

УДК.62-23

Антонюк Я. М. ORCID: 0009-0005-7680-5950 (ІТС НАН України)
канд. техн. наук Корольов А. П. ORCID: 0000-0001-6281-063X (ВІТІ ім. Героїв Крут)
Лапа В. І. ORCID: 0009-0006-8017-0334 (ВІТІ ім. Героїв Крут)
канд. техн. наук Мацаєнко А. М. ORCID: 0000-0003-1149-7318 (ВІТІ ім. Героїв Крут)

ОПТИМІЗАЦІЯ РЕАЛІЗАЦІЇ ЦИФРОВИХ ГРЕБІНЧАСТИХ ФІЛЬТРІВ НА БАЗІ СУЧАСНИХ МІКРОКОНТРОЛЕРІВ ДЛЯ ЗАДАЧ ОБРОБКИ СИГНАЛІВ У РЕАЛЬНОМУ ЧАСІ

У статті представлено результати дослідження та практичної реалізації цифрових гребінчастих фільтрів на базі мікроконтролерів серії STM32 (ядро ARM Cortex-M4). Актуальність теми зумовлена необхідністю підвищення обчислювальної ефективності та зниження апаратних витрат у системах цифрової обробки сигналів реального часу. Вдосконалення методики реалізації гребінчастих фільтрів дозволяє оптимізувати процеси децимації в сучасних телекомунікаційних стандартах та покращити якість фільтрації періодичних завад у складних заводських умовах.

Метою статті є дослідження комбінаційного методу програмної та апаратної реалізації рекурсивних та нерекурсивних гребінчастих фільтрів на базі мікроконтролера STM32 для забезпечення їх роботи у режимі реального часу при мінімальному використанні ресурсу пам'яті, а також оцінка їхніх частотних характеристик при обробці сигналів із заданою частотою дискретизації.

Розглянуто математичні моделі рекурсивних і нерекурсивних гребінчастих фільтрів, а також особливості їхньої адаптації до архітектур із обмеженими обчислювальними ресурсами. Запропоновано методику оптимізації алгоритму через використання кільцевих буферів, механізмів прямого доступу до пам'яті та бібліотеки CMSIS-DSP. Проведено порівняльний аналіз завантаженості центрального процесора при використанні арифметики з плаваючою та фіксованою комою. Експериментально підтверджено ефективність запропонованих рішень для фільтрації періодичних завад у реальному часі.

Практичне значення отриманих результатів: зниження обчислювального навантаження на центральний процесор; створення готових програмних шаблонів на базі бібліотеки CMSIS-DSP, що скорочує термін розробки та налагодження вбудованих систем на базі ядер Cortex-M; можливість реалізації фільтрів високого порядку на мікроконтролерах з обмеженими ресурсами без втрати частоти дискретизації.

Ключові слова: гребінчасті фільтри, цифрова обробка сигналів, STM32, ARM Cortex-M4, реальний час, контролери прямого доступу до пам'яті, CMSIS-DSP, оптимізація.

Y. Antoniuk, A. Korolov, V. Lapa, A. Matsaienko. Optimization of the digital comb filters implementation based on modern microcontrollers for real-time signal processing tasks

The article presents the results of both research and practical implementation of digital comb filters using STM32 series microcontrollers based on the ARM Cortex-M4 core. The topic is important because it addresses the need to improve computational efficiency and reduce hardware costs in real-time digital signal processing systems. Enhancing the methodology for implementing comb filters makes it possible to optimize decimation processes in modern telecommunication standards and improve the suppression of intermittent interference under challenging noise conditions.

The aim of the article is to investigate a combined software–hardware approach to the implementation of recursive and non-recursive comb filters based on the STM32 microcontroller, in order to ensure their real-time operation with minimal memory usage, as well as to evaluate their frequency characteristics during the processing of signals with a specified sampling rate.

Mathematical models of recursive and non-recursive comb filters are considered, as well as the features of their adaptation to architectures with limited computing resources. A method for optimizing the algorithm through the use of ring buffers, direct access to memory and the CMSIS-DSP library is proposed. A comparative analysis of the CPU workload for calculations with floating-point and fixed-point arithmetic is carried out. The effectiveness of the proposed solutions for filtering periodic interference under real time conditions has been experimentally confirmed.

The practical significance of the results includes reduced CPU utilization, the development of software templates based on the CMSIS-DSP library that shorten the development and debugging time of embedded systems based on Cortex-M cores, and the ability to implement high-order filters on resource-constrained microcontrollers without loss of sampling rate.

Keywords: comb filters, digital signal processing, STM32, ARM Cortex-M4, real-time, direct memory access controllers, CMSIS-DSP, optimization.

Постановка задачі в загальному вигляді. На сучасному етапі розвитку систем цифрової обробки сигналів (Digital Signal Processing, DSP) на базі вбудованих рішень ключовою вимогою є забезпечення високої точності фільтрації при мінімальному використанні обчислювальних ресурсів. Одним із найефективніших інструментів для селекції або режекції періодичних складових сигналу є гребінчасті фільтри (*Comb Filters*). Вони знаходять широке застосування в задачах пригнічення мережевих завад, обробці аудіосигналів, децимації в сигма-дельта перетворювачах, а також у системах розпізнавання акустичних сигналів об'єктів.

У статті розглянуто методи оптимізації апаратно-програмного комплексу при побудові гребінчастих фільтрів із використанням мікроконтролерів STM32 (Cortex-M3).

Наукова новизна роботи визначається розробкою цілісної моделі конвеєрної обробки сигналів, яка інтегрує роботу апаратного блоку математичного процесора (Floating Point Unit, FPU) та контролера прямого доступу до пам'яті (Direct Memory Access, DMA) у циклічному режимі для реалізації гребінчастої фільтрації. Це дозволило розширити динамічний діапазон оброблюваних частот без збільшення завантаження центрального процесора, забезпечуючи високу селективність режекції завад у реальному часі.

Аналіз публікацій за темою дослідження. Сучасні дослідження в галузі вбудованих систем [2; 3; 5] демонструють тенденцію до перенесення складних алгоритмів фільтрації безпосередньо на периферійні пристрої. Традиційні методи реалізації гребінчастих фільтрів на базі лінійних буферів часто стикаються з проблемою обмеженої пам'яті та нераціонального використання циклів процесора. Водночас архітектурні особливості сучасних мікроконтролерів серії *STM32*, зокрема наявність спеціалізованого апаратного блоку для виконання математичних операцій над числами з плаваючою комою, контролерів прямого доступу до пам'яті та спеціалізованих інструкцій (Single Instruction, Multiple Data, SIMD), відкривають нові можливості для оптимізації рекурсивних структур фільтрації.

У роботі [2] досліджено і представлено ефективну методіку лише модифікованих косинусних децимаційних фільтрів. Проаналізовано обчислювальну складність лише нерекурсивних фільтрів.

У роботі [3] запропоновано окремий підхід до проектування гребінчастого фільтра з використанням струмового конвеєра для усунення небажаних гармонійних завад. У цьому підході для побудови гребінчастого фільтра використовується ряд інвертованих смугових фільтрів. Але не приділено достатньо уваги фазо-частотним характеристикам (ФЧХ) фільтрів.

У дослідженні [5] проаналізовано вплив окремих алгоритмів на швидкість обчислень на ядрах *ARM Cortex-M4* і *Cortex-M7*. Але не приділено уваги підходу при визначенні кількості відліків.

Таким чином, проаналізовано джерела, автори яких розкрили певні аспекти розробки та використання гребінчастих фільтрів, водночас залишається актуальним наукове завдання оптимізації їх використання з метою забезпечення обробки сигналів в режимі реального часу із урахуванням обмежених обчислювальних потужностей.

Метою роботи є дослідження комбінаційного методу програмної та апаратної реалізації рекурсивних і нерекурсивних гребінчастих фільтрів на базі мікроконтролера *STM32* для забезпечення їхньої роботи у режимі реального часу при мінімальному використанні ресурсу пам'яті, а також оцінка їхніх частотних характеристик при обробці сигналів із заданою частотою дискретизації.

Виклад основного матеріалу

Нерекурсивні гребінчасті фільтри (Finite Impulse Response, *FIR*) описуються різницеvim рівнянням (1) [7]:

$$y[n] = x[n] + \alpha \cdot x[n - K], \quad (1)$$

де $y[n]$ – вихідний сигнал;
 $x[n]$ – вхідний сигнал;
 K – затримка у відліках;
 α – коефіцієнт підсилення (зазвичай $|\alpha| \leq 1$).

Структурна схема такого фільтра наведена на рисунку 1.

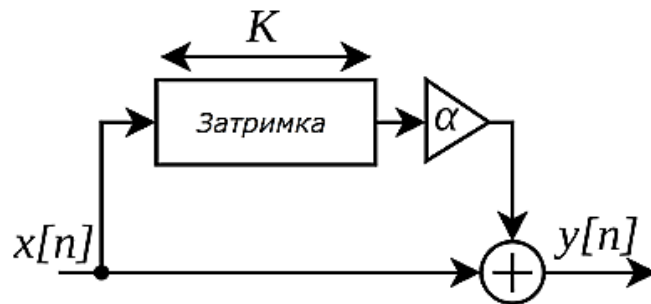


Рис. 1. Схема нерекурсивного гребінчастого фільтра (без зворотного зв'язку)

Передавальна функція гребінчастого FIR-фільтра описується формулою (2) [7]:

$$H(z) = 1 + \alpha \cdot z^{-K}. \quad (2)$$

Для розрахунку амплітудно-частотної характеристики (АЧХ) необхідно z замінити на $e^{j\omega}$ та розрахувати модуль передавальної функції. Приклад розрахунку АЧХ гребінчастого FIR-фільтра ($K = 10$, $\alpha = 0,9$) наведено на рисунку 2. Частота нормована відносно частоти дискретизації. АЧХ має періодичні піки та провали – «зубці». Піки підсилюють певні частоти, а провали повністю або частково послаблюють відповідні частоти. Між провалами амплітуда змінюється доволі плавно. Відстань між «зубцями» залежить від затримки K . Для частоти дискретизації f_s інтервал між ними дорівнює $\Delta f = f_s / K$.

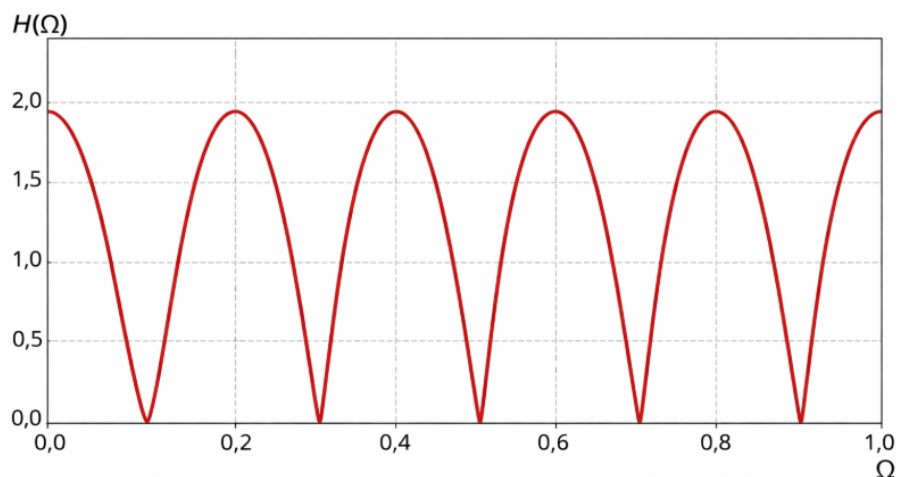


Рис. 2. АЧХ нерекурсивного гребінчастого фільтра ($K = 10$, $\alpha = 0,9$)

На рисунку 3 наведена ФЧХ цього фільтра, яка розраховується як аргумент передавальної функції. Нерекурсивні гребінчасті фільтри мають лінійну фазову характеристику, що важливо для збереження форми сигналу.

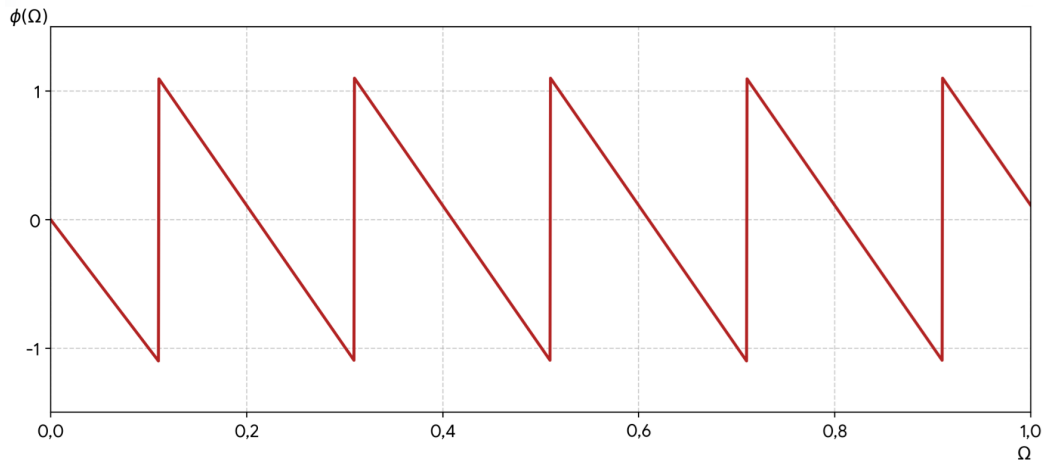


Рис. 3. ФЧХ нерекурсивного гребінчастого фільтра ($K = 10, \alpha = 0,9$)

Рекурсивні гребінчасті фільтри (Infinite Impulse Response, *IIR*) описуються різницевою рівнянням (3) [7]:

$$y[n] = x[n] + \alpha \cdot y[n - K], \quad (3)$$

де $y[n - K]$ – затримані відліки вихідного сигналу;
 α – коефіцієнт зворотного зв'язку.

Структурна схема такого фільтра наведена на рисунку 4.

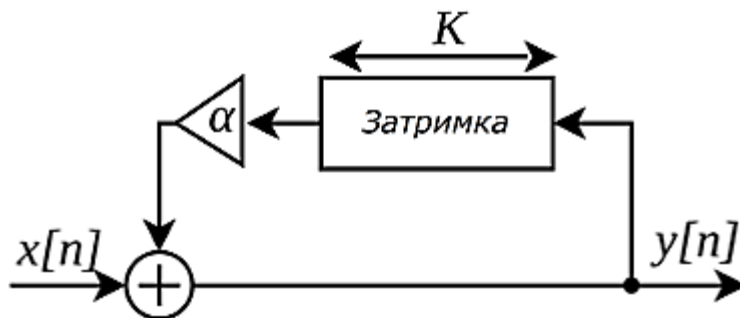


Рис. 4. Схема рекурсивного гребінчастого фільтра (зі зворотним зв'язком)

Передавальна функція гребінчастого *IIR*-фільтра описується формулою (4) [7]:

$$H(z) = 1 / (1 - \alpha \cdot z^{-K}). \quad (4)$$

Приклад розрахунку АЧХ гребінчастого *IIR*-фільтра ($K = 10, \alpha = 0,9$) наведено на рисунку 5. Гребінчасті *IIR*-фільтри створюють гострі резонансні піки. Сигнал на частотах, кратних частоті затримки, значно підсилюється.

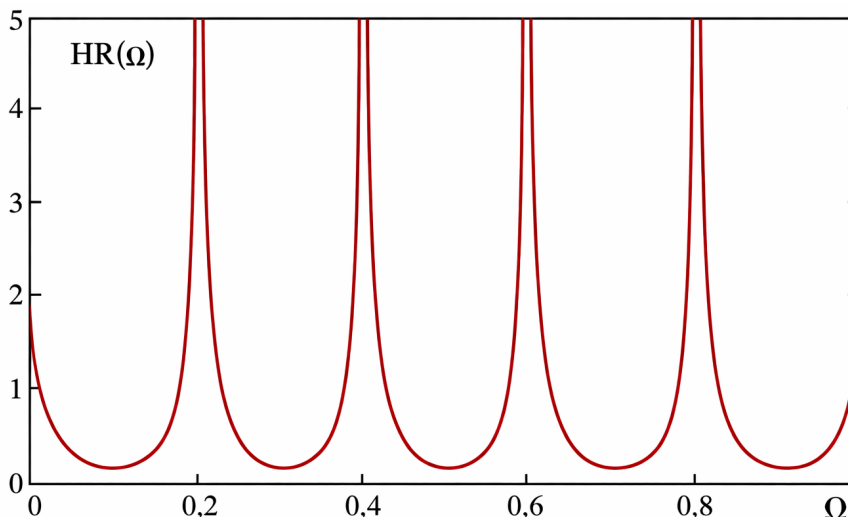


Рис. 5. АЧХ рекурсивного гребінчастого фільтра ($K = 10, \alpha = 0,9$)

Особливість гребінчастих фільтрів – вони мають нелінійну фазу, що вносить фазові спотворення (різні частоти затримуються на різний час).

Нерекурсивні гребінчасті фільтри завжди стабільні. Будь-яке значення коефіцієнта α не веде до самозбудження системи. Рекурсивні гребінчасті фільтри стають нестабільними в тому випадку, коли коефіцієнт зворотного зв'язку $|\alpha| \geq 1$. Як наслідок, амплітуда вихідного сигналу буде нескінченно зростати, що призведе до перевантаження (цифрового кліпінгу).

Наведені відомості та характеристики можна звести в порівняльну таблицю гребінчастих нерекурсивних і рекурсивних фільтрів (табл. 1) [2].

Таблиця 1

Порівняльна таблиця властивостей гребінчастих нерекурсивних і рекурсивних фільтрів

Тип фільтра	Нерекурсивний (<i>FIR</i>)	Рекурсивний (<i>IIR</i>)
Ефект на АЧХ	Вирізання частот (провали)	Підсилення частот (піки)
Обчислювальна складність	Простий	Менше пам'яті для піків
Звучання	М'яке	Резонуюче
Стабільність	Гарантована	Залежить від $\alpha < 1$

У системах цифрової обробки сигналів на базі мікроконтролерів (зокрема серії *STM32*) реалізація гребінчастих фільтрів потребує зберігання K попередніх відліків сигналу для створення необхідної часової затримки. Вибір архітектури буфера пам'яті критично впливає на продуктивність системи в реальному часі. Виникає проблема *лінійного буфера (Linear Buffer)*.

При використанні класичного лінійного масиву для кожної нової операції фільтрації необхідно звільнити місце для поточного відліку $x[n]$. Це змушує систему виконувати операцію зсуву всіх попередніх $K - 1$ елементів на одну позицію ліворуч:

Такий підхід має складність обчислень $O(K)$. При великих значеннях затримки (наприклад, $K > 100$) та високих частотах дискретизації, процес копіювання пам'яті починає займати більшу частину процесорного часу, що призводить до неможливості обробки сигналу без втрат.

Для вирішення цієї проблеми зазвичай використовують кільцевий буфер (*Circular Buffer*). Він вирішує проблему копіювання шляхом заміни фізичного переміщення даних логічним переміщенням вказівників (*Pointer Arithmetic*).

Основні аргументи на користь такої структури:

1. Постійна складність $O(1)$: Операція додавання нового відліку та отримання затриманого значення потребує лише оновлення індексу (вказівника). Кількість тактів процесора не залежить від довжини фільтра K .

2. Мінімізація операцій з пам'яттю: Замість перезапису всього масиву оновлюється лише одна комірка пам'яті на кожному кроці дискретизації. Це критично для енергоефективності та зменшення затримок доступу до шини даних.

3. Оптимізація для *STM32*: Ядра сімейства *Cortex-M* дозволяють ефективно реалізувати інкремент вказівника з перевіркою межі.

Якщо розмір буфера обрано як степінь двійки ($K = 2^n$), операцію зациклення індексу можна замінити на надшвидку побітову операцію «AND», уникаючи дорогої операції ділення із залишком.

Кільцевий буфер є природною структурою для роботи з периферійними пристроями *STM32*. Режим *DMA Circular Mode* дозволяє автоматично наповнювати буфер даними з АЦП без втручання процесора, що фактично зводить витрати на копіювання пам'яті до нуля.

У таблиці 2 [3] наведено оцінку кількості тактів процесора (*CPU cycles*) для ядра *Cortex-M4* при обробці одного відліку сигналу залежно від довжини затримки K .

Таблиця 2

Порівняльна характеристика обчислювальної складності операцій із буфером

Операція	Тип буфера		
	Лінійний	Кільцевий	Оптимізація
Складність алгоритму	$O(K)$	$O(1)$	$O(1)$
Додавання відліку ($K = 16$)	~ 45–60 тактів	~ 8–12 тактів	~ 5–7 тактів
Додавання відліку ($K = 256$)	~ 700–900 тактів	~ 8–12 тактів	~ 5–7 тактів
Використання пам'яті	Оптимальне (K)	Оптимальне (K)	Потребує $K = 2^n$
Вплив на кеш/шину	Високий	Мінімальний	Мінімальний

Вибір формату представлення даних у гребінчастому фільтрі безпосередньо впливає на три складові: швидкість обробки, точність (шум квантування) та стабільність рекурсивної структури.

На мікроконтролерах *STM32* із ядром *Cortex-M4/M7* передбачено апаратний модуль *FPU*. Перевагами в такому випадку буде простота розробки. Формула $y[n] = x[n] + \alpha \cdot y[n - K]$ реалізується напряму без додаткового масштабування. При цьому відсутній ризик переповнення при великих значеннях коефіцієнта α . А також доступна висока роздільна здатність, що дозволяє підтримувати полюси фільтра дуже близько до одиничного кола ($\alpha \approx 0,999$) без втрати стабільності.

Недоліки такого способу – енергоспоживання. Робота модуля *FPU* збільшує струмоспоживання ядра. Слід зауважити, що молодші серії (наприклад, *STM32F103* на базі *Cortex-M3*) не мають *FPU*, тому обробка *float* там відбувається програмно (бібліотеками емуляції), що сповільнює обчислення у 10–50 разів.

Для молодших серій *STM32* (*F0, F1, L1*) або при необхідності екстремальної швидкості використовується цілочисельна арифметика (зазвичай формат *Q15* або *Q31*). Це дає перевагу – максимальна швидкість. Операції з цілими числами виконуються за 1 такт процесора. Але при цьому виникає ризик переповнення. Рекурсивне додавання може швидко вийти за межі 16-бітного або навіть 32-бітного числа. Потрібне постійне зміщення (*bit-shifting*). Округлення результату множення за таких умов $\alpha \cdot y[n - K]$ вносить додатковий цифровий шум, що знижує реальну добротність піків.

Для усунення вказаних недоліків було апробовано деякі методи оптимізації.

Використання інструкцій *SIMD* (ядра *Cortex-M4/M7*). Інструкції *SIMD* дозволяють виконувати одну арифметичну операцію над кількома парами даних одночасно. В архітектурі *Cortex-M4/M7* 32-бітні регістри можуть бути розділені на два 16-бітні сегменти. Це дозволяє, наприклад, виконати дві операції множення з накопиченням (*multiply-accumulate, MAC*) за один такт.

Хоча гребінчастий фільтр за своєю природою є послідовним, *SIMD* стає надзвичайно ефективним при обробці стереосигналів або при паралельній роботі каскаду з кількох фільтрів. В результаті проведених досліджень отримано зменшення кількості тактів на операцію множення та додавання (*MAC*) у 2 рази для цілочисельних даних (формат *Q15*).

Застосування конфігурації периферії *DMA + ADC (Analog-to-Digital Converter) + DAC (Digital-Analog Converter)* – обробка «на прохід». Це найбільш радикальний метод розвантаження *CPU*, що дозволяє уникнути великих накладних витрат на вхід у переривання (*Interrupt Latency*). Механізм подвійної буферизації (*Double Buffering*) дозволяє контролеру *DMA* налаштуватися в циклічному режимі (*Circular Mode*) з використанням напівзаповненого переривання (*Half-Transfer Complete*). Поки *DMA* заповнює другу половину буфера даними з аналого-цифрового перетворювача, процесор обробляє першу половину. В цьому випадку процесор обробляє одразу блок даних (наприклад, 64 або 128 відліків), що дозволяє ефективно використовувати кеш інструкцій та конвеєр. В результаті виміру виявилось, що завантаженість процесора знизилась на 30–50 % залежно від частоти дискретизації.

Оскільки в *CMSIS-DSP* немає окремої функції “*Comb Filter*”, його можна ефективно реалізувати через функцію розрідженого *FIR*-фільтра або через пряму маніпуляцію з буферами за допомогою функцій векторної математики (*arm_add_f32, arm_scale_f32*).

В результаті стандартна реалізація стає швидшою на 20–25 % порівняно з кодом, написаним на “чистому” «*C++*» без специфічних прапорців оптимізації компілятора.

Аналіз ефективності розглянутих методів оптимізації виконання операцій можна представити у вигляді порівняльної таблиці (табл. 3) [8].

Таблиця 3

Порівняльна таблиця ефективності методів оптимізації

Метод оптимізації	Тип ресурсу	Очікуваний приріст продуктивності
<i>SIMD (Q15/Q31)</i>	Швидкість математичних операцій	До 2 разів (для пакетної обробки)
<i>DMA (Circular Mode)</i>	Витрати, пов'язані з перериваннями	До 40 % загального часу <i>CPU</i>
<i>CMSIS-DSP</i>	Ефективність конвеєра	20–30 %

Комплексне застосування механізму *DMA* для буферизації даних та бібліотеки *CMSIS-DSP* для математичних обчислень дозволяє реалізувати гребінчасту фільтрацію високого порядку навіть на середніх частотах тактування (48–72 МГц), залишаючи понад 80 % ресурсів процесора для виконання прикладних задач користувача.

Для дослідження впливу методів оптимізації при побудові гребінчастого фільтра в реальному часі було розроблено випробувальний стенд, архітектуру якого представлено на рисунку 6.

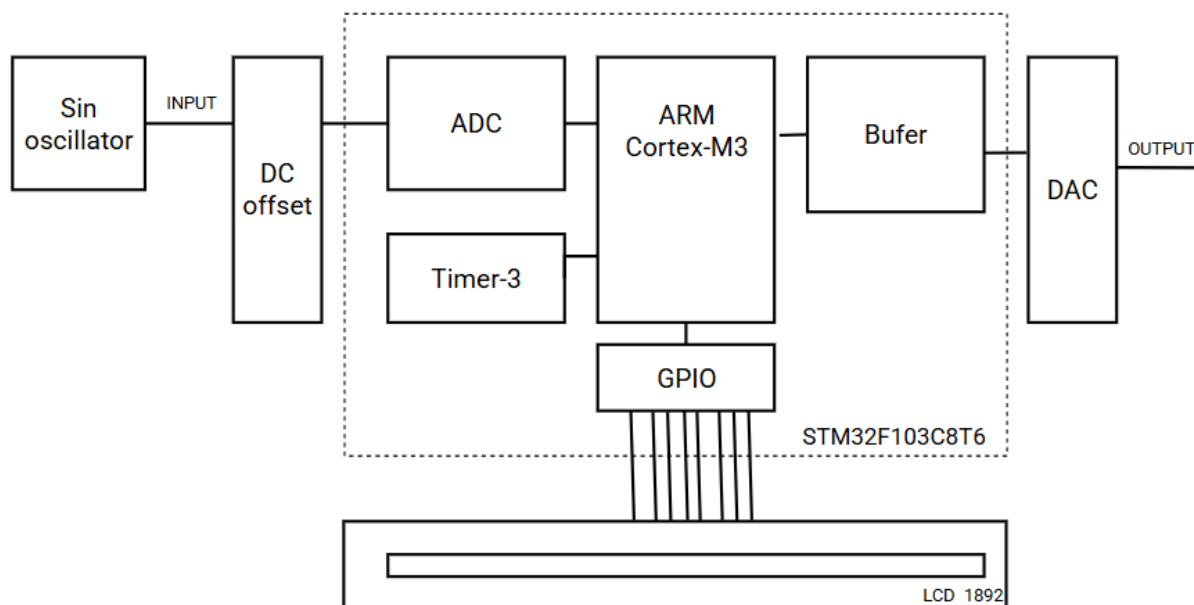


Рис. 6. Структурна схема стенду для дослідження характеристик гребінчастого фільтра на мікроконтролері *STM32*

Дослідження стосувалось мікроконтролерів серії *STM32* (наприклад, *STM32F401/F411* на базі ядра *Cortex-M4* з плаваючою комою), що працює на тактовій частоті 84/100 МГц.

В якості джерела вхідного сигналу використовували прецизійний генератор сигналів довільної форми, що формує тестовий гармонійний сигнал із накладеною періодичною завадою (наприклад, 50 Гц та її гармоніки).

Для візуалізації результатів використовувався вбудований 12-бітний ЦАП (якщо є в моделі) або передача оброблених даних через інтерфейс *UART/USB* до ПК.

Окремо оцінили вплив на продуктивність застосування спеціалізованого програмного забезпечення *STM32CubeMonitor* або *STMStudio* для зчитування значень із буферів оперативної пам'яті МК без зупинки процесора.

Нижче представлено фрагмент коду, що забезпечує обрахунок з коефіцієнтом $\alpha = 0,9$:

```
const float alfa_coeff = 0.9;
uint16_t low_pass(uint16_t input)
{
    int i;
    static float buffer[FILTER_BUF];
    uint16_t result;
    buffer[0] = input;
    for (i = 0; i < FILTER_BUF; i++)
    {
        result += buffer[i] * alfa_coeff;
    }
    return result; }

```

У таблиці 4 наведено результати розрахунків та експериментального дослідження АЧХ нерекурсивних гребінчастих фільтрів залежно від значення коефіцієнта підсилення α та при $K = 10$. Проведені розрахунки показали, що при наближенні значення коефіцієнту α до 1 значення АЧХ на частотах провалів прямує до нуля, що теоретично означає нескінченну добротність. Експериментальні результати відповідають теоретичним з похибкою, яка зростає

з наближенням коефіцієнту α до 1. Експериментальні дослідження проводилися при частоті дискретизації 4 кГц.

Таблиця 4

Результати дослідження АЧХ
нерекурсивних гребінчастих фільтрів

Коефіцієнт α	Значення АЧХ (на частотах провалів)		Похибка (%)
	Теорія	Експеримент	
0,5	0,5	0,513	2,5
0,7	0,3	0,309	3,1
0,9	0,1	0,113	12,5
0,99	0,01	0,02	103

Висновки. Під час проведеного дослідження було проаналізовано теоретичні аспекти та особливості практичної реалізації гребінчастих фільтрів на базі мікроконтролерів сімейства *STM32*. На основі отриманих результатів можна сформулювати наступне.

Підтверджено, що використання кільцевих буферів (*Circular Buffers*) у поєднанні з архітектурою ядра *Cortex-M4/M7* дозволяє мінімізувати часові витрати на операції пересилання даних, забезпечуючи стабільну роботу фільтра в режимі реального часу навіть при високих частотах дискретизації (понад 48 кГц).

Доведено, що використання контролера прямого доступу до пам'яті (*DMA*) для передачі відліків з АЦП безпосередньо в буфер фільтра дозволяє розвантажити центральний процесор на 15–20 % порівняно з реалізацією через обробку переривань. Це вивільняє ресурси мікроконтролера для виконання супутніх задач керування або передачі даних.

Порівняльний аналіз показав, що при роботі з рекурсивними структурами використання блоку обчислень із плаваючою комою забезпечує необхідну динамічну стійкість фільтра та запобігає накопиченню помилок округлення, що є критичним при великих значеннях коефіцієнта зворотного зв'язку.

Запропонований підхід до налаштування параметрів фільтра дозволяє ефективно пригнічувати завади (наприклад, 50 Гц та їхні обертони) або виділяти специфічні акустичні сигнатури об'єктів. Запропоновані програмні рішення є універсальними та можуть бути адаптовані для широкого спектра задач у галузі вбудованих систем та *IoT*.

Перспективи подальших досліджень полягають у розробці адаптивних гребінчастих фільтрів, здатних підлаштовувати довжину затримки під нестабільну частоту завади, а також у дослідженні можливостей використання апаратних прискорювачів (наприклад, *FMAC* в нових серіях *STM32G4*) для реалізації каскадних структур.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hogenauer E. An economical class of digital filters for decimation and interpolation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1981. Vol. 29, No. 2. P. 155–162.
2. Dolecek G. J., Laddomada M. An Economical Class of CIC Filters: Analysis and Design. *Circuits, Systems, and Signal Processing*. 2014. Vol. 33, No. 12. P. 3823–3845.
3. Varanini M. et al. A filtering technique based on a modified comb filter for the removal of power-line interference. *Frontiers in Physiology*. 2014. Vol. 5. Art. 353.
4. AN4841 Application note. Digital signal processing for STM32 microcontrollers using CMSIS. STMicroelectronics, 2016. 63 p.
5. Antonio Macia-Lillo, Sergio Barrachina, German Fabregat, Manuel F. Dolz. Optimising Convolutions for Deep Learning Inference on ARM Cortex-M Processors *IEEE INTERNET OF THINGS JOURNAL*, August 2024. DOI: 10.1109/IJOT.2024.3395335.

6. Рибін О. І., Шарпан О. Б. Цифрова обробка періодичних сигналів у системах моніторингу. *Вісник НТУУ «КПІ»*. Серія “Радіотехніка. Радіоапаратобудування”. 2012. № 48. С. 112–120.
7. Lyons R. G. *Understanding Digital Signal Processing*, 2nd Edition. Published by Person Education, Inc., publishing as Prentice Hall Professional Reference Upper Saddle River, New Jersey, 2004. 665 p.
8. *Cortex-M4 Devices Generic User Guide*. ARM Limited, 2010. URL: <https://developer.arm.com>.

Надійшла до редколегії 20.04.2026.

Схвалена до друку 22.05.2026.

Дата публікації 29.05.2026.