

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
імені адмірала Макарова

Навчально-науковий інститут автоматики і електротехніки

(повне найменування інституту, назва факультету)

Кафедра програмованої електроніки, електротехніки і телекомунікацій

(повна назва кафедри)

«Допущений до захисту»

Завідувач кафедри  В. М. Рябенський
д.т.н., професор

« 16 » грудня 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «магістр»

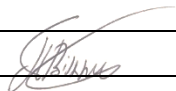
на тему: Програмна реалізація систем цифрової обробки сигналів для
аналізу та обробки аудіопотоку

Виконав: студент 6 курсу, групи 6321М
спеціальності 171

«Електроніка»

шифр і назва спеціальності)

ОПП «Електронні системи»

Пелипас В.В. 

(прізвище та ініціали студента)

Керівник Ушкаренко О.О. 

(прізвище та ініціали)

Рецензент д.т.н., проф. Черно О.О. 

(прізвище та ініціали)

м. Миколаїв – 2024 року

Національний університет кораблебудування імені адмірала Макарова

ННІ	автоматики та електротехніки
Кафедра	програмованої електроніки, електротехніки і телекомунікацій
Рівень вищої освіти	другий (магістерський)
Спеціальність	171 «Електроніка»
ОПП	Електронні системи

ЗАТВЕРДЖУЮ

Завідувач кафедри



В. М. Рябенський

д.т.н., професор

« 16 » жовтня 2024 р.

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ АТЕСТАЦІЙНУ РОБОТУ СТУДЕНТУ

Пелипасу Віктору Валентиновичу

(прізвище, ім'я, по батькові)

1. Тема роботи Програмна реалізація систем цифрової обробки сигналів для аналізу та обробки аудіопотоку

керівник роботи Ушкаренко Олександр Олегович, д.т.н., проф.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу

від « 14 » жовтня 2024 року № 1075-уч.

2. Строк подання студентом роботи 15 грудня 2024 року

3. Вихідні дані до роботи діапазон частот аудіо сигналу 20Гц-20кГц;
типи цифрових фільтрів – з кінцевою імпульсною характеристикою, з нескінченною імпульсною характеристикою; розрядність даних після аналого-цифрового перетворення 10-12 біт;

4. Мета дослідження полягає в подальшому розвитку методів програмної реалізації алгоритмів цифрової обробки аудіо сигналів за рахунок поєднання в одному програмному модулі засобів реалізації, візуалізації та валідації цифрових фільтрів та розробка інструментальних засобів для дослідження алгоритмів ЦОС

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналітичний огляд систем вимірювання аналогових сигналів. 2. Проектування апаратної частини системи аналого-цифрової обробки аудіо сигналів. 3. Реалізація алгоритмів цифрової обробки аудіо сигналів. 4. Охорона праці.

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

1. Схема підсилювача аудіо сигналу

2. Схема аналогового фільтру нижніх частот та АЦП

3. Результат спектрального аналізу білого шуму

4. Спектри аудіо сигналу після цифрової обробки різними типами фільтрів

5. Структури цифрових фільтрів для створення звукових ефектів

6. Спектр аудіо сигналу після його динамічної корекції

7. Консультанти розділів роботи

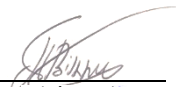
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Тубальцев А.М.		

8. Дата видачі завдання « 16 » жовтня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Аналітичний огляд інструментальних засобів для аналізу та дослідження аналогових сигналів	20.10.2023	виконано
2.	Аналіз факторів, які необхідно враховувати при проектуванні апаратної частини системи обробки аудіо сигналів	15.11.2023	виконано
3.	Проектування апаратної частини попередньої обробки аналогових сигналів звукового діапазону частот	10.03.2024	виконано
4.	Розрахунок надійності апаратної частини	12.04.2024	виконано
5.	Реалізація алгоритму частотної корекції аудіо сигналів	17.05.2024	виконано
6.	Динамічна та спектральна обробка аудіо сигналів	14.06.2024	виконано
7.	Програмна реалізація різноманітних звукових ефектів	15.09.2024	виконано
8.	Використання звукової системи персонального комп'ютера для генерування аудіо потоку	25.10.2024	виконано
9.	Частотна корекція та динамічна обробка звукових сигналів, отриманих з мікрофону	19.11.2024	виконано
10.	Підготовка розділу з охорони праці	15.12.2024	виконано

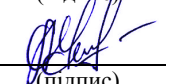
Студент


(підпис)

В.В. Пелипас

(прізвище та ініціали)

Керівник роботи


(підпис)

О.О. Ушкаренко

(прізвище та ініціали)

Анотація

В магістерській кваліфікаційній роботі, яка складається з 81 сторінки, розглянуто особливості цифрової обробки сигналів звукового діапазону частот. У системах обробки аудіо потоку розроблені програмні засоби, що зазвичай виконуються цифровими процесорами обробки сигналів, вирішують завдання аналізу, розпізнавання та синтезу мовлення, стиснення мовлення у системах телекомунікації. При обробці цифрових звукових сигналів використовувалися алгоритми цифрової фільтрації та спектрального аналізу, алгоритми кореляційного аналізу, зворотного згортки, спеціальні алгоритми лінійного передбачення.

Ключові слова: аудіо потік, цифровий фільтр, динамічна корекція, спектральний аналіз, частотна характеристика

Abstract

The master's qualification work, which has 81 pages, examines the features of digital processing of audio frequency range signals. In audio stream processing systems, software tools have been developed, usually performed by digital signal processing processors, solving problems of speech analysis, recognition and synthesis, speech compression in telecommunication systems. When processing digital audio signals, digital filtering and spectral analysis algorithms, correlation analysis algorithms, inverse convolution, special linear prediction algorithms were used.

Keywords: audio stream, digital filter, dynamic correction, spectral analysis, frequency response

ЗМІСТ

Зміст	3
Перелік умовних позначень, символів, одиниць, скорочень та термінів..	5
Вступ.....	6
Розділ 1. Аналітичний огляд систем вимірювання та аналізу аналогових сигналів	9
1.1. Порівняльна характеристика аналогових та цифрових осцилографів	9
1.2. Критерії вибору обладнання для вимірювання характеристик аудіо сигналів	10
1.3. Основні виробники вимірювальної апаратури.....	13
1.4. Методика налагодження програм.....	17
Розділ 2. Проектування апаратної частини системи аналого-цифрової обробки аудіосигналів	22
2.1. Розробка підсилювального каскаду.....	22
2.2. Проектування фільтру низьких частот.....	25
2.3. Аналого-цифровий перетворювач	28
2.4. Розрахунок надійності апаратної частини системи	31
Розділ 3. Реалізація алгоритмів цифрової обробки аудіосигналів	35
3.1. Частотна корекція сигналів	35
3.2. Динамічна і спектральна обробка звукових сигналів.....	49
3.3. Програмна реалізація різноманітних звукових ефектів	52
3.4. Використання звукової системи персонального комп'ютера.....	56
Розділ 4. Охорона праці	62
4.1. Огляд правової безпеки	62
4.2. Розрахунок захисного заземлення	63
4.3. Захист від ураження струмом	66
4.4. Заходи з техніки безпеки	67
Висновки	70
Список використаних джерел	71
Додаток А.....	73

Лістинг програми синтезу звуку	73
Додаток Б	79
Лістинг програми децимації аудіо сигналу.....	79

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ ТА ТЕРМІНІВ**

АЦП – аналогово-цифровий перетворювач
АЧХ – амплітудно-частотна характеристика
ЕОМ – електронно-обчислювальна машина
ПЛІС – програмована логічна інтегральна схема
КІХ – кінцева імпульсна характеристика
НІХ – нескінченна імпульсна характеристика
ОП – операційний підсилювач
САПР – система автоматизації проектних робіт
ФНЧ – фільтр низьких частот
ФЧХ – фазово-частотна характеристика
ЦАП – цифрово-аналоговий перетворювач
ЦОС – цифрова обробка сигналів
ЦСП – цифровий сигнальний процесор
ЦФ – цифровий фільтр
ШПФ – швидке перетворення Фур'є

					171.6321М.КР.ПЗ.Скорочення	Арк.
						5
Змн.	Арк.	№ документу	Підпис	Дата		

ВСТУП

У системах обробки звуку цифрові сигнальні процесори вирішують завдання аналізу, розпізнавання та синтезу мовлення, а також стиснення мовних сигналів у телекомунікаційних системах. Під час обробки цифрових звукових сигналів використовуються алгоритми цифрової фільтрації та спектрального аналізу, алгоритми кореляційного аналізу, зворотної згортки та спеціальні алгоритми лінійного прогнозування. Використовувані процесори цифрової обробки сигналу надають можливість дослідження спектра сигналу за допомогою аналізу із застосуванням швидкого перетворення Фур'є.

Мета дослідження полягає в подальшому розвитку методів програмної реалізації алгоритмів цифрової обробки аудіо сигналів за рахунок поєднання в одному програмному модулі засобів реалізації, візуалізації та валідації цифрових фільтрів та розробка інструментальних засобів для дослідження алгоритмів ЦОС.

Об'єкт дослідження: Цифрові сигнали та алгоритми їх обробки в аудіо-та відеосистемах.

Предмет дослідження: Методи й алгоритми цифрової фільтрації, спектрального та кореляційного аналізу для обробки звукових сигналів та зображень.

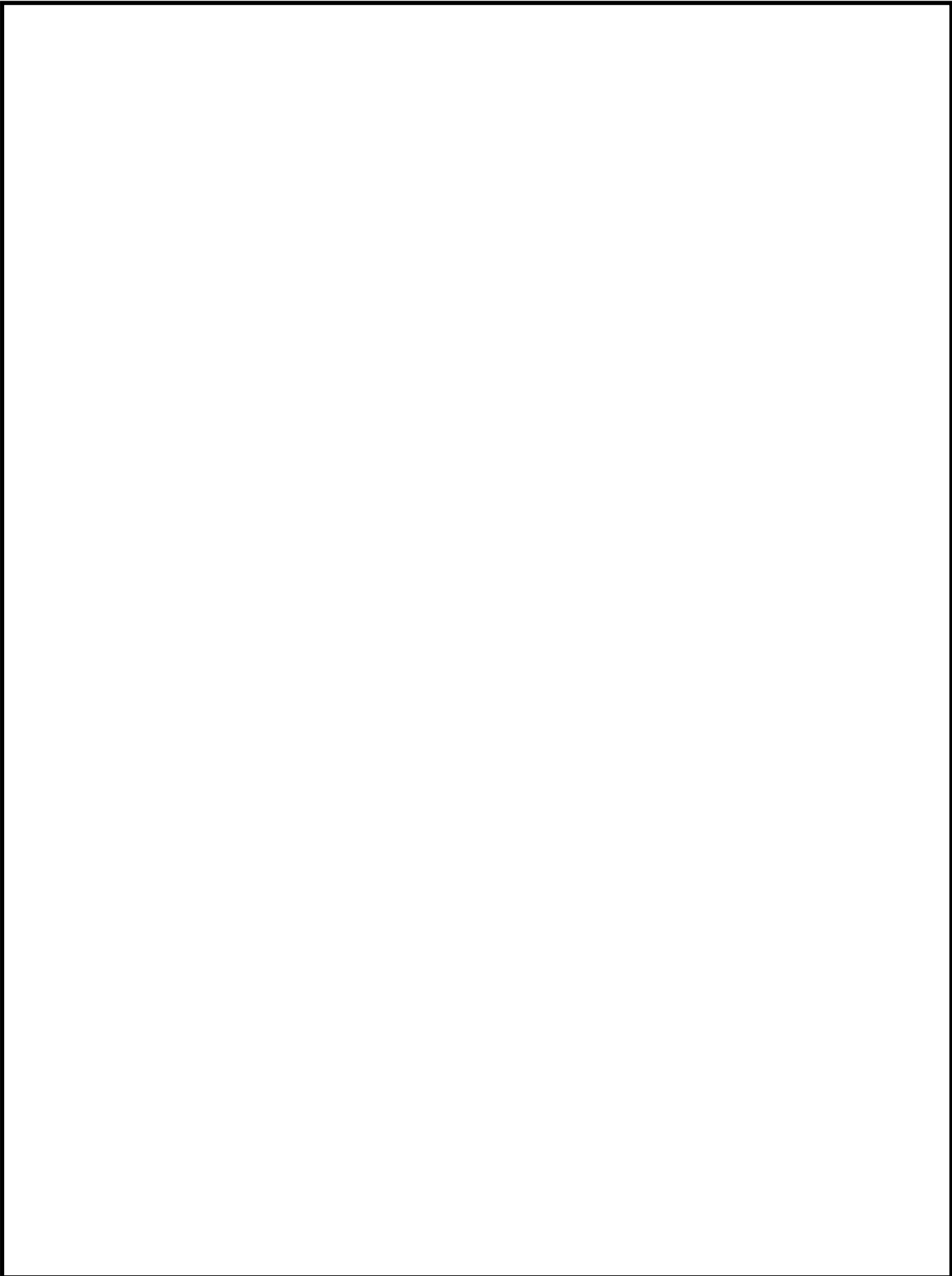
Використані методи дослідження:

- Цифрова фільтрація для виділення потрібних частотних компонентів.
- Спектральний аналіз для оцінки частотних характеристик аудіо сигналів.
- Кореляційний аналіз для виявлення схожості та закономірностей між аудіо сигналами.
- Зворотна згортка для відновлення спотворених аудіо сигналів.
- Алгоритми лінійного прогнозування для передбачення динаміки зміни аудіо сигналу.

					171.6321М.КР.ПЗ.Вступ	Арк.
						6
Змн.	Арк.	№ документу	Підпис	Дата		

Такий підхід забезпечує ефективну обробку звукових та зображувальних даних у сучасних телекомунікаційних та мультимедійних системах.

					171.6321М.КР.ПЗ.Вступ	Арк.
						7
Змн.	Арк.	№ документу	Підпис	Дата		



					171.6321М.КР.ПЗ.Р1			
Змн.	Арк.	№ документа	Підпис	Дата				
Розробник		Пелипас В.В.			Аналітичний огляд систем вимірювання та аналізу аналогових сигналів	Літ.	Арк.	Аркуші
Консультант							8	
Н. контр.		Добрінова Л. П.				НУК імені адмірала Макарова		
Керівник		Ушкаренко О.О.						
Зав. каф.		Рябенський В. М.						

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД СИСТЕМ ВИМІРЮВАННЯ ТА АНАЛІЗУ АНАЛОГОВИХ СИГНАЛІВ

1.1. Порівняльна характеристика аналогових та цифрових осцилографів

З появою на ринку цифрових осцилографів від різних виробників, вибір цього відомого інструмента стає досить складним. Просте порівняння характеристик і можливостей осцилографів не завжди допоможе визначити, який пристрій найбільше відповідатиме вашим конкретним потребам.

Одним з перших питань при виборі осцилографа, ймовірно, буде його ціна. Вартість цифрових осцилографів залежить від багатьох факторів, таких як смуга пропускання, частота дискретизації, кількість каналів та обсяг пам'яті. Якщо ви будете орієнтуватися лише на ціну, це може призвести до незадоволеності іншими характеристиками пристрою.

Цифрові та аналогові осцилографи мають свої переваги і недоліки. Постійний розвиток цифрових технологій дозволив створити цифрові прилади, які є більш потужними та продуктивними порівняно з аналоговими. Крім того, різниця в ціні між ними зменшується, і цифрові осцилографи стають дедалі доступнішими.

У таблиці 1.1 наведені переваги та недоліки цифрових і аналогових осцилографів.

Таблиця 1.1 – Порівняння аналогових та цифрових осцилографів

Вид осцилографа	Переваги	Недоліки
1	2	3
Аналоговий	Знайомий інтерфейс; миттєве відновлення екрана при відображенні сигналів, що швидко змінюються, у часі прямі; зрозумілі засоби керування для часто використ. налаштувань; низька вартість.	Низька точність; мерехтіння або тьмяність екрана залежно від частоти сигналу й коефіцієнта розгорнення; обмежена смуга пропускання; висока експлуатаційна вартість.

1	2	3
Цифровий	Можливість «заморожування» зображення на довільний час; висока точність вимірів; широка смуга пропускання; яскравий, добре сфальцьований екран на будь-якій швидкості розгорнення; можливість відображення сигналу до моменту, що запускає (в "негативному" часі) можливість виявлення імпульсних перешкод; автоматичні засоби виміру параметрів сигналів; можливість підключення до комп'ютера, принтеру або плоттеру; можливості математичної й статистичної обробки сигналу; засоби самодіагностики й самокалібрування.	Більш висока вартість, більше складні в керуванні.

1.2. Критерії вибору обладнання для вимірювання характеристик аудіо сигналів

Смуга пропускання.

Прилади, призначені для вимірювання змінних сигналів, мають максимальну частоту, після якої точність вимірювання починає знижуватися. Ця частота визначає смугу пропускання пристрою і зазвичай є тією частотою, на якій амплітуда сигналу зменшується на 3 дБ. Вибір необхідної смуги пропускання залежить від типу сигналів, які потрібно вимірювати, і бажаної точності результатів.

					171.6321М.КР.ПЗ.Р1	Арк.
						10
Змн.	Арк.	№ документу	Підпис	Дата		

Для цифрових осцилографів існують два основних типи смуги пропускання: для повторюваних сигналів (аналогова) і для однотактних сигналів. Багато цифрових осцилографів мають смугу пропускання для повторюваних сигналів, що перевищує їх частоту дискретизації. Оскільки повторюваний сигнал може бути відтворено за кілька запусків, осцилографу не потрібно оцифровувати весь сигнал одночасно, що робить цю характеристику незалежною від частоти дискретизації. Це властивість аналогових підсилювачів осцилографа. Смуга пропускання для однотактних сигналів важлива для неперіодичних сигналів, які захоплюються і оцифровуються за один такт; у цьому випадку вона залежить від частоти дискретизації. Співвідношення між частотою дискретизації і смугою пропускання для однотактних сигналів може варіюватися: з інтерполяцією — 4:1, без — 10:1.

Багато сигналів містять частотні компоненти, які значно перевищують основну частоту досліджуваного сигналу. Наприклад, прямокутний сигнал може мати компоненти, які в десять разів вищі. Осцилографи з більшою смугою пропускання забезпечать детальнішу інформацію про ці високочастотні компоненти.

Осцилограф зі смугою пропускання 500 МГц надає найдетальнішу інформацію про сигнал і краще відображає його фронти. У той час як осцилограф зі смугою 150 МГц може зрізати високочастотні компоненти, що спотворює вигляд сигналу. Прилад зі смугою 100 МГц ще більше уповільнює фронти сигналів.

Кількість каналів.

Вибір кількості каналів залежить від специфіки досліджуваного об'єкта. Найпопулярнішими є двоканальні осцилографи, проте чотирьохканальні моделі також затребувані для розширених завдань.

Частота дискретизації.

Частота дискретизації є критично важливим параметром для вимірювання однотактних або перехідних процесів. Вона вказує, з якою

					171.6321M.KP.ПЗ.Р1	Арк.
						11
Змн.	Арк.	№ документа	Підпис	Дата		

швидкістю осцилограф може оцифровувати вхідний сигнал. Вища частота дискретизації забезпечує ширшу смугу пропускання для одноканальних сигналів та кращу роздільну здатність. Більшість виробників дотримуються співвідношення 4:1 (з інтерполяцією) або 10:1 (без неї), щоб уникнути спотворення сигналу. Оскільки обсяг пам'яті осцилографа обмежений, максимальна частота дискретизації досягається тільки при найшвидших швидкостях розгорнення; на повільніших швидкостях частота автоматично зменшується.

Обсяг пам'яті.

Необхідний обсяг пам'яті залежить від тривалості сигналу і бажаної роздільної здатності. Для тривалих сигналів з високою роздільною здатністю потрібен більший обсяг пам'яті, що дозволяє підтримувати високу частоту дискретизації на повільніших швидкостях розгорнення, зменшуючи ризик спотворення сигналу і отримуючи більше інформації. Формула для розрахунку обсягу пам'яті виглядає так:

Обсяг пам'яті = проміжок часу (с) / роздільна здатність (с).

Недолік осцилографів з великим обсягом пам'яті полягає в тому, що це може знизити їх реакцію на дії оператора або зміни вхідного сигналу, що робить їх менш зручними у використанні.

Виявлення імпульсних перешкод.

На здатність осцилографа виявляти імпульсні перешкоди впливають три ключові фактори:

1. Частота відображення сигналів. Цифрові осцилографи спочатку оцифровують вхідний сигнал, обробляють його і відображають на екрані. Частота відображення визначає, як швидко виконуються ці операції. Прилади з високою частотою відновлення мають кращу здатність виявляти рідкісні імпульсні перешкоди. Осцилографи з багатопроцесорною архітектурою можуть мати частоту відновлення, що значно перевищує традиційні однопроцесорні моделі, що робить їх більш ефективними для виявлення імпульсів.

					171.6321M.KP.ПЗ.Р1	Арк.
						12
Змн.	Арк.	№ документа	Підпис	Дата		

2. Можливості визначення амплітуди. Багато цифрових осцилографів знижують частоту дискретизації на повільних швидкостях розгорнення, що може призвести до втрати коротких імпульсів. Спеціальний режим "виявлення пікової амплітуди" дозволяє підтримувати максимальну частоту дискретизації на всіх швидкостях, записуючи мінімальні і максимальні значення вибірок.

3. Можливості запуску по імпульсній перешкоді. Осцилографи, що підтримують запуск по імпульсній перешкоді, дають змогу ізолювати важко виявлені імпульсні перешкоди та ініціювати запис осцилографа по цих сигналах, що допомагає в діагностиці ненормальної роботи схем.

Аналіз сигналів.

Функції автоматичних вимірів та вбудовані засоби аналізу можуть суттєво зекономити час і спростити роботу. Цифрові осцилографи пропонують безліч функцій, недоступних на аналогових моделях. Математичні операції, такі як додавання, віднімання, множення, ділення, інтеграція та диференціювання, дозволяють проводити глибший аналіз. Статистичні дані (мінімальні, максимальні та середні значення) допомагають оцінити точність вимірів, особливо при аналізі шумових сигналів. Деякі цифрові осцилографи також здатні проводити частотний аналіз сигналів за допомогою швидкого перетворення Фур'є.

1.3. Основні виробники вимірювальної апаратури

Сучасні осцилографи представляють собою інноваційні інструменти, які використовують передові технології і перетворилися з простих приладів для відображення тимчасових залежностей напруги на екрані в багатофункціональні інтегровані контрольно-вимірювальні системи. Завдяки інтеграції осцилографів із такими популярними пристроями, як мультиметри, функціональні перетворювачі, частотоміри, логічні та спектральні аналізатори, користувачі можуть легко і швидко виконувати різноманітні завдання. Прагнення залучити більше потенційних клієнтів спонукає

					171.6321M.KP.ПЗ.Р1	Арк.
						13
Змн.	Арк.	№ документа	Підпис	Дата		

розробників осцилографів розширювати функціональність своїх пристроїв і пропонувати вигідне співвідношення ціни та якості.



Рисунок 1.1 – Зовнішній вигляд цифрових осцилографів

Ринок осцилографів в цілому домінують американські виробники, які займають понад 70% світових продажів. Обсяги продажів постійно зростають і щорічно складають від 50 до 70 мільйонів доларів. Значний приріст продажів у останні роки спостерігається в Азіатсько-Тихоокеанському регіоні. Основними споживачами осцилографів є різні промислові галузі, такі як комунікації, автомобільна промисловість, виробництво побутової електроніки та високочастотного обладнання, розробка напівпровідникових матеріалів і мікросхем, аерокосмічна, військова та оборонна промисловість. Нові технології у сферах зв'язку та передачі даних стимулюють удосконалення інтернет-додатків, поширення широкосмугових систем, волоконно-оптичних ліній зв'язку та бездротових технологій. Це зумовлює необхідність створення нових протоколів і використання більш швидких і складних електронних сигналів, а отже, й відповідних приладів для їхнього аналізу. Успіх будь-якої компанії, що займається розробкою та продажем контрольно-вимірювального обладнання, залежить від здатності швидко створювати та впроваджувати в ринок затребувані продукти.



Рисунок 1.2 – Стаціонарний осцилограф

Останнім часом виробництво частково перемістилося в Китай. Вся продукція компанії має сертифікацію, включаючи ISO 9001:2000. Компанія стабільно займає провідні позиції на ринку контрольно-вимірювальних приладів, виготовляючи обладнання як для електроніки, так і для суміжних галузей. Це включає осцилографи змішаних сигналів, спектральні аналізатори в реальному часі, генератори сигналів різних форм, логічні аналізатори та різноманітні пробники, які розширюють функціональність вимірювальних приладів Tektronix для роботи з різними вхідними сигналами.

Agilent Technologies була заснована в 1939 році Біллом Х'юлеттом і Дейвом Паккардом як компанія Hewlett-Packard, що стала основою для розвитку Силіконової долини. У 1999 році Agilent Technologies, Inc. стала незалежною компанією, яка на сьогодні пропонує контрольно-вимірювальне обладнання, системи автоматизованого контролю, програмне забезпечення для електронного моніторингу, інтерфейсні плати та підтримку осцилографів. У фінансовому році 2007 обсяги продажів Agilent Technologies перевищили 5,4 мільярда доларів, що підтверджує її статус одного з лідерів у галузі вимірювальної техніки. Основна діяльність компанії зосереджена на розробці та виробництві приладів для прийому, аналізу, відображення та передачі даних, а також вимірювального обладнання для біотехнологій та хімічного аналізу.

					171.6321M.KP.ПЗ.P1	Арк.
						15
Змн.	Арк.	№ документу	Підпис	Дата		

LeCroy спеціалізується на виробництві цифрових осцилографів, що складають 95% її асортименту, включаючи прикладні та спеціалізовані моделі. Компанія має великі підрозділи в Женеві (Швейцарія) і Токіо (Японія), а також представництва в інших європейських та азіатських країнах. LeCroy також розробляє, виготовляє, продає та патентує обладнання для прецизійних досліджень, яке складає 15% її продукції. Вироби LeCroy забезпечують можливість захоплення і обробки цифрових потоків з високою швидкістю та частотою дискретизації, а також мають розширений обсяг пам'яті для детального аналізу форм сигналів. Компанія також розробляє додаткові засоби, які допомагають операторам краще розуміти характеристики сигналів.

RIGOL Technologies, Inc. — швидко зростаюча китайська компанія, заснована в 1998 році, яка спеціалізується на виробництві вимірювальної техніки високого класу. Вона здобула світову популярність завдяки успішним проектам у створенні цифрових осцилографів, ставши першою китайською компанією, що випустила осцилографи з частотою дискретизації 1 ГГц. Серія DS5000CA стала основою для виробництва осцилографів DSO3000 для Agilent Technologies. У 2006 році RIGOL зайняла друге місце у світі за випуском цифрових осцилографів бюджетного класу, а її змішані сигналові осцилографи DS1000CD отримали нагороду журналу EDN як «Кращий продукт 2006 року» у категорії вимірювального обладнання, ставши першим виробником з Азії, який здобув цю відзнаку.



Рисунок 1.3 – Осцилограф з ЕПТ

					171.6321M.KP.ПЗ.P1	Арк.
Змн.	Арк.	№ документу	Підпис	Дата		16

1.4. Методика налагодження програм

Після успішного завантаження програмних модулів, їх розміщення за визначеними адресами та взаємозв'язку, для налагодження програми можна використовувати кілька методів:

- внутрішньосхемний загальнокласовий;
- вбудований програмний відладчик;
- зовнішній програмний відладчик;
- пристрій, що налагоджується, із записаним двійковим кодом програми.

Внутрішньосхемний загальнокласовий метод, що відображає змінні мови програмування на дисплеї комп'ютера, суттєво полегшує налагодження безпосередньо на апаратному рівні. Цей спосіб надає зручне середовище для зупинки програми, контролю виконання за вихідним кодом, а також стану зовнішніх портів і внутрішніх змінних, що визначені як у мікросхемі, так і в коді. Важливо зазначити, що для використання цього методу налагодження необхідно включати символічну інформацію в об'єктні модулі.

Раніше внутрішньосхемний загальнокласовий відладчик був окремим пристроєм, підключеним до плати замість мікроконтролера. Сучасні мікроконтролери часто містять вбудований внутрішньосхемний загальнокласовий. Найзручнішим інтерфейсом для зв'язку з комп'ютером є JTAG, хоча деякі виробники пропонують свої варіанти.

При використанні інтегрованого середовища програмування, розробникам надається зручний інтерфейс для налагодження. Наразі стандартом де-факто є інтерфейс, схожий на оболонку Visual C, приклад якої наведено на рисунку 1.4.

					171.6321M.KP.ПЗ.Р1	Арк.
						17
Змн.	Арк.	№ документу	Підпис	Дата		

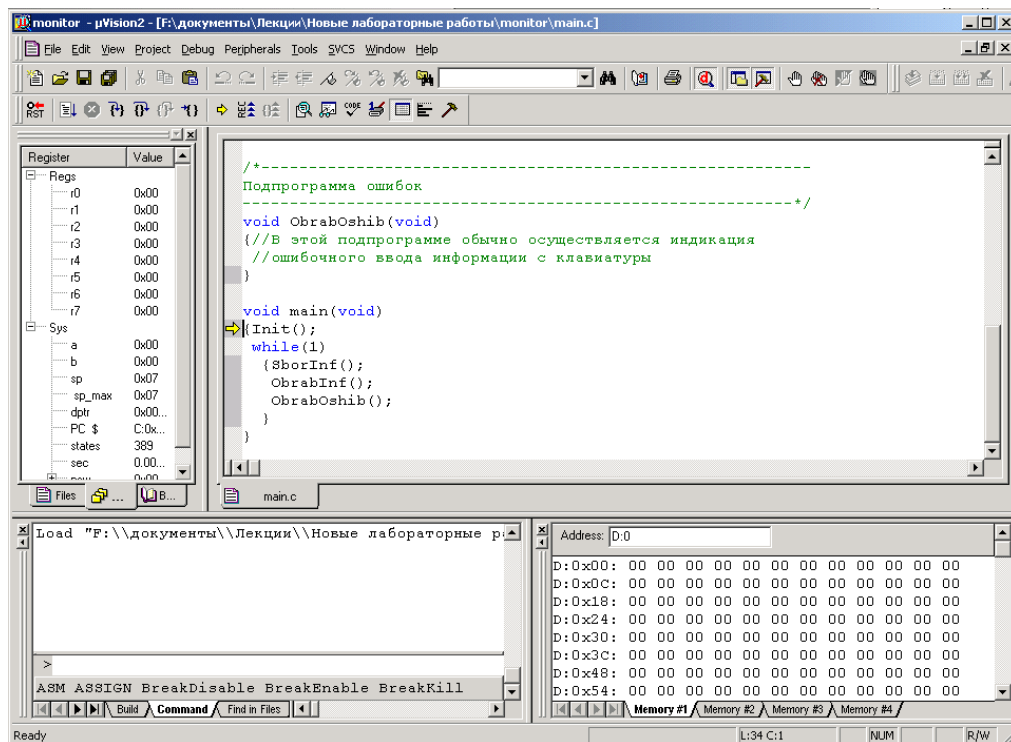


Рисунок 1.4 – Приклад зовнішнього вигляду відлагоджувача інтегрованої системи налагодження програмного забезпечення

Вбудований програмний відладчик, що є частиною інтегрованого середовища програмування, не дозволяє контролювати роботу апаратури, що підключена до зовнішніх виводів мікроконтролера. Проте він суттєво знижує вартість необхідного відладочного комплексу для написання програм для мікроконтролерів. Сигнали, які повинні надходити до мікроконтролера, визначаються програмістом, що може призвести до помилок через неправильне розуміння роботи апаратури. Ручне введення цих сигналів істотно уповільнює процес налагодження програмного забезпечення.

Зовнішній програмний відладчик використовується в ситуаціях, коли не застосовується інтегроване середовище програмування, а тільки окремий транслятор для вибраної мови. У цьому випадку можна використовувати будь-яку програму, що емулює відповідний тип мікроконтролера. Оскільки об'єктні формати різних трансляторів можуть дещо відрізнятися, вхідним файлом часто є завантажувальний модуль у двійковому або шістнадцятковому форматах, в яких відсутня налагоджувальна інформація.

Тому налагодження таких програм можна проводити лише за допомогою вбудованого дизасемблера та роздрукованого (або відкритого в іншому вікні) виходу програми. Це робить цей метод ще менш зручним у порівнянні з використанням інтегрованого середовища програмування та внутрішньосхемного відладчика.

Програмування мікросхеми передбачає запис готового завантажувального модуля у пам'ять мікроконтролера за допомогою програматора, що входить до складу багатьох сучасних мікроконтролерів. Після цього здійснюється ретельне тестування розробленого пристрою для виявлення можливих помилок у схемі та програмному забезпеченні. Тільки після успішного завершення тестування програма вважається повністю написаною та налагодженою.

У випадках, коли програми є досить простими, налагодження може обмежуватися лише цим фінальним етапом тестування. Проте виявлення, пошук та усунення помилок за допомогою тільки цього методу є дуже трудомістким процесом, подібно до налаштування апаратури без спеціальних інструментів.

Налагодження програм полягає у перевірці їхньої правильності та функціонування апаратури. Програма може бути вільною від синтаксичних помилок, але все ж містити логічні помилки, які заважають їй виконувати свої функції. Логічні помилки можуть бути спричинені неправильним алгоритмом або неправильним розумінням роботи апаратури, що підключена до мікроконтролера.

Вбудований відладчик інтегрованого середовища програмування дозволяє налагоджувати код, який не залежить від зовнішньої апаратури, що не є частиною мікроконтролера. Це, як правило, стосується математичних обчислень або конвертацій форматів даних.

Для налагодження програм використовуються три основні методи:

Покрокове налагодження з заходом у підпрограми;

Покрокове налагодження з виконанням підпрограми як єдиного

					171.6321M.KP.ПЗ.Р1	Арк.
						19
Змн.	Арк.	№ документа	Підпис	Дата		

оператора;

Виконання програми до встановленої точки зупинки.

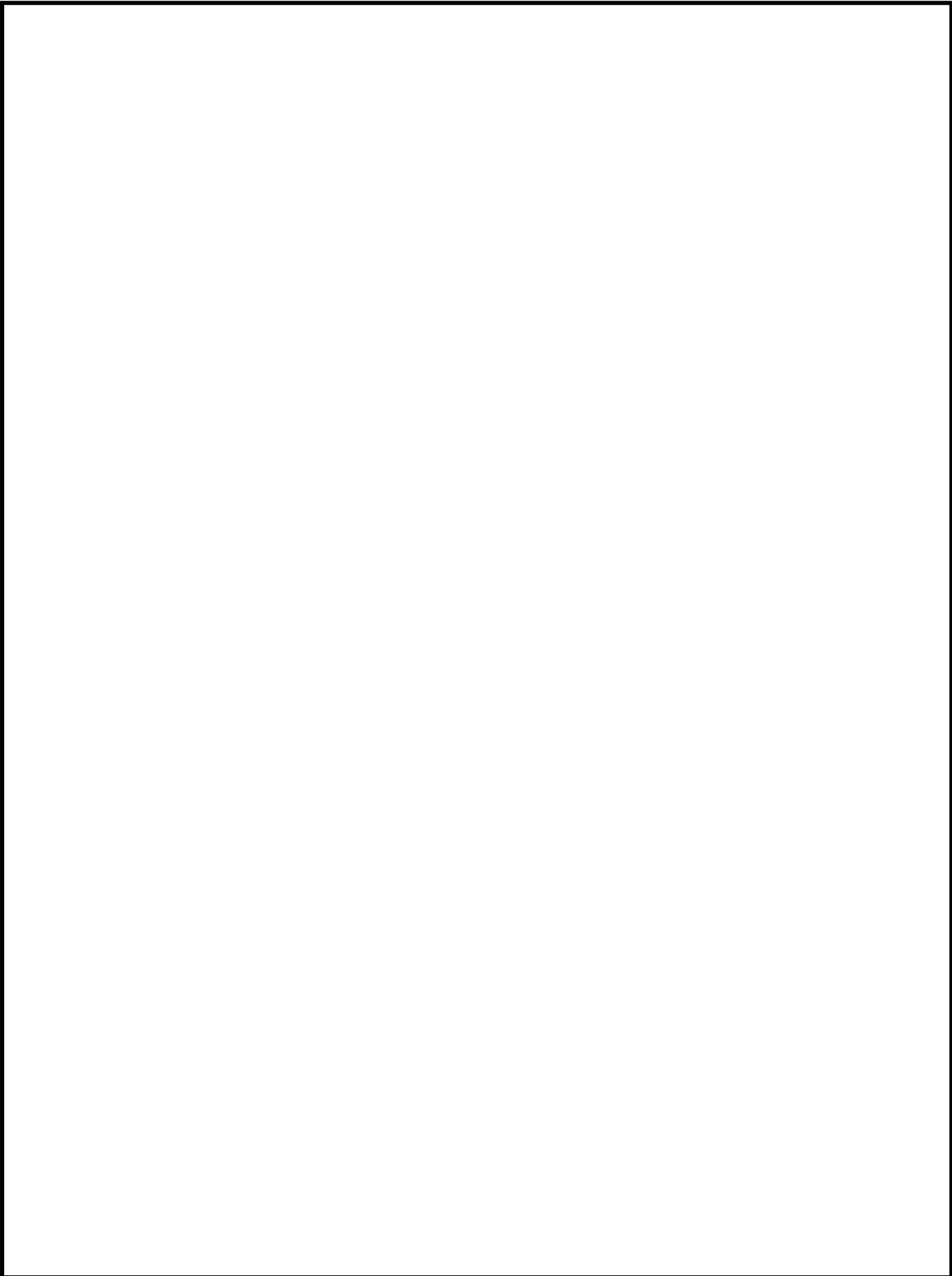
Покрокове налагодження полягає в поетапному виконанні операторів програми з подальшим контролем змінних, на які вони впливають. Якщо програма містить вже перевірені підпрограми, їх можна розглядати як один оператор і використовувати другий метод налагодження.

У випадку, коли в програмі є велика перевірена частина, її можна виконати без контролю змінних, на які вона впливає. Використання точок зупинки дозволяє пропустити вже налагоджені фрагменти програми. Точки зупинки встановлюються в місцях, де потрібно перевірити змінні або контролювати передачу управління.

Усі сучасні відладчики підтримують цю функцію, а також виконання програми до курсору та вихід з підпрограми. Після цього налагодження продовжується в покроковому режимі з контролем локальних та глобальних змінних, а також внутрішніх реєстрів мікроконтролера і напруги на його виходах.

Контроль даних під час покрокового налагодження. Вбудований відладчик дозволяє контролювати змінні за допомогою вікна Watch, що зазвичай зручніше, ніж перегляд пам'яті. У цьому вікні змінні відображаються у форматі, в якому вони були оголошені в програмі, а перетворення даних здійснюється вбудованими засобами відладчика. Якщо відладжувальна інформація втрачена або програма написана на асемблері, стан змінних можна перевірити, переглянувши вміст внутрішньої та зовнішньої пам'яті мікроконтролера. Для визначення конкретної комірки пам'яті, зайнятої під змінну, можна скористатися лістингом, що створюється редактором зв'язків компілятора або самим компілятором. Більшість відладчиків також дозволяють перевіряти вміст робочих реєстрів поточного банку.

					171.6321M.KP.ПЗ.Р1	Арк.
						20
Змн.	Арк.	№ документу	Підпис	Дата		



					171.6321М.КР.ПЗ.Р2			
Змн.	Арк.	№ документа	Підпис	Дата				
Розробник		Пелипас В.В.			Проектування апаратної частини системи аналого-цифрової обробки аудіосигналів	Літ.	Арк.	Аркуші
Консультант							21	
Н. контр.		Добрінова Л. П.				НУК імені адмірала Макарова		
Керівник		Ушкаренко О.О.						
Зав. каф.		Рябенський В. М.						

РОЗДІЛ 2. ПРОЕКТУВАННЯ АПАРАТНОЇ ЧАСТИНИ СИСТЕМИ АНАЛОГО-ЦИФРОВОЇ ОБРОБКИ АУДИОСИГНАЛІВ

2.1. Розробка підсилювального каскаду

Сигнал, який необхідно виміряти, через кабель (50-70 Ом) надходить на вхід J1. Конденсатор C1 (рис. 2.1) фільтрує постійну складову змінної напруги. Ланцюг C2, C3 визначає вхідну ємність осцилографа.

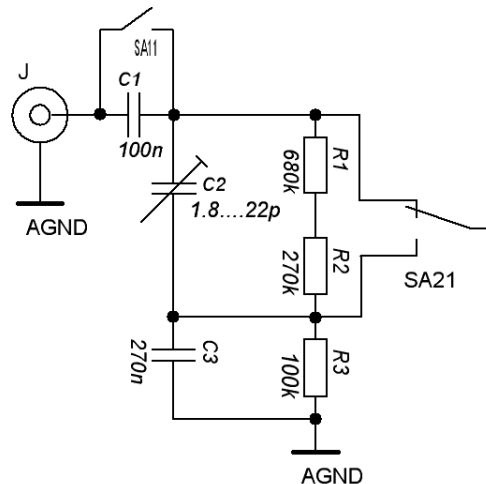


Рисунок 2.1 – Вхідний каскад

При роботі в режимі постійної напруги сигнал подається відразу на дільник, а при роботі зі змінною напругою для відсікання постійних складових сигналу на вхід ставиться конденсатор. Конденсатор має комплексний опір

$$Z_c = \frac{1}{j \cdot 2 \cdot \pi \cdot f \cdot C} \quad (2.1)$$

На частоті 10 МГц при $C = 100$ нФ $Z_c \rightarrow 0$, що нам і потрібно.

Зробимо розрахунок дільника напруги для ослаблення вхідного сигналу по амплітуді.

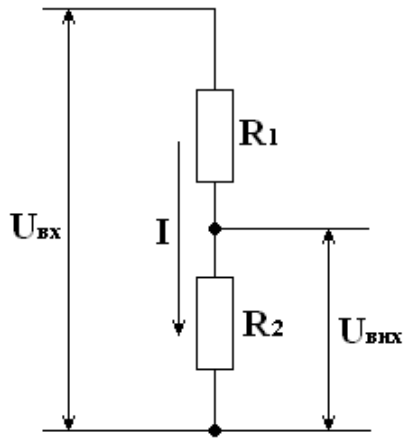


Рисунок 2.2 – Дільник напруги

Припустимо, що навантаження немає, тоді за законом Ома

$$I = \frac{U_{\text{вх}}}{R_1 + R_2}, \quad (2.2)$$

а

$$U_{\text{вих}} = I \cdot R_2 = \frac{R_2}{R_1 + R_2} \cdot U_{\text{вх}}. \quad (2.3)$$

Для різних дільників буде мінятися коефіцієнт ослаблення

$$K_{\text{осл}} = \frac{R_2}{R_1 + R_2}. \quad (2.4)$$

В нашому випадку $R_1 = 100 \text{ кОм}$, $R_2 = 950 \text{ кОм}$. Отже $K_{\text{осл}} = 0,1$.

Резистори на високих частотах мають паразитну ємність на підложку. Дротові резистори поведуться добре на частотах до 50 кГц, вуглецеві резистори використовуються на частотах до 1 МГц. Для даної схеми потрібно використовувати плівкові резистори, які мають стабільні параметри на частотах до 100 МГц.

Передпідсилювачем електричних сигналів даного каскаду є мостовий підсилювач, зібраний на польових транзисторах VT1, VT2 (КП303Е, КП103Е). Транзистор VT1 підсилює позитивну напівхвилю досліджуваного сигналу, VT2 - негативну. Щоб уникнути порушення нормальної роботи транзисторів або їхнього пробоя від статичної напруги, ставляться захисні діоди VD1, VD2 (1N4148). У ролі захисту від високочастотних перешкод по живленню виступають конденсатори C4, C5.

					171.6321М.КР.ПЗ.Р2	Арк.
						23
Змн.	Арк.	№ документа	Підпис	Дата		

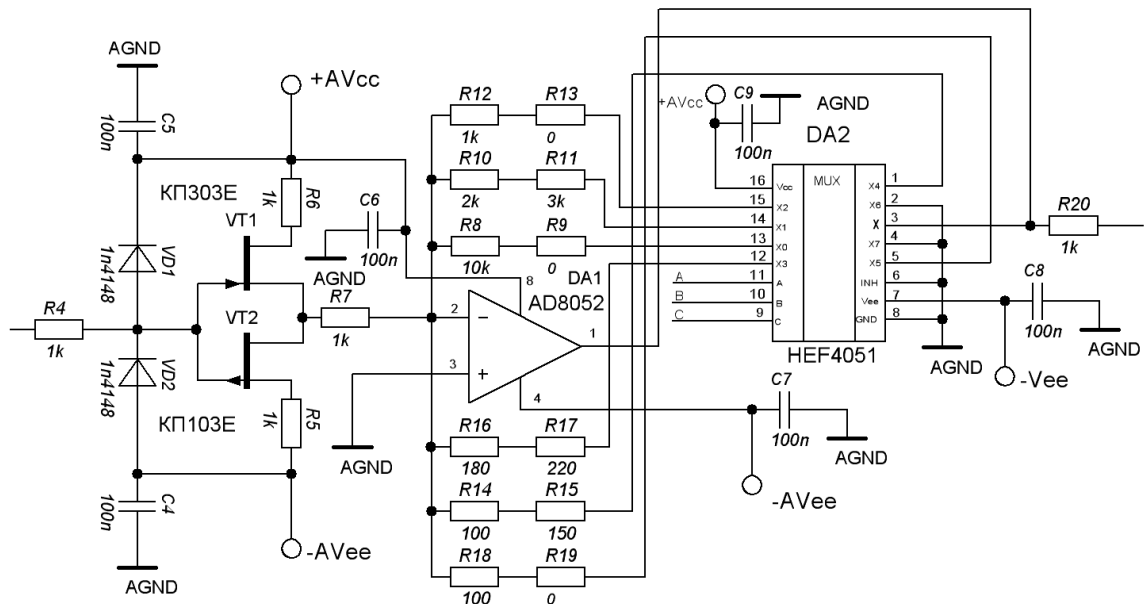


Рисунок 2.3 – Схема підсилювача аудіо сигналу

Для посилення досліджуваного сигналу використовуємо ОП AD8052. У даній схемі використовуємо інверсну схему включення ОП.

У цій схемі ОП охоплений паралельним негативним зворотним зв'язком по напрузі. На інверсний вхід подається сигнал, обумовлений сумою вхідної й вихідної напруг ділянкою на опорах R_1 , R_2 . Тому що не інверсний вхід ОП з'єднаний із загальним виводом, а $U_{диф.} \approx 0$, то інверсна напруга на вході також дорівнює нулю. У результаті для схеми можна записати рівняння:

$$\frac{U_{вх}}{R_1} = -\frac{U_{вих}}{R_2}. \quad (2.5)$$

Звідки знаходимо коефіцієнт підсилення підсилювача:

$$K = \frac{U_{вих}}{U_{вх}} = -\frac{R_2}{R_1}. \quad (2.6)$$

Вхідний опір інверсного підсилювача визначається у вигляді

$$r_{вх\text{ инв.}} = R_1 + r_{вх.диф.} \parallel [R_2 / (K+1)], \quad (2.7)$$

де другий доданок – опір паралельно включених вхідного опору ОП й зменшеного в $(K+1)$ раз опору резистора зворотного зв'язку R_2 . Приблизно $r_{вх\text{ инв.}} \approx R_1$.

Значення коефіцієнтів підсилення заносимо в таблицю 2.1.

Таблиця 2.1.

Позначення резисторів на схемі	$R1, \text{Ом}$	$R2, \text{Ом}$	коефіцієнт підсилення K
$R8- R9$	1000	10000	10
$R10- R11$	1000	5000	5
$R12- R13$	1000	1000	1
$R14- R15$	1000	250	0,25
$R16- R17$	1000	400	0,4
$R18- R19$	1000	100	0,1

«Інтелектуальний» підсилювач, зібраний на ОП $DA1$ мультиплексорі $DA2$, служить для посилення або ослаблення досліджуваного сигналу, щоб досліджуваний сигнал відображався на дисплеї, що не виходить за рамки й занадто малим. Це досягається шляхом перемикання відповідної вітки з опором негативного зворотного зв'язку ОП. Перемикачем служить мультиплексор HEF4051 із двонарпавленими входами. Перемикання відбувається шляхом подачі відповідної комбінації імпульсів на розв'язні входи мультиплексора А, В, С. Подача імпульсів управляється мікроконтролером. Для захисту $DA1, DA2$ від високочастотних перешкод по живленню ставляться конденсатори $C6-C9$.

2.2. Проектування фільтру низьких частот

Фільтр низьких частот (ФНЧ) усуває ефект накладення спектрів та небажані сигнали, що виходять за межі його смуги пропускання. Оскільки цей фільтр обробляє аналогові сигнали, він також є аналоговим пристроєм.

Коли йдеться про дискретизацію низькочастотного сигналу, мається на увазі, що сигнал, який підлягає дискретизації, розташований у першій зоні Найквіста. Важливо зазначити, що без попередньої фільтрації на вході ідеального дискретизатора будь-який частотний компонент (як сигнал, так і шум), що перевищує "смугу Найквіста", може створити низькочастотну складову в першій зоні Найквіста. Саме тому ФНЧ зазвичай використовується з усіма аналого-цифровими перетворювачами (АЦП) для придушення небажаних сигналів.

Згідно з теоремою Котельникова, частота дискретизації (f_d) повинна бути вдвічі більшою за верхню частоту сигналу (f_v). Проте будь-який сигнал має нескінченний спектр, що робить фільтрацію необхідною.

Ключовим моментом є правильне формулювання вимог до характеристик аналогового фільтра, що обмежує спектр сигналу на вході АЦП. Фільтр має смугу пропускання і смугу непропускання. Його слід реалізувати так, щоб амплітудні та фазові спотворення в смузі пропускання були мінімальними, а на частоті ($f_d - f_v$) – забезпечувати достатнє ослаблення, щоб сигнал не перевищував рівень шуму квантування АЦП.

При розробці ФНЧ орієнтуються на необхідне ослаблення на частоті ($f_d - f_v$) і допустимі параметри фазового та амплітудного спотворення. Спектр дискретизованого сигналу без використання ФНЧ продемонстрований на рисунку 2.2.

					171.6321М.КР.ПЗ.Р2	Арк.
						26
Змн.	Арк.	№ документу	Підпис	Дата		

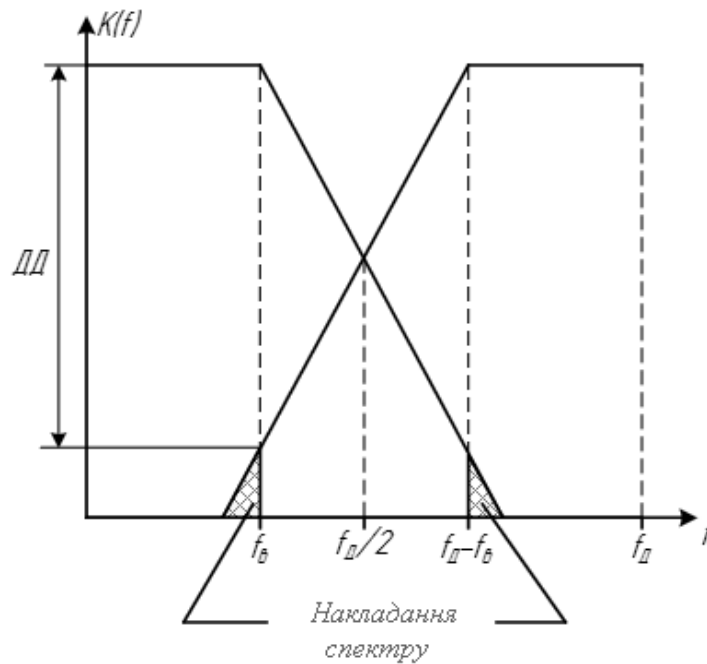


Рисунок 2.2 – Частотна характеристика фільтра

Необхідне загасання аналогового фільтра в смузі затримування визначається динамічним діапазоном корисного сигналу. Динамічний діапазон цифрового пристрою обирається на основі заданої точності відображення сигналу. При цьому нижня межа динамічного діапазону залежить від рівня всіх перешкод, що потрапляють у смугу частот корисного сигналу.

Різні типи фільтрів мають різну крутість спаду та частотні характеристики. Наприклад, фільтр Баттерворта забезпечує ослаблення 6 дБ на октаву для кожного порядку. Щоб досягти ослаблення 60 дБ у діапазоні від 1 МГц до 2 МГц (1 октава), необхідно використовувати фільтр не нижче 10-го порядку, що є досить складним у розробці. Тому для високошвидкісних застосувань, які вимагають більшої крутості спаду та менших спотворень у робочій смузі при лінійній фазовій характеристиці, доцільніше використовувати фільтри інших типів.

Шум квантування SNR (у дБ) в N -Розрядному АЦП перебувати по наступній формулі:

$$SNR = 6,02N + 1,76 \text{ дБ.} \quad (2.9)$$

Виходить, в 8-розрядному АЦП на частоті ($f_d - f_v$) повинне бути ослаблення:

$$SNR = 6,02 \cdot 8 \text{ дБ} + 1,76 \text{ дБ} = 49,92 \text{ дБ}. \quad (2.10)$$

Ще одна важлива вимога до ФНЧ – лінійність фазо-частотної характеристики (ФЧХ) у смузі пропускання або постійний час групової затримки сигналу. Нерівномірність амплітудної характеристики може бути надалі скомпенсована цифровою фільтрацією.

Схема фільтру нижніх частот зображена на рис. 2.3.

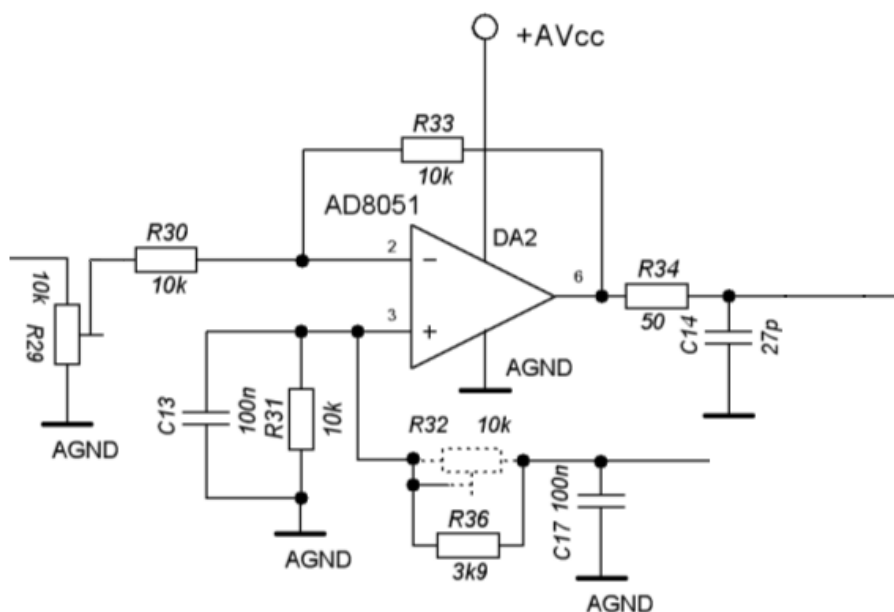


Рисунок 2.3 – Схема аналогового ФНЧ

2.3. Аналого-цифровий перетворювач

Перетворення аналогового сигналу в цифровий показано на схемі рис.

2.4.

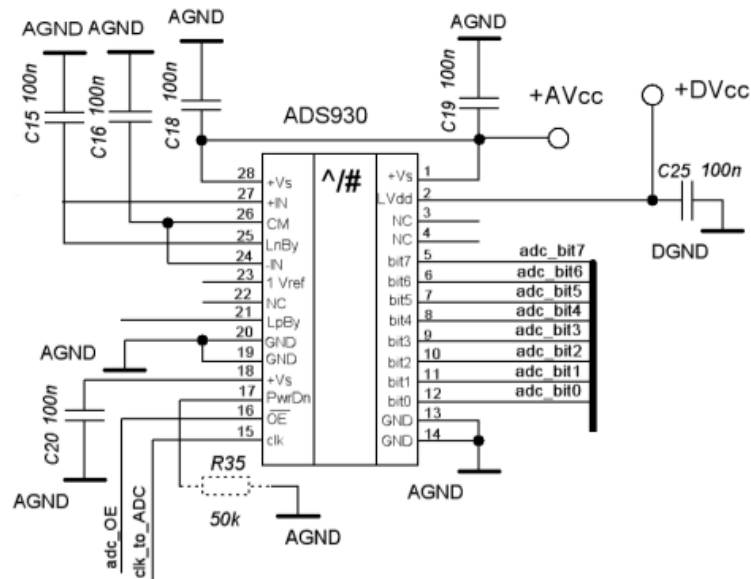


Рисунок 2.4 – Схема перетворення аналогового сигналу в цифровий

Розглянемо детальніше процеси відцифровки аналогового сигналу.
На рис. 2.5 зображена структурна схема мікросхеми АЦП.

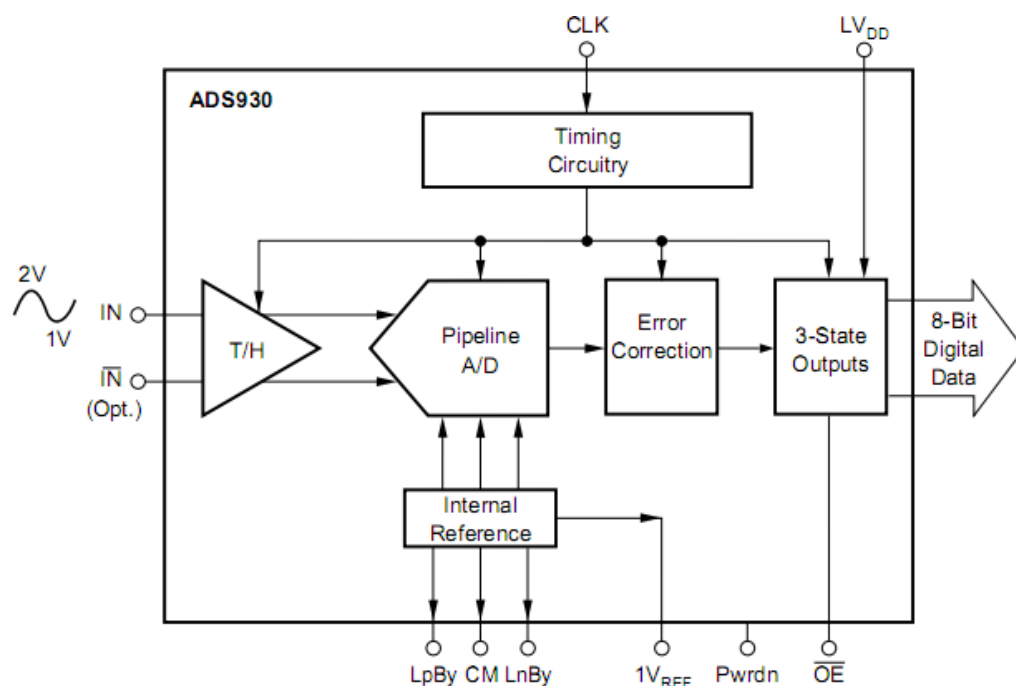


Рисунок 2.5 – Мікросхема АЦП

На рис. 2.6 зображені часові діаграми роботи АЦП.

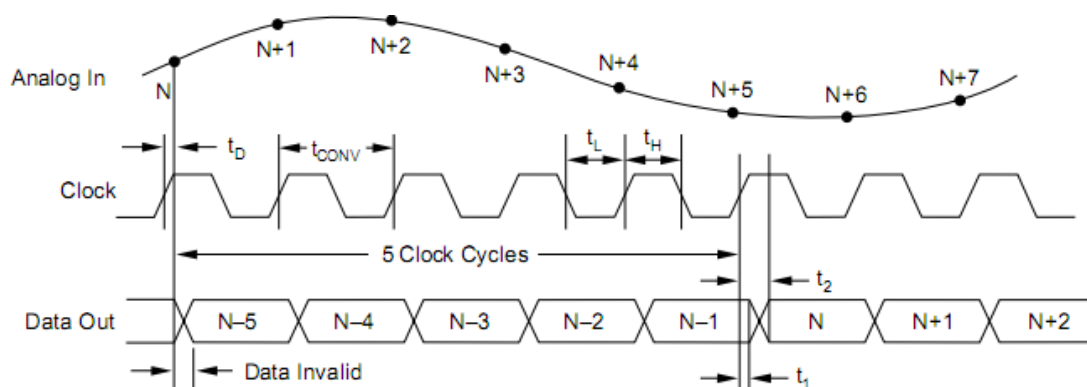


Рисунок 2.6 – Діаграми роботи АЦП

Таблиця 2.2 – Характеристики АЦП

Позначення	Пояснення	t_{min} , нс	t_{typ} , нс	t_{max} , нс
t_{conv}	Час перетворення сигналу	33		10^5
t_L	Час низького рівня імпульсу	15,5	16,5	
t_H	Час високого рівня імпульсу	15,5	16,5	
t_D	Затримка		2	
t_1	Часова затримка, при $C_L=0$ pF	3,9		
t_2	Часова затримка, при $C_L=15$ pF			12

ADS930 - висока швидкість, що робить вибірку аналого-цифрового конвертера, зробленого по конвеєрній архітектурі. Повністю диференціальна топологія й цифрове усунення помилки гарантують 8-бітову розв'язну здатність. Простежимо роботу одного каналу через еквівалентну схему, показану на рис. 2.7.

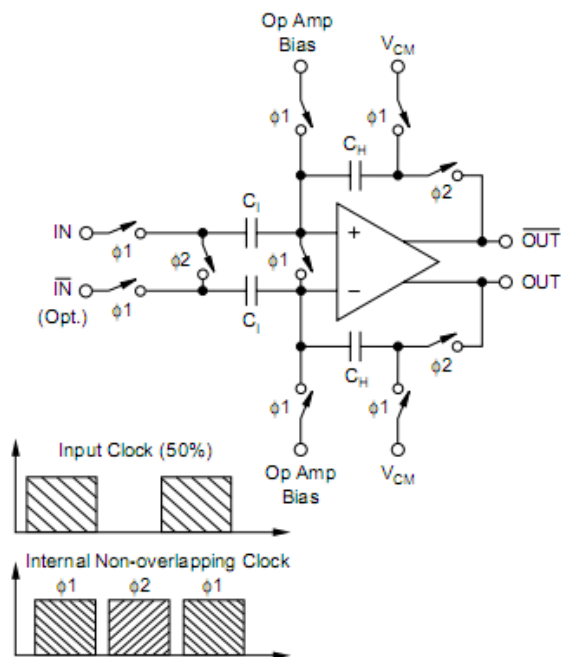


Рисунок 2.7 – Еквівалентна схема

Вимикачами управляє внутрішній тактовий генератор, у якого є неперекривання на два сигнали фази, $\phi 1$ і $\phi 2$. Час вибірки вхідного сигналу визначається вхідними конденсаторами. У наступній фазі тактового генератора, $\phi 2$, клеми вхідних конденсаторів підключені разом, і конденсатори зворотного зв'язка переключені до Op Amp виводу. У цей час заряд перерозподіляється між C_1 і C_H , завершуючи кожний track/hold цикл. У нормальному режимі роботи, додатковому введення прив'язане до загальної напруги. У цьому випадку, track/hold канал перетворює асиметричний вхідний сигнал у повністю диференціальний сигнал для квантизатора. Отже, вхідний сигнал підсилюється у два рази, що поліпшує продуктивність сигналу до переключування. Інші параметри, такі як малий сигнал, смуга пропускання, і широкополосе переключування також визначені в цій стадії.

2.4. Розрахунок надійності апаратної частини системи

Для визначення вірогідності безвідмовної роботи системи потрібно знати потоки відмов кожного з елементів схеми. Потоки відмов елементів схеми зведені в таблиці 2.4.

					171.6321M.KP.ПЗ.P2	Арк.
						31
Змн.	Арк.	№ документа	Підпис	Дата		

Таблиця 2.4 – Потоки відмов елементів

Елементи	$\lambda_i \cdot 10^{-5},$ годин ⁻¹	кількість
Напівпровідникові прилади	0,5	18
Резистори	0,25	33
Конденсатори	0,15	44
Кварц	0,03	2

Сумарна інтенсивність відмов визначається за формулою:

$$\lambda_0 = \sum_{i=1}^n n_i \cdot \lambda_i, \quad (2.11)$$

де n_i – кількість елементів, λ_i – інтенсивність відмов елементів.

$$\lambda_0 = 0,5 \cdot 18 + 0,25 \cdot 33 + 0,15 \cdot 44 + 0,03 \cdot 2 = 23,91 \cdot 10^{-5} \text{ (год.}^{-1}\text{)}.$$

Середній час безвідмовної роботи дорівнює:

$$m_t = \frac{1}{\lambda_0} = \frac{1}{23,91 \cdot 10^{-5}} = 4182 \text{ (год.)}.$$

Вірогідність безвідмовної роботи розраховується за формулою:

$$P(t) = e^{-\lambda_0 \cdot t}. \quad (2.12)$$

На рис. 2.8 представлено графік вірогідності безвідмовної роботи.

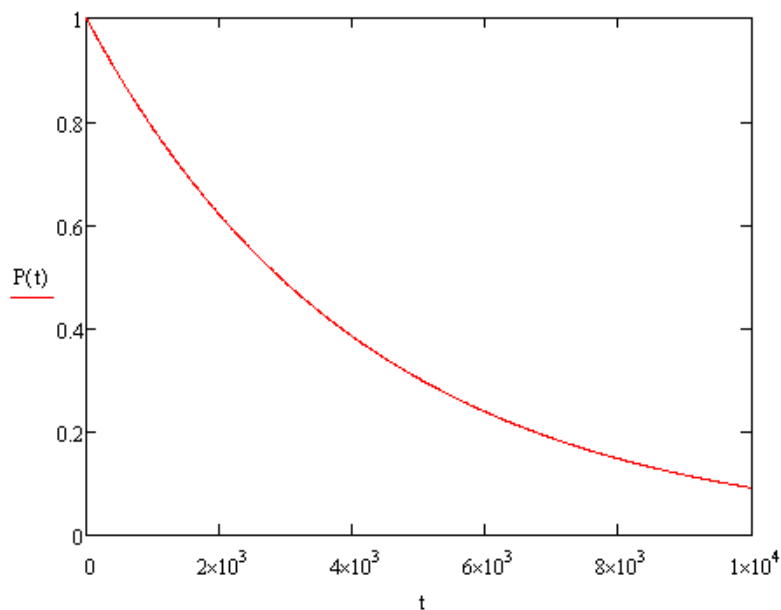


Рисунок 2.8 – Графік вірогідності безвідмовної роботи

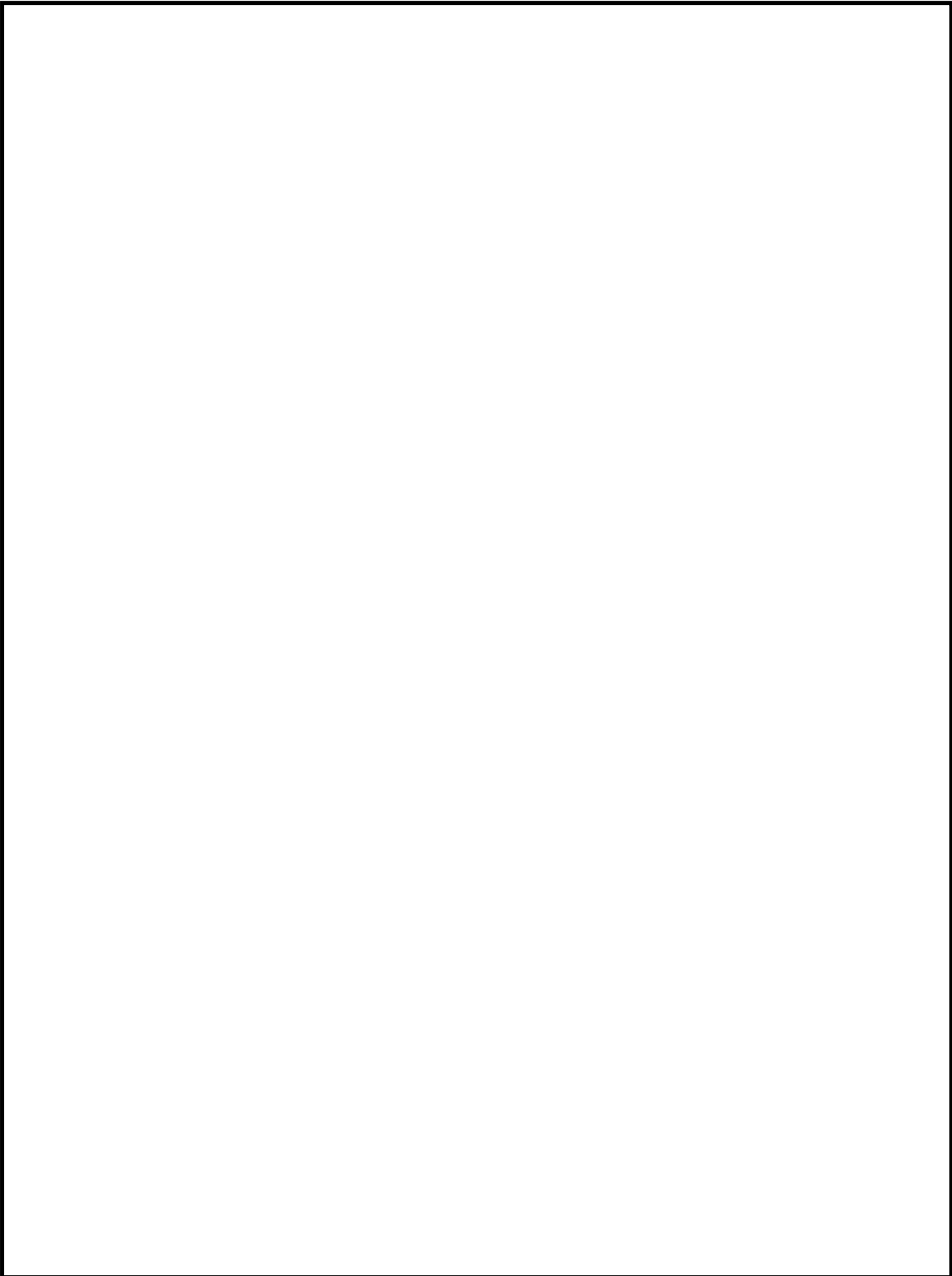
Час безвідмовної роботи визначається при вірогідності безвідмовної роботи, рівній 0,37. Виходячи з формули, час безвідмовної роботи визначається за формулою:

$$t = -\frac{1}{\lambda_0} \cdot \ln(P), \quad (2.13)$$

$$t = -\frac{1}{23,91 \cdot 10^{-5}} \cdot \ln(0,37) = 4158 \text{ (год.)}.$$

Із розрахунків видно, що час безвідмовної роботи складає 4158 годин, що дорівнює приблизно 6 місяцям безперервної роботи.

					171.6321М.КР.ПЗ.Р2	Арк.
						33
Змн.	Арк.	№ документу	Підпис	Дата		



					171.6321М.КР.ПЗ.РЗ			
Змн.	Арк.	№ документу	Підпис	Дата				
Розробник		Пелипас В.В.			Реалізація алгоритмів цифрової обробки аудіосигналів	Літ.	Арк.	Аркуші
Консультант							34	
Н. контр.		Добрінова Л. П.				НУК імені адмірала Макарова		
Керівник		Ушкаренко О.О.						
Зав. каф.		Рябенський В. М.						

РОЗДІЛ 3. РЕАЛІЗАЦІЯ АЛГОРИТМІВ ЦИФРОВОЇ ОБРОБКИ АУДІОСИГНАЛІВ

3.1. Частотна корекція сигналів

Під обробкою звукового сигналу розуміється зміна його частотної або фазової характеристики, звуження або розширення динамічного діапазону, застосування амплітудної, частотної або фазової модуляції, видалення шумів, а також створення затриманих копій цього сигналу. Метою обробки можуть бути як суто технічні завдання, такі як узгодження параметрів сигналу з характеристиками електроакустичного тракту, так і художні, що визначаються звуорежисером, зокрема це можуть бути різні звукові ефекти (тремоло, вібрато, хор, луна, реверберація та інші).

Обробка звукових сигналів проводиться переважно у цифровому вигляді за допомогою звукових процесорів. Програмне забезпечення дозволяє здійснювати як завгодно складні перетворення звукових сигналів і створювати найнеймовірніші звукові ефекти. Дуже важливо, що для різних маніпуляцій зі звуком не потрібна постійна зміна обладнання, достатньо змінити програмне забезпечення.

Щоб зібрати всі просторово-часові події та зробити з них музику, в якій кожна частина ідеально підходить до іншої, потрібно бути трохи художником, трохи вченим. Необхідно знати фізичні основи перетворень і вміти грамотно користуватися обладнанням. Науковий аспект роботи полягає в тому, щоб знати, як поєднати все в єдину систему і як керувати параметрами, що впливають на обробку звуку. Художній аспект включається, коли ви приймаєте рішення, які ефекти та звуки використовувати, яким має бути баланс та як розмістити різні партії у остаточному міксі.

Під частотною корекцією розуміється підвищення чи зниження рівня спектральних складових звукових сигналів у вибраних смугах з допомогою фільтрів без внесення нових складових спектра. Необхідність серйозної частотної корекції звуковідтворювальної апаратури найчастіше зумовлена

					171.6321М.КР.ПЗ.РЗ	Арк.
						35
Змн.	Арк.	№ документу	Підпис	Дата		

поганими акустичними характеристиками приміщень, де проводиться концерт чи прослуховується звукозапис. Якщо, наприклад, у залі є рівні тверді поверхні сцени та підлоги, бетонні або цегляні стіни, бляшаний дах, то все це може почати гриміти та деренчити, а в кращому разі вокаліст та слухачі перестануть розуміти слова через зниження розбірливості. Серйозні проблеми з якісним сприйняттям звуку виникають у салоні автомобіля. Апаратура частотної корекції звукових сигналів є сполучною ланкою між звучанням звуковідтворювальної системи та відгуком приміщення, і вона значною мірою може вирішити такі проблеми.

Дані, що стосуються мультимедіа (звук, відео) зберігаються у файлах у так званому форматі RIFF (Resource Interchange File Format – формат файлу для обміну ресурсами), представлений в табл. 3.1. WAV-файли, що містять звук, мають формат RIFF.

Таблиця 3.1 – Формат WAV-файлу

Зсув	Назва поля	Опис	Розмір
0	formatId	Містить символи "RIFF" в кодуванні ASCII. Є початком RIFF-ланцюжка.	4
4	blockLength	Це розмір ланцюжка, що залишився, починаючи з цієї позиції. Інакше кажучи, це розмір файлу - 8, тобто виключені поля chunkId і chunkSize.	4
8	soundId	Містить символи "WAVE".	4
12	subId	Містить символи "fmt".	4
16	headerLength	16 для формату PCM. Це розмір підчіпки, що залишився, починаючи з цієї позиції.	4
20	formatType	Аудіо формат, повний перелік можна отримати тут. Для PCM = 1 (тобто Лінійне квантування). Значення, відмінні від 1, позначають певний формат стиснення.	2
22	channels	Кількість каналів. Моно = 1, стерео = 2.	2

24	frequency	Частота дискретизації: 8000 Гц, 44 100 Гц і т.д.	4
28	dataRate	Кількість байт, надісланих за секунду відтворення.	4
32	sampleBytes	Кількість байт для однієї вибірки, у тому числі всі канали.	2
34	sampleWidth	Кількість біт у вибірці. Так звана "глибина" чи точність звучання. 8 біт, 16 біт і т.д.	2
36	dataId	Містить символи "data"	4
40	dataLength	Кількість байт у сфері даних.	4
44	data	Безпосередньо WAV-дані.	data Length

У моно варіанті значення амплітуди розташовані послідовно. У стерео спочатку розташовується значення амплітуди для лівого каналу, потім правого. Для монофонічного сигналу з дискретністю 8 біт звукові дані є масивом однобайтових значень, кожне з яких є вибіркою сигналу.

Для стереофонічного сигналу з дискретністю 8 біт звукових даних мають формат масиву двобайтових слів, причому молодший байт слова відповідає лівому каналу, а старший – правому.

Формат звукових даних із дискретністю 16 біт виглядає аналогічно. Для монофонічного сигналу дані зберігаються у масиві 16-бітових слів. Для стереофонічного використовують масив подвійних слів, причому молодшому слову відповідає лівий канал, а старшому – правий.

Діапазон зміни значень вибірок сигналу визначається дискретизацією. Для 8-бітових даних він становить від 0 до 255 (0xFF), причому відсутність сигналу (повної тиші) відповідає значення 128 (0x80). Для 16-бітових даних діапазон зміни становить від -32768 (-0x8000) до 32767 (0x7FFF), відсутності сигналу відповідає значення 0.

Фільтр нижніх частот ковзного середнього – різновид цифрового фільтра з кінцевою імпульсною характеристикою. На рис. 3.1 представлені АЧХ фільтра ковзного середнього різних порядків.

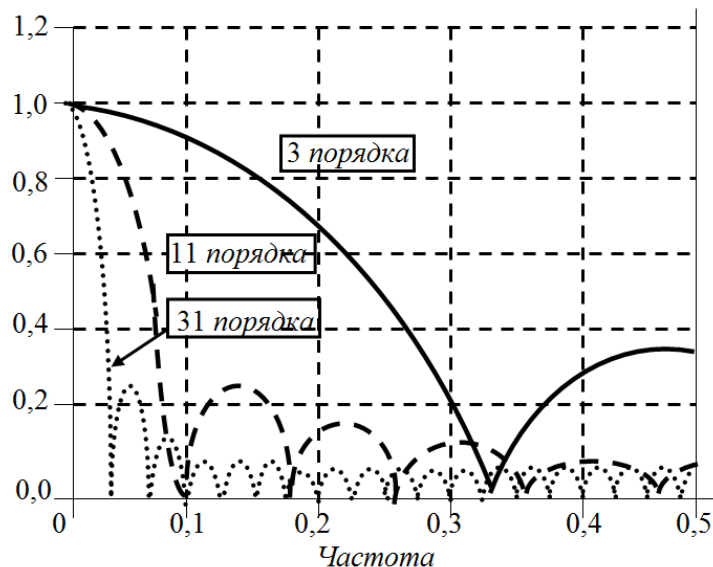


Рисунок 3.1 – Частотні характеристики фільтрів різних порядків

Створюємо проект Win32 Console Application із підтримкою MFC. Текст програми, в якій реалізований 8-точковий фільтр ковзного середнього, представлений нижче.

```
#include "stdafx.h"          // Підключення стандартних заголовків
#include "WaveSample.h"      // Підключення заголовку для роботи з WAV
#pragma comment(lib, "winmm.lib") // Підключення бібліотеки winmm для роботи з аудіо

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

CWinApp theApp; // Ініціалізація застосунку
using namespace std; // Використання простору імен std
```

// Структура для збереження параметрів WAV-файлу

struct WAVESTRUCT

```
{
    char formatId[4]; // Ідентифікатор формату (наприклад, "RIFF")
    int blockLength; // Довжина блоку даних
    char soundId[4]; // Ідентифікатор звуку ("WAVE")
    char subId[4]; // Підсекція формату ("fmt ")
    int headerLength; // Довжина заголовку
    short formatType; // Тип формату (1 = PCM)
    short channels; // Кількість каналів (1 = моно)
    int frequency; // Частота дискретизації (Гц)
    int dataRate; // Швидкість передачі даних
    short sampleBytes; // Кількість байтів на семпл
    short sampleWidth; // Глибина семплу (біт)
    char dataId[4]; // Ідентифікатор даних ("data")
    int dataLength; // Довжина даних
};
```

int main(int argc, TCHAR* argv[], TCHAR* envp[])

```
{
    int nRetCode = 0; // Код завершення програми

    // Ініціалізація MFC (Microsoft Foundation Classes)
    if (!AfxWinInit(::GetModuleHandle(NULL), NULL,
::GetCommandLine(), 0))
    {
        cerr << _T("Фатальна помилка: Ініціалізація MFC не вдалася") <<
endl;

        nRetCode = 1;
    }
}
```

					171.6321M.KP.ПЗ.РЗ	Арк.
						39
Змн.	Арк.	№ документа	Підпис	Дата		

```

    }
else
{
    // Заповнення масиву вибірок випадковими даними
    char data[88400];
    for (int i = 0; i < 88400; i++)
        data[i] = 128 + rand() % 127;

    // Реалізація KIX-фільтра (ковзне середнє)
    int tmp = 0; // Тимчасова змінна для підсумку
    int N = 8;   // Розмір вікна для фільтрації
    int Y[88400]; // Масив для фільтрованих даних
    memset(Y, 0, 88400 * sizeof(int)); // Ініціалізація нулями

    // Застосування фільтра ковзного середнього
    for (int i = N; i < 88400; i++)
    {
        for (int k = 0; k < N; k++)
        {
            tmp = tmp + data[i - k]; // Підсумовування N вибірок
        }
        Y[i] = tmp; // Збереження підсумку
        tmp = 0;    // Скидання підсумку
    }

    // Ділення кожного елементу на N для обчислення середнього
    значення
    for (int i = 0; i < 88400; i++)
        data[i] = Y[i] / N;

```

					171.6321М.КР.ПЗ.РЗ	Арк.
						40
Змн.	Арк.	№ документа	Підпис	Дата		

```

// Заповнення структури WAVESTRUCT параметрами файлу
WAVESTRUCT sws;
strcpy(sws.formatId, "RIFF");
sws.blockLength = sizeof(data) + 36;
strcpy(sws.soundId, "WAVE");
strcpy(sws.subId, "fmt ");
sws.headerLength = 16;
sws.formatType = 1; // PCM формат
sws.channels = 1; // Моно
sws.frequency = 22100; // Частота дискретизації 22.1 кГц
sws.dataRate = 22100; // Швидкість передачі даних
sws.sampleBytes = 1; // 1 байт на семпл
sws.sampleWidth = 8; // 8 біт на семпл
strcpy(sws.dataId, "data");
sws.dataLength = 88400;

// Створення та запис файлу WAV
CFile nf;
nf.Open("e:\\test.wav", CFile::modeCreate

```

Для демонстрації роботи фільтра у програмі реалізований генератор, який формує білий шум із рівномірним спектром у всьому діапазоні частот. Після обробки такого сигналу фільтром вихідний сигнал записується у WAV-файл. Спектральний аналіз виконується у програмі SpectraLab. На рис. 3.2 показано вікно із настройками програми.

					171.6321М.КР.ПЗ.РЗ	Арк.
						41
Змн.	Арк.	№ документу	Підпис	Дата		

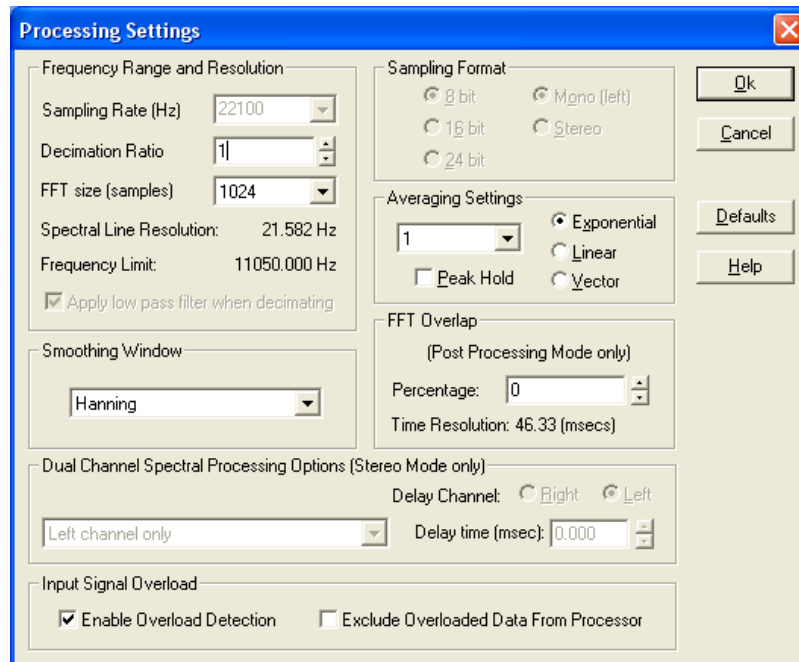


Рисунок 3.2 – Вікно із настройками програми

На рис. 3.3 представлено вікно програми SpectraLab з результатами аналізу WAV-файлу, який записаний білий шум. З малюнка видно, що спектральні складові розподілені рівномірно аналізованому діапазоні частот.

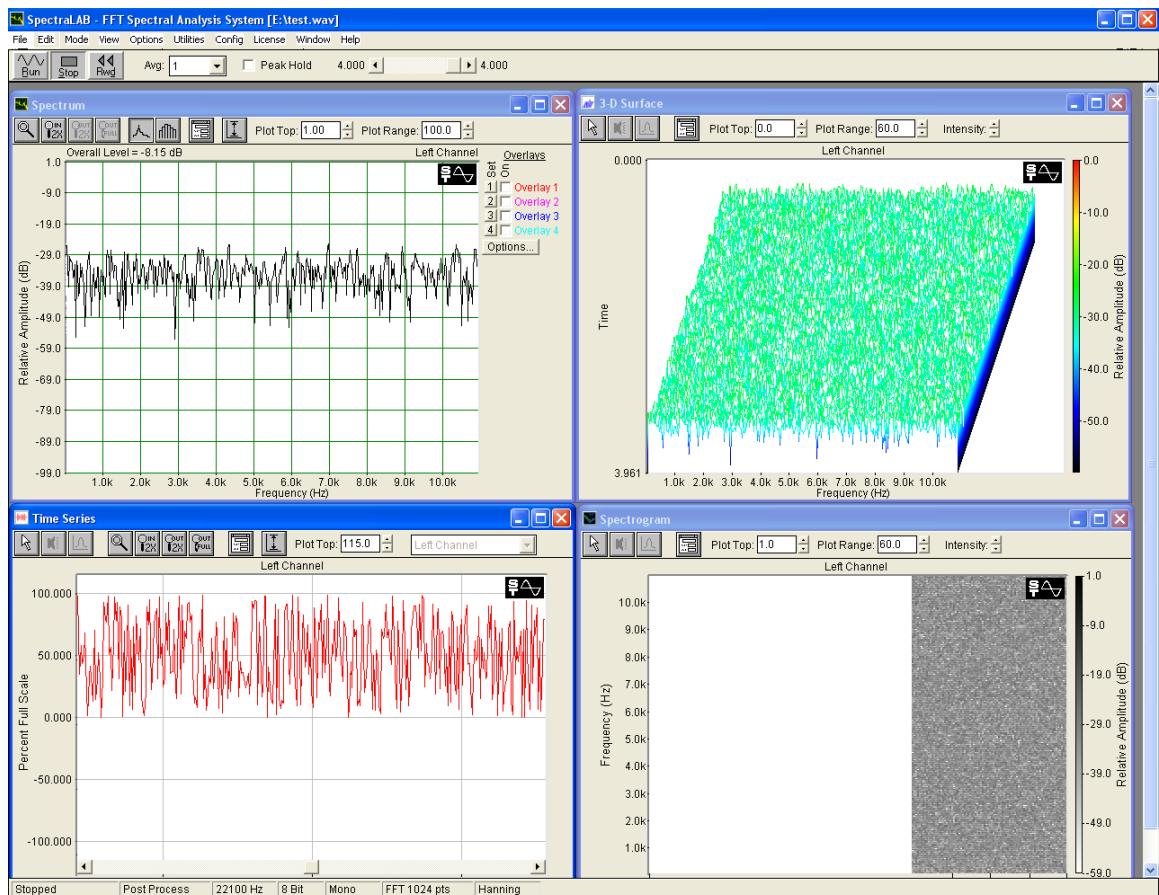


Рисунок 3.3 – Результати аналізу сигналу

Програмна реалізація фільтра ковзного середнього має вигляд:

```
int сумарнеЗначення = 0; // Тимчасова змінна для підсумку
const int розмірВікна = 8; // Розмір вікна для фільтрації
int фільтрованіДані[88400]; // Масив для збереження відфільтрованих
даних
```

```
// Ініціалізація масиву фільтрованих даних нулями
memset(фільтрованіДані, 0, sizeof(фільтрованіДані));
```

```
// Застосування фільтра ковзного середнього
for (int i = розмірВікна; i < 88400; i++)
{
    for (int k = 0; k < розмірВікна; k++)
    {
        сумарнеЗначення += data[i - k]; // Підсумовування N вибірок
    }
    фільтрованіДані[i] = сумарнеЗначення; // Збереження підсумку
    сумарнеЗначення = 0; // Скидання підсумку
}
```

```
// Обчислення середнього значення для кожної вибірки
for (int i = 0; i < 88400; i++)
{
    data[i] = фільтрованіДані[i] / розмірВікна; // Ділення на N
}
```

На рис. 3.4 представлено вікно програми SpectraLab з результатами спектрального аналізу сигналу, що є білим шумом, пропущеним через 8-точковий фільтр ковзного середнього.

					171.6321М.КР.ПЗ.РЗ	Арк.
						43
Змн.	Арк.	№ документи	Підпис	Дата		

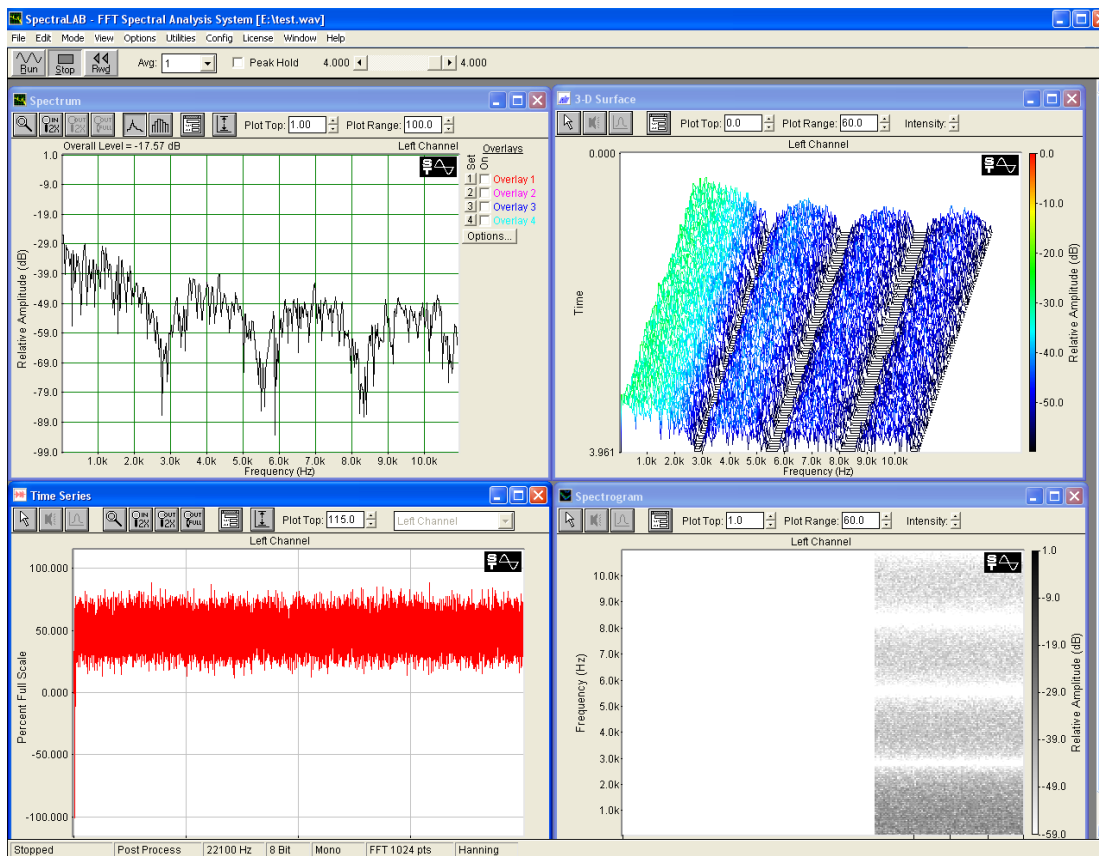


Рисунок 3.4 – Обробка аудіо сигналу фільтром ковзаючого середнього

Реалізація фільтрів верхніх частот. За допомогою програми `siirf` можна розрахувати коефіцієнти фільтра з нескінченною імпульсною характеристикою (рекурсивного), коригувати ці коефіцієнти одночасно спостерігаючи за зміною амплітудно-частотної характеристики. Фільтр реалізований у вигляді послідовного з'єднання секцій другого порядку для ФНЧ та ФВЧ, і четвертого порядку для смугового та режекторного фільтрів. Програмна реалізація такого фільтра представлена нижче.

// Filter coefficients

double coeffA0 = 0.018307;

double coeffA1 = -0.036614;

double coeffA2 = 0.018307;

double coeffB1 = -1.582450;

double coeffB2 = -0.655667;

```

// Array to store processed data
unsigned int tempData[88400];

// Initialize the array with zeros
memset(tempData, 0, sizeof(tempData));

// Apply the filter to the data array
for (int n = 2; n < 88400; n++)
{
    tempData[n] =
        coeffB1 * tempData[n - 1] +
        coeffB2 * tempData[n - 2] +
        coeffA0 * data[n] +
        coeffA1 * data[n - 1] +
        coeffA2 * data[n - 2];
}

// Copy processed data back to the data array
for (int n = 0; n < 88400; n++)
{
    data[n] = tempData[n];
}

```

Результат спектрального аналізу сигналу, отриманого з виходу фільтра, представлений на рис. 3.5.

					171.6321М.КР.ПЗ.РЗ	Арк.
						45
Змн.	Арк.	№ документа	Підпис	Дата		

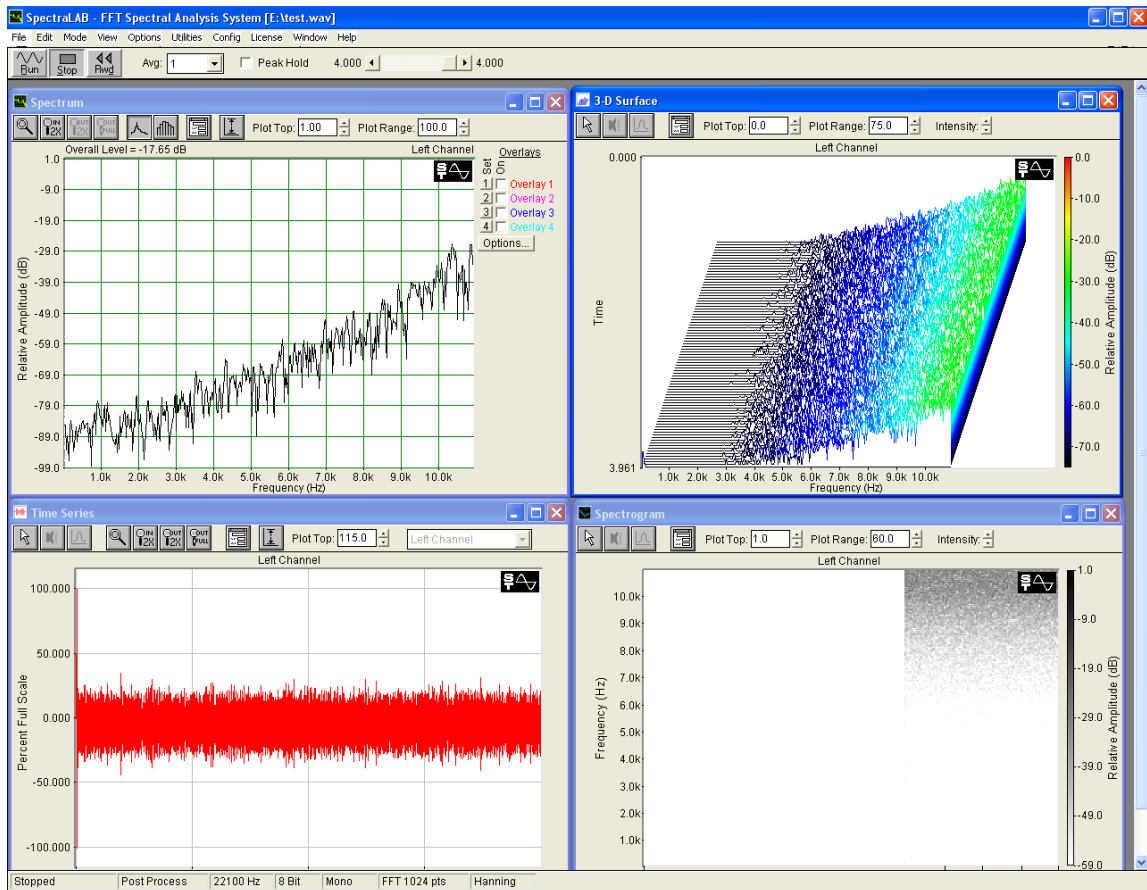


Рисунок 3.5 – Обробка аудіо сигналу за допомогою ФВЧ

Програмна реалізація смугового фільтра 4 системи представлена далі.

```
unsigned int outputData[88400]; // Array to store the filtered output
memset(outputData, 0, sizeof(outputData)); // Initialize the array with zeros
```

```
// First filter coefficients
```

```
double coeffA0 = 0.052539;
```

```
double coeffA1 = 0;
```

```
double coeffA2 = -0.105078;
```

```
double coeffA3 = 0;
```

```
double coeffA4 = 0.052539;
```

```
double coeffB1 = 0.484212;
```

```
double coeffB2 = -1.184490;
```

```
double coeffB3 = 0.278339;
```

```

double coeffB4 = -0.335580;

// Apply the first filter
for (int n = 4; n < 88400; n++)
{
    outputData[n] =
        coeffB1 * outputData[n - 1] +
        coeffB2 * outputData[n - 2] +
        coeffB3 * outputData[n - 3] +
        coeffB4 * outputData[n - 4] +
        coeffA0 * data[n] +
        coeffA1 * data[n - 1] +
        coeffA2 * data[n - 2] +
        coeffA3 * data[n - 3] +
        coeffA4 * data[n - 4];
}

// Scale down the output
for (int n = 0; n < 88400; n++)
{
    data[n] = outputData[n] / 2;
}

// Second filter coefficients
coeffA0 = 0.065234;
coeffA1 = 0;
coeffA2 = -0.130468;
coeffA3 = 0;
coeffA4 = 0.065234;
coeffB1 = 0.526342;

```

					171.6321M.KP.ПЗ.РЗ	Арк.
						47
Змн.	Арк.	№ документи	Підпис	Дата		

```

coeffB2 = -1.470700;
coeffB3 = 0.420462;
coeffB4 = -0.658290;

// Reset the output array for the second filter
memset(outputData, 0, sizeof(outputData));

// Apply the second filter
for (int n = 4; n < 88400; n++)
{
    outputData[n] =
        coeffB1 * outputData[n - 1] +
        coeffB2 * outputData[n - 2] +
        coeffB3 * outputData[n - 3] +
        coeffB4 * outputData[n - 4] +
        coeffA0 * data[n] +
        coeffA1 * data[n - 1] +
        coeffA2 * data[n - 2] +
        coeffA3 * data[n - 3] +
        coeffA4 * data[n - 4];
}

// Scale down the output for the second filter
for (int n = 0; n < 88400; n++)
{
    data[n] = outputData[n] / 2;
}

```

					171.6321M.KP.ПЗ.РЗ	Арк.
						48
Змн.	Арк.	№ документи	Підпис	Дата		

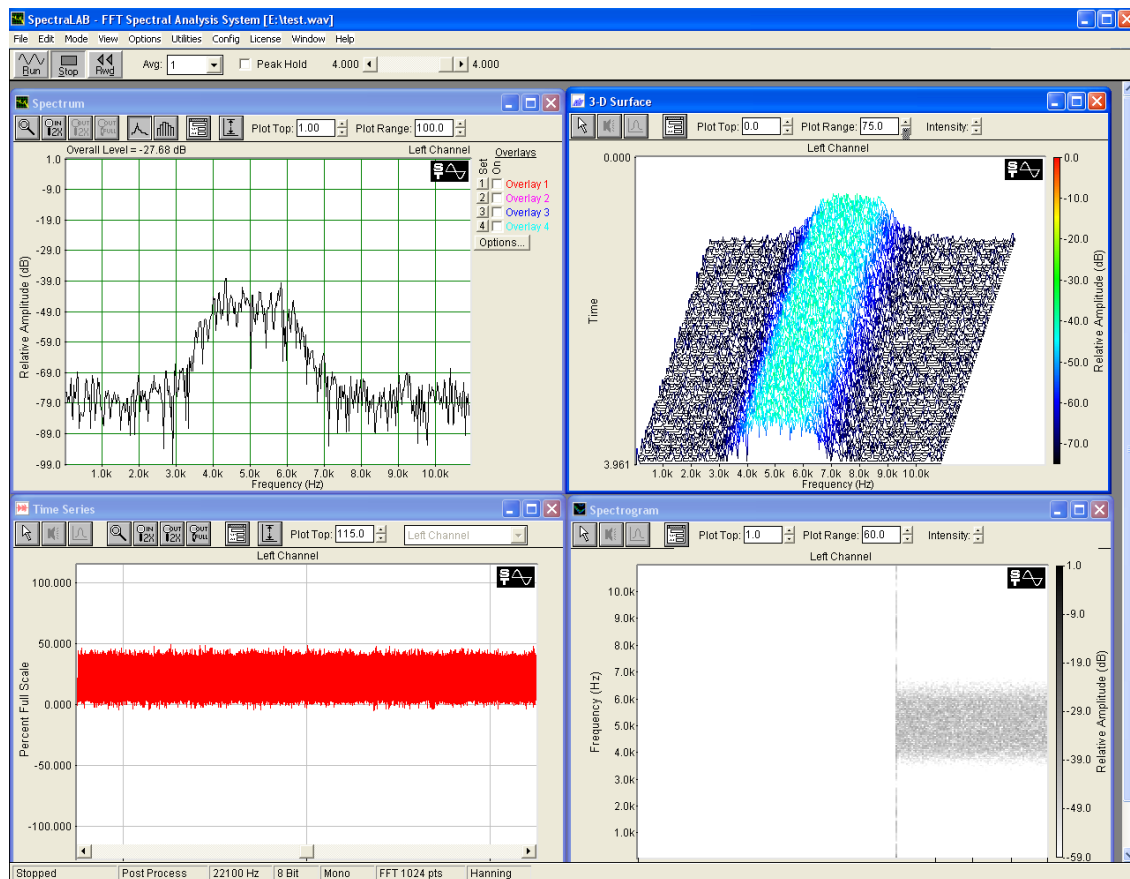


Рисунок 3.7 – Результат застосування смугового фільтру

3.2. Динамічна і спектральна обробка звукових сигналів

Динамічна обробка варта скорочення динамічного діапазону звукових сигналів. Звукові сигнали в радіомовленні, телебаченні та звукозаписі завжди піддаються такій обробці, незалежно аналогові вони чи цифрові. Це пов'язано з тим, що часто динамічний діапазон природних звуків, звуків музики та мови значно ширший за динамічний діапазон електроакустичних трактів сучасної апаратури. Динамічна обробка звукових сигналів проводиться за допомогою лімітерів, максимайзерів, компресорів, експандерів та гейтів. Це все порогові пристрої, в яких при досягненні сигналу встановленого рівня їхній коефіцієнт передачі змінюється стрибком. У такий спосіб можна як стиснути, і розширити динамічний діапазон сигналу. У системах передачі звукових сигналів лініями зв'язку скорочення і зворотне розширення динамічного діапазону проводиться за допомогою компандерної системи компресування. На вході лінії встановлюється компресор, а на виході

експандер, тому в такій системі сигнал компресований тільки в лінії зв'язку. Назви приладів такі ж, як і при динамічній обробці, але принцип роботи зовсім інший, на це потрібно звернути увагу.

Спектральні процесори звукових сигналів призначені зміни тембру звуку музичних інструментів і вокальних виконавців шляхом навмисного зміни його спектрального складу. При спектральному перетворенні спотворення можуть створюватися як гармонік і субгармонік, які були відсутні у вихідному звуковому матеріалі. Спотворення можуть вноситися в звукових смугах і окремих музичних інструментів чи солістів. Вони можуть створюватися у всьому звуковому діапазоні. При цьому може змінюватися частотний та амплітудний склад обертонів музичних звуків. На відміну від частотної корекції тембру за допомогою еквайзерів при спектральній обробці звукових сигналів створюються обертони (гармоніки), яких не було. Отже, спектральне перетворення – це принципово нелінійний процес. Такі перетворення здійснюються до створення художніх ефектів звучання, і навіть під час реставрації старих записів, у яких безповоротно втрачені вищі гармоніки. Нерідко доводиться використовувати спектральну корекцію відновлення динаміки музики у сфері високих частот, порушеної при компресуванні. До спектральних процесорів звуку відносяться ексайтери, енхансери, віталайзери, спектралайзери та багато інших. Їх назви переважно є брендом фірми-виробника.

Лістинг програми, що демонструє реалізацію фільтра, який виконує децимацію сигналу, наведено в додатку Б.

На рис. 3.8 представлений результат спектрального аналізу вихідного сигналу, але в рис. 3.9 – сигнал після децимації.

					171.6321M.KP.ПЗ.РЗ	Арк.
						50
Змн.	Арк.	№ документу	Підпис	Дата		

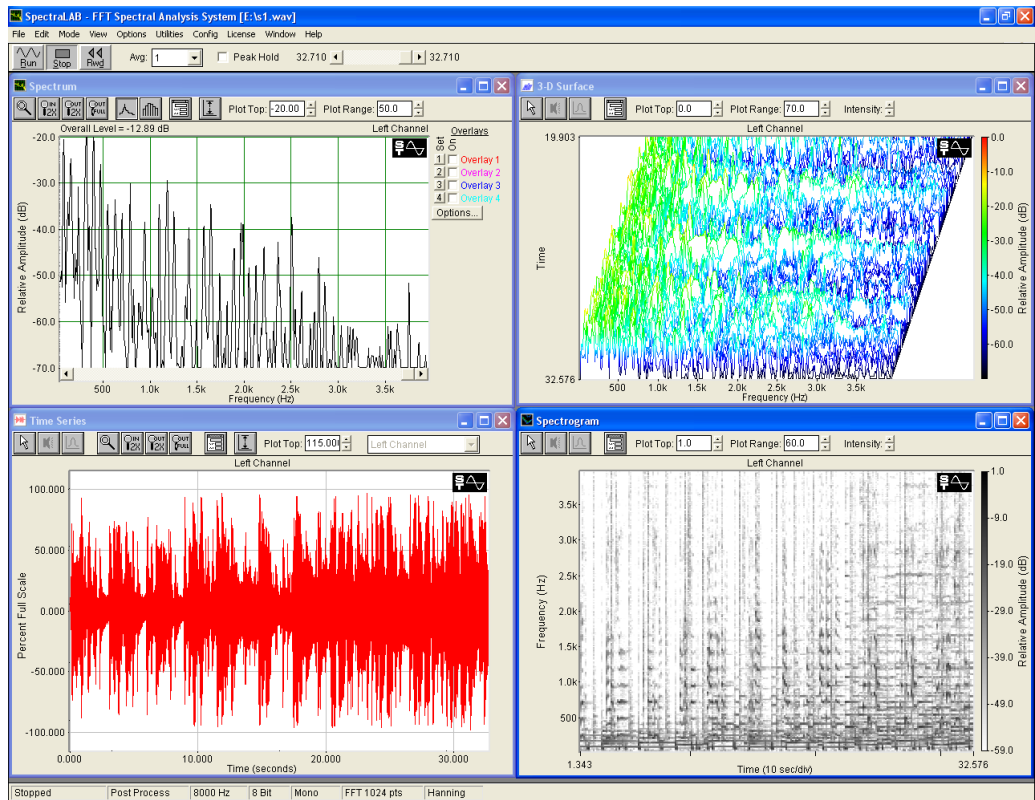


Рисунок 3.8 – Результат спектрального аналізу вихідного сигналу

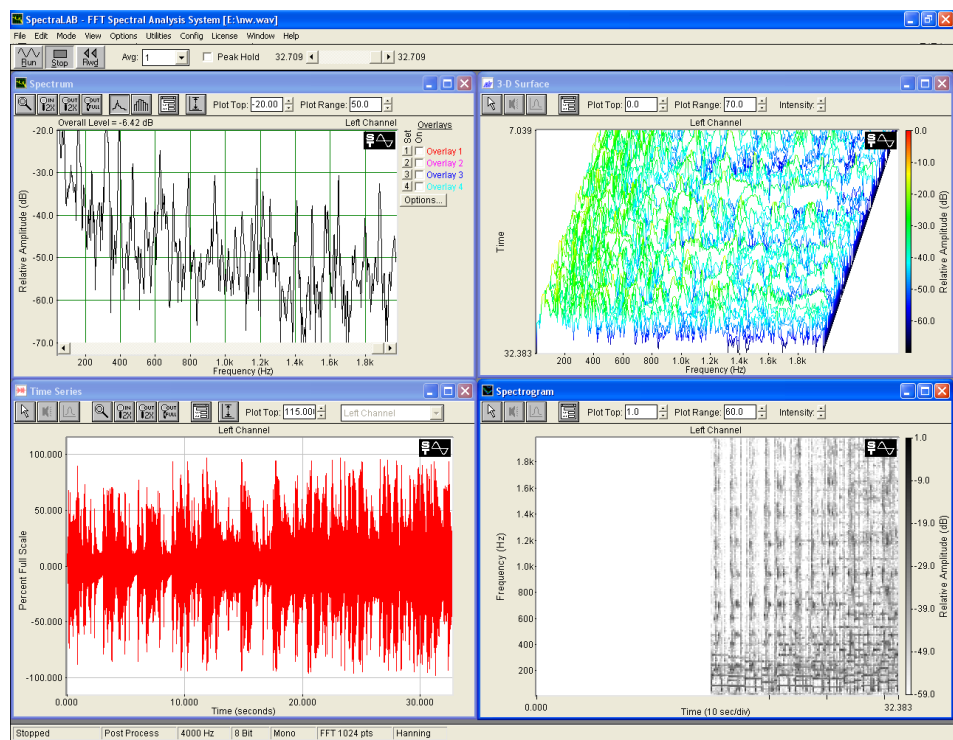


Рисунок 3.9 – Результат аналізу сигналу після децимації

При прослуховуванні звукового файлу після децимації можна почути неприємні клацання та тріск. Зменшити цей ефект можна за допомогою фільтрації.

3.3. Програмна реалізація різноманітних звукових ефектів

Під час підготовки аудіоматеріалу до відтворення до нього можуть бути внесені деякі звукові ефекти, що додають до нього додаткові фарби. Ефекти можуть використовуватися не тільки при обробці музичних творів, але і при відтворенні мовного матеріалу для зниження монотонності під час прослуховування тривалих фрагментів. Однак у цьому випадку слід бути особливо обережним, оскільки подібні обробки зазвичай призводять до зниження розбірливості мовного сигналу. Тому необхідно постійно стежити за тим, щоб позитивний ефект, що досягається, не був би зведений нанівець зниженням розбірливості мовного матеріалу, що призводить, у свою чергу, до підвищення навантаження слухача.

Звуковий ефект «луна» та «реверберація». Один із найпоширеніших класів є звукові ефекти, засновані на затримці сигналу. У реальній ситуації затримка звукового сигналу може виникнути при відображенні звукової хвилі від твердої перешкоди, наприклад, від стіни.

Величину затримки, що виникає при цьому, легко визначити, знаючи пройдений звуковою хвилею шлях і враховуючи, що швидкість звуку можна вважати рівною 330 м/с. Таким чином, за одну мілісекунду звукова хвиля проходить 33 сантиметри. Для розрахунків зручніше пам'ятати, що один метр проходить за 3 мс. Знаючи частоту дискретизації аналогового сигналу, легко визначити кількість вибірок, які будуть отримані за вказаний проміжок часу.

Відлуння виникає в тунелях, на стадіонах, в гірських ущелинах печерах та інших місцях через відображення звукових хвиль від будь-яких перешкод, що заважають їх поширенню, але тільки у разі запізнення відбиття на час більше 50 мс. Цей час дорівнює постійному часу слуху людини. Поки запізнення відбитих сигналів менше цього часу вони сприймаються як єдиний звук, луна немає.

Звуковий ефект «луна» виходить, якщо один або кілька затриманих сигналів додаються до оригінального. Щоб ефект сприймався як відлуння, затримка повинна бути близько 50 мс або більше. Ефект може бути

					171.6321М.КР.ПЗ.РЗ	Арк.
						52
Змн.	Арк.	№ документа	Підпис	Дата		

досягнутий як аналогової, так і цифрової обробки. Якщо значна кількість затриманих сигналів відтворюється протягом кількох секунд, результуючий сигнал створює ефект присутності у великому приміщенні та сприймається як ефект реверберації.

Розглянемо реалізацію одноразового відображення звукової хвилі. Цей ефект є базовим для більш складних ефектів, таких як імітація реверберації і хору.

Блок-схему реалізації одноразової затримки наведено на рис. 3.10.

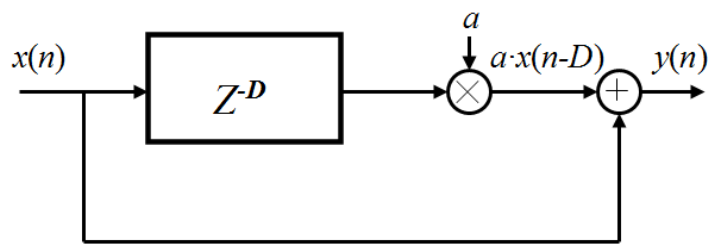


Рисунок 3.10 – Блок-схема реалізації одноразової затримки

Вхідний сигнал $x(n)$ одночасно надходить на лінію затримки та на суматор, що формує вихідний сигнал. Затриманий сигнал $x(n)$ множиться на масштабний коефіцієнт a також надходить на суматор. Перетворення, що здійснюються даною схемою, можуть бути виражені такою формулою:

$$y(n) = x(n) + ax(n - D).$$

Дана схема є лінійним фільтром з передавальною функцією:

$$H(z) = 1 + az^{-D}.$$

Така передатна характеристика є характеристикою гребінчастого фільтра. Відмінною особливістю даного класу фільтрів є виконання для них рівності:

$$H(j\omega) = H(j\omega + \frac{2\pi}{D}).$$

Оскільки за будь-якого відображенні звуковий сигнал послаблюється, то $-1 < a < 1$. Негативні значення відповідають відображенню протифазі. Ця модель досить груба, оскільки як поширення звукової хвилі, так і її відображення є частотно-залежними процесами. Однак у більшості випадків

ця модель є адекватною. Величина D визначається різницею часу між приходом прямої та відбитої звукової хвилі. Програмна реалізація фільтра наведена нижче.

```
int delaySamples = 200; // Delay in samples
for (int i = 0; i < size; i++)
{
    if (i < delaySamples)
    {
        // For the first 'D' samples, copy data directly
        decimatedData[i] = audioData[i];
    }
    else
    {
        // Apply a delayed average for samples after the delay
        decimatedData[i] = 0.5 * audioData[i] + 0.5 * audioData[i -
delaySamples] / 2;
    }
}
```

Частота дискретизації в файлі, що обробляється, дорівнює 8 кГц. Розмір D=200 відповідає затримці 25 мс. При затримці понад 50 мс (D>400) слухач почує відлуння. Розглянутий приклад моделює одноразове відображення звукової хвилі, розташованої на відстані 4,3 метра від слухача (звукова хвиля долає відстань до стіни за 12,5 мс, і повертається назад за такий самий час). Для того щоб при виконанні цієї операції не виникало переповнення розрядної сітки, обидва сигнали складаються з коефіцієнтом 0,5. При зменшенні затримки нижче 10 мс сигнали можуть виникнути небажані биття, оскільки сума двох гармонійних сигналів з однаковою

					171.6321M.KP.ПЗ.РЗ	Арк.
						54
Змн.	Арк.	№ документу	Підпис	Дата		

амплітудою може бути представлена як амплітудна модуляція сигналу з частотою, що дорівнює різниці частот вихідних сигналів.

Введення затримки в один із каналів дозволяє створити ефект псевдостереофонічного прослуховування. І тут один із стереоканалів подається вихідний монофонічний сигнал, а інший – із затримкою його копія. Тимчасова затримка між каналами має перевищувати 15–20 мс. В іншому випадку стереосигнал суб'єктивно розбиватиметься на два затримані відносно один одного монофонічні сигнали. Іноді для створення псевдостереофонічного ефекту обидва канали подаються суми вихідного і затриманого сигналів, що відрізняються величиною затримки і ваговими коефіцієнтами, що використовуються при створенні вихідного сигналу у відповідному каналі.

Одноразова затримка часто використовується для пожвавлення "сухої" музики. В цьому випадку як основний, так і затриманий аудіосигнали складаються з однаковою амплітудою.

Область використання одноразових затримок сигналу дуже обмежена, оскільки в реальному житті рідко зустрічається ситуація, коли в акустичному полі присутня тільки одна поверхня, що відбиває. Крім того, навіть у цьому випадку для підвищення адекватності моделі необхідно використовувати формуючий фільтр, що враховує частотну характеристику поверхні, що відбиває, і, можливо, частотну характеристику середовища поширення сигналу. Для моделювання акустичних полів з декількома поверхнями, що відбивають, і обліку частотних характеристик процесів відображення і поширення звукової хвилі використовуються багаторазові затримки.

Багаторазова затримка може бути досягнута двома шляхами: доступом до різних елементів, що зберігаються в одній лінії затримки, або створення декількох окремих ліній, що мають різний час затримки.

Багаторазова затримка сигналу може бути реалізована як із використанням трансверсальних (KIX) фільтрів, так і з використанням рекурсивних фільтрів.

					171.6321М.КР.ПЗ.РЗ	Арк.
						55
Змн.	Арк.	№ документа	Підпис	Дата		

Блок-схема реалізації багаторазової затримки сигналу з використанням КІХ-фільтра наведена на рис. 3.11.

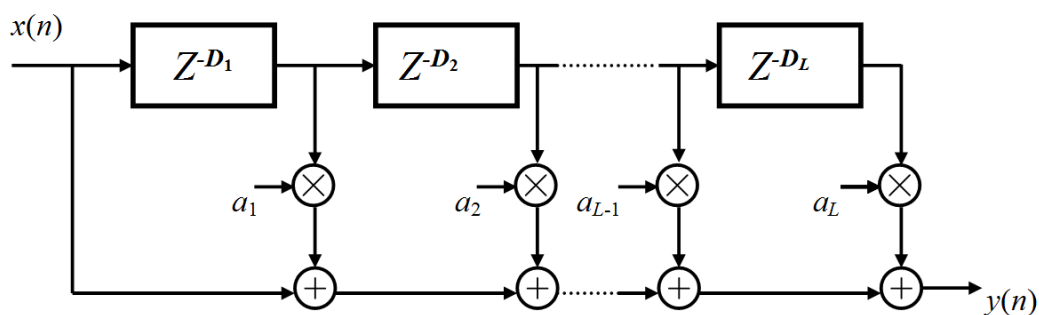


Рисунок 3.11 – Блок-схема реалізації багаторазової затримки сигналу

Різницяння рівняння такого фільтра може бути записане у вигляді:

$$y(n) = x(n) + \sum_{i=1}^L a_i x(n - \sum_{k=1}^i D_k)$$

У цьому фільтрі використовуються різні величини затримок вхідного сигналу, що дозволяє вважати цей фільтр гребінчастим. Використання кінцевого числа ліній затримки дає можливість врахувати вплив обмеженої кількості поверхонь, що відбивають.

3.4. Використання звукової системи персонального комп'ютера

Звернення до звукових систем в ОС Windows відбувається за допомогою спеціальних об'єктів, що є інтерфейсами звукового обладнання (рис. 3.12). Використання таких об'єктів дозволяє створювати вторинні звукові буфери, дані з яких надходять та сумуються у первинному звуковому буфері. Потім вибирається формат аудіо-потoku, в якому вказуються частота дискретизації сигналу, кількість вибірок, кількість вторинних буферів, кількість звукових каналів, розрядність потоку та інше.

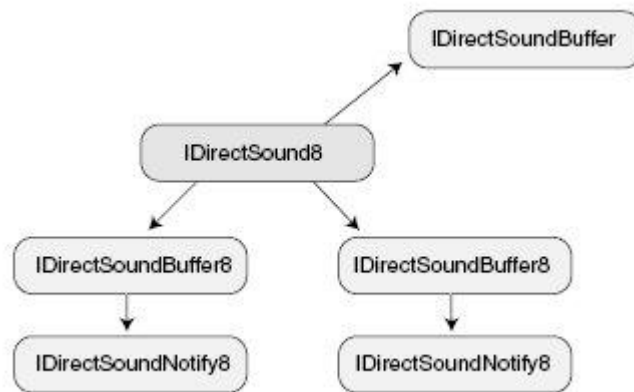


Рисунок 3.12 – Інтерфейси звукового обладнання

Для роботи з об'єктами існують стандартні бібліотеки DirectSound, в яких реалізовано основні методи та функції звукових інтерфейсів. Для роботи з ними необхідно використовувати Microsoft DirectX SDK, шлях, до якого вказується в проекті, що розробляється. Також необхідно підключити ряд бібліотек та заголовних файлів, залежно від набору необхідних методів обробки аудіо-потoku.

У роботі використовується функція для роботи з мультимедіа, які є в стандартній бібліотеці mmsystem.h.

Первинний звуковий буфер є кільцевою буфер (рис. 3.13).

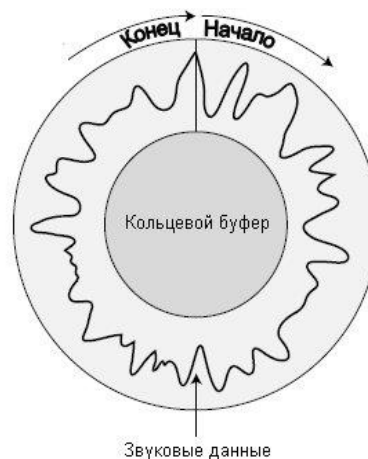


Рисунок 3.13 – Кільцевий буфер

Для формування безперервного звукового потоку використовують вторинні буфери. На рис. 3.14 представлено схему, що пояснює принцип використання вторинних звукових буферів.

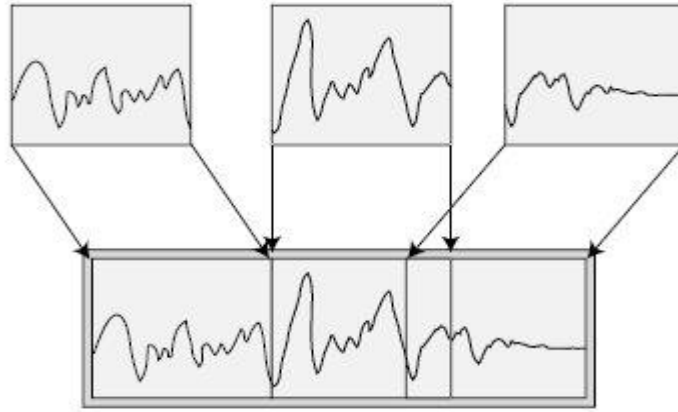


Рисунок 3.14 – Формування вторинних буферів

Для дослідження алгоритмів обробки звукових сигналів необхідно створити проект на основі діалогової програми та додати на діалогове вікно елемент «Button» (кнопка). У властивостях кнопки змінити текст у полі Caption (назва) на Record (запис). Лістинг програми наведено в додатку А.

На рис. 3.15 представлений 3D спектр вихідного записаного звукового фрагмента. Лініями відзначені найбільш яскраво виражені спектральні складові записаного звукового фрагмента, якими зручно аналізувати зміна спектра сигналу після частотної корекції.

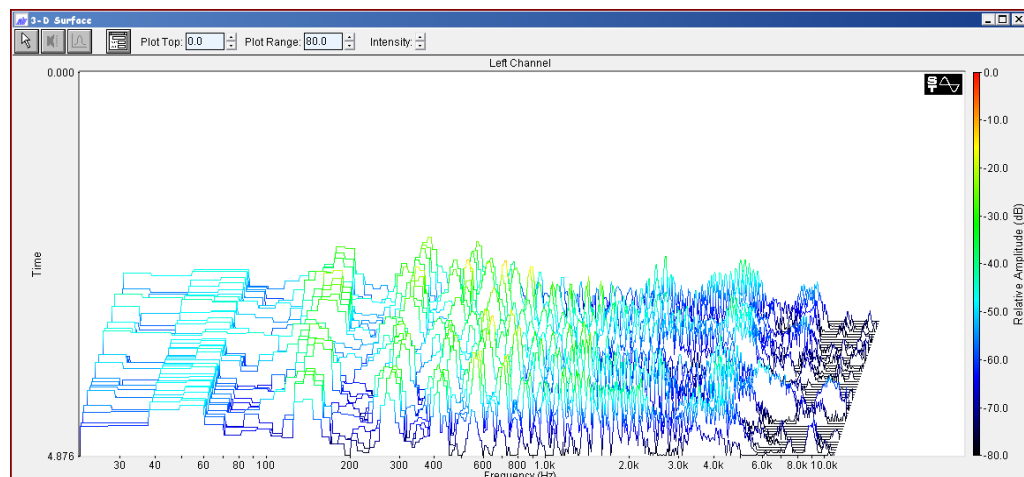


Рисунок 3.15 – 3D спектр вихідного записаного звукового фрагмента

Після частотної корекції спектр сигналу змістився в область високих частот (рис. 3.16) та область низьких частот (рис. 3.17).

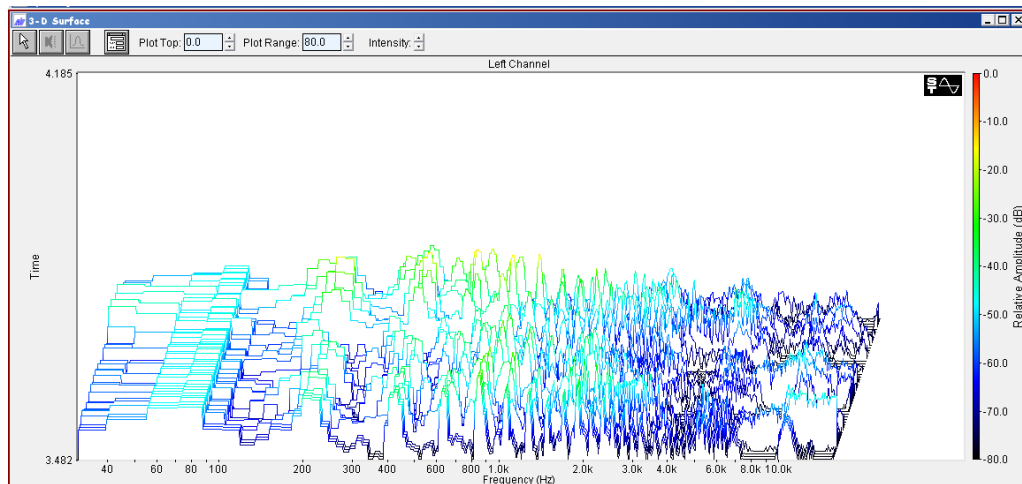


Рисунок 3.16 – Спектр сигналу змістився в область високих частот

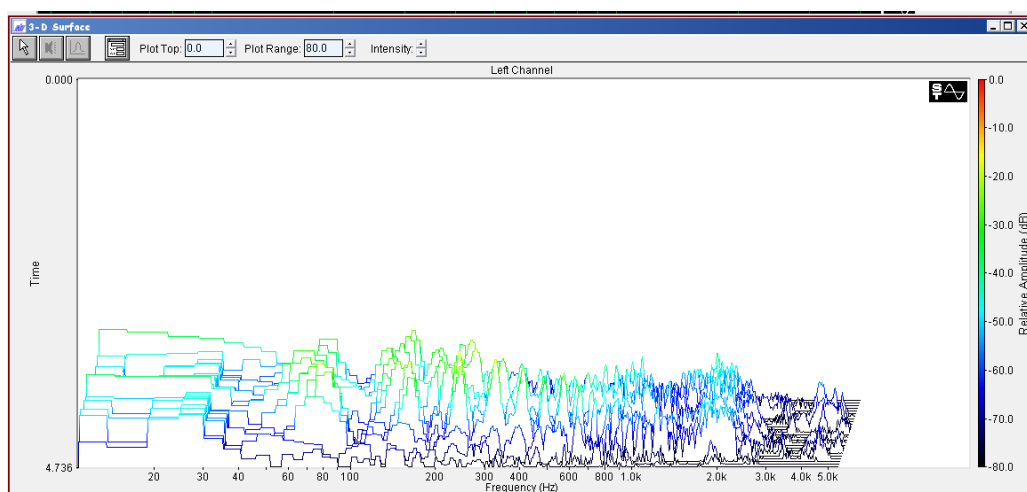


Рисунок 3.17 – Спектр сигналу змістився в область низьких частот

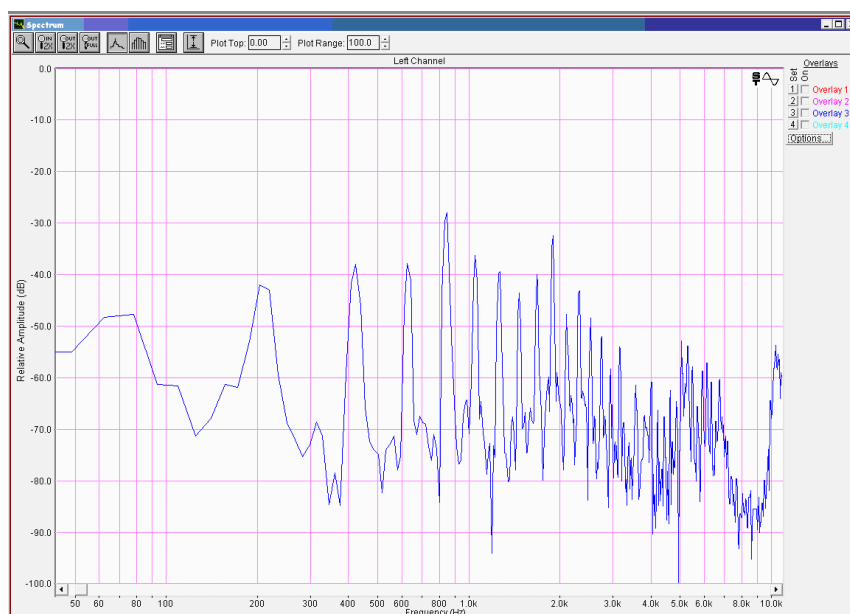
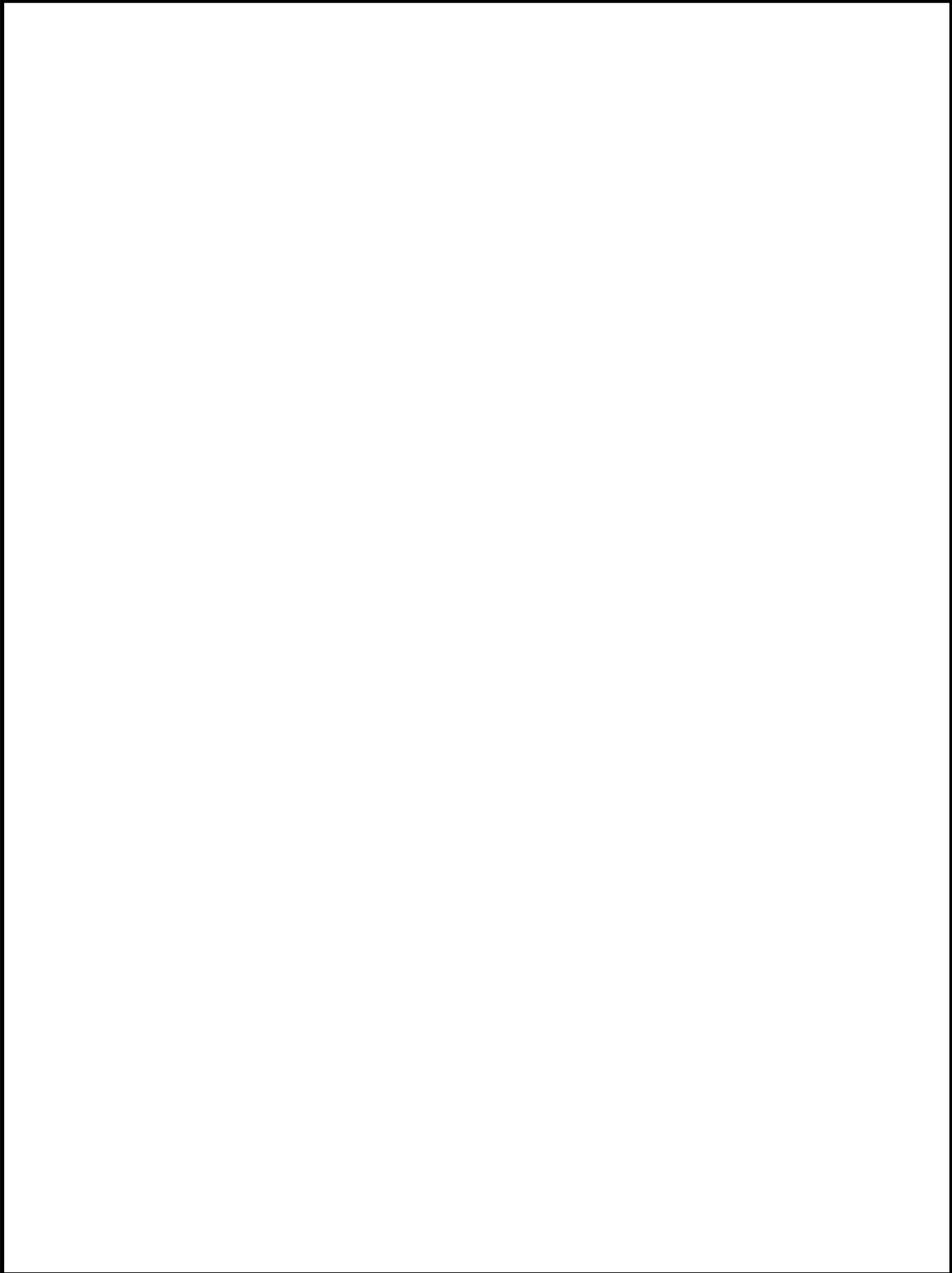


Рисунок 3.18 – Вигляд спектру аудіо сигналу після динамічної корекції

Використання методів та структур бібліотеки мультимедіа (mmsystem.h) дозволяє використовувати звукову карту в якості найпростішого осцилографа зі смугою пропускання до 22050 Гц, наявністю функцій захоплення та цифрової обробки сигналу, функцій формування звукового потоку та його конфігурації. Також можна використовувати звукову карту як аналізатор спектру аудіо-потoku, визначення амплітудно-частотних характеристик, системи визначення звукових образів та інших технічних завдань.

					171.6321М.КР.ПЗ.РЗ	Арк.
						60
Змн.	Арк.	№ документа	Підпис	Дата		



					171.6321М.КР.ПЗ.Р4			
Змн.	Арк.	№ документу	Підпис	Дата				
Розробник		Пелипас В.В.			Охорона праці	Літ.	Арк.	Аркуші
Консультант		Губальцев А. М.					61	
Н. контр.		Добрінова Л. П.				НУК імені адмірала Макарова		
Керівник		Ушкаренко О.О.						
Зав. каф.		Рябенський В. М.						

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1. Огляд правової безпеки

Одна з ключових задач держави — забезпечення безпеки та збереження здоров'я населення під час їхньої трудової діяльності шляхом створення комфортних і безпечних умов праці. За інформацією Міжнародної організації праці, смертність від нещасних випадків займає третє місце після серцево-судинних та онкологічних хвороб, причому молоді люди працездатного віку до 40 років частіше стають жертвами. Аналіз смертності в Україні за період 1994–1998 років показує, що головними чинниками смертності серед чоловіків працездатного віку є нещасні випадки, отруєння та травми, які складають близько 30–35% усіх смертей у цій групі.

Охорона праці має вагоме значення як суспільний фактор, оскільки навіть видатні професійні досягнення не здатні компенсувати втрату здоров'я чи життя людини. Окрім соціальної значущості, охорона праці також має економічну цінність: підвищує ефективність роботи, знижує витрати на лікування, а також скорочує виплати за важкі та небезпечні умови праці. За оцінками експертів МОП, економічні втрати від нещасних випадків становлять близько 1% світового валового національного продукту.

Відповідно до статистики МОП, кількість нещасних випадків на робочих місцях продовжує зростати й щорічно досягає 125 мільйонів випадків, із яких приблизно 220 тисяч закінчуються смертю. Рівень травматизму та професійних захворювань у країнах, що розвиваються, значно перевищує показники промислово розвинених держав.

Закордонні фахівці, які працювали в Україні в рамках програми МОП, визначили п'ять головних причин високого рівня смертельних нещасних випадків: недостатню підготовку працівників та роботодавців із питань безпеки, слабкий контроль за безпечними умовами на робочих місцях, погане забезпечення працівників індивідуальними засобами захисту, повільне впровадження колективних заходів безпеки та використання застарілого обладнання.

					171.6321М.КР.ПЗ.Р4	Арк.
						62
Змн.	Арк.	№ документу	Підпис	Дата		

Правовою основою охорони праці в Україні є Конституція та низка законів: «Про охорону праці», «Про охорону здоров'я», «Про пожежну безпеку», «Про використання ядерної енергії та радіаційний захист», «Про санітарне та епідеміологічне благополуччя населення», а також Кодекс законів про працю. Основним нормативним актом у цій сфері є Закон «Про охорону праці», який поширюється на всі підприємства незалежно від форми власності, галузі діяльності та кількості працівників. Охорона праці — це сукупність заходів, спрямованих на збереження здоров'я та працездатності людини під час виконання робочих обов'язків.

4.2. Розрахунок захисного заземлення

Вихідні дані.

- Виконати розрахунок захисного заземлення для електроустановки з напругою до 1000 В.

- Припустимий опір заземлюючого пристрою: $R_0 = 40 \text{ Ом}$;
- Тип заземлювача – стрижневий у землі;
- Довжина стержня: $l = 5 \text{ м}$;
- Діаметр стержня: $d = 16 \text{ мм} = 16 \cdot 10^{-3} \text{ м}$, глибина закладення:

$t_0 = 0.7 \text{ м}$;

- Сполучна смуга – протяжна смугова в землі;
- Ширина смуги: $b = 45 \text{ мм} = 45 \cdot 10^{-3} \text{ м}$.
- Ґрунт – глина.

Розрахунок.

1. Припустимий опір заземлюючого пристрою $R_0 = 40 \text{ Ом}$.

2. Визначаємо питомий опір ґрунту для чорнозему:

$$\rho = 60 \text{ Ом} \cdot \text{м}.$$

3. Вибираємо матеріали й розміри заземлювача. Для обраного типу й розмірів визначаємо опір розтіканню одного заземлювача:

					171.6321М.КР.ПЗ.Р4	Арк.
						63
Змн.	Арк.	№ документу	Підпис	Дата		

$$R_1 = \frac{\rho}{2 \cdot \pi \cdot l} \cdot \left(\ln \frac{2l}{d} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l}{4 \cdot t - l} \right), \quad (4.1)$$

R_1 - опір розтікання;

l - довжина стержня;

t_0 - глибина закладення труби від поверхні землі;

t - глибина закладення труби рівна відстані від поверхні землі до середини труби.

$$t = t_0 + \frac{1}{2} \cdot l; \quad (4.2)$$

$$t = 0.7 + \frac{1}{2} \cdot 5 = 3.2 \text{ м};$$

$$R_1 = \frac{60}{2 \cdot \pi \cdot 5} \cdot \left(\ln \frac{2 \cdot 5}{16 \cdot 10^{-3}} + \frac{1}{2} \cdot \ln \frac{4 \cdot 3.2 + 5}{4 \cdot 3.2 - 5} \right); \quad R_1 = 13.083 \text{ Ом}.$$

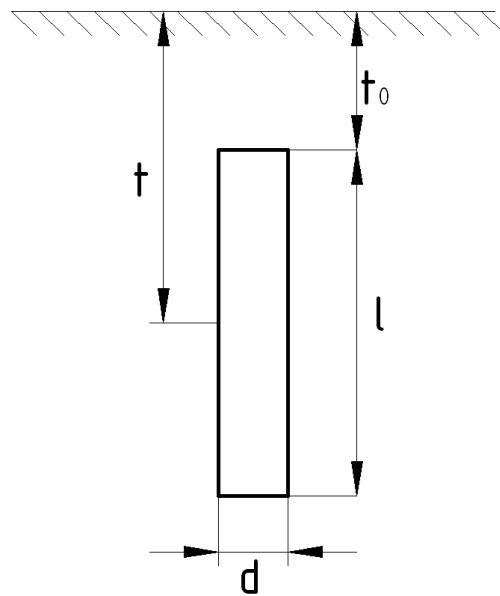


Рисунок 4.1 – Стрижневий заземлювач у землі

4. З винятку $R_1 > R_0$ необхідно застосовувати декілька штучних заземлювачів з'єднаних паралельно.

5. Визначаємо необхідне число паралельно з'єднаних заземлювачів:

$$n = \frac{R_1}{\eta_e \cdot R_0}, \quad (4.3)$$

η_e - коефіцієнт використання вертикальних заземлювачів;

					171.6321М.КР.ПЗ.Р4	Арк.
						64
Змн.	Арк.	№ документа	Підпис	Дата		

$$\eta_g = 1.$$

$$n = \frac{13.083}{1.4}; n = 3.271 \Rightarrow n = 3.$$

Спочатку приймаємо $n_g = 1$, отримане число заземлювачів округлимо до цілого числа n . Знаходимо фактичний коефіцієнт використання заземлювачів η_ϕ , визначаємо фактичний опір вертикальних штучних заземлювачів $R_{\phi\phi}$:

$$R_{\phi\phi} = \frac{R_1}{n \cdot \eta_\phi}; \quad (4.4)$$

$$R_{\phi\phi} = \frac{13.083}{3 \cdot 0.76}; R_{\phi\phi} = 5.738 \text{ Ом}.$$

6. Для зв'язку вертикальних заземлювачів застосовуємо сполучну смугу. Опір розтікання сполучної смуги визначаємо як опір розтікання протяжної смуговий у землі:

$$R_{1II} = \frac{\rho}{2 \cdot \pi \cdot l} \cdot \ln \frac{2 \cdot l^2}{b \cdot t}, \quad (4.5)$$

l - довжина смуги.

a - відстань між вертикальними заземлювачами.

Довжину смуги з'єднаних вертикально заземлювачів знаходимо по формулі:

$$l = 1.05 \cdot a \cdot (n - 1); \quad (4.6)$$

$$a = 1 \cdot l_{эл}; \quad (4.7)$$

$$l = 1.05 \cdot 5 \cdot 3; l = 10.5 \text{ м}.$$

Визначаємо опір:

$$R_{1II} = \frac{60}{2 \cdot \pi \cdot l} \cdot \ln \frac{2 \cdot l^2}{b \cdot t};$$

$$R_{1II} = \frac{60}{2 \cdot \pi \cdot 10.5} \cdot \ln \frac{2 \cdot 10.5^2}{45 \cdot 10^{-3} \cdot 0.7};$$

$$R_{1II} = 8.06 \text{ Ом}.$$

З урахуванням коефіцієнта використання смуги електрода або горизонтального заземлювача:

$$R_{II} = \frac{R_{1II}}{\eta_r}. \quad (4.8)$$

					171.6321М.КР.ПЗ.Р4	Арк.
						65
Змн.	Арк.	№ документа	Підпис	Дата		

$$\eta_{\Gamma} = 0.77 .$$

$$R_{\Pi} = \frac{8.06}{0.77} ; R_{\Pi} = 10.46 \text{ Ом} .$$

7. Фактичний опір заземлюючого пристрою з вертикальних заземлювачів сполучної смуги, визначаємо як еквівалент опору при паралельному з'єднанні:

$$R_3 = \frac{R_{\phi\phi} \cdot R_{\Pi}}{R_{\phi\phi} + R_{\Pi}} ; \quad (4.9)$$

$$R_3 = \frac{5.738 \cdot 10.46}{5.738 + 10.46} ;$$

$$R_3 = 3.7 \text{ Ом} .$$

$R_3 \leq R_0$ отже, розрахунок зроблений правильно.

4.3. Захист від ураження струмом

Ізоляція та контроль струмоведучих частин

Ізоляція струмоведучих елементів із системами контролю забезпечує стабільну роботу електроустановок та захищає від ураження струмом. Якщо основна ізоляція пошкоджується, для безпеки передбачається додаткова. У поєднанні ці два шари утворюють подвійну ізоляцію. Якщо ж ізоляція одного шару забезпечує такий самий рівень захисту, як подвійна, її називають посиленою ізоляцією.

Контроль ізоляції може бути періодичним або безперервним. Для вимірювання опору ізоляції використовуються мегомметри. Перевірка виконується:

- під час приймально-здавальних випробувань після монтажу чи ремонту;
- при виявленні ознак пошкодження ізоляції;
- у межах встановлених нормативних строків технічного обслуговування.

Нормативи опору ізоляції регламентуються правилами улаштування електроустановок:

					171.6321М.КР.ПЗ.Р4	Арк.
						66
Змн.	Арк.	№ документу	Підпис	Дата		

- Для котушок, пускачів, контактів, силових щитів та освітлювальних систем: не менше 0,5 МОм.

- Для вторинних ланцюгів: не менше 1 МОм.

Постійний контроль ізоляції виконується за допомогою спеціальних пристроїв, які інтегруються в коло разом з електроустановкою. Вони автоматично вимірюють опір та подають сигнал, якщо значення опору знижується до небезпечного рівня.

Огородження та захист від контакту зі струмоведучими частинами

Для попередження випадкового дотику до струмоведучих частин або небезпечного наближення до них використовуються огороження. Основні вимоги до таких засобів:

- Механічна міцність — здатність витримувати зовнішні впливи без пошкоджень.

- Електрична надійність — забезпечення ефективного захисту від ураження струмом протягом усього терміну експлуатації.

4.4. Заходи з техніки безпеки

Інструкції з техніки безпеки під час роботи з електронним обладнанням

1. Осцилограф та безпека

- Заборонено вмикати осцилограф без кришки, щоб уникнути ризику травм.

- Перед заміною запобіжників обов'язково відключіть пристрій від електромережі.

- Використовуйте лише стандартні запобіжники — саморобні не допускаються.

- Не залишайте ввімкнений осцилограф без нагляду.

- Якщо в осцилограф потрапила рідина або він перебував у вологих умовах, вмикати його можна лише через 24 години.

- Вентиляційні отвори осцилографа мають залишатися відкритими для запобігання перегріванню.

					171.6321М.КР.ПЗ.Р4	Арк.
						67
Змн.	Арк.	№ документа	Підпис	Дата		

- Розміщуйте осцилограф подалі від джерел тепла.

2. Заходи безпечної експлуатації електроустановок

Системи безпеки в електроустановках поділяються на дві групи:

- Умови нормальної експлуатації: захист користувача в штатних ситуаціях.

- Аварійний режим: заходи безпеки для попередження травм або виходу обладнання з ладу під час позаштатних ситуацій.

3. Ізоляція як захисний механізм

Електрична ізоляція забезпечує захист шляхом обмеження протікання струму завдяки діелектричному покриттю. В електроустановках напругою до 1000 В ізольовані дроти є ефективним захистом від ураження струмом. Однак при напрузі понад 1000 В навіть ізольовані проводи залишаються небезпечними, тому використовуються огорожувальні пристрої — суцільні або сітчасті.

4. Блокування та фізичний доступ

Блокування — це автоматична система, що запобігає небезпечним діям. Електричні блокування вимикають напругу при відкритті захисних огорожень чи знятті кришок. Також безпечно розміщення проводів на недосяжній висоті зменшує ризик ураження струмом.

5. Низькі напруги та їхні джерела

Струм напругою до 42 В вважається безпечним для людини. Джерела низької напруги включають знижувальні трансформатори, акумулятори та випрямні установки.

6. Вирівнювання потенціалів та заземлення

Вирівнювання потенціалів мінімізує напругу між точками, до яких людина може одночасно доторкнутися. Захисне заземлення полягає у з'єднанні металевих частин із землею для зниження напруги дотику та попередження травм при замиканні на корпус.

7. Захисне вимкнення

					171.6321M.KP.ПЗ.P4	Арк.
						68
Змн.	Арк.	№ документу	Підпис	Дата		

Автоматичне вимкнення електроустановок при небезпеці ураження струмом забезпечує швидке реагування на аварії та знижує ризик травмування.

8. Електрозахисні засоби

Електрозахисні засоби поділяються на основні та допоміжні. Основні засоби, як-от ізольовані рукоятки інструментів, витримують робочу напругу та дозволяють працювати з елементами під напругою. Допоміжні засоби — це додаткові пристрої, які підсилюють захист, але не можуть повністю замінити основні засоби.

					171.6321М.КР.ПЗ.Р4	Арк.
						69
Змн.	Арк.	№ документу	Підпис	Дата		

ВИСНОВКИ

У процесі дослідження цифрової обробки аудіо сигналів стало очевидним, що вона є невід'ємною частиною сучасних технологій, що використовуються в різних сферах, від музичної індустрії до телекомунікацій. Аналіз показав, що цифрова обробка сигналів суттєво підвищує якість і гнучкість обробки звукових даних, дозволяючи здійснювати ефективне управління аудіосигналами в реальному часі.

Основним інструментом у цифровій обробці є дискретизація сигналу, яка дозволяє перетворювати аналогові сигнали в цифрову форму для подальшої обробки. Важливими аспектами є використання фільтрів низьких частот, які усувають небажані спектри та спотворення, що виникають в процесі дискретизації. Це дозволяє запобігти накладенню спектрів і зберегти цілісність корисних сигналів.

Розробка і оптимізація характеристик аналогових фільтрів, таких як фільтри Баттерворта, є критично важливими для забезпечення необхідного рівня загасання і запобігання впливу шуму на якість обробки. Динамічний діапазон та точність представлення сигналу також вимагають ретельної уваги, адже від них залежить здатність системи коректно сприймати і обробляти корисні дані в умовах наявності перешкод.

Наукова новина, що підкреслює важливість цих технологій, стосується нещодавно опублікованого дослідження, яке демонструє новий алгоритм для цифрової обробки аудіо сигналів, здатний зменшити шум до 90% без втрати якості звучання. Дослідники з університету Стенфорда застосували методи глибокого навчання для оптимізації фільтрації сигналів, що відкриває нові горизонти в створенні акустичних систем.

Отже, цифрова обробка аудіосигналів є складним, але надзвичайно важливим процесом, що вимагає мультидисциплінарного підходу для досягнення оптимальних результатів. Розвиток нових фільтрів та технологій обробки сприятиме підвищенню якості аудіо та створенню інноваційних рішень у цій галузі.

					171.6321М.КР.ПЗ.Висновки	Арк.
						70
Змн.	Арк.	№ документу	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Качан В. Л., Привалов І. Ю., Опанасенко П. М. Цифрова обробка сигналів: підручник. – Київ: Національний технічний університет України "КПІ", 2020. – 268 с.
2. Привалов І. Ю. Методи фільтрації та корекції аудіосигналів у цифрових системах: монографія. – Харків: ХНУРЕ, 2018. – 310 с.
3. Опанасенко П. М. Основи програмування звукових карт на базі Arduino. – Львів: Видавництво "Світ", 2021. – 198 с.
4. Smith S. W. The Scientist and Engineer's Guide to Digital Signal Processing. – California Technical Pub, 1999. – 626 p.
5. Zölzer U. Digital Audio Signal Processing. – 2nd ed. – Wiley & Sons, 2002. – 384 p.
6. Bosi M., Goldberg R. E. Introduction to Digital Audio Coding and Standards. – Springer, 2003. – 450 p.
7. Голік Ю. М. Цифрові методи аналізу та синтезу аудіосигналів. – Дніпро: ДНУ, 2017. – 245 с.
8. Ekholm T. Digital Sound Processing Techniques for Music Applications. – Cambridge University Press, 2015. – 275 p.
9. Davis G. F. Sound System Engineering. – 4th ed. – Focal Press, 2007. – 490 p.
10. Іванов С. О. Спектральний аналіз сигналів у звукових системах. – Запоріжжя: ЗНТУ, 2022. – 340 с.
11. Case A. Mix Smart: Pro Audio Tips for Your Multitrack Mix. – Focal Press, 2012. – 312 p.
12. Кравчук М. Л. Комп'ютерне відтворення та обробка аудіо. – Вінниця: ВНТУ, 2019. – 185 с.
13. Гончар Д. В. Системи контролю та корекції аудіо сигналів у реальному часі. – Одеса: ОНПУ, 2018. – 298 с.
14. Pohlmann K. C. Principles of Digital Audio. – 6th ed. – McGraw-Hill, 2010. – 672 p.

					171.6321М.КР.ПЗ.Джерела	Арк.
						71
Змн.	Арк.	№ документу	Підпис	Дата		

15. Moorer J. A. Audio Signal Processing in the Time Domain. – IEEE Press, 1992. – 215 p.
16. Brandenburg K. Applications of Digital Signal Processing to Audio and Acoustics. – Springer, 1999. – 344 p.
17. Lathi B. P., Green Z. Linear Systems and Signals. – 3rd ed. – Oxford University Press, 2014. – 992 p.

					171.6321М.КР.ПЗ.Джерела	Арк.
						72
Змн.	Арк.	№ документу	Підпис	Дата		

ДОДАТОК А

Лістинг програми синтезу звуку

```
struct WAVESTRUCT
{
    char formatId[4];          // Містить символи "RIFF" в ASCII
                                // (0x52494646 в big-endian представленні).
                                // Це початок RIFF-ланцюга.

    int blockLength;           // Залишковий розмір ланцюга,
                                // починаючи з цієї позиції.
                                // Іншими словами, це розмір файлу
                                // мінус 8 байт.

    char soundId[4];           // Містить символи "WAVE" (0x57415645
                                // в big-endian представленні).

    char subId[4];             // Містить символи "fmt " (0x666d7420
                                // в big-endian представленні).

    int headerLength;          // 16 для формату PCM. Це залишковий
                                // розмір підланцюга,
                                // починаючи з цієї позиції.

    short formatType;           // Аудіо формат, повний список можна
                                // знайти тут. Для PCM = 1 (тобто,
                                // Лінійне квантування). Значення, що
                                // відрізняються від 1, позначають
                                // деякий формат стиснення.

    short channels;             // Кількість каналів. Моно = 1, Стерео
                                // = 2.

    int frequency;             // Частота дискретизації. 8000 Гц,
                                // 44100 Гц тощо.

    int dataRate;              // Кількість байтів, переданих за
                                // секунду відтворення.

    short sampleBytes;          // Кількість байтів для одного зразка,
                                // включаючи всі канали.

    short sampleWidth;          // Кількість біт у зразку. Так звана
                                // "глибина" або точність
                                // звучання. 8 біт, 16 біт тощо.
```

					171.6321М.КР.ПЗ.Додатки	Арк.
						73
Змн.	Арк.	№ документу	Підпис	Дата		

```

    char dataId[4];          // Містить символи "data" (0x64617461 в
big-endian представленні).
    int dataLength;          // Кількість байтів у області даних.
};

void CSound3391Dlg::OnButton14()
{
    HWAVEIN micro_handle; // Ідентифікатор пристрою, отриманий
функцією waveInOpen;
    WAVEHDR wave_header;   // Структура містить опис буфера та
    // посилання на масив
    // передаваних даних

    const int t = 5;

    const int num_pts = 22050 * t; // Кількість вибірок за t
секунд
    int sample_rate = 22050;        // Частота дискретизації
    short int wave_in[num_pts];     // 'short int' це 16-
розрядний тип даних
    // для 8-розрядних
використовується 'unsigned char' або 'BYTE'
    MMRESULT result = 0;

    WAVEFORMATEX format;
    format.wFormatTag = WAVE_FORMAT_PCM; // Простий, не
стиснутий формат
    format.wBitsPerSample = 16; // 16 для високої
якості, 8 для телефонного
    format.nChannels = 1; // 1 = моно, 2 =
стерео
    format.nSamplesPerSec = sample_rate; // 22050
    format.nAvgBytesPerSec = format.nSamplesPerSec *
format.nChannels * format.wBitsPerSample / 8; // =
nSamplesPerSec * n.Channels * wBitsPerSample / 8

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						74
Змн.	Арк.	№ документу	Підпис	Дата		

```

format.nBlockAlign          =          format.nChannels          *
format.wBitsPerSample / 8; // = n.Channels * wBitsPerSample / 8
format.cbSize = 0;

result = waveInOpen(&micro_handle, WAVE_MAPPER, &format, 0L,
0L, WAVE_FORMAT_DIRECT);

/* Коментар
Функція
UINT waveInOpen(
    LPHWAVEIN lphWaveIn,          // Вказівник на ідентифікатор
пристрою
    UINT wDeviceID,              // Номер відкриваемого пристрою
    LPWAVEFORMAT lpFormat,       // Вказівник на структуру WAVEFORMAT
    DWORD dwCallback,            // Адреса функції зворотного виклику
                                // або ідентифікатор вікна
    DWORD dwCallbackInstance,    // Дані для функції зворотного
виклику
    DWORD dwFlags);              // Режим відкриття пристрою

if (result)
{
    Sleep(1000);
    return;
}

// Налаштування та підготовка структури заголовка даних для
запису
wave_header.lpData = (LPSTR)wave_in;
wave_header.dwBufferLength = num_pts * 2;
wave_header.dwBytesRecorded = 0;
wave_header.dwUser = 0L;
wave_header.dwFlags = 0L;
wave_header.dwLoops = 0L;
waveInPrepareHeader(micro_handle,          &wave_header,
sizeof(WAVEHDR));

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						75
Змн.	Арк.	№ документа	Підпис	Дата		

```

/* Коментар
Функція
UINT waveInPrepareHeader(
    HWAVEIN hWaveIn,          // Ідентифікатор пристрою
    LPWAVEHDR lpWaveInHdr,    // Вказівник на структуру WAVEHDR
    UINT wSize);              // Розмір структури WAVEHDR

typedef struct wavehdr_tag {
    LPSTR lpData;              // Адреса буфера даних - посилання на
СИМВОЛЬНИЙ МАСИВ
    DWORD dwBufferLength;      // Розмір буфера даних
    DWORD dwBytesRecorded;     // Кількість записаних байтів
    (тільки при запису)
    DWORD dwUser;              // Користувацькі дані
    DWORD dwFlags;             // Прапори стану буфера даних
    DWORD dwLoops;             // Кількість повторень (тільки
при відтворенні)
    struct wavehdr_tag far * lpNext; // Зарезервовано
    DWORD reserved;            // Зарезервовано
} WAVEHDR;
*/

// Додавання первинного буфера
result = waveInAddBuffer(micro_handle, &wave_header,
sizeof(WAVEHDR));

/* Коментар
Функція
UINT waveInAddBuffer(
    HWAVEIN hWaveIn,          // Ідентифікатор пристрою
    LPWAVEHDR lpWaveInHdr,    // Вказівник на структуру WAVEHDR
    UINT wSize);              // Розмір структури WAVEHDR
Передає підготовлений буфер пам'яті драйверу пристрою вводу.
Параметри аналогічні функції waveInPrepareHeader.
*/

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						76
Змн.	Арк.	№ документа	Підпис	Дата		

```

if (result)
{
    Sleep(1000);
    return;
}

result = waveInStart(micro_handle);

/* Коментар
Функція
UINT waveInStart (HWAVEIN hWaveIn) // Ідентифікатор
пристрою, отриманий waveInOpen.
Запускає процес вводу даних.
*/

if (result)
{
    Sleep(1000);
    return;
}

// Очікування, поки виконується запис даних у буфер
do {} while (waveInUnprepareHeader(micro_handle,
&wave_header, sizeof(WAVEHDR)) == WAVERR_STILLPLAYING);

/* Коментар
Функція
UINT waveInUnprepareHeader(
HWAVEIN hWaveIn, // Ідентифікатор пристрою вводу
LPWAVEHDR lpWaveInHdr, // Вказівник на структуру WAVEHDR
UINT wSize); // Розмір структури WAVEHDR

Звільняє буфер вводу.
*/

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						77
Змн.	Арк.	№ документа	Підпис	Дата		

```

waveInClose (micro_handle);

/* Коментар
Функція
UINT waveInClose (HWAVEIN hWaveIn); // Ідентифікатор
пристрою вводу
Закриває пристрій вводу.
*/

// Відтворення звуку
int fd = 22050;

WAVESTRUCT sws;
strcpy (sws.formatId, "RIFF");
sws.blockLength = sizeof (data) + 36;
strcpy (sws.soundId, "WAVE");
strcpy (sws.subId, "fmt ");
sws.headerLength = 16;
sws.formatType = 1;
sws.channels = 1;
sws.frequency = fd;
sws.dataRate = fd;
sws.sampleBytes = 2;
sws.sampleWidth = 16;
strcpy (sws.dataId, "data");
sws.dataLength = fd * sizeof (short) * t;

// Динамічна корекція сигналу
for (int i = 0; i < num_pts; i++)
{
    if (wave_in[i] < 8096)
        wave_in[i] = wave_in[i] * 2; // Підсилення слабкого
сигналу в 2 рази
}

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						78
Змн.	Арк.	№ документа	Підпис	Дата		

ДОДАТОК Б

Лістинг програми децимації аудіо сигналу

```
#include "stdafx.h"
#include "WaveSample.h"
#pragma comment(lib, "winmm.lib")

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

CWinApp theApp;
using namespace std;

struct WAVESTRUCT
{
    char formatId[4];
    int blockLength;
    char soundId[4];
    char subId[4];
    int headerLength;
    short formatType;
    short channels;
    int frequency;
    int dataRate;
    short sampleBytes;
    short sampleWidth;
    char dataId[4];
    int dataLength;
};

int main(int argc, TCHAR* argv[], TCHAR* envp[])
{
    int nRetCode = 0;
```

					171.6321М.КР.ПЗ.Додатки	Арк.
						79
Змн.	Арк.	№ документи	Підпис	Дата		

```

// Initialize MFC
if (!AfxWinInit(::GetModuleHandle(NULL), NULL,
::GetCommandLine(), 0))
{
    cerr << _T("Fatal Error: MFC initialization failed")
<< endl;

    nRetCode = 1;
}
else
{
    // File handling
    WAVESTRUCT wavStruct;
    CFile inputFile;
    CFile outputFile;
    const char* inputPath = "e:\\s1.wav";
    const char* outputPath = "e:\\nw.wav";

    // Open input WAV file
    if (!inputFile.Open(inputPath, CFile::modeRead |
CFile::typeBinary))
    {
        cout << "File error!";
        return 0;
    }

    // Read the WAV header
    inputFile.Read(&wavStruct, sizeof(wavStruct));
    int dataSize = wavStruct.dataLength;

    // Allocate memory for audio data
    byte* audioData = new byte[dataSize];
    inputFile.Read(audioData, dataSize);
    inputFile.Close();

    // Open output WAV file

```

					171.6321М.КР.ПЗ.Додатки	Арк.
						80
Змн.	Арк.	№ документи	Підпис	Дата		

```

        outputFile.Open(outputPath, CFile::modeWrite |
CFile::typeBinary | CFile::modeCreate);
        byte* decimatedData = new byte[dataSize];
        memset(decimatedData, 0, dataSize);

        // Update WAV header for decimated data
        wavStruct.frequency /= 2;
        wavStruct.dataRate /= 2;
        wavStruct.dataLength = dataSize / 2;

        // Implement decimation
        for (int i = 0; i < dataSize / 2; i++)
        {
            decimatedData[i] = audioData[2 * i];
        }

        // Write new WAV file
        outputFile.Write(&wavStruct, sizeof(wavStruct));
        outputFile.Write(decimatedData,
wavStruct.dataLength);
        outputFile.Close();

        // Clean up memory
        delete[] audioData;
        delete[] decimatedData;
    }

    // Play the new WAV file
    PlaySound("c:\\new.wav", NULL, SND_FILENAME |
SND_ASYNC);
    return nRetCode;
}

```