

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Модель адаптивного цифрового фільтру звукових сигналів на ПЛІС
(тема)

Виконав:
студент 2 курсу, групи СКСм-21-1
Дубов І.Г.
(прізвище, ініціали)

Спеціальність 123 Комп'ютерна інженерія
(код і повна назва спеціальності)


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

Освітня програма Спеціалізовані комп'ютерні системи
(повна назва освітньої програми)

Керівник доц. Філіппенко І.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри АПОТ

 Чумаченко С.В.
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління

Кафедра Автоматизації проектування обчислювальної техніки

Рівень вищої освіти другий (магістерський)

Спеціальність 123 Комп'ютерна інженерія
(код і повна назва)

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

Освітня програма Спеціалізовані комп'ютерні системи
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

«_____» _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Дубову Івану Геннадійовичу
(прізвище, ім'я, по батькові)

1. Тема роботи Модель адаптивного цифрового фільтра звукових сигналів на ПЛІС

затверджена наказом університету від 14 листопада 2022 р. № 1478Ст

2. Термін подання студентом роботи до екзаменаційної комісії 17 12 2022 р.

3. Вихідні дані до роботи _____

цифровий фільтр
ПЛІС

4. Перелік питань, що потрібно опрацювати в роботі _____

Аналіз предметної області та постановка задачі.

Розробка моделі

Програмна реалізація


Аналіз отриманих даних.


5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) _____
17 слайдів

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання	04.11.2022 - 05.11.2022	
2	Аналіз літератури	12.11.2022 - 16.11.2022	
3	Розробка моделі	17.11.2022 - 26.11.2022	
4	Реалізація моделі	29.11.2022 - 05.12.2022	
5	Тестування отриманих даних	06.12.2022 - 10.12.2022	
6	Оформлення пояснювальної записки	13.12.2022 - 17.12.2022	

Дата видачі завдання __05__ __09__ 2022 р.

Студент  _____
(підпис)

Керівник роботи  _____ доц. Філіппенко І.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 53 сторінки, 20 рисунків, 24 джерела за переліком посилань.

ЦИФРОВА ОБРОБКА СИГНАЛІВ, ЦИФРОВІ ФІЛЬТРИ, MATLAB,
ПЛІС

В ході виконання кваліфікаційної роботи проведено аналіз алгоритмів адаптивних фільтрів. Розглянуті етапи проектування цифрових фільтрів за допомогою MatLab і Simulink.

Було виконане проектування адаптивного цифрового фільтру на ПЛІС. Було отримано VHDL опис фільтру. Проведено синтез та тестування спроектованого фільтру у середовищі пакету Altera Quartus.

ABSTRACT

Bachelor's thesis contains 53 pages, 20 figures, 24 sources according to the list of links.

DIGITAL SIGNAL PROCESSING, DIGITAL FILTERS, MATLAB, FPGA

In the course of the qualification work, an analysis of adaptive filter algorithms was carried out. Considered stages of designing digital filters using MatLab and Simulink.

The design of the adaptive digital filter on the FPGA was performed. A VHDL description of the filter was obtained. The designed filter was synthesized and tested in the environment of the Altera Quartus package.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	8
ВСТУП.....	9
1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ АДАПТИВНОЇ ФІЛЬТРІВ	11
1.1 Адаптивні алгоритми у цифровій обробці інформації.....	11
1.2 Загальна структура адаптивного фільтра	12
1.3 Алгоритми адаптації	14
1.4 Алгоритм Вінера	17
1.5 Алгоритм RLS.....	18
1.6 Алгоритм афінних проєкцій	19
1.7 Порівняння алгоритмів адаптивної фільтрації	20
2 РОЗРОБКА МОДЕЛІ АДАПТИВНОГО ФІЛЬТРА	22
2.1 Математичні пакети для побудови моделі фільтра	22
2.2 Вимоги до моделі адаптивного фільтра звукових сигналів	24
2.3 RLS фільтр.....	24
2.4 Побудова моделі адаптивного фільтра.....	28
3 РОЗРОБКА АПАРАТНОЇ РЕАЛІЗАЦІЇ АДАПТИВНОГО ФІЛЬТРА	40
3.1 Вибір апаратної платформи	40
3.2 Реалізація фільтру	42
ВИСНОВКИ	49
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	51
ДОДАТОК А	Ошибка! Закладка не определена.
ДОДАТОК Б	Ошибка! Закладка не определена.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ

- АЦП – аналогово-цифровий перетворювач;
ЦАП – цифроаналоговий перетворювач;
ЦОС – цифрова обробка сигналів;
ЦСП (DSP) – цифровий сигнальний процесор;
АЧХ – амплітудно-частотна характеристика;
НІХ – фільтр з нескінченною імпульсною характеристикою;
СІХ – фільтр зі скінченною імпульсною характеристикою;
ФЧХ – фазочастотна характеристика фільтра;
ПЛІС – програмована логічна інтегральна схема;
ФНЧ – фільтр нижніх частот;
ЦФ – цифровий фільтр;
ФАПЧ – фазове автопідстроювання частоти;
LMS – least mean squares;
RLS – recursive least squares.

ВСТУП

У даний час неухильно зростає потреба вирішення питань з цифрової обробки сигналу в системах різноманітного призначення. Це зумовлено значними нароботками у теорії та практиці цифрової обробки інформації за останній час, а також технологічними досягненнями в радіоелектронній техніці та збільшенню кількості вентелей в кристалі. Ці досягнення дозволили створювати високопродуктивні та одночасно компактні цифрові пристрої для реалізації алгоритмів цифрової обробки інформації.

Цифрова обробка сигналів (ЦОС) сьогодні широко використовується в радіолокаційних та радіонавігаційних системах, в провідному та оптичному зв'язку й радіозв'язку, обробці медичних сигналів, в обчислювальній та побутовій техніці. Відмінною особливістю цих задач є великий обсяг обчислень, які необхідно виконувати в реальному часі.

Однією з таких пріоритетних задач є вирішення питання придушення завад, що виникають в перевідображенні радіохвиль. Для вирішення даної проблеми застосовуються різні підходи, в тому числі в технології цифрової адаптивної фільтрації сигналів.

Адаптивна обробка сигналів, або іншими словами адаптивна фільтрація, використовується, коли з поставленим завданням не можуть впоратися фільтри з фіксованими параметрами. Це відбувається, як правило, у випадках, коли умови фільтрації невідомі або змінюються, тому вимоги до параметрів фільтра не можуть бути сформульовані заздалегідь.

Сьогодні існує багато радіоелектронних виробів, ефективна робота яких неможлива без використання адаптивних фільтрів. Це адаптивні антенні решітки, компенсатори сигналів електричної та акустичної луни, еквалайзери (вирівнювачі) електричних характеристик каналів зв'язку, компенсатори шумів та ряд інших пристроїв.

Таким чином є актуальною проблема розробки адаптивного фільтру, який можна буде використовувати для фільтрації сигналів звукового діапазону для приглушення акустичних завад в різноманітних пристроях в режимі реального часу.

1 АНАЛІЗ СУЧАСНИХ МЕТОДІВ АДАПТИВНОЇ ФІЛЬТРІВ

1.1 Адаптивні алгоритми у цифровій обробці інформації

На сьогоднішній день існує багато радіоелектронних виробів, ефективна робота яких неможлива без використання адаптивних фільтрів. Основні області застосування адаптивної фільтрації – очищення даних від нестабільних сигналів, що заважають, і шумів, що перекриваються по спектру зі спектром корисних сигналів, або коли смуга частот що заважають невідома, змінна і не може бути задана апріорно для розрахунку параметричних фільтрів [1]. Ефективне вирішення цих проблем можливе лише з використанням адаптивних фільтрів.

Як правило, адаптивну фільтрацію застосовують у тих випадках, коли фільтри з фіксованими параметрами не можуть виконувати поставлену задачу. Наприклад, якщо умови фільтрації змінюються, тому вимоги до фільтра не можуть бути сформульовані заздалегідь [2]. Частотна характеристика адаптивних фільтрів автоматично регулюється або модифікується відповідно до певного критерію, що дозволяє фільтру адаптуватися до змін характеристик вхідного сигналу. Вони досить широко використовуються в радіо- та гідролокації, в системах навігації, у виділенні біомедичних сигналів та багатьох інших галузях техніки.

Адаптивною обробкою сигналів займалися такі вчені, як Н. Вінер, А.Н.Колмогорів, А.Крайн, М.Левінсон. Вперше адаптивну обробку сигналів на практиці застосував Бернард Уїдрей. Він розробив і реалізував перший адаптивний фільтр на базі Least Mean Square (LMS) алгоритму, також відомого як алгоритм по методу найменшого.

Основні області застосування адаптивної фільтрації – очищення даних від нестабільних сигналів, що заважають, і шумів, що перекриваються по спектру зі спектром корисних сигналів, або коли смуга заважаючих частот

невідомо, змінна і не може бути задана апріорно для розрахунку параметричних фільтрів. Так, наприклад, цифровий зв'язок з сильною активною перешкодою може інтерферувати з корисним сигналом, а при передачі цифрової інформації по каналам з поганими частотними характеристиками може спостерігатися міжсимвольна інтерференція цифрових кодів. Ефективне вирішення цих проблем можливе лише за допомогою адаптивних фільтрів [3].

Адаптивний фільтр – це фільтр з змінними в процесі роботи параметрами, набір яких багато в чому залежить від критерію роботи адаптивного фільтра. Цим критерієм часто є досягнення мінімальної цільової функції, як правило, квадратичної функції помилки між необхідним та вихідним сигналами адаптивного фільтра. Досягнення мінімальної цільової функції означає, що вихідний сигнал адаптивного фільтра в певній мірі наближений до потрібного сигналу, фізична природа якого визначається конкретним додатком адаптивного фільтра.

У залежності від вирішуваної задачі і типу оброблюваного сигналу адаптивні фільтри можуть бути одноканальними або багатоканальними з дійними або комплексними ваговими коефіцієнтами. Алгоритми обчислення вісових коефіцієнтів можуть бути простими, як, наприклад, LMS-алгоритм, або складними, як, наприклад, Recursive Least Squares (RLS) алгоритм.

1.2 Загальна структура адаптивного фільтра

Розглянемо загальну структуру адаптивного фільтра, яка наведена на рисунку 1.1. Вхідний дискретний сигнал $x(k)$ обробляється дискретним фільтром, в результаті чого формується сигнал $y(k)$. Цей вихідний сигнал порівнюється з сигналом-зразком $d(k)$. Різниця між цими двома утворює сигнал помилки $e(k)$.

Задачею адаптивного фільтра є мінімізація помилки відтворення зразкового сигналу. З цією метою блок адаптації після обробки кожного відліка аналізує сигнал похибки і додаткові дані, що надходять з фільтра, використовує ці дані для налаштування параметрів фільтра, а саме його коефіцієнтів.

Можливий інший варіант адаптації, при якому зразковий сигнал не використовується. Такий режим роботи називається «сліпою адаптацією» [3]. Цей метод є більш складною обчислювальною задачею, чим адаптація з використанням зразкового сигналу.

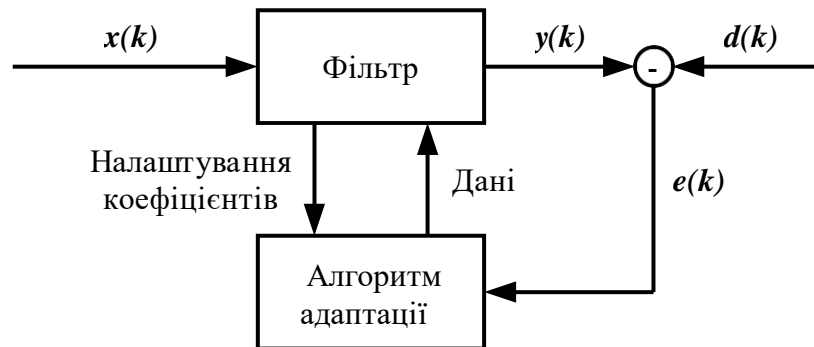


Рисунок 1.1 – Загальна структура адаптивного фільтра

Може здатися, що алгоритми з використанням зразкового сигналу є незрозумілими, оскільки вихідний сигнал повинен бути відомим заздалегідь. Однак, з'являється безліч практичних задач, при вирішенні яких зразковий сигнал виявляється недоступним. У ряді випадків при цьому корисний сигнал є не вихідний сигнал фільтра, а сигнал помилки, тобто це різниця між зразком і виходом.

Завади, з якими приходиться справлятися за допомогою адаптивного фільтра можна розподілити на акустичні і електричні. Причиною першого є порушення умов балансу диференціальних систем, які використовуються у провідних системах для розв'язки двох і чотирьохпроводових ліній. У результаті цього частина енергії сигналу, що приймається, у вигляді

пошкодженої та затриманої копії потрапляє в коло передавача, тим самим, погіршуючи якісні показники системи зв'язку.

Звуковий сигнал-завада виникає в тому випадку, коли звукова хвиля, відхиляючись від прилеглих об'єктів, повертається назад до джерела повідомлень. У системах телекомунікації це відбувається в випадках, коли звукова хвиля попадає в мікрофонне коло за рахунок перешкод або поганої розв'язки приймального або передавального кола. Хоча акустичне відлуння прийме як у провідових, так і в безпроводових системах телекомунікацій, найбільший вплив цього негативного ефекту визначається в таких технічних додатках як радіотелефонія, системи телеконференцій та мобільний зв'язок

Причинами появи цього типу перешкод є:

- пульсації випрямленого струму в джерелах живлення, електричні і магнітні «наводки» від кола перемінного струму;
- хаотичний рух (флуктації) електронів або інших заряджених частинок у провідниках, резисторах, електронних приладах;
- постійні електричні і магнітні поля, створювані трансформаторами, електродвигунами, колами перемінного струму, сучасними кабелів;
- специфічні недоліки носіїв запису, які визначаються як безпосередні коливання різної фізичної природи, що відрізняються складністю часової і спектральної структури;
- сторонні джерела в радіодіапазонах (інші радіостанції, атмосферні електричні розряди, промислова, медична, побутова електроапаратура).

1.3 Алгоритми адаптації

На даний час відома множина алгоритмів для побудови адаптаційних фільтрів. Адаптивні алгоритми базуються на теорії оптимальної вінерівської фільтрації. Опис цієї теорії та основних різновидів адаптивних алгоритмів, що існують сьогодні, можна знайти у спеціалізованих книгах, наприклад, у

[1-4]. Прообразами більшості адаптивних алгоритмів є алгоритм Ньютонa, алгоритм якнайшвидшого спуску і алгоритм за критерієм найменших квадратів (рис. 1.1). До простих відносяться різновиди градієнтних адаптивних алгоритмів за критерієм найменшого квадрата (Least Mean Square, LMS), його нормалізованої версії (Normalized LMS, NLMS).

До складних можна віднести різновиди рекурсивних алгоритмів за критерієм найменших квадратів (Recursive Least Squares, RLS) використання леми про звернення матриці (Matrix Inversion Lemma, MIL) та QR-розкладання.

В свою чергу проміжний клас утворюють алгоритми афінних проєкцій (Affine Projections, AP). Також можна виділити швидкі (тобто обчислювально ефективні) алгоритми AP (Fast AP, FAP) та швидкі RLS-алгоритми, включаючи сходові (що отримали назву через обчислювальні процедури, структурно нагадують сходи або ґрати).

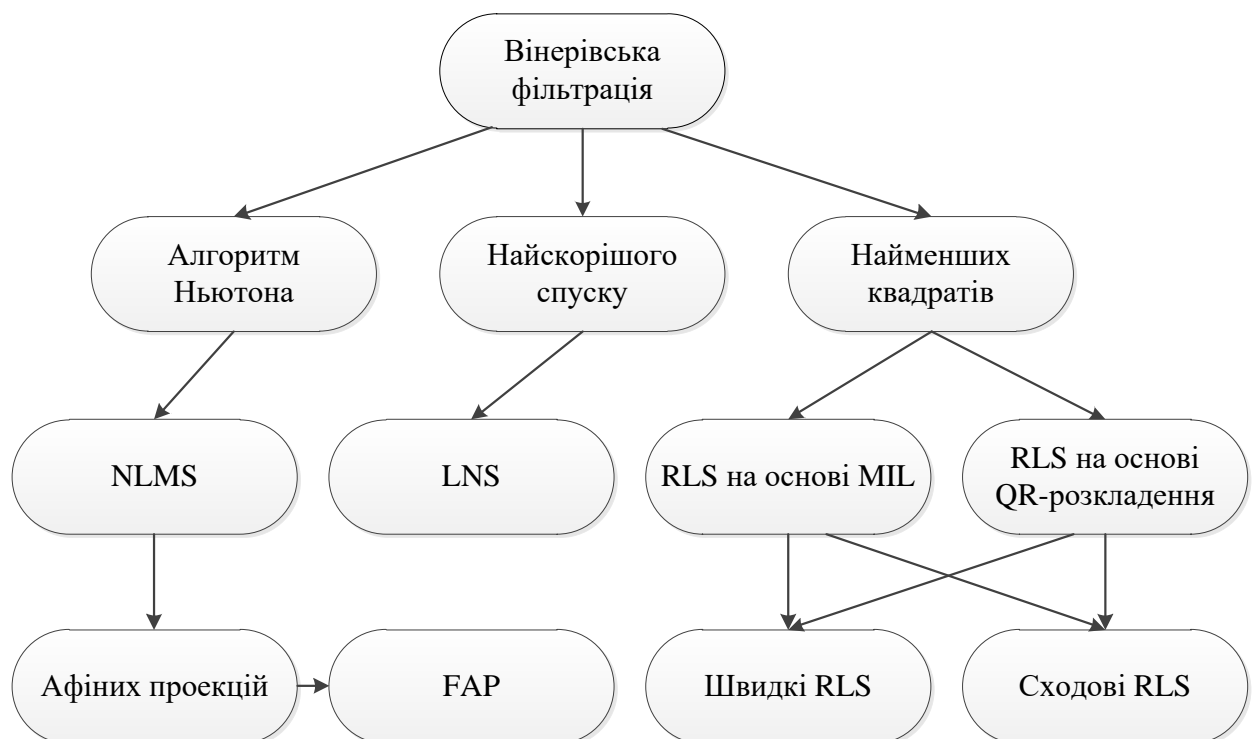


Рисунок 1.2 – Алгоритми адаптивної фільтрації сигналів

Адаптивний алгоритм – це процедура обчислення вагових коефіцієнтів, щоб забезпечити мінімізацію цільової функції адаптивного фільтра, тобто

виконання критерію роботи цього фільтра. Алгоритм характеризується видом використовуваної цільової функції, методом пошуку оптимального рішення та природою сигналів помилок.

В адаптивних алгоритмах використовуються арифметичні операції як над апіорними, і над апостеріорними помилками. Тому обчислення вихідного сигналу адаптивного фільтра, як зазначалося, сприймається як складова адаптивного алгоритму. Адаптивний алгоритм – це зазвичай ітераційна процедура. Його ітерації зазвичай збігаються тривалістю з періодом дискретизації оброблюваних сигналів. При цьому змінні, що обчислюються, оновлюються на кожній ітерації шляхом додавання деяких добавок до попередніх значень або віднімання цих добавок з попередніх значень.

Робота більшості алгоритмів адаптивної фільтрації базується на безумовній або умовній мінімізації певної функції помилок між вхідним та вихідним сигналами адаптивного фільтра. При умовній оптимізації, як правило, застосовуються лінійні обмеження, що накладаються на значення цільової функції, що мінімізується. Адаптивні алгоритми, що впливають з такої мінімізації, називаються лінійно-обмеженими і існують або можуть бути отримані практично для більшості однойменних алгоритмів, що базуються на безумовній мінімізації.

Наявність різноманітних алгоритмів адаптивної фільтрації дає можливість розробникам радіоелектронної апаратури приймати компромісні рішення між обчислювальною складністю (необхідним числом арифметичних операцій на одну ітерацію роботи адаптивного фільтра) та ефективністю (тривалістю перехідного процесу, значенням помилки в стані, що встановився) при розробці фільтра.

Прості градієнтні алгоритми зазвичай використовуються при апаратній реалізації адаптивних фільтрів. Складні RLS- та AP-алгоритми, включаючи їх швидкі версії, переважно орієнтовані на програмну реалізацію. Незважаючи

на складність, RLS-алгоритми найефективніші, а тому також становлять практичний інтерес поряд із простими градієнтними алгоритмами.

1.4 Алгоритм Вінера

Розглянемо адаптивний алгоритм LMS (Least Mean Square), або алгоритм Вінера, який є найпоширенішим. Він заснований на пошуку мінімальної цільової функції за допомогою методу найшвидшого спуску, або іншими словами шляхом знаходження коефіцієнтів фільтра, які відносяться до отримання найменшого середнього квадрата сигналу помилки (різниці між бажаним і фактичним сигналами). Тобто метод стохастичного градієнтного спуску виконується на основі помилки в даний час.

Був запропонований в 1960 професором Бернардом Відроу (Стенфордський університет).

При використанні даного алгоритму оптимізуємий функціонал обчислюються за формулою:

$$J(w) = \overline{e^2(k)} \rightarrow \min \quad (1.1)$$

де $\overline{e^2(k)}$ – середній квадрат помилки, k – крок, який є натуральне число.

$$w(k+1) = w(k) + \mu e(k)u(k), \quad (1.2)$$

де μ – позитивний коефіцієнт, це є розмір кроку, $u(k)$ – вектор-стовпець вмісту лінії затримки фільтра на k -му кроці.

Обчислення вагових коефіцієнтів у адаптивному алгоритмі за критерієм найменшого середнього квадрата LMS виконується як $h_N(k) = h_N(k-1) + \mu x_N(k)\alpha^*(k)$, тобто як

$$\left\{ \begin{array}{l} \text{поточне} \\ \text{значення} \\ \text{вектора} \end{array} \right\} = \left\{ \begin{array}{l} \text{попереднє} \\ \text{значення} \\ \text{вектора } BK \end{array} \right\} + \left\{ \begin{array}{l} \text{крок} \\ \text{збіжності} \end{array} \right\} \left\{ \begin{array}{l} \text{вектор} \\ \text{вхідних} \\ \text{сигналів} \end{array} \right\} \left\{ \begin{array}{l} \text{сигнал} \\ \text{похибки} \end{array} \right\}$$

Перевагою алгоритму LMS є відносна обчислювальна простота. Для обчислювання коефіцієнтів фільтра на кожному кроці потрібно виконати декілька пар операцій множення та додавання. Однак даний алгоритм має «повільну» збіжність та достатньо велику міру розсіяння значень похибки, тому що коефіцієнти фільтра мають відхилення проміж оптимальних значень і це додає помилку до рівня вхідного шуму.

1.5 Алгоритм RLS

Алгоритм RLS (Recursive Least Square) відрізняється від LMS тим, що його цільова функція спрямована не на вирахування середньоквадратичної помилки, а на мінімізацію норми помилки.

Формула оптимальної цільової функції має наступний вигляд:

$$J(w) = \sum_{k=0}^{K-1} |e(k)|^2 \rightarrow \min ,$$

де K – вектор-стовпець коефіцієнтів збільшення.

Формула для рекурсивного оновлення коефіцієнтів виглядає наступним чином:

$$w(k+1) = w(k) + K(k+1)e(k+1).$$

У рекурсивних алгоритмах за критерієм найменших квадратів RLS обчислення BK виконується як $h_N(k) = h_N(k-1) + g_N(k)\alpha^*(k)$, тобто як

$$\left\{ \begin{array}{l} \text{поточне} \\ \text{значення} \\ \text{вектора} \end{array} \right\} = \left\{ \begin{array}{l} \text{попереднє} \\ \text{значення} \\ \text{вектора BK} \end{array} \right\} + \left\{ \begin{array}{l} \text{вектор} \\ \text{коефіцієнтів} \\ \text{Калмана} \end{array} \right\} \left\{ \begin{array}{l} \text{сигнал} \\ \text{похибки} \end{array} \right\}$$

На відміну на LMS алгоритму головною перевагою алгоритму RLS є швидка збіжність. Однак досягається це за рахунок значно більш високої обчислювальної складності. При оптимальній організації вирахувань для оновлення коефіцієнтів фільтра на кожному такті потрібно у рази більше операцій «множення-складання».

1.6 Алгоритм афінних проекцій

Алгоритм афінних проекцій займає проміжне місце по обчислювальній складності і продуктивності серед розглянутих алгоритмів. А саме LMS алгоритм має достатньо повільну збіжність, а той же час RLS, має швидку збіжність, але має велику складність обчислень

Вагові коефіцієнти обчислюються за даним методом за наступною формулою:

$$w(k+1) = w(k) + \mu X_{NL}(k)e(k),$$

$$\text{де } X_{NL} = [x_N(k), x_N(k-1), \dots, x_N(k-L+1)];$$

$$x_N(k) = [x(k), x(k-1), \dots, x(k-N_m+1)];$$

$x_N(k)$ – вектор вхідних сигналів.

Алгоритм афінних проекцій має більш швидкий процес обчислення, тобто його швидкість вище, ніж у LMS алгоритму. Деякі модифікації цього алгоритму мають ефективність близьку до ефективності RLS алгоритмів. Однак, така ефективність досягається за допомогою збільшення вагових коефіцієнтів, що спричиняє за собою збільшення обчислювальної складності.

1.7 Порівняння алгоритмів адаптивної фільтрації

Як можна побачити з проведеного аналізу вибір адаптивного алгоритму обумовлений наступними вимогами: забезпеченням оптимального рішення; тривалістю перехідного процесу, що характеризує швидкість збіжності та слідчі властивості адаптивного фільтра; значенням залишкових помилок у режимі, що характеризує точність знаходження оптимального рішення; обчислювальною складністю алгоритму, що характеризує обсяг ресурсів, необхідних його апаратної чи програмної реалізації. До цих вимог слід додати «алгоритмічну складність», під якою розуміється складність математичного представлення алгоритмів, обумовлена числом та різноманітністю математичних виразів, що є власне алгоритмом.

З точки зору арифметичної та алгоритмічної складності, алгоритми адаптивної обробки сигналів можна умовно поділити кілька груп. Існують обчислювальні прості, тобто з малою кількістю арифметичних операцій за однією ітерацією алгоритму. Це алгоритми з лінійною обчислювальною складністю та малою алгоритмічною складністю, тобто описувані лише кількома простими математичними висловлюваннями. Прикладами таких алгоритмів є алгоритм LMS та інші адаптивні алгоритми на основі стратегії градієнтного пошуку вагових коефіцієнтів.

Алгоритм за найменшим критерієм квадратів характеризується великою обчислювальною складністю та малою алгоритмічною складністю. RLS алгоритми характеризуються середньою обчислювальною складністю та середньою алгоритмічною складністю. Швидкі, тобто обчислювально ефективні або з малим числом арифметичних операцій на одну ітерацію, алгоритми RLS характеризуються низькою обчислювальною складністю, але великою алгоритмічною складністю, оскільки використовують досить велику кількість різнорідних формул.

Раніше такому відношенню до алгоритмів адаптивної фільтрації різної складності сприяла низька продуктивність цифрової елементної основи. Щоб реалізація складних алгоритмів забезпечувала потрібну частоту дискретизації сигналів, були потрібні певні обчислювальні ресурси. Можна висловити припущення про те, що задоволені результати можуть бути отримані при використанні цифрового фільтра використовуючи ПЛІС.

Для реалізації фільтру було обрано RLS алгоритм, який має швидку швидкість сходження на відміну від алгоритмів LMS та алгоритмів афінних проекцій.

Такий підхід дозволить збільшити швидкість обробки сигналу, тобто забезпечити обробку сигналу у режимі реального часу, що дозволить розширити сфери застосування і забезпечити універсальність фільтра на відміну від відомих рішень, реалізованих на базі DSP-процесорів.

2 РОЗРОБКА МОДЕЛІ АДАПТИВНОГО ФІЛЬТРА

2.1 Математичні пакети для побудови моделі фільтра

Для побудови моделі фільтра на сьогоднішній день можна використовувати значну низку математичних пакетів, що реалізують різноманітні чисельні методи і провідні аналітичні матриці. Найбільш відомими сьогодні є наступні пакети прикладних програм та математичні бібліотеки:

- Matlab;
- MathCAD;
- Maple;
- Математика;
- LabVIEW.

Matlab (MATrix LABoratory) був розроблений фірмою The MathWorks. Matlab призначений для обчислення будь-яких математичних розрахунків і моделювання технічних і фізичних систем, а також виконання научних та інженерних моделей та розрахунків.

Інтерактивний інструмент Simulink з використанням Matlab System Blockнадає наступні можливості для моделювання:

- сумісне використання одного і того ж об'єкта System в Matlab і Simulink;
- виділена інтеграція об'єктів системи з Simulink;
- модульний тест алгоритму в Matlab перед його використанням в Simulink;
- налаштування діалогового вікна;
- ефективне моделювання з кращою ініціалізацією;
- обробку станів системи;
- налаштування портів блоку з мітками;

– доступ до двох режимів стимуляції.

MathCAD є інтегрованою системою програмування, орієнтованою на проведення математичних і інженерно-технічних розрахунків.

Maple фірми Waterloo Maple Inc. – програмний пакет, запропонований як система комп'ютерної алгебри. Система Maple призначена для символічних обчислень, яка має ряд засобів для чисельного рішення диференціальних рівнянь і знаходження інтегралів. Володіє розвиненими графічними інструментами.

Математика від Wolfram Research – це система комп'ютерної алгебри. Утримує безліч функцій, як для аналітичних перетворень, так і для чисельних розрахунків. Крім того, програма підтримує роботу з графікою і звуком.

LabVIEW компанії National Instruments – це математичне середовище для візуальної мови програмування. Метою даного пакету є автоматизація використання обчислювального та вимірювального лабораторного обладнання. Інструментарій проектування (синтезу) цифрових фільтрів в LabVIEW (LabVIEW Digital Filter Design Toolkit) складається з декількох розширених засобів для проектування фільтра, аналізу та моделювання цифрових фільтрів з фіксованою і плаваючою комою. Процес проектування є ітераційним.

Всі вище перераховані математичні пакети мають обширний функціонал, великий набір інструментів і бібліотек для роботи в багатьох областях, включаючи цифрову обробку сигналів. Для проектування моделі фільтру з перерахованих пакетів було обрано Matlab тому, що він відрізняється високою швидкістю чисельних виразів, має середу програмування Simulinc, яка є зручною для отримання та аналізу результатів моделювання при побудові фільтру. Також за використовуючи Xilinx System Generator Subsystem можливо моделювати проекти за допомогою блоків як від Simulink, так і Xilinx, і автоматично генерувати інтегрований HDL-код. HDL Coder™ генерує HDL-код із блоків Simulink та використовує Xilinx

System Generator для генерації HDL-коду з блоків Xilinx System Generator Subsystem. Таким чином, як інструмент моделювання адаптивного фільтра шумозаглушення був вибраний пакет прикладних програм Matlab.

2.2 Вимоги до моделі адаптивного фільтра звукових сигналів

Розглянемо вимоги, які необхідно виконати при проектуванні адаптивного фільтра звукових сигналів. Такий адаптивний фільтр повинен пропускати без спотворень корисні складові сигналу і значно послабляти, а в окремих випадках повністю, зменшити будь-які завади вхідного сигналу.

Для цього при побудові адаптивного фільтра необхідно експериментувати з різними технічними вимогами при проектуванні або вибирати методи розробки відповідного цифрового фільтра для застосування. Іноді, можливо, потрібно переглянути технічні вимоги або змінити метод проектування після перевірки фільтра, особливо при розробці фільтрів з фіксованою комою через кінцеву точність коефіцієнтів. При побудові моделі адаптивного фільтра слід врахувати, що фільтр шумокорекції повинен функціонувати в режимі реального часу. Швидкість адаптації фільтра повинна бути достатньою для забезпечення високої якості вихідного сигналу.

Як було розглянуто раніше такі властивості повинен мати RLS алгоритм, який має достатньо високу збіжність.

2.3 RLS фільтр

Для побудови моделі був використаний блок RLS Filter в Matlab. Це блок реалізований на основі алгоритму RLS. Він обчислює відфільтрований сигнал, помилку фільтра та ваги фільтра для заданого вхідного та бажаного сигналу за допомогою алгоритму адаптивного фільтра RLS.

Фільтри RLS мінімізують функцію витрат, або іншими словами функцію зважених помилок методом найменших квадратів, шляхом

відповідного вибору коефіцієнтів фільтра $w(n)$ і оновлення фільтра, коли надходять нові дані. Функція витрат визначається цим рівнянням:

$$C(w_n) = \sum_{i=0}^n \lambda^{n-i} e^2(i),$$

де w_n — коефіцієнти адаптивного фільтра RLS;

$e(i)$ — помилка між бажаним сигналом d і оцінкою бажаного сигналу $dest$ на поточний індекс часу; сигнал $dest$ є виходом фільтра RLS, і тому неявно залежить від поточних коефіцієнтів фільтра;

λ — коефіцієнт забування, який надає експоненціальне меншу вагу старішим зразкам, визначений у діапазоні $0 < \lambda \leq 1$; коли $\lambda = 1$, усі попередні помилки вважаються рівними вагою в загальній помилці; коли λ наближається до нуля, минулі помилки відіграють меншу роль у загальній сумі.

Наприклад, коли $\lambda = 0,1$, алгоритм RLS множить значення помилки з 50 зразків у минулому на коефіцієнт ослаблення $0,150 = 1 \times 10^{-50}$, значно зменшуючи вплив минулих помилок на поточну загальну помилку. У випадках, коли значення помилки може походити від хибної точки або точок вхідних даних, коефіцієнт забування дозволяє алгоритму RLS зменшити значущість старих даних помилки шляхом множення старих даних на коефіцієнт забування.

Блок RLS Filter наведений на рисунку 2.1.

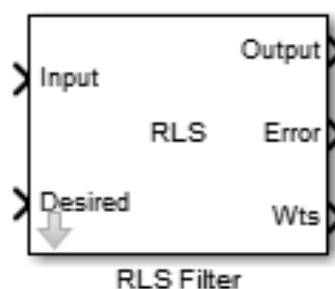


Рисунок 2.1 – Блок RLS Filter

Блок RLS Filter рекурсивно обчислює оцінку найменших квадратів (RLS) ваг фільтра FIR. Блок оцінює ваги фільтра або коефіцієнти, необхідні для перетворення вхідного сигналу в бажаний сигнал. Для цього на вход блоку подається сигнал, який потрібно відфільтрувати, до вхідного порту. Вхідним сигналом може бути скаляр або вектор-стовпець. Вхідний сигнал повинен мати той самий тип даних, складність і розміри, що й вхідний сигнал.

На вихідний порт виводиться відфільтрований вхідний сигнал. Порт Error виводить результат віднімання вихідного сигналу з потрібного сигналу. Відповідний фільтр RLS виражається в матричній формі як:

$$k(n) = \frac{\lambda^{-1} P(n-1) u(n)}{1 + \lambda^{-1} u^H(n) P(n-1) u(n)},$$

$$y(n) = w(n-1) u(n),$$

$$e(n) = d(n) - y(n),$$

$$w(n) = w(n-1) + k^H(n) e(n),$$

$$P(n) = \lambda^{-1} P(n-1) - \lambda^{-1} k(n) u^H(n) P(n-1),$$

де λ^{-1} – позначає зворотну величину експоненціального вагового коефіцієнта. Змінні наведені у таблиці 2.1.

Таблиця 2.1 – Змінні алгоритму

Змінна	Опис змінної
n	Індекс поточного часу
$u(n)$	Вектор буферизованих вхідних зразків на кроці n
$P(n)$	Обернена коваріаційна матриця на кроці n

$k(n)$	Вектор посилення на кроці n
$w(n)$	Вектор оцінок фільтр-відведення на кроці n
$y(n)$	Відфільтрований вихід на кроці n
$e(n)$	Помилка оцінки на кроці n
$d(n)$	Бажана відповідь на кроці n
λ	коефіцієнт забування, який дає експоненціально меншу вагу більш старим виборам помилок

Реалізація алгоритму в блоці оптимізована за рахунок використання симетрії оберненої коваріаційної матриці $P(n)$. Це зменшує загальну кількість обчислень у два рази.

Параметр коефіцієнта забування (від 0 до 1) відповідає λ у рівняннях. Він визначає, як швидко фільтр «забуває» інформацію про попередні зразки. Налаштування $\lambda=1$ визначає нескінченну пам'ять. Як правило,

$$1 - \frac{1}{2L} < \lambda < 1,$$

де L – довжина фільтра.

Коефіцієнт забування вказується за допомогою вхідного порту, для цього необхідно ввести значення в параметр «Коефіцієнт забування» (від 0 до 1) у діалоговому вікні Параметри блоку: фільтр RLS. Блок використовує скалярне значення для створення вектора ваг фільтра. Довжина цього вектора дорівнює довжині фільтра, а всі його значення дорівнюють скалярному значенню. Початкове значення $P(n)$ дорівнює $\frac{1}{\sigma^2} I$, де необхідно вказати σ^2 у параметрі оцінки початкової вхідної дисперсії.

Коли вхідні дані цього порту відмінні від нуля, блок постійно оновлює ваги фільтра. Коли на вході цього порту дорівнює нулю, ваги фільтра

залишаються на поточних значеннях. Для кожної ітерації блок виводить поточні оновлені ваги фільтра з цього порту.

Особливістю RLS фільтра є те, що вихідне значення кожного відліку формується не тільки маніпуляціями з поточними вхідними значеннями, але і з значеннями вихідних відліків, порахованих в попередніх розрахунках підлітків.

Це дозволяє уникати великих і невиправданих обчислювальних витрат на вирахування важливих коефіцієнтів фільтра. При використанні адаптивного алгоритму RLS необхідно на кожному часовому відліку виконати наступні кроки. При надходженні нових вхідних даних $x(k)$ виконується фільтрація сигналу з використанням поточних коефіцієнтів фільтра та обчислення величини похибки зразкового сигналу.

Початкове значення вектора вагових коефіцієнтів зазвичай приймається нульовим, а в якості вихідної оцінки матриці приймається матриця наступного виду:

Зазвичай діагональ заповнюється значеннями, наприклад $100/\sigma_n^2$.

$$P(-1) = \begin{bmatrix} \infty & 0 & 0 & \dots & 0 \\ 0 & \infty & 0 & \dots & 0 \\ 0 & 0 & \infty & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \infty \end{bmatrix}$$

2.4 Побудова моделі адаптивного фільтра

Для побудови моделі адаптивного фільтра необхідно вибрати структуру фільтра. Всі способи застосування адаптивних фільтрів зводяться до вирішення задачі ідентифікації, тобто визначення характеристик деякої системи. Ідентифікація – це процес, в результаті якого забезпечується збіг

або подібність (за заданим критерієм) її вхідного та вихідного сигналів з вхідним та вихідним сигналами відомої системи. Існуючі підходи до вирішення задачі ідентифікації відгуку невідомих параметрів для визначення характеристик системи можна розділити на пряму ідентифікацію та зворотню.

У разі прямої ідентифікації вхідний сигнал є загальним для дослідної системи та адаптивного фільтра, а вихідний сигнал системи є для адаптивного фільтра зразковим. У відношенні адаптації часові і частотні характеристики фільтра будуть спрямовуватися до відповідних характеристик системи, що досліджується.

Тобто пряма ідентифікація – це збіг вхідних сигналів невідомої та відомої систем за подібністю (за заданим критерієм) їх вихідних сигналів. Схема прямої ідентифікації наведена на рисунку 2.2.

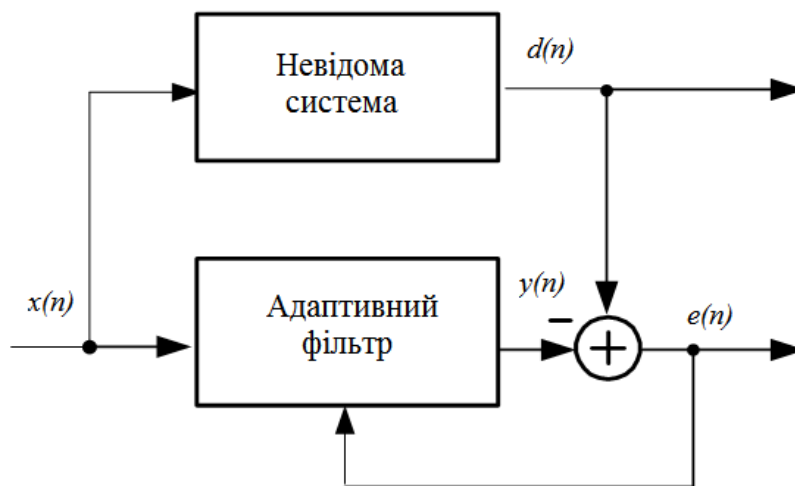


Рисунок 2.2 – Схема прямої ідентифікації

При зворотній ідентифікації адаптивний фільтр включається послідовно з системою, що досліджується. Вихідний сигнал системи подається на вхід адаптивного фільтра, а вхідний сигнал є зразком для адаптивного фільтра. Таким чином, фільтр прагне компенсувати вплив системи та відновити вихідний сигнал, усунувши внесені системою пошкодження.

Тобто зворотня ідентифікація – збіг вихідного сигналу невідомої системи з вхідним сигналом відомої системи за подібністю (за заданим критерієм) вхідного сигналу невідомої системи з вихідним сигналом відомої системи. Схема зворотної ідентифікації наведена на рисунку 2.3, де позначено z^{-D} – елемент затримки на D відліків, $d(n-D)$ – затриманий зразковий сигнал.

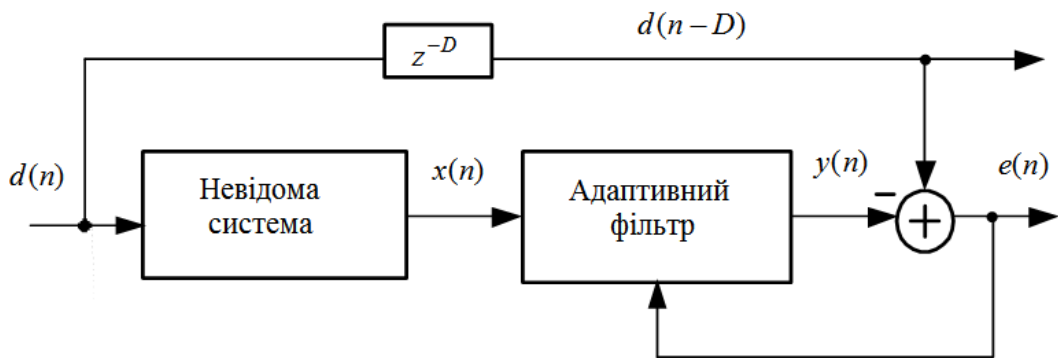


Рисунок 2.3 – Схема зворотної ідентифікації

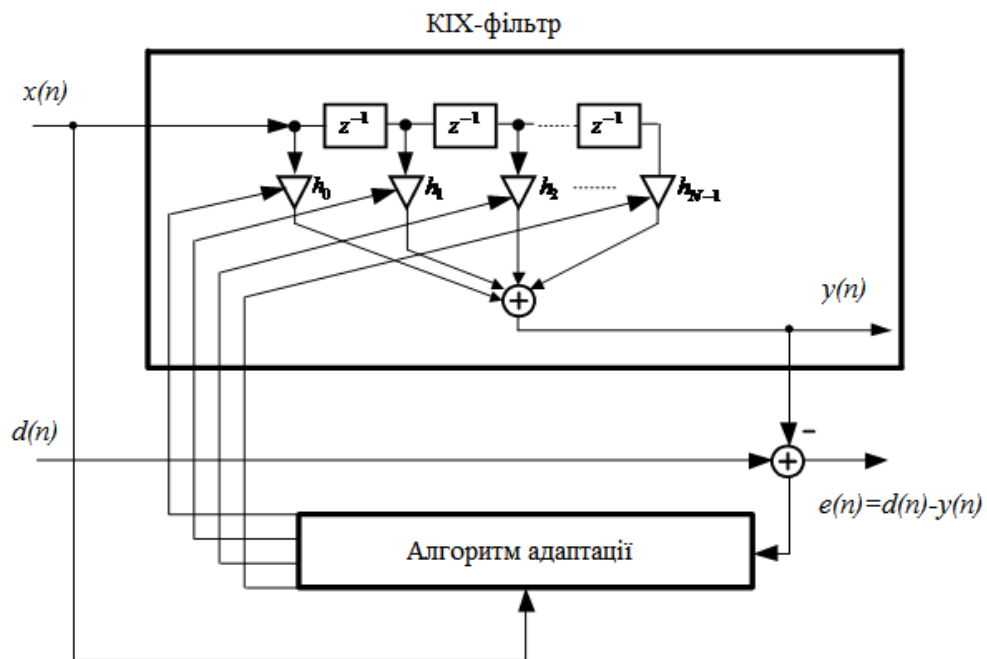


Рисунок 2.4 – Структурна схема зворотної ідентифікації за допомогою КІХ-фільтра

На підставі розглянутого вище була розроблена модель адаптивного фільтру (рис. 2.5). Модель була розроблена за допомогою бібліотеки візуального програмування Simulink з використанням Matlab System Block.

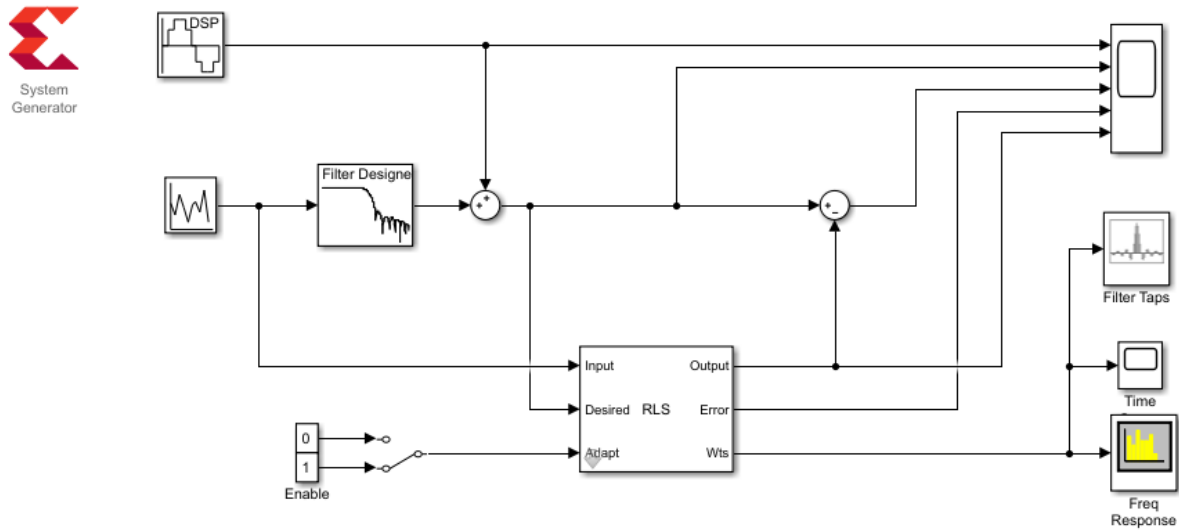


Рисунок 2.5 – Модель фільтра

Для імітації вхідних сигналів були використані блоки генерації сигналів та блок генерації шуму або блок затримки. Для симуляції генератора шуму використовувався Гаусов шум. Потім сигнал плюс шум подавався на вхід адаптивного фільтру. Формування вхідного сигналу з завадами наведено на рисунку 2.6.

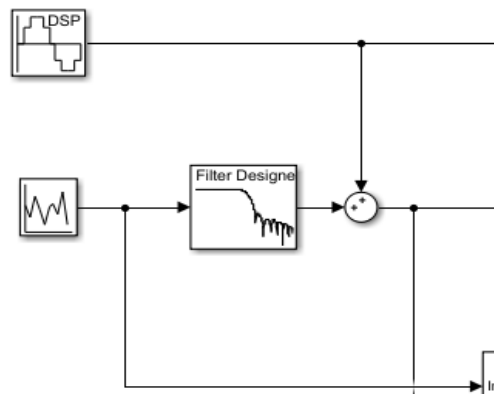


Рисунок 2.6 – Формування вхідного сигналу змішаного з шумом

Також для формування вхідного сигналу з відлунням використовувалась наступна модель сигналу, яка наведена на рисунку 2.7. Сигнал формується наступним чином: початковий сигнал поступає на вхід лінії затримки, а потім подається на вхід цифрового фільтру. Після цього початковий сигнал складається з сформованим у суматорі.

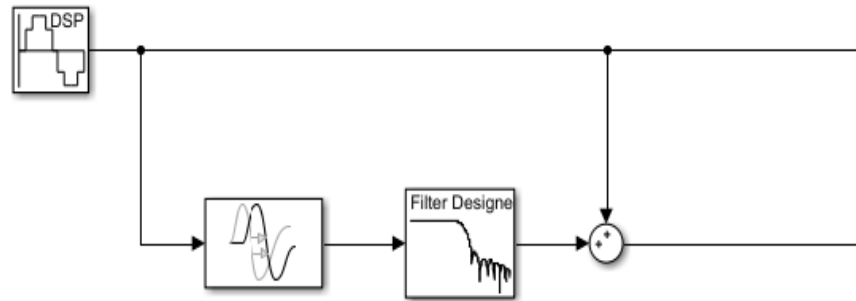


Рисунок 2.7 – Формування вхідного сигналу з відлунням

Модель адаптивного фільтру працює наступним чином. На вхід фільтру подаються опорний сигнал на вхідному порту та бажаний сигнал на бажаному порту. У міру збіжності адаптивної фільтрації відфільтрований шум повинен бути повністю віднятий із сигналу сигнал + шум, а сигнал помилки повинен містити тільки вихідний сигнал. Блок обчислює ваги фільтра на основі алгоритму рекурсивних найменших квадратів (RLS) для адаптивної фільтрації вхідного сигналу. Вхід Adapt port необхідний для того, щоб запускати механізм постійного оновлювання ваг фільтра, коли вхідні дані цього порту відмінні від нуля. Якщо вхід до цього порту дорівнює нулю, ваги фільтра залишаються постійними. Якщо у моделі використовується порт Reset та він увімкнено, то відбувається подія reset, блок скидає ваги фільтра до їхніх початкових значень.

Параметри ФНЧ фільтру, який використовується в запропонованій моделі наведено на рисунку 2.8.

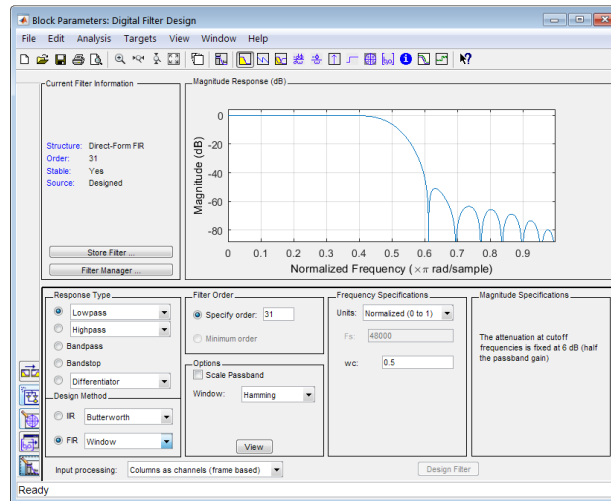


Рисунок 2.8 – Параметри фільтру

Параметри адаптивного фільтру задаються за допомогою діалогового вікна, що наведено на рисунку 2.9.

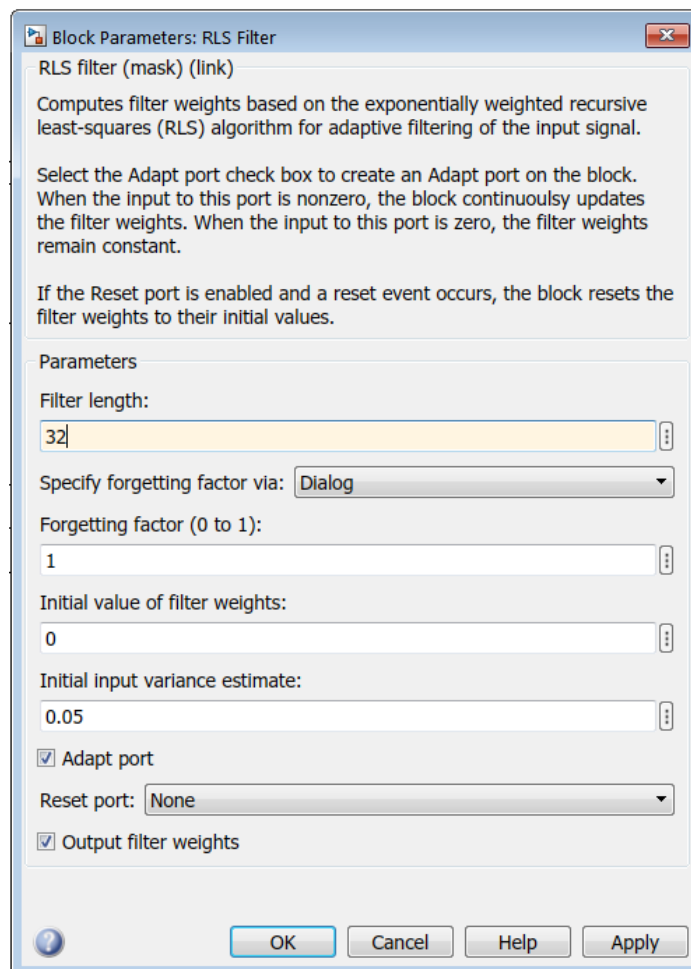


Рисунок 2.9 – Діалогове вікно блоку адаптивного фільтру

У діалоговому вікні можна налаштувати наступні параметри фільтра: порядок фільтра (кількість вагових коефіцієнтів); коефіцієнт забування (експонентне зменшує вагу минулих значень сигналу помилки); початковий вектор коефіцієнтів (дозволяє ініціалізувати вагові коефіцієнти); зворотню корпоративну матрицю (визначає величини значень діагоналі зворотної корпоративної матриці); дозвіл на зміну коефіцієнтів.

Прапор для дозволу на зміну коефіцієнтів дозволяє включити (відключити) адаптацію, тобто зафіксувати коефіцієнти фільтра. Це можна використовувати в тому випадку, коли фільтр адаптувався під вхідний сигнал.

2.5 Результати моделювання

Для отримання результатів роботи запропонованої моделі при моделюванні адаптивного цифрового фільтра на вхід подавалися сигнали різної складності, а саме сигнали математичних функцій синус, косинус та інше різної частоти, з яких формувалися сигнали з завадами або сигнали з відлунням.

Для оцінки результатів моделювання в моделі використовувалися осцилограф, блоки побудови амплітудно-частотної характеристики, коефіцієнтів фільтру у реальному масштабі часу.

На рисунку 2.10 можна побачити вихідний сигнал, сигнал з додаванням шуму, от фільтрований сигнал за допомогою адаптивного фільтру та сигнал похибки, який компенсується за допомогою адаптивного фільтру.

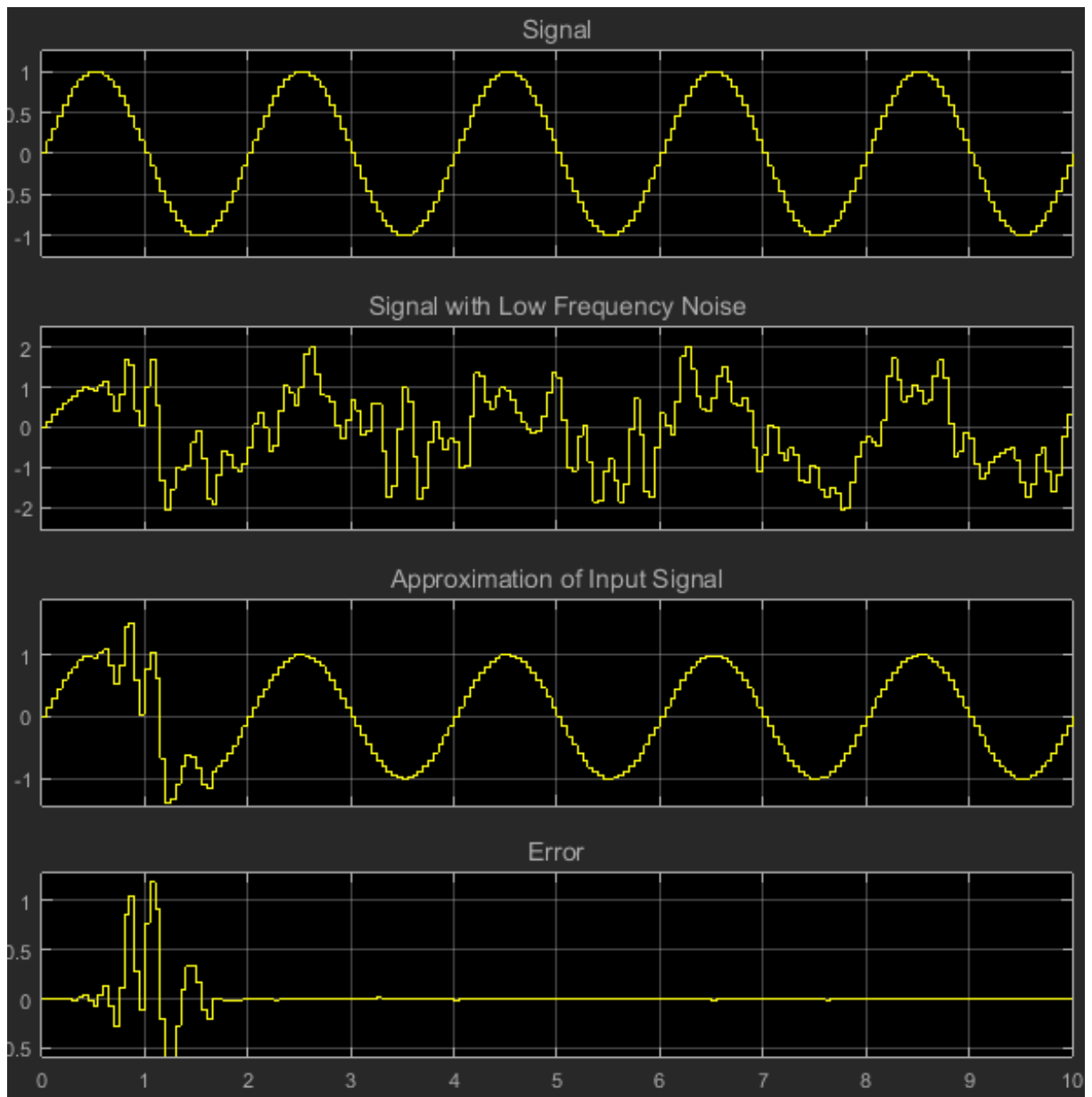


Рисунок 2.10 – Графіки моделювання сигналу

На рисунку 2.11 наведені результати моделювання фільтрування різноманітних сигналів з завадами.

З графіків видно, що в початковий момент часу сигнали помилки та відфільтрований сигнал ідентичні, однак у наступний час адаптації помилка практично повністю зникає.

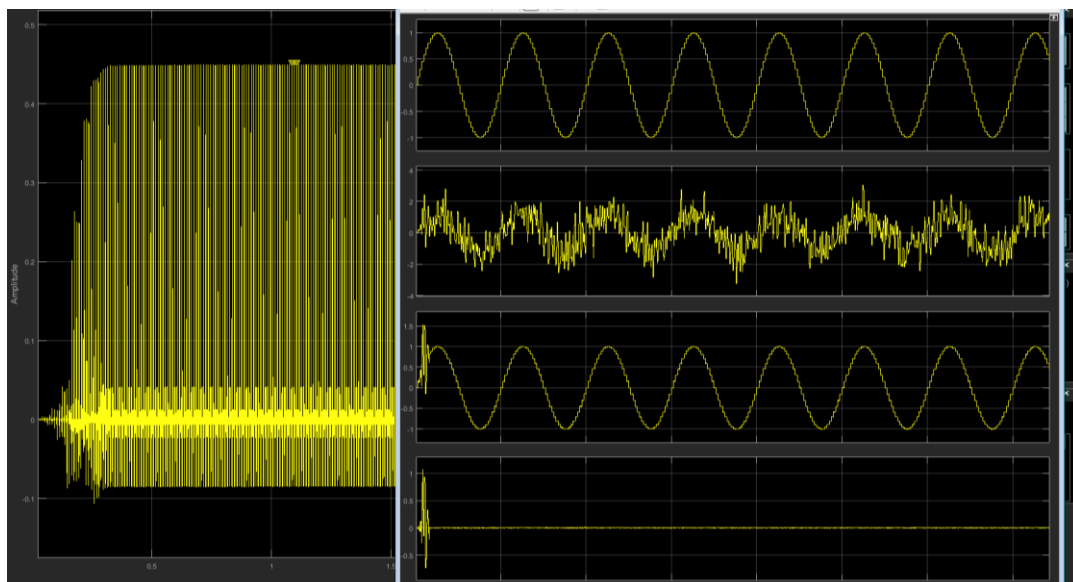
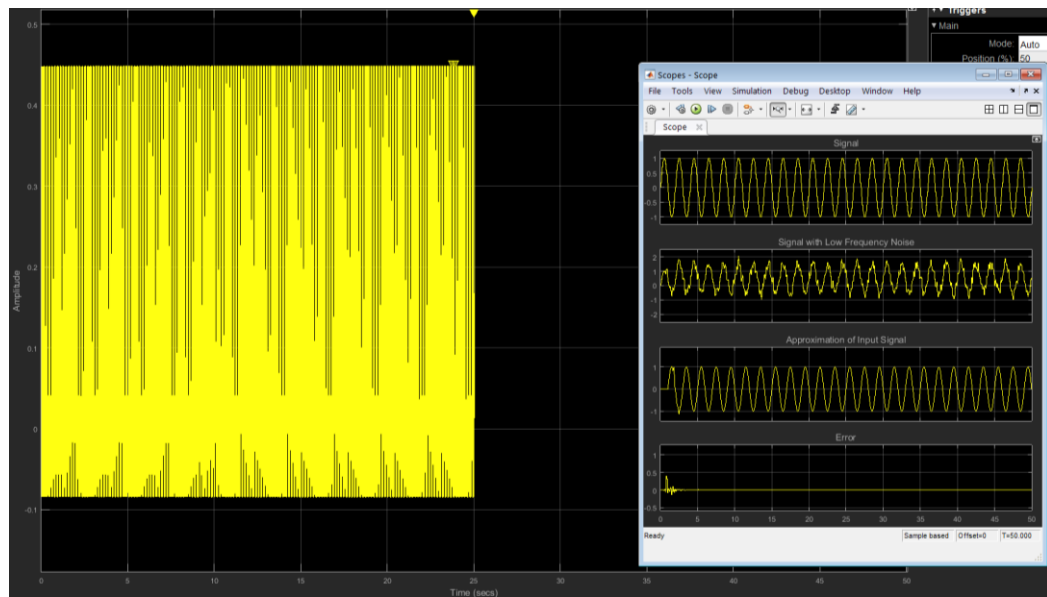
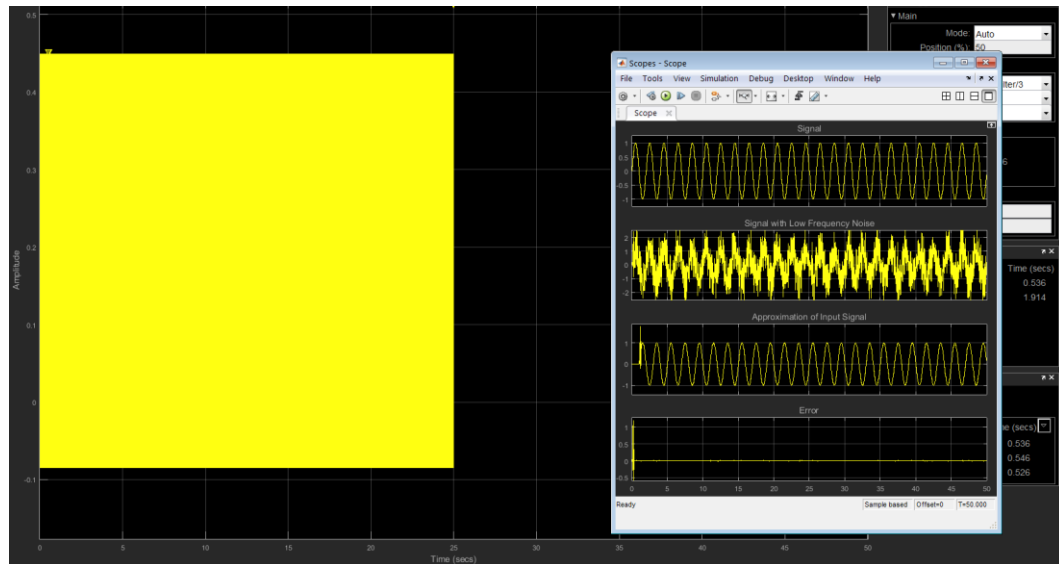


Рисунок 2.11 – Результати моделювання

На рисунку 2.12 наведені результати фільтрування сигналу з відлунням.

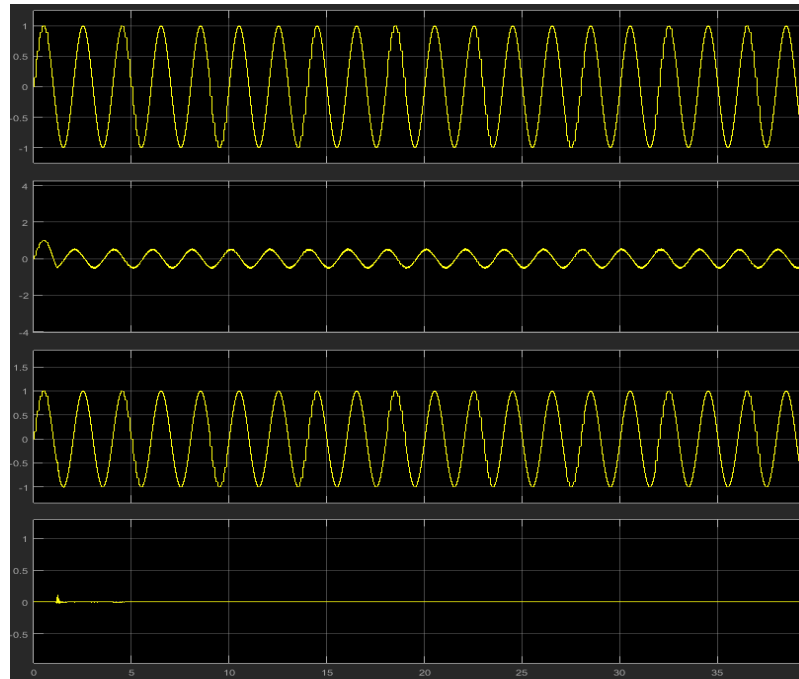


Рисунок 2.12 – Результати моделювання

На рисунках 2.13-2.14 представлені графіки вагових коефіцієнтів різні моменти часу і амплітудно-частотна характеристика фільтру. За графіками видно, що після основної частини адаптації значення вагових коефіцієнтів змінюється у малих діапазонах.

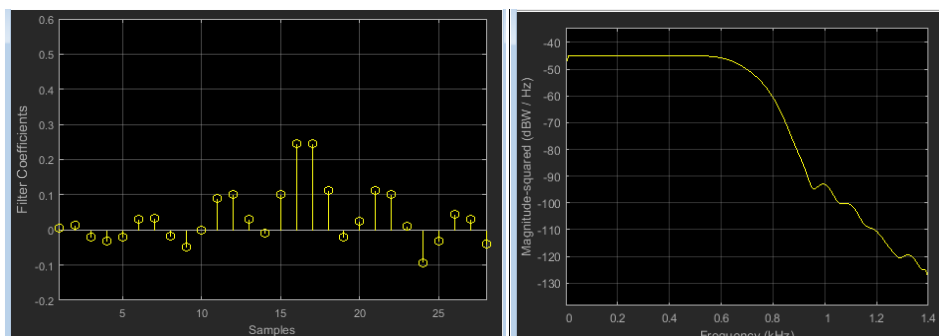


Рисунок 2.13 – Результати моделювання в початковий момент часу

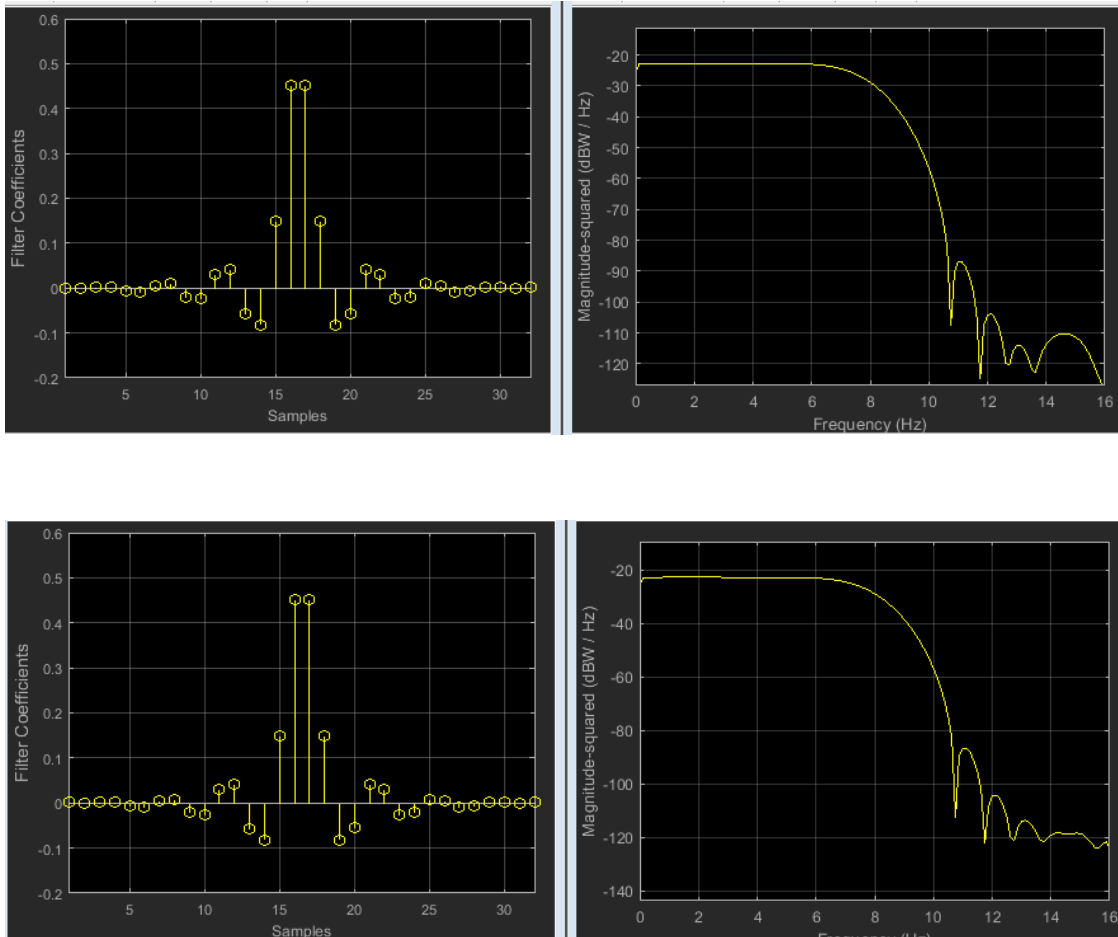


Рисунок 2.14 – Графіки вагових коефіцієнтів і амплітудно-частотних характеристик адаптивного фільтра в різні моменти часу

В результаті моделювання було розроблено модель на основі адаптивного фільтра. Визначені вимоги адаптивного фільтра. Було обрано розрахунковий алгоритм адаптації за методом RLS.

Також було обрано середу програмного моделювання. Розроблена програмна модель адаптивного фільтра. Проведено моделювання фільтра з використанням сигналів різної складності.

Моделювання показало, що при використанні адаптивного фільтра основна компенсація завади або відлуння відбувається протягом 150 відліків. При дальшій адаптації вагові коефіцієнти фільтра природної динаміки не мають. Амплітудно-частотна характеристика також практично не змінюється. На частотах звукового діапазону фільтр справляється зі своєю основною задачею, однак, на частоту вище 40 кГц фільтр не спроможний виконувати

фільтрування. Це пов'язано з тим, що фільтр при високій частоті вхідного сигналу не успіває адаптуватися, так як обчислення вагових коефіцієнтів вимагає деякого часу.

3 РОЗРОБКА АПАРАТНОЇ РЕАЛІЗАЦІЇ АДАПТИВНОГО ФІЛЬТРА

3.1 Вибір апаратної платформи

Для реалізації фільтру необхідно обрати апаратну платформу, яка забезпечить режим роботи фільтру у реальному часі. Дискретний фільтр можна реалізувати:

- на базі замовної ІС;
- на базі мікроконтролера;
- на базі DSP процесора;
- на базі ПЛІС.

Варіант реалізації на базі замовної ІС є дуже важкозатратним, тривалим і дорогоцінним процесом.

Варіант розробки адаптивного фільтру на базі мікроконтролеру є реалізуємим, але він обмежений технічними можливостями чіпа, такими як його швидкодією та інше.

Розробка на базі DSP процесорів не дозволить досягти бажаних результатів, оскільки деякі функції доведеться реалізовувати програмно, що також накладає певні обмеження, а саме швидкодію, розмірність шини даних та інше.

Варіант розробки адаптивного фільтру на базі ПЛІС набагато продуктивніший. Основна перевага ПЛІС полягає у можливості паралельного виконання кількох сотень операцій множення з накопиченням, тоді як сигнальні процесори можуть виконати у кращому разі 2-4 таких операції одночасно.

Ще однією перевагою, пов'язаною з можливістю програмування зв'язків між апаратними ресурсами та різноманітністю цих ресурсів, є різноманіття способів реалізації фільтру. Реалізувати адаптивний фільтр на

ПЛІС можна за допомогою її програмованих ресурсів. Виграшним варіантом побудови адаптивного фільтру у таких випадках може виявитися використання методу розподілених обчислень, що дозволить не лише обмежитися дешевшим кристалом, а й знизити латентність фільтра.

Також перевагою використання ПЛІС є спеціалізовані апаратні модулі цифрової обробки сигналів, що вбудовуються у сучасні програмовані матриці. Ефективна реалізація складних алгоритмів на ПЛІС вимагає спеціальних архітектур і модифікації алгоритмів. Головна перевага апаратних модулів це відомі заздалегідь гарантовані часові характеристики, причому значно кращі, ніж у пристроїв, що збираються на програмованих ресурсах. Відмінами таких модулів є наявність помножувача, що являє собою програмований суматор, що виконує додавання, віднімання або акумулювання змінних, а також набір регістрів та мультиплексорів для керування вхідними, вихідними та внутрішніми сигналами.

При програмному моделюванні адаптивних фільтрів обчислення зазвичай виконуються в арифметиці з плаваючою крапкою. Однак ПЛІС не має вбудованих модулів по обробці чисел з плаваючою крапкою. Реалізація математичних операцій з плаваючою крапкою вимагає великих апаратних витрат, що не необхідним при проектуванні систем реального часу.

Хоча обчислення арифметики з плаваючою крапкою в ПЛІС можливо, це зазвичай досягається за допомогою спеціальної моделі з плаваючою крапкою. Ці прилади дуже потребують багато ресурсів з точки зору логічних венти́лей. Тому у проекті може використовуватися тільки лише невелика кількість моделей з плаваючою крапкою. Це не дає повної переваги розподілення, яке можливо в ПЛІС, і, відповідно, не є найбільш ефективним методом. Тому всі розрахунки повинні бути проведені в арифметиці з фіксованою крапкою, але це може привести до деяких помилок.

Помилки, які можуть виникати:

– помилки квантування АЦП – результат подання вхідних даних обмеженим числом бітів;

- помилки квантування вагових коефіцієнтів – викликані наданням вагових коефіцієнтів кінцевим числом бітів;
- помилки переповнення – викликані складенням (примноженням) двох великих чисел одного знака, що призводить до переповнення, який перевищує допустиму довжину слова;
- помилки округлення - виникає, коли результат множення округлення (або урізання) до найближчого дискретного значення або допустимої довжини слова.

3.2 Реалізація фільтру

На основі розробленої моделі було згенеровано VHDL опис адаптивного фільтру. Для генерації коду розробленої моделі використовувався Simulink HDL Coder. Це високорівневий інструмент проектування, який генерує HDL код з моделей Simulink. Simulink Coder HDL також надає інтерфейси для комбінування написаного вручну HDL коди, блоки симуляції HDL і блоки оперативної пам'яті в своєму середовищі.

Алгоритм RLS був розроблений, синтезований та змодельований за допомогою програмного забезпечення Altera Quartus II, тому що це середовище забезпечує повне середовище проектування для проектування системи на програмованому чіпі (рис.3.1).

У якості впровадження та тестування використовується було обрано ПЛІС FPGA Cyclone III (EP3C5E144C8N).

EP3C5E144C8N є ПЛІС Cyclone III, яка складається з 5136 вертикально встановлених логічних елементів (LEs), має 414КБ вбудованої пам'яті та 23 вбудованих помножувачів 18x18.

Архітектура включає високоефективні мережі синхронізації, що забезпечують зв'язок між логічними структурами для тактових сигналів і сигналів даних. Структури логіки та маршрутизації оточені портами

вводу/виведення та ФАПЧ. Сімейство ПЛІС Cyclone III малої потужності є третім поколінням серії ПЛІС Altera Cyclone.

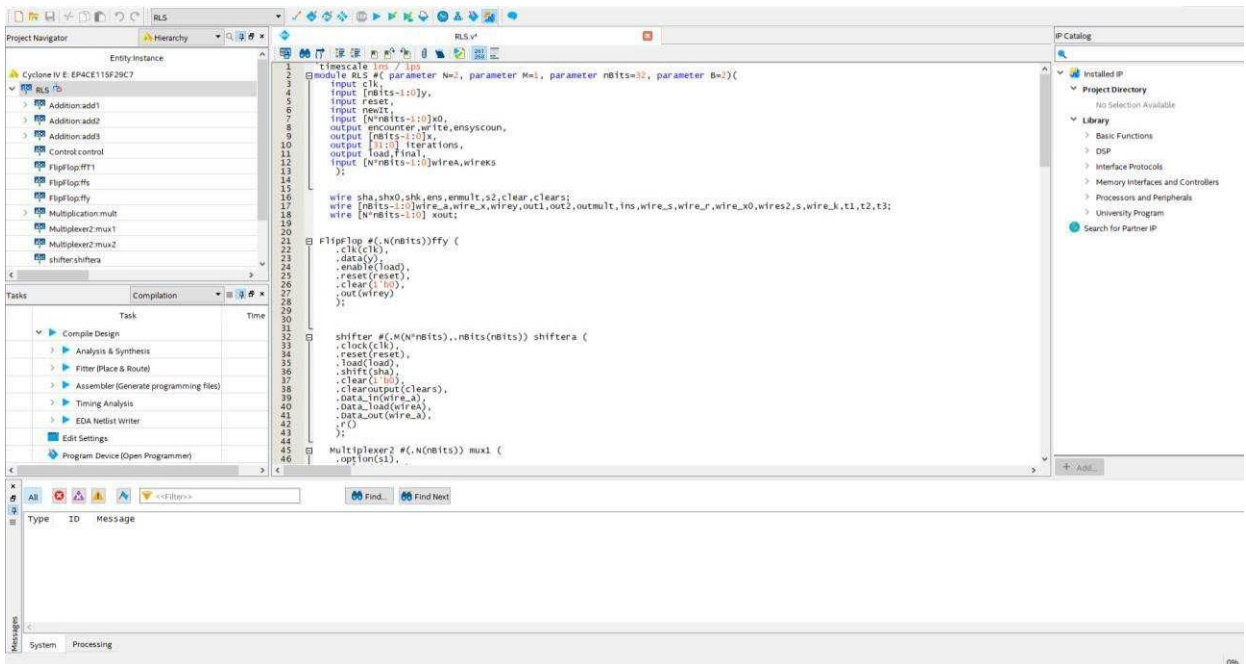


Рисунок 3.1 – Інтерфейс Altera Quartus

Завдяки безпрецедентному поєднанню низької потужності та високої функціональності, сімейство ПЛІС Cyclone III збільшує кількість високопродуктивних та економічних пристроїв, які можуть отримати вигоду з ПЛІС. Версія Cyclone III LS розширює сімейство за допомогою вищої щільності, більшої пам'яті, зменшених корпусів та функцій безпеки для захисту вашої інтелектуальної власності (IP).

Затримка генерації верхньої трикутної матриці становить 777 циклів а зворотне заміщення має затримку 156 циклів. Ці затримки можна зменшити, додавши більше граничних або внутрішніх вузлів до дизайну або зменшення вимог до довжини слова. Дизайн легко розширюється до інших розмірів матриці шляхом зміни керування. Існує компроміс між кількістю вентилів, тобто площею дизайну, і пропускною здатністю. Для великих матриць, якщо пропускна здатність менша ніж необхідно збільшити кількість комірок і використовувати напівпаралельний підхід замість поточної складеної моделі.

Було сгенеровано VHDL-опис адаптивного фільтру для Cyclone III. Також було сгенеровано тестбенч для здійснення моделювання та перевірки результатів синтезу. Потім виконано синтез, імплементацію, та тестування запропонованого проекту.

В лістингу 3.1 наведено блок Entity з вхідними сигналами. Вхідними сигналами для фільтру є тактовий сигнал Clk, сигнал початку фільтрації clk_enable, сигнал скидання reset. Також сигнал вхідних даних, що фільтруються data_in. Вихідними даними є середній квадрат помилки. Отфільтровані дані отримуються на виході data_out.

Лістинг 3.1

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
USE ieee.math_real.all;

ENTITY filter IS
  PORT(
    clk      : in   STD_LOGIC;
    reset    : in   STD_LOGIC;
    ce       : in   STD_LOGIC;
    data_in_rdy : out STD_LOGIC;
    data_in   : in   STD_LOGIC_VECTOR (WIDTH_IN-1
downto 0);
    data_in_vld : in   STD_LOGIC;
    cs_valid   : out  STD_LOGIC;
    cs         : out  STD_LOGIC_VECTOR (WIDTH_COS-1
downto 0);
    sn_vld     : out  STD_LOGIC;
    sn         : out  STD_LOGIC_VECTOR (WIDTH_SIN-1
downto 0);
    data_out_vld : out STD_LOGIC;
    data_out    : out STD_LOGIC_VECTOR (WIDTH_OUT-1
downto 0));
END filter;
```

Блок фільтрації наведено у лістингу 3.2.

```
begin

process(clk, reset)
```

```

begin
    if reset = '1' then
        data_in_buff <= (others => '0');
        data_out      <= (others => '0');
    elsif rising_edge(clk) then
        if ce = '1' then
            if data_in_vld = '1' then
                data_in_buffer <= data_in;
            end if;
            data_out <= data_out_buff;
        end if;
    end if;
end process;

process(clk, reset)
begin
    if reset = '1' then
        data_in_sq <= (others => '0');
        r_sq       <= (others => '0');
        r_sqr_buff <= (others => '0');
    elsif rising_edge(clk) then
        if ce = '1' then
            data_in_sq <= data_in_buffer * data_in_buff;
            r_sq       <= data_in_sq +
lamda * r_sq_buff;
            r_sq_buff <= trounding(r_sq, , );
        end if;
    end if;
end process;

--
process(clk, reset)
begin
    if reset = '1' then

    elsif rising_edge(clk) then
        if ce = '1' then

        end if;
    end if;
end process;

end struct;

```

Для тестування було розроблено TestBench, який частково наведено в листингу 3.3.

```
-- TestBench Template
```

```

LIBRARY std;

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
USE ieee.math_real.all;

library work;

ENTITY testbench IS
    Generic (
        IN_VEC      : string := "t_vect.txt";
        OUT_VEC     : string := "r_vect.txt");
END testbench;

ARCHITECTURE fir OF testbench IS

    begin
        file_open(out_file, filename, WRITE_MODE);

BEGIN

-- Clock
    clk_process: process
    begin
        clk <= '0';
        wait for clk_period/2;
        clk <= '1';
        wait for clk_period/2;
    end process;

    tb_read : PROCESS
    BEGIN
        data_in <= (others => '0');
        reset <= '1';
        ce <= '0';
        wait for 100 ns;
        data_in_ram <= (others => (others => '0'));
        reset <= '0';
        ce <= '1';

        data_in_ram <= fill_ram(INVEC);

        for i in 0 to VECTOR_LENGTH-1 loop
            wait until rising_edge(clk);
            data_in <= data_in_ram(i);
        end loop;
        wait for clk_period;
        ce <= '0';

```

```

    wait;
END PROCESS tb_read;

tb_write : PROCESS
BEGIN

    for j in 0 to VECTOR_LENGTH-1 loop
        wait until data_out_vld = '1' and rising_edge(clk);
        data_out_ram(j) <= data_out;
    end loop;

    export_ram(OUTVEC, data_out_ram);

end process tb_write;

END;
```

Результати моделювання наведені на часовій діаграмі (рис. 3.3)

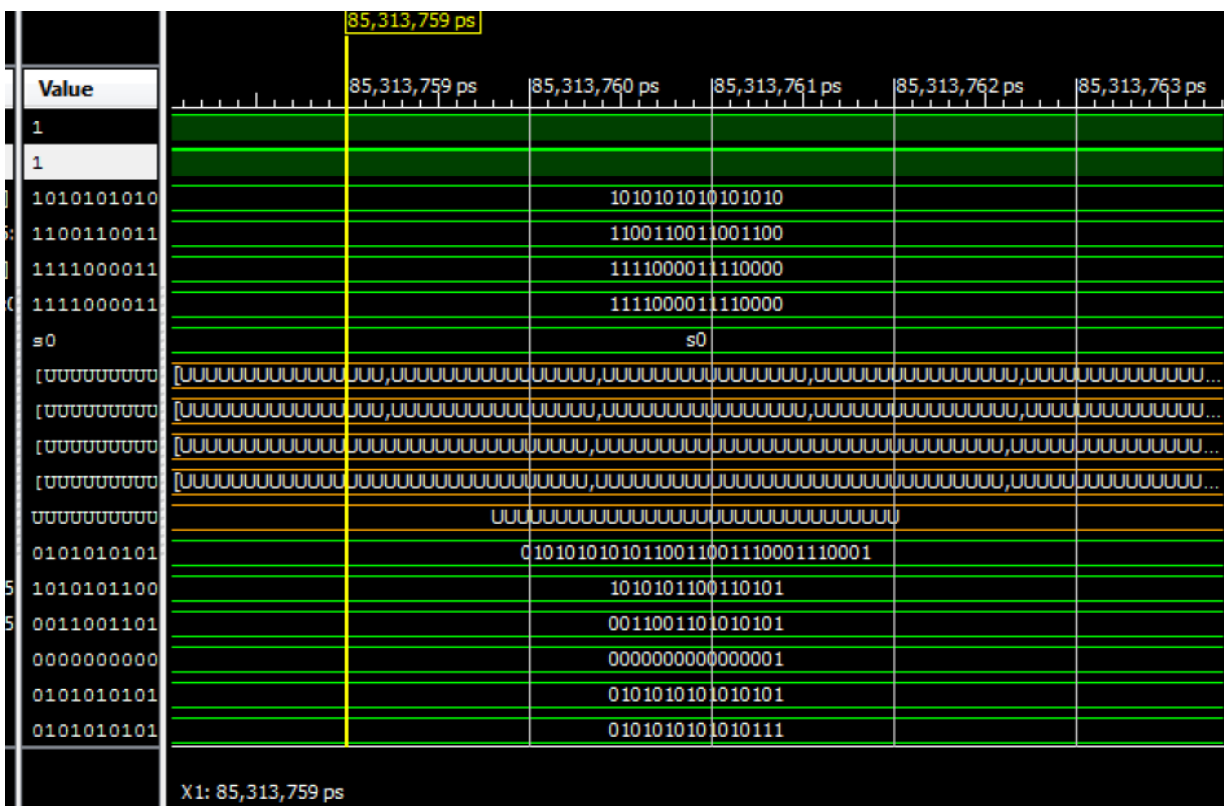


Рисунок 3.2 – Результати тестування

Результати тестування наведені на рис. 3.2 показали, що результати роботи фільтра є правильними. Тобто фільтр працює правильно.

На рисунках 3.3 – 3.4 наведено синтезована схема адаптивного цифрового фільтра та розведення топологічних елементів в ПЛІС .

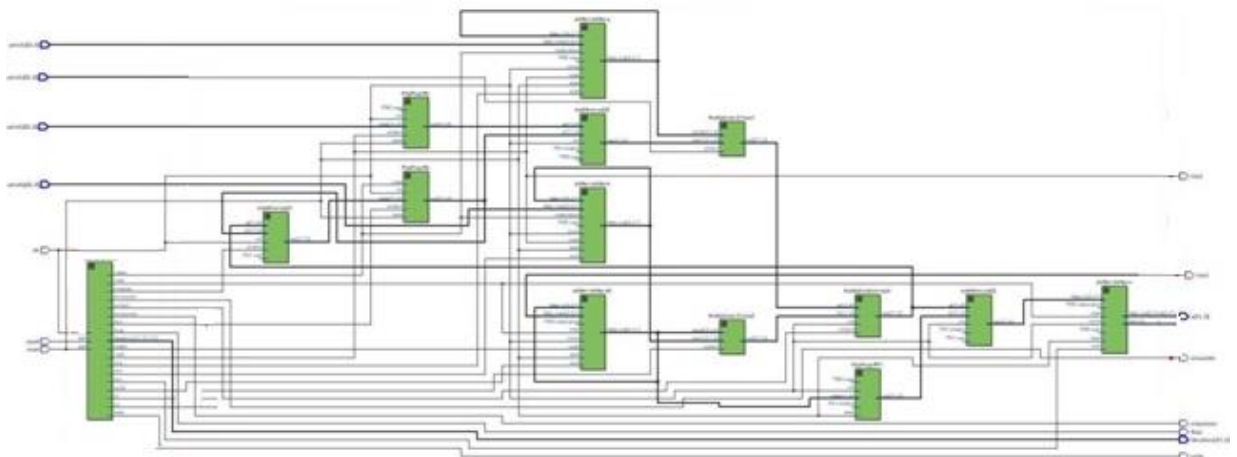


Рисунок 3.3 – Схема адаптивного цифрового фільтра



Рисунок 3.4 – Розведення топологічних елементів в ПЛІС

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було виконане проектування адаптивного цифрового фільтру з застосуванням ПЛІС.

Для цього було проведено аналіз існуючих адаптивних алгоритмів, розглянуті їх недоліки та переваги. Розглянута загальна структура адаптивного фільтру та сформульовані вимоги до вибору адаптивного алгоритму, який повинен бути реалізований апаратно.

Проведено порівняльний аналіз адаптивних алгоритмів. Для реалізації фільтру було обрано RLS алгоритм, тому що він має швидку швидкість сходимості на відміну від алгоритмів інших розглянутих адаптивних алгоритмів.

Були розглянуті існуючі на даний момент математичні пакети для розробки та аналізу моделі фільтра. На основі проведеного аналізу був вибраний пакет Matlab/Simulink.

Була розроблена математична модель адаптивного фільтру у пакеті Matlab/Simulink та проведене моделювання. Для реалізації моделювання використовувалися вхідні сигнали різної складності, а саме сигнали різної частоти математичних функцій синус, косинус та інше. Результат моделювання показав, що основна компенсація завади або відлуння відбувається протягом 150 відлітків.

Були проаналізовані можливості апаратної реалізації цифрових фільтрів в DSP, мікроконтролерах та ПЛІС. Аналіз показав, що реалізація адаптивного фільтру на ПЛІС обумовлено наступними перевагами, це можливість паралельного виконання кількох сотень операцій множення з накопиченням, що є важливим для реалізації фільтру.

На основі розробленої моделі в Matlab/Simulink було отримано VHDL опис адаптивного фільтру.

Проведено синтез та тестування адаптивного фільтру у за допомогою

програмного забезпечення Altera Quartus II.

Результати тестування показали, що результати фільтрації співпадають з тестовою послідовністю, що означає, що фільтр працює правильно.

Таким чином у кваліфікаційній роботі було розроблено адаптивний фільтр, який можна буде використовувати для фільтрації сигналів звукового діапазону для приглушення акустичних завад в різноманітних пристроях в режимі реального часу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Айфичер, Э. С. Цифровая обработка сигналов: практический подход [Текст]: пер. с англ. / Э. С. Айфичер, Б. У. Джервис. 2-е изд. М.: Издат. дом «Вильямс», 2004.
2. Смит С. Цифровая обработка сигналов. Практическое руководство для инженеров и научных работников / Стивен Смит – М. : Додэка_XXI, 2011. 720 с.
3. Прокис Дж. Цифровая связь: Пер. с англ. / Под ред. Д. Д. Кловского.– М.: Радио и связь, 2000.— 800 с.
4. Сергиенко, А. Б. Цифровая обработка сигналов [Текст]: учеб. пособие / А. Б. Сергиенко. 2-е изд. СПб.: Питер, 2006.
5. Проектирование цифровых схем с использованием языка VHDL: Учебное пособие/ В.В. Семенец, И.В. Хаханова, В.И. Хаханов. – ХНУРЭ, 2003. 492 с.
6. MATLAB [Электронный ресурс]: Офіційний сайт компанії Mathworks. – Режим доступа: <https://www.mathworks.com/products/matlab/> (дата обращения: 01.10.2022).
7. Maple [Электронный ресурс]: Офіційний сайт компанії Waterloo Maple Inc. – Режим доступа: <https://www.maplesoft.com/products/Maple/> (дата обращения: 01.10.2022).
8. MathCAD [Электронный ресурс]: Офіційний сайт компанії PTC. – Режим доступа: <https://www.mathcad.com/en/> (дата обращения: 01.10.2022).
9. Liang, J. Floating point unit generation and evaluation for FPGAs / J. Liang, R. Tessier, O. Mencer // Annual IEEE Symposium on Field-Programmable Custom Computing Machines. – 2003. – № 11. – С. 185–194.
10. Ifeasnor, E. C. Digital Signal Processing / E. C. Ifeasnor, B. W. Jervis. – Изд. 2-е. – Prentice Hall, 2002. – 960 с.

11. Sayed A.H. Fundamentals of adaptive filtering. John Wiley and Sons, 2003.
12. Adaptive signal processing: applications to realworld problems. J. Benesty, Y. Huang, Eds. Springer, 2003.
13. Poularikas A.D., Ramadan Z.M. Adaptive filtering primer with MATLAB. CRC Press, 2006.
14. Ogunfunmi T. Adaptive nonlinear system identification: the Volterra and Wiener model approaches. Springer Science + Business Media, 2007.
15. Dzing P.S.R. Adaptive filtering algorithms and practical implementation. 3rd ed. Springer Science + Business Media, 2008.
16. Djigan V.I. Recursive least squares – an idea whose time has come. Proc. of the 7th Intern. Workshop on Spectral Methods and Multirate Signal Processing. Moscow, 2007. PP. 255–260.
17. Treichler J., Larimore M. New processing techniques based on the constant modulus adaptive algorithm. IEEE Trans. Acoustics, Speech and Signal Processing. 1985. Vol. 33. № 2. PP. 420–431.
18. Widrow B. Thinking about thinking: the discovery of the LMS algorithm – DSP history. IEEE Signal Processing Magazine. 2005. Vol. 22. № 1. PP. 100–106.
19. Chen W.Y. Simulation techniques and standards development for digital subscriber linesystems. Macmillan Technical Publishing, 1998.
20. Starr T., Cioffi J.M., Silverman P.J. Understanding digital subscriber line technology. Prentice Hall, 1999.
21. Allen J.B., Berkley D.A. Image method for efficiently simulating smallroom acoustics. J. Acoustic Society of America. 1979. Vol. 65. № 4. PP. 943–950.
22. Digital network echo cancellers. ITUT Recommendation G.168, 04/2000. Geneva, 2001.

23. Deller J.R., Proakis J.G., Hansen G.H.L. Discretetime processing of speech signals. Prentice Hall, 1993.

24. Chen Y., LeNgoc T., Champagne B., Xu C. Recursive least squares constant modulus algorithm for blind adaptive array. IEEE Trans. Signal Processing. 2004. Vol. 52. № 5. PP. 1452–1456.

