

РОЗРОБКА АЛГОРИТМУ ЦИФРОВОЇ ФІЛЬТРАЦІЇ ВХІДНОЇ ІНФОРМАЦІЇ РОБОТОТЕХНІЧНОЇ СИСТЕМИ

Анотація: У ряді областей техніки форму сигналів пов'язують з об'єктом дослідження, прикладом цього служать радіолокація, технічна та медична діагностика, телеметрія та ін. Як правило, тут мають місце нестационарні випадкові сигнали малої тривалості у часі. В результаті обробки таких сигналів, наприклад, за допомогою лінійного цифрового фільтра, їх форма, а, отже, діагностичні ознаки що містяться в ньому можуть бути сильно спотворені. У зв'язку з цим особливої актуальності набуває розробка алгоритмів цифрової фільтрації сигналів, спрямованих на збереження їх первісної форми.

Метою даної роботи є розробка алгоритму цифрової фільтрації сигналів на основі математичного апарату швидкого перетворення Фур'є (ШПФ). Для досягнення поставленої мети в роботі був проведений аналіз існуючих способів цифрової фільтрації сигналів та здійснено синтез алгоритму цифрової фільтрації на основі математичного апарату ШПФ. В результаті отриманих досліджень було визначено, що для збільшення швидкодії ШПФ необхідно організувати таким чином, щоб на кожній стадії дані для обробки можна було б зчитувати з пам'яті послідовно. Для цього на попередній стадії результати обчислень потрібно зберігати в необхідному порядку. Розглянутий алгоритм був реалізований на процесорі ARM Cortex-M3.

Ключові слова: цифро-аналоговий перетворювач, цифрова обробка сигналів, швидке перетворення Фур'є, мікроконтролер ARM Cortex-M3.

Опис проблеми та постановка задачі

У ряді областей техніки форму сигналів пов'язують з об'єктом дослідження, прикладом цього служать радіолокація, технічна та медична діагностика, телеметрія та ін. Як правило, тут мають місце нестационарні випадкові сигнали малої тривалості у часі. В результаті обробки таких сигналів, наприклад, за допомогою лінійного цифрового фільтра, їх форма, а, отже, діагностичні ознаки що містяться в ньому можуть бути сильно спотворені. У зв'язку з цим особливої актуальності набуває розробка алгоритмів цифрової фільтрації сигналів, спрямованих на збереження їх первісної форми.

Якщо для цифрової фільтрації зі збереженням форми сигналів використовуються адаптивні алгоритми (адаптивні цифрові фільтри), то для них також виникає ряд складнощів, так як метою застосування алгоритму адаптивної фільтрації сигналів є досягнення локального або глобального екстремуму функціоналу якості. У задачі

Міжвідомчий науково-технічний збірник «Адаптивні системи автоматичного управління» № 2' (33) 2018

збереження вихідної форми сигналу під функціоналом якості розуміється залежність значень середнього квадрата помилки від параметрів адаптації цифрового фільтра. Якщо статистичні властивості сигналів змінюються в часі, то функціонал якості можна вважати «розмитим» або нечітким. В цьому випадку процес адаптації полягає не тільки в русі до точки екстремуму, але й в стеженні за цією точкою, оскільки вона змінює своє місце розташування в просторі. У розглянутих умовах використання адаптивних алгоритмів на основі принципів оптимальної лінійної фільтрації є неефективним і нерациональним з точки зору обчислювальних витрат.

Таким чином, для вирішення задач цифрової фільтрації зі збереженням форми сигналів особливої актуальності набуває розробка альтернативних алгоритмів цифрової фільтрації сигналів, на основі математичного апарату швидкого перетворення Фур'є.

Метою роботи є розробка алгоритму цифрової фільтрації сигналів на основі математичного апарату швидкого перетворення Фур'є.

Для досягнення поставленої мети у роботі необхідно виконати низку завдань:

- виконати аналіз існуючих способів цифрової фільтрації;
- здійснити синтез алгоритму цифрової фільтрації на основі математичного апарату швидкого перетворення Фур'є;
- запропонувати програмну реалізацію алгоритму;
- навести пропозиції щодо технічної реалізації синтезованого алгоритму цифрової фільтрації.

Об'єктом роботи є алгоритм цифрової фільтрації сигналів.

Предметом є процес розробки алгоритму цифрової фільтрації сигналів на основі мат. апарату швидкого перетворення Фур'є.

Основна частина

Тенденція розвитку мікропроцесорних пристроїв в даний час носить експонентний характер. Використання мікропроцесорів дозволяє не лише зменшити масогабаритні показники, а й істотно поліпшити технічні характеристики, в тому числі швидкодію пристроїв. Так, наприклад, реалізація частотних фільтрів на мікропроцесорних пристроях дозволяє забезпечити сталість характеристик, прецизійне налаштування і уникнути негативних впливів старіння аналогових елементів. Сучасні мікропроцесорні пристрої являють собою сукупність декількох спеціалізованих інтегральних схем, які фактично виконують функції процесора ЕОМ.

Реалізація алгоритмів цифрової обробки сигналів (ЦОС) зводиться до написання і налагодження програм на мові *C* або асемблері. Реалізація програм на мові асемблер дозволяє створювати компактні, досить ефективні програми, але вимагає докладного вивчення архітектурних особливостей використовуваного

мікропроцесора. Такі програми, як правило, «економічно» використовують пам'ять мікропроцесора. Реалізація програм на асемблері дозволяє використовувати всі можливості мікропроцесора при жорстких умовах обмеження розміру пам'яті та необхідної швидкодії.

Проведений аналіз літератури показує, що написання програм для мікропроцесорних пристроїв найчастіше здійснюється на мові C. Ця закономірність очевидна. Мова високого рівня C дозволяє полегшити процес створення і налагодження програм. Створені програми при цьому не відрізняються компактністю коду, ефективністю виконання і мінімізацією використання пам'яті. Цю обставину необхідно враховувати при розробці програм призначених для ЦГЗ в режимі реального часу.

Серед мікропроцесорів для вбудованих систем одними з найперспективніших і швидкодіючих є мікропроцесори побудовані на ядрі ARM. Серед цих пристроїв лідируюче місце займають 32-розрядні мікропроцесори Cortex-M3 [1]. Вони виділяються оптимальною ціною і продуктивністю. На випуску цих мікропроцесорів спеціалізуються багато відомих фірм, такі як *Atmel*, *Texas Instruments*, *ST Microelectronics*, *NXP*, та ін. Перелічимо основні переваги мікроконтролерів ARM Cortex-M3 [1, 2].

1. Повністю 32-бітна архітектура:

- всі регістри 32-бітові, арифметичні операції працюють з 32-бітними даними;
- операція множення 32-розрядних чисел виконується за 1 такт, розподіл за 2-12 тактів.

2. Вбудований контролер переривань (*Nested Vectored Interrupt Controller (NVIC)*) підтримує до 240 переривань і до 256 пріоритетів, при цьому забезпечує високошвидкісну реакцію на переривання.

3. Повноцінна підтримка і реалізація прямого доступу до пам'яті (*Direct Memory Access*).

4. Спеціалізований набір інструкцій *Thumb-2* [2] дозволяє створювати досить компактний виконуваний код. Розроблений набір інструкцій орієнтований на роботу з популярним компілятором мови високого рівня C.

Розглянуті мікроконтролери Cortex-M3 знаходять широке застосування в пристроях ЦОС. Cortex-M3 мають продуктивність в десятки MIPS, низьке споживання енергії і, як правило, наявність різноманітних периферійних пристроїв, виконаних на кристалі, що робить даний тип мікроконтролерів підходящим засобом для реалізації цифрової обробки сигналів в недорогих пристроях.

Завдання визначення спектрального складу – амплітуд, частот і фаз гармонічних складових складного сигналу зустрічається при аналізі багаточастотних коливань в системах управління, при розпізнаванні багаточастотних сигналів

Міжвідомчий науково-технічний збірник «Адаптивні системи автоматичного управління» № 2' (33) 2018 в лініях зв'язку і т.п. Для вирішення цього завдання найчастіше застосовується дискретне перетворення Фур'є (ДПФ).

Алгоритм швидкого перетворення Фур'є (ШПФ) є одним з найбільш часто використовуваних при цифровій обробці сигналів. У тому чи іншому варіанті він застосовується при гармонійному аналізі, кодуванні різного роду мультимедіа інформації, систем цифрової діагностики, фільтрації сигналів і для вирішення інших завдань, в зв'язку з чим велике значення має ефективна реалізація цього алгоритму для різних цифрових і універсальних процесорів.

Під назвою ШПФ ховається велика кількість різних алгоритмів, що ДПФ, кожен з яких має свої переваги.

Алгоритм винограду (*Winograd*) проводить перетворення з мінімальною кількістю множень, а алгоритм Джонсона-Баррасса (*Johnson-Burrus*) має мінімальну сумарну кількість операцій додавання і множення. Найбільше практичне поширення набули алгоритми Кулі-Тьюки (*Cooley-Tukey*), які мають зручну структуру для програмної реалізації [3].

Алгоритми Кулі-Тьюки дозволяють організувати обчислювальний процес таким чином, щоб на кожній стадії обчислень результати розміщувалися в тих же осередках пам'яті, що і вхідні дані (так звані обчислення «на місці», *in-place*). Ця схема виявилася зручною в реалізації і набула широкого поширення.

Разом з тим обчислення ШПФ «на місці» має низку недоліків. Основний недолік пов'язаний з тим, що на кожній стадії дані з пам'яті потрібно зчитувати не послідовно, а в певному порядку. У деяких обчислювальних системах послідовне зчитування даних з пам'яті виконується швидше, ніж при довільному доступі до пам'яті; крім того, для довільного доступу потрібне використання адресної арифметики.

Для збільшення швидкодії ШПФ запропонована наступна ідея: обчислювальний процес потрібно організувати таким чином, щоб на кожній стадії дані для обробки можна було б зчитувати з пам'яті послідовно. Для цього на попередній стадії результати обчислень потрібно зберігати в необхідному порядку.

При реалізації цієї ідеї дані на різних стадіях обчислень будуть розташовані по-різному, тобто організувати обчислення «на місці» при даному підході не вийде. Тому для зберігання даних потрібно використовувати два буфера. У поточній стадії один з них використовується для вхідних даних, інший – для вихідних, на наступній стадії вони міняються місцями. У російській мові для позначення такого підходу часто застосовується термін «пінг-понг», англійською, по аналогії зі звичайною схемою, більш вдалим буде назва «*out-place*».

Стосовно до ШПФ можна реалізувати різні схеми типу «*out-place*». В даному випадку викладається один з можливих підходів до побудови таких схем, що дозволяє, отримати найбільш ефективні і прості в реалізації алгоритми.

Для викладу цього підходу відзначимо деякі моменти, що лежать в основі алгоритмів Кулі-Тьюки.

ДПФ для N точок, де $N = M \times L$, можна виконати більш ефективно в такий спосіб:

- вихідний масив з N елементів представляється у вигляді двовимірної матриці розміром $M \times L$;
- обчислюються L -точкові ДПФ всіх стовпців;
- всі елементи матриці множаться на так звані множники, що повертають (*twiddle factors*);
- обчислюються M -точкові ДПФ усіх рядків;
- міняються місцями рядки і стовпчики з отриманням спектру сигналу.

Дана схема може застосовуватися рекурсивно, тобто в тому випадку, якщо L або M складене число, ДПФ рядків (або стовпчиків) також обчислюється за допомогою уявлення рядків (шпальти) у вигляді матриці.

У тому випадку, якщо N є ступенем двійки, рекурсивні обчислення можна реалізувати, залишаючи на кожній стадії стовпці (рядки) з двох елементів і виконуючи обчислення ДПФ для двох точок. Це буде алгоритм Кулі-Тьюки по підставі 2. Якщо на кожній стадії виконувати ДПФ для чотирьох точок, то вийде алгоритм по підставі 4.

Обмін рядків і стовпців після кожної стадії обчислень не дозволяє організувати обчислення «*in-place*», тому застосовується інший підхід – один раз до або після виконання основної операції (в залежності від того, для рядків або стовпців виконується доточкове ДПФ) виконується так звана біт реверсне сортування елементів. Відповідно розрізняють так звані алгоритми з проріджуванням по частоті і з проріджуванням по часу [2].

Розглянемо можливий варіант реалізації «*out-place*» алгоритму Кулі-Тьюки за основою 2.

На вході алгоритму ми має N точок вхідного сигналу,

$$N = 2^K,$$

де K - кількість стадій алгоритму.

На першій стадії представляємо вхідні дані у вигляді матриці з двох рядків. Перший рядок розташовується в першій половині буфера, другий рядок - у другій половині. Над елементами кожного стовпчика виконується двоточкове ШПФ, множаться потрібні елементи на повертаються множники і записуються у вихідний буфер в наступному рядку: перший елемент першого двоточкового ШПФ, другий елемент першого двоточкового ШПФ і далі елементи інших ШПФ. Графічно ця операція зображена на рис. 1 на ньому прямокутниками позначені буфери,

точками – комплексні числа, над якими виконується ШПФ, квадратами – операція двоточкового ШПФ з множенням на множник, що повертається. Буфери – це масиви, порядок даних в яких відповідає розташуванню точок зліва направо. Різниця в розташуванні точок по висоті не впливає на розташування чисел в пам'яті і введено лише для більш наочного викладу алгоритму. Для завершення ДПФ необхідно виконати ДПФ рядків, тобто ДПФ парних і непарних елементів (на рис. 1 вони обведені прямокутниками із закругленими кутами).

На другій стадії паралельно виконується ДПФ двох рядків, кожен з яких, в свою чергу, представляється у вигляді матриці з двох рядків. Перший рядок розташовується в першій половині буфера, другий – в другій половині. Над елементами кожного стовпчика виконується двоточкове ШПФ, множаться потрібні елементи на повертають множники і записуються у вихідний буфер в наступному порядку: перший і другий елемент першого двоточкового ШПФ першого рядка, перший і другий елемент першого двоточкового ШПФ другого рядка і далі елементи інших ШПФ. Для однакових елементів кожного з рядків, для яких виконується ДПФ, використовується один множник.

На кожній наступній стадії число рядків, над яким паралельно виконується ДПФ, подвоюється. На останній стадії в рядках залишається по два елементи, над якими виконується двоточкове перетворення. Після закінчення останньої стадії необхідно виконати біт-реверсне сортування.

У розглянутому алгоритмі дані на кожній стадії зчитуються в два потоки (з першої і другої половини буфера) і зберігаються в один потік, тобто дані зчитуються послідовно і не потрібне використання адресної арифметики.

Аналогічним чином можна побудувати «пінг-понг». Зчитування даних в цьому алгоритмі може проводитися в 4, а збереження в один потік; коефіцієнти для кожної четвірки чисел завантажуються один раз. За аналогією з алгоритмами проріджування за часом і по частоті, можна побудувати «пінг-понг» алгоритми, що зчитують дані в один потік, а зберігають у 2 або 4 потоку.

Розглянутий алгоритм був реалізований на процесорі *ARM Cortex-M3*. Час виконання перетворення Фур'є для 128 точок (вхід - комплексні 16-бітові цілі числа в форматі Q15 нормовані на половину діапазону, для запобігання переповнення після кожної стадії дані діляться на два) включаючи збереження контексту складала 1900 тактів (менше 15 тактів на точку). Цей результат на 20% швидше найкращого з відомих результатів (2400 тактів, менш 19 тактів на точку) для алгоритмів, що виконують обчислення «на місці» [3]. Схема реалізації наведена на рис. 2.

На перерахованих розмірах вибірки мікропроцесор *ARM Cortex-M3* дозволяє проводити обчислення ШПФ в режимі реального часу із затримкою необхідною для накопичення необхідного числа відліків дискретизованого сигналу. Реа-

лізований алгоритм ШПФ може бути використаний в якості базового при розрахунку частотно-тимчасових кореляційних функцій [4,5].

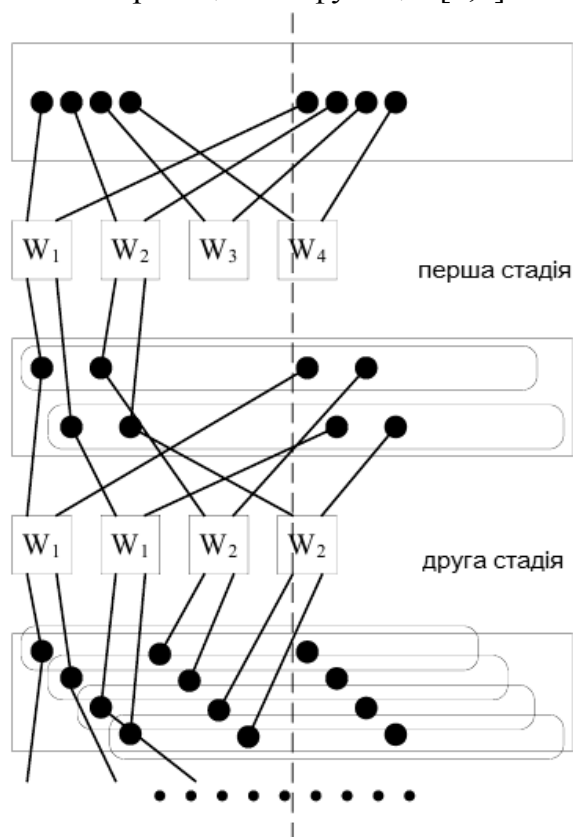


Рис. 1. Структура обробки даних

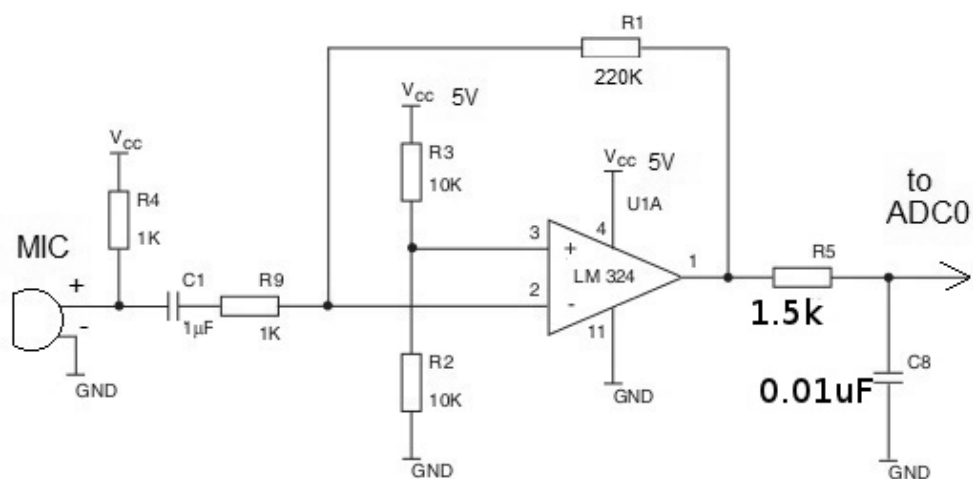


Рис. 2. Швидке перетворення Фур'є на ARM мікроконтролері

Результати отриманих даних наведені у таблиці 1.

Таблиця 1. Результат отриманих даних

Довжина перетворення	Час виконання, мкс	Час перетворення, мкм
8	327,7	317,7
16	467,9	455,8
32	688,6	667,5
64	1246,7	1160,2
128	2400	1900

Графічний результат виведено на діаграми, що представлені на рис. 3-4.

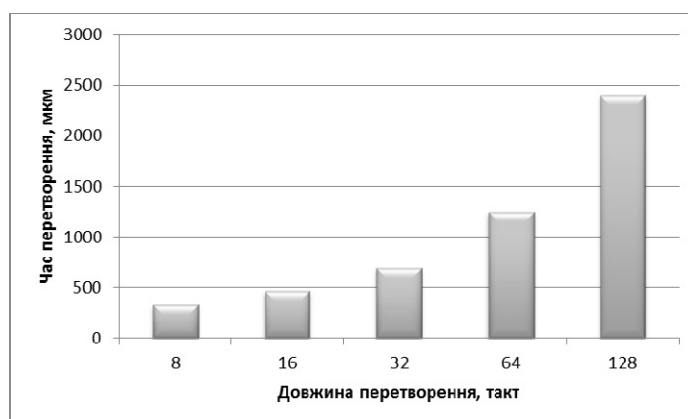


Рис. 3. Діаграма результату опрацювання, поточні значення

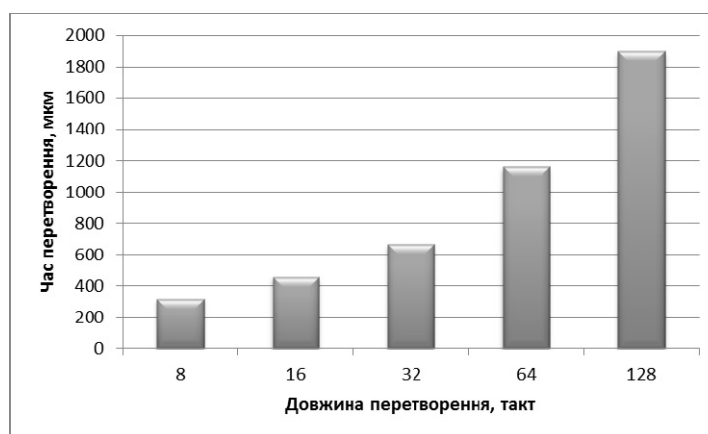


Рис. 4. Діаграма результату опрацювання, перетворення Фур'є.

Таким чином, алгоритми типу «пінг-понг» дозволяють отримувати в загальному випадку, більш швидкі і прості функції для обчислення спектрів сигналів.

Вони були використані при розробці функцій ШПФ математичної бібліотеки для процесорів на ядрах ZSP. Розроблені функції мають продуктивність на 10-20% вище оціночних значень продуктивності класичних алгоритмів.

Висновки

У межах даної роботи здійснено синтез алгоритму цифрової фільтрації на основі математичного апарату швидкого перетворення Фур'є. На основі проведення досліджень варто зазначити, що для збільшення швидкодії ШПФ необхідно організувати таким чином, щоб на кожній стадії дані для обробки можна було б зчитувати з пам'яті послідовно. Для цього на попередній стадії результати обчислень потрібно зберігати в необхідному порядку.

Розглянутий алгоритм був реалізований на процесорі *ARM Cortex-M3*. Час виконання перетворення Фур'є для 128 точок (вхід - комплексні 16-бітові цілі числа в форматі Q15 нормовані на половину діапазону, для запобігання переповнення після кожної стадії дані діляться на два) включаючи збереження контексту склав 1900 тактів (менше 15 тактів на точку). Цей результат на 20% швидше найкращого з відомих результатів (2400 тактів, менш 19 тактів на точку) для алгоритмів, що виконують обчислення «на місці».

Таким чином, алгоритми типу «пінг-понг» дозволяють отримувати в загальному випадку, більш швидкі і прості функції для обчислення спектрів сигналів. Вони були використані при розробці функцій ШПФ математичної бібліотеки для процесорів на ядрах *ZSP*. Розроблені функції мають продуктивність на 10-20% вище оціночних значень продуктивності класичних алгоритмів.

Список використаних джерел

1. Новиков И.Я. Основы теории всплесков / И.Я. Новиков, С.Б. Стечкин // Успехи математических наук. 1998. – №. 53. – № 6. – С. 9-13.
2. Петухов А.П. Введение в теорию базисов всплесков. / А.П. Петухов. - СПб.: Изд. СПбГТУ, 1999. – 131 с.
3. Martin Vatterli, Jelena Kovačević. Wavelets and Subband Coding. - Prentice Hall, New Jersey, 2007. – 522 p.
4. Лукин А. Введение в цифровую обработку сигналов. / А. Лукин. – М.: Наука, 2002. – 44 с.
5. Parkhomey I. Method for UAV trajectory parameters estimation using additional radar data. / V. Kozlovski, I. Parkhomey, R. Odarchenko // Methods and Systems of Navigation and Motion Control, 4th International conference 18-20 Oct. – 2016. - P. 19-20.