

Олег Новосад¹, Сергій Щербовських²

¹ Кафедра систем автоматизованого проектування, Національний університет “Львівська політехніка”, вул. С. Бандери, Львів, Україна, E-mail: oleh.v.novosad@lpnu.ua, ORCID 0009-0003-6590-457X

² Кафедра систем автоматизованого проектування, Національний університет “Львівська політехніка”, вул. С. Бандери, Львів, Україна, E-mail: serhiy.v.shcherbovskykh@lpnu.ua, ORCID 0000-0001-8535-733X

ПІДХОДИ ДО СИНТЕЗУ ВИСОКОЧАСТОТНИХ ФІЛЬТРІВ ТИПУ КІНЦЕВОЇ ІМПУЛЬСНОЇ ХАРАКТЕРИСТИКИ НА ОСНОВІ АЛГОРИТМУ ЗГОРТКИ

Отримано: березень 20, 2024 / Переглянуто: квітень 02, 2024 / Прийнято: квітень 08, 2024

© Новосад О., Щербовських С., 2024

<https://doi.org/>

Анотація. Цифрові фільтри стали невід'ємною частиною сучасного світу обробки аудіо та відео, телекомунікацій та інших галузей обробки даних. Ці інструменти дозволяють видаляти з сигналу небажані компоненти, покращувати якість сигналу та вирішувати різноманітні задачі обробки сигналів. Ця стаття охоплює основні принципи цифрової фільтрації, включаючи етапи від проектування до реалізації цифрового фільтра. Вона дає цілісне розуміння процесу фільтрації, від параметризації до аналізу реальних даних на виході фільтра. Автори зосереджуються на різних типах цифрових фільтрів, зокрема на фільтрах з нескінченною імпульсною характеристикою (IIR) та фільтрах з кінцевою імпульсною характеристикою (FIR). Кожен тип має свої особливості, переваги та недоліки, які допоможуть вам вибрати найкращий фільтр для конкретного завдання. Важливою частиною статті є обговорення того, як цифрові фільтри можуть бути використані для вирішення різних завдань обробки сигналів, в тому числі для зменшення шуму. Розглянуто різні сценарії використання фільтрів у реальних застосуваннях, що допоможе читачеві зрозуміти практичний потенціал цих інструментів. Загалом, ця стаття допоможе читачам зрозуміти, як працюють цифрові фільтри, як вибрати правильний фільтр для конкретного завдання і як використовувати ці фільтри для поліпшення якості сигналу. Вона є важливим ресурсом для техніків, інженерів, студентів і всіх, хто цікавиться обробкою сигналів.

Ключові слова: високочастотний фільтр, кінцева імпульсна характеристика, нескінченна імпульсна характеристика, алгоритм

Вступ

Цифрова обробка сигналів є невід'ємною частиною сучасних технологій, і вона знаходить застосування в найрізноманітніших областях – від бездротових комунікацій до медичної техніки. Основним інструментом в цифровій обробці сигналів є цифрові фільтри, які дозволяють проводити детальний аналіз сигналів, виділяти потрібні компоненти.

У цьому контексті, однією з найважливіших задач є проектування та реалізація високочастотних фільтрів (High pass filters), які дозволяють пропускати сигнали високої частоти, при цьому блокуючи сигнали нижчої частоти.

У даній роботі для реалізації високочастотного фільтра було використано алгоритм FIR (Finite Impulse Response), написаний на мові програмування C. Застосування FIR-фільтра забезпечує стабільність, а проектування FIR-фільтра є відносно простим завданням, що робить його вибором для багатьох додатків.

Реалізація високочастотного FIR-фільтра була проведена на мікроконтролері, що дозволяє виконувати обробку сигналів у реальному часі і забезпечує високу ефективність системи.

Мікроконтролери, завдяки їх малому розміру, низькому споживанню енергії та високій продуктивності, є ідеальним рішенням для вбудованих систем обробки сигналів.

Таким чином, ця робота демонструє застосування алгоритмів FIR та мікроконтролерів у реалізації високочастотних фільтрів, і показує, як ці технології можуть бути використані для вирішення конкретних задач цифрової обробки сигналів.

Огляд сучасних джерел інформації за тематикою публікації

Літературний огляд виконано на основі джерел, пов'язаних із цифровими фільтрами.

В [1] автори зосереджені на будівництві та ефективності різних типів фільтрів та генераторів, особливо фільтр високих частот (ФВЧ), смуговий фільтр (СФ), фільтр низьких частот (ФНЧ), інвертуючий фільтр низьких частот (ІФНЧ) та біквадратичний фільтр. Розробка та реалізація цих фільтрів виконувалася за допомогою двох заземлених конденсаторів, чотирьох резисторів та методу аналітичного синтезу.

В [2] пропонують дві нові та вдосконалені біквадратичні фільтра з електронним регулюванням за напругою, які використовують три готові інтегральні схеми (ІС) LT1228, два заземлені конденсатори (ЗК) та шість резисторів. Ці компоненти дозволяють незалежно керувати коефіцієнтами підсилення для смугової (СС), верхньочастотної (ВЧ) та нижньочастотної/регулярної режекторної (НЧ/РН) фільтрації.

В [3] аналізується виклик отримання точного кутового положення і швидкості з сигналів резольвера, що вимагає резольверно-цифрового перетворення (RDC). Підкреслюється проблема гармонік, присутніх в сигналах датчиків, які можуть суттєво погіршити точність RDC. Ці гармоніки можна приглушити за допомогою фільтрів низьких частот (ФНЧ), але фазовий затримка ФНЧ може призвести до додаткових помилок в ЦАП, особливо при спробі придушення гармонік нижчих порядків.

В [4], [7] представлений метод, який обіцяє полегшити процес високошвидкісного аналого-цифрового перетворення. Цей метод ґрунтується на багатоканальній дискретизації аналогових сигналів, використовуючи концепцію узагальненої дискретизації. Він також включає в себе використання аналогових передніх фільтрів з чистою затримкою і -відгалужувачів, а також цифрового фільтра реконструкції БІХ для -каналу. Одним з важливих аспектів цього методу є вибір цифрових фільтрів реконструкції БПФ таким чином, що вираз аналогових фільтрів у замкненому вигляді можна отримати через процес інвертування матриці Вандермонда. Ця обернена матриця може бути записана як добуток двох трикутних матриць: верхньої трикутної матриці та нижньої трикутної матриці.

В [5] розглядається адаптивну решітку зі зменшеною розмірністю як обіцяючий метод зниження обчислювальної складності адаптивного формувача променя. Пропонується простий метод проектування адаптивної антени зі зменшеною розмірністю, який ґрунтується на просторових фільтрах. Цей метод включає використання набору фільтрів зі скінченною імпульсною характеристикою для створення матриці перетворення з меншим рангом, дозволяючи перетворити прийняті дані з простору елементів у простір променів. Результати моделювання показують, що цей метод ефективний для зменшення розмірності адаптивної решітки, зберігаючи при цьому продуктивність на рівні з просторовою решіткою.

В [6], [10] розглядає мінімаксий дизайн фільтра з нескінченною імпульсною характеристикою (ІІР). Стаття починає з застосування методу часткових дробів для розкладення передавальної функції на суму дробів низького порядку. Цей крок дозволяє формулювати задачу проектування мінімаксного ІІР-фільтра з використанням умов стійкості як обмежень. Далі аналізуються різні можливі розклади передавальної функції і доводиться, що розклад другого порядку постійно дає хороше наближення до оптимального розв'язку. Це суттєвий внесок у розуміння того, як можна підхопити певні властивості розкладу передавальної функції для досягнення бажаних результатів у проектуванні ІІР-фільтрів.

В [8] розглядають новий підхід до аналогових інтегральних схем, який може зменшити шуми та спотворення парного порядку, характерні для таких схем. Суть цього підходу полягає в переході від диференціальних структур до односторонніх. Це дозволяє спростити схему, уникнути синфазного зворотного зв'язку та конверсії між однополярними та диференціальними сигналами на вхідних та вихідних портах. Односторонні схеми дозволяють використовувати однокантні підсилювачі, що можуть бути значно простішими в побудові, зокрема через використання інверторних схем.

В [9] розглядають програмований аналоговий фільтр четвертого порядку зі скінченною імпульсною характеристикою (AFIR), що працює на частоті 3,25 ГГц/с. Цей фільтр призначений для гнучкої обробки аналогових сигналів в дискретних часових інтервалах (DT-ASP). Пропонований фільтр розрахований на повне використання смуги пропускання аж до швидкості Найквіста, і програмується за допомогою набору коефіцієнтів множника. У використаній реалізації функції передавальної характеристики в якості множників коефіцієнтів використовуються цифро-аналогові перетворювачі з розділеною ємністю (split CDAC). Це забезпечує високу лінійність у всьому діапазоні частот і низьке енергоспоживання.

Постановка проблеми

У сучасному світі, де переважає цифрова інформація, цифрова обробка сигналів стає все більш важливою. Однак, цифрові сигнали часто містять різні частотні компоненти, з яких деякі можуть бути небажаними або заважають аналізу основних даних. Саме тут на допомогу приходять цифрові фільтри.

Однією з ключових проблем, які стають перед розробниками і інженерами, є проектування та реалізація високочастотних фільтрів. Високочастотні фільтри є важливими, оскільки вони дозволяють пропускати сигнали високої частоти та відфільтровувати сигнали нижчої частоти. Вони зокрема використовуються в радіокомунікаційних системах, аудіотехніці, біомедичних дослідженнях та багатьох інших областях.

Однак, реалізація високочастотних фільтрів ставить перед нами ряд викликів. По-перше, потрібно вибрати правильний тип фільтра (IIR чи FIR) в залежності від конкретних вимог до системи. По-друге, проектування та реалізація високочастотних фільтрів вимагає глибоких знань у математиці та цифровій обробці сигналів. По-третє, високочастотні фільтри мають бути реалізовані таким чином, щоб вони були ефективні та надійні у роботі.

В результаті, ця проблема вимагає розробки ефективних методів проектування та реалізації високочастотних фільтрів, які б могли задовольнити потреби різних областей застосування.

Загальна формула для FIR високочастотного фільтра (1):

$$y(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) + \dots + b_Mx(n-M) - a_1y(n-1) - a_2y(n-2) - \dots - a_Ny(n-N), \quad (1)$$

де ' $x(n)$ ' – вхідний аудіосигнал в час ' n ', ' $y(n)$ ' – вихідний аудіосигнал в час ' n ', ' b_0 ', ' b_1 ', ' b_2 ', ..., ' b_M ' – передні коефіцієнти фільтра (також відомі як ваги фільтра), ' a_1 ', ' a_2 ', ' a_3 ', ..., ' a_N ' – зворотні коефіцієнти (також відомі як коефіцієнти коефіцієнтів минулих значень вихідного сигналу), ' M ' і ' N ' – порядок фільтра, який визначає складність фільтра та частоту зрізу.

Огляд:

Вхідний аудіосигнал ' $x(n)$ ' множиться на передні коефіцієнти ' b_0 ', ' b_1 ', ' b_2 ', ..., ' b_M ' та сумується.

Сума потім вичитана на продукт зворотніх коефіцієнтів ' a_1 ', ' a_2 ', ' a_3 ', ..., ' a_N ' та минулих значень вихідного сигналу ' $y(n-1)$ ', ' $y(n-2)$ ', ..., ' $y(n-N)$ '.

Результат – вихідний аудіосигнал ' $y(n)$ '.

Фільтр FIR високопрохідний дозволяє проходити компоненти низьких частот, а компоненти високих частот пригнічує. Частота зрізу визначає точку, з якої фільтр починає пригнічувати сигнал. Порядок та коефіцієнти фільтра $'b_0', 'b_1', 'b_2', \dots, 'b_M'$ та $'a_1', 'a_2', 'a_3', \dots, 'a_N'$ обираються на основі бажаної частоти зрізу та характеристик фільтра (наприклад, фільтр Баттерворта, Чебішева або Еліптичний фільтр).

Виклад основного матеріалу

Високопрохідний фільтр (HPF) –це тип цифрового фільтра, який дозволяє проходити сигнали з низькими частотами та пригнічує сигнали з високими частотами. Він зазвичай використовується в аудіообробці для видалення шумів низьких частот або для підвищення якості звуку.

HPF працює за принципом зміни фази вхідного сигналу в залежності від його частоти. Сигнали з низькими частотами проходять через фільтр без значних змін, тоді як сигнали з високими частотами затримуються або пригнічуються.

Ця поведінка досягається за допомогою характеристики відгуку фільтру, яка визначає, як фільтр змінює фазу вхідного сигналу в залежності від його частоти. Для HPF характеристика відгуку має певну "вирізку" на певній частоті, яка називається частотою зрізу. Сигнали з частотами нижчими від частоти зрізу проходять через фільтр без змін, а сигнали з частотами вищими за частоту зрізу пригнічуються.

Існує декілька типів фільтрів, які можуть бути використані для реалізації HPF, кожен з яких має свої переваги та недоліки:

Алгоритм Butterworth – це метод проектування фільтрів, який дозволяє отримати фільтр Butterworth, який має найпростішу структуру та найрівномірнішу амплітудну відповідь на частотах вище від частоти зрізу. Він зазвичай використовується в тих випадках, коли стабілізація є ключовою вимогою, а відгук на частотах вище від частоти зрізу менш важливий.

Створення Butterworth фільтру починається з вибору його порядку, який визначає кількість секцій, які фільтр має. Вищий порядок фільтру дозволяє отримати більш плавну характеристику відгуку, але також збільшує його складність.

Коефіцієнти фільтру обчислюються за допомогою формули (2):

$$b_n = \frac{1}{1 + \left(\frac{\omega_n}{\omega_c}\right)^{2N}} \sin(\omega_n - \omega_c), \quad (2)$$

де ω_n – n-та частота зрізу, яка обчислюється як

$$\omega_n = \frac{(2\pi \cdot f_0) \cdot n}{N}, \quad (3)$$

для $n = 1, 2, \dots, N$.

Основні переваги використання алгоритму Butterworth:

- Найпростіша структура: Butterworth фільтри мають найпростішу структуру серед усіх типів фільтрів, що робить їх легкими для реалізації.

- Найкраща стабілізація: Butterworth фільтри мають найкращу стабілізацію, що означає, що вони мають найменші спотворення навіть при значних відхиленнях вхідного сигналу.

- Рівномірна амплітудна відповідь: Butterworth фільтри мають найрівномірнішу амплітудну відповідь на частотах вище від частоти зрізу.

Основні недоліки використання алгоритму Butterworth:

- Більш низька амплітудна відповідь: Butterworth фільтри мають менш агресивну характеристику відгуку, що означає, що вони мають більш низьку амплітудну відповідь на частотах вище від частоти зрізу.

- Більш низька селективність: Butterworth фільтри мають менш селективну характеристику відгуку, що означає, що вони можуть пропускати сигнали з частотами, які близькі до частоти зрізу.

Алгоритм Chebyshev – це метод проектування фільтрів, що дозволяє отримати фільтр Chebyshev, та має більш агресивну характеристику відгуку, ніж Butterworth фільтри, проте має більш неоднорідну амплітудну відповідь. Він зазвичай використовується в тих випадках, коли потрібна більш агресивна характеристика відгуку, а відгук на частотах вище від частоти зрізу є менш важливим.

Створення Chebyshev фільтру починається з вибору його порядку, який визначає кількість секцій, які фільтр має. Вищий порядок фільтру дозволяє отримати більш агресивну характеристику відгуку, але також збільшує його складність.

Коефіцієнти фільтру обчислюються за допомогою формули (4):

$$b_n = \frac{\cos(\omega_n - \omega_c) - \cos(\omega_n + \omega_c)}{2(1 - \cos(2\omega_n)) + 2 \cos(\omega_n - \omega_c) \cos(\omega_n + \omega_c)}, \quad (4)$$

де ω_n – n-та частота зрізу, яка обчислюється як (3)

Основні переваги використання алгоритму Chebyshev:

- Більш агресивна характеристика відгуку: Chebyshev фільтри мають більш агресивну характеристику відгуку, що дозволяє їм пропускати сигнали з частотами, які близькі до частоти зрізу.

- Менше ресурси для реалізації: Chebyshev фільтри вимагають менше ресурсів для реалізації, ніж інші типи фільтрів.

Основні недоліки використання алгоритму Chebyshev:

- Більш неоднорідна амплітудна відповідь: Chebyshev фільтри мають більш неоднорідну амплітудну відповідь, ніж Butterworth фільтри.

- Більше спотворень: Chebyshev фільтри можуть мати більше спотворень, ніж Butterworth фільтри.

Алгоритм Elliptic – це метод проектування фільтрів, який дозволяє отримати фільтр Elliptic, який має найкращу амплітудну відповідь на частотах вище від частоти зрізу, але вимагає більше обчислювальних ресурсів для реалізації. Він зазвичай використовується в тих випадках, коли важливо мати найкращу амплітудну відповідь на частотах вище від частоти зрізу, а відгук на частотах вище від частоти зрізу менш важливий.

Створення Elliptic фільтру починається з вибору його порядку, який визначає кількість секцій, які фільтр має. Вищий порядок фільтру дозволяє отримати більш плавну характеристику відгуку, але також збільшує його складність.

Коефіцієнти фільтру обчислюються за допомогою формули (5):

$$b_n = \frac{\sin(\omega_n - \omega_c)(1 + \cos(\omega_n - \omega_c))}{\sin(2\omega_n - 2\omega_c)}, \quad (5)$$

де ω_n – n-та частота зрізу, яка обчислюється як (3).

Основні переваги використання алгоритму Elliptic:

- Найкраща амплітудна відповідь: Elliptic фільтри мають найкращу амплітудну відповідь на частотах вище від частоти зрізу.

- Плавна характеристика відгуку: Elliptic фільтри мають плавну характеристику відгуку, що дозволяє їм пропускати сигнали з частотами, які близькі до частоти зрізу.

Основні недоліки використання алгоритму Elliptic:

- Більше обчислювальних ресурсів: Elliptic фільтри вимагають більше обчислювальних ресурсів для реалізації, ніж інші типи фільтрів.

- Більше спотворень: Elliptic фільтри можуть мати більше спотворень, ніж інші типи фільтрів.

Алгоритм Bessel – це метод проектування фільтрів, який дозволяє отримати фільтр Bessel, який має найкращу амплітудну відповідь на частотах вище від частоти зрізу, але має менш агресивну характеристику відгуку, ніж Chebyshev фільтри. Він зазвичай використовується в тих випадках, коли важливо мати найкращу амплітудну відповідь на частотах вище від частоти зрізу, а відгук на частотах вище від частоти зрізу менш важливий.

Створення Bessel фільтру починається з вибору його порядку, який визначає кількість секцій, які фільтр має. Вищий порядок фільтру дозволяє отримати більш плавну характеристику відгуку, але також збільшує його складність.

Коефіцієнти фільтру обчислюються за допомогою формули (6):

$$b_n = \frac{\left(\frac{2}{(n + 0,4) \sin(\omega_n - \omega_c)} \right)}{(\omega_n - \omega_c)}, \quad (6)$$

де ω_n – n-та частота зрізу, яка обчислюється як (3).

Основні переваги використання алгоритму Bessel:

- Найкраща амплітудна відповідь: Bessel фільтри мають найкращу амплітудну відповідь на частотах вище від частоти зрізу.
- Менш агресивна характеристика відгуку: Bessel фільтри мають менш агресивну характеристику відгуку, ніж Chebyshev фільтри.

Основні недоліки використання алгоритму Bessel:

- Більше спотворень: Bessel фільтри можуть мати більше спотворень, ніж інші типи фільтрів.
- Більше ресурси для реалізації: Bessel фільтри вимагають більше ресурсів для реалізації, ніж інші типи фільтрів.

У статті хотів б розглянути два основних типа HPF:

IIR (Infinite Impulse Response) – це тип цифрових фільтрів, які мають нескінченну імпульсну відповідь. Вони відрізняються від FIR (Finite Impulse Response) фільтрів, які мають скінченну імпульсну відповідь. IIR фільтри зазвичай використовуються в тих випадках, коли потрібно отримати фільтр з певною характеристикою відгуку, але з меншим числом коефіцієнтів, ніж FIR фільтри.

IIR фільтри можуть бути реалізовані за допомогою різних архітектур, таких як Sallen-Key, активний RC фільтр або цифровий фільтр. Їх коефіцієнти обчислюються за допомогою наступних формул (7,8):

$$a_n = \frac{1}{b_n}, \quad (7)$$

b_n = коефіцієнти фільтру

$$y_n = b_n x_n + a_n y_{(n-1)} + a_n e_n, \quad (8)$$

де x_n – вхідний сигнал, y_n – вихідний сигнал, e_n – шум, який додається до вихідного сигналу.

Основні переваги використання IIR фільтрів:

- Менше коефіцієнтів: IIR фільтри можуть бути реалізовані з меншим числом коефіцієнтів, ніж FIR фільтри.
- Менше спотворень: IIR фільтри можуть мати менше спотворень, ніж FIR фільтри.

Основні недоліки використання IIR фільтрів:

- Більше обчислювальних ресурсів: IIR фільтри вимагають більше обчислювальних ресурсів для реалізації, ніж FIR фільтри.
- Більше складність: IIR фільтри можуть бути більш складними для проектування та реалізації, ніж FIR фільтри.

Далі розглянемо тип FIR з використанням алгоритму згортки:

```
// the FIR filter function
void applyHighPassFilter(int16_t *input, int16_t *output, int length)
{
    int32_t acc; // accumulator for MACs
    int16_t *coeffp; // pointer to coefficients
    int16_t *inputp; // pointer to input samples
    int n;
    int k;
    // put the new samples at the high end of the buffer
    memcpy(&insamp[FILTER_LEN - 1], input,
           length * sizeof(int16_t));
    // apply the filter to each input sample
    for (n = 0; n < length; n++)
    {
        // calculate output n
        coeffp = coeffs;
        inputp = &insamp[FILTER_LEN - 1 + n];
        // load rounding constant
        acc = 1 << 14;
        // perform the multiply-accumulate
        for (k = 0; k < FILTER_LEN; k++)
        {
            acc += (int32_t)(*coeffp++) * (int32_t)(*inputp--);
        }
        // saturate the result
        if (acc > 0x3fffffff)
        {
            acc = 0x3fffffff;
        }
        else if (acc < -0x40000000)
        {
            acc = -0x40000000;
        }
        // convert from Q30 to Q15
        output[n] = (int16_t)(acc >> 15);
    }
    // shift input samples back in time for next time
    memmove(&insamp[0], &insamp[length],
            (FILTER_LEN - 1) * sizeof(int16_t));
}
```

Цей код C реалізує фільтр з кінцевою імпульсною характеристикою (FIR) високих частот. Ось детальний опис його роботи:

Застосування фільтру:

Функція *applyHighPassFilter()* очікує масив вхідних зразків, масив для зберігання вихідних зразків, та довжину цих масивів як вхід.

- Спочатку вона копіює нові вхідні зразки у буфер *insamp[]*.
- Потім, для кожного вхідного зразка, вона використовує вкладений цикл для виконання операції згортки. Ця операція включає множення кожного з коефіцієнтів фільтру на відповідний вхідний зразок, сумування цих добутків, та зберігання результату у вихідному масиві. Змінна *acc* служить накопичувачем для цих множень та додавань.
- Вона також включає механізм насичення для запобігання числового переповнення, яке може виникнути через використання арифметики з фіксованою комою. Результат потім масштабується вниз з формату Q30 у формат Q15 і зберігається у вихідному масиві.

Підходи до синтезу високочастотних фільтрів типу кінцевої імпульсної...

- Наостанок, вона "зсуває" буфер `insamp[]`, переміщуючи елементи, щоб зробити місце для нових вхідних зразків у наступному виклику функції. Це здійснюється за допомогою функції `memmove`.

Це стандартна реалізація фільтру FIR із фіксованими коефіцієнтами у сценарії вбудованої системи. Він використовує прямолінійний підхід для виконання операції фільтрації FIR та ефективний щодо використання пам'яті та обчислювальних вимог.

Також для полегшення пошуку коефіцієнтів було написано Python script, для пошуку коефіцієнтів, за вказаними параметрами:

```
import numpy as np
import scipy.signal as signal
# Define filter specifications
sample_rate = 16000 # Hz
cutoff_frequency = 200 # Hz
filter_order = 101 # Adjust the filter order as needed
# Design the high-pass FIR filter using the Kaiser window method
nyquist = 0.5 * sample_rate
cutoff_normalized = cutoff_frequency / nyquist
beta = 5.0 # Adjust the beta parameter for the Kaiser window (higher values result
in a narrower transition band)
# Design the filter
filter_coefficients = signal.firwin(filter_order, cutoff_normalized, pass_zero=False,
window=('kaiser', beta))
# Print the coefficients (scaled for fixed-point representation)
scaled_coefficients = np.round(filter_coefficients * (2**15 - 1)).astype(np.int16)
print(len(scaled_coefficients))
print(' '.join(str(scaled_coefficients).split()).replace(" ", ", "))
```

- *Імпорт необхідних бібліотек:* Код починається з імпорту двох бібліотек Python – `numpy` та `scipy.signal`. `Numpy` – це бібліотека для Python, яка додає підтримку великих багатовимірних масивів та матриць, а також велику колекцію математичних функцій для роботи з цими масивами. `Scipy` – це ще одна бібліотека python, що використовується для високорівневих обчислень.

- *Визначте специфікації фільтра:* Воно визначає певні параметри фільтра, такі як ‘`sample_rate`’, ‘`cutoff_frequency`’ та ‘`filter_order`’. Вони представляють частоту, за якою сигнал вибірково відтворюється, бажану частоту зрізу високочастотного фільтра та порядок фільтра відповідно.

- *Створення високочастотного FIR-фільтра:* Він створює високочастотний FIR-фільтр за допомогою функції `"firwin"` з модуля `scipy.signal`. Функція `"firwin"` використовується для створення FIR-фільтра з методом вікна Кайзера. Частота зрізу нормалізується шляхом ділення її на частоту Найквіста (яка становить половину частоти дискретизації).

- *Масштабування коефіцієнтів для представлення з фіксованою комою:* Коефіцієнти фільтра масштабуються для представлення з фіксованою комою шляхом множення їх на ‘`(2**15 - 1)`’. Після цього масштабовані коефіцієнти округлюються до найближчого цілого числа і перетворюються на 16-бітні знакові цілі числа за допомогою методу ‘`astype`’ `numpy`.

- *Друк масштабованих коефіцієнтів:* Нарешті, друкуються масштабовані коефіцієнти. Простір в навколишньому середовищі видаляється і замінюється комами.

В цілому, цей код корисний для створення високочастотних FIR-фільтрів у Python та представляє коефіцієнти у форматі, який можна використовувати в середовищах з арифметикою фіксованої точки, таких як вбудовані системи або мови програмування C чи C++.

Після виконання Python ми отримуємо масив (рис. 1) з коефіцієнтами які слід застосувати для нашого High Pass Filter


```
#define FILTER_LEN 101
int16_t coeffs[FILTER_LEN] =
{
    5, 6, 7, 8, 8, 8, 8, 7, 6, 3,
    0, -4, -9, -16, -24, -33, -43, -56, -69, -85,
    -102, -121, -141, -164, -188, -213, -240, -268, -298, -328,
    -360, -392, -425, -458, -491, -523, -555, -587, -617, -646,
    -674, -700, -723, -745, -764, -780, -794, -805, -813, -817,
    31943, -817, -813, -805, -794, -780, -764, -745, -723, -700,
    -674, -646, -617, -587, -555, -523, -491, -458, -425, -392,
    -360, -328, -298, -268, -240, -213, -188, -164, -141, -121,
    -102, -85, -69, -56, -43, -33, -24, -16, -9, -4,
    0, 3, 6, 7, 8, 8, 8, 8, 7, 6, 5};
```

Рис. 1. Масив коефіцієнтів

По виконанні програми на мікроконтролері ми отримуємо вихідний аудіо файл (рис. 2) з відсіченими шумами на рівні частоти 200 Hz.

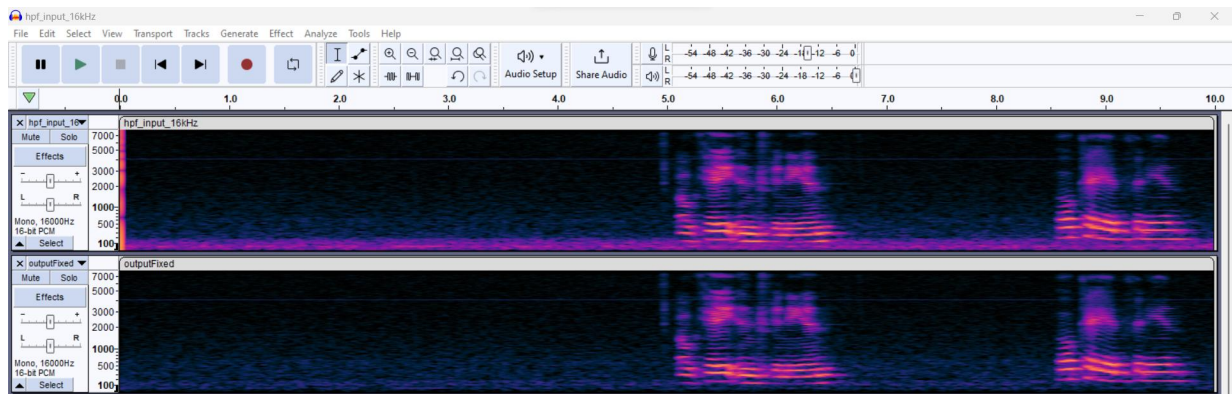


Рис. 2. Спектограма вхідних та вихідних даних

Висновки

Високочастотні фільтри з кінцевою імпульсною характеристикою (FIR) є важливими інструментами в цифровій обробці сигналів. Вони використовуються для видалення низькочастотних компонентів з сигналу або для відокремлення високочастотних компонентів. Вони можуть бути реалізовані в різних мовах програмування, включаючи C та Python. Процес проектування фільтра зазвичай включає визначення специфікацій фільтра, створення коефіцієнтів фільтра та їх масштабування для використання у фіксованій точці або іншому середовищі обчислень.

У Python є корисні бібліотеки, такі як `scipy` та `numpy`, які надають готові до використання функції для проектування FIR-фільтрів. З іншого боку, мови програмування низького рівня, як-от C, дозволяють більшу контроль над процесом обробки сигналу та можуть бути використані для вбудованих систем, які мають обмежені ресурси. Однак, незалежно від використовуваної мови або підходу, ключовою частиною проектування FIR-фільтра є правильне визначення його характеристик (таких як частота зрізу та порядок фільтра) та використання відповідного методу для розрахунку коефіцієнтів фільтра.

Перелік використаних джерел

- [1] S. -F. Wang, H. -P. Chen, Y. Ku and M. -X. Zhong, "Analytical Synthesis of High-Pass, Band-Pass and Low-Pass Biquadratic Filters and its Quadrature Oscillator Application Using Current-Feedback Operational Amplifiers," in IEEE Access, vol. 9, pp. 13330-13343, 2021, <https://doi.org/10.1109/ACCESS.2021.3050751>
- [2] H. -P. Chen, S. -F. Wang, C. -Y. Chang, S. -J. Chen and Y. -H. Chen, "Two New and Improved Electronically Adjustable Voltage-Mode Multifunction Biquadratic Filters With Independent Control of Band-Pass, High-Pass, and Low-Pass/Regular Notch Filter Gains," in IEEE Access, vol. 11, pp. 20494-20514, 2023, <https://doi.org/10.1109/ACCESS.2023.3249790>

- [3] R. Wang, Z. Wu and Y. Shi, "Suppressing Harmonics in Resolver Signals via FLL-Based Complementary Filters," in *IEEE Access*, vol. 9, pp. 158402-158411, 2021, <https://doi.org/10.1109/ACCESS.2021.3130921>
- [4] B. D. Sahoo and G. Manganaro, "Generalized Sampling-Based Multi-Channel Sampling of Signals Realized With Pure Delay Analog Filters and Digital FIR Reconstruction Filters," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 9, pp. 3323-3327, Sept. 2023, <https://doi.org/10.1109/TCSII.2023.3275650>
- [5] P. Yang and W. Zhou, "Design and Experiment of Reduced-Dimension Adaptive Array Based on FIR Filters," in *IEEE Antennas and Wireless Propagation Letters*, vol. 20, no. 2, pp. 204-208, Feb. 2021, <https://doi.org/10.1109/LAWP.2020.3044303>
- [6] H. Qi, Z. G. Feng, K. F. C. Yiu and S. Nordholm, "Optimal Design of IIR Filters via the Partial Fraction Decomposition Method," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 8, pp. 1461-1465, Aug. 2019, <https://doi.org/10.1109/TCSII.2018.2879914>
- [7] X. X. Zheng, J. Yang, S. Y. Yang, W. Chen, L. Y. Huang and X. Y. Zhang, "Synthesis of Linear-Phase FIR Filters With a Complex Exponential Impulse Response," in *IEEE Transactions on Signal Processing*, vol. 69, pp. 6101-6115, 2021, <https://doi.org/10.1109/TSP.2021.3115352>
- [8] T. He, M. Kareppagoudr, Y. Zhang, E. Caceres, U. -K. Moon and G. C. Temes, "Noise Filtering and Linearization of Single-Ended Sampled-Data Circuits," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 4, pp. 1331-1341, April 2019, <https://doi.org/10.1109/TCSI.2018.2885802>
- [9] S. Park, D. Shin, K. -J. Koh and S. Raman, "A Low-Power, High-Linearity Wideband 3.25 GS/s Fourth-Order Programmable Analog FIR Filter Using Split-CDAC Coefficient Multipliers," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 4, pp. 1576-1590, April 2020, <https://doi.org/10.1109/TMTT.2020.2967388>
- [10] H. Qi, Z. G. Feng, K. F. C. Yiu and S. Nordholm, "Optimal Design of IIR Filters via the Partial Fraction Decomposition Method," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 8, pp. 1461-1465, Aug. 2019, <https://doi.org/10.1109/TCSII.2018.2879914>

Oleh Novosad¹, Serhiy Shcherbovskykh²

¹ Computer Aided Design Department, Lviv Polytechnic National University, Ukraine, Lviv, S. Bandery street 12, E-mail: oleh.v.novosad@lpnu.ua, ORCID 0009-0003-6590-457X

² Computer Aided Design Department, Lviv Polytechnic National University, Ukraine, Lviv, S. Bandery street 12, E-mail: serhiy.v.shcherbovskykh@lpnu.ua, ORCID 0000-0001-8535-733X

APPROACHES TO SYNTHESIS OF HIGH-FREQUENCY FILTERS OF FINITE IMPULSE RESPONSE TYPE BASED ON CONVOLUTION ALGORITHM

Received: March 20, 2024 / Revised: April 02, 2024 / Accepted: April 08, 2024

© Novosad O., Shcherbovskykh S., 2024

Abstract. Digital filters are widely used in audio and video processing, telecommunications, and other data processing industries. They enable the removal of unwanted components from signals, improve signal quality, and solve various signal processing tasks. This article covers the basic principles of digital filtering, including the stages from design to implementation of a digital filter. The text provides a comprehensive overview of the filtering process, covering parameterization and analysis of real data at the filter output. The authors discuss various types of digital filters, such as infinite impulse response (IIR) and finite impulse response (FIR) filters, each with its own unique characteristics, advantages, and disadvantages. This information will assist in selecting the most appropriate filter for a given task. The article discusses the use of digital filters for signal processing tasks, including noise reduction, in various real-world scenarios. The language used is clear, concise, and objective, with a formal register and precise word choice. The article discusses the use of digital filters for signal processing tasks, including noise reduction, in various real-world scenarios. The sentences and paragraphs have a logical flow of information with causal connections between statements. The text is free from grammatical errors, spelling mistakes, and punctuation errors. It aims to help readers understand the functionality of digital filters, how to select the appropriate filter for a specific task, and how to use them to enhance signal quality. The text adheres to conventional academic structure and formatting, with consistent citation and footnote style. No changes in content have been made to the original text. It is an important resource for technicians, engineers, students, and anyone interested in signal processing.

Keywords: high-pass filter, finite impulse response, infinite impulse response, algorithm