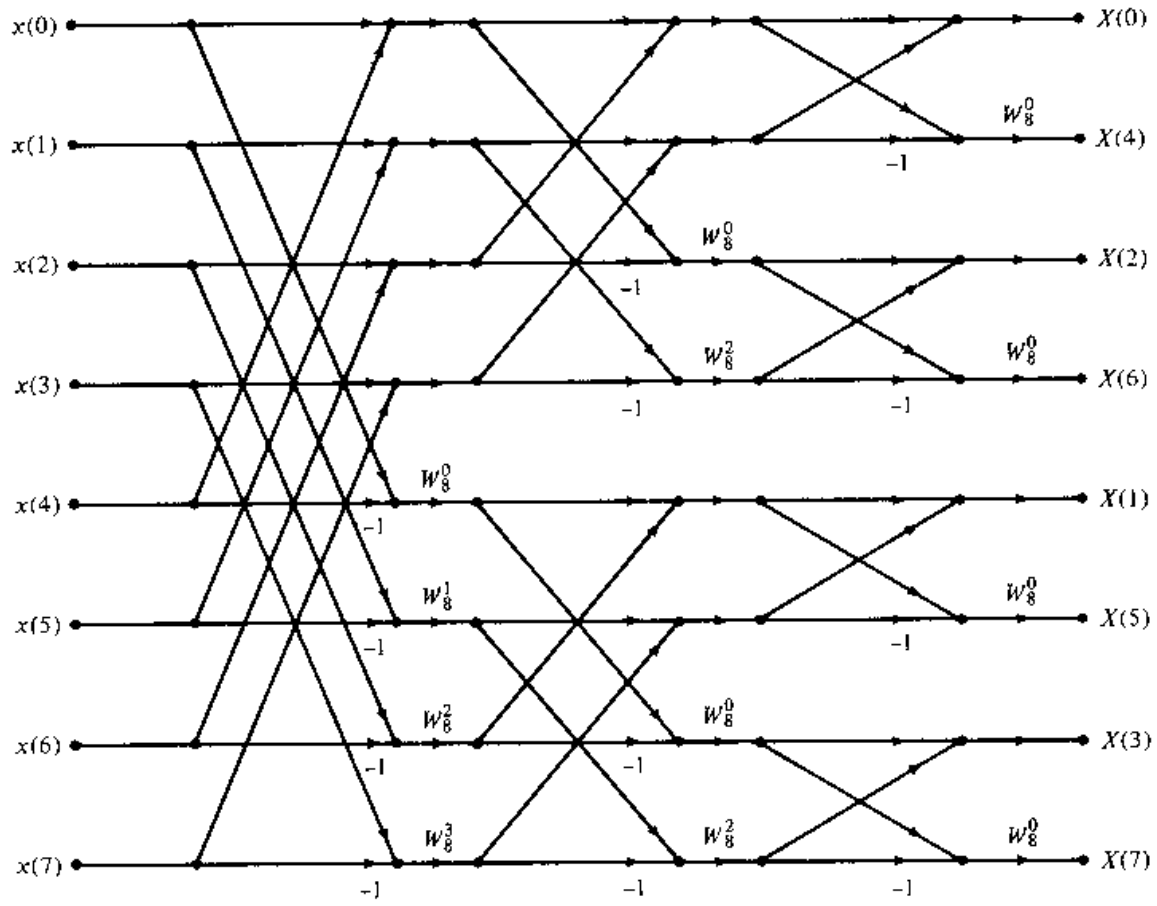


Рисунок 1.8. “Метелик” ШПФ, алгоритм декомпозиції сигналу з 8 точок.

Способів застосування ШПФ дуже багато: від розв’язання диференціальних рівнянь та вивчення структури кристалів до радарів, гідролокаторів, Wi-Fi й 5G скрізь, де потрібно обробляти сигнали з великою ймовірністю використовується цей алгоритм. З розвиненою мережею сейсмометрів, сучасними комп’ютерами та ШПФ можна фіксувати й розпізнавати підземні поштовхи з магнітудою 3 бали, які відповідають вибуху 1 кілотонни в практично будь-якій точці світу. ШПФ лягло в основу більшості алгоритмів стиснення. Для стиснення зображення картинка досліджується по пікселях, береться окремо кожний ряд пікселів і досліджується значення яскравості пікселя та проводиться ШПФ. Фактично визначаються які частоти присутні у значення яскравості у кожному рядку зображення. Яскравість відображає величину кожного компонента частоти, а колір – фазу. Потім ШПФ застосовується для стовпчиків пікселів (порядок не

має значення, можна спочатку стовпчики, а потім рядки). Для більшості зображень багато значень у перетворенні близькі до нуля, тому з перетвореного зображення можна викинути багато інформації, що скоротить розмір зображення, причому при зворотному перетворенні отримане зображення мало відрізняється від початкового. На комп'ютері більшість зображень зберігається у такому вигляді як двовимірне ШПФ, а при повторному перегляді комп'ютер їх інвертує. Зараз практично завжди використовують саме варіант ШПФ [7].

За теоремою відліків Найквіста спектр неперервного сигналу можна однозначно відновити без втрат за дискретними відліками з частотою дискретизації хоча б удвічі більшою за максимальну частоту сигналу. Чому саме удвічі, бо так на сигналі даної частоти вдається записати значення локального максимуму і мінімуму на кожному періоді, якщо сигнал буде мати частоту в 5 разів менше, то запис буде 5 значень на кожному періоді, що збільшить точність, тобто чим більша частота дискретизації за частоту сигналу, тим вища точність результатів, а коли значення ці частоти рівні (або й дискретизація повільніша), то записується 1 значення кожного періоду, що не дасть можливість відновити сигнал і будуть спотворення. Таким чином при необхідності якісного розглядання сигналів усього звукового діапазону на 20000 Гц потрібно мати частоту дискретизації як мінімум 40000 Гц. Звісно можливо розглядати й вищі частоти, які людина не чує, бо частоти звуку не обмежуються лише тими що ми чуємо, ті що ми не чуємо здатні також впливати на людський організм. Усі реальні сигнали скінченні, тому аналоговий сигнал можливо відновити з необхідною точністю за його дискретними відліками з частотою вдвічі більшою частотою обмеженого спектра реального сигналу та при перевищенні реальної частоти половини значенню дискретизації створюються спотворення в аналоговому сигналі, які неможливо уникнути при даних конфігураціях [8].

Труднощі роботи з ШПФ на мікроконтролерах:

- необхідна висока частота обробки (повинна бути вдвічі більшою досліджуваного звука, якщо необхідних весь діапазон звукових хвиль, що здатна чути людина, то необхідно 40 кГц);
- під час вибірки на мікроконтролері не повинні перемішуватися сторонні функції як Wi-Fi;
- розрахунки ШПФ значно навантажують центральний процесор, на відносно слабкому мікроконтролері з обмеженістю по пам’яті не вийде зробити ШПФ довгого сигналу, проте його можна розбити на менші блоки;
- групування по частоті треба налаштовувати вручну.

До недоліків методу ще можна віднести те що в результаті отримується частотно-амплітудна залежність без дослідження часу, що часто буває важливим параметром. Наприклад, якщо розглядати роботу світлофора за допомогою ШПФ, прийнявши колір світлофора як 3 різні частоти та час роботи кожного кольору як інтенсивність даної частоти, і слідкувати за коректністю роботи системи, то це не дасть повної правильної картини. Кінцевий спектр лише передає інтенсивність частот і не передає коли це відбулося, тобто після перетворення неможливо побачити спочатку світився секунду зелене світло, потім жовте і червоне, чи спочатку було секунду зелене світло, потім червоне і жовте, бо в обох випадках інтенсивність кожної частота не змінилась, змінився лише час, що не передається перетворенням. Проблему можна вирішити розглянувши сигнал не повністю відразу, а по частинах, це можна досягти за допомогою віконного перетворення Фур’є.

1.4. Віконне перетворення Фур’є

ПФ забезпечує представлення сигналу глобальними періодичними функціями, тобто функція розглядається одночасно цілком. Щоб дозволити локалізовані описи, потрібно зосередити аналіз Фур’є лише на частині сигналу. Це може бути досягнуто за допомогою віконної функції, g(t), яка

зберігає лише частину сигналу, що аналізується, і який потім може бути послідовно зміщений в інші місця, щоб охопити всю необхідну часову область. Віконні функції необхідні для кращої чіткості спектра при ПФ. Цей підхід відомий як віконне перетворення Фур'є (ВПФ) і був запропонований Габором, який прийняв для цієї мети функцію Гауса, враховуючи його оптимальні властивості з погляду локалізації, отриманої у часовій і в частотній областях. ВПФ – змінне в часі ПФ, яке необхідне, щоб визначити синусоїдні частоти й вміст фаз частини сигналу. Одночасна локалізація в часовій і частотній областях, обмежена нижньою межею, твердженням, яке зазвичай називають принципом невизначеності Гейзенберга. В цьому випадку така функція є добутком віконної функції та комплексної експоненти, тобто $g(t)e^{-i\omega t}$. ВПФ визначається рівнянням (4), з відповідною формулою реконструкції, заданою рівнянням (5), де можна оцінити його надлишковий характер, оскільки він використовує дві змінні для представлення функції однієї змінної [9].

$$F(\omega, b) = \int_{-\infty}^{\infty} f(t) g(t-b) e^{-i\omega t}\, dt \qquad (4)$$

$$f(t) = \frac{1}{||g||} \iint F(\omega, b) g(t-b) e^{-i\omega t} d\omega\, db \qquad (5)$$

Прямокутники в частотно-часовій площині (рисунок 1.9, а) вказують області, де ВПФ передає інформацію про сигнал. Ці області мають однакову форму при низьких і при високих частотних областях, що створює деякі проблеми при реалізації цієї методики. Масштаб гаусового вікна має бути визначений відповідно до основних характеристик очікуваного сигналу, потрібно визнати, що інші досить чіткі шаблони не будуть адекватно описані таким перетворенням, як для низьких, так і високих частот [10].

При необхідності часової компоненти існують компромісні методи, що дає і час, і частоту, одним з найпопулярніших і найефективніших методів аналізу даних є вейвлет-перетворення.

1.5. Вейвлет-перетворення

Суть застосуванню і принципу, грубо кажучи, досить схожий з ПФ, проте має істотні зміни. ПФ розбиває графік на нескінченні циклічні синусоїди, вейвлет – це сімейство скінченних функцій, на які можна розбити графік. Є можливість створення власного вейвлету. Для створення вейвлетів мають виконуватися 2 умови: значення рівне нулю (тобто сумарна площа на від'ємній ординаті графіку повинна дорівнювати сумарній площі на додатній ординаті) та скінченна енергія (площа квадрата функції має бути меншою за нескінченність). Вейвлети мають дійсну та уявну частину (також їх ще поділяють на материнську і батьківську частини) і вони є зручним інструментом в дослідженні частот нестаціонарного сигналу, оскільки вони гарно частотно-часово адаптовані. Математично вейвлет-перетворення представлені інтегралом (6)

$$W(a,b) = \int_{-\infty}^{\infty} f(t) a^{-\frac{1}{2}} \psi\left(\frac{t-b}{a}\right) dt \quad (6)$$

де a – параметр масштабу, b – параметр перетворення, f(t) – оригінальний сигнал.

На відміну від спектра ПФ, спектри ВП більш деталізовані. ВП розглядає часові функції в термінах коливань, локалізованими по часу й частоті. При аналізі сигналу використовується неперервне ВП. Основна ідея ВП – розклад сигналу на дві складові: грубу (що апроксимує) і точну (що деталізує). Комплексні вейвлети з дійсними та уявними частинами розширюють можливості візуалізації вейвлет-аналізу.Суть ВП втому, що перетворення повинно дозволити лише зміни в продовження часу, не форму, що залежить від вибору основних функцій. Принцип невизначеності обробки сигналів стверджує $\Delta t * \Delta\omega \geq 1/2$, t – час, ω – кутова частота, $\omega = 2\pi f$, f – частота. Вища детальність часу дає нижчу детальність частоти. Тобто при великому Δt дозвіл по часу буде низький, по частоті – високий, також це спричинить низьку частоту з великим коефіцієнтом масштабування. При малому Δt – навпаки: високий по часу, низький по частоті й висока частота з

малим коефіцієнтом масштабування. Також на величину впливає розширення вікна аналізу сигналу: метод ВП містить частотно-часове рухоме вікно, що налаштовано автоматично та якісно виявляє низькі й високі частоти сигналу при різних масштабах часу, фільтри, які допомагають для зменшення шумів і виділяють бажані компоненти сигналу. ВП використовують для стиснення даних (до найбільш розповсюджених форматів фотографій відносять JPEG 2000 та DjVu), електрокардіографічних сигналів, також має застосування для аналізу сигналів прискорень для швидкості, для розроблення електрокардіостимуляторів малої потужності тощо [11].

Послідовність як виділяти ВП компоненти сигналу:

1. Обрати значення максимальної кількості ітерацій (розкладення) сигналу i_{max} і початкове значення ітерацій $_{i=1}$.
2. Обрахувати початкове значення ентропії сигналу H_S за формулою (7), врахувати що $H_S=H_1$.

$$H(s) = \sum_i {s_i}^2 log_2({s_i}^2) \quad (7)$$

3. Обрати вейвлет-декомпозицію на рівні i=2.
4. Розбити сигнал s на глибину i, отримавши 1-шу компоненту W_i сигналу s.
5. Отримати 2-гу компоненту сигналу R_i віднявши від сигналу W_i.
6. Обчислити H_S (7) для обох компонентів сигналу H_{Wi} та H_{Ri}.
7. Обрахувати суму ентропії обох компонентів сигналу H_i за формулою (8).

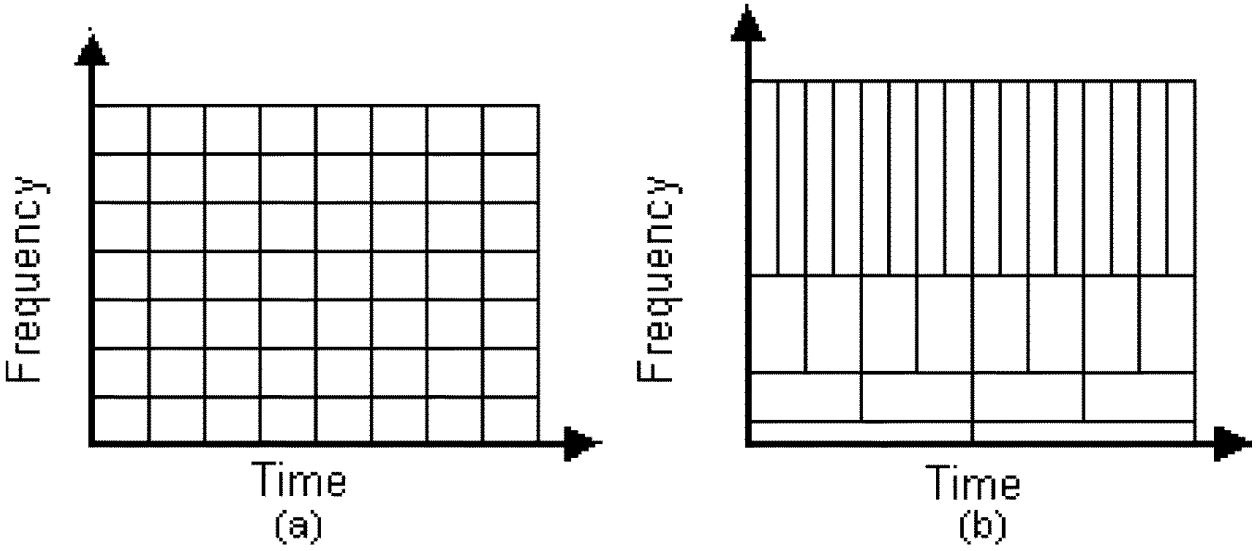
$H_i=H_{Wi}+H_{Ri}$ (8)

8. При $i \leq i_{max}$ зробити i=i+1 і повернутися на крок 4, повторювати поки i та i_{max} не стануть рівними [12].

1.6. Порівняння ШПФ, ВП та ВПФ

Порівняно з ВПФ, що має рівномірну сітку в частотно-часової області й погано адаптований для нерівномірних сигналів з різним масштабом, ВП має нерівномірну роздільну здатність, що надає можливість локального чи

повного дослідження сигналу (рисунок 1.9). ВП набув популярності для аналізу як стаціонарних, так і нестаціонарних сигналів. Частота і період обернено пропорційні, тому потрібно обрати більше вузьке вікно для локалізації високої частоти сигналу і ширше для низьких частот. Короткочасне перетворення Фур'є допустимо застосовувати для сигналу з порівняно вузькою смугою частот [13].



1.9. Різниця масштабу вікна ВПФ (a) та ВП (b).

В контексті визначення частоти та часу варто зазначити що цей спосіб жертвує точністю обох параметрів, щоб відображати обидва одночасно. Це не являється помилкою методу, бо фундаментально мати одночасно точні значення усіх цих параметрів неможливо.

Суть обох методів досить схожа – розбиття оригінального сигналу на складові (амплітуда, частота і час), проте кожний метод робить по-своєму, маючи власні переваги та недоліки порівняно з іншим. При виборі яким користуватися було розглянуто їх окремо та порівняно. ВП надає можливість виділити одночасно частотну і часову компоненти, проаналізувати зміну спектра частот в часі, що не може надати перетворення Фур’є. Обидва перетворення зворотні, тобто сигнал можна розбити на складові та скласти його назад маючи необхідні параметри. Отримавши запис звуку через ШПФ

його можна розбити як графік на складові синусоїди, що сумарно дають вихідну функцію (ВП своєю чергою розбиває на складові вейвлети).

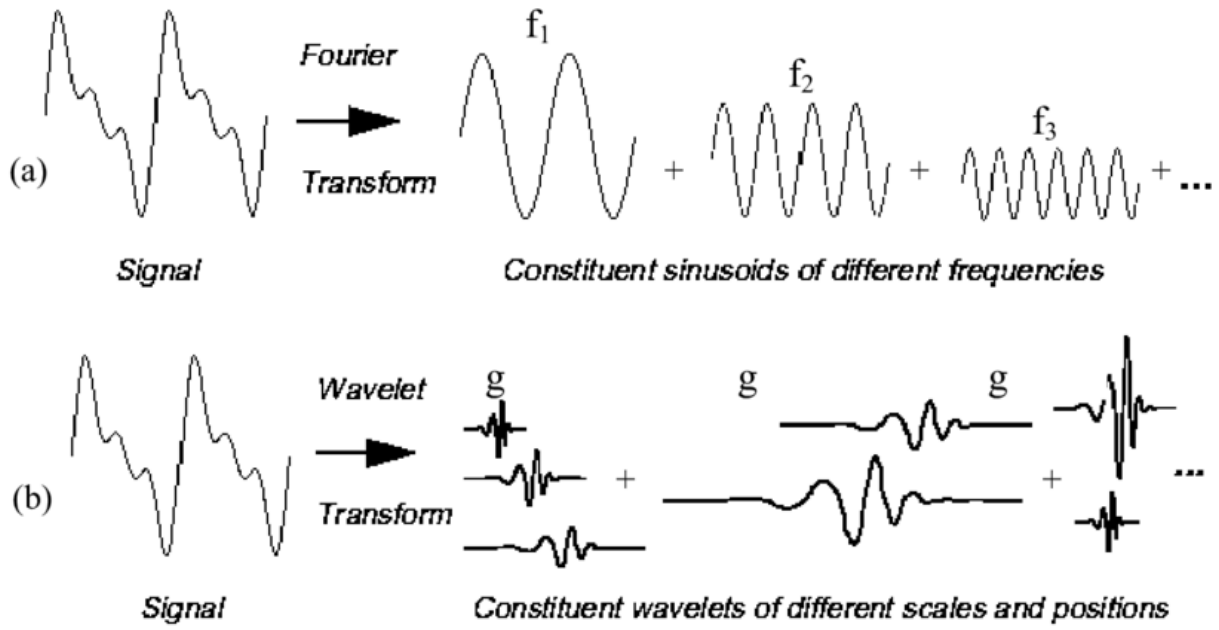


Рисунок 1.10. Розбиття сигналу через ПФ (а) і ВП (б).

Важливо зазначити що в результаті ШПФ ми отримуємо двовимірний графік з частотою та амплітудою без часу. Тому можливо спостерігати з якою інтенсивністю які були частоти, але неможливо ідентифікувати коли вони відбулися. Повна відсутність інформація про час створює проблеми для застосування перетворення Фур'є. Позбавившись виміру часу перетворення Фур'є знаходить точно частоту. Точність амплітуди та частоти – головна перевага методу, відсутність часової розмірності – головний недолік. Нескінченність на періодичність синусоїди для перетворення Фур'є може бути як позитивною стороною, так і негативною. З ШПФ можна отримати 2-вимірний графік, з ВП – 3. Фур'є показує лише точну частоту, проте на ньому відсутній час, вейвлет – наявні частота й час, але з гіршою точністю обох параметрів ближче до крайніх значень. Фур'є розкладає графік на нескінченні синусоїди, вейвлет – може розкласти на будь-які вейвлети. Фур'є застосовується для стаціонарних сигналів, ВП – для стаціонарних чи нестаціонарних сигналів. ВП більш комплексне для роботи та розуміння і

складніше для інтерпретації результатів, також граничні значення результатів дещо розмиті та не можуть дати точні правильні значення.

1.7. TinyML

Проблема влаштованих систем та машинного навчання полягає в розмірах та живленні, зменшення вбудованих пристроїв при низькому споживанні енергії поєднуючи з оптимізацією моделей ML для кращої ефективності використанням пам’яті – ці умови проклали шлях до створення TinyML, що націлений саме на цю потребу: впровадження моделей і методів машинного навчання на периферійних пристроях і фреймворку IP, надає можливість обробляти сигнали та дані на пристроях IP і надає вбудований штучний інтелект, тому не потрібно передавати дані на хмарні платформи для обробки. Впровадження технології призведе до покращення надійності конфіденційності та ефективності при цьому енергетичні затрати будуть понижені, також це може надати локальну аналітику при необхідності чи проблемам з підключенням. Цю технологію без перебільшення можна назвати проривом в області штучного інтелекту, так за прогнозами ABI Research до 2030 року вийдуть близько 2.5 мільярди пристроїв з TinyML, а за дослідженнями консалтингової компанії Silent Intelligence прогнозує що до 2026 року економічна оцінка ринку TinyML може перевищити понад 70 мільярдів доларів. Більшість IP пристрої виконують конкретну задачу, вони отримують дані через датчик, проводячи обрахунки, потім роблять щось залежно від отриманого значення або відправляють результат далі, зазвичай на централізований сервер, де потім реалізується машинне навчання. Натомість TinyML відкриває нові можливості для IP пристроїв як телевізори, автомобілі, кавомолки, годинники тощо, надаючи їм інтелектуальні функції, які нині широко розповсюджені й наявні виключно в комп’ютерах і смартфонах [14].

Одну з найпопулярніших застосувань технології – це розпізнавання мови. Люди здатні легко розпізнати вимовлені голосні та приголосні звуки. Однак цей сигнал не завжди однаковий для всіх, тому що різні люди мають різні акценти, коли говорять те саме речення вказує на те, як людина сприймає частоту звукових коливань, якщо вона вимовляється вищим або нижчим тоном. З іншого боку, гучність еквівалентна енергії звуку. Людські вуха більш чутливі до звуків нижчої тональності, тому часто для посилення високого звуку нам часто потрібен попередній фільтр. Можливо використати ПФ для аналізу спектра потужності звукової хвилі, щоб отримати більше інформації в частотній області. Одним з ефективних способів застосування ПФ до аудіосигналів є ШПФ, яке широко використовується для цифрового спектрального аналізу. У обробці мовлення це найважливіший алгоритм для отримання інформації. ШПФ є ітеративним і менш складним процесом із меншим часом обчислення, ніж прямий метод ДПФ [15].

TinyML – інструмент машинного навчання, що може проводити на пристрої аналітику, щоб сприймати різними способами як зір, звук та мова. TinyML являється об’єднанням програмних забезпечень, алгоритмів та апаратних забезпечень, вони взаємодіють один з одним і працюють разом для забезпечення необхідної швидкості та продуктивності роботи. Ця технологія є однією з найшвидшою по розвитку полем машинного навчання, вона все більше отримує практичного застосування, побутового використання, більшої інтеграції в системи. Ключова відмінність машинного навчання цього методу полягає в тому, що машинне навчання здійснюється не віддалено в великих дата центрах чи хмарах, а виконується прямо на пристрої, що надзвичайно покращує швидкість відгуку. Традиційний метод посилання даних не тільки більш енергетично затратний, а й розташовується віддалено від пристрою, що створює значну і різну затримку залежно від місця розташування необхідного дата центру, ця проблема стала однією з головних передумов цілей створення цієї технології. TinyML можна використовувати на низько енергійних пристроях як мікроконтролери, також

це надає автономності й енергетичної ефективності. Програми цього машинного навчання ставлять 2 умови для використання: можливість для масштабування мільярдів дешевих вбудованих систем та надання можливості збереження коду меншого кількох КБ на оперативній пам’яті приладу. Є спеціальні інструменти для запуску моделі машинного навчання на IP пристроях, найвідоміший з них – Tensorflow Lite, за допомогою якого можна групувати свої моделі Tensorflow для роботи з вбудованими системами, він надає невеликі двійкові файли, які здатні працювати з влаштованими системами при низькому живленні. Tensorflow відіграє вирішальну роль у розробці вбудованого машинного навчання. Основною частиною програми TinyML є API, створений на основі Tensorflow Lite/мікроядра. Ядро Tensorflow Lite використовується для розгортання моделей ML на мобільних і периферійних пристроях, а саме на смартфонах або на одноплатних комп’ютерах, таких як Raspberry Pi або Jetson Nano. Прикладом використання цієї технології машинного навчання є датчики стану навколишнього середовища, що здатні визначати температуру, наявність горючих речовин в лісі, здатні оцінювати ризики пожежі та при цьому довго працювати без заміни живлення. Рекомендованою мовою програмування для створення моделей ML є Python, проте для Tensorflow Lite це можна реалізувати також на C, C++ або Java, також Tensorflow Lite можна розгорнути моделі машинного навчання без необхідності підключення до Інтернету [16].

Програми для розпізнавання мови та звуку зазвичай мають такі методи зв’язку, які рахують всі дані необхідними й передають їх, але останніми роками з’явилася і набуває популярності як альтернатива семантична комунікація, яка передає лише контекст чи значення даних. Вона може бути здійснена за допомогою TinyML. Для подолання складнощів потреб потужностей та вимог до енергії на пристрої для розпізнавання слів, виявлення мови, спілкування створено бібліотеку TinySpeech. Вона надає можливість розробнику маючи низький обчислювальний ресурс створювати

архітектуру, що створює об’єкт зберігання малого об’єму використовуючи глибокі згорткові мережі. Широко відомі програми для розпізнавання мовлення, розповсюджені у побуті та знайомі багатьом голосові помічники як Siri, Google Assistant та Alexa, субтитри наживо, голосові команди, розумні будинки, медичні пристрої, моніторинг в дикій природі тощо, застосування обмежується лише фантазією. Реальне покращення обробки голосу від TinyML відразу на пристрої полягає в покращенні голосових помічників оскільки після отримання даних вони кожного разу перевіряють хмарну платформу чи базу даних що створює проблеми затримки відповіді, конфіденційність відправленої на базу даних і безпека даних, натомість запропонований інструмент не лише суттєво зменшує час відгуку, розпізнаючи мову прямо на самому пристрої, а й прибирає необхідність перенесення даних на хмарні платформи чи бази даних, що прибирає проблему конфіденційності (бо дані не відправляються на сервері та не зберігаються) та цілісності даних. Для використання TinyML з метою покращення мовлення, необхідно першим ділом переконатися що розмір моделі модернізації відповідала обмеженням апарату, тому застосували структуроване скорочення та ціле чисельне квантування моделі покращення мовлення PHM. В результаті вийшло зменшити розмір моделі практично в 12 разів, а кількість операцій – в 3 рази. Питання зменшення розмірів, кількості операцій та ефективність використання ресурсів і енергії досі відкрите й актуальне, над ним ще працюють та вдосконалюють технологію, так з’явилася ідея використати перетворювач на основі телефону системам розпізнавання голосу, замінивши предиктори ДКП на рівень Conv1D для зменшення необхідності обчислення на периферійних пристроях, що дало позитивні наслідки, CPM призвів до зменшення моделі, а декодування на основі ЗПКС надало кращу гнучкість для зміщення покращення моделі [17].

Периферійні обчислення – один з головних методів використання TinyML, бо поширення приладів IP для їх під’єднання по всій планеті необхідно налаштувати крайні прилади, що мінімізує навантаження хмарних

архітектур, не покладаючись на хмари ці прилади будуть мати центри обробки даних, що нададуть можливість автономного виконання обчислення на високому рівні самим приладам, тому збільшить хмарну незалежність, зменшить час відгуку, покращить безпеку та конфіденційність інформації та зменшить пропускну здатність. Використовуючи методи TinyML прилади Edge, стануть в пригоді для усунення поточних обмежень від вимог потужностей, пам’яті й обчислення (рисунок 1.11). Також технологія здатна підвищити якість застосування БПЛА, прибираючи недоліки нинішніх обмежень які притаманні цим апаратам. TinyML здатний бути контролером БПЛА що дозволить розробникам зробити енергетично ефективний прилад при цьому маючи низьку затримку і високу обчислювальну потужність. Найпопулярнішими готовими наборами навченими моделей ML від Tensorflow Lite є розпізнавання предметів (використовується для розпізнавання різних об’єктів на фотографії), розумні відповіді (генеруються відповіді при взаємодії з діалоговим штучним інтелектом чи чат-ботом) та рекомендації (пропонуються системі індивідуальні рекомендації на основі поведінки та вибору користувача). Як альтернативи існують CoreML для пристроїв машинного навчання на iOS від Apple та PyTouch Mobile – мобільна версія бібліотеки глибокого навчання PyTouch від Facebook [18].

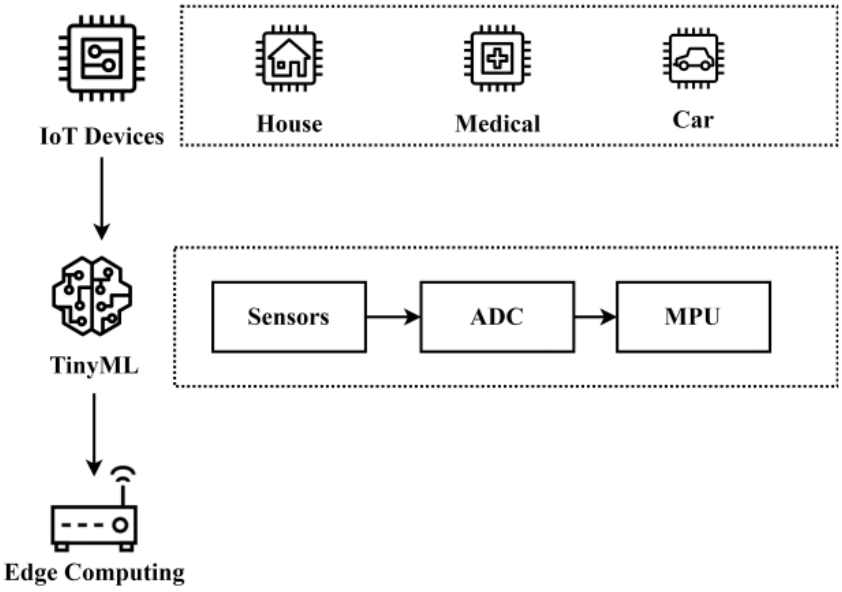


Рисунок 1.11. Пристрій Edge, що використовує алгоритми TinyML.

TinyML – актуальна і цікава тема індустрії ML та штучного інтелекту через потенційні використання, що обмежується лише уявою, до них можна віднести: програми на основі мови чи зору, діагностику і моніторинг стану здоров'я, дослідження змін в організмі при впливі довкілля, стиснення даних, периферійне обчислення, класифікацію шаблонів даних, інтерфейс керування мозком тощо. До переваг TinyML можна віднести:

- безпеку даних: вища гарантія конфіденційності даних, бо нема потреби передачі інформації назовні;
- економія енергії: передача інформації потребує значної серверної інфраструктури, оскільки нема потреби в передачі даних, то енергія та ресурси економляться і відповідно менші затрати;
- незалежність від з'єднання: на відміну від звичайного випадку коли пристрою потрібен Інтернет для відправлення даних на сервер, передача даних локально не потребує наявність Інтернету;
- мінімальність затримки: передача даних зазвичай потребує часу і часто створює затримку, проте коли передача даних не є частиною робочого процесу, то результат буде значно швидшим, практично без затримок [19].

У програмі розпізнавання мовлення, якщо необроблені зразки аудіо даних, які введені з мікрофона, передаються безпосередньо на модель, тоді сигнал міститиме забагато даних. Таким чином, для розрахунку будуть потрібні надлишкові ресурси. Крім того, якщо різні люди вимовляють те саме слово, вони отримають однаковий результат класифікації. Однак форма хвилі має інші форми порівняно з формою хвилі необроблених аудіоданих. Пристрої Edge використовують методи попередньої обробки, такі як ШПФ, щоб зменшити обсяг обчислень і підвищити ефективність класифікації моделі [20].

ШПФ – важливий метод обробки сигналів, який широко використовується багатьма способами, включаючи прискорення згорток, виділення ознак і регуляризацію моделей. Розподілене ШПФ пропонує спосіб обчислення перетворень Фур'є в моделях, які працюють із наборами

даних, схожими на зображення, які занадто великі, щоб поміститися в пам’ять одного прискорювача. Алгоритм розподіленого ШПФ реалізований для Tensorflow v1 як бібліотека. Ця робота представляє нещодавно додану власну підтримку в Tensorflow v2 для розподіленого ШПФ через новий API розповсюдження Tensorflow, DTensor, що є розширенням Tensorflow для синхронних розподілених обчислень. Він поширює програму та тензори через процедуру, що називається розширенням однієї програми, кількох даних (SPMD). DTensor пропонує єдиний API для традиційних шаблонів паралелізму даних і моделей, які широко використовуються в машинному навчанні.

Інтерфейс API для розподіленого ШПФ такий самий, як і оригінальний ШПФ у Tensorflow. Користувачам просто потрібно передати сегментований тензор як вхідні дані для наявних операцій ШПФ у TensorFlow, таких як tf.signal.fft2d. Вихід розподіленого ШПФ також стає сегментованим. Розподілене ШПФ може обробляти більше даних, ніж нерозподілене, використовуючи пам’ять на кількох пристроях. Компромісом є витрати додаткового часу на зв’язок і транспонування даних, що сповільнює швидкість обчислення. Реалізація розподіленого ШПФ у TensorFlow дотримується методу простого перемішування і локального ШПФ, який також використовується іншими популярними розподіленими бібліотеками ШПФ, такими як FFTW і PFFT. Примітно, що дві локальні операції ШПФ займають лише 3,6% загального часу (15 мс). Це приблизно 1/3 часу для нерозподіленого fft2d. Більшість обчислювального часу витрачається на перетасування даних, яке представлене операцією ncclAllToAll (рисунок 1.12) [21].

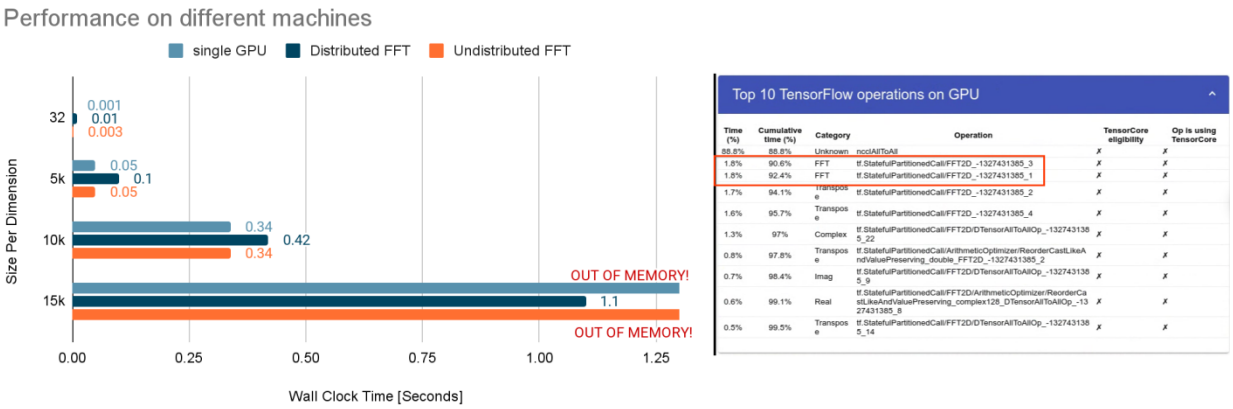


Рисунок 1.12. Продуктивність на різних машинах.

Розділ 2.

Практична частина

2.1. Аналіз апаратного забезпечення для спектрального аналізу

Таблиця 1. Компоненти схеми.

	Елемент	Модель	Розмір, мм	Ціна, грн	Частота, кГц
1	Мікроконтролер	DCCduino Nano CH340 TYPE C	18x43	165	16000
2	Мікроконтролер	RP2040w	51x21	356	133000
3	Мікрофонний модуль	MAX4466	20x14	65	6.8
4	Датчик звуку	LM393	40x17	22	10
5	Дисплей	OLED	0.96" 27x27	126	
6	Макетна плата	400 точок	55x80	47	
7	Кабелі	Папа-мама	200	45 (40 шт)	
8	Кабелі	Папа-папа	200	45 (40 шт)	

Спочатку був вибір між Arduino та Raspberry Pi, які схожі по загальному використанню, але їхнє порівняння схоже до порівняння калькулятора з комп’ютером. 2 моделі мікроконтролерів, з якими проводилась робота, були Arduino Nano (точніше DCCduino Nano CH340 TYPE C, який є аналогом цієї моделі, має майже ідентичні характеристики, дешевший, більш зручний для роботи порт і для нього підходять драйвери та код під Arduino Nano v3.0) та RP2040w (Raspberry Pi, 2040 значить: 2 ядра процесора, тип процесора M0+, 264 КБ оперативного SRAM пам’яті, 0 зовнішньої енергонезалежної пам’яті, Wi-Fi). Ардуїно краще за все підходить для контролю електронного компонента: мотору, світла, сенсорів тощо, гарно підходить для створення роботів і потребують небагато живлення. Натомість Raspberry містить мікропроцесор, якому потрібні чіпи для підтримки,

збереження та виконування коду, що можна порівняти з материнською платою комп'ютера з компонентами, які під'єднані до неї та необхідні для роботи комп'ютера, цей продукт сам по собі є комп'ютером, до нього можна під'єднати клавіатуру, монітор, мишку, живлення і працювати як на звичайному комп'ютері на операційній системі Linux. Raspberry Pi має більші потужності, але тому потребує більше енергії та є більш комплексним, тому складнішою моделлю для керування. Це не значить що Ардуїно не може виконувати основні функції Paspberry Pi та навпаки, проте потужностей першого не достатньо для повної реалізації потенціалу другого, RP2040w своєю чергою може повністю виконувати всі функції Arduino Nano, проте для характерної задачі Ардуїно Raspberry Pi не зможе продемонструвати весь свій потенціал (також ця модель дорожча та складніша в роботі з базовими елементами). Raspberry Pi 2040 підтримує програмування на assembly, C, C++, Free Pascal, Rust, Go, MicroPython, CircuitPython, Ada і TypeScript, Arduino Nano використовує варіант мови C/C++ (код написаний цією мовою і має додаткові методи та функції в ArduinoIDE). Також є можливість запустити python на Arduino. Для роботи з сенсорами та роботизованими елементами краще підходить Ардуїно, якщо необхідно додати елементи комп'ютера до приладу як вебкамеру, зробити сервер тощо, то краще підходить Raspberry. Як приклад можна сказати що для машини на радіоуправлінні краще взяти Ардуїно, а для машини на радіоуправлінні з камерою, що веде пряму трансляцію – Raspberry Pi.

Для акустичного спектроскопа мікрофон підбирається індивідуально для потреб поставленої задачі, вибирають по чутливості сенсора, ширини діапазону сприйнятливих частот, надійності, ціни тощо. В даній роботі було використано декілька звичайних цифрових мікрофонів. Готові подібні прилади (навіть низької якості) коштують по завищеній ціні, ніж можна зібрати вручну: загальна сума всіх використаних компонент не перевищує 500 грн. Як мікрофон був вибраний цифровий датчик звуку з компаратором LM393. Обрано роботу з ним, замість звичайного мікрофона, оскільки на

ньому ще є потенціометр, за допомогою якого було налаштовано оптимальну чутливість. Було обрано модель через низьку ціну і гарні відгуки. Робоча напруга 3.3-5 В і легко підключається до Arduino. По характеристиках робоча частота моделі 50-10000 Гц, тобто діапазон звуків, що може сприймати сенсор вдвічі менший – до 5000 Гц. Хоч людина і сприймає частоти умовно до 20000 Гц, але вище 5000 Гц йдуть дуже високі неприємні звуки, а також великий діапазон звуків лежить до межі в 5000 Гц. Сенсор має дискретний вихід, рівень гучності можна налаштувати потенціометром вручну. Також проведено порівняльні експерименти замінивши на мікрофонний модуль MAX4466 зі схожими результатами, але з параметрів якого звук був в меншому діапазоні (до 3400 Гц порівняно з попередніми 5000 Гц).

2.2. Вибір мови програмування

Працюючи з Arduino було використано звичний для роботи з цим мікроконтролером Arduino IDE. Середовище ArduinoIDE було обране як найбільш розповсюджена стандартна мова програмування Arduino, це інтегроване середовище розробки для Windows, macOS і Linux, воно розроблене на C та C++ та потрібне для програмування на Arduino чи Arduino-сумісних платах. Ці мови не єдині, є ще можливості програмувати на python, blockly тощо через треті програми. Програма комфортна в роботі, наявна велика кількість різноманітних корисних бібліотек, в роботі було використано бібліотеки fix_fft для виконання ШПФ, ППІ для загальної роботи з мікроконтролером, Wire.h, Adafruit_GFX.h і Adafruit_SSD1306.h для OLED дисплею і графіків на ньому на ATmega328P. Для роботи DCCduino Nano CH340 було встановлено в ArduinoIDE драйвер для підтримки плати Arduino Nano, оскільки мікроконтролер є аналогом цієї моделі та він підтримував цю версію. Повна версія коду представлена нижче [Додаток 2]. Щоб позбутись шумів уловлюються звуки від частоти 60 Гц і застосовано зниження шуму.

2.3. Опис експериментальної установки й застосування технології

Розроблено портативний акустичний спектрометр для отримання спектра звуку, що дозволяє проводити аналіз спектра частот діапазоном близько 60-2300 Гц та бачити відносну інтенсивність кожного стовпчика частоти. В результаті портативний акустичний спектрометр [Додаток 3]. Принципова і монтажна схема представлена нижче (рисунок 2.1, 2.2). Чутливість залишає бажати кращого, але це і не дивно, оскільки мікрофон дешевий і необхідний лише для наочності, при заміні його на якісніший при потребі можна досягти більшого діапазону і кращої чутливості. Прилад

працює стабільно безперебійно, суттєвих порушень в роботі чи значущих похибок не виявлено.

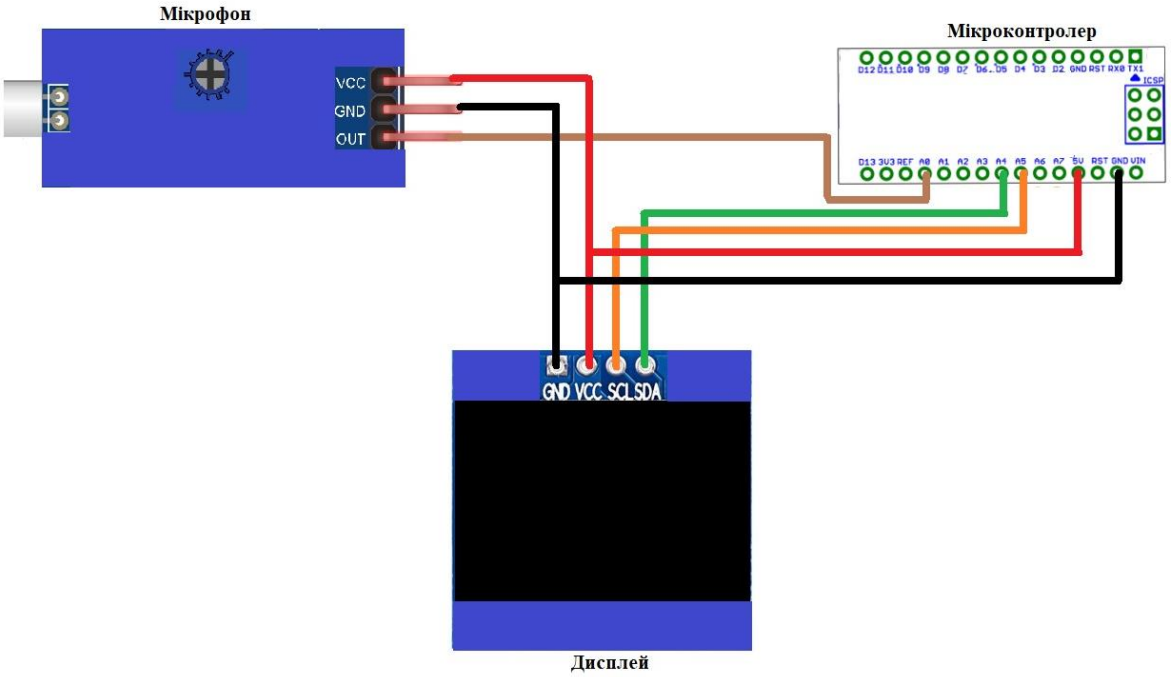


Рисунок 2.1. Принципова схема.

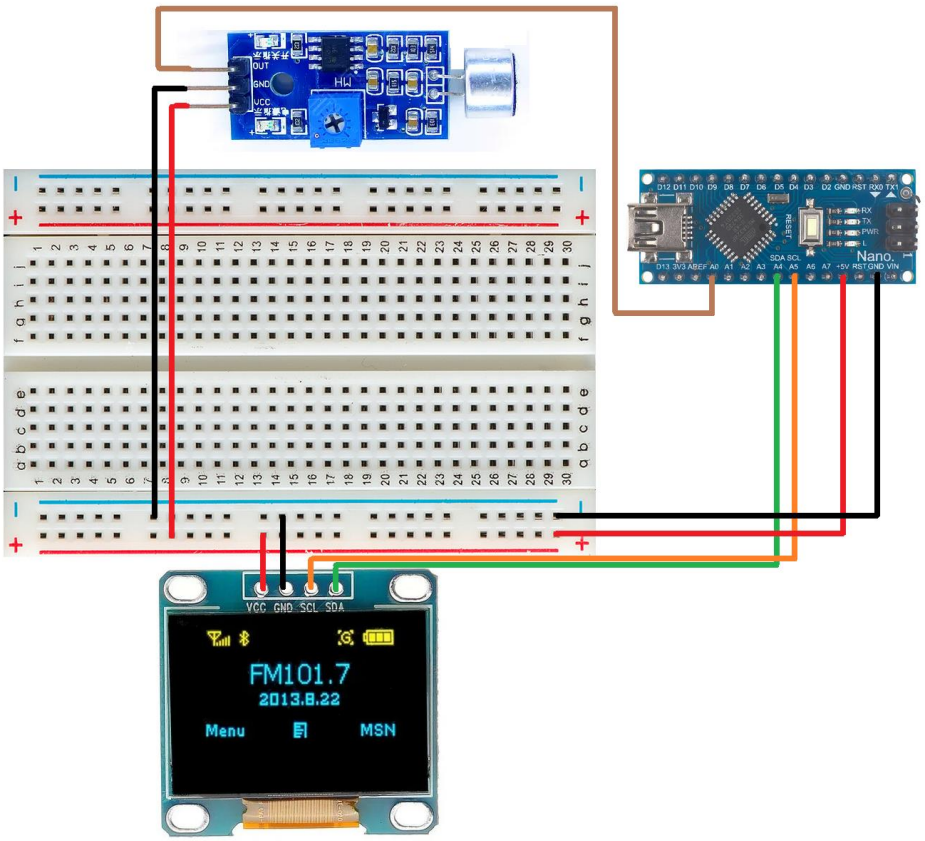


Рисунок 2.2. Монтажна схема.

Маючи частоту дискретизації та деяку кількість стовпчиків можна знайти частоту кожного зразка, для цього необхідно поділити частоту дискретизації на кількість стовпчиків (нульовий бін не враховується, спостерігаємо лише за половиною зразків, бо одна половина має додатні значення, а інша – від'ємні, що не цікавлять). Наприклад, для пристрою з частотою дискретизації 40000 Гц та 16 зразками частота кожного зразка буде 40000/16 = 2500 Гц, тоді діапазон частоти бінів від 0-го до 1-го буде 0-2.5 кГц, від 1-го до 2-го 2.5-5 кГц тощо. Це міняється коли розглядати сигнал не лінійно, а логарифмічно, оскільки найцікавіше відбувається на нижчих частотах і менше цікавить діапазон як 17500-20000 Гц (який багато хто навіть і не чує). Логарифмічну зручнішу шкалу можна розрахувати для персональних потреб, так якщо поставити мінімум 80 Гц, максимум 12000 Гц і розподілити це між 8 бінами, то одержимо такий результат: 0 бін – 80 Гц, 1 – 164 Гц, 2 – 335 Гц, 3 – 685 Гц, 4 – 1401 Гц, 5 – 2867 Гц, 6 – 5866 Гц, 7 – 12000 Гц. Більша кількість бінів дасть більшу наочність і точність, якщо збільшити з 8 до 16 отримаємо такі результати: 0 бін – 80 Гц, 1 – 112 Гц, 2 – 156 Гц, 3 – 218 Гц, 4 – 304 Гц, 5 – 425 Гц, 6 – 594 Гц, 7 – 829 Гц, 8 – 1158 Гц, 9 – 1617 Гц, 10 – 2258 Гц, 11 – 3154 Гц, 12 – 4405 Гц, 13 – 6152 Гц, 14 – 8592 Гц, 15 – 12000 Гц. Природно логарифмічний відлік зручніший, оскільки людське вухо також чує не лінійно і людині легше відрізнити 300 та 400 Гц ніж 12100 і 12200 Гц. Для відображення частоти зазвичай використовують логарифмічну шкалу і для дослідження широкого діапазону звуку, проте в даній роботі було використано лінійну шкалу, оскільки діапазон не дуже великий, не всі сприйнятливі для людини 20 кГц.

Більше кількість зразків створює вищу точність визначення частоти звука. Тип шкали можна змінити по потребі, ширшому діапазоні тощо. На екрані видаються дискретні значення, звісно стовпчиків не 2240 на кожний Гц окремо, було взято 32 по 70 Гц на кожний. Коли звук стає на межі частот стовпчиків помітно як затухає один і починає підійматися інший, це показує кращу плавність переходу частот, проте це робиться шляхом зменшення

точності інтенсивності, оскільки стовпчик опускається при незмінній гучності звуку, проте зміна висоти вказує на зворотне. Для роботи, де головна увага зосереджується на частотах, це більше перевага ніж недолік, оскільки це дозволяє побачити наближення частоти до верхнього чи нижнього значення порога стовпчика.

Де тільки не можна застосувати знання звукових характеристик та хвиль загалом: музика, слухові апарати, голосові команди, комунікація, керування голосом, розпізнавання мови та емоцій, відтворення мови людини машиною та надання їй більш живого звучання досліджуючи як змінюється вимова людини при різному настрої, розпізнавання і відтворення акцентів, моніторингу навколишнього середовища, навіть в оборонному секторі є застосування тощо.

2.4. Можливості покращення приладу

Можливостей для покращення приладу безліч і обмежується лише фантазією. З покращень функціонала приладу найпростіше й очевидна зміна це новий мікрофон. Мікрофони є вартістю по декілька тисяч і дорожчі, також можливо покращити сам сенсор, наприклад, установивши сферичне дзеркало і сам мікрофон у фокусі для кращого фокусування звуку і збору інформації. Можна придбати кращий екран, на якому можна відобразити більше інформації. Можливо перейти на кращий потужніший мікроконтролер, вважаю Raspberry Pi підійде краще також для подальших модернізацій і нових можливостей приладу. Зміна мови коду на мову з ширшим функціоналом та кращими прикладними бібліотеками здатна надати більші можливості функціонала приладу й покращити його. Існує варіант змінити технологію підходу обробки даних і, наприклад, перейти з ШПФ на ВП, що дадуть всі необхідні характеристики для повноцінного аналізу. Можливо додати функцію не тільки аналізу, а й запису та порівняння результатів, можна створити власну бібліотеку бажаних звуків для детектування і

порівнювати навколишні звуки з заданими. Можливо покращити метод зменшення шуму в аудіо файлі й отримувати якісніші та чіткіші після обробки. Застосування TinyML значно покращить обрахункові можливості та сполучення мікроконтролера з сенсорами та середовищем загалом, може зменшити витрати енергії, покращити надійність тощо, ця технологія буде не лише покращенням, а наступним кроком еволюції акустичного спектрометра.

Висновки

ШПФ та ВП є корисними методами для роботи з аналізом частотного спектра сигналу, кожний з яких має свої переваги та недоліки, які були наведені в роботі. Практичну частину було реалізовано на DCCduino Nano CH340, код написано з допомогою бібліотеки ШПФ. Після теоретичного і практичного порівняння датчика звуку з компаратором LM393 отримано кращі результати ніж з мікрофонним модулем MAX4466 (причому він ще й значно дорожчий). Чутливість обох є досить низькою (необхідно тримати близько до джерела звуку), тому логічним наступним покращенням може бути заміна мікрофона на якісніший (це також може збільшити діапазон досліджуваних частот та точність результатів). Також можливо додати сферичне дзеркало з мікрофоном у фокусі, це зробить прослуховування звуку більш спрямованим: покращить звук, що збирає в фокусі, проте інші звуки буде чутно гірше.

Для загального дослідження і для певних поставлених цілей ШПФ достатньо, проте для глибшого аналізу є можливості покращити установку або змінити підхід роботи, до того ж значення частот і амплітуд точніший для цього методу, отримані результати досить добре передають точність частоти з похибкою (зазвичай) до 35 Гц. В побуті можливо застосувати для порівняння частот звука блискавок та вибухів, щоб ідентифікувати джерело звуку. Можливо дослідження буде продовжено в майбутньому і розвинуто можливості портативного акустичного спектрометра іншими запропонованими шляхами покращення системи. Як один з першочергових майбутніх кроків стоїть реалізація обох методів для практичного порівняння результатів, що заодно надасть можливості кращого порівняння точність теоретичних передбачень і мати що необхідно з результатів.

Список використаних джерел

1. Audio Archiving Guide: Part 1 – М. Серано – 2008
2. Перетворення Фур'є, Лапласа: узагальнення та застосування: навчально-методичний посібник – Г. П. Лопушанська, А. О. Лопушанський, О. М. М'яус – 2014
3. Функції комплексної змінної. Перетворення Фур'є та Лапласа // Навчальний посібник для студентів технічного спеціального вищого закладу освіти – Я. І. Дасюк, В. С. Ільків, П. І. Каленюк, П. П. Костробій, З. М. Нітребич – 1999
4. Generalized Discrete Fourier Transform With Nonlinear Phase – А. Аканцу – 2010
5. Uniqueness of the discrete Fourier transform – І. Баракін, Н. Ратіер – 2023
6. The Fast Fourier Transform – І. Брігам – 2002
7. A modified split-radix FFT with fewer arithmetic operations – Матео, Фріго, Джонсон – 2007
8. Certain topics in telegraph transmission theory, Translation AIEE – Г. Найквіст – 1928
9. Short Time Spectral Analysis, Synthesis, and Modification by Discrete Fourier Transform – Б. Ален – 1977
10. Comprehensive Chemometrics - С. Браун, Р. Таулер, Б. Вальчак - 2009
11. Вейвлет-перетворення у компресії та попередній обробці зображень – О. В. Капшій, О. І. Коваль, Б. П. Русин – НАН України, Фізико-механічний інститут імені Г. В. Карпенка. – 2008
12. Дослідження зміни ентропії і енергії на етапах декомпозиції сигналу // Радіоелектроніка, інформатика, управління. – В. І. Дубровін, Ю. В. Твердохлєб – 2013
13. Wavelet-Based Identification of Delamination Defect in CMP (Cu-Low k) Using Nonstationary Acoustic Emission Signal – Р. Ганесан, К. Тапас, К. Дас, К. Аран, А. Кумар – 2003

14. TinyML: програми, обмеження та його використання в пристроях IoT і Edge – К. Кейрівал – 2023
15. Implementation of a Speech-command-interfaceon Microcontroller with TinyML – Д. А. Фам – 2021
16. What is TinyML, and why does it matter? – Ж. Рібейро – 2020
17. TinyML : Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers – П. Уорден, Д. Сітунаяке – 2019
18. TinyML Made Easy: Sound Classification (KWS) – М. Роваі – 2022
19. Why the Future of Machine Learning is Tiny – П. Уорден – 2018
20. Hardware/Software Co-Design for TinyML Voice-Recognition Application on Resource Frugal Edge Devices – Дж. Квон – 2021
21. Distributed Fast Fourier Transform in TensorFlow – Р. Сан – 2023

Додаток 1.

Система акустичних сенсорів Zvook

Додаток 2.

Код

```
#include "fix_fft.h"
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define OLED_RESET 4
Adafruit_SSD1306 display(OLED_RESET);
char im[128], data[128];
char x = 0, ylim = 60;
int i = 0, val;
void setup()
{
  Serial.begin(9600);
  display.begin(SSD1306_SWITCHCAPVCC);
  display.setTextSize(1);
  display.setTextColor(WHITE);
  display.clearDisplay();
  analogReference(DEFAULT);
};
void loop()
{
  int min=1024, max=0; // 1024 впливає на ширину стовпчика, (νmax - νmin)/1024
  for (i = 0; i < 128; i++) {
    val = analogRead(A0);
    data[i] = val / 4 - 128;
    im[i] = 0;
    if(val>max) max=val;
    if(val<min) min=val;
```

```
};
  fix_fft(data, im, 7, 0);
  display.clearDisplay();
  for (i = 1; i < 64; i++) {
    int dat = sqrt(data[i] * data[i] + im[i] * im[i]);
    display.drawLine(i*2 + x, ylim, i*2 + x, ylim - dat, WHITE);
};
  display.drawLine(0, 10, data[0], 10, WHITE);
  display.drawLine(0, 11, data[0], 11, WHITE);
  display.setCursor(0,0);
  display.print("min=");
  display.print(min,1);
  display.setCursor(8*6,0);
  display.print("max=");
  display.print(max);
  display.setCursor(16*6,0);
  display.print("v=");
  display.print((int)data[0]);
  display.display();
};
```

Додаток 3.

Готовий прилад