

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)
(рівень вищої освіти)

Методи аналізу ефективності перетворення Фур'є та Хартлі
для спектрального аналізу звукових сигналів

(тема)

Виконав: студент 2 курсу, групи СКСм-21-1

Пристинський Д.В.

(прізвище, ініціали)

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані

комп'ютерні системи


(повна назва освітньої програми)

Керівник роботи доцент Шкіль О. С.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри



(підпис)

Чумаченко С.В.
(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерної інженерії та управління _____
Кафедра _____ Автоматизації проектування обчислювальної техніки _____
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність _____ 123 Комп'ютерна інженерія _____
(шифр і назва)
Тип програми _____ Освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)
Освітня програма _____ Спеціалізовані комп'ютерні системи _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав.кафедри _____
(підпис)
“ _____ ” _____ 20 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Пристинському Данилу Валерійовичу _____
(прізвище, ім'я, по батькові)

- Тема роботи (проекту) Методи аналізу ефективності перетворення Фур'є та Хартлі для спектрального аналізу звукових сигналів
затверджена наказом по університету від "13" _____ 05 _____ 2022 р. № 607 Ст
- Термін подання студентом роботи до екзаменаційної комісії _____ 15.12.2022
- Вихідні дані до роботи (проекту) _____
Перетворення Фур'є та швидке перетворення Фур'є _____
Перетворення Хартлі та швидке перетворення Хартлі _____
Розкладання звуку до спектру _____
Аналіз звукового сигналу _____
Мова програмування C++ _____
Середовище розробки QtCreator _____
- Перелік питань, що потрібно опрацювати у роботі _____
Особливості роботи зі звуком _____
Порівняння перетворення Фур'є та Хартлі _____
Ефективність перетворень для дискретизованого звукового сигналу _____
Вибір платформи та інструменті для реалізації середи для аналізу _____
Проектування середи для аналізу ефективності перетворень звуку _____

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 15 слайдів

6. Консультанти розділів роботи (проекту)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

7. Дата видачі завдання 27.04.2022

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи (проекту)	Термін виконання етапів проекту (роботи)	Примітка
1	Видача теми проекту, узгодження і затвердження	01.09.2022-01.05.2022	
2	Аналіз особливостей роботи зі звуком	01.05.2022 -10.06.2022	
3	Порівняння перетворень Фур'є та Хартлі	10.06.2022 -17.07.2022	
4	Аналіз ефективності перетворень для дискретизованого звукового сигналу	18.07.2022 -25.08.2022	
5	Вибір платформи та інструменті для реалізації середи для аналізу	25.08.2022 -30.09.2022	
6	Проектування середи для аналізу ефективності перетворень звуку	31.09.2022 -05.10.2022	
7	Оформлення пояснювальної записки	05.10.2022 -07.11.2022	
8	Перевірка виконаного проекту керівником	08.11.2022 -10.12.2022	
9	Захист проекту	15.12.2022 -22.12.2022	

Студент _____

(підпис)

Керівник роботи (проекту) _____

(підпис)

доцент Шкіль О.С.

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 61 сторінку, 15 рисунків, 8 джерел за переліком посилань.

ПФ, ШПФ, ДПФ, ПХ, ДПХ, ШПХ, C++, Qt, СПЕКТР ЗВУКУ, СПЕКТРОГРАМА, АНАЛІЗ ЗВУКУ

Розглянуті особливості роботи зі звуком, методи аналізу звуку. Були розглянуті основні методи розкладання звуку до спектру та проаналізовані такі перетворення: перетворення Фур'є, дискретне перетворення Фур'є, перетворення Хартлі, дискретне перетворення Хартлі, та алгоритм швидкого перетворення Фур'є, алгоритм швидкого перетворення Хартлі та їх порівняння. Було розроблене середовище для аналізу ефективності швидких перетворень у контексті роботи зі звуком в одному просторі. Середовище аналізу було представлено як прикладна програма.

ABSTRACT

Thesis contains 61 pages, 15 figures, 8 sources according to the list of links.

FT, FFT, DFT, HT, DHT, FHT, C++, Qt, SPECTRUM, SPECROGRAM,
SOUND ANALYSIS

Features of working with sound, methods of sound analysis are considered. The main methods of decomposition of sound into a spectrum were considered and the following transformations were analyzed: Fourier transform, discrete Fourier transform, Hartley transform, discrete Hartley transform, and fast Fourier transform algorithm, fast Hartley transform algorithm and their comparison. An environment was developed to analyze the effectiveness of fast transformations in the context of working with sound in one space. The analysis environment was presented as an application program.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП.....	8
1.1 Звук, як явище.....	9
1.2 Класифікація звуків	11
1.3 Характеристики звуку	12
1.4 Фізичні параметри звуку	13
1.5 Реєстрація, запис та відтворення звуку	17
2 ПЕРЕТВОРЕННЯ ФУР'Є ТА ХАРТЛІ.....	18
2.1 Перетворення Фур'є	18
2.2 Дискретне перетворення Фур'є	20
2.3 Перетворення Хартлі та дискретне перетворення Хартлі	22
2.4 Порівняння перетворень Фур'є та Хартлі	24
2.5 Швидке перетворення Фур'є	26
2.6 Швидке перетворення Хартлі	29
2.7 Порівняння між ШПФ та ШПХ	32
3 РОЗРОБКА СЕРЕДИ ДЛЯ АНАЛІЗУ ЕФЕКТИВНОСТІ ШПХ ТА ШПФ ДЛЯ РОБОТИ ЗІ ЗВУКОМ.....	34
3.1 Вибір платформи та вимоги	34
3.2 Вибір засобів розробки.....	34
3.2.1 Мова програмування C++	34
3.2.2 Фреймворк Qt.....	36
3.2.3 Бібліотека FFmpeg	37
3.2.4 Бібліотека FFTW	37
3.3 Особливості роботи перетвореннями із бібліотеки FFTW.....	39
3.3.1 Одновимірні ШПФ реальних даних	39
3.5.2 ШПФ у напівкомплексному форматі	41
4 РОЗРОБКА ПРИКЛАДНОЇ ПРОГРАМИ	44
4.1 Інтерфейс програми.....	44
4.2 Перетворення у прикладні програмі.....	46
4.2.1 Файл transformation.cpp.....	46

4.2.2 Файл dfttest.cpp	50
5 ОГЛЯД ПІДСУМКОВОЇ ПРИКЛАДНОЇ ПРОГРАМИ ДЛЯ АНАЛІЗУ ЕФЕКТИВНОСТІ ПЕРЕТВОРЕНЬ ШПФ ТА ШПХ ДЛЯ РОБОТИ ЗІ ЗВУКОМ.....	53
5.1 Огляд прикладної програми та її функціоналу	53
5.1.1 Файловий режим	53
5.1.2 Audio Input Mode	58
ВИСНОВКИ	60
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	61
ДОДАТОК А	Ошибка! Закладка не определена.
ДОДАТОК Б	Ошибка! Закладка не определена.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ

ПФ – Перетворення Фур'є

ДПФ – Дискретне перетворення Фур'є

ШПФ – Швидке перетворення Фур'є

ПХ – Перетворення Хартлі

ДПХ – Дискретне перетворення Хартлі

ШПХ – Швидке перетворення Хартлі

ВСТУП

Аналіз звуку є комплексною задачею і перетворення Фур'є є ключовим елементом для розкладання сигналу до частотного простору. Але часто працюючи зі звуком, мається на увазі вже дискретизований звуковий сигнал, це означає, що приходиться працювати з послідовністю дійсних чисел.

В свою чергу перетворення Фур'є передбачає роботу з комплексними числами, а комплексний частина кінцевої послідовності часто відкидається і більша частина роботи проводиться з дійсною частиною кінцевої послідовності.

Також існує перетворення Хартлі, яке має тісний зв'язок з перетворенням Фур'є, але його головна ідея перетворювати дійсні послідовності до дійсних послідовностей, а також просте зворотне перетворення.

Була висунута гіпотеза, що для реальних послідовностей в одному вимірі швидке перетворення Хартлі буде ефективніше ніж швидке перетворення Фур'є. Для перевірки цієї гіпотези потрібне середовище для аналізу ефективності перетворень для різних задач. Наприклад обробка звуку в реальному часі, обробка великих музичних файлів, робота с шумом на записах.

У цій роботі буде розроблене середовище для аналізу ефективності для ШПФ та ШПХ, будуть проаналізовані перетворення з математичної точки зору в контексті роботи саме з дійсними послідовностями в контексті розкладання звукового сигналу до частотного простору.

АНАЛІЗ ОСОБЛИВОСТЕЙ РОБОТИ ЗІ ЗВУКОМ

1.1 Звук, як явище

Звук – коливальний рух частинок середовища, що поширюється у вигляді хвиль у газі, рідині чи твердому тілі та сприймається слухом. У вузькому значенні терміном звук визначають коливання, які сприймаються сенсорною системою тварин і людини. В цьому випадку маються на увазі збурення, що поширюються в повітрі. Органи слуху людини сприймають звуки у відносно вузькому частотному діапазоні. Слуховий апарат багатьох тварин сприймає звуки в значно ширшому інтервалі частот. В загальному випадку цим терміном визначається процес поширення збурень в різних за фізичними властивостями середовищах, в яких відновлювальною силою, що намагається повернути збурену частинку в положення рівноваги, є сила пружності. Хвильові збурення, що визначаються як звук, є об'єктивною реальністю і існують незалежно від сприйняття їх будь-якими живими істотами. Вивчення закономірностей генерації, поширення та сприйняття звуків в різних середовищах є приналежним до такої наукової дисципліни як акустика.

Більшість явищ у природі супроводжуються характерними звуками, які сприймаються та розпізнаються вухом людини і тварин і служать для орієнтування та спілкування. Специфіка сприйняття коливальних рухів частинок повітря слуховим апаратом людини зумовлює поділ звуків на приємні, гармонійні (звуки мови, музичні звуки, спів пташок, наприклад) та звуки зі специфічним спектральним наповненням, часто небажані та дратівні, які визначаються як шум.

Зміщення в просторі частинок середовища у звукових збуреннях характеризується такою величиною як амплітуда коливань. Для опису змін стану частинок середовища в часі використовується така характеристика як частота. Збурення, що сприймаються органами слуху людини та різних

тварин це лише невелика частка збурень, що можуть існувати в навколишньому світі. Людське вухо може сприймати лише звуки в певних обмежених інтервалах частот та амплітуд. Для багатьох випадків практичного використання звуків велике значення мають звуки, в яких основна частина енергії зосереджена в області частот, що не сприймаються людським вухом. В зв'язку з цим окремо виділяють такі особливі типи збурень, як ультразвук та інфразвук.

У філософії, психології й екології засобів комунікації звук досліджується у зв'язку з його впливом на сприйняття й мислення (мова йде, наприклад, про акустичний простір як простір, створюваний впливом електронних засобів комунікації).

Термін звук використовується для характеристики процесів поширення збурень в матеріальних середовищах, що мають пружні властивості. Прикладом таких середовищ є гази, рідини, тверді пружні тіла. В таких середовищах на будь-яку частинку, зміщену з положення рівноваги, діє сила пружності, яка намагається повернути її в вихідне положення. Характер руху частинок середовища при цьому може бути різним. При коротких збуреннях (постріл, плеск в долоні та ін.) здійснюються короткочасні рухи, після яких частинки повертаються до початкового рівноважного стану. При дії довготривалих, змінних у часі, збурень частинки середовища здійснюють коливальні рухи поблизу положення рівноваги.

Звукові хвилі при певних умовах можуть бути прикладом коливального процесу. Усяке коливання пов'язане з порушенням рівноважного стану системи й виражається у відхиленні її характеристик від рівноважних значень. При поширенні звукових хвиль в середовищі зазнають змін такі характеристики як густина, тиск, швидкість руху частинок, механічні (нормальні та дотичні) напруження, температура. Зміна положення частинок характеризується вектором швидкості, який є функцією координат і часу. Силкові фактори взаємодії між частинками акустичного середовища в випадку

рідин та газів характеризуються зміною величини тиску. Для пружних тіл силова взаємодія між частинками описується тензором напружень.

Якщо зробити різкий стиск пружного середовища в одному місці, наприклад, за допомогою поршня, то в цьому місці збільшиться тиск. Завдяки пружним зв'язкам між атомами та молекулами тиск передається на сусідні частинки, які, у свою чергу, впливають на наступні, і область підвищеного тиску немов переміщується в пружному середовищі. За областю підвищеного тиску слідує область зниженого тиску, і, таким чином, утвориться чергування областей стиску й розрідження, що поширюються в середовищі у вигляді хвилі. Кожна частка пружного середовища в цьому випадку здійснює коливальні рухи.

У рідких і газоподібних середовищах акустичні хвилі мають поздовжній характер, тобто напрямком коливання частинок збігається (або протилежний) з напрямком поширення хвилі. У твердих тілах, крім поздовжніх деформацій, виникають також пружні деформації зсуву, що зумовлює можливість існування хвиль з рухом частинок в напрямках, перпендикулярних до напрямку поширення хвилі. Такі хвилі називають поперечними.

1.2 Класифікація звуків

Більшість звуків, які зустрічаються в природі складні, тобто є суперпозицією (сумою) хвиль різної частоти. Суперпозиція хвиль різної частоти надає чутному звуку забарвлення, яке називають тембром.

За частотними характеристиками акустичної хвилі розрізняють прості та складні тони, шум. Прості тони – синусоїдні коливання, звукові коливання простого тону в достатньо хорошому наближенні випромінюють звукові генератори і камертон. Складні тони розділяють на гармонічні та негармонічні.

Гармонічні – визначеної звуковисотності, що складається з основного тону та обертонів. Звуки такого спектрального складу дають музичні інструменти. Тембр звуку, тобто співвідношення обертонів та основного тону надає кожному музичному інструменту своє характерне звучання. Негармонічний – приблизно визначеної звуковисотності, що складається з основного тону та негармонічних обертонів

Шум поділяють на білий або кольоровий:

- білий шум – хаотичні коливання, спектральні складові розміщуються рівномірно по всьому діапазону;
- кольоровий шум – хаотичні коливання, спектральні складові розміщуються нерівномірно по всьому діапазону, як правило з поступовим зменшенням інтенсивності від низьких до високих частот.

1.3 Характеристики звуку

При описі звукових полів використовують такі фізичні характеристики, як частота, довжина хвилі, амплітуда, швидкість звуку, тиск та швидкість частинок середовища.

У зв'язку з особливістю руху частинок середовища при поширенні звукових хвиль цю швидкість називають коливальною швидкістю. Коли мова йде про сприйняття звуків людиною використовують так звані фізіологічні характеристики звуку: тембр, гучність, висота тону. Звичайно середньостатистична людина чує коливання, передані по повітрю, у діапазоні частот від 16–20 Гц до 15–20 кГц.

Звук нижче діапазону чутності людини називають інфразвуком; вище: до 1 ГГц, – ультразвуком, від 1 ГГц – гіперзвуком. Представники тваринного світу здатні сприймати звуки в значно ширшому діапазоні частот. Кажани для орієнтації та знаходження здобичі використовують звуки в діапазоні 20-120 кГц. Дельфіни, наприклад, використовують для орієнтації та полювання

звуки з частотою понад 100 кГц[4] Значно вищі 20 кГц частоти здатні сприймати і собаки.

Серед вловимих звуків варто особливо виділити фонетичні, мовні звуки й фонемі (з яких складається усне мовлення) і музичні звуки (з яких складається музика).

Розрізняють поздовжні й поперечні звукові хвилі залежно від співвідношення напрямку поширення хвилі й напрямки механічних коливань часток середовища поширення.

1.4 Фізичні параметри звуку

В багатьох практично важливих випадках для вивчення процесів генерації, поширення звукових хвиль та їх взаємодії з перешкодами використовується модель ідеальної стисливої рідини (газу) або ідеального пружного тіла. В цьому випадку процес поширення хвиль описується силовими характеристиками (тиск або тензор напружень) та кінематичними характеристиками (вектор коливальної швидкості частинок середовища. Коливальна швидкість вимірюється в м/с або см/с. Зміна температури в процесі адіабатичної деформації частинок середовища може бути обчислена додатково. Для врахування втрат енергії при деформації середовища слід використовувати моделі, що враховують в'язкість. При поширенні гармонічних хвиль в'язкі втрати часто враховуються введенням комплексних хвильових чисел чи комплексних модулів пружності.

В енергетичному відношенні реальні коливальні системи характеризуються зміною енергії внаслідок часткової її витрати на роботу проти сил тертя й випромінювання в навколишній простір. У пружному середовищі коливання поступово загасають. Для характеристики згасних коливань використовуються коефіцієнт загасання (S), логарифмічний декремент (D) і добротність (Q).

Коефіцієнт загасання відбиває швидкість спадання амплітуди із часом.

Якщо позначити час, протягом якого амплітуда зменшується в e разів, через τ , то:

$$S = \frac{1}{\tau} \dots\dots\dots(1.1)$$

Зменшення амплітуди за один цикл характеризується логарифмічним декрементом. Логарифмічний декремент дорівнює відношенню періоду коливань до часу загасання τ :

$$D = \frac{T}{\tau} \dots\dots\dots(1.2)$$

Якщо на коливальну систему із втратами діяти періодичною силою, то виникають вимушені коливання, характер яких тією чи іншою мірою повторює зміни зовнішньої сили. Частота вимушених коливань не залежить від параметрів коливальної системи. Навпаки, амплітуда залежить від маси, механічного опору й гнучкості системи, якими визначаються її власні частоти. Таке явище, коли амплітуда коливальної швидкості досягає максимального значення, називається механічним резонансом. При цьому частота вимушених коливань збігається із частотою власних незатухаючих коливань механічної системи. Характерно, що максимальне відхилення частинок середовища від положення рівноваги досягається при дещо іншому значенні частоти зовнішньої сили. Різниця між цими характерними частотами визначається величиною демпфування в системі.

При частотах впливу, значно менших резонансної, зовнішня гармонійна сила врівноважується практично тільки силою пружності. При частотах порушення, близьких до резонансного, головну роль грають сили тертя. За умови, коли частота зовнішнього впливу значно більше резонансної, поведження коливальної системи залежить від сили інерції або маси.

Властивість середовища проводити акустичну енергію, у тому числі й ультразвукову, характеризується акустичним опором. Як характеристика

середовища величина акустичного опору може бути визначена при аналізі поширення плоскої хвилі. В цьому випадку вводять для характеристики середовища поняття питомого акустичного опору. Його величина визначається відношенням амплітуди звукового тиску в середовищі до амплітуди коливальної швидкості його часток. Чисельно, питомий акустичний опір середовища (Z) дорівнює добутку густини (ρ) на швидкість (c) поширення звукових хвиль в ньому.

$$Z = \rho c \dots\dots\dots(1.3)$$

Звуковий або акустичний тиск у середовищі являє собою різницю між миттєвим значенням тиску в даній точці середовища при наявності звукових коливань і статичного тиску в тій же точці при їхній відсутності. Іншими словами, звуковий тиск є змінний тиск у середовищі, обумовлений акустичними коливаннями. Максимальне значення змінного акустичного тиску (амплітуда тиску) може бути розраховане через амплітуду коливання часток:

$$P = 2\pi\nu\rho cA \dots\dots\dots(1.4)$$

Найпростішим типом звуку є звук, в якому тиск у кожній точці простору змінюється за синусоїдним законом, тобто здійснює гармонічні коливання з певною частотою.

Частота – це кількість коливань певної точки середовища, в якому поширюються гармонічні звукової хвилі, в секунду. Одному циклу коливання в секунду відповідає величина 1 Гц (1/с). В загальному випадку довільних збурень зміна положення точки середовища в часі може характеризуватися набором дискретних значень частот, або неперервним інтервалом частот (скінченим чи нескінченим). В цьому випадку говорять про дискретний чи неперервний частотний спектр звуку.

Людина чує звук з частотами від 16 Гц до 20 кГц. Границі чутності визначені не строго і змінюються від людини до людини. Деякі тварини можуть чути звуки з частотою, нижчою від 16 Гц, інші – з частотою понад 20

кГц.

Діапазон від 16 Гц до 20 кГц називають чутним діапазоном. Звуки з частотами до 16 Гц називаються інфразвуком, понад 20000 Гц – ультразвуком. Звуки з частотою 109–1013 Гц називають гіперзвуком. Людське вухо сприймає та розрізняє частоту звукових коливань як висоту звуку або тон.

Швидкість звуку – швидкість поширення звукових хвиль у середовищі.

Як правило, в газах швидкість звуку менша, ніж в рідинах, а в рідинах швидкість звуку менша, ніж у твердих тілах, що пов'язано в основному зі зменшенням стисливості (зростанням об'ємного модуля пружності) речовин у цих фазових станах відповідно.

Швидкість звуку в повітрі за нормальних умов становить 340 м/с. Вона дещо зростає з підвищенням температури і зменшується при її пониженні. Швидкість звуку в повітрі практично не залежить від частоти, тому звук розповсюджується на великі відстані без спотворень. В газах швидкість звуку складає декілька сотень метрів на секунду.

Гучність звуку – суб'єктивне сприйняття сили звуку (абсолютна величина слухового відчуття). Гучність головним чином залежить від звукового тиску, амплітуди й частоти звукових коливань. Також на гучність звуку впливають його спектральний склад, локалізація в просторі, тембр, тривалість впливу звукових коливань, індивідуальна чутливість слухового аналізатора людини й інші фактори

Гучність звуку визначається амплітудою коливань, однак гучність – суб'єктивна характеристика інтенсивності звуку, тоді як об'єктивною фізичною характеристикою є звуковий тиск.

Людське вухо сприймає гучність у приблизно логарифмічному масштабі за законом Вебера-Фехнера, тому гучність вимірюється в логарифмічних одиницях – децибелах, тоді як звуковий тиск вимірюється в паскалях. Логарифмічний масштаб сприйняття означає, що людина може почути новий звук на деякому звуковому тлі тільки тоді, коли його амплітуда

перевищує амплітуду тла не на деяку певну абсолютну величину, а на певний множник, який залежить від частоти. Аналогічно, у логарифмічному масштабі людське вухо розрізняє тони.

1.5 Реєстрація, запис та відтворення звуку

При розв'язанні завдання реєстрації звуків використовують два типи сенсорів – вимірювач тиску чи вимірювач коливальної швидкості. Звичайний мікрофон та людське вухо є реєстраторами тиску. Практично лише такого типу реєстратори використовуються при аналізі звукових полів в газових середовищах. В гідроакустиці досить часто використовують реєстратори коливальної швидкості.

Звук реєструють за допомогою мікрофонів – приладів, що перетворюють звукові коливання у електричні. Зареєстровані звукові коливання можна передати на віддаль засобами телекомунікації – телефоном або радіо або записати на носії інформації. Переданий або записаний звук відтворюється за допомогою гучномовців, які перетворюють електричні коливання у звукові хвилі.

В сучасних умовах для реєстрації та обробки звукових сигналів широко використовуються комп'ютери. При цьому звуковий сигнал кодується цифровим кодом з допомогою спеціальних аналого-цифрових перетворювачів. При цьому формується цифровий звук. При створенні цифрових звукових файлів використовується велика кількість різних звукових форматів. Широко використовуються, наприклад, формати MP3, MP4. При використанні аналого-цифрових перетворювачів ставиться завдання збереження характеристик звуку в певному частотному діапазоні, у відповідності до якого вибирається частота дискретизації.

2 ПЕРЕТВОРЕННЯ ФУР'Є ТА ХАРТЛІ

2.1 Перетворення Фур'є

Перетворення Фур'є (ПФ) – це математичне перетворення, яке розкладає функції на частотні компоненти, які представлені результатом перетворення як функцією частоти. Найчастіше функції часу або простору перетворюються, що виводить функцію залежно від часової або просторової частоти відповідно. Цей процес ще називають аналізом. Прикладом застосування може бути розкладання форми хвилі музичного акорду в термінах інтенсивності його складових тонів. Термін «перетворення Фур'є» відноситься як до представлення частотної області, так і до математичної операції, яка пов'язує представлення частотної області з функцією простору або часу.

Перетворення Фур'є функції – це функція з комплексним значенням, що представляє комплексні синусоїди, які складають вихідну функцію. Для кожної частоти величина (абсолютне значення) комплексного значення представляє амплітуду складової комплексної синусоїди з цією частотою, а аргумент комплексного значення представляє зміщення фази цієї комплексної синусоїди. Якщо частота відсутня, перетворення має значення 0 для цієї частоти. Перетворення Фур'є не обмежується функціями часу, але область початкової функції зазвичай називають часовою областю. Теорема обернення Фур'є забезпечує процес синтезу, який відтворює вихідну функцію з її подання в частотній області.

Функції, локалізовані в часовій області, мають перетворення Фур'є, які розповсюджуються по частотній області, і навпаки, явище, відоме як принцип невизначеності. Критичним випадком для цього принципу є функція Гауса, яка має суттєве значення в теорії ймовірностей і статистиці, а також у вивченні фізичних явищ, що мають нормальний розподіл (наприклад,

дифузія). Перетворення Фур'є функції Гауса є іншою функцією Гауса. Жозеф Фур'є запровадив перетворення у своєму вивченні теплообміну, де функції Гауса виступають як розв'язки рівняння теплопровідності.

Перетворення Фур'є також можна узагальнити на функції кількох змінних у евклідовому просторі, посилаючи функцію 3-вимірного «простору положення» у функцію 3-вимірного імпульсу (або функцію простору та часу у функцію 4-імпульсу). Ця ідея робить просторове перетворення Фур'є дуже природним у вивченні хвиль, а також у квантовій механіці, де важливо мати можливість представляти хвильові рішення як функції позиції або імпульсу, а іноді і того, і іншого. Загалом, функції, до яких застосовуються методи Фур'є, є комплекснозначними та, можливо, векторнозначними.

Ще можливе подальше узагальнення для функцій на групах, які, окрім оригінального перетворення Фур'є на \mathbb{R} або \mathbb{R}^n (розглядаються як групи під додаванням), зокрема, включає дискретне перетворення Фур'є (DTFT, група $= \mathbb{Z}$), дискретне перетворення Фур'є (ДПФ, група $= \mathbb{Z} \bmod N$) і ряд Фур'є або кругове перетворення Фур'є (група $= S^1$, одиничне коло \approx замкнутий кінцевий інтервал з визначеними кінцевими точками).

Перетворення Фур'є функції f дійсної змінної є невід'ємною частиною і задається наступною формулою:

$$\hat{f}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-ix\omega} dx. \quad (2.1)$$

Різні джерела можуть давати визначення, що відрізняються від приведеного вище вибору множника перед інтегралом (так званого нормованого множника, який відноситься до питання про нормування перетворення Фур'є), а також знак « $-$ » у показниках експонентів. Але поза залежністю від таких варіацій всі властивості будуть зберігати свою силу, хоча вид деяких формул може змінитися.

$$\hat{f}(\omega) = \sqrt{\frac{|b|}{(2\pi)^{1-a}}} \int_{-\infty}^{\infty} f(x) e^{-ibx\omega} dx. \quad (2.2)$$

Зворотнє перетворення визначається так:

$$f(x) = \sqrt{\frac{|b|}{(2\pi)^{1+a}}} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{ib\omega x} d\omega. \quad (2.3)$$

При виборі $a = 0$ і $b = 2\pi$ або $b = -2\pi$ і формули стають особливо прості, в них зникають нормувальні множники і формули відрізняються лише знаком ступеня, внаслідок чого більшість наведених вище формул спрощуються на постійні константи.

2.2 Дискретне перетворення Фур'є

В математиці дискретне перетворення Фур'є (ДПФ) перетворює скінченну послідовність рівновіддалених відліків функції в однакову довжину рівновіддалених відліків дискретного перетворення Фур'є (ДПФ), яке є комплексним відображенням функція частоти. Обернене ДПФ – це ряд Фур'є, в якому використовуються вибірки ДЧПФ як коефіцієнти комплексних синусоїд на відповідних частотах ДЧПФ. Він має ті самі значення вибірки, що й вихідна вхідна послідовність.

Таким чином, ДПФ називається представленням початкової вхідної послідовності в частотній області. Якщо вихідна послідовність охоплює всі ненульові значення функції, її ДПФ є безперервним (і періодичним), а ДПФ забезпечує дискретні вибірки одного циклу. Якщо вихідна послідовність є одним циклом періодичної функції, ДПФ забезпечує всі ненульові значення одного циклу ДЧПФ.

ДПФ є найважливішим дискретним перетворенням, яке використовується для виконання аналізу Фур'є в багатьох практичних застосуваннях. У цифровій обробці сигналу функція – це будь-яка величина або сигнал, який змінюється з часом, наприклад тиск звукової хвилі, радіосигнал або щоденні показники температури, відібрані протягом кінцевого інтервалу часу (часто визначається віконною функцією). Під час обробки зображень вибірками можуть бути значення пікселів у рядку чи стовпці растрового зображення. ДПФ також використовується для ефективного вирішення диференціальних рівнянь із частинними похідними та для виконання інших операцій, таких як згортка або множення великих цілих чисел.

Оскільки він має справу з обмеженою кількістю даних, його можна реалізувати в комп'ютерах за допомогою числових алгоритмів або навіть спеціального апаратного забезпечення. Ці реалізації зазвичай використовують ефективні алгоритми швидкого перетворення Фур'є (ШПФ) настільки, що терміни «ШПФ» і «ДПФ» часто використовуються як синоніми.

Дискретне перетворення Фур'є перетворює послідовність з N комплексних чисел $\{x\} := x_0, x_1, \dots, x_{N-1}$ до іншої послідовності комплексних чисел $\{X_k\} := X_0, X_1, \dots, X_{N-1}$, що визначаються за формулою, де останній вираз впливає з першого за формулою Ейлера:

$$\begin{aligned} X_k &= \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{i2\pi}{N} kn} \\ &= \sum_{n=0}^{N-1} x_n \cdot \left[\cos\left(\frac{2\pi}{N} kn\right) - i \cdot \sin\left(\frac{2\pi}{N} kn\right) \right] \end{aligned} \quad (2.4)$$

Дискретне перетворення Фур'є є оборотним лінійним перетворенням

$$\mathcal{F}: \mathbb{C}^N \rightarrow \mathbb{C}^N \quad (2.5)$$

з \mathbb{C} , що позначає набір комплексних чисел. Його обернене відоме як зворотне дискретне перетворення Фур'є (ОДПФ). Іншими словами, для будь-якого $N > 0$ N -вимірний комплексний вектор має ДПФ та ОДПФ, які, у свою

чергу, є N -вимірними комплексними векторами.

Обернене перетворення визначається як:

$$X_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k \cdot e^{\frac{i2\pi}{N} kn} \quad (2.6)$$

Таблиця 2.1 – Дійсна та умовна частина

Властивості	Часова область x_n	Частотний домен X_k
Дійсна частина в часі	$\Re(x_n)$	$\frac{1}{2} (X_k + X_{N-k}^*)$
Уявна частина часу	$\Im(x_n)$	$\frac{1}{2i} (X_k - X_{N-k}^*)$
Дійсна частина в частоті	$\frac{1}{2} (x_n + x_{N-n}^*)$	$\Re(X_k)$
Уявна частина частоти	$\frac{1}{2i} (x_n - x_{N-n}^*)$	$\Im(X_k)$

У цій таблиці показано деякі математичні операції над x_n в часовій області та відповідний вплив на її ДПФ X_k у частотній області.

2.3 Перетворення Хартлі та дискретне перетворення Хартлі

Перетворення Хартлі – інтегральне перетворення, тісно пов'язане з перетворенням Фур'є, але на відміну від останнього перетворює одні дійсні функції в інші дійсні функції.

Перетворення було запропоновано в якості альтернативи перетворення Фур'є Р. Хартлі в 1942 році. Перетворення Хартлі є одним із багатьох відомих типів перетворень Фур'є. Перетворення Хартлі може бути зворотнім.

Можливо, одним із найтриваліших внесків Хартлі був більш симетричний інтеграл Фур'є, спочатку розроблений для стаціонарного та перехідного аналізу проблем системи телефонної передачі. Хоча це

перетворення залишалося в стані спокою більше 40 років, перетворення Хартлі було заново відкрито більше десяти років тому Вангом і Брейсуеллом, які є авторами остаточних трактатів на цю тему. Дискретний варіант перетворення Хартлі був представлений Рональдом Брейсуеллом у 1983 року.

Пряме перетворення:

$$H(\omega) = \{\mathcal{H}f\}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) \text{cas}(\omega t) dt, \quad (2.7)$$

$$\text{cas}(t) = \cos(t) + \sin(t) = \sqrt{2} \sin\left(t + \frac{\pi}{4}\right) = \sqrt{2} \cos\left(t - \frac{\pi}{4}\right) \quad (2.8)$$

Де (ω) може бути кутовою частотою і (2.8) є косинус-синус (cas) або ядро Хартлі. Інженерно кажучи, це перетворення переносить сигнал (функцію) з часової області в спектральну область Хартлі (частотну область).

Зворотне перетворення Хартлі

$$f = \{\mathcal{H}\{\mathcal{H}f\}\} \quad (2.9)$$

Зворотне перетворення (2.9) виходить за принципом інволюції.

Це перетворення відрізняється від класичного перетворення Фур'є $F(\omega) = \mathcal{F}\{f(t)\}(\omega)$ виборам ядра перетворення.

У перетворенні Фур'є ми маємо експоненціальне ядро:

$$\exp(-i\omega t) = \cos(\omega t) - i\sin(\omega t) \exp(-i\omega t) = \cos(\omega t) - i\sin(\omega t) \quad (2.10)$$

Однак ці два перетворення тісно пов'язані, і перетворення Фур'є (за умови, що воно використовує ту саму умову нормалізації $1/\sqrt{2\pi}$ обчислено з перетворення Хартлі через:

$$F(\omega) = \frac{H(\omega) + H(-\omega)}{2} - i \frac{H(\omega) - H(-\omega)}{2} \quad (2.11)$$

Тобто дійсна та уявна частини перетворення Фур'є просто задані парною та непарною частинами перетворення Хартлі відповідно. І навпаки, для дійсних функцій $f(t)$ перетворення Хартлі задається з дійсної та уявної частин перетворення Фур'є, де \Re є дійсною частиною, а \Im уявною частиною:

$$\{\mathcal{H}f\} = \Re\{\mathcal{F}f\} - \Im\{\mathcal{F}f\} = \Re\{\mathcal{F}f \cdot (1 + i)\} \quad (2.12)$$

Дискретне перетворення Хартлі (ДПХ) – це пов’язане з Фур’є перетворення дискретних періодичних даних, подібне до дискретного перетворення Фур’є (ДПФ), з аналогічним застосуванням в обробці сигналів і пов’язаних областях. Його головна відмінність від ДПФ полягає в тому, що він перетворює дійсні вхідні дані в дійсні виходи без внутрішньої участі комплексних чисел. Подібно до того, як DFT є дискретним аналогом безперервного перетворення Фур’є (ПФ), DHT є дискретним аналогом безперервного перетворення Хартлі (ПХ), представленого Ральфом В.Л. Хартлі в 1942 році.

Формально дискретне перетворення Хартлі є лінійною оборотною функцією $H: \mathbb{R}^n \rightarrow \mathbb{R}^n$ (де \mathbb{R} позначає множину дійсних чисел). N дійсних чисел x_0, \dots, x_{N-1} перетворюються на N дійсних чисел H_0, \dots, H_{N-1} за формулою

$$H_k = \sum_{n=0}^{N-1} x_n \operatorname{cas}\left(\frac{2\pi}{N}nk\right) = \sum_{n=0}^{N-1} x_n \left[\cos\left(\frac{2\pi}{N}nk\right) + \sin\left(\frac{2\pi}{N}nk\right) \right] \\ k = 0, \dots, N - 1. \quad (2.13)$$

$$\cos(z) + \sin(z) = \sqrt{2} \cos\left(z - \frac{\pi}{4}\right) \quad (2.14)$$

Комбінація (2.14) іноді позначається як $\operatorname{cas}(z)$, і її не слід плутати з $\operatorname{cis}(z) = e^{iz} = \cos(z) + i \sin(z)$ або $e^{-iz} = \operatorname{cis}(-z)$, які з’являються в визначенні ДПФ (де i – уявна одиниця).

Як і у випадку з ДПФ, загальний масштабний коефіцієнт перед перетворенням і знак синуса є умовністю. Хоча ці конвенції час від часу відрізняються між авторами, вони не впливають на основні властивості перетворення.

2.4 Порівняння перетворень Фур’є та Хартлі

Дискретне перетворення Фур’є (ДПФ) перетворює послідовність

дійсних чисел із часової області в послідовність комплексних чисел у частотній області. Половина чисел у частотній області зайві, оскільки інформація в негативній частотній області частоти повторюють інформацію на позитивних частотах. Іншими словами, якщо $X_f(k)$ є перетворенням Фур'є дійсної послідовності дійсних чисел, то

$$\text{Real}(X_f(k)) = \text{Real}(X_f(-k)) \quad (2.15)$$

$$\text{Imag}(X_f(k)) = -\text{Imag}(X_f(-k)) \quad (2.16)$$

Для множення комплексних чисел у комп'ютері потрібно чотири множення з плаваючою комою та два додавання з плаваючою комою. Через необхідний обсяг пам'яті, надлишкову інформацію та кількість необхідних обчислень перетворення Фур'є не є найефективнішим методом перетворення дійсних чисел у частотну область. Якщо вхідна послідовність не складається з комплексних чисел, перетворення Фур'є неефективне. Деякі програми DFT використовують алгоритм, який перепаковує числа, щоб усунути надмірність. Наприклад, негативні реальні значення частоти замінюються позитивними уявними значеннями, а всі інші уявні значення ігноруються.

$$X_h(k) = \text{Real}[X_f(k)] - j \text{Imag}[X_f(k)] \quad (2.17)$$

Дискретне перетворення Хартлі (ДПХ) перетворює послідовність дійсних чисел у часовій області в послідовність дійсних чисел у частотній області. Перетворення Хартлі можна розглядати, як алгоритм, який усуває надлишковість в області Фур'є, перепаковуючи числа через відношення (2.17), де $X_h(k)$ – це перетворення Хартлі послідовності, а $X_f(k)$ – це перетворення Фур'є тієї ж послідовності.

Як видно з рівняння (2.17), результатом перетворення Хартлі є дійсні числа. Жодна інформація не втрачається, оскільки дані в домені Хартлі можуть бути перетворені назад в домен Фур'є за допомогою відношень

$$\text{Real}[X_f(k)] = [X_h(k) + X_h(-k)]/2 \quad (2.18)$$

$$\text{Imag}[X_f(k)] = [X_h(k) - X_h(-k)]/2 \quad (2.19)$$

Оскільки жодна інформація не втрачається під час перетворення, реальна послідовність у частотній області використовує мінімально необхідний простір пам'яті. Представлення частоти є реальним, що означає, що перетворення Хартлі вимагає менше множень. Виходячи з цих основних критеріїв, перетворення Хартлі ефективніше, ніж пряме перетворення Фур'є.

Перетворення Фур'є та Хартлі дуже схожі та мають багато властивостей. Серед цих властивостей – правило зсуву та періодичні властивості їх ядер. Обидві ці властивості корисні для побудови швидкого перетворення Фур'є (ШПФ).

2.5 Швидке перетворення Фур'є

N-точкове дискретне перетворення Фур'є часової послідовності $x(n)$ має формулу (2.4). Використовуючи симетричні властивості перетворення цього перетворення можливо розбити оригінальну N-точкову послідовність на дві менших (N/2)-точкові послідовності:

$$x_1(n) = x(2n), n = 0, 1, \dots, \frac{N}{2} - 1 \quad (2.20)$$

$$x_2(n) = x(2n + 1), n = 0, 1, \dots, \frac{N}{2} - 1 \quad (2.21)$$

Перша послідовність $x_1(n)$ містить парні члени $x(n)$, а $x_2(n)$ містить непарні члени $x(n)$. Дві (N/2)-точкові послідовності перетворюються Фур'є та об'єднуються в N-точкове ДПФ $x(n)$ за формулою:

$$X_f(k) = \sum_{n=0}^{N/2-1} x(2n)e^{-\frac{2\pi i}{N}2kn} + \sum_{n=0}^{N/2-1} x(2n + 1)e^{-\frac{2\pi i}{N}(2n+1)k} \quad (2.22a)$$

Другий член у рівнянні (2.22a) не є справжнім ДПФ, оскільки ядро у другому члені не є ядром ДПФ.

$$e^{-\frac{2\pi i}{N}(2n+1)k} = e^{-\frac{2\pi i}{N}k} e^{-\frac{2\pi i}{N}2nk} \quad (2.23)$$

Використовуючи тотожність (2.23) до рівняння (2.22a) отримаємо:

$$X_f(k) = \sum_{n=0}^{N/2-1} x(2n)e^{-\frac{2\pi i}{N}2nk} + e^{-\frac{2\pi i}{N}k} \sum_{n=0}^{N/2-1} x(2n+1)e^{-\frac{2\pi i}{N}2nk} \quad (2.226)$$

Другий член рівняння (2.226) тепер є справжнім ДПФ. Рівняння (4а) можна переписати, так що X_{f1} – дискретне перетворення Фур'є $x_1(n)$, $X_{f2}(k)$ – дискретне перетворення $x_2(n)$.

$$X_f(k) = X_{f1}(k) + e^{-\frac{2\pi i}{N}k} X_{f2}(k) \quad (2.24)$$

Комплексна експонента у другому члені рівняння (2.24) виходить з правила зсуву перетворення Фур'є, яке задається формулою (2.25), де c є константою:

$$X_f(k+c) = e^{-i\frac{2\pi}{N}c} X_f(k) \quad (2.25)$$

Коли член $e^{-i\frac{2\pi}{N}k}$ було винесено з другого доданка у рівнянні (2.226) часову послідовність була зміщена на один вліво. Здвигнута частина $x_2(n)$ була перетворення за допомогою ПФ до $X_{f2}(k)$. Частина здвигу $e^{-i\frac{2\pi}{N}k}$ в рівнянні (2.24) здвигує заново $x_2(n)$ до його початкового положення відносно часового виміру. Коли $X_{f2}(k)$ повернулось на місце, сума двох частин в рівнянні (2.22а) має місце. Правило здвигу дозволяє ШПФ алгоритму взяти дві $N/2$ -точкову послідовність і скомбінувати до N -точкової послідовності.

Частини $X_{f1}(k)$ та $X_{f2}(k)$ в рівнянні (2.24) визначені на проміжку $0 \leq k \leq N/2 - 1$. Так як $X_f(k)$ визначена на проміжку $0 \leq k \leq N - 1$, необхідне правило, щоб отримати значення для $X_f(k)$ де $k \geq N/2$. Для отримання правила використовуються періодичні властивості ДПФ. Ядро ДПФ, $e^{-i\frac{2\pi}{N}k}$, періодичне та має період N . Наприклад, $e^{-i\frac{2\pi}{N}2nk}$ у виразі (2.226) має період $(N/2)$. Коли k стає більше $(N/2 - 1)$, тоді $(N/2)$ може бути ввідняте від k і отримати:

$$e^{-i\frac{2\pi}{N}2(k-N/2)} = e^{-i\frac{2\pi}{N}2k} \quad (2.26)$$

Обидві частини в рівнянні (2.24) $X_{f1}(k)$ та $X_{f2}(k)$ має період $(N/2)$. Комбінуючи рівняння (2.226) та (2.26), $X_f(k)$ для усіх k становить

$$X_f(k) = \begin{cases} X_{f_1}(k) + e^{-i(2\pi/N)k} X_{f_2}(k) & 0 \leq k \leq \frac{N}{2} - 1 \\ X_{f_1}(k - N/2) + e^{-i(2\pi/N)k} X_{f_2}(k - N/2) & \frac{N}{2} \leq k \leq N - 1 \end{cases} \quad (2.27)$$

Використовуючи факт, що частина $e^{-i\frac{2\pi}{N}k}$ в рівнянні (2.24) має період N та знаючи тотожність (2.23) рівняння (2.27) може бути переписане:

$$X_f(k) = \begin{cases} X_{f_1}(k) + e^{-i(2\pi/N)k} X_{f_2}(k) & 0 \leq k \leq \frac{N}{2} - 1 \\ X_{f_1}(k - N/2) - e^{-i(2\pi/N)(k - N/2)} X_{f_2}(k - N/2) & \frac{N}{2} \leq k \leq N - 1 \end{cases} \quad (2.28)$$

Періодична властивість ДПФ дозволяє алгоритму ШПФ обчислювати всі значення N -точкової послідовності із двох $(N/2)$ -точкових послідовностей.

У верхній частині рисунка 2.1 показано потокові графіки для обчислення восьмиточкового ДПФ із двох чотирьохточкових перетворень. Вхідні послідовності ліворуч – $X_{f_1}(k)$ та $X_{f_2}(k)$, а вихідна послідовність праворуч – $X_f(k)$. Рядки на малюнку 2.1 показують потік даних зліва на праву сторону. Нижня частина рисунка 2.1 показує загальний графік метеликового потоку ШПФ. Якщо a і b є вхідними даними з лівого боку графіка потоку, то вихідні дані з правого боку є $(a + wb)$ і $(a - wb)$. У верхній частині рисунка 2.1 $X_{f_1}(k)$ відповідає до a , $X_{f_2}(k)$ відповідає до b , w відповідає до фактору $e^{-i\frac{2\pi}{N}k}$.

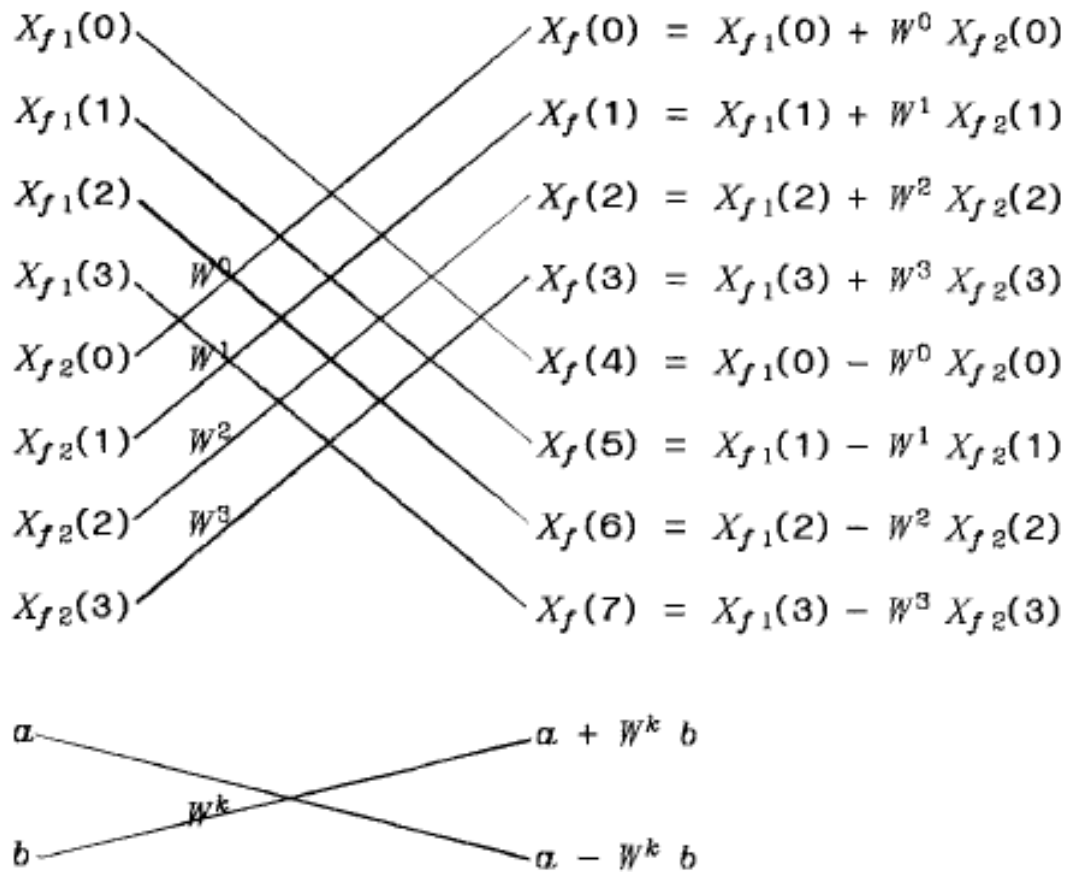


Рисунок 2.1 – Метеликовий потоковий графік для ШПФ

Метелик ШПФ використовується рекурсивно для об'єднання коротших послідовностей ДПФ у довші послідовності ДПФ.

На рисунку 2.1 показано перетворення для восьмиточкового перетворення. ШПФ метелик працює з будь-якою послідовністю, довжина якої дорівнює двійці. ШПФ починається з виконання двоточкових перетворень вхідної послідовності. ШПФ отримує вихідні дані двоточкових перетворень і повторює процес, але з чотириточковим перетворенням. ШПФ продовжується таким чином, доки не буде досягнуто N .

2.6 Швидке перетворення Хартлі

Алгоритм ФНТ починається так само, як і FFT. По-перше, ФНТ розбиває N -точкову послідовність $x^{(n)}$ на дві менші $(N/2)$ -точкові

послідовності $x_1(n)$ і $x_2(n)$. Дві послідовності трансформуються Хартлі в $X_{h1}(k)$ і $X_{h2}(k)$ та об'єднуються в $X_h(k)$, як це робить ШПФ у рівнянні (2.23а). Однак перетворення Хартлі поєднує $X_{h1}(k)$ і $X_{h2}(k)$ за допомогою рівняння:

$$X_h(k) = \sum_{n=0}^{N/2-1} x(2n) \cos((2\pi/N) 2nk) + \sum_{n=0}^{N/2-1} x(2n+1) \cos((2\pi/N) (2n+1)k). \quad (2.29)$$

Як і для рівняння (2.23а), другий член у рівнянні (2.29) не є ДНТ, оскільки часова послідовність $x_2(n)$ зміщена вліво. Часову послідовність $x_2(n)$ потрібно зсунути вліво на одиницю за правилом зсуву ДНТ, яке визначається формулою (2.30), де c є константою:

$$X_h(k+c) = X_h(k) \cos(c) + X_h(-k) \sin(c) \quad (2.30)$$

$$X_h(k) = X_{h1}(k) + \cos((2\pi/N)k) X_{h2}(k) + \sin((2\pi/N)k) X_{h2}(-k). \quad (2.31)$$

Після зсуву $x_2(n)$ за допомогою рівняння (2.30) і рівняння (2.29) отримуємо рівняння (2.31), де $X_{h1}(k)$ є перетворення Хартлі послідовності $x_1(n)$, а $X_{h2}(k)$ – перетворення Хартлі послідовності $x_2(n)$. Використовуючи правило здвигу, дві $(N/2)$ -точкові послідовності ДПХ можуть бути скомбіновані до однієї N -точковою послідовності ДПХ.

Правило обчислення значень $X_h(k)$, коли $k \geq N/2$, ідентичне правилу для перетворення Фур'є. Ядро ДНТ має такі ж періодичні властивості, як і ядро ДПФ.

$$\cos\left(\frac{2\pi}{N}(n+N)\right) = \cos\left(\frac{2\pi}{N}n\right) \quad (2.32)$$

$$\cos\left(\frac{2\pi}{N}(n+N/2)\right) = -\cos\left(\frac{2\pi}{N}n\right) \quad (2.33)$$

Використовуючи періодичні властивості з виразів вище значення $X_h(k)$ для всіх k може бути описано системою (2.34)

$$X_h(k) = \begin{cases} X_{h_1}(k) + \cos((2\pi/N)k) X_{h_2}(k) + \sin((2\pi/N)k) X_{h_2}(-k) & 0 \leq k \leq \frac{N}{2} - 1 \\ X_{h_1}(k - N/2) - \cos((2\pi/N)(k - N/2)) X_{h_2}(k - N/2) - \sin((2\pi/N)(k - N/2)) X_{h_2}(-k + N/2) & \frac{N}{2} \leq k \leq N - 1 \end{cases} \quad (2.34)$$

Рівняння (2.34) дає основну формулу для алгоритму ШПФ. Основна відмінність між рівнянням (2.34) для ШПФ і рівнянням (2.28) для ШПФ полягає в тому, що для кожного k , $X_h(k)$ вимагає трьох значень, а $X_f(k)$ вимагає лише двох значень. Оскільки рівняння (2.34) потребує трьох значень, стандартний графік метеликового потоку ШПФ, показаний на малюнку 2.1, не можна використовувати для ШПХ. Натомість для реалізації рівняння (2.28) ШПФ вимагає подвійного графіка потоку метелика.

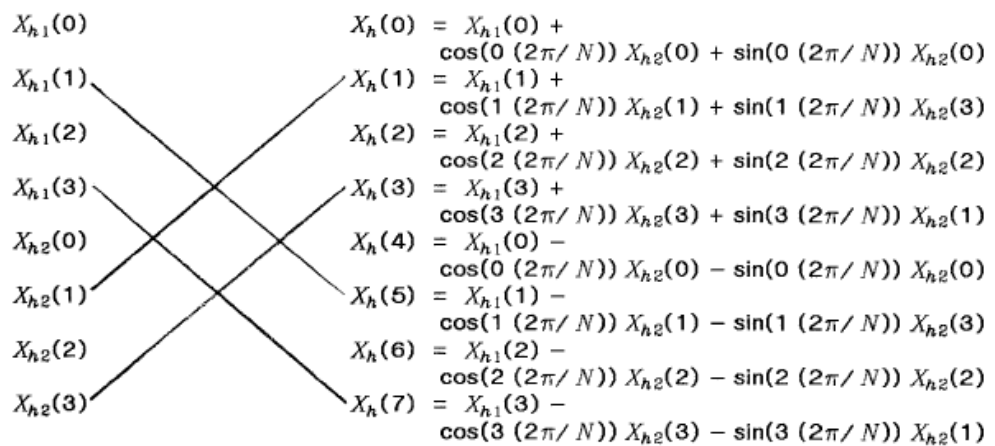


Рисунок 2.2 – Подвійний метелик ГНТ для послідовності з восьми точок.

На рисунку 2.2 показаний один із подвійних метеликів, який використовується для оцінки восьмиточкового перетворення два чотириточкових перетворення. Подвійний метелик використовує чотири входи і створює чотири виходи. Входи з лівої частини $X_{h_1}(1)$, $X_{h_1}(3)$, $X_{h_2}(1)$ та $X_{h_2}(3)$, виходи з правої частини $X_h(1)$, $X_h(3)$, $X_h(5)$ та $X_h(7)$. Є причина, чому подвійний метелик потребує чотирьох входів, коли рівняння (2.34) вимагає лише трьох. Відповідно до рівняння (2.34), вихід $X_h(1)$ вимагає входів $X_{h_1}(1)$,

$X_{h2}(1)$ та $X_{h2}(-1)$.

Оскільки $X_{h2}(k)$ не визначено для $k < 0$, потрібен відповідний запис. Негативні компоненти частоти N-точкової послідовності зберігаються після $(N/2)$.

2.7 Порівняння між ШПФ та ШПХ

Існує два критерії, за якими порівнюються переваги та недоліки ШПФ і ШПХ: швидкість виконання та вимоги до пам'яті. Очевидно, що алгоритм ШПФ потребує менше пам'яті, оскільки всі дані зберігаються в масивах дійсних чисел. ШПФ використовує складні масиви, які потребують вдвічі більше пам'яті, ніж реальний масив. У більшості систем геофізичної обробки комп'ютер має великий обсяг основної пам'яті, тому швидкість обчислення є найважливішим фактором. Більшість обчислень перетворень ШПФ і ШПХ виконано в межах петель метелика.

Існують деякі алгоритми ШПФ, які мають приблизно таку саму швидкість, що й алгоритм ШПХ. Ці алгоритми ШПФ є швидкими, оскільки вони попередньо обробляють дані шляхом перепакування дійсних чисел у комплексні числа, таким чином скорочуючи довжину вхідної послідовності вдвічі. ШПФ завжди виконуватиме певну кількість складних множень і додавання. Коли реальні дані стискаються до складної форми, алгоритм ефективніше використовує необхідні множення та додавання.

Однією з переваг обчислень у частотній області є те, що потрібно менше операцій. Наприклад, згортка відбувається швидше в частотній області, ніж у часовій області. Якщо використовувати перетворення Хартлі замість перетворення Фур'є, обчислення в простір Хартлі займає менше часу, ніж простір Фур'є.

Деякі з важливих операцій у частотній області включають або парну, або непарну функцію. Наприклад, простий фільтр низьких частот є парною функцією в частотній області. Симетрія парних і непарних функцій

позбавить від деякої кількості операцій з плаваючою комою. У наведених нижче порівняннях розглядається кількість операцій з плаваючою комою для парної функції та загальної функції. Наведені нижче рівняння для парних функцій є такими самими рівняннями для непарної функції, тому кількість обчислень однакова для парних і непарних функцій.

3 РОЗРОБКА СЕРЕДИ ДЛЯ АНАЛІЗУ ЕФЕКТИВНОСТІ ШПХ ТА ШПФ ДЛЯ РОБОТИ ЗІ ЗВУКОМ

3.1 Вибір платформи та вимоги

Платформою були обрані desktop-пристрої, а середовищем аналізу було обрано формат прикладної програми. До прикладної програми аналізу ефективності ШПФ та ШПХ були поставлені такі вимоги:

- програма повинна працювати на ОС Windows та Linux;
- програма повинна обробляти звуковий сигнал в реальному часі;
- програма повинна підтримувати хоча б два перетворення ШПФ та ШПХ;
- програма має працювати з музичними файлами;
- програма повинна малювати графік амплітуди та частот в реальному часі;
- програма повинна малювати спектрограму для аудіо-файлів;
- програма має підтримувати відображення часу виконання та зайняту пам'ять для ШПФ та ШПХ.

3.2 Вибір засобів розробки

Засоби розробки або стек розробки – це набір інструментів, які будуть використовуватись при розробці. Сюди відносять мови програмування, фреймворки бібліотеки, компілятори та інше.

Для розробки була обрана мова програмування C++.

3.2.1 Мова програмування C++

Мова C++ – компіювана, статично типізована мова програмування

загального призначення. Підтримує такі парадигми програмування, як процедурне програмування, об'єктно-орієнтоване програмування, узагальнене програмування. Мова має багату стандартну бібліотеку, яка включає поширені контейнери і алгоритми, введення-виведення, регулярні висловлювання, підтримку багатопоточності та інші можливості. C++ поєднує властивості як високорівневих, і низькорівневих мов. У порівнянні з його попередником – мовою C - найбільшу увагу приділено підтримці об'єктно-орієнтованого та узагальненого програмування.

C++ широко використовується для розробки програмного забезпечення, будучи однією з найпопулярніших мов програмування. Область його застосування включає створення операційних систем, різноманітних прикладних програм, драйверів пристроїв, додатків для систем, високопродуктивних серверів, а також комп'ютерних ігор. Існує безліч реалізацій мови C++ як безкоштовних, так і комерційних і для різних платформ.

Синтаксис C++ успадкований від мови C. Спочатку одним із принципів розробки було збереження сумісності з C. Тим не менш, C++ не є в строгому сенсі надмножиною C; безліч програм, які можуть однаково успішно транслюватися як компіляторами C, і компіляторами C++, досить велика, але включає всі можливі програми на C.

Вона дозволяє досягнути оптимальної швидкості для роботи в реальному часі та підходить для роботи з драйверами звуку та гарно працює з байтами, що є безцінним при роботі зі звуком у реальному часі. Також мова C++ має декілька дуже корисних фреймворків для розробки desktop-додатків.

Для побудови віконного додатку, що може бути легко перенесений до різних ОС(Windows,Linux), було прийняте рішення використати фреймворк Qt, його IDE(Integrated development environment) Qt Creator, бібліотеки FFTW та FFmpeg, компілятор MinGW та qmake.

3.2.2 Фреймворк Qt

Qt – фреймворк для розробки кроссплатформенного програмного забезпечення мовою програмування C++.

Qt дозволяє запускати написане з його допомогою програмне забезпечення у більшості сучасних операційних систем шляхом простої компіляції програми для кожної системи без зміни вихідного коду. Включає всі основні класи, які можуть знадобитися при розробці прикладного програмного забезпечення, починаючи від елементів графічного інтерфейсу і закінчуючи класами для роботи з мережею, базами даних і XML. Є повністю об'єктно-орієнтованим, таким, що розширюється і підтримує техніку компонентного програмування.

Відмінна риса - використання метаоб'єктного компілятора – попередньої системи обробки вихідного коду. Розширення можливостей забезпечується системою плагінів, які можна розміщувати безпосередньо на панелі візуального редактора. Також існує можливість розширення звичної функціональності віджетів, пов'язаної з розміщенням їх на екрані, відображенням, перемальовуванням при зміні розмірів вікна.

Комплектується візуальним середовищем розробки графічного інтерфейсу Qt Designer, що дозволяє створювати діалоги та форм. У поставці Qt є Qt Linguist - графічна утиліта, що дозволяє спростити локалізацію та переклад програми на багато мов; і Qt Assistant – довідкова система Qt, що спрощує роботу з документацією по бібліотеці, а також дозволяє створювати кроссплатформну довідку для програмного забезпечення, що розробляється на основі Qt.

Починаючи з версії 4.5.0, в комплект включено середовище розробки Qt Creator, яке включає редактор коду, довідку, графічні засоби Qt Designer і можливість налагодження додатків. Qt Creator може використовувати GCC або Microsoft VC++ як компілятор і GDB як відладчик. Для Windows-версій бібліотека комплектується компілятором, заголовками та об'єктними

файлами MinGW.

3.2.3 Бібліотека FFmpeg

FFmpeg — це безкоштовний програмний проект із відкритим вихідним кодом, який складається з набору бібліотек і програм для обробки відео, аудіо та інших мультимедійних файлів і потоків. Його ядром є сам інструмент командного рядка FFmpeg, призначений для обробки відео та аудіо-файлів. Він широко використовується для перекодування формату, базового редагування (обрізання та об'єднання), масштабування відео, ефектів пост-продакшну відео та відповідності стандартам (SMPTE, ITU).

FFmpeg також містить інші інструменти: `ffplay`, простий медіаплеєр і `ffprobe`, інструмент командного рядка для відображення медіаінформації. Серед включених бібліотек є `libavcodec`, бібліотека аудіо/відеокодеків, яка використовується багатьма комерційними та безкоштовними програмними продуктами, `libavformat` (Lavf), бібліотека мультиплексування та демультіплексування контейнерів аудіо/відео, і `libavfilter`, бібліотека для вдосконалення та редагування фільтрів через Gstreamer- як `filtergraph`.

FFmpeg є частиною робочого процесу багатьох інших проектів програмного забезпечення, а його бібліотеки є основною частиною програмних медіаплеєрів, таких як VLC, і включено до основної обробки для YouTube і Bilibili. Кодери та декодери для багатьох форматів аудіо- та відеофайлів є включено, що робить його дуже корисним для перекодування поширених і незвичайних медіафайлів.

FFmpeg публікується під ліцензією LGPL-2.1-або пізнішою або GPL-2.0-або пізнішою, залежно від того, які параметри ввімкнено.

3.2.4 Бібліотека FFTW

The Fastest Fourier Transform in the West (FFTW) — це бібліотека

програмного забезпечення для обчислення дискретних перетворень Фур'є (ДПФ), розроблена Маттео Фріго та Стівеном Г. Джонсоном з Массачусетського технологічного інституту.

FFTW є одним із найшвидших безкоштовних програмних реалізацій швидкого перетворення Фур'є (ШПФ). Він реалізує алгоритм ШПФ для дійсних і комплекснозначних масивів довільного розміру та розмірності. FFTW оперативно перетворює дані, підтримуючи різноманітні алгоритми та вибираючи один (конкретне розкладання перетворення на менші перетворення), який він оцінює або вимірює як кращий за конкретних обставин. Він найкраще працює з масивами розмірів із малими простими множниками, причому степені двійки є оптимальними, а великі прості числа є найгіршим випадком (але все одно $O(n \log n)$).

Щоб розкласти перетворення складених розмірів на менші перетворення, він вибирає один із кількох варіантів алгоритму ШПФ Кулі-Тьюкі (що відповідає різним факторизаціям і/або різним шаблонам доступу до пам'яті), тоді як для простих розмірів він використовує алгоритм ШПФ Рейдера або Блуштейна.

Після того як перетворення було розбито на підперетворення достатньо малих розмірів, FFTW використовує жорстко закодовані розгорнуті ШПФ для цих невеликих розмірів, які були створені (під час компіляції, а не під час виконання) генерацією коду; ці підпрограми використовують різноманітні алгоритми, включаючи варіанти Кулі-Тьюкі, алгоритм Рейдера та алгоритми ШПФ з простим множником.

Для достатньо великої кількості повторюваних перетворень корисно вимірювати продуктивність деяких або всіх підтримуваних алгоритмів на заданому розмірі масиву та платформі. Ці вимірювання, які автори називають «мудрістю», можна зберегти у файлі або рядку для подальшого використання.

FFTW має «інтерфейс гуру», який має на меті «розкрити якомога більше гнучкості базової архітектури FFTW». Це дозволяє, серед іншого,

багатовимірні перетворення та численні перетворення за один виклик (наприклад, коли дані чергуються в пам'яті).

FFTW має обмежену підтримку для перетворень поза порядком (з використанням версії інтерфейсу передачі повідомлень (MPI)). Перевпорядкування даних викликає накладні витрати, яких нетривіально уникнути для перетворень на місці довільного розміру. Задokumentовано невідомо, для яких трансформацій ці накладні витрати є значними.

FFTW ліцензовано відповідно до GNU General Public License. Він також комерційно ліцензований (за вартістю до 12 500 доларів США) MIT і використовується в комерційному пакеті матриць MATLAB для розрахунку ШПФ. FFTW написано мовою C, але існують інтерфейси Fortran і Ada, а також інтерфейси для кількох інших мов. Хоча сама бібліотека є C, код фактично генерується з програми під назвою «genfft», яка написана на OCaml.

У 1999 році FFTW отримав премію Дж. Х. Вілкінсона за програмне забезпечення для чисельних програм.

3.3 Особливості роботи перетвореннями із бібліотеки FFTW

3.3.1 Одновимірні ШПФ реальних даних

У багатьох практичних застосуваннях вхідні дані $in[i]$ є суто дійсними числами, і в цьому випадку вихідні дані ДПФ задовольняють «ермітівську» надлишковість: $out[i]$ є сполученим $out[n-i]$. Можна скористатися цими обставинами, щоб досягти приблизно подвійного покращення як швидкості, так і використання пам'яті.

В обмін на ці переваги у швидкості та просторі користувач жертвує частиною простоти складних перетворень FFTW. По-перше, вхідні та вихідні масиви мають різні розміри та типи: вхід – n дійсних чисел, а вихід – $n/2+1$ комплексних чисел (ненадлишкові виходи); це також вимагає невеликого

«доповнення» вхідного масиву для перетворень на місці. По-друге, зворотне перетворення (від комплексного до дійсного) має побічний ефект перезапису вхідного масиву за замовчуванням. Жодна з цих незручностей не повинна становити серйозної проблеми для користувачів, але важливо знати про них.

Процедури для виконання перетворень реальних даних майже такі ж, як і для складних перетворень: виділяються масиви `double` та/або `fftw_complex` (бажано використовуючи `fftw_malloc` або `fftw_alloc_complex`), створюється `fftw_plan`, виконується скільки завгодно разів за допомогою `fftw_execute(plan)` і очищається за допомогою `fftw_destroy_plan(plan)` (і `fftw_free`). Єдина відмінність полягає в тому, що вхід (або вихід) має тип `double` і існують нові підпрограми для створення плану.

```
fftw_plan fftw_plan_dft_r2c_1d(int n, double *in, fftw_complex *out,  
                               unsigned flags);  
fftw_plan fftw_plan_dft_c2r_1d(int n, fftw_complex *in, double *out,  
                               unsigned flags);
```

Рисунок 3.1 – Функції плану для повного ШПФ

Для перетворень реального входу в комплексно-ермітівський вихід (r2c) і комплексно-ермітівський вхід у реальний вихід (c2r). На відміну від складного планувальника ДПФ, тут немає знакового аргументу. Натомість r2c DFT завжди `FFTW_FORWARD`, а c2r DFT завжди `FFTW_BACKWARD`. (Для одинарної/довгої подвійної точності `fftwf` і `fftwl double` слід замінити на `float` і `long double` відповідно.)

Тут n – «логічний» розмір ДПФ, не обов'язково фізичний розмір масиву. Зокрема, дійсний (подвійний) масив має n елементів, тоді як складний (`fftw_complex`) масив має $n/2+1$ елементів (де ділення округлено в меншу сторону). Для перетворення на місці вхід і вихід прив'язуються до одного масиву, який має бути достатньо великим, щоб вмістити обидва; отже, реальний масив насправді матиме $2*(n/2+1)$ елементів, де елементи поза першим n є невикористаним доповненням. (Зауважте, що це дуже

відрізняється від концепції «нульового доповнення» перетворення до більшої довжини, яке змінює логічний розмір ДПФ шляхом фактичного додавання нових вхідних даних.) k -й елемент комплексного масиву точно такий же, як як k -й елемент відповідного комплексного ДПФ. Усі позитивні n підтримуються; добутки малих множників є найбільш ефективними, але алгоритм $O(n \log n)$ використовується навіть для простих розмірів.

Як зазначалося вище, перетворення $c2r$ знищує свій вхідний масив навіть для перетворень, які не є на місці. Цьому можна запобігти, якщо необхідно, включивши `FFTW_PRESERVE_INPUT` до прапорців, на жаль, з деякими жертвами в продуктивності. Цей прапор також наразі не підтримується для багатовимірних реальних ДПФ.

Пам'ятаймо, що 0 -й («DC») і $n/2$ -й (частота «Найквіста», коли n є парним) елементи комплексного результату є суто реальними. Тому деякі реалізації зберігають елемент Найквіста там, де буде розміщена уявна частина DC, щоб зробити вхідний і вихідний масиви однакового розміру. Таке пакування, однак, погано узагальнює багатовимірні перетворення, і економія місця в будь-якому випадку незначна; FFTW не підтримує.

Альтернативний інтерфейс для одновимірних DFT $r2c$ і $c2r$ можна знайти в інтерфейсі « $r2r$ » (див. ДПФ у напівкомплексному форматі), з виводом у форматі «напівкомплексного», який має той самий розмір (і тип), що й вхідний масив. Цей інтерфейс, хоча він і не дуже корисний для багатовимірних перетворень, іноді може забезпечувати кращу продуктивність.

3.5.2 ШПФ у напівкомплексному форматі

Вид $r2r$ `FFTW_R2HC` ($r2hc$) відповідає $r2c$ DFT (див. Одновимірні DFT реальних даних), але з виводом у «напівскладному» форматі, і іноді може бути швидшим та/або зручнішим, ніж останній.

Зворотне перетворення $hc2r$ має вигляд `FFTW_HC2R`. Це складається з

ненадлишкової половини комплексного виведення для 1d реального вхідного ДПФ розміром n , що зберігається як послідовність n дійсних чисел (double) у форматі:

$$r_0, r_1, r_2, \dots, r_{n/2}, i_{(n+1)/2-1}, \dots, i_2, i_1$$

Рисунок 3.2 – Розміщення в пам'яті вихідної послідовності

Тут r_k – дійсна частина k -го виходу, а i_k – уявна частина. (Ділення на 2 округлюється в меншу сторону.) Для напівкомплексного масиву $hc[n]$ k -й компонент, таким чином, має дійсну частину в $hc[k]$, а уявну частину — в $hc[n - k]$, за винятком $k = 0$ або $n/2$ (останнє, лише якщо n парне) – у цих двох випадках уявна частина дорівнює нулю через симетрію дійсного вхідного ДПФ і не зберігається. Таким чином, перетворення $r2hc$ n дійсних значень є напівкомплексним масивом довжини n , і навпаки для $hc2r$.

Окрім іншого формату, вихід FFTW_R2HC/FFTW_HC2R в іншому випадку точно такий же, як і для відповідного 1d перетворення $r2c/c2r$ (тобто перетворення FFTW_FORWARD/FFTW_BACKWARD, відповідно). Пам'ятайте, що ці перетворення є ненормалізованими, тому $r2hc$, за яким слідує $hc2r$, призведе до вихідних даних, помножених на n . Крім того, подібно до перетворення $c2r$, перетворення $hc2r$, що не є на місці, знищить його вхідний масив.

Хоча ці напівкомплексні перетворення можна використовувати з багатовимірним інтерфейсом $r2r$, інтерпретація такого роздільного продукту перетворень уздовж кожного виміру проблематична.

Наприклад, розглянемо двовимірне перетворення n_0 на n_1 , $r2hc$ на $r2hc$, заплановане за допомогою `fftw_plan_r2r_2d(n0, n1, in, out, FFTW_R2HC, FFTW_R2HC, FFTW_MEASURE)`. Концептуально, FFTW спочатку перетворює рядки (розміру n_1), щоб отримати напівкомплексні рядки, а

потім перетворює стовпці (розміру n_0). Половина цих перетворень стовпців, однак, складається з уявних частин, і тому їх слід помножити на i та поєднати з перетвореннями r_2 реальних стовпців, щоб отримати $2d$ амплітуди ДПФ; Перетворення r_2 FFTW не виконує цю комбінацію.

4 РОЗРОБКА ПРИКЛАДНОЇ ПРОГРАМИ

4.1 Інтерфейс програми

Спочатку треба підключити необхідні модулі Qt для цього в `qmake(.pro)` файлі додати модулі Qt, що будуть потрібні для розробки.

Лістинг 4.1 – Модулі Qt

```
QT += core gui printsupport multimedia
```

Ці модулі дозволяють створити графічний інтерфейс, а також малювати на графіках в реальному часі та працювати за аудіо-файлами та іншими джерелами аудіо даних.

Так як за основу була взята об'єктно орієнтовна мова C++ структура програми поділена на класи. У цьому розділі будуть розглянуті класи, що відповідають лише за інтерфейс прикладної програми.

Кінцевий інтерфейс програми буде мати такий вигляд:

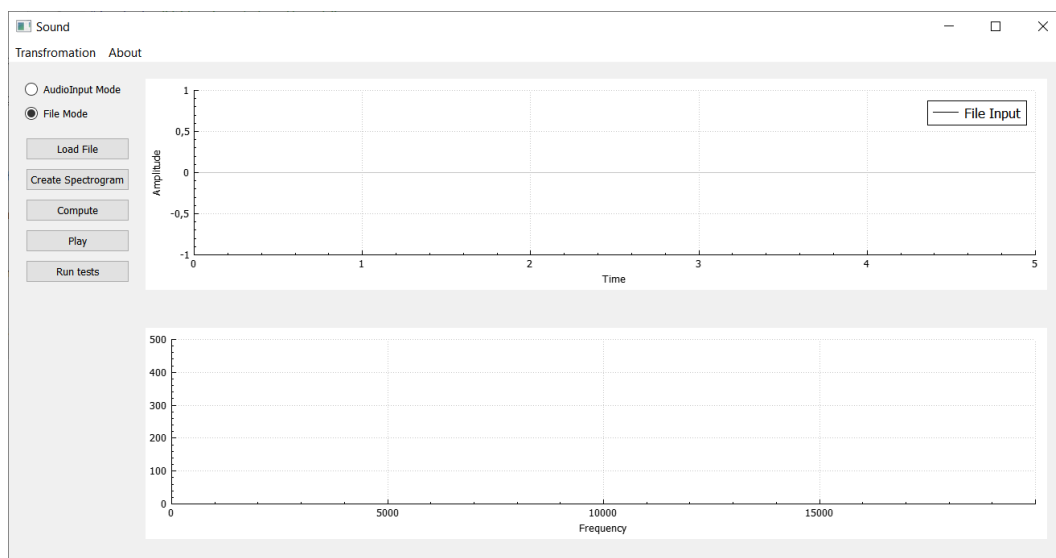


Рисунок 4.1 – Кінцевий вигляд інтерфейсу програми

Головний клас вікна приведено на лістингу 4.2

Лістинг 4.2 – Клас MainWindow

```
class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private slots:
    void onUseButtonPressed( const QAudioDeviceInfo& dev );
    void initControls( bool checked );
    void createSpectrogram( const QString& filename );
    void setFFT();
    void setFFThc();
    void setFHT();
    void openHelpWindow();

private:
    Ui::MainWindow *ui;

    Spectrogram::SpekSpectrogram* m_specWidget = nullptr;
    AudioInputHandler m_audioInput;
    AudioInputControls m_audioInputContr;
    FileControls m_fileControls;
    FileInputHandler m_fileHandler;
    QAction* m_naiveDft;
    QAction* m_FFT;
    QAction* m_FFThc;
    QAction* m_FHT;
    QAction* m_openEfficiencyWindow;
    QAction* m_openHelpWindow;
};
```

На цьому лістингу можна побачити, які класи були використанні. Клас `MainWindow` має такі приватні члени класу, як `SpekSpectrogram` – це є клас

віджету, що відповідає за малювання спектрограми wav файлу, `AudioInputHandler` – клас для роботи з сигналом із зовнішніх мікрофону, `AudioInputControls` – клас віджету для відображення елементів управління для роботи з зовнішніми мікрофонами, `FileControls` – клас віджету для відображення управління елементів для роботи з wav аудіо файлів, `FileInputHandler` – клас для отримання сигналу з wav аудіо файлів та декілька `QAction` для подій у програмі.

За допомогою об'єкту `ui` та графічного конструктору вбудованого в Qt Creator були побудовані графіки амплітуди та частоти, меню, два елемента `RadioButton`.

Однак в цієї роботі більше уваги приділено саме перетворенням сигналу звуку, тому відкриття файлу, малювання та побудову інтерфейсу не буде розглянуте у цієї роботі зокрема заготовочного класу `MainWindow` наведеного вище.

4.2 Перетворення у прикладні програми

4.2.1 Файл `transformation.cpp`.

Лістинг 4.3 – `include` директиви для `transformation.cpp`

```
#include "transformations.h"  
#include "fftw/fftw3.h"  
#include <QDebug>  
#include <cmath>  
#include <chrono>
```

У лістингу 4.3 підключаються бібліотеки необхідні для побудови функцій розкладення до спектру. Для перетворень Фур'є та Хартлі буде використана бібліотека FFTW та її заголовний файл `fftw3.h`, для виведення до консолі налагодження використовується `QDebug`, для побудови віконної функції та нормалізації виведення даних до вихідної послідовності

використовується `cmath`, а для оцінки скільки перетворення зайняло часу використовуються функціонал `chrono`.

Лістинг 4.3 – Повне ШПФ

```
QPair<double, int> FFTc( QVector<double> inVec, QVector<double>&
outVec, bool log = false )
{
    int nfft = inVec.size();
    QPair<double,int> returnValue;
    returnValue.second = 2*nfft;
    double* fftIn = fftw_alloc_real( nfft );
    fftw_complex* fftOut = fftw_alloc_complex( nfft );
    fftw_plan fftPlan = fftw_plan_dft_r2c_1d( nfft, fftIn,
fftOut, FFTW_ESTIMATE );
    auto start = std::chrono::steady_clock::now();
    memcpy( fftIn, inVec.data(), nfft*sizeof(double) );
    hamming( fftIn, nfft );
    fftw_execute( fftPlan );
    outVec.clear();
    countSpectrum( fftOut, outVec, nfft, log );
    auto end = std::chrono::steady_clock::now();
    std::chrono::duration<double> elapsed_seconds = end-
start;
    returnValue.first = elapsed_seconds.count();
    fftw_free(fftIn);
    fftw_free(fftOut);
    fftw_destroy_plan(fftPlan);
    return returnValue;
}
```

На лістингу 4.3 показана функція повного перетворення Фур'є і розкладання перетвореної послідовності до частотного спектру, яку використовує всі класи окрім `SpekSpectrogram`.

На вхід функції подається послідовність типу `double`, посилання на виділене місце в пам'яті для вихідної послідовності, та зміна типу `bool`, що вказує на необхідність логарифмування вихідної послідовності. Вихідне значення функції має змістовні частини ефективності перетворення(пам'ять та швидкість).

Для ШПФ використовуються план `fftw_plan_dft_r2c_1d` та для нормалізації відображеного сигналу використовуються функція вікна Вікно Хеммінга вказана на лістингу 4.4.

Лістинг 4.4 – Вікно Хеммінга

```
void hamming( QVector<double> &signal )
{
    int N = signal.size();
    for (int i = 0; i < N; i++)
        signal[i] *= ( 0.54 - 0.46 * cos( 2*PI * i/N) );
}
```

Виконується копіювання вхідної послідовності, так як використовуючи план `fftw_plan_dft_r2c_1d` бібліотеки `FFTW` вхідну послідовність буде зруйновано. Після виконання перетворення за допомогою `fftw_execute` треба розкласти отриману послідовність до частотного спектру. Це робиться за допомогою функції `countSpectrum`, що наведена в лістингу 4.5.

Лістинг 4.5 – Функція для рахування спектру

```
void countSpectrum( fftw_complex* input, QVector<double>& spec,
int size, bool log )
{
    for( int i = 1; i < ( size +1 ) / 2; ++i )
    {
        double enrValue = input[i][0]*input[i][0] +
input[i][1] * input[i][1];
        if( log )
            spec.append( 10 * std::log( enrValue ) );
        else
            spec.append( sqrt( enrValue ) );
    }
}
```

В залежності від значення змінної `log` буде виконано перетворення до `dB`(децибел) за допомогою логарифма або буде записана величина частотного спектру цієї послідовності. За допомогою `chrono` отримуємо час на перетворення.

ШПХ та напів-комплексне ШПФ виглядає аналогічно на лістингу 4.6 та 4.7 відповідно.

Лістинг 4.6 – Функція перетворення ШПХ та розкладання до частотного спектру

```

QPair<double,int> FHT_func( QVector<double> inVec,
QVector<double>& outVec, bool log = false )
{
    int nfft = inVec.size();
    QPair<double,int> returnValue;
    returnValue.second = nfft;
    double* fftIn = fftw_alloc_real( nfft );
    double* fftOut = fftw_alloc_real( nfft );
    fftw_plan fftPlan = fftw_plan_r2r_1d( nfft, fftIn,
fftOut, FFTW_DHT, FFTW_ESTIMATE );
    auto start = std::chrono::steady_clock::now();
    memcpy( fftIn, inVec.data(), nfft*sizeof(double) );
    hamming( fftIn, nfft );
    fftw_execute( fftPlan );
    outVec.clear();
    countSpectrum( fftOut, outVec, nfft, log );
    auto end = std::chrono::steady_clock::now();
    std::chrono::duration<double> elapsed_seconds = end-
start;
    returnValue.first = elapsed_seconds.count();
    fftw_free(fftIn);
    fftw_free(fftOut);
    fftw_destroy_plan(fftPlan);
    return returnValue;
}

```

Лістинг 4.7 – Функція напів-комплексного ШПФ та розкладання до частотного спектру

```

QPair<double,int> FFThc( const QVector<double>& inVec,
QVector<double>& outVec, bool log = false )
{
    int nfft = inVec.size();
    QPair<double,int> returnValue;

```

```

        returnValue.second = nfft;
        double* fftIn  = fftw_alloc_real( nfft );
        double* fftOut = fftw_alloc_real( nfft );
        fftw_plan fftPlan = fftw_plan_r2r_1d( nfft, fftIn,
fftOut, FFTW_R2HC, FFTW_ESTIMATE );
        auto start = std::chrono::steady_clock::now();
        memcpy( fftIn, inVec.data(), nfft*sizeof(double) );
        hamming( fftIn, nfft );
        fftw_execute( fftPlan );
        outVec.clear();
        countSpectrum( fftOut, outVec, nfft, log );
        auto end = std::chrono::steady_clock::now();
        std::chrono::duration<double> elapsed_seconds = end-
start;

        returnValue.first = elapsed_seconds.count();
        qDebug() << "Duration" << elapsed_seconds.count();
        fftw_free( fftIn );
        fftw_free( fftOut );
        fftw_destroy_plan(fftPlan);
        return returnValue;
    }

```

Для спектрограми використовуються схожий принцип, але були використані інструменти бібліотеки FFMPEG. Більш детально ознайомитись з кодом можна у додатку А.

4.2.2 Файл dfttest.cpp

Лістинг 4.8 – include директиви для dfttest.cpp

```

#include "dfttest.h"
#include <QVector>
#include <chrono>
#include "fftw/fftw3.h"

```

Для файлу `dfttest.cpp`, як і для файлу трансформацій знадобляться бібліотека `FFTW` для перетворень та `chrono` для оцінки швидкості перетворення у секундах. Але так як тут тестується тільки пряме та зворотне перетворення без роботи з перетвореною послідовність заголовков `smath` тут не потрібен. В тестах використовується вихідний формат, який вказаний у лістингу 4.9.

Лістинг 4.9 – Вихідний формат даних тестів

```
struct TestResult
{
    int sampleSize;
    int memoryUsage;
    double timeOfTransform;
    double timeOfReverseTransform;
    double fullTime;
};
```

У цьому форматі змінна `sampleSize` є розміром у байтах вхідної послідовності, `memoryUsage` – розмір вихідної послідовності, `timeOfTransform` – час витрачений на пряме перетворення, `timeOfReverseTransform` – час витрачений на зворотне перетворення, `fullTime` – час на пряме та зворотнє перетворення разом.

Лістинг 4.10 – Тест для ШПФ

```
TestResult DFTTest::runFTTtest()
{
    TestResult r;
    QVector<double> samples( m_file.samples[0].begin(),
m_file.samples[0].end() );
    r.sampleSize = samples.size();
    r.memoryUsage = samples.size() * 2;
    int nfft = samples.size();

    auto start = std::chrono::steady_clock::now();
    fftw_complex* complexV = fftw_alloc_complex( nfft );
```

```

    fftw_plan fftPlan = fftw_plan_dft_r2c_1d( nfft,
samples.data(), complexV, FFTW_ESTIMATE );
    fftw_execute( fftPlan );
    auto end = std::chrono::steady_clock::now();
    std::chrono::duration<double> elapsed_seconds = end-start;
    r.timeOfTransform = elapsed_seconds.count();

    start = std::chrono::steady_clock::now();
    double* realV = fftw_alloc_real( nfft );
    fftw_plan rFFtPlan = fftw_plan_dft_c2r_1d( nfft, complexV,
realV, FFTW_ESTIMATE );
    fftw_execute( fftPlan );
    end = std::chrono::steady_clock::now();
    elapsed_seconds = end-start;
    r.timeOfReverseTransfrom = elapsed_seconds.count();

    r.fullTime = r.timeOfReverseTransfrom + r.timeOfTransform;

    fftw_free(complexV);
    fftw_free(realV);
    fftw_destroy_plan(fftPlan);
    fftw_destroy_plan(rFFtPlan);
    return r;
}

```

У лістингу 4.10 представлений тест для ШПФ та зворотного ШПФ. За допомогою функцій отримання часу та функцій реалізацій ШПФ із бібліотеки FFTW отримуємо швидкість виконання перетворення в секундах. Аналогічний алгоритм використовується для тестування ШПХ та напів-комплексного ШПФ.

5 ОГЛЯД ПІДСУМКОВОЇ ПРИКЛАДНОЇ ПРОГРАМИ ДЛЯ АНАЛІЗУ ЕФЕКТИВНОСТІ ПЕРЕТВОРЕНЬ ШПФ ТА ШПХ ДЛЯ РОБОТИ ЗІ ЗВУКОМ

5.1 Огляд прикладної програми та її функціоналу

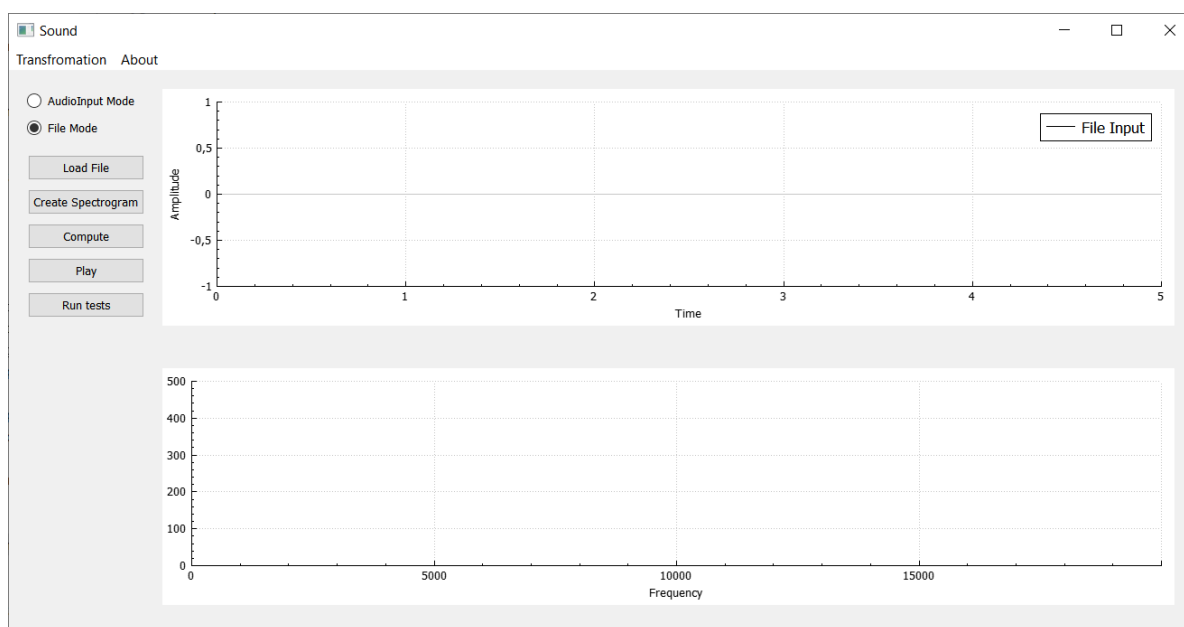


Рисунок 5.1 – Вигляд програми при першому включенні

Програма працює у двох режимах аудіо режим та файловий режим, що переключаються за допомогою двох кнопок AudioInput Mode та File Mode відповідно. За замовчуванням програма відкривається у файловому режимі.

5.1.1 Файловий режим

У файловому режимі, щоб скористатися програмою потрібно загрузити файл який буде використовуватись для всього набору функцій прикладної програми. Коли користувач натискає на кнопку Load File буде відкрите стандартне вікно пошуку у вашій ОС.

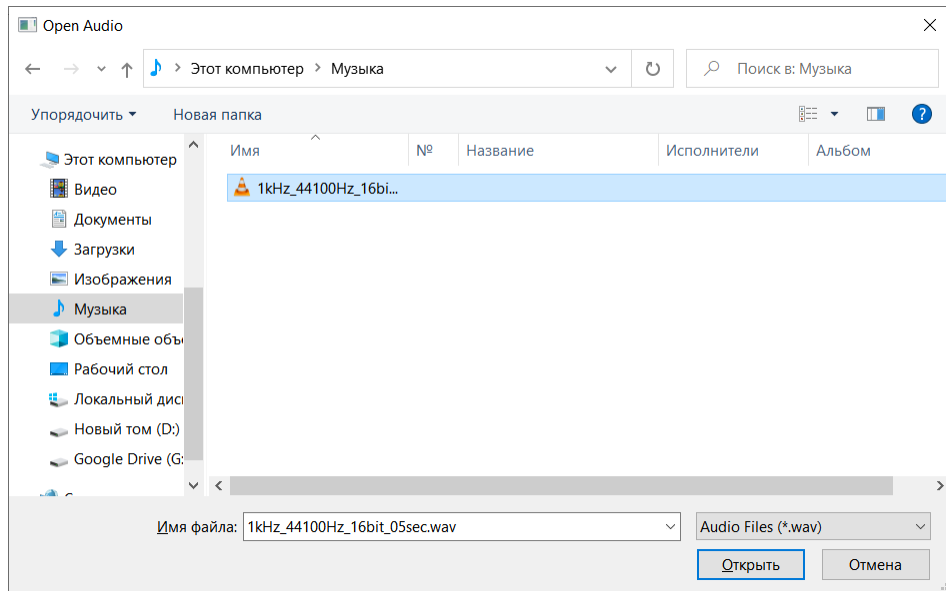


Рисунок 5.2 – Стандартне вікно пошуку для Windows

Недоліком цієї програми у файловому режимі є сприйняття тільки файли з розширенням wav. Успішно вибравши файл у користувача є декілька функцій на вибір :

- обчислити спектр та амплітуду та побудувати їх на графіку(кнопка Compute);
- намалювати спектрограму(кнопка Create Spectrogram);
- програти звукозапис(кнопка Play);
- запустити тестові прямі і зворотні перетворення для цього файлу та вивести результати на екран(кнопка Run tests).

Графіки можна масштабувати та здвигати використовуючи колесо миші та кнопки миші. Використовуючи простий тон з частотою 10kHz тривалістю 5 секунд протестуємо програму.

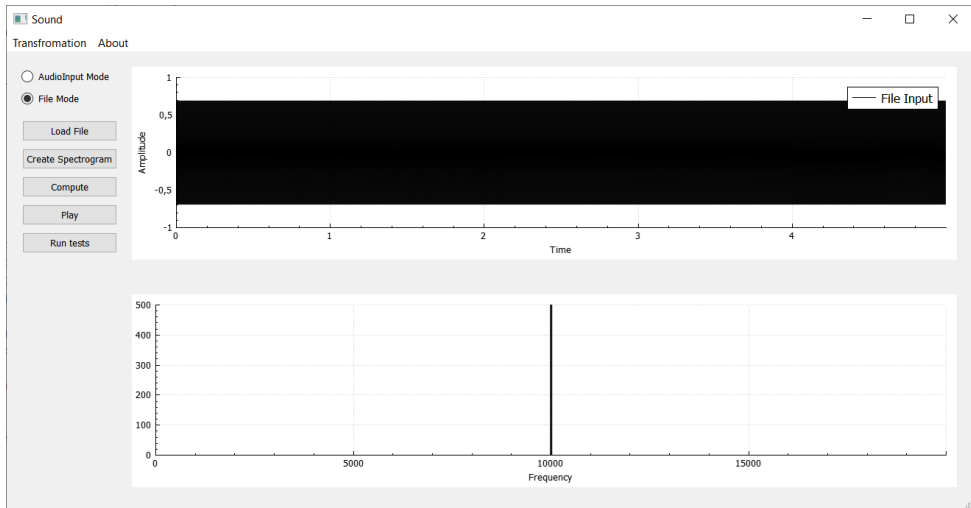


Рисунок 5.3 – Результати обчислень частоти та амплітуди після натискання кнопки Compute

	Sample Size(bytes)	Used Memory(bytes)	FowardTransform(s)	BackwardTransform(s)	TotalTime(s)
FFT	220500	441000	0.0142323	0.0034951	0.0177274
FFT _{hc}	220500	220500	0.004825	0.0025651	0.0073901
FHT	220500	220500	0.0049069	0.002765	0.0076719

Рисунок 5.4 – Результати тестів щодо перетворень ШПХ,ШПФ та напівкомплексного ШПФ після натискання кнопки Run Tests

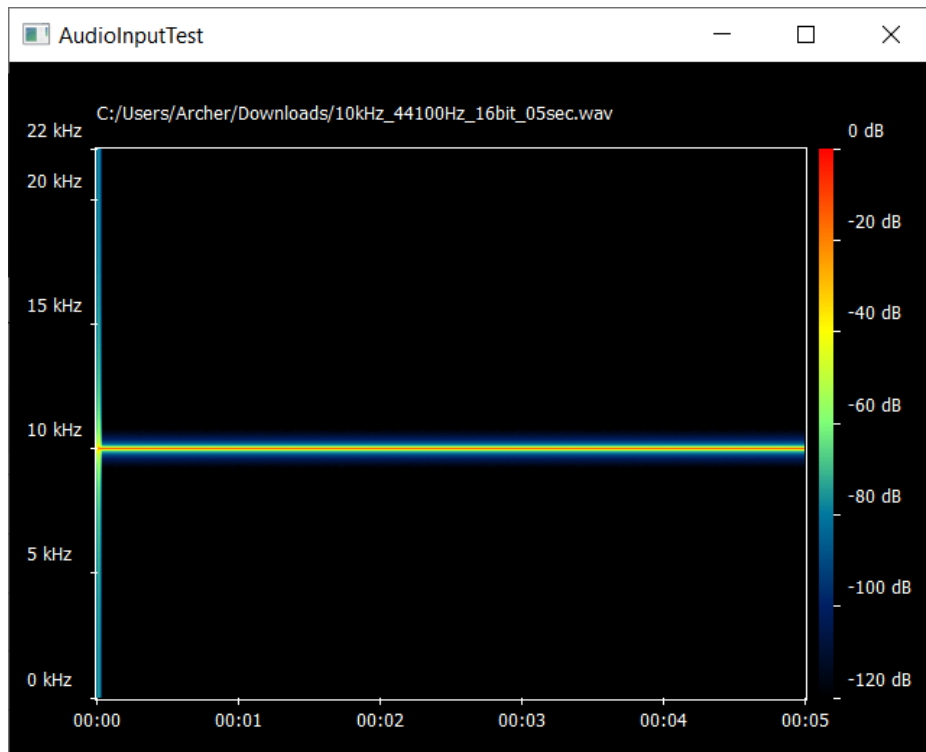


Рисунок 5.5 – Спектрограма для простого тону 10kHz після натискання кнопки Create Spectrogram

Аналогічно протестуємо програму використовуючи музикальний аудіо файл тривалістю в 16 секунди

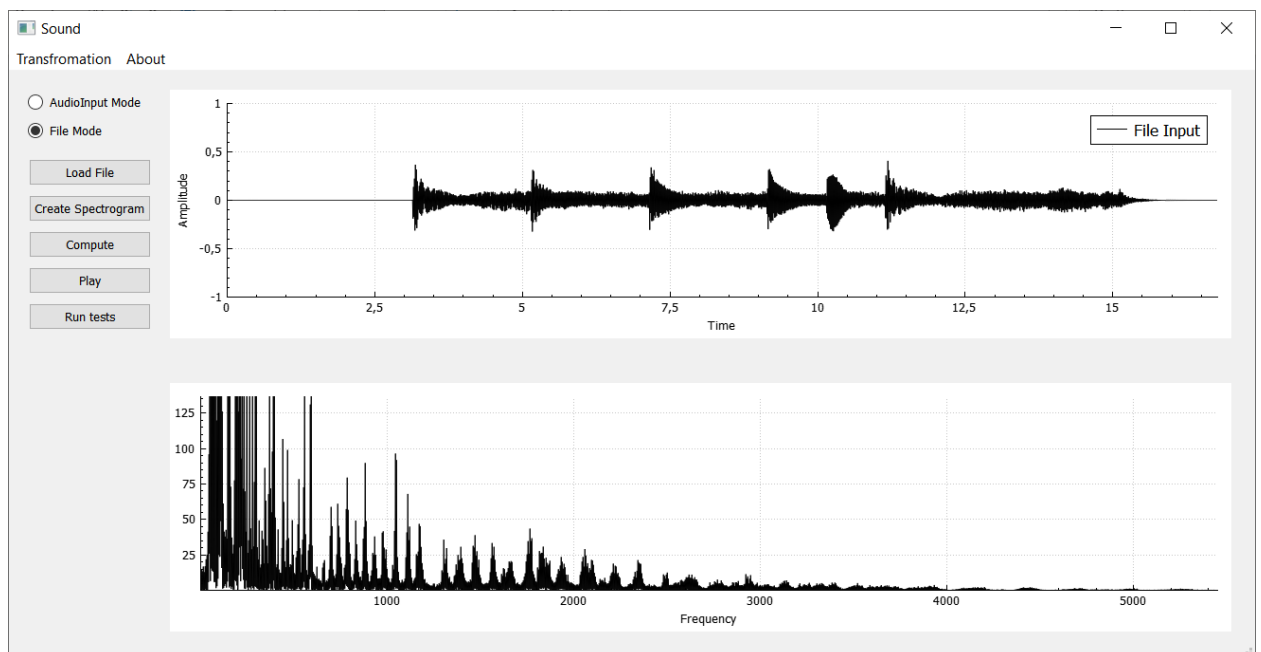


Рисунок 5.6 – Результати обчислень частоти та амплітуди після натискання кнопки Compute

	Sample Size(bytes)	Used Memory(bytes)	FowardTransform(s)	BackwardTransform(s)	TotalTime(s)
FFT	402624	805248	0.0315363	0.0308569	0.0623932
FFTnc	402624	402624	0.241385	0.191029	0.432415
FHT	402624	402624	0.231065	0.210156	0.441221

Рисунок 5.7 – Результати тестів щодо перетворень ШПХ,ШПФ та напів-омплексного ШПФ після натискання кнопки Run Tests

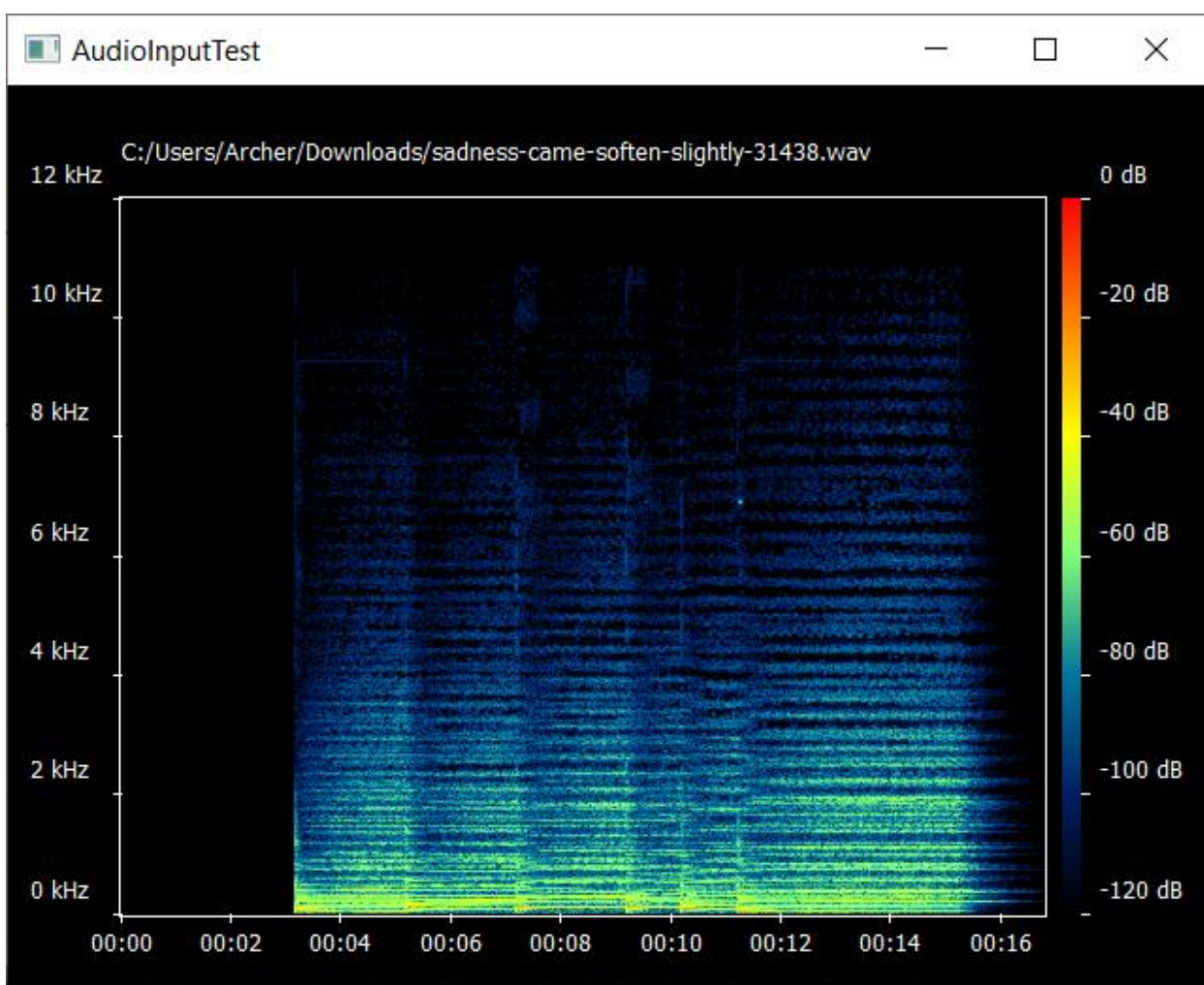


Рисунок 5.8 – Спектрограма для складного тону гітарної струни після натискання кнопки Create Spectrogram

Використовуючи меню Transformation вибрав різні види перетворень, особу увагу приділив результатам тестів можна сказати, що найбільш ефективним перетворення для дискретизованого сигналу звуку, що представлений у вигляді одномірної послідовності дійсних чисел, є напів-комплексне швидке перетворення Фур'є та перетворення Хартлі, так як різницею, яка була представлена у рамках тестів, можна знехтувати.

5.1.2 Audio Input Mode

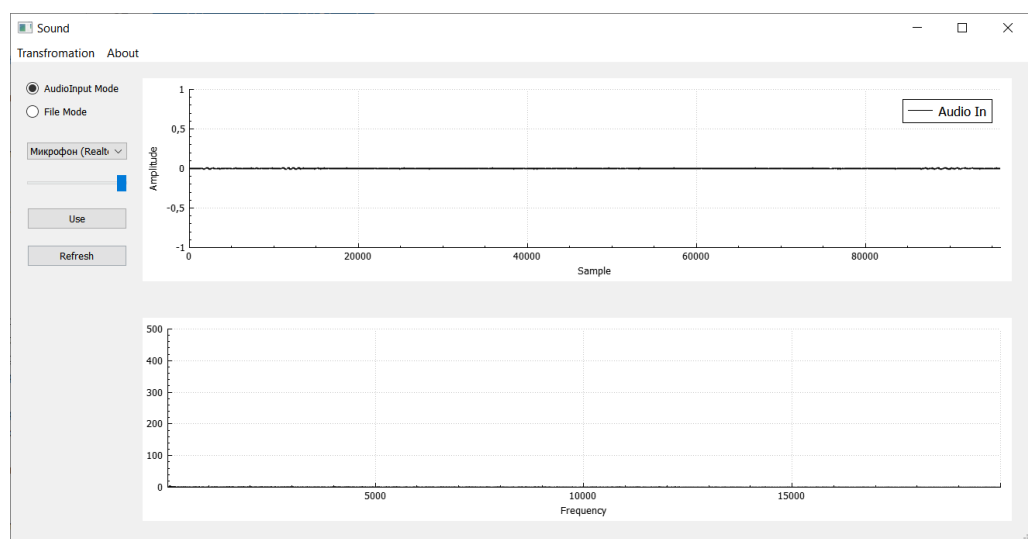


Рисунок 5.9 – Програма у режимі Audio Input Mode

У цьому режимі інформація поступає до програми не з музичного файлу, а зі зовнішніх мікрофонів. В лівій верхній частині програми є випадаючий список з усіма представленими зовнішніми мікрофонами, що під'єднані до системи. Якщо бажаний мікрофон не було знайдено треба перевірити його підключення та натиснути кнопку Refresh. Вибравши бажаний мікрофон та натиснувши кнопку Use програма почне зчитувати звук з мікрофона та відобразити його частотну та амплітудну характеристику на графіку.

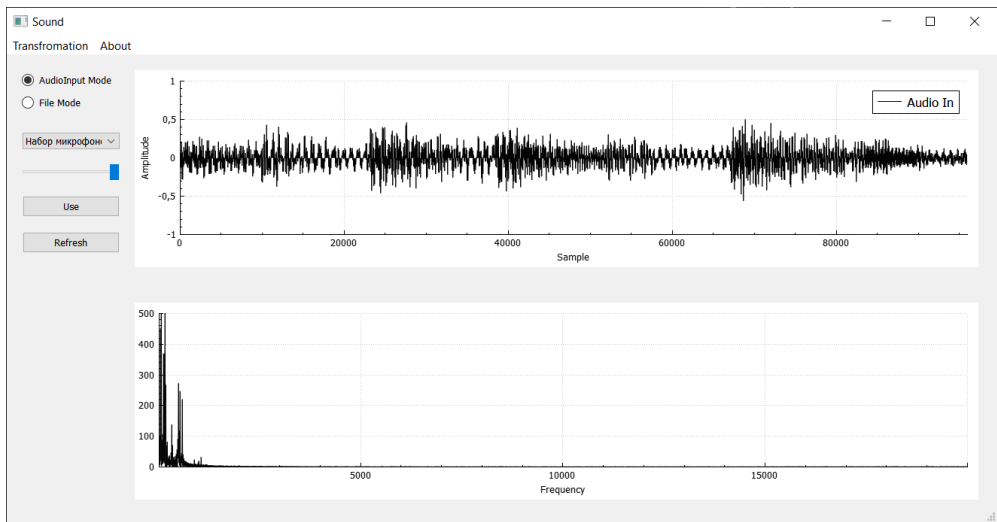


Рисунок 5.10 – Відображення частотної та амплітудної характеристики у реальному часі

Використовуючи повзунок у лівій частині інтерфейсу можна регулювати гучність звуку. Також можна вибрати бажане перетворення у меню Transformation зверху.

ВИСНОВКИ

У роботі розглянуті особливості роботи зі звуком, методи аналізу звуку. Були розглянуті основні методи розкладання звуку до спектру та проаналізовані такі перетворення: перетворення Фур'є, дискретне перетворення Фур'є, перетворення Хартлі, дискретне перетворення Хартлі, та алгоритм швидкого перетворення Фур'є, алгоритм швидкого перетворення Хартлі та їх порівняння.

Була розроблене середовище для аналізу ефективності швидких перетворень у контексті роботи зі звуком в одному просторі. Середовище аналізу було представлено у вигляді прикладної програми.

Програма має такі можливості, як малювання амплітудно-часового графіку та графіку спектру для аудіо файлу з розширенням wav, малювання спектрограми, розкладення звуку з мікрофонів до амплітудного та частотного графіку у реальному часі, формування таблиць з такими даними, як швидкість перетворення в секундах та пам'ять, яка була використана для перетворення.

За підсумками роботи було винесено вирок, що ШПХ дійсно ефективніше ніж не спеціалізоване швидке перетворення Фур'є(ШПФ) для одного виміру, але використовуючи спеціалізоване швидке перетворення для одного виміру, напівкомплексне швидке перетворення Фур'є, різниця між ними є незначною.

Для послідовностей дійсних чисел в одному вимірі найбільш ефективно використовувати або спеціалізоване напівкомплексне ШПФ або ШПХ.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Звуковой диапазон частот [Электронний ресурс] / studfile.net. – Режим доступу: www / URL: <https://studfile.net/preview/1640603/page:5> – 08.06.2021 г. – Загол. з екрану.
2. Пономарева, О. В. Быстрое параметрическое дискретное преобразование Фурье действительных последовательностей [Текст] / О. В. Пономарева // Цифровая обработка сигналов. – 2012. – №. 2. – С. 2–5.
3. Солонина, А. И. Алгоритмы и процессоры цифровой обработки сигналов [Текст] / А. И. Солонина, Д. А. Улахович, Л. А. Яковлев. – СПб. : БХВ Петербург, 2002. – 464 с.
4. Аверченко, А. П. Преобразование Фурье и преобразование Хартли [Текст]/ А. П. Аверченко, В. К. Воропаев, Б. Д. Женатов // Технические науки в России и за рубежом : материалы III Междунар. науч. конф. (г. Москва, июль 2014 г.). — 2014. — С22-24.
5. Qt Documentation [Электроний ресурс] / doc.qt.io/qt-5.15 – Режим доступу: www / URL: <https://doc.qt.io/qt-5.15> – 15.12.2022 г. – Загл. с экрана.
6. Harris, Fredric J. "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform" [Текст] / Harris, Fredric J. – М. : IEEE, 1978. – 94 с.
7. Randall, R.V., and Tech, B. "Frequency Analysis, 3rd Edition", [Текст] / Randall, R.V., and Tech, B. – М. : Bruël and Kjør, 1979. –168 с.
8. FFTW Documentation [Электроний ресурс] / fftw.org/fftw3_doc/– Режим доступу: www / URL: https://www.fftw.org/fftw3_doc/ – 15.12.2022 г. – Загл. с экрана.