

Р.В. Петросян, ст. викладач

А.Р. Петросян, аспірант

Державний університет «Житомирська політехніка»

О.О. Шелуха, к.т.н.

Національний авіаційний університет

І.І. Сугоняк, к.т.н., доц.

Державний університет «Житомирська політехніка»

Впорядкування каскадів нерекурсивного цифрового фільтра при послідовній формі реалізації за допомогою генетичного алгоритму

У статті пропонується алгоритм мінімізації вихідної похибки КІХ-фільтра, обумовленої квантуванням результатів операцій множення, за рахунок оптимального розташування каскадів фільтра. Цифрові фільтри широко використовуються в різних галузях науки та техніки. КІХ-фільтри мають деякі переваги перед іншими типами фільтрів. Для реалізації КІХ-фільтрів часто використовують послідовну форму, яка забезпечує менші відхилення нулів передатної функції. Ефект квантування призводить до збільшення вихідної похибки фільтра, яка буде залежати від розташування каскадів. Вибір оптимального порядку розташування каскадів є складною задачею, оскільки вона належить до класу NP-повних задач. Було обрано шумову модель послідовної форми реалізації КІХ-фільтра. Відповідно до шумової моделі розроблено алгоритм впорядкування каскадів фільтра на основі генетичного алгоритму. Для розв'язання такої задачі використано генетичний алгоритм, тому що він дозволяє отримати квазіоптимальне рішення за менший час порівняно з алгоритмами, які дають точні рішення. Крім того, використання генетичного алгоритму має перевагу перед іншими евристичними методами.

У роботі запропоновано фітнес-функцію для мінімізації похибки при впорядкуванні каскадів КІХ-фільтра. Проведено детальний аналіз методів селекції, схрещування та мутації, а також здійснено вибір методів, які найбільш підходять для вирішення поставленого завдання. Виконано налаштування гіперпараметрів генетичного алгоритму для отримання більшої ефективності.

Для перевірки роботи алгоритму проведено низку експериментів. Було синтезовано декілька КІХ-фільтрів різного типу та порядку. В результаті експериментальних досліджень вдалося з'ясувати, що розроблений алгоритм дійсно має високу швидкість порівняно з алгоритмами, які дозволяють отримати точні рішення. Запропонований алгоритм більш ефективний при впорядкуванні великої кількості каскадів, тому що суттєво зменшує витрати часу на пошук розв'язання задачі.

Ключові слова: впорядкування каскадів КІХ-фільтра; послідовна форма реалізації КІХ-фільтра; проектування КІХ-фільтрів; генетичний алгоритм.

Актуальність теми. Цифрові фільтри є важливим інструментом обробки сигналів і знаходять широке застосування в різних сферах: енергетиці, обробці зображень, акустиці, телекомунікаціях, медичному устаткуванні тощо. Одним із типів цифрових фільтрів (ЦФ) є нерекурсивні ЦФ (фільтри з кінцевою імпульсною характеристикою, або скорочено – КІХ-фільтри), які мають низку переваг перед іншими типами фільтрів, наприклад, можна забезпечити лінійну фазо-частотну характеристику.

Поширеною формою реалізації цифрових фільтрів є послідовна. У цьому випадку передатна функція ЦФ, що розробляється, представляється як добуток передатних функцій декількох ЦФ (каскадів) меншого порядку, наприклад, ЦФ другого порядку. Наявність ефекту квантування призводить до того, що похибка залежить від взаємного розташування каскадів, тому виникає задача знаходження такого порядку розташування каскадів, щоб мінімізувати вихідну похибку ЦФ.

У цій статті пропонується алгоритм впорядкування каскадів КІХ-фільтра на основі генетичного алгоритму для мінімізації похибки, що виникає внаслідок квантування результатів множення (кінцевої розрядності регістрів).

Аналіз останніх досліджень та публікацій, на які спираються автори. КІХ-фільтри є одними з найпоширеніших типів фільтрів у цифровій обробці сигналів завдяки своїй простоті реалізації, лінійності, масштабованості та високій продуктивності при обробці сигналів. Вони використовуються для виділення симетричних складових у трифазній мережі [1], фільтрації механічних вібрацій у безпілотних повітряних суднах [2], видалення шуму в акустичних системах [3] тощо, тому інтерес до них постійно зростає.

На рисунку 1 представлено амплітудно-частотну характеристику (АЧХ) ідеального фільтра нижніх частот (ФНЧ). Однак при цифровій обробці сигналів завжди виникають похибки у вихідному сигналі, у тому числі при використанні ЦФ.

Це зумовлено низкою причин:

- похибка, яка обумовлена неідеальними частотними характеристиками цифрового фільтра, тобто для того ж ЦФ нижніх частот неможливо отримати АЧХ, що показано на рисунку 1;
- похибки коефіцієнтів цифрового фільтра;
- похибки квантування результатів множення у цифровому фільтрі.

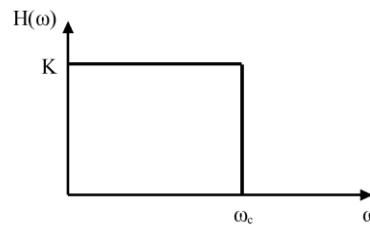


Рис. 1. АЧХ ідеального цифрового фільтра нижніх частот

Похибка, яка обумовлена неідеальними частотними характеристиками ЦФ, залежить від використаного методу проектування та порядку фільтра. Очевидно, щоб зменшити цю похибку, необхідно підвищити порядок ЦФ, який проектується, та/або змінити метод проектування. Існує безліч таких методів, серед яких можна виокремити класичні (метод зважування; метод частотної вибірки; метод найменших квадратів, метод найкращої рівномірної апроксимації) [4–6] та евристичні методи проектування КІХ-фільтрів [7–10]. Серед евристичних алгоритмів можна вирізнити генетичний алгоритм (ГА) [11, 12], який має низку переваг перед іншими оптимізаційними алгоритмами: належить до глобальних методів оптимізації; є паралельно масштабованим; може використовуватися для багатокритеріальної оптимізації тощо. У [5, 6, 13] наводиться опис та порівняльний аналіз деяких методів проектування КІХ-фільтрів.

Похибки коефіцієнтів ЦФ виникають під час проектування КІХ-фільтрів, а також через обмежену розрядність регістрів мікропроцесора або мікроконтролера [5, 6]. Від того, яка структура КІХ-фільтра була обрана під час проектування, залежить чутливість АЧХ до похибок коефіцієнтів ЦФ. Найбільш поширена структура тих фільтрів, яка має низьку чутливість АЧХ до похибок коефіцієнтів ЦФ, – це послідовна (каскадна) форма реалізації КІХ-фільтрів.

Похибки квантування результатів множення у ЦФ виникають при фільтрації вхідного сигналу у зв'язку з обмеженою розрядністю регістрів мікропроцесора або мікроконтролера [5, 6]. Від того, яка структура КІХ-фільтра обрана при проектуванні, залежить величина вихідної похибки цього фільтра, що обумовлено квантуванням результатів операцій множення. Однак при використанні послідовної форми реалізації ЦФ така похибка залежить також від порядку розташування каскадів у фільтрі.

Складність вибору оптимального порядку розташування каскадів КІХ-фільтра полягає в тому, що зі зростанням кількості каскадів фільтра суттєво збільшується кількість варіантів його реалізації, яка буде визначатися відповідно до виразу (1):

$$N = S!, \quad (1)$$

де S – кількість каскадів КІХ-фільтра.

Таким чином, можна зазначити, що задача впорядкування каскадів у послідовній формі реалізації КІХ-фільтра належить до класу NP-повних задач.

У [6] даються загальні рекомендації щодо розташування каскадів у КІХ-фільтрі, які дозволяють зменшити цю похибку. У [14] для отримання оптимального розташування каскадів ЦФ, щоб зменшити вихідну похибку, використовується метод динамічного програмування Беллмана. Цей метод дозволяє отримати оптимальне розташування каскадів фільтра, але складність такого алгоритму має експоненційний характер. Можна зауважити, що алгоритми, які дозволяють отримати оптимальне розв'язання нашої задачі, потребують тривалого часу. Для розв'язання такого класу задач часто використовують евристичні алгоритми, тому що вони дозволяють отримати квазіоптимальне рішення, проте час на пошук цього рішення значно скорочується. До таких робіт належать [15, 16], де пропонується на базі евристичних алгоритмів виконати впорядкування та поділ коефіцієнтів при реалізації структур КІХ-фільтрів з метою мінімізації енергоспоживання НВІС (надвеликих інтегральних схем).

Метою статті є розробка алгоритму наближеного розв'язання задачі впорядкування каскадів у послідовній формі реалізації КІХ-фільтра для мінімізації максимальної вихідної похибки за допомогою генетичного алгоритму.

Викладення основного матеріалу. Узагальнену схему послідовної форми реалізації ЦФ можна зобразити відповідно до рисунка 2. Будемо вважати, що ЦФ синтезований, розкладений на каскади і коефіцієнти ЦФ обмежені необхідною розрядністю, тому залишається визначити таке розташування каскадів, яке забезпечить мінімальну вихідну похибку ЦФ.

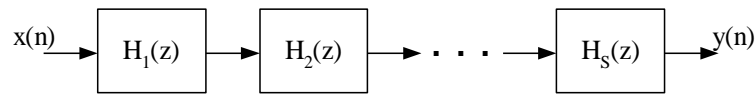


Рис. 2. Послідовна форма реалізації ЦФ ($x(n)$, $y(n)$ – вхідна і вихідна послідовності даних відповідно)

Кожний каскад є джерелом похибок квантування ε_s . У цьому випадку вплив усіх складових похибок можна показати за допомогою наступної схеми (рис. 3). В цьому випадку похибка квантування операції множення показана на виході кожного каскаду, однак в деяких джерелах ця похибку приводять до входу ЦФ.

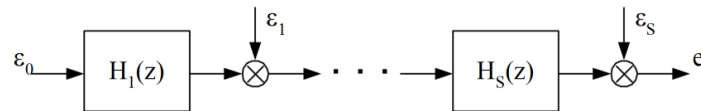


Рис. 3. Шумова модель ЦФ при послідовній реалізації

Відповідно до рисунка 3, сумарна похибка на виході кожного каскаду ЦФ буде визначатися відповідно до виразу (2):

$$e_s(z) = \sum_{i=0}^{s-1} \varepsilon_s(z) \prod_{t=i+1}^s H_t(z) + \varepsilon_s, \quad (2)$$

де $e_s(z)$ – сумарна похибка на виході S -го каскаду фільтра, при цьому можна вважати $\varepsilon_0 \equiv 0$, тому що дана похибка є вхідною для усього фільтра і не впливає на розташування каскадів; $H_t(z)$ – передатна функція кожного каскаду фільтра.

Похибка квантування на виході фільтра буде визначатися відповідно до виразу (3):

$$e(z) = \sum_{s=1}^S \varepsilon_s(z) H'_{s+1}(z), \quad (3)$$

де $\varepsilon_s(z)$ – похибка квантування кожного s -го каскаду фільтра; $H'_s(z)$ – передатна функція останніх $S-s+1$ каскадів, тобто $H'_s(z) = \prod_{u=s}^S H_u(z)$ (вважати $H'_{S+1}(z) = 1$).

Вираз (3) можна розглядати як функціонал (фітнес-функцію), що дозволяє мінімізувати вихідну похибку КІХ-фільтра, обумовлену квантуванням результату операцій множення. Використавши фітнес-функцію, можна виконати її мінімізацію за допомогою ГА.

Генетичний алгоритм – евристичний алгоритм, що є різновидом еволюційних алгоритмів, за допомогою яких розв'язуються оптимізаційні задачі з використанням методів еволюції, подібних до природного відбору в навколишній природі. Ключовим поняттям ГА є особа, яка кодує можливе розв'язання задачі. Характеризується особа хромосомою чи сукупністю хромосом. Атомарною одиницею хромосоми є ген (найчастіше кодується одним бітом). Під час вирішення завдання створюється популяція осіб. Кожна особа оцінюється ступенем пристосованості, що визначається фітнес-функцією. Таким чином, визначаються особи, які краще пристосувалися до «довкілля» (мають найкраще рішення). ГА є ітераційним, тому на кожній ітерації породжується нова популяція осіб, що має кращу пристосованість, ніж попередня. Для застосування ГА до нашої задачі необхідно обрати:

- фітнес-функцію, яка обмежує простір пошуку рішення;
- операторів генетичного алгоритму (методи селекції, схрещування, мутації);
- гіперпараметри, які мають забезпечити швидке обчислення осіб з найкращою пристосованістю, однак не повинні призводити до локального рішення.

Як фітнес-функцію (функцію пристосованості) використаємо вираз (3), який необхідно мінімізувати (4):

$$|e(z)|_{\max} \rightarrow \min. \quad (4)$$

Створення нової популяції відбувається застосуванням генетичних операторів (селекції, схрещування, мутації) до поточних осіб.

Селекція – оператор, спрямований на відбір осіб відповідно до певного критерію. Існує досить багато різних методів селекції, але більш розповсюджені такі методи: метод рулетки, метод ранжування, турнірний відбір. Кожен із методів селекції має переваги та недоліки. Недоліком методу рулетки є те, що особи з дуже малим значенням функції пристосованості занадто швидко виключаються з популяції, що може призвести до передчасної збіжності ГА. Для запобігання такому ефекту застосовується масштабування функції пристосованості. Недоліком методу ранжирування є те, що особи можуть

отримувати однаковий ранг, а це означає, що вони матимуть однакову можливість вибору, незважаючи на відмінності в їх пристосованості. До недоліків турнірного методу варто зарахувати складність вибору розміру турніру (якщо розмір турніру занадто малий, то є ймовірність того, що найкращі особини не вибиратимуться для батьківського пулу). При розв'язанні задачі використано останній метод селекції – турнірний метод із 3 особами як більш ефективний.

Схрещування – оператор, який застосовується до двох батьківських осіб. Найчастіше кожна з них ділиться на дві частини в одній і тій самій випадковій позиції гена. Утворення осіб є комбінацією першої та другої частин хромосом різних батьків. Існують й інші методи схрещування, проте при їх виборі необхідно враховувати особливості фенотипу задачі. У нашій задачі потрібно отримати порядок розташування каскадів КІХ-фільтра, тому хромосома представлятиме список номерів каскадів. Вочевидь, що більшість методів схрещування для нашої задачі не підходять, тому що номер каскаду не може повторюватися у списку. Серед методів схрещування, які можна використати для розв'язання нашої задачі, можна вирізнити: упорядковане схрещування; частково відображуване схрещування; циклічне схрещування. Аналіз показав, що суттєвих відмінностей у рішеннях немає, тому будь-який з методів може бути використаний. У нашому випадку зупинилися на упорядкованому схрещуванні.

Мутація – оператор, який здійснює зміну генів у випадкових позиціях хромосоми. Мутація призначена для зниження ймовірності оптимізації в локальному екстремумі. З урахуванням фенотипу задачі можна використовувати такі методи мутації: мутація перетасовування та мутація обміном. При розв'язанні задачі використано перший метод, який є більш ефективним, тому що здійснює значніші зміни пристосованості осіб.

Алгоритм впорядкування каскадів КІХ-фільтра з використанням ГА буде мати такий вигляд:

1. Формуємо вихідну популяцію ГА, де кожна особа представляє список розташування каскадів КІХ-фільтра;
2. Обчислюємо пристосованість кожної особи (розраховуємо вихідну похибку квантування на виході КІХ-фільтра);
3. Виконуємо селекцію осіб з найкращою пристосованістю для створення нової популяції (обираємо найбільш вдалі рішення);
4. Якщо умову зупинки виконано, повертаємо особу з найкращою пристосованістю;
5. Виконуємо операцію схрещування обраних осіб для утворення нової популяції;
6. Виконуємо операцію мутації;
7. Переходимо до пункту 2 для обробки нової популяції.

Умовою зупинення алгоритму може бути як формування певної кількості поколінь, так і несуттєві зміни значень функції пристосованості.

Для перевірки роботи алгоритму проведено низку експериментів. Було синтезовано декілька КІХ-фільтрів різного типу та порядку. Деякі фільтри розглянуто далі й результати роботи алгоритму будуть порівнюватися з методом повного перебору (brute force).

Модельовання проводилося за допомогою мови програмування Python [17]. Для реалізації ГА необхідно налаштувати гіперпараметри. У нашому випадку вони матимуть такий вигляд:

```
POPULATION_SIZE = 10 # кількість осіб у популяції
P_CROSSOVER = 0.9 # ймовірність схрещування особи
P_MUTATION = 0.1 # ймовірність мутації особи
MAX_GENERATIONS = 50 # максимальна кількість поколінь
```

На рисунку 4 представлено АЧХ КІХ-фільтра 8-го порядку та результати впорядкування його каскадів за допомогою ГА.

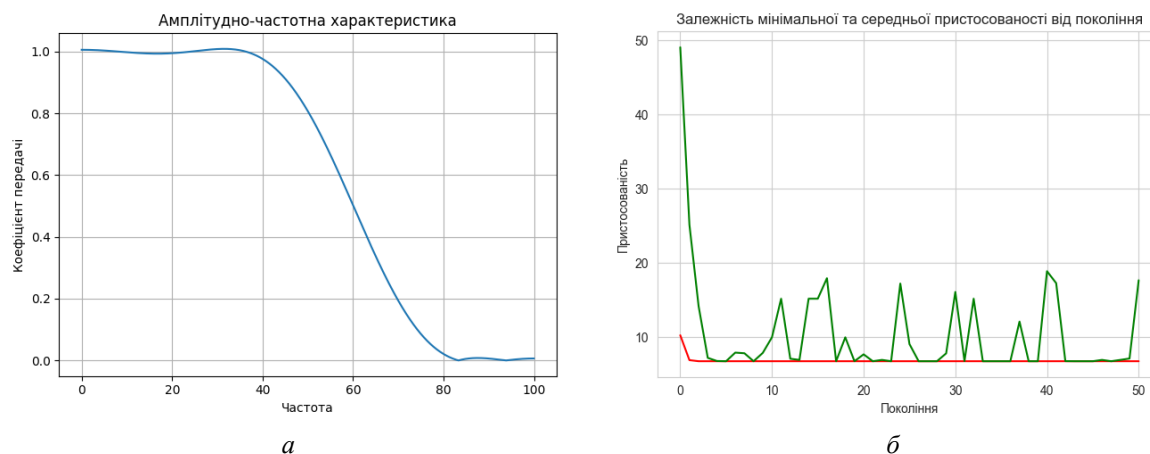


Рис. 4. ФНЧ 8 порядку: а – АЧХ; б – мінімальна та середня пристосованість залежно від покоління

Фільтр складатиметься з 4-х каскадів з такими коефіцієнтами: 1. [0.02827533, -0.12520231, 0.21901412]; 2. [1, 1.73074291, 1]; 3. [1, 1.96213129, 1]; 4. [1, -0.57166319, 0.12910277].

У таблиці 1 виконано повний перебір можливих варіантів розташування каскадів і розраховано сумарну похибку квантування на виході фільтра при такому розташуванні каскадів.

Таблиця 1
Залежність похибки квантування на виході фільтра від порядку розташування каскадів

№ з/п	Порядок каскадів фільтра	Похибка квантування на виході фільтра, Δ
1	0, 1, 2, 3	85,86
2	0, 1, 3, 2	91,084
3	0, 2, 1, 3	85,466
4	0, 2, 3, 1	90,227
5	0, 3, 1, 2	109,946
6	0, 3, 2, 1	109,715
7	1, 0, 2, 3	26,427
8	1, 0, 3, 2	31,65
9	1, 2, 0, 3	10,246
10	1, 2, 3, 0	6,889
11	1, 3, 0, 2	18,348
12	1, 3, 2, 0	8,834
13	2, 0, 1, 3	25,1
14	2, 0, 3, 1	29,86
15	2, 1, 0, 3	10,1
16	2, 1, 3, 0	6,742
17	2, 3, 0, 1	17,335
18	2, 3, 1, 0	8,515
19	3, 0, 1, 2	60,319
20	3, 0, 2, 1	60,088
21	3, 1, 0, 2	25,374
22	3, 1, 2, 0	15,86
23	3, 2, 0, 1	24,594
24	3, 2, 1, 0	15,774

Як видно з таблиці 1, значення похибок може відрізнятись більш ніж на порядок навіть при 4-х каскадах у фільтрі. Мінімальне значення відповідає розташуванню каскадів (2, 1, 3, 0) і дорівнює значенню 6,74 Δ .

У таблиці 2 наведено результати пошуку оптимального розташування каскадів КІХ-фільтра 8-го порядку з використанням ГА.

Таблиця 2
Залежність мінімальної та середньої пристосованості від покоління для КІХ-фільтра 8-го порядку

Покоління	Мінімальна пристосованість, Δ	Середня пристосованість, Δ
0	10,246	49,098
1	6,889	25,156
2	6,742	14,182
3	6,742	7,195
4	6,742	6,771
...
46	6,742	6,914
47	6,742	6,742
48	6,742	6,919
49	6,742	7,107
50	6,742	17,667

Як видно з рисунка 4, б і відповідної таблиці 2, результат був знайдений після 3-х поколінь. І хоча середня пристосованість поколінь змінювалася для пошуку кращого рішення, проте мінімальна пристосованість залишалася незмінною. Очевидно, це і є найкраще рішення, яке збіглося з методом повного перебору. Час пошуку ГА співпорівняно з методом повного перебору для 4 каскадів. Для кращої апроксимації вихідного фільтра збільшимо порядок КІХ-фільтра до 12 (рис. 5, а). Цей фільтр складатиметься з 6 каскадів з такими коефіцієнтами: 1. [-0.0090766, 0.02517961, -0.04670202]; 2. [1, -1.37608242, -2.63768746]; 3. [1, 1.90770586, 1]; 4. [1, 1.69782604, 1]; 5. [1, -0.53915458, 0.19435138]; 6 [1, 0.52170033, -0.37911997].

Результати перебору не представлятимемо у вигляді таблиці, тому що кількість варіантів становитимемо вже 720. Обмежимося лише кінцевим результатом. Значення похибок у цьому випадку може відрізнятися більш ніж на два порядки. Мінімальне значення відповідає розташуванню каскадів (2, 3, 1, 5, 4, 0) і дорівнює значенню 10,259Δ.

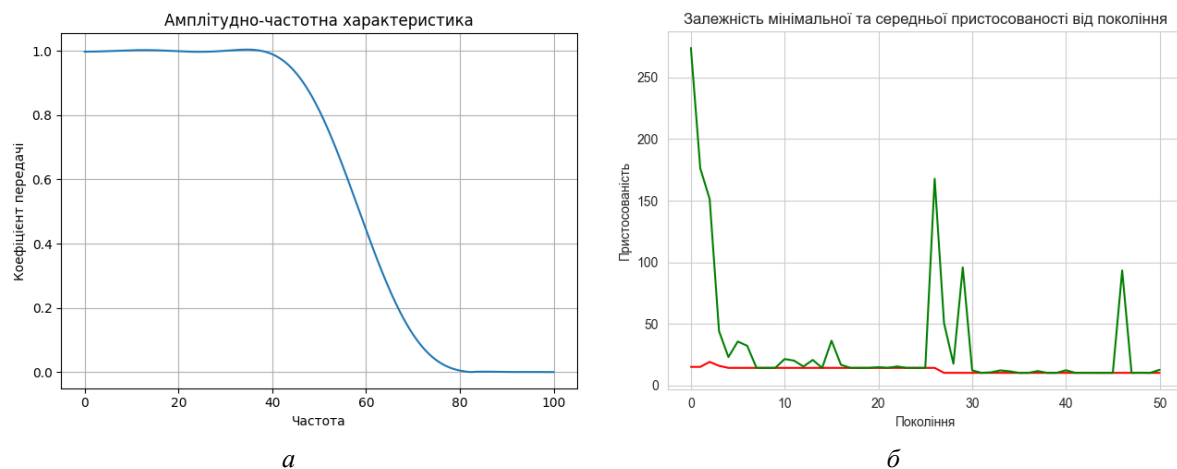


Рис. 5. ФНЧ 12 порядку: а – АЧХ; б – мінімальна та середня пристосованість залежно від покоління

У таблиці 3 наведено результати пошуку оптимального розташування каскадів КІХ-фільтра 12-го порядку з використанням ГА.

Таблиця 3

Залежність мінімальної та середньої пристосованості від покоління для КІХ-фільтра 12-го порядку

Покоління	Мінімальна пристосованість, Δ	Середня пристосованість, Δ
0	15,108	274,038
1	15,108	176,236
2	19,164	151,384
3	15,964	44,12
4	14,298	23,069
...
26	14,298	167,841
27	10,259	50,6
...
49	10,259	10,259
50	10,259	12,778

Як видно з рисунка 5, б і відповідної таблиці 3, результат був знайдений після 27 поколінь. Однак досить вдале рішення було знайдено вже після 4 поколінь, а оптимальніше рішення було знайдено після мутації 26 покоління. Це є найкраще рішення, яке повністю збіглося з методом повного перебору. Видно, що час пошуку ГА виявився в 2 рази меншим за метод повного перебору для 6 каскадів.

Розглянемо ще один варіант. Збільшимо порядок КІХ-фільтра до 20 (рис. 6, а). Цей фільтр складатиметься з 10 каскадів з такими коефіцієнтами: 1. [6.32109790e-04, -3.17828355e-03, 4.23190299e-03]; 2. [1, -3.47495702, 5.62408813]; 3. [1, -1.06171835, 4.05576043]; 4. [1, 1.77252184, 1]; 5. [1, 1.98892234, 1]; 6. [1, 1.6528648, 1]; 7. [1, 1.90612165, 1]; 8. [1, -2.61780342e-01, 2.46562887e-01]; 9. [1, -6.17870300e-01, 1.77806602e-01]; 10. [1, -7.51029398e-01, 1.49367741e-01].

В результаті повного перебору було отримано результат, проте час перебору склав 7 хв. Мінімальне значення відповідає значенню 130,214Δ. Результат пошуку оптимального розташування каскадів КІХ-фільтра 20-го порядку з використанням ГА видно з рисунка 6, б. Досить вдале рішення було знайдено після 20 поколінь (133,405Δ), а оптимальніше рішення було знайдено після 38 поколінь (130,826Δ). Це не є найкраще рішення, тому що не збігається з методом повного перебору і відповідає розташуванню каскадів (1, 4, 6, 2, 9, 3, 7, 5, 8, 0). Різниця у похибці між варіантами дуже мала, однак час пошуку рішення суттєво відрізняється (час пошуку ГА склав менше секунди).

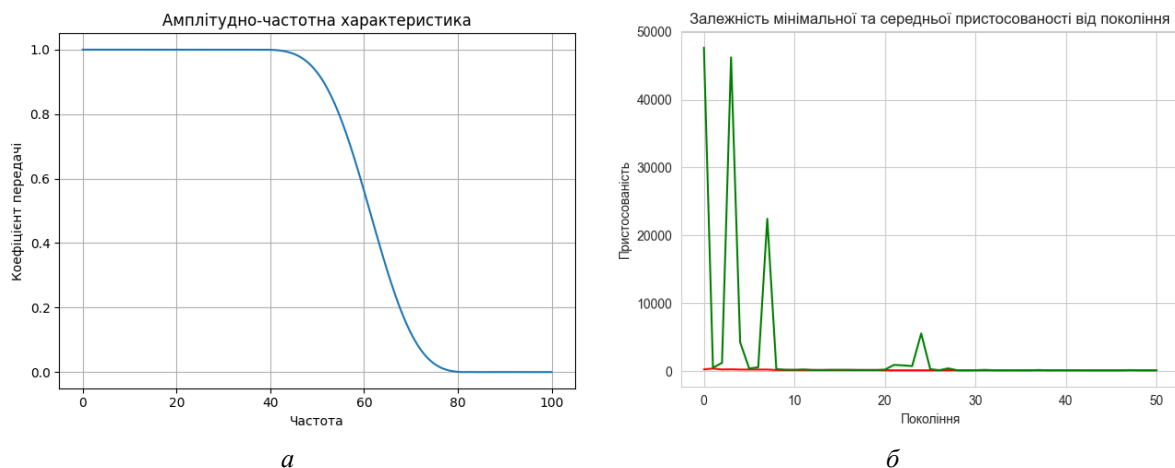


Рис. 6. ФНЧ 20 порядку: а – АЧХ; б – мінімальна та середня пристосованість залежно від покоління

Висновки та перспективи подальших досліджень. У статті наведено алгоритм мінімізації вихідної похибки КІХ-фільтра, обумовленої квантуванням результатів операцій множення, за рахунок оптимального розташування каскадів ЦФ.

Для виконання поставленого завдання використано апарат цифрової обробки сигналів та еволюційних алгоритмів. У ході аналізу було обрано шумову модель послідовної реалізації КІХ-фільтра. Відповідно до шумової моделі розроблено алгоритм впорядкування каскадів фільтра на основі ГА, який мінімізує сумарну похибку квантування на виході фільтра. Для розв'язання задачі оптимізації було обрано: турнірний метод селекції з трьома особами; метод упорядкованого схрещування; метод мутації на базі перетасовування.

В результаті експериментальних досліджень вдалося з'ясувати, що розроблений алгоритм має високу швидкодію порівняно з алгоритмами, які дозволяють отримати точні рішення. Запропонований алгоритм більш ефективний при впорядкуванні великої кількості каскадів (6 та більше), тому що суттєво зменшує витрати часу на пошук розв'язання задачі (можливо доведеться збільшити розмір популяції і кількість поколінь, але аналіз показав, що до 20 каскадів впорядкування цього можна не робити).

Список використаної літератури:

1. Петросян Р.В. Синтез цифрових фільтрів симетричних складових на базі нерекурсивних цифрових фільтрів з лінійною фазою / Р.В. Петросян // Вісник ЖДТУ. Сер. : Технічні науки. – 2007. – № 2 (41). – С. 114–118 [Електронний ресурс]. – Режим доступу : <https://vtn.ztu.edu.ua/article/view/93626/90686>.
2. Betaflight 4.3 Tuning Notes // Betaflight [Electronic resource]. – Access mode : <https://betaflight.com/docs/tuning/4.3-Tuning-Notes>.
3. Kavita Baghel Remove Noise From Input Signal by Using IIR and Fir Filter Combined Circuit / Kavita Baghel, Brijendra Mishra // International Journal of Advanced Research in Science, Communication and Technology. – 2023. – Vol. 3. – P. 199–203. DOI: 10.48175/ijarsct-7866.
4. Gazi O. Understanding Digital Signal Processing / O.Gazi. – Singapore : Springer, 2018. DOI: 10.1007/978-981-10-4962-0.
5. Manolakis D.G. Digital signal processing / D.G. Manolakis, J.G. Proakis // New International Edition. – Pearson Higher Ed, 2013.
6. Rabiner L.R. Theory and application of digital signal processing / L.R. Rabiner, B.Gold // Prentice-Hall, Englewood Cliffs. – N.J. – 1975.
7. Petrosian R. Method for calculating the FIR filter based on genetic algorithm / R.Petrosian, O.Kuzmenko, A.Petrosian // Computer Systems and Information Technologies. – 2021. – Vol. 1. – P. 19–24 [Electronic resource]. – Access mode : <https://csitjournal.khmnu.edu.ua/index.php/csit/article/view/45/33>.
8. Thakare V.V. Genetic Algorithm for Optimization of MSE & Ripples in Linear Phase Low Pass FIR Filter & Also Compare with Cosine Window Techniques / V.V. Thakare, K.A. Sahur // International Journals Digital Communication and Analog Signals. – 2015. – Vol. 1, № 1. – P. 9–15. DOI: 10.37628/jdcas.v1i1.42.
9. Optimal design of digital FIR filters based on back propagation neural network / Yang J. et al. // IEICE Electronics Express. – 2023. – Vol. 20, № 1. – P. 20220491–20220491. DOI: 10.1587/elex.19.20220491.
10. Petrosian R. Development of a method for synthesis the FIR filters with a cascade structure based on genetic algorithm / R.Petrosian, V.Chukhov, A.Petrosian // Technology audit and production reserves. – 2021. – Vol. 4, № 2 (60). – P. 6–11. DOI: 10.15587/2706-5448.2021.237271.
11. Mutingi M. Grouping genetic algorithms: Advances and Applications / M.Mutingi, C.Mbohwa. – Switzerland : Springer International Publishing. – 2017. – 243 p.
12. Runwei Cheng J. Parallel Genetic Algorithms with GPU Computing / J.Runwei Cheng, M.Gen // Industry 4.0 – Impact on Intelligent Logistics and Manufacturing. – 2020. DOI: 10.5772/intechopen.89152.

13. Kaur S. Different Design Approaches for the Optimization of FIR Filter Coefficients / S.Kaur, B.Singh, M.Singh // Research Cell: An International Journal Of Engineering Sciences. – 2016. – Vol. 17. – P. 229–236 [Electronic resource]. – Access mode : <http://ijoes.vidyapublications.com/paper/Vol17/32-Vol17.pdf>.
14. Петросян Р.В. Впорядковування каскадів цифрових фільтрів при каскадній реалізації / Р.В. Петросян // Вісник ЖДТУ. Сер. : Технічні науки. – 2008. – № 2 (45). – С. 90–92 [Електронний ресурс]. – Режим доступу : <http://vtn.ztu.edu.ua/article/view/81607/79196>.
15. A.G. da Luz Ordering and partitioning of coefficients based on heuristic algorithms for low power FIR filter realization / A.G. da Luz, E.A.C. da Costa, M.S. de Aguiar // Proceedings of the 23rd symposium on Integrated circuits and system design. – 2010. – P. 180–185. DOI: 10.1145/1854153.1854198.
16. A.G. da Luz Exploring the use of heuristic-based algorithms for the ordering and partitioning of coefficients for power efficient fir filters realization / A.G. da Luz, E.A.C. da Costa, M.S. de Aguiar // Proceedings of the 24th symposium (João Pessoa, Brazil, 30 August – 2 September 2011). – New York, USA, 2011. – P. 180–185. DOI: 10.1145/2020876.2020898.
17. Downey A. Think DSP: digital signal processing in Python / A.Downey. – O'Reilly Media, Inc., 2016. – 168 p.

References:

1. Petrosian, R.V. (2007), «Syntez tsyfrovyykh filtriv symetrychnykh skladovykh na bazi nerekursyvnykh tsyfrovyykh filtriv z liniinoiu fazoiu», *Visnyk ZhDTU. Ser. Tekhnichni nauky*, No. 2 (41), pp. 114–118, [Online], available at: <https://vtn.ztu.edu.ua/article/view/93626/90686>
2. «Betaflight 4.3 Tuning Notes», *Betaflight*, [Online], available at: <https://betaflight.com/docs/tuning/4.3-Tuning-Notes>
3. Kavita, Baghel and Brijendra, Mishra (2023), «Remove Noise From Input Signal by Using IIR and Fir Filter Combined Circuit», *International Journal of Advanced Research in Science, Communication and Technology*, Vol. 3, pp. 199–203, doi: 10.48175/ijarsct-7866.
4. Gazi, O. (2018), *Understanding Digital Signal Processing*, Springer, Singapore, doi: 10.1007/978-981-10-4962-0.
5. Manolakis, D.G. and Proakis, J.G. (2013), «Digital signal processing», *New International Edition. Pearson Higher Ed.*
6. Rabiner, L.R. and Gold, B. (1975), «Theory and application of digital signal processing», *Prentice-Hall, Englewood Cliffs*, N.J.
7. Petrosian, R., Kuzmenko, O. and Petrosian, A. (2021), «Method for calculating the FIR filter based on genetic algorithm», *Computer Systems and Information Technologies*, Vol. 1, pp. 19–24, [Online], available at: <https://csitjournal.khmnu.edu.ua/index.php/csit/article/view/45/33>
8. Thakare, V.V. and Sahur, K.A. (2015), «Genetic Algorithm for Optimization of MSE & Ripples in Linear Phase Low Pass FIR Filter & Also Compare with Cosine Window Techniques», *International Journals Digital Communication and Analog Signals*, Vol. 1, No. 1, pp. 9–15, doi: 10.37628/jdcas.v1i1.42.
9. Yang, J. et al. (2023), «Optimal design of digital FIR filters based on back propagation neural network», *IEICE Electronics Express*, Vol. 20, No. 1, pp. 20220491–20220491, doi: 10.1587/elex.19.20220491.
10. Petrosian, R., Chukhov, V. and Petrosian, A. (2021), «Development of a method for synthesis the FIR filters with a cascade structure based on genetic algorithm», *Technology audit and production reserves*, Vol. 4, No. 2 (60), pp. 6–11, doi: 10.15587/2706-5448.2021.237271.
11. Mutingi, M. and Mbohwa, C. (2017), «Grouping genetic algorithms: Advances and Applications», Springer International Publishing, Switzerland, 243 p.
12. Runwei Cheng, J. and Gen, M. (2020), «Parallel Genetic Algorithms with GPU Computing», *Industry 4.0 – Impact on Intelligent Logistics and Manufacturing*, doi: 10.5772/intechopen.89152.
13. Kaur, S., Singh, B. and Singh, M. (2016), «Different Design Approaches for the Optimization of FIR Filter Coefficients», *Research Cell: An International Journal Of Engineering Sciences*, Vol. 17, pp. 229–236, [Online], available at: <http://ijoes.vidyapublications.com/paper/Vol17/32-Vol17.pdf>
14. Petrosian, R.V. (2008), «Vporiadkovuvannya kaskadiv tsyfrovyykh filtriv pry kaskadnii realizatsii», *Visnyk ZhDTU. Ser. Tekhnichni nauky*, No. 2 (45), pp. 90–92, [Online], available at: <http://vtn.ztu.edu.ua/article/view/81607/79196>.
15. A.G. da Luz, E.A.C. da Costa and M.S. de Aguiar (2010), «Ordering and partitioning of coefficients based on heuristic algorithms for low power FIR filter realization», *Proceedings of the 23rd symposium on Integrated circuits and system design*, pp. 180–185, doi: 10.1145/1854153.1854198.
16. A.G. da Luz, E.A.C. da Costa and M.S. de Aguiar (2011), «Exploring the use of heuristic-based algorithms for the ordering and partitioning of coefficients for power efficient fir filters realization», *Proceedings of the 24th symposium (João Pessoa, Brazil, 30 August – 2 September 2011)*, USA, New York, pp. 180–185, doi: 10.1145/2020876.2020898.
17. Downey, A. (2016), *Think DSP: digital signal processing in Python*, O'Reilly Media, Inc., 168 p.

Петросян Руслан Валерікович – старший викладач кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

<http://orcid.org/0000-0002-0388-8821>.

Наукові інтереси:

- цифрова обробка;
- штучний інтелект;
- математичне моделювання;
- комп'ютерні системи спеціального призначення;
- вебтехнології.

Петросян Арсен Русланович – аспірант кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

<http://orcid.org/0000-0003-0960-8461>.

Наукові інтереси:

- комп'ютерні науки;
- інтернет речей;
- безпілотні повітряні судна;
- вебпрограмування та вебтехнології;
- олімпіадне програмування.

Шелуха Олексій Олегович – кандидат технічних наук, доцент кафедри комп'ютеризованих електротехнічних систем та технологій Національного авіаційного університету.

<https://orcid.org/0000-0002-6088-8262>.

Наукові інтереси:

- комп'ютерні науки;
- відновлювана та цифрова енергетика.

Сугоняк Інна Іванівна – доцент, кандидат технічних наук, доцент кафедри комп'ютерних наук Державного університету «Житомирська політехніка».

<https://orcid.org/0000-0002-0484-4839>.

Наукові інтереси:

- проектування сховищ даних;
- моделювання та інтелектуальний аналіз даних.

Petrosian R.V., Petrosian A.R., Shelukha O.O., Suhoniak I.I.

Ordering the cascades of a nonrecursive digital filter in a sequential form of implementation using a genetic algorithm

The paper proposes an algorithm for minimizing the output error of a FIR filter caused by quantization of the results of multiplication operations by optimizing the arrangement of filter cascades. Digital filters are widely used in various fields of science and technology. FIR filters have some advantages over other types of filters. To implement FIR filters, a sequential form is often used, which provides smaller deviations of the zeros of the transfer function. The quantization effect leads to an increase in the output error of the filter, which will depend on the arrangement of the cascades. Choosing the optimal order of the cascades is a difficult challenge, as it belongs to the class of NP-complete problems. The noise model of the sequential form of the FIR filter implementation was chosen. In accordance with the noise model, an algorithm for ordering the filter cascades based on a genetic algorithm was developed. To solve this problem, a genetic algorithm was used because it allows us to obtain a quasi-optimal solution in less time compared to algorithms that allow us to obtain exact solutions. In addition, the use of a genetic algorithm has an advantage over other heuristic methods.

In this paper, a fitness function is proposed to minimize the error in the ordering of FIR filter cascades. A detailed analysis of the methods of selection, crossing, and mutation is carried out, and the methods that are most suitable for solving the problem are selected. The hyperparameters of the genetic algorithm were adjusted to obtain greater efficiency.

A number of experiments were conducted to test the algorithm. Several FIR filters of different types and orders were synthesized. As a result of experimental studies, it was found that the developed algorithm really has high performance compared to algorithms that allow obtaining exact solutions. The proposed algorithm is more effective in ordering a large number of cascades, as it significantly reduces the time spent on finding a solution to the problem.

Keywords: ordering of FIR filter cascades; sequential form of FIR filter implementation; FIR filter design; genetic algorithm.

Стаття надійшла до редакції 17.04.2023.