**ЗМІСТ**

[Перелік скорочень, умовних позначень, термінів 4](#_Toc453071282)

[ВСТУП 5](#_Toc453071283)

[1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ІДЕНТИФІКАЦІЇ ОБ’ЄКТІВ ЗА ЙОГО ЗВУКОВИМ ПОРТРЕТОМ 6](#_Toc453071284)

[1.1. Задача ідентифікації звуку 6](#_Toc453071285)

[1.2. Частотний метод 9](#_Toc453071286)

[1.3. Метод нейронних мереж 10](#_Toc453071287)

[1.4. Метод фільтрації рівня гучності 11](#_Toc453071288)

[1.5. Направлене прослуховування ефіру 12](#_Toc453071289)

[1.6. Фрактальний та мультифрактальний методи 13](#_Toc453071290)

[1.7. Вейвлет-перетворення 15](#_Toc453071291)

[1.8. Перетворення Фур’є 18](#_Toc453071292)

[1.9. Метод аудіо-відбитків 19](#_Toc453071293)

[Висновки 22](#_Toc453071294)

[2. ОПИС МАТЕМАТИЧНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ЗАДАЧІ АНАЛІЗУ ХАРАКТЕРИСТИК СИГНАЛУ 23](#_Toc453071295)

[2.1. Пояснення вибору метода та його предметна галузь 23](#_Toc453071296)

[2.2. Віконне перетворення Фур’є 26](#_Toc453071297)

[2.3. Вейвлет-перетворення 29](#_Toc453071298)

[2.4. Числова характеристика форми графіків сигналів 33](#_Toc453071299)

[Висновки 35](#_Toc453071300)

[3. ОПИС ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ 36](#_Toc453071301)

[3.1. Обґрунтування вибору мови програмування Python 36](#_Toc453071302)

[3.2. Принцип та алгоритм роботи програми 37](#_Toc453071303)

[3.3. Структура програми 38](#_Toc453071304)

[3.4. Використання бібліотеки pywt 40](#_Toc453071305)

[3.5. Організація користувацького інтерфейсу ПЗ 41](#_Toc453071306)

[Висновки 42](#_Toc453071307)

[4. ПЕРЕВІРКА ПРАВИЛЬНОСТІ ІДЕНТИФІКАЦІЇ ОБ’ЄКТІВ 43](#_Toc453071308)

[4.1. Опис умов тестування 43](#_Toc453071309)

[4.2. Результати тестів 44](#_Toc453071310)

[4.3. Аналіз результатів тестування 54](#_Toc453071311)

[4.4. Висновки 55](#_Toc453071312)

[ВИСНОВОК 56](#_Toc453071313)

[СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ 57](#_Toc453071314)

**ДОДАТКИ**

**Додаток 1. Копії графічних матеріалів**

* ІАЛЦ.045420.005 Д1. Алгоритм завантаження файлів. Блок-схема
* ІАЛЦ.045420.006 Д2. Алгоритм програми. Блок-схема
* ІАЛЦ.045420.007 Д3. Структура програми. Структурна схема
* ІАЛЦ.045420.008 Д4. Блоки перетворення сигналу. Структурна схема

**Додаток 2. Фрагменти програмного коду**

# Перелік скорочень, умовних позначень, термінів

АП – акустичний портрет.

ВП – вейвлет перетворення.

ІКЗ – інтенсивні короткочасні звуки.

КМА – кратномасштабний аналіз.

ПЗ – програмне забезпечення.

ПФ – перетворення Фур’є.

ФР – фрактальна розмірність.

Pylab – засоби графічного подання даних, створені Python

Python – мова програмування високого рівня.

# ВСТУП

Завдання ідентифікації об’єктів на основі акустичного портрету зводиться до спектрального аналізу звуку. Задача ідентифікації звуків найчастіше зустрічається у випадках розпізнавання голосу та виявлення звуків немовного походження. Такі програми знаходять широке застосування, наприклад: у промисловості – детектування аварійних випадків при пошкодженні ємностей з хімічними речовинами; у космонавтиці – при відстикуванні модулів; у системах устаткуванні домів – для контролю за відкриттям дверей, вікон, спрацьовуванню кондиціонера; у військовій сфері – для ідентифікації дронів та стеженням за ворогом. Сьогодні існує декілька підходів до вирішення даної задачі, вони відрізняються швидкістю обробки даних, залежністю від інших методів та потребою у початковому налаштуванні системи сенсорів.

Існує ряд методів, які допомагають вирішити поставлену задачу розпізнавання звуків, проте відповідно до вхідних умов придатними залишаються лише декілька з них.

Для простого і обґрунтованого подання рішення задачі розпізнавання звуку широко використовується перетворення та операції з математичного аналізу, що приводять масиви даних випадкової величини у зручний вигляд, що спрощує їх подальше обчислення.

Правильний вибір підходу до вирішення завдання та математичних засобів дозволяє успішно подати розв’язок у вигляді програмних структур даних та алгоритмів, що автоматизують процес обчислення. Процес вибору програмних засобів впливає на швидкість та якість розробки. Наявність існуючої бібліотеки обраних програмних засобів може значно спростити складність поставленої задачі.

# АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ІДЕНТИФІКАЦІЇ ОБ’ЄКТІВ ЗА ЇХ ЗВУКОВИМ ПОРТРЕТОМ

## Задача ідентифікації звуку

Задача ідентифікації об’єктів різного походження зустрічається у наступних варіантах:

* розпізнавання людського голосу;
* фільтрування шумів і виділення людського голосу;
* розпізнавання звуків немовного походження.

Типовий випадок розпізнавання звуків немовного походження є задача ідентифікації інтенсивних короткочасних звуків – звукових коливань тривалістю менше 0.5 секунди, гучність яких значно вища (на 15 і більше дБ), ніж фонові звуки. Прикладами таких звуків є вистріл, плескіт і політ дрона.

Програми, що ідентифіковують звуки, мають широке застосування, наприклад:

* у промисловості – визначення плескоту, вибухів для миттєвого запуску системи безпеки у випадку аварії чи пошкодження ємностей із хімічними небезпечними речовинами,
* у космонавтиці – для своєчасної реакції операторів щодо змін у роботі устаткування,
* на випробувальних полігонах чи у стрілецьких тирах – для автоматизації роботи з контрольно-вимірювальним устаткуванням,
* в охоронних системах – для подання сигналу про порушення захисту,
* у системах “Розумний дім” – як для розпізнавання керуючих дій, так і з ціллю отримання інформації про події, що потребують уваги, наприклад про відкрите від вітру вікно.

Оскільки предмет дипломної роботи зводиться до дослідження випроміненого звуку, необхідно володіти базовими даними щодо його цифрового подання.

Цифровим поданням звуку є масив чисел, що містить послідовність потужності звуку в кожен момент часу.

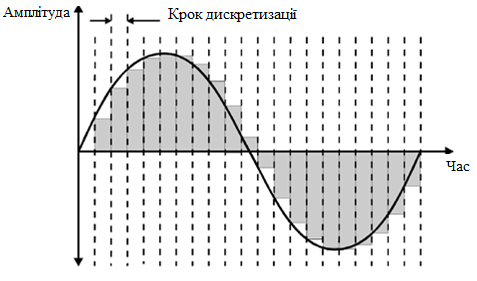


Рисунок 1.1 Цифрове подання звукових даних

Значення у цифровому масиві відповідають визначеному проміжку часу – кроку дискретизації. Важливо врахувати, що для запису висоти кожного блока використовують дискретний набір значень, тому вони не завжди точно співпадають із числовою мірою амплітуди хвилі, що може призвести до виникнення похибки.

Таким чином на практиці звук представлений набором значень його потужності у заданому часовому проміжку із обраним кроком дискретизації. Детальна форма запису і збереження заданих наборів значень залежить від використаного аудіо формату.

Вирішення задачі ідентифікації об’єктів на основі звуку має проходити в режимі реального часу. Це передбачає багаторазове повторення трьох операцій: запису порції звукових коливань, обробки отриманих значень і виведення результату. Для зменшення часу аналізу даних необхідний аудіо формат, що дозволить максимально швидко оброблювати дані.

Формат представлення звукових даних у цифровому вигляді залежить від способу квантування у аналого-цифровому перетворювачі.

Виділяють наступні групи аудіо форматів:

* аудіо формати без стиску (wav, aiff);
* аудіо формати із стиском без втрат (ape, flac);
* аудіо формати із стиском та втратами (mp3, ogg).

Формати, що використовують стиск, призначені для зберігання і відтворення звукових даних, які слухає людина, слух якої не бездоганний.

Для швидкого і порівняно точного аналізу формати із стиском, і тим більше з втратами, непридатні. Наявність втрат неприйнятна, оскільки це зменшує точність отриманих результатів, а необхідність розпаковування даних збільшує час їх обробки. Відповідно, варто використовувати формат без стиску.

Аналіз показав, що всю необхідну інформацію у зручному для обробки вигляді містить формат WAVE (wav).

Аудіофайл WAVE складається з двох блоків. Перший блок – інформаційний заголовок файлу, в якому зберігається така інформація:

* розмір файлу,
* кількість каналів,
* частота дискретизації,
* глибина звуку (bitrate).

Другий блок складається із даних, що зберігають цифровий сигнал – набір значень амплітуд.

Розпізнавання об’єктів відбувається у повітряному середовищі, тому на точність розпізнавання впливають наступні фактори:

1. температура повітря,
2. вологість повітря,
3. атмосферний тиск,
4. відстань між сенсором та джерелом звуку,
5. геометричні параметри простору (приміщення чи вулиці).

Важливим критерієм при проектуванні системи, що ідентифікуватиме об’єкти на основі акустичного портрету, є потреба у початковому налаштуванні мережі сенсорів, що реалізовуватимуть обраний метод. Як відомо, початкове налаштування мережі може тривати дуже довго, що не завжди задовільно у критичних умовах. Крім того, необхідно створити правильні умови, в яких система би “навчалась”. Згідно з заданим критерієм методи можна поділити на дві групи: методи, що потребують налаштування, та методи, що здійснюють обчислення без початкового налаштування.

Для вибору способу ідентифікації об’єктів на основі акустичного портрету необхідно проаналізувати кожен із вже існуючих методів.

## Частотний метод

Частотний метод не потребує початкового налаштування або “навчання” системи. Частотний метод передбачає перехід із часової (рисунок 1.1) форми представлення звуку в частотну, де кожному значенню частоти відповідає деяка величина потужності звуку. В частотній області для кожного звуку значення потужності унікальні, що дозволяє із високою ефективністю застосувати його для задачі розпізнавання інтенсивних короткочасних звуків. Критерієм для порівняння є набір значень потужності звуку у заданих інтервалах частот. Але використання даного методу вимагає додатковий етап обчислень, тому використання лише даного методу для аналізу звуку в режимі реального часу не може бути застосовано.

## Метод нейронних мереж

Розпізнавання інтенсивних короткочасних звуків на основі нейронних мереж – самостійний метод, що потребує “навчання” системи. При використанні даного методу вирішується задача розпізнавання образів по шаблону. При цьому кожен нейрон (див. рисунок 1.2) є еталонним значенням залежності потужності звуку від часу.

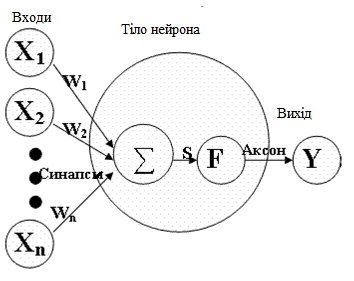


Рисунок 1.2 Схема роботи нейромереж

Синоптичні зв’язки характеризуються вагами wi, на вхід яких подаються значення рівня гучності звуку на деякому інтервалі. Поточний стан S нейрона дорівнює зваженій сумі входів.



Функція S далі перетворюється активаційною функцією F і дає вихідний сигнал Y нейрона. Найбільше розповсюдженою у нейронних мережах активаційною функцією є логістична функція, що обчислюється так:



Головною рисою нейронних мереж є можливість їх початкового “навчання”, що суттєво збільшує імовірність правильного розпізнавання.

З точки зору розпізнавання інтенсивних короткочасних звуків необхідність у налаштуванні нейронної мережі і, відповідно, складання бази шаблонів – недолік, оскільки без початкових даних немає можливості швидко ввести устаткування в експлуатацію.

Недоліки застосування методів ідентифікації об’єктів змушують шукати інші способи вирішення цієї задачі.

## Метод фільтрації рівня гучності

Метод фільтрації рівня гучності інтенсивних короткочасних звуків – несамостійний метод, що не потребує налаштування системи. Форма вхідних сигналів значно простіша, ніж форма хвиль людського голосу, тому для розпізнавання сигналів можна задіяти метод, що базується на різниці рівня гучності сигналів і фону.

Для виділення звуків задається фіксоване значення гучності звука L і зміщення ∆L у відсотках від L (див. рисунок 1.3). Отримані значення L+∆L і L-∆L є верхньою і нижньою межею чутливості мікрофона відповідно. Таким чином, потрібні звуки попадають у діапазон і розпізнаються, а шум відсікається.

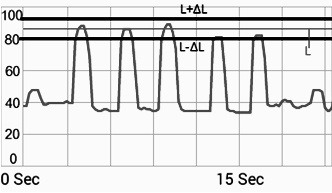


Рисунок 1.3 Діапазон звуків при фільтрації

Зміщення ∆L відносно граничного значення, треба визначати експериментально залежно від природи звуку.

Спосіб дуже простий у реалізації, але для високої надійності розпізнавання потребує точного визначення зміщення ∆L, оскільки навіть у одного й того ж джерела звуку рівень гучності може суттєво коливатись. Сенсор може зреагувати на інші джерела звуку, які мають схожі характеристики, або відфільтрувати необхідний звук, оскільки він не попав у заданий інтервал. Але, з іншого боку, можна припустити, що при правильному визначенні зміщення і при врахуванні отриманої залежності гучності звуку від відстані, даний спосіб буде високоефективним при розпізнаванні в умовах шумів чи при великій кількості джерел шуму із схожою природою на коротких відстанях.

Даний метод може працювати у парі як з частотним методом, так із методом нейронних мереж, що суттєво пришвидшить обчислення у режимі реального часу.

## Направлене прослуховування ефіру

Щоб підвищити надійність розпізнавання інтенсивних короткочасних звуків можна використовувати направлене прослуховування ефіру двома мікрофонами. При цьому простір ділиться на n рівних частин, наприклад 8. Використовуючи дані, отримані з двох мікрофонних пристроїв, визначається сектор, в якому знаходиться джерело звуку. Дані з решти напрямків ігноруються.

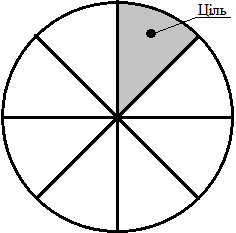


Рисунок 1.4 Розташування джерела звуку у секторі

Даний метод може не давати високої точності, оскільки результати залежать від параметрів задіяних пристроїв, що визначають:

* якість звукозапису,
* програмний доступ двох мікрофонів.

Залежно від добротності запису звуку, отриманого від мікрофона, може неправильно визначатися положення джерела звука у просторі (попадання в інший сектор). Додатково, можливе отримання сигналу від інших об’єктів, що знаходяться у тій області простору, де розміщене необхідне джерело звуку. До того ж, не завжди є можливість для програмної (для поставленої задачі) реалізації, оскільки не кожен пристрій має 2 програмно доступних мікрофони.

Даний метод є допоміжним і лише полегшує задачу розпізнавання звуків у режимі реального часу, зменшуючи навантаження процесора.

## Фрактальний та мультифрактальний методи

Основою поняття фрактального метода є фрактальна розмірність, яка є невід’ємним дійсним числом, та відображає геометричну складність аналізованої структури (сигналу, зображення і т.д.). Різноманітна складність об’єктів спостереження призводить до різних величин вхідного сигналу. Дана властивість надає можливість використовувати поняття фрактальної розмірності для аналізу сигналу. Алгоритми фрактального аналізу достатньо прості і ефективні у реалізації.

Існує багато способів розрахунку розмірності, але всі вони містять обчислення об’єму, площі чи довжини фрактальної форми і того, як вони змінюються при масштабуванні. Найбільш використовуваною є розмірність Хаусдорфа-Безиковича із використанням способу покриття, що буде оперуватися при аналізі величини фрактальної розмірності сигналу.

Спосіб покриття ФР за допомогою квадратів містить такі кроки:

1. Задається значення , часова область вхідного сигналу розкладається на квадрати зі стороною ε і підраховується, скільки квадратів покривають всі локальні екстремуми.

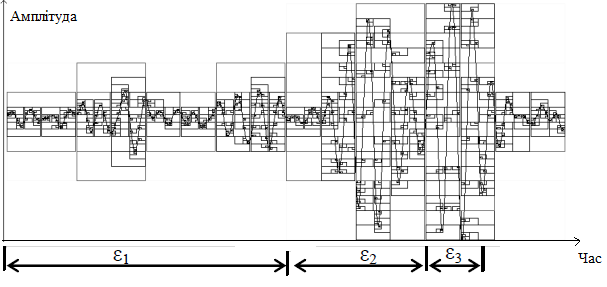


Рисунок 1.5 Фрактальний метод аналізу сигналів

1. Припустимо, що були проведені обчислення N(ε) для різних довжин сторони ε (на рис. 1.5 значення ε1, ε2= ε1/2, ε3= ε1/4). На основі отриманих даних будується залежність log N(ε) від log 1/ ε (рис. 1.6).

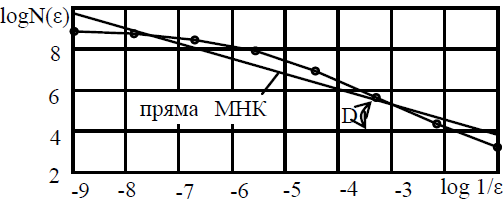


Рисунок 1.6 Побудова прямої методом МНК

1. Оцінка ФР на основі отриманих значень зводиться до пошуку “найбільш лінійної” ділянки залежності. Припустимо, що були проведені обчислення N(ε) для різних довжин сторони ε (на рис. 1.5 значення ε1, ε2= ε1/2, ε3= ε1/4). На основі отриманих даних будується залежність log N(ε) від log(1/ε) (рис. 1.6); до побудови на цій ділянці лінійної апроксимації вигляду log N(ε) = -b × log(ε) + С, наприклад, методом найменших квадратів (МНК); як оцінку фрактальної розмірності беремо D = -b, тобто кут нахилу прямої МНК.

Ідея мультифрактального аналізу сигналів полягає у розкладанні заданої множини із складною статистикою по множинам однорідних фракталів із чітко виділеною фрактальною розмірністю.

## Вейвлет-перетворення

Вейвлет-перетворення може послужити “магічним склом”, дозволяючи побачити локальні короткочасні особливості сигналу, які неможливо виявити при його дослідженні класичними методами. Прикладами є збій у індикаторах сенсорів, нерівність поверхні та специфічні звукові характеристики в акустичному спектрі.

Розглянемо синусоїдний сигнал, що має невелике локальне викривлення (виділено квадратом на рисунку 1.7). На спектрограмі сигналу описана особливість не спостерігається – помітна лише гармонічна складова. Проте на діаграмі коефіцієнтів деталізації ця особливість виражена дуже наочно.

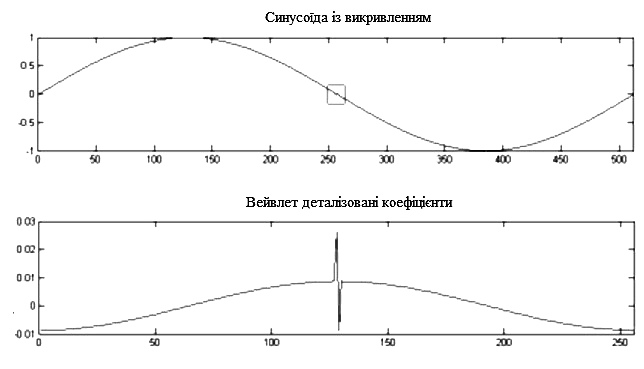


Рисунок 1.7 Вейвлет-перетворення

Інший приклад – порушення неперервності другої похідної сигналу (рисунок 1.7). Графік цієї функції виглядає як гладка крива, але насправді складається з двох експоненціальних компонентів, що з’єднуються у точці t = 500. Неперервність порушується лише у другій похідній, але і цей момент також помітний на графіку при вейвлет-розкладанні.

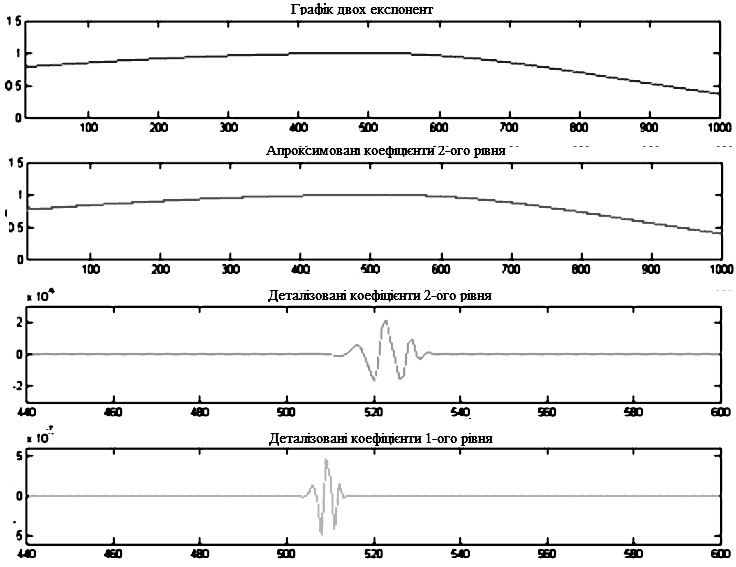


Рисунок 1.8 Коефіцієнти ВП

При дослідженні та обробці сигналу багато корисної інформації можна отримати із аналізу його частотних характеристик. Оскільки вейвлети володіють хорошою частотно-часовою адаптацією, вони могли б послужити зручним інструментом для дослідження частотних характеристик сигналу.

Крім цього, інтуїтивно зрозумілий зв’язок між масштабом вейвлет-перетворення і його частотною областю: чим більший масштаб, тим більш низькі частоти аналізуються. Однак, для практичного використання вейвлетів у спектральному аналізі сигналів необхідно встановити взаємозв’язок між масштабом вейвлета та його частотними характеристиками.

Вейвлети є вузькочастотними фільтрами і можуть виділяти із сигналу складові, частота яких відповідає їх піку (максимальному значенню на заданому проміжку). Ця частота називається центральною частотою вейвлета.

Приведемо приклад. Візьмемо сім синусоїд та дослідимо їх за допомогою вейвлета. Частоту першої синусоїди візьмемо рівною центральній частоті задіяного вейвлета. А частоти решти синусоїд зменшуватимемо вдвічі послідовно. Застосуємо алгоритм швидкого вейвлет-перетворення. Результат перетворення показано на рисунку 1.9.

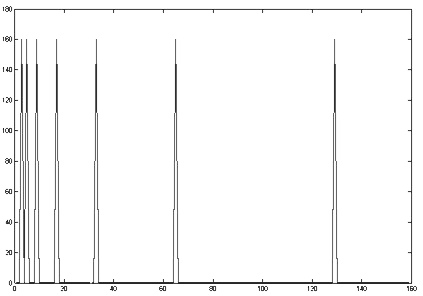


Рисунок 1.9 Частотно-часова характеристика сигналу

## Перетворення Фур’є

При ідентифікації періодичних (квазіперіодичних) і хаотичних сигналів часто використовується процедура перетворення Фур’є. Сигнал, що має період повторення, називається стаціонарним; сигнал, що є випадковим та не повторюється - нестаціонарним. ПФ, на відміну від вейвлет-перетворення, може бути застосований лише для стаціонарних сигналів.

Функція F(ω) називається перетворенням Фур’є або спектральною щільністю сигналу f(t), де ω – є показником, що відповідає частоті складової сигналу. Функцію частоти F(ω) називають енергетичним спектром стаціонарного випадкового сигналу. Цей спектр дає лише загальну картину розподілу енергії по частоті елементарних гармонічних складових, але не враховує її фазової структури.

Перетворення Фур’є періодичної функції f(t) на інтервалі [0,T], що описує форму сигналу, використовує як базисні функції синус і косинус, представлені експонентою:

При отриманні фазового портрету, задача ідентифікації об’єкту зводиться до аналізу амплітуд при відповідних значеннях частот.

Даний метод є зручним у випадках, коли є характерні значення частоти досліджуваного об’єкта без врахування часових даних на певних частотах, і може використовуватися разом із частотним методом.

## Метод аудіо-відбитків

Даний метод є самостійним та широко використовується у додатках щодо розпізнавання музичних файлів.

Цей спосіб ідентифікації звуку використовує швидке перетворення Фур’є для формування спектрограми на визначеному дискретному проміжку часу. Після того їх об’єднують та формують повний спектральний портрет звукового файлу. Водночас зберігають значення амплітуди у кожному проміжку часу. Як підсумок – отримано двовимірний масив із амплітудою, що є функцією від часу та частоти. Важливо пам’ятати, що значення частоти та часу – цілі, проте амплітуди – дійсні. Вигляд спектрограми показано на рисунку 1.10.

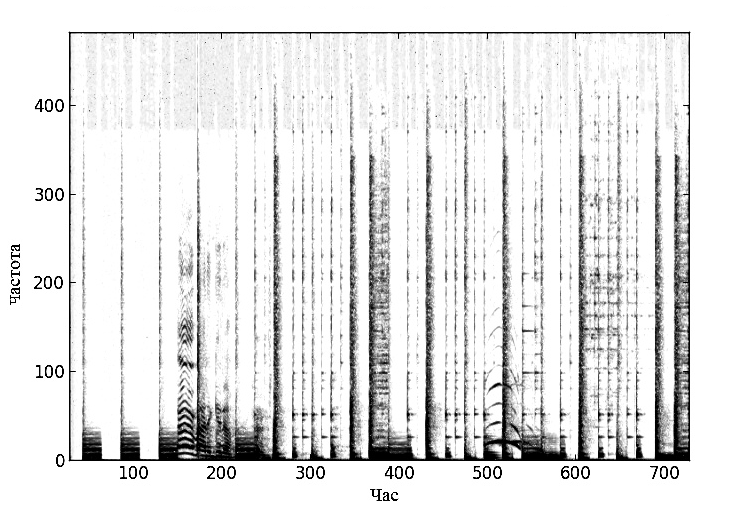


Рисунок 1.10 Спектрограма mp3-файлу

На рисунку 1.10 можна помітити характерний сплеск амплітуди уздовж всієї горизонтальної осі координат.

Базовою для вирішення задачі розпізнавання звуку є ідея пікових значень амплітуди. Кожне значення піку є парою значення час і частота, що відповідає найбільшому значенню амплітуди у локальній області, тобто є локальним максимумом. Ширина області визначається експериментально та задається константним способом.

Після отримання пікових значень покажемо їх на рисунку 1.11, сумістивши із виглядом спектрограми.

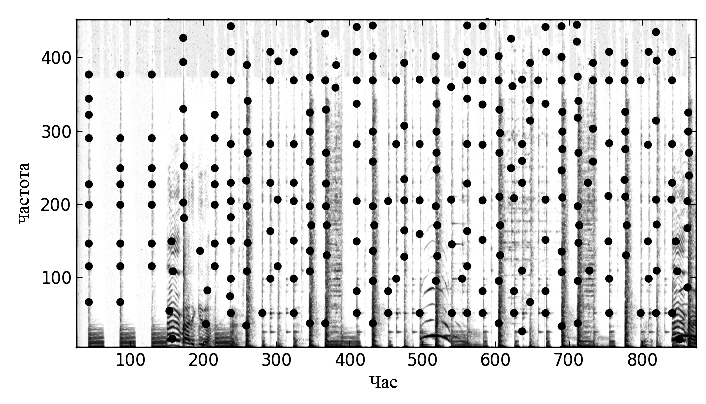


Рисунок 1.11 Пікові значення на спектрограмі

Для спрощення операції порівняння значення із зразком введемо поняття хеш-функції, що повертає унікальне значення на основі двох і більше параметрів.

У даному випадку, хеш-значення аудіо-відбитку буде рівне хеш-функції від частоти пікових значень та різниці часу між ними.

Далі відбувається запам’ятовування аудіо-відбитків.

При запуску програми, що реалізує приведений метод, здійснюється порівняння хеш-значень, і чим більше виявилося збігів, тим більша імовірність ідентифікації того ж звукового файлу. Даний метод також ідентифікує схожість нецілих та відрізаних файлів. З іншого боку, він потребує тієї ж послідовності потужності вхідного музичного файлу, що і зразку, що може виявитись проблемою при розпізнаванні короткочасних випадкових звуків.

## 

## Висновки

У цьому розділі проаналізовані всі можливі підходи до вирішення задачі ідентифікації об’єктів на основі АП.

Умовно їх можна розділити на самостійні методи, що є цілісними та не потребують додаткових обчислень, але можуть бути поєднанням інших способів, та на несамостійні, які є допоміжними в обчисленнях та не можуть самотужки вирішити задачу розпізнавання звуку.

Також важливим є критерій “навчання систем”. Процес налаштування та початкового введення даних у систему може займати тривалий час (до кількох місяців). Тому при вирішенні задачі предметної галузі потрібно зважати на фактор часу, який може стати визначальним для методу нейронних мереж. З іншого боку, такі методи доволі прості у реалізації та володіють кращою точністю щодо виявлення збігів із зразком.

Ще одним критерієм при обранні методу є розрахунок обчислювальної складності алгоритму, що реалізовуватиме заданий метод. Оскільки всі обчислення відбуваються у режимі реального часу, що означає безперервне надходження даних для їх аналізу, то у випадку недостатніх ресурсів (частота процесора, об’єм оперативної пам’яті) масиви значень не будуть вчасно обробленими, тобто будуть знищеними.

Правильне вирішення задачі містить враховує вибір формату даних, який не може мати втрат і стиску, які б ускладнювали задачу обробки даних.

# ОПИС МАТЕМАТИЧНИХ ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ЗАДАЧІ АНАЛІЗУ ХАРАКТЕРИСТИК СИГНАЛУ

## Пояснення вибору метода та його предметна галузь

Щоб обрати правильний метод для рішення задачі розпізнавання об’єктів на основі АП, необхідно проаналізувати його предметну галузь застосування.

Як предметну галузь було обрано звуки пострілів із гвинтівки, пістолету чи револьвера. Такий вибір пояснюється його короткочасністю, легким виявленням людським слухом, аби перевірити правильність роботи системи. Крім того, постріл має характерний звуковий спектр, який можна аналізувати. Також виявлення звуку пострілу є актуальним у сучасний період, адже розмістивши сенсори на певній території, можна проаналізувати: чи ведуться бойові дії поблизу. Іншим застосуванням є контроль за браконьєрством у лісах та заповідниках, де заборонене знищення природних ресурсів.

Важливо відзначити, що аналіз звукового спектру має здійснюватися без попередньої обробки у таких ПЗ, як SoundBooth чи SoundHack. Система має працювати автономно, отримуючи як вхідні дані цифровий сигнал, обчислений за допомогою сенсорів. Такі умови безпосередньо впливають на точність обчислення.

Частотний метод, навіть як допоміжний, не може бути задіяним для вирішення поставленого завдання, адже як вхідні дані можуть бути цифрові сигнали, отримані із різноманітних сенсорів, модель яких визначає запис частотних характеристик звукових хвиль. До того ж умови зчитування звуку враховують випадкову відстань та взаємного розташування щодо сенсорів. Якщо запис відбуватиметься у замкнутому приміщенні, де яскраво помітний ефект еха, то відбиті хвилі сильно вплинуть на частотну характеристику сигналу, що змінить його вигляд відносно зразка.

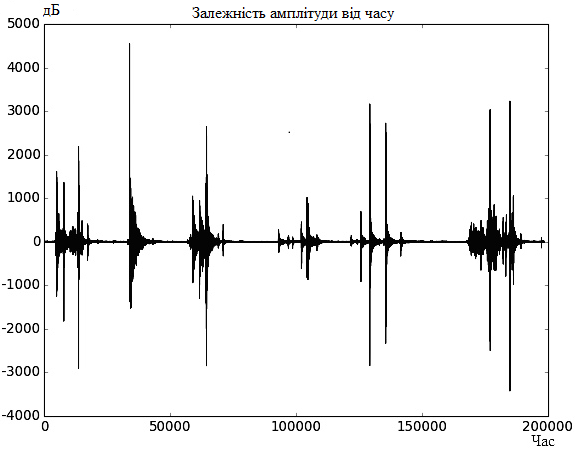


Рисунок 2.1 Амплітудна характеристика вхідного сигналу

На рисунку 2.1 за допомогою програмних засобів Pylab показано графік залежності амплітуди від частоти певного вхідного сигналу. На рисунку 2.2 зображена частотна характеристика цього ж сигналу. Із рисунку помічаємо, що не можна розділити кілька спектральних складових. На основі частотної характеристики сигналу помітно тільки те, що на високих частотах середнє значення потужності звуку нижче, ніж на низьких, що характерно для безліч випадків. Враховуючи необхідність аналізу пострілу, який має визначений інтервал тривалості та інші часові особливості, даний метод непридатний для вирішення задачі ідентифікації об’єктів на основі АП.



Рисунок 2.2 Частотна характеристика сигналу

Метод нейронних мереж потребує тривалого часу для запам’ятовування зв’язків між нейронами, особливо при виявленні сигналу, що матиме різноманітні значення амплітуд залежно від відстані до джерела звуку. Приблизна оцінка часу навчання системи становить 3-4 місяці. Крім того, при частковій зміні предметної галузі, тобто при обробці вибуху міни, а не пострілу із гвинтівки чи пістолета, доведеться по-новому перенавчати нейронну мережу.

Метод фільтрації рівня гучності добре підходить для початкового аналізу сигналу. Як відомо, постріл має різке збільшення амплітуди звуку на початку та поступове затухання сигналу, що дає можливість позбутися зайвого аналізу шуму, що має нижчі амплітуди звуку. Значення ∆L при налаштуванні задаватиметься у відсотках від максимального значення амплітуди сигналу на заданому проміжку.

Метод направленого прослуховування ефіру не буде задіяний, адже немає уточнень предметною галуззю щодо дозволеної кількості сенсорів та узгоджень їх взаємного розміщення у просторі. Як відомо, підключення і аналіз одночасно двох каналів з двох сенсорів потребує додаткового дослідження, що не зачіпає тематики даного дипломного проекту.

Фрактальний метод цілком підходить для вирішення даного завдання, але через не наочність процесу ідентифікації він не використовуватиметься.

Перетворення Фур’є дозволяє аналізувати лише значення амплітуди при кожному значенню частоти. Сигнал, зображений на рисунку 2.1, видозмінений за допомогою ПФ до вигляду графіку, зображеному на рисунку 2.2. Частотний метод непридатний для заданої предметної галузі, тому потреба у ПФ відпадає.

## Віконне перетворення Фур’є

Математичним фундаментом для вирішення задачі ідентифікації об’єктів на основі АП є безперервне вейвлет – перетворення. Вперше ідея ВП була запропонована як подолання проблеми ПФ щодо частотно-часового відображення. Частково цю проблему вдалося розв’язати використавши віконне перетворення Фур’є. Проте залишилась проблема явища, що називається принципом невизначеності Гейзенберга. Цей принцип стосовно ВПФ наголошує, що неможливо отримати точне частотно-часове відображення сигналу, тобто не можна визначити для якого моменту часу які спектральні компоненти присутні у сигналі. Можемо тільки знати, власне, часові інтервали, протягом яких у сигналі існують смуги частот. Ця проблема ще називається проблемою роздільної здатності.

Оскільки ВП придумано як рішення проблеми ВПФ та цілком базується на ідеї ВПФ, розглянемо базові математичні поняття віконного перетворення Фур’є.

Оскільки ПФ працює лише з стаціонарними сигналами, то суть ВПФ полягає у тому, щоб розділити всю часову шкалу на проміжки, які надалі називатимемо вікном, щоб сигнал всередині нього виглядав стаціонарним. Для цього щодо сигналу застосовується віконна функція w, ширина якої має бути рівною ширині вікна.

Якщо ширина віконної функції T секунд, тоді у момент часу t=0 вона перекривається на T/2 секунди сигналу. Віконна функція і сигнал перемножуються. Якщо віконна функція прямокутна і одиничної висоти, то сигнал не змінюється. В іншому разі він зважується з віконною функцією. Тоді добуток підлягає перетворенню Фур’є.

У результаті отримуємо ПФ перших T/2 секунди вхідного сигналу. Якщо цей відрізок стаціонарний, то отриманий результат перетворення коректно відображає частотне наповнення перших T/2 секунди сигналу.

Наступним кроком є зсув віконної функції на деяку величину t сек. Зсунута функція знову множиться на сигнал, виконується ПФ добутку. Дана процедура повторюється до досягнення кінця вхідного сигналу,

де x(t) – вхідний сигнал, w(t) – віконна функція, w\* - означає комплексне спряження. Іншими словами, ВПФ – це ПФ сигналу, що множиться на віконну функцію.

Для кожного t’ і f обчислюється свій коефіцієнт ВПФ.

Таким чином ми отримуємо частотно-часове перетворення сигналу.

Покажемо на рисунку 2.3 різні частотні компоненти: від 0 до 250 мс – 300 Гц, і, далі 200, 100 та 50 Гц. ВПФ цього нестаціонарного сигналу зображено на рисунку 2.4

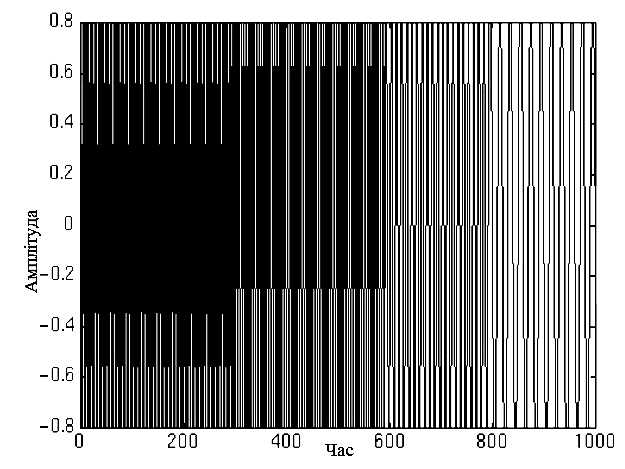


Рисунок 2.3 Вигляд вхідного сигналу

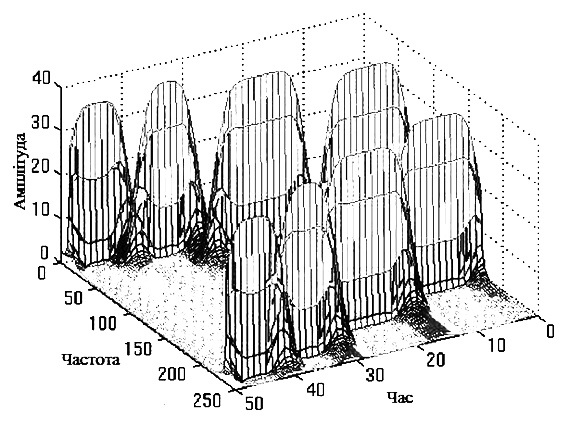


Рисунок 2.4 Результат ВПФ

Це тривимірний графік. По осях х та у відкладені час і частота відповідно. Графік симетричний відносно осі частот (не показано на рисунку).

На рисунку помітна наявність чотирьох піків, що відповідають чотирьом частотним компонентам. На відміну від ПФ, ці піки локалізовані у різноманітних часових інтервалах, як це було і у вхідному сигналі.

Ми отримали частотно-часову характеристику сигналу, проте за принципом невизначеності Гейзенберга, описаним раніше, дані частоти не є точними. Оптимальна частотна роздільна здатність досягається при ПФ за рахунок того, що вікно є функцією exp (j×w×t), що простягається від мінус до плюс нескінченності. При ВПФ вікно має визначену кінцеву довжину, тому накриває лише частину сигналу, тому і частотна роздільна здатність зменшується. Вузьке вікно забезпечує кращу часову роздільну здатність, а широке – часову. Тому ми не знаємо точно присутніх у сигналі частот, а лише їх смуги.

## Вейвлет-перетворення

Отож покажемо, як ця проблема вирішується у математичній реалізації ВП. Існує можливість аналізу спектру сигналу за допомогою альтернативного підходу, що називається кратномасштабним аналізом (КМА). КМА аналізує сигнал на різноманітних частотах і при різній роздільній здатності водночас. Кожна спектральна компонента не аналізується окремо, як це було у випадку з ВПФ.

КМА дозволяє отримати хорошу роздільну здатність у часі (погану по частоті) на високих частотах та хорошу роздільну здатність по частоті (погану у часі) на низьких частотах. Даний підхід є особливо ефективним, коли сигнал має високочастотні компоненти короткої тривалості і довгі низькочастотні. Саме такі сигнали трапляються найбільш часто. Для пояснення задачі роздільної здатності при ВП зазвичай використовується рисунок 2.5. Кожен прямокутник відповідає значенню ВП на частотно-часовій площині. Площа прямокутників ненульова, це означає те, що ми не можемо обчислити будь-яку точку площини. Всі точки, що належать одному прямокутнику, подаються як одне значення ВП.

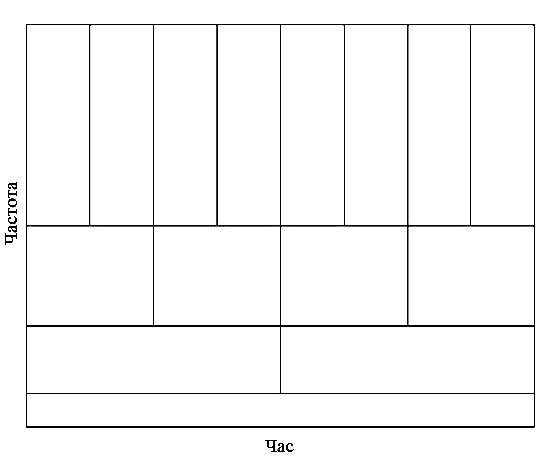


Рисунок 2.5 Роздільна здатність ФП

Безперервний вейвлет-аналіз виконується аналогічно ВПФ, тобто сигнал множиться на функцію (вейвлет), і обчислення виконується окремо для різних проміжків часу сигналу. Але є дві суттєві відмінності між ВП і ВПФ:

* Не виконується ПФ зваженого з вейвлетом сигналу. Тому значення одиниці відповідає синусоїді, тобто від’ємні частоти не обчислюються.
* Ширина вікна змінюється, тому перетворення здійснюються для кожної спектральної компоненти, що є важливою властивістю ВП.

Безперервне ВП визначається так:

Як помітно із рівності, перетворений сигнал є функцією від двох змінних, та s, параметрів зсуву та масштабу відповідно. (t) – функція перетворення, що називається материнським вейвлетом.

Термін зсув використовується у тому ж сенсі, що і при ПФ: він відноситься до місця розташування вікна, і вікно рухається уздовж сигналу. Цей термін відноситься до часової інформації, присутньої у результаті перетворення. Але при ВП ми не маємо частотного параметру, як це було при ВПФ. Замість нього маємо параметр масштабу, який можна визначити як величину, обернену частоті.

Параметр масштабу у ВП має аналогію з масштабом географічних карт. Більші значення масштабу відповідають малій кількості деталей, глобальному представленню сигналу, а низькі значення масштабу дозволяють відрізнити деталі. Тобто низькі частоти відповідають глобальній інформації про сигнал (яка міститься на всій його протяжності), а високі частоти – детальній інформації, прихованим особливостям, що тримають недовго.

Якщо f(t) вихідна функція, то f(s×t) відповідає стисненій версії f(t), якщо s > 1, і розширеній версії, якщо s < 1.

У формулі коефіцієнт масштабу є ще у знаменнику. Тому s > 1 розширює сигнал, а s < 1 стискає його.

Процедура аналізу стартує із масштабу s = 1 і продовжується, поступово збільшуючи значення s, тобто аналіз розпочинається із високих частот і проводиться у сторону низьких частот. Перше значення s відповідає найбільш стиснутому вейвлету. При збільшенні значення s вейвлет розширюється.

Вейвлет розташовується на початку сигналу. Вейвлет-функція масштабу s = 1 множиться із сигналом та інтегрується на всій часовій шкалі. Інтеграл множиться на константу 1/sqrt(s) для нормалізації, тобто для того, щоб сигнал при кожному значенню масштабу мав би однакову енергію.

Вейвлет масштабу s=1 потім зсовується вправо на τ до точки t=τ, і процедура повторюється. Процедура триває, доки вейвлет не досягне кінця сигналу.

На рисунку 2.6 показаний рух вейвлета вздовж осі часу для чотирьох різних значень τ. Значення масштабу 1 відповідає найменшому значенню, або найвищій частоті.

Якщо у сигналі присутні спектральні компоненти, що відповідають певному значенню s (на рисунку 2.6 s=1), то добуток вейвлета із сигналом у інтервалі, де розміщена спектральна компонента, дасть відносно велике значення. В іншому разі – добуток буде малим або рівним нулю. Сигнал, показаний на рисунку 2.6, має широкі спектральні компоненти на інтервалі близько t=100 мс.

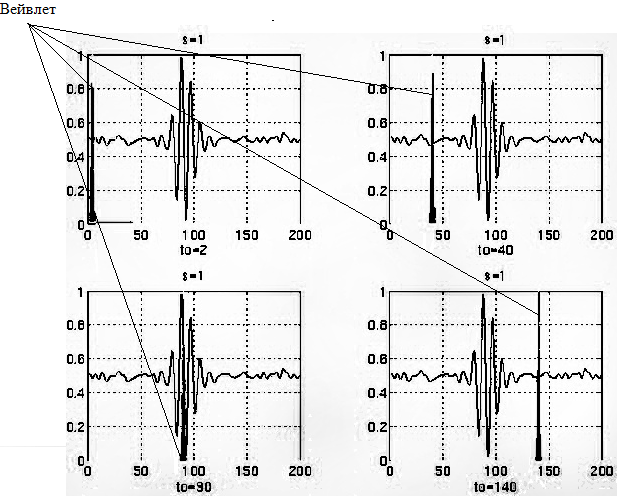


Рисунок 2.6 Рух вейвлета по часовій шкалі

## Числова характеристика форми графіків сигналів

У ході ідентифікації звуку на основі АП двох сигналів необхідний численний метод порівняння їх спектральних складових. Для цього зручно скористуватися операцією згортки із функціонального аналізу, що є особливим видом інтегрального перетворення. Операцію згортки можна інтерпретувати як схожість із зсунутою копією іншої. Результатом операції є функція, максимальним значенням якої можна вважати найбільшу схожість сигналів при зсуві по осі часу, аргумент x – координата найбільшої схожості графіків.

f, g – функції двох сигналів,

t – аргумент, що служить для зсуву по всій шкалі функції g,

x – точка на осі, вздовж якої відбувається зсув.

На рисунку 2.7 нижній графік показує зростаючу “схожість” функцій f1: y = x та f2: y = x2 ближче до x=1.

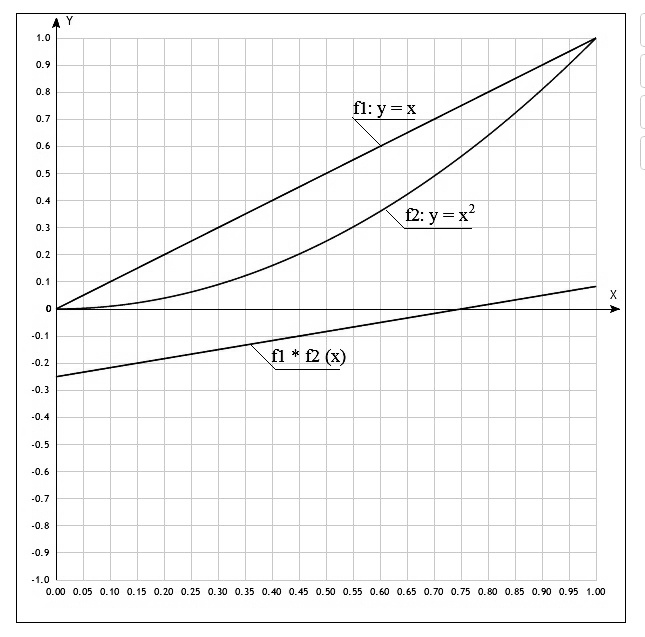


Рисунок 2.7 Залежність значення згортки f1 \* f2 від схожості f1 і f2

Враховуючи потребу у аналізі енергії, якою володіє певний сигнал, у процесі обчислень будемо використовувати фізичний смисл визначеного інтегралу, що рівний площі на деякому відрізку [a; b], обмеженої віссю абсцис та функції, яка інтегрується.

Враховуючи специфічний спектральний вигляд вхідного сигналу пострілу, де основна енергія сконцентрована на початку, можемо отримати числове співвідношення енергії зліва та справа від точки на осі часу x = x0. Нехай функція (x) буде значенням, яке показує числову міру даного співвідношення у будь-якій точці на осі абсцис.

## Висновки

Серед усіх запропонованих підходів до розв’язання задачі дипломного проекту у розділі 1, ми обрали метод фільтрації рівня гучності та вейвлет-перетворення. Перший метод дозволить працювати лише із високими частотами, а ВП дає змогу отримати більш інформативні для обчислення коефіцієнти.

Як математичний апарат використовується безперервне ВП, що є модифікованим ВПФ, а ВПФ є спробою вирішення проблеми принципу невизначеності Гейзенберга методу перетворення Фур’є.

При ВП можемо задавати глибину рівня декомпозиції сигналу, що визначає міру масштабного коефіцієнта s, який є числовою мірою ширини вікна вейвлета. У результаті ВП отримуємо деталізовані та апроксимовані (наближені) коефіцієнти, які полегшують процес аналізу особливостей вхідного сигналу.

Для підрахунку схожості вхідного та зразкового сигналів зручно використати операцію згортки із функціонального аналізу, яка ідентифікує сигнал у випадку його відмінності у фазі.

# ОПИС ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ

## Обґрунтування вибору мови програмування Python

Програма для ідентифікації об’єктів на основі АП реалізована мовою високого рівня програмування Python. Такий вибір пояснюється:

1. легкою і швидкою реалізацією цією мовою програмування,
2. відсутністю потреби у оперуванні елементарними типами змінних при роботі із великими структурами даних (масив байтів сигналу),
3. наявністю бібліотечних засобів щодо обробки звуку,
4. простим способом графічного відображення даних за допомогою бібліотеки Pylab,
5. скриптовістю, що дозволяє автоматизувати повторювані дії, такі як викачування та обробка кількасот файлів.

Для розробки складного ПЗ на першому етапі зручно використовувати для моделювання та оцінки майбутнього результату саме мову програмування Python, що суттєво зекономить час. Потім при потребі варто переробити проект мовою більш низького рівня, такою як C++, Java чи Lisp, що значно пришвидшить виконання програми.

Для швидкої і наочної розробки ПЗ були використані бібліотеки, що спрощують алгоритми ідентифікації звуків, серед них:

* scipy.io.wavfile – модуль для зчитування WAV-файлу із жорсткого диску,
* pywt – модуль для отримання вейвлет-коефіцієнтів на основі вхідного сигналу,
* peakutils, numpy – модулі обчислення пікових значень у вхідному сигналі із кроком дискретизації та мінімальним пороговим значенням піку,
* peakutils.plot – для виведення графіків із наведеними піковими значеннями на них,
* scipy.interpolate.UnivariateSpline – модуль для інтерполювання сигналу,
* pylab, matplotlib – бібліотека графічних засобів Python для побудови спектральних складових,
* os – для лістингу списку файлів у заданій директорії на жорсткому диску,

## Принцип та алгоритм роботи програми

Принцип роботи програми: заголовки wav-файлів подаються як вхідні дані ПЗ, у кожному з файлів виділяються елементарні сигнали, кожен з яких пропускається через 4 фільтри, ціллю яких є виявити сигнали, які точно не є звуком пострілу. Якщо один із фільтрів повертає значення False, то далі сигнал не обробляється. За умови, що всі виділені сигнали з даного wav-файлу не пройшли через фільтри, програма формує відповідь про відсутність звуку пострілу у заданому звуковому файлі.

Алгоритм роботи програми:

1. Задається директорія, на основі якої формується список назв файлів для обробки.
2. Обирається заголовок із списку
3. Зчитується файл по назві як масив байтів завдяки scipy.io.wavfile
4. Здійснюється селекція спектральних складових із масиву сигналу у новий список.
5. Спектральні складові очищуються від зайвої послідовності нулів.
6. Елементарні звуки пропускаються через фільтри.
7. Якщо хоч одна спектральна складова пройшла через всі фільтри, формуємо повідомлення успішної ідентифікації звуку.
8. Якщо є необроблений заголовок звукового файлу, то переходимо на пункт 2.
9. Завершення програми.

## Структура програми

Програма складається із трьох основних модулів:

• LoadScript.py – здійснює послідовний аналіз та ідентифікацію звуку пострілу у для кожного із заданого списку файлів, при потребі служить для скриптового завантаження файлів із певного Web-ресурсу.

• SoundProcessing.py – містить базові засоби для зчитування wav-файлів, обробки, селекції спектральних складових у сигналі, інтерполювання графіків, оцінки степені схожості двох сигналів, очищення спектральної складової, виведення на екран графіків сигналу.

• Filters.py – містить функції для фільтрації вхідного сигналу за формою, тривалістю, розподілом енергії сигналу та локалізацією пікового значення сигналу; повертає True якщо сигнал пройшов усі фільтри, тобто є постріл у файлі.

Опис функцій у кожному модулі:

Модуль *SoundProcessing.py:*

* *def read\_signal(f, wavelet='db1'):* зчитує файл з назвою f, замість від’ємних значень потужності звуку беремо їх модулі (працюємо лише з додатними величинами), застосовує до нього ВП (за замовчуванням обирається вейвлет *‘db1’*), та повертає бітрейт звукового файлу із масивом вейвлет-коефіцієнтів.
* *def select\_basic\_signals(wave, rat, min\_t=0.1, max\_t=3.0)*: розкладає сигнал спектральні складові мінімальної тривалості *min****\_****t* та максимальної *max\_t*.
* *def interpolate\_signal(wave, new\_len)*: стискає сигнал у часі, зберігаючи його початкову форму
* *def show\_signals(waves, mes='default'):* виводить список спектральних складових *waves* на графік, для візуальної оцінки їх форм.
* *def clean\_signal(wave):* очищає спектральну складову від довгої послідовності нулей для спрощення її аналізу.
* *def count\_scaled\_similarity(w1, w2)*: виводить значення схожості двох сигналів у форматі подвійної точності: якщо 1.0 – сигнали однакові, 0.0 – сигнали різні.

Модуль *Filters.py*:

* *def filter\_by\_duration(wave, rat, min\_sec=0.01, max\_sec=10):* фільтрує сигнал за мінімальним значенням *min\_sec* та максимальним значенням *max\_sec* тривалості спеткральної складової.
* *def filter\_energy(wave, min\_val=1.5):* фільтрує складову за співвідношенням енергії до та після піку. Якщо це співвідношення менше за *min\_val,* то звук не проходить через фільтр.
* *def filter\_by\_peak(wave, area=0.2):* фільтрує складову за положенням піку, якщо він не розміщений на перших *area* відсотках всього сигналу, то складова не є звуком пострілу.
* *def filter\_by\_form(w, pattern, thres=0.08, n=50):* фільтрує сигнал за формою, повертає True, якщо вхідний сигнал схожий на зразок.
* *def run\_filters(name1, pattern='wav\_success/pattern.wav'):* розкладає цілісний звуковий файл на спектральні складові, послідовно їх пропускає через всі фільтри, виводить кінцевий результат як наявність зразкової спектральної складової у вхідному сигналі.

Модуль *LoadScript.py:*

* *def download\_sound(url, s):* завантажує звуковий файл із сторінки по адресі *url* на жорсткий диск із назвою *s*.
* *def download\_page(url):* завантажує сторінку по адресі *url* та повертає список її байт-кодів.
* *def refs\_list\_of\_sounds\_from\_url(url):* шукає посилання на звукові файли на сторінці за адресою *url*
* *def search\_url\_to\_next\_page(url):* на сторінці по адресі *url* шукає посилання на наступну сторінку*.*
* *def get\_download\_url\_and\_name(url):* на сторінці по адресі *url* шукає та повертає назву звукового файлу та посилання на нього.
* *def load\_site(current\_page):* завантажує всі можливі звукові файли по заданій сторінці *page.*
* *def run\_tests(folder):* запускає обробку заданої кількості звукових файлів, послідовно показуючи результати ідентифікації пострілу.

## Використання бібліотеки pywt

Бібліотека дозволяє виконувати:

* Декомпозицію сигналу одного рівня за допомогою

*(cA, cD) = dwt(data, wavelet, mode='sym')*

де

data – вхідний сигнал, заданий як одновимірний масив дійсних значень,

wavelet – графік математичного вейвлета,

mode – необов’язковий параметр, що характеризує спосіб вирішення проблеми граничних значень.

cA – одновимірний масив апроксимованих (наближених) вейвлет-коефіцієнтів,

cD – одновимірний масив деталізованих вейвлет-коефіцієнтів.

У ході обчислень використовуватимемо деталізовані коефіцієнти, які дають більше інформації на високих частотах.

* Багаторівневу декомпозицію

*pywt.wavedec(data, wavelet, mode='sym', level=None)*

Параметри ті ж самі, за винятком *level*, що задає рівень декомпозиції. Функція повертає список масивів коефіцієнтів у такому форматі:

*[cA\_n, cD\_n, cD\_n-1, ..., cD2, cD1]*

Де *cA\_n* – масив апроксимованих коефіцієнтів, *cD\_i* – масив вейвлет-коефіцієнтів на *i*-ому рівні декомпозиції.

Оскільки ПЗ повинне працювати із звуками, записаними в умовах відкритої місцевості, високий рівень декомпозиції непотрібний.

## Організація користувацького інтерфейсу ПЗ

Режими запуску процесу аналізу звукового файлу:

* запуск одного файлу, вказавши повний шлях до нього;
* запуск множини файлів, задавши повний шлях директорії.

Графічна частина містить виведення:

* виведення амплітудного графіку вхідного сигналу;
* виведення пікових значень амплітуд сигналу;
* виведення інтерпольованого за часом графіку сигналу.

Так як обробка великої кількості звукових файлів може тривати протягом 10-20 хвилин, тому у ході обчислення здійснюється текстове виведення стану програми.

Після закінчення аналізу сигналу програма виводить булеве значення, що показує чи містить даний файл спектральну складову звуку.

## Висновки

Вибір мови програмування був здійснений на основі критерію простоти та швидкості розробки завдяки її високому рівню та відсутністю складних синтаксичних конструкцій, що є невід’ємним засобом при роботі з об’ємними структурами даних, такими як файли звукозапису.

Для зрозумілої логіки побудови ПЗ використовується поняття фільтрів, ціллю яких є знаходження відмінності вхідного сигналу від збереженого і проаналізованого раніше зразку.

Структура ПЗ представлена у вигляді модулів, кожен з яких забезпечує уніфікований функціонал всієї програми.

Модуль обробки сигналу містить функції зчитування, селекції, інтерполяції, виведення та обчислення схожості вхідних сигналів. Модуль фільтрів має функції перевірки сигналу за його тривалістю, положенням піку, розподілом енергії сигналу та за його формою.

Ключовою у роботі програми є бібліотека *pywt*, що містить метод *dwt* для отримання вейвлет-коефіцієнтів.

# ПЕРЕВІРКА ПРАВИЛЬНОСТІ ІДЕНТИФІКАЦІЇ ОБ’ЄКТІВ

## Опис умов тестування

Як вхідні дані для розробленого ПЗ подається список із різноманітних звукових файлів. ПЗ працює лише із форматом “.wav”, що пояснюється у пункті 1 розділу 1.

Для перевірки та аналізу правильності роботи програми необхідний ресурс із великою кількістю звукових файлів заданого формату. Враховуючи зручні пошукові засоби та зрозумілу структуру веб-сторінок, вирішено обрати електронний медіа-ресурс freesound.org.

За допомогою розробленого модуля *LoadScript.py*, опис якого наведений у п 3.3, можна легко завантажити будь-яку кількість файлів. Для чіткого визначення роботоздатності програмних засобів ідентифікації об’єктів на основі АП необхідно отримати більше 1000 файлів

Програма повинна визначати наявність спектральних складових у файлах пострілу, та констатувати відсутність у випадках, де їх немає.

Щоб продемонструвати динамічність роботи ПЗ задаємо файли різних тем, тобто звуки, які містять голос тварин, звуки транспорту, файли, які не містять помітних для людського слуху помітних особливостей (звуковий фон вулиць, природи чи річки) та інші. Для якісного аналізу кількість файлів кожної з тем має бути не менше 20. Такий підхід до тестування програми є GUI-автоматизованим на основі інструменту сценаріїв (Scripting). Оскільки усі файли завантажуються автоматично, немає змоги чітко проконтролювати вміст кожного з них людським слухом. Це створює додаткові труднощі при визначенні роботоздатності програми, адже вміст файлів, завантажених із посилань однієї тематики, не завжди відповідає своїм заголовкам. Тому ми керуватимемося статистичними даними, тобто аналізуватимемо відсоток ідентифікованих звуків. Крім того, ПЗ дозволяє аналізувати файл окремо, це означає те, що, знаючи вміст заданого сигналу, можна перевірити тільки його. Такий підхід називається ручним тестуванням (Manual testing).

Враховуючи низьку якість завантажених файлів, можна передбачити наявність відчутної похибки при аналізі результатів, адже деякі з файлів нелегко розпізнати навіть людським слухом.

## Результати тестів

Отож, як було зазначено у п. 4.1, тестування здійснюватимемо при GUI-автоматизованому та ручному підходах. При першому покажемо лише числові характеристики дослідження, знайдемо процент правильно ідентифікованих звукових файлів.

Побудуємо таблицю, що відображає здатність ПЗ правильно ідентифікувати відсутність звуку пострілу у вхідних сигналах.

Таблиця 4.1

Тестування автоматизованим підходом для файлів без пострілу

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тема звуків | Кількість файлів | Хибно-позитивні (%) | Правильно-негативні (%) | Помилкових файлів (%) |
| Пташки | 155 | 8.33 | 85.26 | 6.41 |
| Коти | 92 | 7.52 | 89.25 | 3.22 |
| Собаки | 93 | 8.51 | 88.3 | 3.19 |
| Гелікоптер | 21 | 9.09 | 81.82 | 9.09 |
| Природа | 139 | 7.14 | 89.29 | 3.57 |

Продовження табл. 4.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тема звуків | Кількість файлів | Хибно-позитивні (%) | Правильно-негативні (%) | Помилкових файлів (%) |
| Літак | 23 | 8.33 | 83.33 | 8.33 |
| Дощ | 29 | 6.67 | 90 | 3.33 |
| Крик | 299 | 6.33 | 91 | 2.67 |
| Тиша | 54 | 10.91 | 85.45 | 3.64 |
| Вулиця | 64 | 7.69 | 80 | 12.3 |
| Поїзд | 73 | 5.4 | 85.1 | 9.46 |

Тепер побудуємо таблицю, що відображає здатність ПЗ розпізнавати звуки пострілів.

Таблиця 4.2

Тестування автоматизованим підходом файлів, що містять постріл

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тема звуків | Кількість файлів | Хибно-негативні (%) | Правильно-позитивні (%) | Помилкових файлів (%) |
| Постріл | 268 | 40.07 | 56.18 | 3.74 |

На основі наведених значень із табл. 4.1 та 4.2 сформуємо узагальнені показники правильності роботи ПЗ у табл. 4.3. Хибно-позитивний результат знайдемо як середнє значення помилково ідентифікованих звуків пострілу у відсотках, хибно-негативний результат – помилково пропущених звуків пострілу, правильно-негативний результат – вірно розпізнаних файлів без складової пострілу, правильно-позитивний – вірно розпізнаних файлів, що містять складову пострілу. Значення імовірності хибного розпізнавання знайдемо як середнє арифметичне хибно-позитивного та хибно-негативного результату. Даний показник характеризує помилковість роботи системи. Імовірність правильного розпізнавання обчислюється аналогічно, і, відповідно, показує шанс отримання правильного результату. Імовірність хибного та правильного розпізнавання необхідно нормувати, щоб отримати значення повної імовірності рівної одиниці, тобто:

ІХР = (7.72 + 40.07) / 2 = 23.89

ІПЗ = (56.18 + 85.26) / 2 = 70.72

Після операції нормування отримаємо

Таблиця 4.3

Показники тестування роботи ПЗ

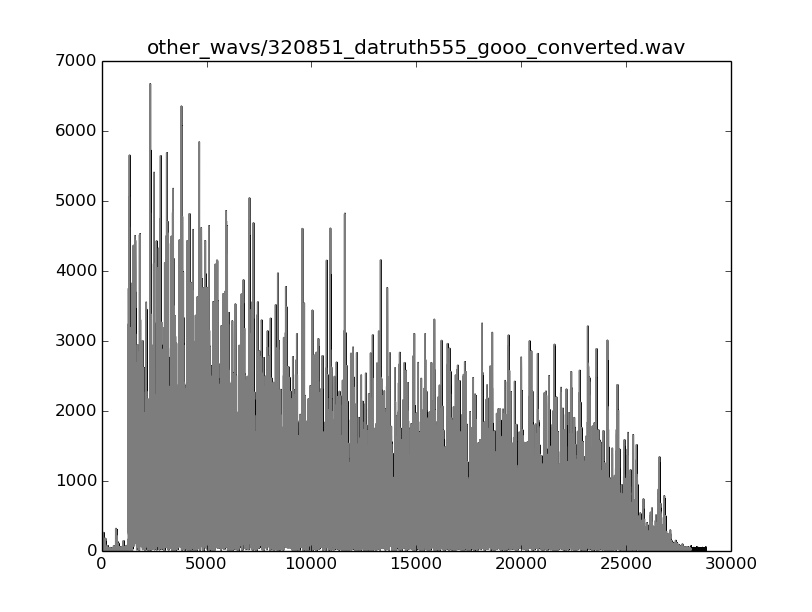
|  |  |
| --- | --- |
| Критерій | Значення (%) |
| Хибно-позитивний результат | 7.72 |
| Хибно-негативний результат | 40.07 |
| Правильно-позитивний результат | 56.18 |
| Правильно-негативний результат | 85.26 |
| Імовірність хибного розпізнавання | 25.25 |
| Імовірність правильного розпізнавання | 74.74 |

Тестування при ручному підході містить перевірку файлу із пострілом та без.

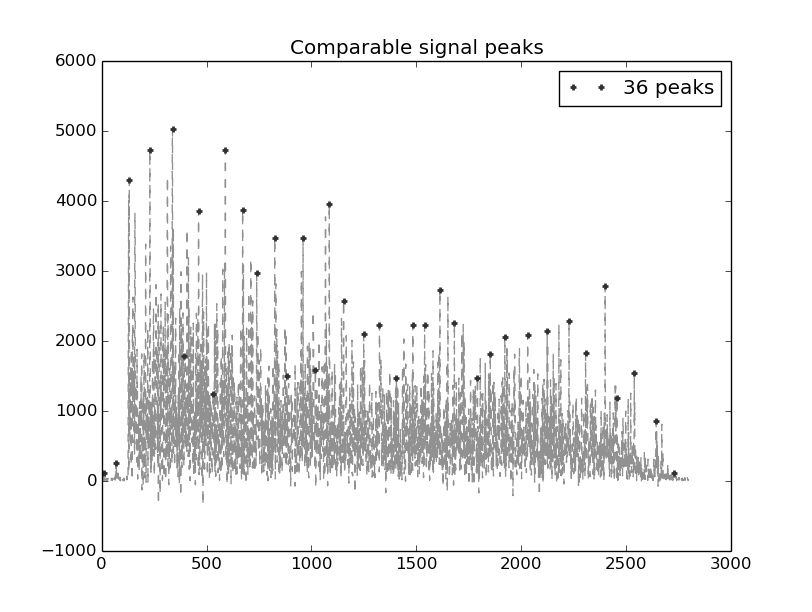
При ручному тестуванні проаналізуємо 2 файли для доведення правильно-позитивного та правильно-негативного результату, показавши обчислення на кожному етапі процесу ідентифікації об’єктів.

Як приклад файлу без пострілу виберемо різкий викрик людини тривалістю до трьох секунд.

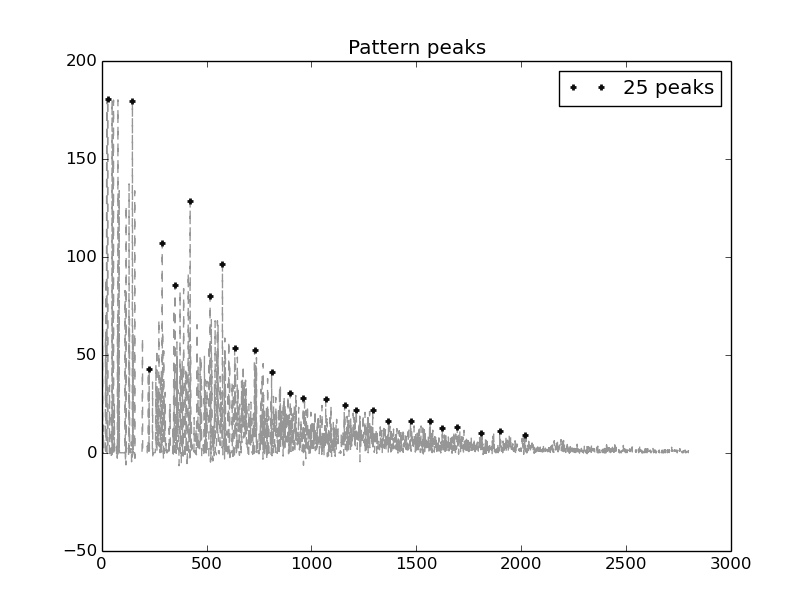
На рисунку 4.1 покажемо виділену спектральну складову сигналу, тобто частину графіку, кожне значення коефіцієнту на певному проміжку є більшим за мінімальний поріг, що підбирається емпірично. Горизонтальна вісь – вісь часу, де одній секунді відповідає кількість байтів, що рівна частоті дискретизації файлу. Вертикальній осі відповідає значення вейвлет-коефіцієнтів.

Рисунок 4.1 Виділена спектральна складова

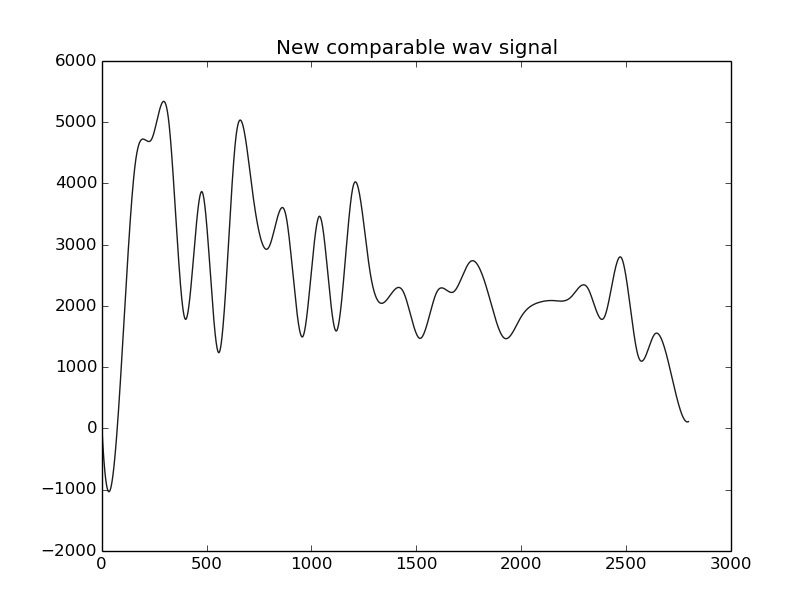
Далі розміщуємо пікові значення коефіцієнтів із емпірично підібраними нижнім порогом та інтервалом. У результаті отримаємо графік на рис. 4.2.

Рисунок 4.2 Піки вхідного сигналу

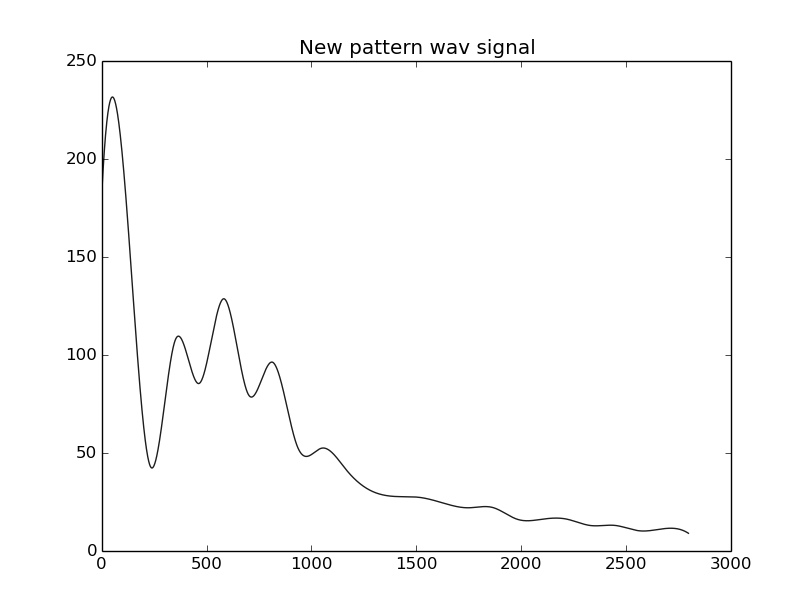
Аналогічну процедуру виконуємо для файлу, який подаємо як зразок.

Рисунок 4.3 Піки сигналу-зразку

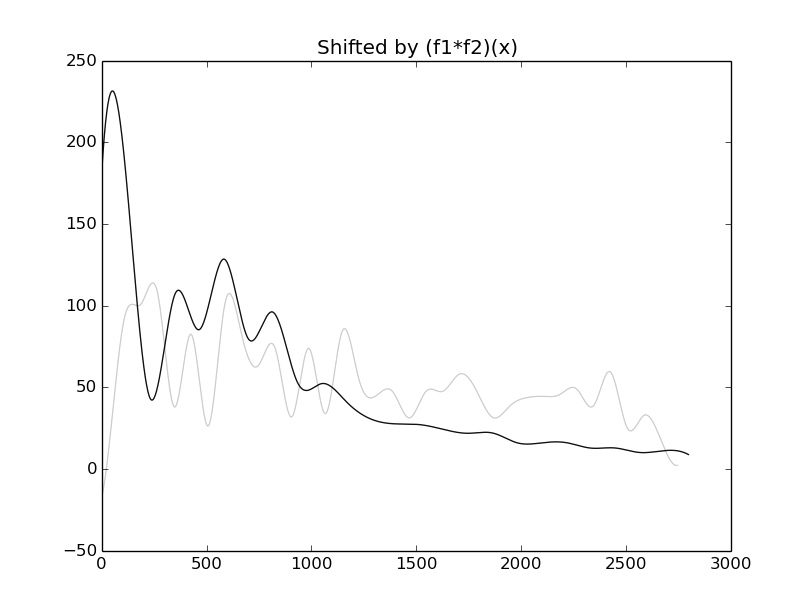
Інтерполюємо графік пікових значень вхідного сигналу, щоб збільшити точність обчислення схожості із сигналом-зразком.

Рисунок 4.4 Інтерпольований графік вхідного сигналу

Інтерполюємо графік пікових значень сигналу-зразку.

Рисунок 4.5 Інтерпольований графік сигналу-зразку

Після операції згортки та зсуву отримаємо 2 графіки, які будемо порівнювати по кожному елементу.

Рисунок 4.6 Порівняння схожості сигналів методом згортки

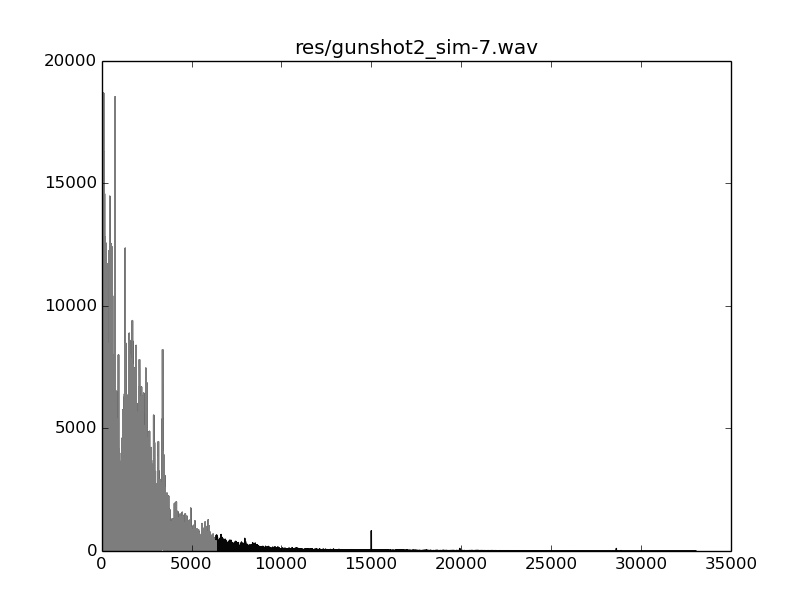
Лістингом роботи програми є:

*1 d->1 e->1 p->1 FALSE [0.709432125066]*

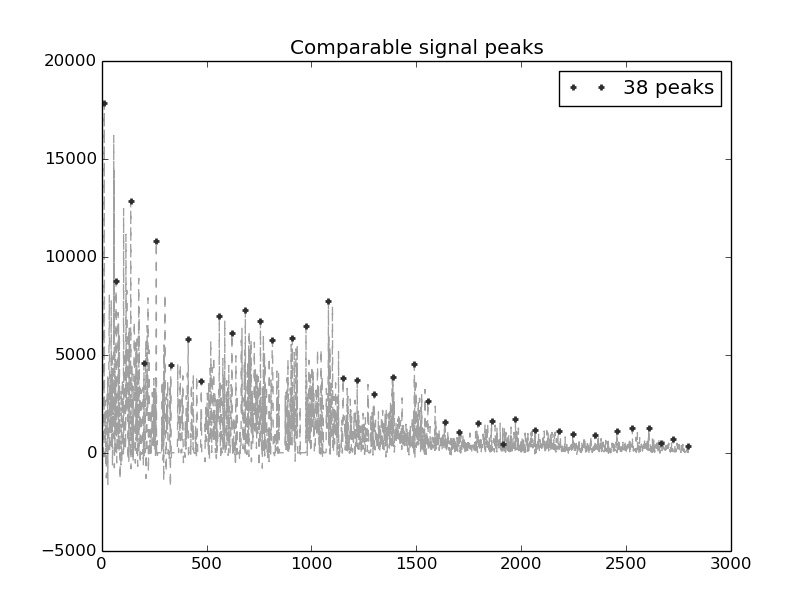
Тобто початковий сигнал був розкладений на одну спектральну складову, що пройшла через фільтри тривалості, енергетичного співвідношення та локації максимального значення. Програма констатує відсутність звуків пострілу у файлі, адже коефіцієнт схожості низький.

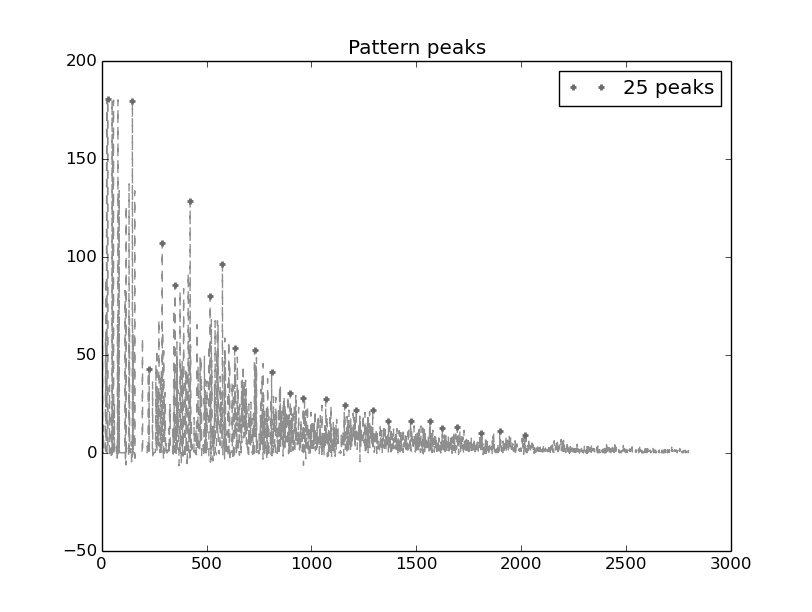
Проаналізуємо файл із пострілом.

На рис. 4.7 показано виділену спектральну складову, що і є пострілом. При значенні 15000 помітний стрибок значень коефіцієнту, проте він не розпізнаний як окрема спектральна складова звуку, адже не має достатньої тривалості.

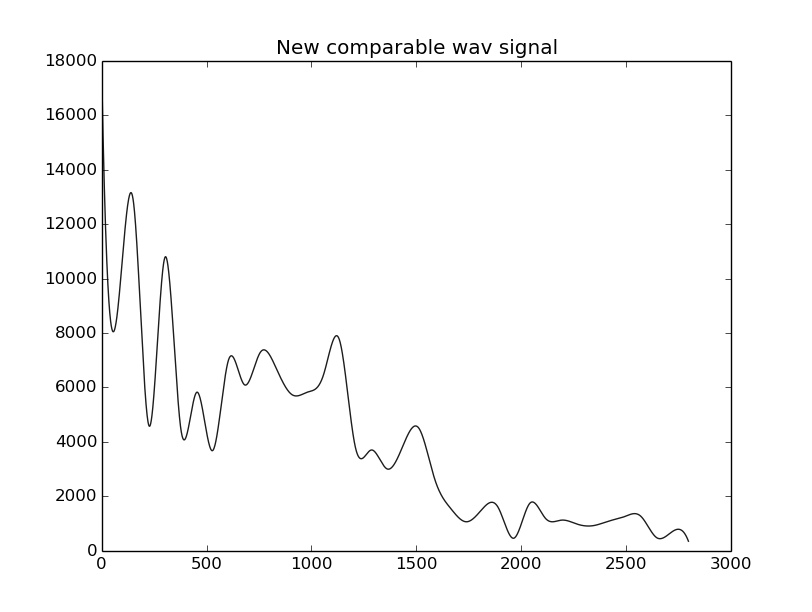
Рисунок 4.7 Вхідний сигнал і виділена складова файлу з пострілом

На рис. 4.8 спостерігаємо більш схожий до форми сигналу пострілу графік, утворений піковими значеннями.

Рисунок 4.8 Піки вхідного сигналу

Рисунок 4.9 Піки сигналу-зразка

На рис. 4.8 максимальне значення вейвлет-коефіцієнтів становить близько 20000, на рис. 4.9 – майже 200. Така відмінність пояснюється умовами якістю сенсора, відстані до джерела випроміненого звуку та станом фізичного середовища. Для належного порівняння складових необхідно нормалізувати вхідний сигнал, тому на рис. 4.11 графіки мають однакове максимальне значення.

Рисунок 4.10 Інтерпольований вхідний сигнал

Інтерпольований графік сигналу-зразка аналогічний як на рисунку 4.5

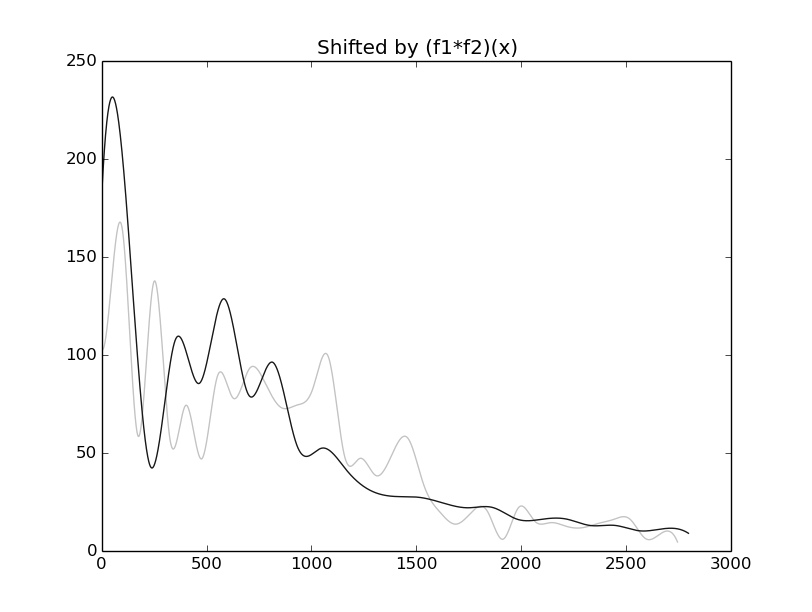


Рисунок 4.11 Порівняння схожості сигналів методом згортки

Лістингом роботи програми є:

*1 d->1 e->1 p->1 | TRUE [0.919552632592]*

Тобто початковий сигнал був розкладений на одну спектральну складову, що пройшла через усі фільтри. Програма констатує наявність звуку пострілу у файлі, адже коефіцієнт схожості високий.

## Аналіз результатів тестування

Аналізуючи тестування автоматизованим підходом для файлів без пострілу, можна помітити, що найбільш часто програма хибно ідентифіковує постріл у файлах із звуком тиші із імовірністю 10.91%. Це легко пояснити, адже будь-які клацання та хрускоти мають різкий перепад частоти у момент появи та поступове затухання коливання, як і звуковий портрет пострілу. Найменш часто програма хибно-позитивно ідентифіковує постріл у файлах із звуками поїзда із імовірністю 5.4%. Поїзд має відносно стабільний графік звукових коливань та не має різкого перепаду амплітуди.

При проведенні експерименту для отримання правильно-позитивного та хибно-негативного показників були отримані результати 40.07% та 56.18% відповідно. Варто врахувати відсоток помилково прочитаних файлів, що пояснюється неправильним внутрішнім форматом файлу та створює труднощі при зчитуванні стандартними засобами бібліотек. Програма ідентифікувала постріл протягом більше половини проведених експериментів, тобто успішно виконала поставлену вимогу.

Аналіз методом випадкової вибірки дозволяє простежити за візуальною схожістю порівнюваних сигналів. В обох випадках тестування ми отримали правильний результат, що демонструє роботоздатність програми, принаймні, для коректно записаних файлів.

Показники у табл. 4.3, загалом, характеризують можливості програми щодо розпізнавання звуку на основі його акустичного портрету. Тобто, з імовірністю 74.74% програма правильно констатує факт наявності чи відсутності спектральної складової пострілу.

## Висновки

Для тестування розробленого ПЗ використовується автоматизований та ручний підхід перевірки правильності його роботи.

Для аналізу роботоздатності програми необхідно здійснити тестування для файлу із звуком пострілу та без. Щоб продемонструвати гнучкість та динамічність обчислення, проаналізовано файли, що не містять звук пострілу з різних тем.

Через низьку якість звукозапису та невідповідність вмісту деяких звукових файлів їх заголовкам виникає помітна похибка.

Протягом ручного тестування ПЗ правильно згенерувало висновок наявності та відсутності спектральної складової пострілу для випадково відібраних файлів.

Отже, дане ПЗ здатне констатувати відсутність та наявність звуку пострілу із середнім значенням імовірності 74.74%.

# ВИСНОВОК

Отже у дипломній роботі проаналізовано ряд існуючих алгоритмів для розв’язання задачі ідентифікації об’єктів на основі акустичного портрету. Враховуючи сформульовані вимоги до розроблених програмних засобів, більшість із існуючих методів вирішення задачі непридатні.

Проведено дослідження перетворення Фур’є, його віконної модифікації та вейвлет-перетворення. Використано матеріал із функціонального аналізу для більш точного порівняння сигналів. Задіяно знання із обчислювального математичного аналізу для гнучкої маніпуляції графіком сигналів.

При проектуванні програмного забезпечення використано бібліотеку ВП pywt мови програмування Python. Програмне забезпечення логічно структуровано на модулі, кожен з яких виконує своє завдання. Для автоматизації підбору вхідних даних завдяки бібліотеці urllib розроблено модуль автоматичного завантаження звукових файлів по заданій url-адресі. Створено графічний інтерфейс для зручного та швидкого виконання необхідних функцій. Надається можливість автоматично аналізувати декілька файлів та отримати детальну статистику виконання обчислення.

Проведено тестування програми ручним та автоматизованим методом. Упродовж тестування ручним способом є можливість візуально прослідкувати за станом і формою вхідного та зразкового сигналу. Кожен етап обчислення за допомогою бібліотеки pylab можна простежити за допомогою графіків, що полегшує контроль за правильністю роботи програми. Автоматизований метод дозволяє отримати величину точності розпізнавання об‘єктів на основі акустичного портрету у відсотках.

Розроблене ПЗ дозволяє правильно констатувати відсутність та наявність звуку пострілу із середнім значенням імовірності 74.74%.

# СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Robi Polikar. Introduction to Wavelet : A tutorial / Robi Polikar 1st edition – Glasboro: Rowan University, 1994. – 59 ст.
2. Приложения вейвлет-анализа [Електронний ресурс]. – 2006. – Режим доступу: <https://basegroup.ru/community/articles/wavelet-applications> – Дата доступу: квітень 2016.
3. Моделирование и обработка стохастических сигналов и структур : учебное пособие / О.М.Вохник [и др.]; -М.: Универститетская книга, 2013. – 125 с.: ил.
4. Audio Fingerprinting with Python and Numpy [Електронний ресурс]. – 2013. – Режим доступу: <http://willdrevo.com/fingerprinting-and-audio-recognition-with-python> – Дата доступу: квітень 2016.
5. Існуючі підходи до вирішення задачі розпізнавання образів [Електронний ресурс]. – 2013. – Режим доступу: engbul.bmstu.ru/file/out/761264- Дата доступу: квітень 2016.
6. Discrete wavelet transform documentation [Електронний ресурс]. – 2012. – Режим доступу: <http://www.pybytes.com/pywavelets/ref/dwt-discrete-wavelet-transform.html> - Дата доступу: квітень 2016.
7. Fourier Transforms example [Електронний ресурс]. – 2016. – Режим доступу: http://docs.scipy.org/doc/scipy/reference/tutorial/fftpack.html - Дата доступу: травень 2016.