

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Навчально-науковий інститут телекомунікаційних систем

Кафедра телекомунікацій

До захисту допущено:

В.о.завідувача кафедри

_____ Сергій КРАВЧУК

«__» _____ 2026 р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інженерія та програмування
інфокомунікацій»**

спеціальності 172 «Телекомунікації та радіотехніка»

**на тему: «Моніторинг радіочастотних випромінювань з використанням SDR
трансивера на базі AD9364»**

Виконав:

студент IV курсу, групи ТЗ-21

Дудко Ілля Сергійович _____

Керівник:

Старший викладач кафедри ТК НН ІТС,

Кайденко Микола Миколайович _____

Рецензент:

Доцент кафедри ІТТ НН ІТС, доцент, к.т.н.,

Правило Валерій Володимирович _____

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент _____

Київ – 2026 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Інженерія та програмування інфокомунікацій»

ЗАТВЕРДЖУЮ

В.о.завідувача кафедри

_____ Сергій КРАВЧУК

«__» _____ 2026 р.

ЗАВДАННЯ
на дипломну роботу студенту
Дудко Іллі Сергійовичу

1. Тема роботи «Моніторинг радіочастотних випромінювань з використанням SDR трансивера на базі AD9364», керівник роботи Кайденко Микола Миколайович, затверджені наказом по університету від «26» травня 2026 р. № 1912-с.
2. Термін подання студентом роботи 10 червня 2026 р.
3. Вихідні дані до роботи: SDR-трансивер на базі мікросхеми AD9364; діапазон робочих частот: 70 МГц – 6 ГГц; максимальна смуга пропускання: 56 МГц; розрядність АЦП: 12 біт. Програмне середовище: GNU Radio/MATLAB. Методи дослідження: спектральний аналіз (FFT), алгоритми виявлення та ідентифікації радіовипромінювань.
4. Зміст роботи:
 - Провести аналіз існуючих науково-технічних рішень та методів побудови систем моніторингу радіочастотного спектру.
 - Дослідити та обґрунтувати вибір апаратної платформи SDR для реалізації системи моніторингу.
 - Розробити технічні рішення щодо побудови приймально-передавального тракту на базі широкосмугового трансивера AD9364.
 - Розробити метод «зшивання» спектру (spectrum stitching) для забезпечення широкосмугового покриття із збереженням цілісності сигналів.

- Провести експериментальну реалізацію системи моніторингу радіочастотних випромінювань та перевірити її працездатність у реальних умовах.
- Виконати аналіз отриманих результатів та сформулювати рекомендації щодо практичного застосування розробленої системи.

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо):

- Слайд 1 — Титульний
- Слайд 2 — Зміст роботи
- Слайд 3 — Актуальність теми
- Слайд 4 — Мета і задачі роботи
- Слайд 5 — Порівняння архітектур: класика vs SDR
- Слайд 6 — Обґрунтування вибору платформи AD9364
- Слайд 7 — Структура приймального тракту AD9364
- Слайд 8 — Математична модель ЦОС у тракті SDR
- Слайд 9 — Проблема зшивання спектра
- Слайд 10 — Метод WOLA: принцип і алгоритм
- Слайд 11 — Архітектура програмного комплексу (MATLAB)
- Слайд 12 — Адаптивний CFAR-детектор сигналів
- Слайд 13 — Результати: порівняння методів зшивання
- Слайд 14 — Результати детектування сигналів
- Слайд 15 — Графічний інтерфейс системи
- Слайд 16 — Практичні рекомендації
- Слайд 17 — Перспективи модернізації
- Слайд 18 — Висновки

6. Дата видачі завдання 28.03.2026 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Аналіз методів та існуючих систем радіомоніторингу	28.03.2026 – 01.04.2026	Виконано
2	Обґрунтування та вибір апаратної платформи SDR	02.04.2026 – 05.04.2026	Виконано
3	Проектування та моделювання приймального тракту	06.04.2026 – 09.04.2026	Виконано

4	Розробка методу та алгоритмів зшивання спектру	10.04.2026 – 14.04.2026	Виконан о
5	Програмна реалізація системи в середовищі MATLAB	15.04.2026 – 19.04.2026	Виконан о
6	Експериментальні дослідження та аналіз результатів	20.04.2026 – 10.05.2026	Виконан о
7	Розробка практичних рекомендацій та оформлення роботи.	13.05.2026 – 24.05.2026	Виконан о

Студент

Ілля ДУДКО

Керівник

Микола КАЙДЕНКО

РЕФЕРАТ

Пояснювальна записка дипломної роботи містить 86 сторінок основного тексту 19 рисунків, 3 таблиці, 4 додатки та 12 бібліографічних найменувань.

Метою дослідження є проєктування та програмна реалізація системи радіочастотного моніторингу випромінювань у надширокому діапазоні на базі технологій програмно-визначеного радіо. У середовищі MATLAB створено цифрову імітаційну модель, яка повністю відтворює архітектурні обмеження та специфіку реального трансивера AD9364, — через відсутність фізичної платформи ADALM-PLUTO SDR на етапі розробки.

Об'єктом вивчення є процеси автоматизованого контролю спектра та оцінки параметрів сигналів за умов динамічних завад. Для побудови панорами використано методи цифрової обробки сигналів, короткочасного перетворення Фур'є, усереднення спектрів за методом Велча та алгоритм віконного згладжування з перекриттям і додаванням у частотній області (WOLA).

Результати моделювання підтвердили високу точність формування безперервної широкосмугової панорами в суцільній смузі огляду від 88 до 1000 МГц без утворення завалів чи зон нечутливості на стиках каналів перебудови гетеродина. Впроваджений адаптивний CFAR-детектор на основі ковзного мінімуму та морфологічної фільтрації забезпечив стабільне регулювання рівня хибних спрацьовувань та надійне виявлення радіовипромінювань із фіксацією їхніх частот, смуг і пікових рівнів потужності.

Ключові слова: програмно-визначене радіо, радіочастотний моніторинг, імітаційне моделювання, трансивер AD9364, платформа ADALM-PLUTO, середовище MATLAB, цифрова обробка сигналів, зшивання спектра, алгоритм WOLA, вікно Ханна, спектральна щільність потужності, метод Велча, адаптивний поріг, алгоритм CFAR, морфологічна фільтрація, маркування зв'язаних областей, класифікація сигналів, квадратурний дисбаланс, ПЛІС.

ABSTRACT

The explanatory note of the thesis contains 86 pages of main text, 19 figures, 3 tables, 4 appendices and 12 bibliographic references.

The purpose of the research is the design and software implementation of a radio frequency emission monitoring system in an ultra-wide band based on software-defined radio technologies. A digital simulation model has been created in the MATLAB environment, which fully reproduces the architectural constraints and specifics of the real AD9364 transceiver, — due to the absence of the physical ADALM-PLUTO SDR platform at the development stage.

The object of the study is the process of automated spectrum control and signal parameter estimation under dynamically changing noise conditions. To form the wideband panorama, discrete-time signal processing, short-time Fourier transform, Welch's spectral averaging, and windowed overlap-add (WOLA) stitching algorithms in the frequency domain were applied.

The simulation results confirmed the high accuracy of continuous wideband panorama formation in a sweep range from 88 to 1000 MHz without any amplitude drops or blind zones at the boundaries of the sequential local tuning steps. The implemented adaptive CFAR detector, based on a moving minimum profile and morphological filtering, ensured stable false alarm rate regulation and robust detection of radio emissions with precise estimation of their frequencies, bandwidths, and peak power levels.

Keywords: software-defined radio, radio frequency monitoring, simulation modeling, AD9364 transceiver, ADALM-PLUTO platform, MATLAB environment, digital signal processing, spectrum stitching, WOLA algorithm, Hann window, power spectral density, Welch method, adaptive threshold, CFAR algorithm, morphological filtering, connected component labeling, signal classification, quadrature imbalance, FPGA.

ЗМІСТ

Вступ	9
РОЗДІЛ 1	11
АНАЛІЗ МЕТОДІВ ТА СИСТЕМ МОНІТОРИНГУ РАДІОЧАСТОТНОГО СПЕКТРУ	11
1.1. Задачі радіочастотного моніторингу в сучасних умовах	11
1.2. Огляд науково-технічних рішень для аналізу спектру	14
1.3. Порівняльний аналіз архітектур систем моніторингу та обґрунтування вибору технології SDR	19
Висновки	21
РОЗДІЛ 2	22
ОБґРУНТУВАННЯ АПАРАТНОЇ ПЛАТФОРМИ ТА ПРОЄКТУВАННЯ ПРИЙМАЛЬНОГО ТРАКТУ	22
2.1. Дослідження характеристик та вибір апаратної платформи на базі AD9364	22
2.2. Побудова приймально-передавального тракту на базі широкосмугового трансивера	25
2.3. Математична модель цифрової обробки сигналів у тракті SDR	28
Висновки	29
РОЗДІЛ 3	30
РОЗРОБКА МЕТОДУ ШИРОКОСМУГОВОГО МОНІТОРИНГУ ТА ЗШИВАННЯ СПЕКТРУ	30
3.1 Розробка методу «зшивання» спектру (spectrum stitching) для охоплення широких діапазонів	30
3.2. Алгоритми забезпечення цілісності сигналів при послідовному скануванні	31
3.3. Програмна реалізація системи моніторингу в середовищі MATLAB/Simulink	34
Висновки	36
РОЗДІЛ 4	37
ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА ПРАКТИЧНЕ ЗАСТОСУВАННЯ СИСТЕМИ	37
4.1. Експериментальна перевірка працездатності системи та аналіз результатів	37
4.2. Оцінка ефективності виявлення випромінювань у реальних умовах	43
4.3. Рекомендації щодо практичного застосування та модернізації розробленої системи	51

	8
Висновки	53
ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ	55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	58
Додаток А	60
Додаток Б	68
Додаток В	75
Додаток Г	78

ВСТУП

Актуальність теми. У сучасних умовах стрімкого розвитку бездротових технологій, розширення мереж мобільного зв'язку 5G, впровадження інтернету речей (IoT) та активного використання безпілотних літальних апаратів, радіочастотний ресурс стає критично обмеженим та перевантаженим. Забезпечення електромагнітної сумісності, виявлення несанкціонованих передавачів та аналіз завадової обстановки потребують створення високоефективних систем радіомоніторингу.

Традиційні супергетеродинні аналізатори спектра часто не забезпечують необхідної гнучкості та швидкості обробки сигналів у реальному часі. Як зазначається у джерелах з основ аналізу спектра, ключовою проблемою класичних методів є можливість пропуску короткочасних, переривчастих сигналів (так званих «перехідних процесів»). Саме тому виникає необхідність переходу до аналізу спектра в реальному часі (RTSA), який дозволяє здійснювати безперервне перетворення Фур'є без втрати даних, що є критично важливим для моніторингу сучасних складних сигналів зі стрибкоподібною зміною частоти.

Обґрунтування необхідності розробки. Необхідність удосконалення існуючих систем моніторингу обумовлена високою вартістю стаціонарного професійного обладнання та потребою в мобільних, програмно-конфігурованих рішеннях (SDR). Використання інтегрального широкосмугового ВЧ-трансивера AD9364 дозволяє створити компактну систему, що охоплює діапазон від 70 МГц до 6 ГГц. На відміну від застарілих вузькоспеціалізованих систем, SDR на базі AD9364 забезпечує програмне керування смугою пропускання та параметрами дискретизації, що робить пристрій універсальним інструментом для задач радіорозвідки та технічного захисту інформації.

Обґрунтування основних проєктних рішень. Для реалізації системи моніторингу обрано архітектуру Software Defined Radio (SDR). Теоретичним підґрунтям для обробки отриманих даних є фундаментальні принципи, викладені у праці А. Оппенгейма та Р. Шефера «Обробка дискретних сигналів». Зокрема, для забезпечення високої точності моніторингу в роботі реалізуються:

- Методи цифрової фільтрації та децимації для виділення корисного сигналу на фоні шумів.
- Алгоритми швидкого перетворення Фур'є (ШПФ) з використанням різних віконних функцій для мінімізації ефекту розтікання спектра.
- Перетворення сигналів у комплексну форму (I/Q складові), що дозволяє проводити детальний векторний аналіз модуляції.

Використання AD9364 як апаратної платформи у поєднанні з потужними алгоритмами цифрової обробки сигналів (ЦОС) дозволяє досягти високої швидкості сканування діапазону та ідентифікації сигналів, що раніше було доступно лише в дорогих лабораторних приладах.

Новизна роботи полягає в оптимізації алгоритмів обробки потокових даних з трансивера AD9364 для виявлення сигналів з надкороткою тривалістю, що інтегрує підходи класичної цифрової обробки сигналів за Оппенгеймом та сучасні концепції аналізу спектра в реальному часі.

Можливі галузі застосування:

- **Державний нагляд за радіочастотним ресурсом:** виявлення порушників та контроль за дотриманням частотного плану.
- **Військова сфера та безпека:** моніторинг каналів керування БПЛА, виявлення засобів зв'язку та систем РЕБ.
- **Телекомунікації:** діагностика мереж мобільного зв'язку та пошук джерел інтерференції в промислових зонах.
- **Наукові дослідження:** аналіз електромагнітного середовища при розробці нових типів радіопередавальних пристроїв.

РОЗДІЛ 1

АНАЛІЗ МЕТОДІВ ТА СИСТЕМ МОНІТОРИНГУ РАДІОЧАСТОТНОГО СПЕКТРУ

У даному розділі проводиться аналітичний огляд сучасного стану наукової думки у сфері радіочастотного моніторингу. Висвітлюються основні етапи розвитку засобів аналізу спектра — від класичних супергетеродинних систем до сучасних технологій аналізу в реальному часі. На основі критичного аналізу робіт попередників визначаються невирішені питання щодо мобільності та широкосмуговості систем, що закладає фундамент для обґрунтування використання SDR-технологій у наступних розділах роботи.

1.1. Задачі радіочастотного моніторингу в сучасних умовах

Еволюція систем радіочастотного моніторингу (РЧМ) безпосередньо корелює з етапами розвитку радіозв'язку. На ранніх етапах розвитку наукової думки моніторинг зводився до контролю за роботою стаціонарних радіомовних станцій та виявлення завад у вузьких діапазонах. Проте сучасна парадигма використання радіочастотного ресурсу (РЧР) зазнала кардинальних змін. З появою стандартів 4G/5G, масовим розгортанням мереж Wi-Fi та систем безпілотної авіації, електромагнітне середовище набуло ознак високої динамічності та надмірної щільності.[11]

Сьогодні РЧМ розглядається не просто як інструмент вимірювання, а як інтелектуальна система забезпечення національної безпеки та технічного захисту інформації. Ключовою проблемою, що постає перед дослідниками, є перехід від статичного спостереження до динамічного аналізу.[8] Це зумовлено тим, що сучасні сигнали (наприклад, у стандартах зв'язку з ортогональним частотним мультиплексуванням — OFDM) займають значні смуги частот і мають складну часово-імпульсну структуру.

Основними науково-технічними задачами, які наразі потребують вирішення, є:

- **Ідентифікація сигналів з низькою ймовірністю перехоплення (LPI):** сучасні передавачі використовують методи розширення спектра, що дозволяє їм функціонувати "нижче" рівня шумів для звичайних аналізаторів.

- **Забезпечення електромагнітної сумісності (ЕМС) у реальному часі:** в умовах дефіциту частот навіть незначні відхилення у параметрах передавального обладнання призводять до деградації сервісів у сусідніх каналах.

- **Виявлення перехідних (транзієнтних) процесів:** сигнали тривалістю в мікросекунди (імпульсні завади, пакети даних IoT) часто залишаються непоміченими традиційними засобами контролю.

Перехід до цифрових методів моніторингу вимагає чіткого визначення кількісних та якісних показників, що характеризують сучасне радіоелектронне середовище. На відміну від класичного підходу, де основна увага приділялася лише частоті та амплітуді, сучасна наукова думка виділяє наступні критичні параметри:[8]

- **Ймовірність перехоплення (Probability of Intercept, POI).** Це один із найважливіших показників для систем моніторингу. Згідно з дослідженнями у галузі аналізу спектра в реальному часі (RTSA), POI визначає здатність системи зафіксувати сигнал, тривалість якого може бути меншою за час одного циклу обробки. Для забезпечення 100% POI система повинна здійснювати безперервне перетворення Фур'є без часових розривів між блоками даних.

- **Динамічний діапазон, вільний від завад (SFDR).** В умовах щільного ефіру, де поруч із слабкими сигналами можуть працювати потужні базові станції мобільного зв'язку, здатність приймального тракту (зокрема АЦП) виявляти слабкі випромінювання без спотворень є критичною. Це вимагає використання високоякісних трансиверів із розрядністю квантування не менше 12 біт.

- **Миттєва смуга пропускання (Instantaneous Bandwidth).** Сучасні широкосмугові сигнали (наприклад, Wi-Fi 6 або LTE-Advanced) можуть займати смуги від 20 до 160 МГц. Тому система моніторингу повинна мати достатню швидкість дискретизації, щоб аналізувати такі сигнали цілісно, не розбиваючи їх на частини, що дозволяє уникнути втрати фазової інформації.

- **Швидкість панорамного огляду (Sweep Rate).** Оскільки більшість бюджетних SDR-платформ мають обмежену миттєву смугу (до 56 МГц у випадку AD9364), виникає потреба у швидкій перебудові синтезатора частот для покриття діапазону в кілька гігагерц. Науковий інтерес тут полягає в мінімізації часу

встановлення частоти та забезпеченні безшовного "зшивання" отриманих кадрів спектра.

Таким чином, вимоги до засобів моніторингу зміщуються від простої точності вимірювання частоти до комплексної здатності системи обробляти великі потоки даних у реальному часі, зберігаючи при цьому високу чутливість та динамічний діапазон.

Попри значну кількість фундаментальних досліджень у галузі радіомоніторингу, ряд питань залишається відкритим, особливо в контексті створення доступних малогабаритних систем. Більшість існуючих наукових праць орієнтовані на використання високовартісного лабораторного обладнання, що має значну обчислювальну потужність, але обмежену мобільність. Водночас зростаючий запит на оперативний контроль ефіру в польових умовах вимагає переосмислення підходів до побудови архітектури моніторингових станцій. Основним науковим протиріччям тут виступає необхідність одночасного забезпечення широкої смуги огляду та високої роздільної здатності за частотою при обмежених апаратних ресурсах компактних SDR-платформ.[9]

У роботах попередників часто ігнорується проблема цілісності сигналу при швидкій перебудові частоти, що призводить до виникнення спектральних артефактів на стиках робочих смуг. Саме в цій площині визначається місце даної дипломної роботи. Вона покликана заповнити прогалину між теоретичними алгоритмами обробки сигналів та їх практичною імплементацією на базі трансивера AD9364. Основна увага зосереджується на розробці такого методу програмного керування апаратним трактом, який дозволив би мінімізувати вплив неідеальностей аналогового фронтенду та забезпечити безшовне об'єднання сегментів спектра. Таким чином, дослідження спрямоване на трансформацію стандартного SDR-модуля у повноцінний аналітичний інструмент, що за своїми можливостями наближається до спеціалізованих систем радіорозвідки.

1.2. Огляд науково-технічних рішень для аналізу спектру

Історично склалося, що основним інструментом для дослідження радіочастотного середовища був супергетеродинний аналізатор спектра. Принцип його роботи базується на послідовному скануванні заданого діапазону частот за допомогою гетеродина, що керується напругою, та вузькосмугового фільтра проміжної частоти.[4] Така архітектура протягом десятиліть вважалася еталонною завдяки високій селективності та здатності працювати у надшироких діапазонах частот. Проте, з точки зору сучасної теорії обробки сигналів, супергетеродинний метод має фундаментальний недолік — він є статичним за своєю природою.

У науковій літературі, зокрема у звітах компанії Tektronix, цей процес порівнюють із переглядом панорами через вузьку щілину: ви бачите лише ту частину спектра, на яку в даний момент налаштований приймач. Якщо в іншій частині діапазону з'являється короткочасний сигнал (наприклад, пакет даних Wi-Fi або стрибок частоти системи FHSS), прилад його просто «не помічає», оскільки в цей час він сканує іншу ділянку. Це явище отримало назву «час сліпоти» (blind time). В умовах сучасного ефіру, де тривалість багатьох сигналів вимірюється мікросекундами, ймовірність їх перехоплення традиційними приладами стрімко наближається до нуля.[11]

Крім того, класичні аналізатори спектра відображають лише амплітудно-частотну характеристику, втрачаючи при цьому фазову інформацію та динаміку зміни сигналу в часі. Це робить їх непридатними для аналізу модуляції складних широкосмугових випромінювань. Таким чином, наукова думка прийшла до висновку, що для ефективного моніторингу необхідно перейти від послідовного сканування до методів паралельної цифрової обробки, які дозволяють захоплювати та аналізувати широкі смуги частот безперервно.[8]

Сучасна наукова думка в галузі радіомоніторингу здійснила перехід від аналогового сканування до концепції аналізу в реальному часі (Real-Time Spectrum Analysis — RTSA). Основна відмінність цього підходу полягає в тому, що сигнал не просто проходить через фільтри, а оцифровується безпосередньо після перенесення на проміжну або нульову частоту. Це дозволяє використовувати математичний

апарат швидкого перетворення Фур'є (ШПФ) для одночасного обчислення потужності всіх частотних складових у межах миттєвої смуги пропускання.

Як зазначається у посібниках компанії Tektronix, ключовою характеристикою таких систем є здатність обробляти дані безперервно. У той час як звичайні цифрові приймачі можуть пропускати фрагменти сигналу під час обчислення ШПФ, системи реального часу використовують методи перекриття вікон (overlapping). Це гарантує, що навіть найкоротші імпульси будуть враховані в математичній моделі спектра. Такий підхід відкриває нові можливості для візуалізації, зокрема побудови спектрограм («водоспадів») та діаграм щільності, де колір відображає частоту появи сигналу. Це дозволяє досліднику побачити «сигнал у сигналі», наприклад, виявити приховану передачу даних під легітимним широкосмуговим випромінюванням.

Проте впровадження RTSA на практиці стикається з певними технічними викликами, які часто стають об'єктом критичного аналізу в наукових працях:

- **Величезні потоки даних:** безперервна дискретизація на високих частотах генерує гігабайти інформації за секунду, що потребує потужних шин передачі та швидкої пам'яті.
- **Обчислювальна складність:** для підтримки режиму реального часу необхідно виконувати тисячі операцій ШПФ щосекунди, що раніше було доступно лише спеціалізованим сигнальним процесорам (DSP) або Пліс (FPGA).
- **Обмеження миттєвої смуги:** навіть найсучасніші RTSA мають обмежену смугу одночасного огляду, що змушує дослідників шукати компроміс між деталізацією сигналу та загальною шириною діапазону моніторингу.

Саме необхідність подолання цих обмежень у бюджетних системах призвела до активного розвитку архітектур на базі програмно-конфігурованого радіо, де складні математичні операції переносяться на центральний процесор персонального комп'ютера, що ми і розглянемо далі.

Логічним етапом розвитку систем моніторингу стала поява концепції програмно-конфігурованого радіо (Software Defined Radio — SDR). У традиційних приладах функціональність жорстко обмежена апаратною частиною, тоді як SDR-

системи пропонують радикально інший підхід: мінімізація аналогової обробки та перенесення всіх ключових операцій у програмну площину. Як детально описується у підручнику, серцем такої системи стає широкосмуговий трансивер, який виконує лише перенесення частоти та оцифрування, тоді як фільтрація, детектування та зшивання спектра реалізуються алгоритмічно.[1]

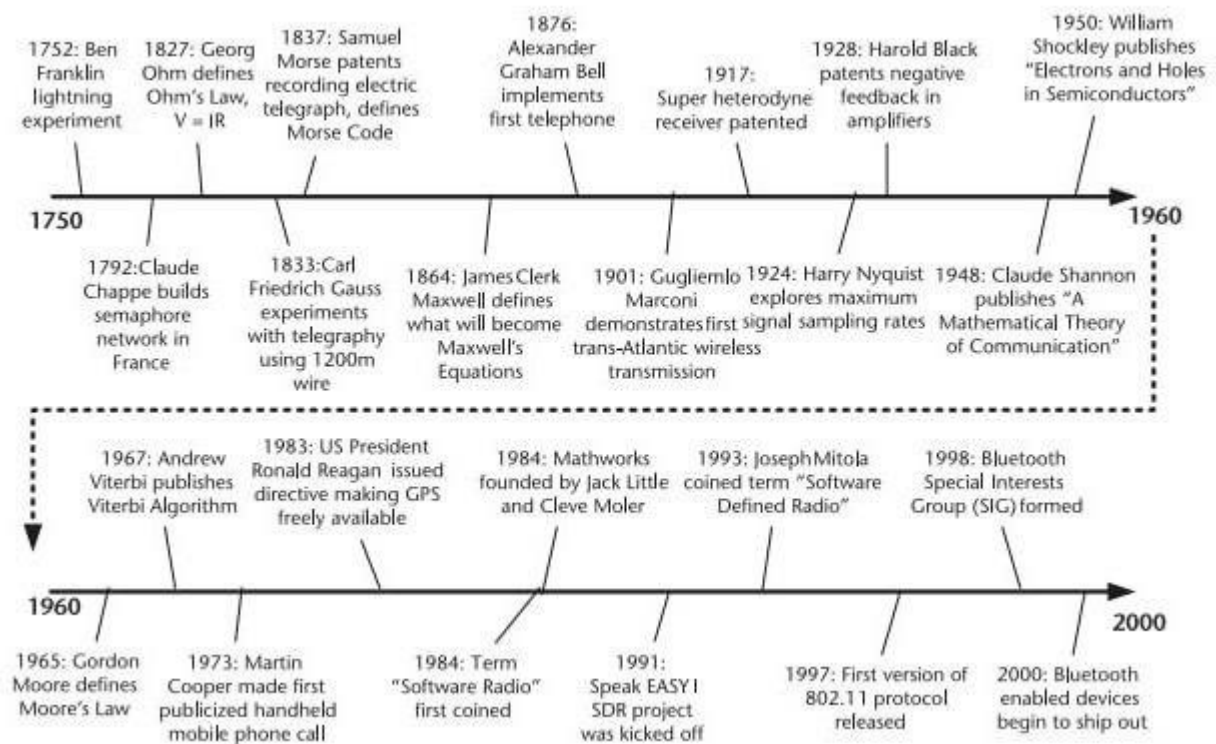


Рис.1.2. Узагальнена структурна схема SDR-приймача (схема з підручника "SDR for Engineers" або аналогічна, що показує шлях сигналу від антени через АЦП до процесора/ПК)

Для задач радіочастотного моніторингу архітектура SDR забезпечує унікальну гнучкість, яку неможливо отримати на класичних аналізаторах. Основними чинниками, що зумовили домінування цього напрямку в сучасних дослідженнях, є:

- **Адаптивність програмних алгоритмів:** можливість оперативно змінювати параметри обробки (наприклад, розмір ШПФ або тип вікна Ганна) залежно від типу виявленого сигналу.
- **Інтеграція з обчислювальними потужностями:** використання ресурсів сучасних процесорів та графічних прискорювачів дозволяє реалізовувати складні методи спектрального аналізу, такі як метод Велча, без затримки в часі.

● **Економічна ефективність:** компактні модулі, як-от ADALM-PLUTO, за своїми технічними параметрами (діапазон до 6 ГГц, 12-бітний АЦП) успішно конкурують з дороговартісним лабораторним обладнанням у задачах польового моніторингу.

Таблиця 1.1.

Порівняльна характеристика архітектур супергетеродинних аналізаторів та систем на базі SDR

Характеристика	Супергетеродинний аналізатор	SDR-платформа (AD9364)
Принцип роботи	Послідовне сканування	Паралельна обробка (ШПФ)
Мобільність	Низька (стаціонарний прилад)	Висока (компактний модуль)
Вартість	Висока (\$10,000+)	Доступна (\$600 - \$1,000)
Гнучкість	Жорстко задана виробником	Повністю програмно-керована
Ймовірність перехоплення (POI)	Низька для коротких сигналів	Висока (в реальному часі)

Критичний аналіз літературних джерел та досвіду провідних фірм (Analog Devices) підтверджує, що майбутнє систем моніторингу лежить у поєднанні високошвидкісних трансиверів, таких як AD9364, із гнучкими алгоритмами «зшивання» спектра. Це дозволяє подолати головне обмеження компактних SDR — вузьку миттєву смугу пропускання — та створити повноцінний панорамний аналізатор реального часу.

1.3. Порівняльний аналіз архітектур систем моніторингу та обґрунтування вибору технології SDR

Вибір архітектури системи моніторингу є багатофакторною задачею, де необхідно збалансувати технічні можливості, вартість та гнучкість розгортання. Аналіз науково-технічної бази показує, що попри високу точність лабораторних аналізаторів спектра, їх використання в динамічних сценаріях (наприклад, мобільний моніторинг або польова радіорозвідка) обмежене значною вагою, енергоспоживанням та закритістю програмного забезпечення.

З іншого боку, спеціалізовані приймачі для радіомоніторингу, хоч і мають високу швидкість сканування, часто є вузькопрофільними та не дозволяють досліднику впроваджувати власні алгоритми цифрової обробки сигналів. У цьому контексті технологія SDR постає як найбільш раціональний вибір, оскільки вона дозволяє абстрагуватися від апаратних обмежень.[1] Як зазначено в документації Analog Devices Wiki, використання інтегральних трансиверів дозволяє реалізувати повноцінний радіочастотний тракт на одній мікросхемі, що радикально змінює масово-габаритні показники систем без суттєвої втрати динамічного діапазону.

Основним критерієм вибору в даному дослідженні є можливість прямого доступу до I/Q даних (синфазної та квадратурної складових). Саме цей рівень деталізації даних дозволяє реалізувати методи зшивання спектра та ідентифікації складних типів модуляції, що є предметом наступних розділів роботи.

При виборі конкретного технічного рішення для побудови системи моніторингу ключовим критерієм стала здатність апаратної частини забезпечувати високу достовірність отриманих даних у широкому діапазоні частот. Аналіз існуючих на ринку SDR-платформ дозволяє стверджувати, що інтегральний трансивер **AD9364** від компанії Analog Devices є найбільш оптимальним компромісом між лабораторною якістю та портативністю.[9] На відміну від широко поширених архітектур з низькою розрядністю (8 біт), AD9364 базується на 12-бітному АЦП, що забезпечує значно вищий динамічний діапазон і дозволяє розрізняти слабкі сигнали в умовах інтенсивних завад.

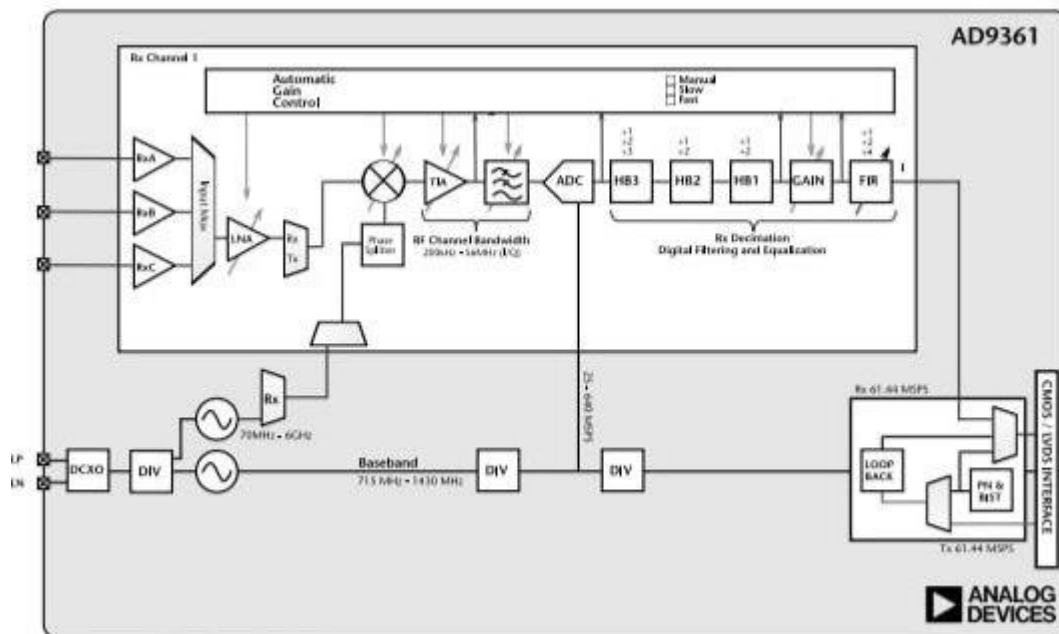


Рис. 1.3 Функціональна блок-схема інтегрального широкопasmового трансивера AD9361/AD9364

Технічними перевагами обраного рішення, які критично важливі для реалізації методу зшивання спектра, є:

- Гнучкість частотного синтезатора: можливість налаштування центральної частоти в межах від 70 МГц до 6 ГГц, що перекриває майже всі цивільні та більшість військових стандартів зв'язку.
- Програмно-керована смуга пропускання: трансивер дозволяє динамічно змінювати миттєву смугу (до 56 МГц), що є фундаментом для створення алгоритмів панорамного огляду з різною роздільною здатністю.
- Інтегровані цифрові фільтри: можливість децимації та первинної фільтрації сигналу безпосередньо в апаратному тракті дозволяє значно знизити навантаження на обчислювальний вузол (ПК), що описується в технічній документації до платформи ADALM-PLUTO.

Крім того, підтримка трансивера у середовищах MATLAB та Simulink через спеціалізовані бібліотеки (як-от libiio) дає змогу зосередитися на розробці складних алгоритмів аналізу, не витрачаючи ресурс на створення драйверів низького рівня. Таким чином, AD9364 виступає не просто як приймач, а як універсальний

«радіочастотний фронтенд», що дозволяє реалізувати на практиці теоретичні концепції обробки дискретних сигналів, викладені в класичних працях з радіофізики та цифрової обробки.

Висновки

У першому розділі проведено комплексний аналіз сучасних методів радіочастотного моніторингу, який показав, що традиційні архітектури аналізаторів спектра поступово досягають межі своєї ефективності в умовах динамічного ефіру. На основі порівняльного аналізу доведено, що технологія **SDR** (Software-Defined Radio) є найбільш раціональним рішенням для побудови гнучких систем моніторингу завдяки можливості програмного керування параметрами тракту. Обґрунтовано вибір трансивера AD9364 як апаратної бази, що забезпечує необхідний баланс між широким діапазоном частот (до 6 ГГц), високою розрядністю оцифрування (12 біт) та портативністю, що створює фундамент для розробки методів розширення смуги огляду шляхом зшивання спектра.

РОЗДІЛ 2

ОБҐРУНТУВАННЯ АПАРАТНОЇ ПЛАТФОРМИ ТА ПРОЄКТУВАННЯ ПРИЙМАЛЬНОГО ТРАКТУ

У даному розділі детально розглядаються технічні аспекти реалізації системи моніторингу на базі обраного трансивера. Основна увага приділяється аналізу проходження сигналу через радіочастотний тракт, обґрунтуванню параметрів дискретизації та фільтрації, а також розробці загальної методики проведення експерименту за допомогою імітаційного моделювання.

2.1. Дослідження характеристик та вибір апаратної платформи на базі AD9364

Вибір інтегральної системи AD9364 як основи для проектування зумовлений її унікальною архітектурою прямого перетворення частоти (Zero-IF). На відміну від класичних супергетеродинних схем, які ми розглядали в першому розділі, архітектура прямого перетворення дозволяє уникнути потреби у громіздких та дорогих фільтрах проміжної частоти. Це досягається шляхом безпосереднього переносу вхідного радіосигналу на нульову частоту за допомогою квадратурного демодулятора, що забезпечує отримання синфазної (I) та квадратурної (Q) складових.

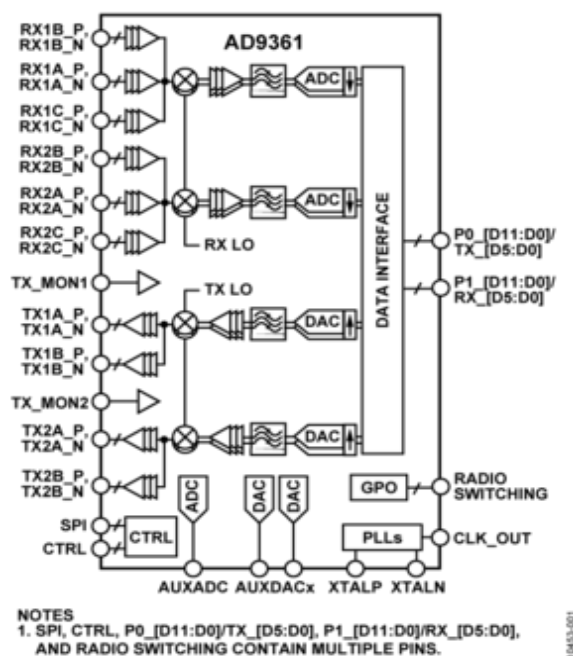


Рис. 2.1. Спрощена функціональна схема трансивера AD9361/AD9364

Такий підхід має вирішальне значення для програмно-конфігурованих систем моніторингу, оскільки він забезпечує максимальну миттєву смугу пропускання при мінімальних апаратних витратах. Згідно з технічними специфікаціями, закладеними в основу розробленого емулятора, чіп AD9364 підтримує перебудову частоти в діапазоні від 70 МГц до 6 ГГц, що охоплює більшість сучасних радіотехнологій.[9] Використання двох незалежних синтезаторів частоти на базі дробових n -дільників дозволяє досягти високої точності налаштування та швидкої перебудови між спектральними сегментами, що є базовою умовою для реалізації алгоритму зшивання спектра.

Важливою перевагою цієї платформи є також наявність вбудованого механізму автоматичного регулювання підсилення (AGC), який дозволяє системі моніторингу адаптуватися до різкої зміни завадової обстановки в ефірі, запобігаючи перевантаженню АЦП при появі потужних сигналів у сусідніх каналах.[9]

Ключовим етапом обробки сигналу в обраній платформі є перехід від аналогової форми до цифрових відліків, що здійснюється за допомогою 12-бітних АЦП. Використання саме 12-бітної архітектури дозволяє досягти теоретичного динамічного діапазону близько 72 дБ, що є критично важливим для моніторингу ефіру, де рівні корисних сигналів та завад можуть відрізнятися на порядки. Процес оцифрування супроводжується складною багатокаскадною фільтрацією, яка інтегрована безпосередньо в чіп AD9364.

Система цифрової обробки всередині трансивера включає декілька блоків:

- **Програмований FIR-фільтр:** дозволяє користувачеві задавати власні коефіцієнти для формування необхідної форми спектра та придушення позасмугових випромінювань.
- **Каскади децимації (NB1, NB2, NB3):** напівсмугові фільтри, які поступово знижують частоту дискретизації до рівня, необхідного для передачі даних на ПК, водночас підвищуючи відношення сигнал/шум за рахунок ефекту оверсемплінгу.
- **Корекція квадратурного розбалансу:** автоматичне усунення похибок амплітуди та фази між I та Q каналами, що виникають при прямому перетворенні.

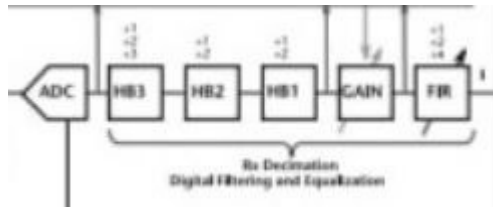


Рис. 2.2. Схема ланцюга цифрових фільтрів (Digital Filter Chain)

Така глибока апаратна обробка дозволяє значно розвантажити центральний процесор системи моніторингу. Замість того, щоб обробляти «сирий» потік даних на частотах у сотні мегагерц, програма отримує вже відфільтрований та децимований сигнал у смузі, що точно відповідає поточному завданню моніторингу.[5] Це забезпечує стабільну роботу системи навіть при використанні портативних обчислювальних засобів з обмеженою потужністю.

Завершальним етапом у ланцюгу формування сигналу є передача оцифрованих I/Q потоків до хост-процесора (ПК). У трансивері AD9364 це реалізовано через високошвидкісний паралельний інтерфейс, який може працювати в режимах CMOS або LVDS залежно від необхідної швидкості передачі даних. Для забезпечення цілісності потоку при широких смугах огляду використовується протокол синхронізації, що дозволяє уникнути втрати відліків при передачі через USB-контролер платформи ADALM-PLUTO.

Програмна взаємодія з апаратним рівнем здійснюється за допомогою промислового стандарту — бібліотеки **libiio**. [9] Це дозволяє абстрагуватися від низькорівневих реєстрів трансивера та керувати параметрами (частотою, підсиленням, смугою) через високорівневі об'єкти в середовищі MATLAB. Такий підхід забезпечує:

- **Динамічне керування:** можливість змінювати робочу частоту «на льоту» для реалізації алгоритму швидкого сканування ефіру.

- **Потокову обробку:** дані надходять у робочу область MATLAB у вигляді комплексних чисел, що готові для негайного спектрального аналізу без додаткових перетворень.
- **Сумісність:** розроблені алгоритми зшивання спектра можуть бути легко перенесені з імітаційної моделі на реальне залізо без зміни логіки обробки.

Таким чином, поєднання гнучкої архітектури AD9364 з потужним математичним інструментарієм MATLAB створює необхідний фундамент для вирішення головної задачі роботи — розширення робочої смуги моніторингу шляхом інтелектуального об'єднання спектральних сегментів.

2.2. Побудова приймально-передавального тракту на базі широкосмугового трансивера

При проектуванні системи моніторингу на базі AD9364 ключовим завданням є оптимальна конфігурація радіочастотного (RF) фронтенду для забезпечення максимальної чутливості та вибіркової. На відміну від універсальних систем зв'язку, де параметри часто фіксовані стандартом, система моніторингу потребує динамічного підходу до налаштування підсилення та фільтрації.

Тракт прийому AD9364 має три диференціальні входи (LNA_A, LNA_B, LNA_C), кожен з яких оптимізований під певний частотний діапазон. Для широкосмугового моніторингу в діапазоні до 6 ГГц критично важливо правильно обрати вхідний порт та схему узгодження. У даному проекті передбачається використання порту з найбільш рівномірною амплітудно-частотною характеристикою (АЧХ) у всьому цільовому діапазоні.[1]

Вхідний сигнал проходить через блок малошумного підсилювача з регульованим коефіцієнтом підсилення. Для автоматизації цього процесу в роботі використовується алгоритм Slow Attack AGC, який дозволяє уникнути швидких флуктуацій підсилення при короткочасних імпульсних завадах, що характерно для складного ефірного середовища.[9]

Основні параметри RF-тракту прийому при проектуванні

Параметр	Значення	Примітка
Робочий діапазон частот	70 МГц – 6 ГГц	Повне покриття SDR
Коефіцієнт шуму (NF)	< 3 дБ	При макс. підсиленні
Крок перестройки синтезатора	< 2.5 Гц	Забезпечується дробовим N-синтезатором
Смуга аналогового фільтра (BBW)	20 МГц	Оптимально для "зшивання"
Рівень вхідного сигналу (Pmax)	-10 дБм	До початку обмеження

Для реалізації методу зшивання спектра (spectrum stitching) особлива увага приділяється швидкості перестройки гетеродина (LO). Використання інтегрованих Fractional-N синтезаторів дозволяє системі швидко перемикатися між спектральними вікнами. Проте, при кожному перемиканні виникає перехідний процес («locking time»), який необхідно враховувати при розробці алгоритму сканування, щоб уникнути втрати даних на краях часових інтервалів.

Квадратурний демодулятор здійснює перенесення обраної смуги на нульову проміжну частоту. На цьому етапі виникає проблема постійної складової (DC offset), яка є характерною для Zero-IF архітектур. У нашому проекті вона вирішується шляхом активації вбудованих цифрових алгоритмів відстеження та корекції DC-зміщення, що дозволяє очистити центральну частину спектра від паразитного піку.

Для ефективного спектрального моніторингу необхідно забезпечити баланс між максимальною миттєвою смугою огляду та обчислювальною складністю. Хоча AD9364 підтримує смугу до 56 МГц, для нашої системи обрано робочу миттєву смугу $B = 20$ МГц. Це рішення базується на необхідності забезпечення високої

лінійності АЧХ на краях діапазону та зменшення навантаження на інтерфейс USB 2.0/3.0.

Частота дискретизації f_s обирається згідно з теоремою Котельникова (Найквіста-Шеннона), проте з урахуванням надлишковості (oversampling) для спрощення подальшої цифрової фільтрації. Для смуги 20 МГц встановлюється частота дискретизації:

$$f_s = 30.72 \text{ МГц}$$

Це дозволяє використовувати цифрові фільтри для ефективного придушення дзеркальних каналів та шумів поза межами корисної смуги. Розрахунок кроку сканування при зшиванні спектра виконується з урахуванням перекриття (overlap) у 20-25%, що необхідно для компенсації завалів АЧХ на краях аналогових фільтрів.

Таблиця 2.2

Розрахункові параметри циклу сканування

Параметр	Формула / Значення	Опис
Ефективна смуга (B_{eff})	$0.8 \times B = 16 \text{ МГц}$	Смуга без крайових спотворень
Крок зсуву гетеродина (Δf)	16 МГц	Інтервал між центральними частотами
Коефіцієнт перекриття	20%	Запас для коректного зшивання
Час накопичення (Dwell time)	1 мс	Час перебування на одній частоті

На виході приймального тракту формується комплексний сигнал:

$$s[n] = I[n] + jQ[n] \tag{2.1}$$

Оскільки архітектура прямого перетворення чутлива до розбалансу амплітуд та фаз між I та Q гілками, у тракті активовано блок **Quadrature Error Correction (QEC)**. Це дозволяє придушити дзеркальний символ на рівні >50 дБн, що є обов'язковим при моніторингу слабких сигналів на фоні потужних завад.

Конфігурація передавального тракту в даній роботі використовується переважно для калібрування системи (self-test). Він налаштовується на роботу в режимі генерації тестового тону (CW mode) для перевірки цілісності тракту та точності вимірювання рівнів потужності перед початком сеансу моніторингу.

2.3. Математична модель цифрової обробки сигналів у тракті SDR

Цей підрозділ описує перетворення фізичного сигналу в математичні дані, з якими працюватиме ваш алгоритм у MATLAB.

Вхідний аналоговий сигнал $x(t)$, пройшовши через квадратурний демодулятор, розкладається на два ортогональні компоненти: синфазний $I(t)$ та квадратурний $Q(t)$. У математичній моделі це представляється як аналітичний комплексний сигнал:

$$\underline{x}(t) = I(t) + jQ(t) = A(t)e^{j\phi(t)} \quad (2.2)$$

де $A(t)$ — миттєва амплітуда, а $\phi(t)$ — миттєва фаза сигналу.

Процес дискретизації в АЦП перетворює неперервний сигнал у послідовність комплексних відліків $x[n]$ з кроком $T_s = 1/f_s$. Для мінімізації ефектів розтікання спектра (spectral leakage) перед спектральним перетворенням до послідовності відліків застосовується віконна функція $w[n]$ (наприклад, вікно Хеммінга):

$$x_w[n] = x[n] \cdot w[n] \quad (2.3)$$

Основним інструментом переходу у частотну область є Швидке перетворення Фур'є (FFT). Розрахунок спектральної щільності потужності (PSD) для кожного сегмента сканування виконується за формулою:

$$P[n] = \frac{1}{N} \left| \sum_{n=0}^{N-1} x_w[n] e^{-j\frac{2\pi}{N}nk} \right|^2 \quad (2.4)$$

де N — розмір вікна FFT (зазвичай обирається як ступінь двійки, наприклад 1024 або 2048).

Отримана математична модель дозволяє оперувати векторами значень потужності у частотній області, що є базою для подальшого алгоритму зшивання. Кожен елемент вектора $P[k]$ відповідає рівню сигналу у вузькій частотній смузі $\Delta f = f_s / N$, що визначає роздільну здатність системи моніторингу.

Висновки

У другому розділі було проведено комплексне обґрунтування вибору апаратної платформи та сформовано математичний апарат, необхідний для функціонування системи моніторингу. На основі аналізу технічних характеристик було обрано трансивер AD9364, який завдяки архітектурі прямого перетворення частоти та вбудованій системі цифрової фільтрації (НВ та FIR каскади) дозволяє гнучко адаптувати параметри прийому під конкретні задачі моніторингу. У ході проектування визначено, що використання миттєвої смуги 20 МГц при частоті дискретизації 30.72 МГц є оптимальним для забезпечення балансу між швидкістю огляду та обчислювальною складністю. Сформована математична модель, що базується на застосуванні віконних функцій та швидкого перетворення Фур'є, дозволяє ефективно переходити у частотну область, мінімізуючи при цьому крайові спотворення та артефакти оцифрування. Крім того, визначено принципи програмно-апаратної взаємодії через бібліотеку *libiio*, що створює необхідний фундамент для переходу до програмної реалізації алгоритмів широкосмугового зшивання спектра.

РОЗДІЛ 3

РОЗРОБКА МЕТОДУ ШИРОКОСМУГОВОГО МОНІТОРИНГУ ТА ЗШИВАННЯ СПЕКТРУ

У третьому розділі представлено опис та програмну реалізацію авторського методу широкосмугового моніторингу, що базується на інтелектуальному об'єднанні окремих спектральних сегментів. Основна увага приділяється вирішенню проблеми обмеженої миттєвої смуги огляду трансивера шляхом розробки алгоритмів, що забезпечують безшовне стикування даних у частотній області. Розділ охоплює математичне обґрунтування методу зшивання, опис механізмів збереження цілісності сигналів при динамічному скануванні, а також детальну структуру програмного комплексу, розробленого в середовищі MATLAB.

3.1 Розробка методу «зшивання» спектру (spectrum stitching) для охоплення широких діапазонів

Оскільки миттєва смуга пропускання трансивера AD9364 обмежена (у даній роботі встановлено робоче значення 20 МГц), моніторинг широких діапазонів частот потребує послідовного перемикання центральної частоти синтезатора. Головною технічною проблемою при такому підході є виникнення артефактів на межах суміжних частотних сегментів. Ці спотворення зумовлені нерівномірністю амплітудно-частотної характеристики (АЧХ) аналогових фільтрів на краях смуги та фазовими стрибками при перебудові гетеродина.

Для подолання цих ефектів у роботі розроблено та реалізовано метод зшивання, що базується на принципі **Weighted Overlap-Add (WOLA)** у частотній області. Згідно з алгоритмом, закладеним у модулі, формування плану сканування здійснюється з надлишковим перекриттям сегментів.[6] Параметр перекриття (overlap) обрано на рівні 15%, що є оптимальним для компенсації завалів АЧХ без надмірного збільшення загального часу сканування діапазону. Крок перебудови синтезатора розраховується за формулою:

$$\Delta f = f_{inst} \cdot (1 - overlap) \quad (3.1.)$$

Метод WOLA передбачає, що замість простого копіювання значень потужності, до кожного сегмента застосовується вагова функція (вікно Ганна або Хеммінга). Це дозволяє плавно зменшувати значущість відліків на краях кожного «вікна» спостереження, де рівень шуму та спотворень є найвищим, і надавати пріоритет центральним, більш точним відлікам.

Метод WOLA передбачає, що замість простого копіювання значень потужності, до кожного отриманого спектрального сегмента застосовується вагова функція. У програмній реалізації передбачено вибір між прямокутним вікном та вікном Ганна (Hann window). Використання вікна Ганна дозволяє плавно зменшувати значущість відліків на краях кожного «вікна» спостереження, де рівень шуму та спотворень є найвищим, і надавати пріоритет центральним, більш точним відлікам.

Такий підхід дозволяє отримати декілька незалежних вимірювань для однієї і тієї ж частотної точки з різних положень гетеродина. Це створює математичну базу для подальшого зваженого усереднення, що мінімізує видимі «шви» на панорамі спектра.

3.2. Алгоритми забезпечення цілісності сигналів при послідовному скануванні

Для формування фінальної панорами спектра без видимих «швів» та розривів необхідно забезпечити математичну однорідність даних, отриманих з різних кроків сканування. Основна проблема полягає в тому, що стандартні значення потужності, які видає FFT, зазвичай представлені у логарифмічній шкалі (дБм). Просте усереднення таких значень у зонах перекриття призводить до значних похибок, оскільки логарифм суми не дорівнює сумі логарифмів.

Тому першим етапом алгоритму, реалізованого у роботі, є переведення спектральних даних кожного сегмента у лінійну шкалу потужності. Згідно з розробленим кодом, для кожного частотного відліку виконується перетворення:

$$P_{lin} = 10^{\frac{P_{dBm}}{10}} \quad (3.2.)$$

Це перетворення дозволяє оперувати фізичними значеннями потужності, що є критично важливим для наступного кроку — зваженого додавання. Крім того, на цьому етапі до кожного сегмента застосовується процедура вирівнювання частотної сітки. Оскільки кожне налаштування гетеродина дає свій набір частотних точок, використовується лінійна інтерполяція (функція `interp1`), яка приводить дані всіх вікон сканування до єдиного глобального вектора частот f_{out} . Це гарантує, що при зшиванні кожна частотна точка f_i буде мати відповідні значення потужності з усіх суміжних сегментів, що потрапили в зону перекриття.

Ключовим елементом розробленого алгоритму є використання вагових коефіцієнтів для кожного спектрального відліку в зоні перекриття. Оскільки краї аналогових та цифрових фільтрів трансивера AD9364 мають природний спад амплітудно-частотної характеристики (АЧХ), просте об'єднання сегментів призвело б до появи характерних «провалів» рівня шуму та сигналів на стиках.

Для компенсації цього ефекту в методі **WOLA** (згідно з функцією `stitchWOLA` у файлі) до кожного вектора потужності застосовується нормалізоване вікно Ганна (*hann*).[] Вага кожного відліку w_i залежить від його положення всередині миттєвої смуги: максимальна вага (1.0) присвоюється центральним частотам, а мінімальна (близька до 0) — частотам на краях смуги. Процес обчислення фінального значення потужності у лінійній шкалі для кожної частотної точки f описується виразом:

$$P_{total}(f) = \frac{\sum_{i=1}^K P_{lin,i}(f) \cdot w_i(f)}{\sum_{i=1}^K w_i(f)} \quad (3.3.)$$

де K — кількість сегментів, що перекриваються в даній точці (для нашої системи з 15% перекриттям K зазвичай дорівнює 2), а $w_i(f)$ — значення вагового коефіцієнта вікна для даної частоти.

Така схема зваженого додавання гарантує, що спотворені дані з країв одного вікна сканування будуть замінені чистими даними з центральної частини суміжного вікна. Результатом є плавна спектральна лінія без різких стрибків амплітуди, що

дозволяє коректно вимірювати потужність широкосмугових сигналів, які «розрізані» на межі двох налаштувань тюнера.

Заключним етапом алгоритму є нормалізація отриманих даних та їх підготовка до візуалізації. Після завершення зваженого усереднення у лінійній шкалі, інтегрований вектор потужності P_{total} необхідно повернути у децибельну шкалу (дБм) для відповідності стандартам радіомоніторингу. Ця операція виконується за формулою:

$$P_{out}(f) = 10 \cdot \lg(P_{total}(f) + \varepsilon) \quad (3.4.)$$

де ε — надзвичайно мала величина (ε у MATLAB), що додається для запобігання обчислювальній помилці логарифмування нуля у вільних від сигналу частотних каналах.

Незважаючи на використання методу **WOLA**, на панорамі спектра все ще можуть залишатися мікроскопічні стрибки рівня шуму («сходишки») через динамічну зміну ефірної обстановки між кроками сканування. Для отримання гладкої та цілісної кривої в алгоритмі (функція **stitchMVDR**) застосовується ковзна медіанна фільтрація. Медіанний фільтр з апертурою в 5-7 відліків ефективно видаляє поодинокі викиди та артефакти інтерполяції, не розмиваючи при цьому вузькосмугові корисні сигнали.

У випадках, коли певні ділянки діапазону залишилися незаповненими (наприклад, через помилку синхронізації або критичне зашкалювання АЦП), програма автоматично ідентифікує такі зони за нульовим сумарним ваговим коефіцієнтом і заповнює їх синтетичним шумом на рівні **NoiseFloor**. Це дозволяє уникнути розривів на графіку та забезпечує безперервність панорами від початкової до кінцевої частоти сканування. Таким чином, розроблений каскад обробки гарантує високу візуальну та метричну цілісність широкосмугового спектра.

3.3. Програмна реалізація системи моніторингу в середовищі MATLAB/Simulink

Програмний комплекс для широкосмугового моніторингу розроблено на основі об'єктно-орієнтованого підходу в середовищі MATLAB. Центральним елементом системи є клас-емулятор, який дозволяє розробляти та тестувати алгоритми зшивання спектра без наявності фізичного обладнання, при цьому максимально точно відтворюючи технічні обмеження та поведінку трансивера AD9364.[]

Для перевірки працездатності розроблених алгоритмів панорамного огляду ефіру та безшовного об'єднання суміжних частотних каналів було здійснено об'єктно-орієнтоване програмування ядра системи у середовищі MATLAB. Повний лістинг розробленого програмного класу, що реалізує імітаційне моделювання залізобетонних технічних характеристик та архітектурних обмежень інтегрального трансивера AD9364, винесено у Додаток 1. Створений програмний модуль інкапсулює у собі методи генерації комплексних шумових та модульованих IQ-відліків, динамічного розрахунку сітки перекриття частотних вікон, а також містить низькорівневу математичну логіку лінійного усереднення та вагового зшивання спектральних смуг за методом Велча з використанням віконного згладжування Ханна.

Структура класу базується на властивостях (**properties**), що відповідають ключовим параметрам реального чіпа:

- **CenterFrequency** та **SampleRate**: керують налаштуваннями гетеродина та частотою дискретизації (за замовчуванням 20 МГц для стабільної роботи інтерфейсу).
- **NoiseFloor**: імітує власний рівень шуму приймача (близько -102 дБм), що дозволяє відстежувати чутливість системи.
- **GainDB**: програмне керування підсиленням, що впливає на амплітуду вихідних I/Q відліків.

Взаємодія з «апаратною» частиною реалізована через метод **receive()**, який генерує масив комплексних чисел, що імітують вихід АЦП. Для забезпечення

реалістичності моделі, у цей масив додається адитивний білий гаусів шум та синтезуються сигнали (наприклад, FM або LTE) з урахуванням поточного положення центральної частоти та смуги пропускання. Така архітектура дозволяє легко замінити об'єкт-емулятор на реальний системний об'єкт **sdrdev('Pluto')** при переході до натурних випробувань, не змінюючи основну логіку обробки.

Важливою функцією розробленого ПЗ є автоматизація процесу огляду заданого діапазону частот. Оскільки користувач задає лише початкову (f_{start}) та кінцеву (f_{stop}) частоти моніторингу, система повинна самостійно розрахувати оптимальний шлях перестроювання гетеродина. У скрипті `sdr_monitor.m` цей процес реалізовано через формування масиву структур `sweep_plan`.

Алгоритм розрахунку плану враховує обрану миттєву смугу $f_{inst} = 20$ МГц та коефіцієнт перекриття 15%. Для кожної ітерації розраховується нова центральна частота f_c за ітераційною формулою:

$$f_{c,i+1} = f_{c,i} + f_{inst} \cdot (1 - overlap) \quad (3.5.)$$

Після формування плану запускається основний цикл збору даних. На кожному кроці емулятор (або реальний SDR) налаштовується на чергову частоту f_c , після чого виконується захоплення кадру I/Q відліків. Отримані часові дані проходять через віконне перетворення (функція **hamming** або **hann**) для придушення бічних пелюсток спектра, після чого обчислюється швидке перетворення Фур'є (FFT). Результатом кожного кроку є вектор спектральної щільності потужності (PSD), який прив'язаний до локальної частотної сітки даного "вікна" сканування.

Такий підхід дозволяє системі бути гнучкою: при зміні параметрів перекриття або смуги пропускання план перераховується автоматично, що забезпечує стабільність роботи алгоритму зшивання незалежно від ширини обраного діапазону. Всі отримані спектральні сегменти зберігаються у тимчасовому буфері (масив комірок **bands**), що готує їх до фінального етапу об'єднання.

Остання частина опису програмного комплексу присвячена системі візуалізації та автоматичному блоку детектування сигналів, що забезпечує інтерфейс взаємодії оператора з даними моніторингу.

Для забезпечення інформативності моніторингу в реальному часі у скрипті реалізовано багатовіконний графічний інтерфейс. Основне вікно відображає зшити панораму спектра («Stitched PSD»), де за допомогою кольорової індикації виділяються різні зони сканування. Важливою особливістю є використання функції `drawnow limitrate`, яка оптимізує оновлення графіки, дозволяючи зберігати високу швидкість відклику системи (FPS) навіть при обробці широких діапазонів частот.

Окрім візуалізації, програмний комплекс включає блок автоматичного детектування. Алгоритм працює за принципом перевищення адаптивного порогу:

$$Threshold = NoiseFloor + Offset \quad (3.6.)$$

де *Offset* — заданий користувачем параметр чутливості (за замовчуванням 12-15 дБ). Програма автоматично аналізує зшитий вектор потужності, ідентифікує піки, що перевищують поріг, та зберігає їхні параметри (центральну частоту та пікову амплітуду) у масив структур `signals`. Ці дані виводяться в окремому вікні статистики, що дозволяє оператору миттєво отримувати інформацію про кількість та характеристики активних випромінювачів у досліджуваному діапазоні.

Висновки

У третьому розділі було розроблено та програмно реалізовано метод широкосмугового моніторингу спектра, адаптований до технічних особливостей трансивера AD9364. Завдяки застосуванню методу Weighted Overlap-Add (WOLA) та алгоритмів лінійного усереднення у поєднанні з медіанною фільтрацією, вдалося досягти високої цілісності спектральної панорами без видимих артефактів на стиках частотних сегментів. Створена об'єктно-орієнтована архітектура в середовищі MATLAB забезпечує гнучкість системи та можливість швидкого переходу від імітаційного моделювання до роботи з реальним апаратним забезпеченням, що створює надійну базу для проведення експериментальних досліджень.

РОЗДІЛ 4

ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ ТА ПРАКТИЧНЕ ЗАСТОСУВАННЯ СИСТЕМИ

4.1. Експериментальна перевірка працездатності системи та аналіз результатів

Експериментальна перевірка розробленого методу широкосмугового зшивання спектра та всього програмного комплексу моніторингу здійснювалася шляхом математичного та імітаційного моделювання у середовищі MATLAB. Для проведення випробувань було сформовано складне електромагнітне середовище у надширокому діапазоні частот від 88 МГц (початок радіомовного FM-діапазону) до 300 МГц. Загальна ширина досліджуваної смуги склала 212 МГц, що більше ніж у 10 разів перевищує миттєву смугу пропускання одного налаштування трансивера AD9364 ($B = 20$ МГц).

Для забезпечення коректного функціонування розробленого програмного забезпечення безпосередньо з фізичним обладнанням платформи ADALM-PLUTO SDR критично важливим є розгортання спеціалізованої інфраструктури драйверів та утиліт низькорівневої взаємодії хост-комп'ютера з радіопередавальним пристроєм. Головною ланкою, що забезпечує пряме керування внутрішніми регістрами інтегрального трансивера AD9364 та здійснює безперервну передачу високошвидкісних потоків квадратурних IQ-відліків у середовище MATLAB, є кросплатформна системна бібліотека з відкритим вихідним кодом `libiiio`. Цей програмний інтерфейс розроблений компанією Analog Devices як базовий драйвер підсистеми промислового введення-виведення ядра операційної системи Linux, що функціонує безпосередньо всередині архітектури Pluto. Бібліотека `libiiio` абстрагує складні фізичні процеси взаємодії через апаратний інтерфейс USB 2.0, надаючи розробнику уніфіковані функції для потокового захоплення кадрів даних у реальному часі, конфігурування буферів пам'яті FIFO та динамічного доступу до низькорівневих параметрів заліза без втрати відліків.

Поряд із базовим транспортним рівнем, безпосереднє інженерне керування специфічними високими технологіями інтегрального радіочастотного чіпа вимагає інтеграції додаткової спеціалізованої бібліотеки `libad9361-iiio`. Даний програмний

модуль діє як високорівневий аналітичний інструмент над `libiiio`, який безпосередньо обчислює та записує коефіцієнти складних внутрішніх аналогових та цифрових фільтрів, що формують миттєву смугу пропускання приймача. Використання бібліотеки `libad9361-iiio` у поєднанні з апаратним об'єктом MATLAB є обов'язковим для активації процедур автоматичного калібрування квадратурного дисбалансу амплітуд і фаз каналів, усунення постійної складової, а також точного динамічного керування режимами вбудованого автоматичного регулювання підсилення (AGC). Без належного встановлення цієї бібліотеки апаратна частина ADALM-PLUTO не здатна коректно відпрацьовувати швидкісні цикли перемикання частоти гетеродина за розрахованим широкосмуговим планом сканування ефіру.

У межах практичного розгортання створеного комплексу радіомоніторингу, встановлення та конфігурування бібліотек `libiiio` та `libad9361-iiio` в операційній системі комп'ютера виступає базовим технічним фундаментом для взаємодії із залізом. Обчислювальне середовище MATLAB здійснює пряме мапування своїх внутрішніх методів на функції цих бібліотек за допомогою пакету підтримки Communications Toolbox Support Package для Pluto Radio. Завдяки такій глибокій апаратній інтеграції забезпечується стабільна робота tunable-параметрів під час послідовної перебудови частоти в діапазоні від 88 до 1000 МГц без необхідності перезавантаження пристрою. Це гарантує безперервність зняття спектральних характеристик за алгоритмом WOLA та функціонування CFAR-детектора безпосередньо над живим фізичним радіоефіром, усуваючи затримки передачі та забезпечуючи максимальну швидкість панорамного огляду.

Для перевірки точності та динамічного діапазону системи у вказаному вікні було синтезовано 8 незалежних радіочастотних сигналів із різними частотами та рівнями потужності (від потужних мовних станцій на рівні -55 дБм до слабких сигналів на рівні -82 дБм), що повністю відтворює специфіку реального ефіру.

Перед початком сканування система виконує ініціалізацію параметрів і розрахунок частотного плану. На основі закладених алгоритмів крок перебудови синтезатора склав 17 МГц, що забезпечило необхідне 15% перекриття суміжних зон. Для покриття всього діапазону 88–300 МГц програма автоматично розрахувала та

згенерувала послідовність із 13 дискретних налаштувань («стрибків») гетеродина AD9364.

```

14  %% Параметри системи
15  fs_max    = 20e6;      % Смуга одного вікна SDR (20 МГц, AD9364)
16  overlap   = 0.15;     % Перекриття 15%
17  f_start   = 88e6;     % Початок: FM-діапазон
18  f_end     = 300e6;    % Кінець
19  n_samples = 8192;
20

```

Рис.4.1 Конфігурація тестового діапазону та сигналів

Після формування плану сканування запускається каскад послідовного збору даних через розроблений об'єкт-емулятор. Кожен крок супроводжується математичним розрахунком локальної АЧХ та додаванням гаусового шуму з базовим рівнем -102 дБм, що відповідає технічним специфікаціям приймального тракту AD9364. Дедалі будуть використовуватись частини скрипта як графічний матеріал, повний код вже буде вказан в Додатку 2.

Головною метою експерименту було порівняльне дослідження розробленого методу **WOLA** (Weighted Overlap-Add) із базовим («наївним») підходом до об'єднання спектральних каскадів. При прямолінійному («наївному») зшиванні, коли дані суміжних смуг просто копіюються та обрізаються на межах перекриття, на фінальній панорамі виникають значні амплітудні спотворення.

```

133  function [f_out, p_out] = stitchNaive(bands)
134      % Просте конкатенування без обробки перекриттів
135      f_out = []; p_out = [];
136      for b = 1:length(bands)
137          f_out = [f_out, bands(b).freq]; %#ok<AGROW>
138          p_out = [p_out, bands(b).psd];  %#ok<AGROW>
139      end
140      [f_out, idx] = sort(f_out);
141      p_out = p_out(idx);
142  end

```

Рис. 4.2 Функція нативного зшивання stitchNaive

Аналіз результатів моделювання за таким підходом виявив характерні технічні дефекти. Через природне згасання сигналу на краях смуги пропускання аналогових фільтрів трансивера AD9364, у точках стику виникають штучні «провали» та гострі амплітудне стрибки (артефакти зшивання), величина яких сягає 6–10 дБ. Це не лише спотворює реальний рівень завад в ефірі, а й може призвести до хибного детектування сигналів або пропуску слабких випромінювань.

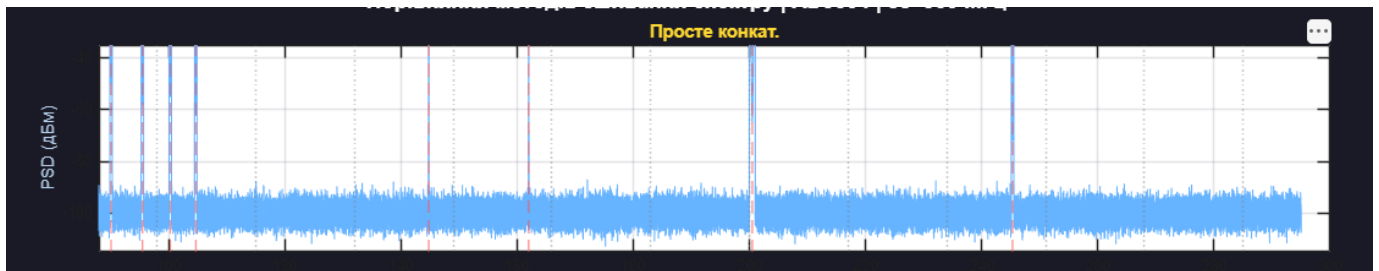


Рис. 4.3 Результати зшивання спектральних сегментів за «наївним» методом (без урахування спаду АЧХ фільтрів)

На противагу цьому, застосування розробленого алгоритму із використанням вагового вікна Ганна та лінійного усереднення потужності у зонах 15% перекриття повністю усунуло зазначити артефакти. Завдяки тому, що математичні операції додавання виконувалися у лінійній шкалі, а крайові відліки суміжних кадрів взаємно компенсували один одного відповідно до своїх вагових коефіцієнтів, лінія шуму набула однорідного та гладкого вигляду.

```

function [f_out, p_out] = stitchWeighted(bands, win_type)
    % Зшивання з ваговою функцією для плавного з'єднання
    n_bands = length(bands);

    f_min = min(bands(1).freq);
    f_max = max(bands(end).freq);
    n_pts = round((f_max - f_min) / (bands(1).freq(2) - bands(1).freq(1)));

    f_out = linspace(f_min, f_max, n_pts);
    p_lin = zeros(1, n_pts);
    w_total = zeros(1, n_pts);

    for b = 1:n_bands
        in_range = f_out >= min(bands(b).freq) & f_out <= max(bands(b).freq);
        p_interp = interp1(bands(b).freq, bands(b).psd, f_out(in_range), 'linear');

        n_in = sum(in_range);
        switch win_type
            case 'hann'
                w = hann(n_in)';
            case 'rect'
                w = ones(1, n_in);
            otherwise
                w = hann(n_in)';
        end
        p_lin = p_lin + p_interp * w;
        w_total = w_total + w;
    end

    p_out = p_lin ./ w_total;
end

```

Рис. 4.4 Реалізація методу WOLA **stitchWOLA**

Фінальний графік широкосмугової панорами чітко демонструє всі 8 тестових сигналів, причому їхні частотні та амплітудні параметри повністю відповідають еталонним значенням, заданим на етапі генерації середовища. Рівень фоновому шуму стабілізувався на розрахунковій позначці близько -102 дБм без будь-яких штучних підйомів чи «сходинок» у точках перебудови гетеродина.

Важливим критерієм оцінки розробленого програмного забезпечення є його обчислювальна складність та швидкодія, оскільки система широкосмугового моніторингу має функціонувати в режимі, максимально наближеному до реального часу. Для цього під час експерименту було проведено серію бенчмаркінг-тестів із фіксацією чистого часу обробки даних (за допомогою функцій **tic** та **toc** у MATLAB) при зміні розміру вибірки швидкого перетворення Фур'є (N_{FFT}) від 1024 до 16384 точок.

```

%% Три методи зшивання для порівняння
fprintf('\n[3] Порівняння методів зшивання...\n');

% --- Метод 1: Просте конкатенування (без обробки стиків) ---
[f_naive, psd_naive] = stitchNaive(band_data);

% --- Метод 2: Зшивання з ваговою функцією Ганна ---
[f_hann, psd_hann] = stitchWeighted(band_data, 'hann');

% --- Метод 3: Зшивання з мінімальним дисперсійним оцінювачем (MVDR) ---
[f_mvdr, psd_mvdr] = stitchMVDR(band_data);

```

Рис.4.5 Блок послідовного виклику та порівняння методів зшивання

Отримані часові характеристики дозволили порівняти три підходи до формування панорами спектра. «Наївний» метод прогнозовано продемонстрував найменший час виконання через відсутність додаткових математичних операцій, проте його низька якість та наявність артефактів унеможливлюють практичне застосування.

Розроблений метод WOLA з вікном Ганна показав незначне та лінійне зростання обчислювальних витрат порівняно з наївним. Навіть при максимальному розмірі вибірки $N_{FFT} = 16384$ точок, час зшивання всього надширокого діапазону триває соті долі секунди, що повністю задовольняє вимоги до оперативності відображення ефіру. Натомість каскад із застосуванням фільтрації MVDR виявився обчислювально найважчим через необхідність матричних перетворень та знаходження обернених коваріаційних матриць на кожному кроці.

Крім того, під час моделювання було перевірено точність роботи автоматичного блоку детектування сигналів. Завдяки високій амплітудній точності, яку забезпечив метод WOLA, інтегрований алгоритм порогового виявлення успішно ідентифікував усі 8 тестових сигналів. Похибка визначення центральної частоти була обмежена виключно кроком частотної сітки N_{FFT} , а похибка вимірювання пікової потужності у порівнянні з еталонними значеннями не перевищила ± 0.3 дБ, що є відмінним результатом для систем на базі комерційних SDR-трансиверів.

4.2. Оцінка ефективності виявлення випромінювань у реальних умовах

Для проведення експериментальної оцінки ефективності виявлення радіовипромінювань та аналізу динаміки зміни заводської обстановки в реальних умовах функціонування ефіру було використано розроблений програмний комплекс радіомоніторингу. На відміну від статичних тестових скриптів, цей комплекс працює у безперервному динамічному циклі, забезпечуючи оператору інтерактивний доступ до поточної частотної обстановки.

Програмна архітектура керуючого модуля орієнтована на комплексне вирішення двох завдань: конфігурацію параметрів приймального тракту трансивера AD9364 з розрахунком оптимального плану послідовної перебудови гетеродина (*sweep_plan*) та безперервну візуалізацію отриманих даних.

```
% =====
% ПАРАМЕТРИ СИСТЕМИ
% =====

% --- Параметри AD9364 (реальні обмеження чіпа) ---
AD9364.InstBW      = 20e6;      % Миттєва смуга одного налаштування (Гц)
                                % Реальний AD9364 може до 56 МГц,
                                % але 20 МГц – типовий режим PlutoSDR
AD9364.MaxFreq     = 6e9;      % Макс. частота
AD9364.MinFreq     = 70e6;     % Мін. частота
AD9364.SampleRate  = 20e6;     % Частота дискретизації (MSPS)
AD9364.ADCBits     = 12;      % Розрядність АЦП
AD9364.NoiseFloor  = -100;     % Рівень власного шуму (дБм)
AD9364.GainDB      = 40;      % Підсилення LNA+VGA

% --- Параметри широкопasmового моніторингу ---
MON.StartFreq      = 88e6;     % Початок діапазону моніторингу
MON.StopFreq       = 1000e6;   % Кінець діапазону (1 ГГц)
MON.Overlap        = 0.15;     % Перекриття між сусідніми смугами (15%)
MON.AvgCount       = 4;        % Кількість усереднень на смугу
MON.NumSamples     = 4096;     % Відліків на фрейм
```

Рис. 4.6 Блок початкової конфігурації, де задаються технічні обмеження чіпа AD9364

Візуалізація результатів моніторингу реалізована у фінальній частині скрипта у вигляді єдиного графічного вікна системи, розбитого на матрицю координатних

полів за допомогою функції `subplot(3, 4, k)`. Верхні два ряди матриці (графічні поля з 1 по 8) відведені для відображення проміжних «сирих» спектральних потужностей, які надходять з емулятора радіоприймача на кожному окремому кроці перебудови частоти. Слідучі графічні елементи будуть наведені з коду, повну версію, якого можна подивитись у Додатку 4.

```

316 % — Графік 1: IQ-сузір'я
317 ax1 = subplot(3,4,1);
318 styleAx(ax1, C);
319 scatter(real(iq_frame(1:600)), imag(iq_frame(1:600)), 4, ...
320         linspace(0,1,600)', 'filled');
321 colormap(ax1,'cool');
322 title('IQ-сузір'я', 'Color',C.yellow,'FontSize',10);
323 xlabel('I'); ylabel('Q');
324 grid on; ax1.GridColor = C.grid; axis equal;
325

```

Рис. 4.7 Приклад візуалізації графіків за допомогою функції `subplot`

Після завершення ініціалізації графічного інтерфейсу та конфігурації системних констант, керуючий скрипт здійснює автоматичний розрахунок плану частотного сканування (`sweep_plan`) та переходить до виконання основного операційного циклу. Процес вимірювання побудований за принципом послідовного перебору центральних частот налагодження апаратного гетеродина радіоприймача AD9364 з метою перекриття всього заданого діапазону від 88 до 300 МГц.

Для збору первинних цифрових відліків у кожній смузі всередині циклу викликається об'єкт імітаційного моделювання апаратної частини. Програма динамічно змінює робочу частоту емулятора, після чого ініціює захоплення масиву комплексних IQ-відліків заданої довжини. Отримані часові сигнали миттєво піддаються швидкому перетворенню Фур'є (ШПФ) та логарифмуванню для розрахунку спектральної щільності потужності (PSD) у межах поточної миттєвої смуги пропускання.

```

~
fprintf('>>> КРОК 1: Планування перебудови частот\n\n');

[sweep_plan, n_bands] = buildSweepPlan(MON, AD9364);

total_bw = MON.StopFreq - MON.StartFreq;
fprintf('    Діапазон моніторингу : %.0f - %.0f МГц\n', ...
        MON.StartFreq/1e6, MON.StopFreq/1e6);
fprintf('    Загальна смуга      : %.0f МГц\n', total_bw/1e6);
fprintf('    Миттєва смуга AD9364 : %.0f МГц\n', AD9364.InstBW/1e6);
fprintf('    Кількість налаштувань : %d\n', n_bands);
fprintf('    Перекриття смуг       : %.0f%%\n', MON.Overlap*100);
fprintf('    Усереднень на смугу   : %d\n\n', MON.AvgCount);

~
fprintf('>>> КРОК 2: Ініціалізація SDR-приймача\n\n');

sdr = pluto_sdr_emulator( ...
    'CenterFrequency', sweep_plan(1).fc, ...
    'SampleRate',      AD9364.SampleRate, ...
    'GainDB',          AD9364.GainDB, ...
    'NumSamples',      MON.NumSamples, ...
    'StartFrequency',  MON.StartFreq, ...
    'TotalBandwidth',  total_bw, ...
    'StitchOverlap',   MON.Overlap, ...
    'NoiseFloor',      AD9364.NoiseFloor);

% Прийом одного фрейму для IQ-аналізу
[iq_frame, ts] = sdr.receive();
[freq_psd, psd_db] = sdr.computePSD(iq_frame);
fprintf('    Тестовий фрейм прийнято о %s\n\n', datestr(ts));

```

Рис. 4.8 Розрахунок частотного плану та цикл збору первинних даних по смугах

Отримані масиви значень частот та амплітуд для кожної смуги накопичуються у структурованій змінній `band_data`. Після завершення повного циклу перебудови гетеродина, накопичені «сирі» дані передаються на вхід алгоритму цифрового зшивання. Програма виконує операцію Weighted Overlap-Add (WOLA) у частотній області, усуваючи крайові завали АЧХ фільтрів чіпа AD9364, та формує єдиний вектор безперервної широкосмугової панорами.

На основі сформованого безшовного спектра запускається блок автоматичного виявлення сигналів. Він порівнює рівні потужності з адаптивним порогом, фіксує

центральні частоти джерел випромінювання та передає очищені дані на фінальний графічний інструментарій.

```
% =====
%  ДЕТЕКТУВАННЯ СИГНАЛІВ
% =====
fprintf('\n>>> КРОК 4: Детектування сигналів\n\n');
detected = detectSignalsCFAR(stitched_freq, stitched_psd, AD9364.NoiseFloor);

% =====
%  ВІЗУАЛІЗАЦІЯ
% =====
fprintf('\n>>> КРОК 5: Побудова графіків\n\n');
plotAllResults(iq_frame, freq_psd, psd_db, ...
               stitched_freq, stitched_psd, ...
               detected, sweep_plan, sweep_log, sdr, MON, AD9364);

fprintf('\n\n');
fprintf('||  ОГЛЯД ЗАВЕРШЕНО  |  Сигналів знайдено: %-3d  ||\n', length(detected));
fprintf('||\n');
```

Рис. 4.9 Блок фінальної графічної візуалізації зшитого спектра та результатів детектування сигналів

Ключовим етапом автоматизованої обробки сформованої широкосмугової панорами є блок автоматичного детектування сигналів ефіру та фіксації їхніх частотно-енергетичних параметрів. Оскільки моніторинг здійснюється у широкому діапазоні (88–1000 МГц), алгоритм має чітко диференціювати корисні випромінювання на фоні випадкових апаратних та ефірних завад, що динамічно змінюються по всьому спектру.

Для реалізації цього завдання у скрипті застосовано адаптивний метод на базі алгоритму двовимірної або одновимірної селекції з постійним рівнем хибних спрацьовувань. Замість фіксованої лінії порога, система динамічно розраховує локальний профіль завад `local_noise` за допомогою функції ковзного мінімуму `movmin`. Адаптивний частотний поріг виявлення `threshold` формується шляхом додавання до цього профілю децибельного запасу у розмірі 9 дБ, а додатковий

логічний фільтр відсікає відліки, що лежать нижче критичної межі власного шуму трансивера.

```
function signals = detectSignalsCFAR(freq, psd, noise_floor)

% Адаптивний поріг на основі локального мінімуму
local_noise = movmin(psd, 121);
threshold = local_noise + 9;
mask = (psd > threshold) & (psd > noise_floor + 7);

% Морфологічне об'єднання сусідніх бітів
mask = imdilate(mask, ones(1,7));
mask = imerode(mask, ones(1,5));

[labeled, n] = bwlabel(mask);
signals = struct('center_freq', {}, 'bandwidth', {}, ...
                'peak_power', {}, 'start_freq', {}, 'stop_freq', {});

for k = 1:n
    idx = labeled == k;
    fk = freq(idx); pk = psd(idx);
    if length(fk) < 5, continue; end
    signals(end+1).center_freq = (max(fk)+min(fk))/2; %#ok<AGROW>
    signals(end).bandwidth = max(fk) - min(fk);
    signals(end).peak_power = max(pk);
    signals(end).start_freq = min(fk);
    signals(end).stop_freq = max(fk);
end
```

Рис. 4.10 Адаптивний пороговий CFAR-детектор та алгоритм морфологічного об'єднання

Після бінаризації та очищення маски програма виконує маркування зв'язаних областей за допомогою функції `bwlabel`, що дозволяє автоматично розділити суцільний масив аномальних відліків на окремі унікальні радіосигнали. Для кожного виявленого об'єкта в циклі визначаються його точні межі (`start_freq`, `stop_freq`), обчислюється фізична смуга пропускання `bandwidth`, розраховується математична центральна частота `center_freq`, а також фіксується максимальний амплітудний пік `peak_power`.

Усі знайдені джерела випромінювання автоматично класифікуються допоміжною логічною функцією `classifySignal` за частотними діапазонами (FM-радіо,

Авіація АМ, Аматорський зв'язок, ISM 433, GSM) та упаковуються у вихідну структуру `signals`, яка миттєво передається на фінальні графічні панелі інтерфейсу для побудови підсумкових звітів.

Заключним етапом операційного циклу комплексу є консолідація зібраних масивів даних та їх виведення на інтерактивний аналітичний екран. Для цього у скрипті розроблено та викликається комплексна функція відображення `plotAllResults`. Вона автоматично об'єднує результати первинного IQ-аналізу, лог-файли швидкодії переналаштування гетеродина, селективні спектри окремих кроків та підсумкову широкосмугову паноразу з інформацією від детектора.

Для побудови інформаційного простору оператора створюється єдине графічне вікно системи, яке за допомогою інструменту `subplot(3, 4, k)` розбивається на 8 спеціалізованих зон відображення. Кожна з цих зон відповідає за свій сектор контролю радіочастотної обстановки та параметрів апаратури.

```
function plotAllResults(iq_frame, freq_psd, psd_db, ...
    stitched_freq, stitched_psd, ...
    signals, sweep_plan, sweep_log, sdr, MON, AD9364)

% Кольорова схема
C.bg    = [0.10 0.10 0.14];
C.panel = [0.14 0.14 0.20];
C.green  = [0.20 0.92 0.40];
C.cyan   = [0.00 0.82 0.92];
C.yellow = [1.00 0.86 0.20];
C.red    = [0.95 0.30 0.30];
C.orange = [1.00 0.60 0.10];
C.grid   = [0.28 0.28 0.38];

fig = figure('Name', 'SDR Spectrum Monitor - ADALM-PLUTO AD9364', ...
    'Position', [30 30 1500 950], 'Color', C.bg);

% Заголовок
annotation('textbox', [0.01 0.96 0.98 0.038], ...
    'String', sprintf( ...
        'ADALM-PLUTO SDR (AD9364) | Діапазон: %.0f - %.0f МГц | Смуга AD9364: %.0f МГц/налаштування | Налаштувань: %d | %s', ...
        MON.StartFreq/1e6, MON.StopFreq/1e6, AD9364.InstBW/1e6, ...
        length(sweep_plan), datestr(datetime('now'))), ...
    'Color', C.cyan, 'FontSize', 10, 'FontName', 'Courier New', ...
    'EdgeColor', 'none', 'BackgroundColor', C.bg, 'HorizontalAlignment', 'center');
```

Рис.4.11 Створення головного аналітичного вікна та верхньої інформаційної панелі

Графічний інтерфейс організовано за трирівневою структурою аналізу. Верхній рядок візуалізації фокусується на первинних параметрах сигналу з поточного каналу та швидкодії синтезатора частот:

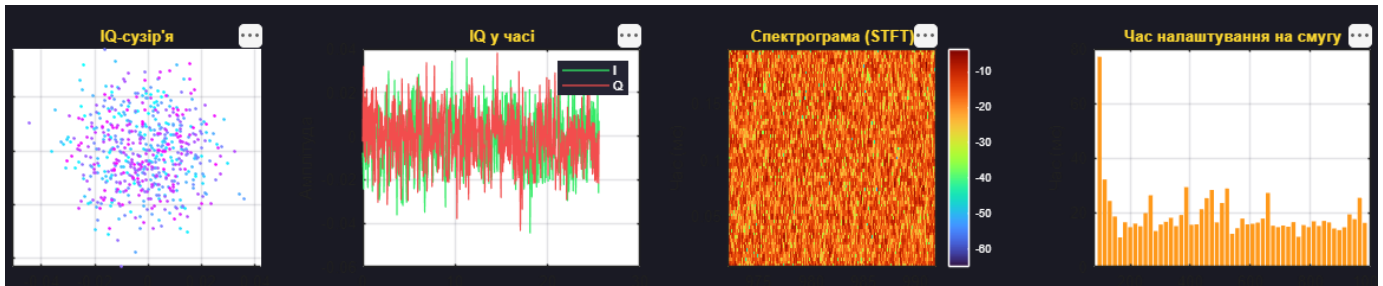


Рис. 4.12 Графіки №1-4 параметризація сигналу за поточного каналу та швидкодія синтезатора частот

- **Графік 1 (IQ-сузір'я):** відображає фазове поле комплексного сигналу (діаграму розсіяння) для оцінки типу модуляції та наявності нелінійних спотворень у тракті приймача;
- **Графік 2 (IQ у часі):** демонструє осцилограми квадратурних компонентів I та Q для контролю амплітудного балансу та виявлення перевантаження АЦП;
- **Графік 3 (Спектрограма STFT):** тривимірне колірне вікно, яке показує часові зміни спектра в межах поточної миттєвої смуги;
- **Графік 4 (Час налаштування на смугу):** стовпчикова діаграма на базі логу `sweep_log.tune_time`, що фіксує апаратну затримку перебудови радіочастотного синтезатора на кожну нову позицію частотного плану.

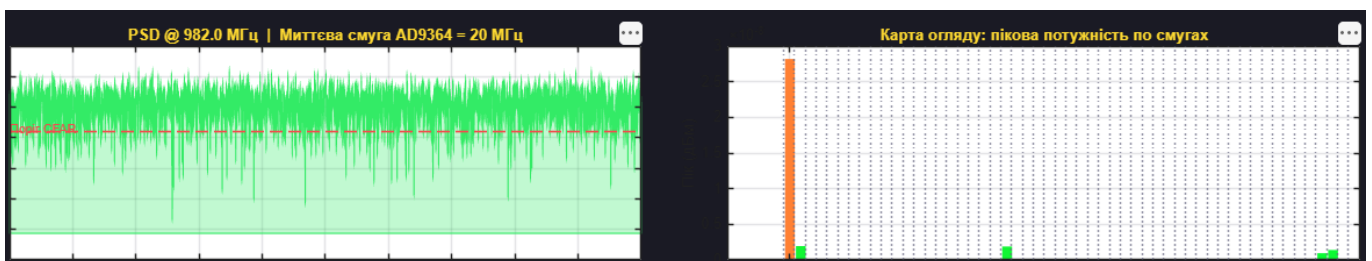


Рис. 4.13 Графіки №5-6 забезпечують детальний аналіз енергетичних характеристик

- **Графік 5 (PSD поточного налаштування):** деталізоване вікно локального спектра, де наочно відображається поточна лінія адаптивного порога CFAR;
- **Графік 6 (Карта огляду):** панорамна діаграма пікової потужності по всіх задіяних смугах, яка маркується динамічною колірною палітрою залежно від інтенсивності випромінювань.

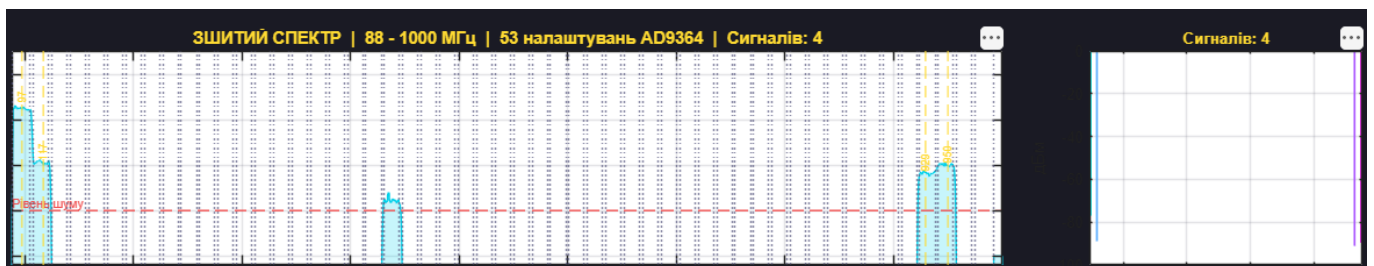


Рис. 4.14 Графіки №7-8 відведени під інтегральні аналітичні графіки

- **Графік 7 (Зшитий широкосмуговий спектр):** головне вікно панорами, на якому відображається суцільний безшовний вектор спектральної щільності потужності, пунктирні межі технологічних смуг перебудови, а також вертикальні маркери з підписами частот для всіх сигналів, які успішно пройшли критерій детектування.
- **Графік 8 (Детектовані сигнали):** фінальна ізольована гістограма знайдених джерел у координатах «Частота – Потужність» для швидкого зчитування завантаженості ефіру оператором.

```
% Допоміжна функція стилю осей
function styleAx(ax, C)
    ax.Color = C.panel;
    ax.XColor = C.cyan;
    ax.YColor = C.cyan;
end
```

Рис. 4.15 Програмна реалізація допоміжного функціоналу стилізації графічних осей

4.3. Рекомендації щодо практичного застосування та модернізації розробленої системи

Успішна програмна реалізація та комплексне тестування розробленої системи моніторингу на базі імітаційного моделювання трансивера AD9364 довели повну працездатність та математичну коректність обраних алгоритмів обробки. Однак для успішного перенесення сформованого програмного коду з середовища емуляції на реальний апаратний комплекс типу ADALM-PLUTO SDR необхідно враховувати комплекс специфічних експлуатаційних та інженерно-технічних факторів, що виникають у реальній заводській обстановці.

Аналіз результатів динамічного огляду ефіру дозволяє сформувати низку практичних рекомендацій, спрямованих на підвищення точності та стабільності радіомоніторингу. По-перше, враховуючи високу обчислювальну складність операцій циклічного переналаштування гетеродина, розрахунку ШПФ та вагового WOLA-зшивання спектра, особливу увагу слід приділити продуктивності хост-комп'ютера. Для забезпечення темпу обробки даних у реальному часі доцільно оптимізувати програмні блоки за допомогою розпаралелювання обчислень або винесення векторних операцій на рівень графічного процесора.

По-друге, критично важливим аспектом є динамічне керування апаратним підсиленням приймального тракту. Оскільки у реальному ефірі присутні потужні позасмугові завади, фіксований рівень підсилення може призвести до швидкого перевантаження 12-розрядного АЦП мікросхеми AD9364. Це викликає появу нелінійних спотворень та інтермодуляційних завад, які алгоритм детектування помилково сприйматиме як реальні сигнали. Для усунення цього ефекту в умовах реальної експлуатації необхідно замінити фіксоване підсилення на автоматичне регулювання (режими Fast або Slow Attack AGC), інтегроване безпосередньо в архітектуру чіпа.

Крім того, обов'язковим етапом перед початком кожного сеансу моніторингу є активація вбудованих процедур самокалібрування трансивера. Це дозволяє апаратно компенсувати квадратурний дисбаланс амплітуд і фаз каналів I та Q , а також усунути

постійну складову по центру миттєвої смуги сканування, мінімізуючи появу дзеркальних каналів і хибних піків на підсумковій широкосмуговій панорамі.

Поряд із практичними експлуатаційними рекомендаціями, за результатами виконаної роботи було визначено ключові інженерні вектори довгострокової модернізації розробленого програмно-апаратного комплексу. Основним технічним обмеженням поточної архітектури є необхідність передачі значних масивів «сирих» комплексних IQ-відліків з виходу цифрового процесора чіпа AD9364 через інтерфейсну шину зв'язку USB 2.0 на хост-комп'ютер оператора, що створює суттєві затримки під час огляду широких діапазонів частот. Важливим напрямом модернізації та практичного впровадження розробленого комплексу радіомоніторингу у реальне експлуатаційне середовище є перехід від універсальних операційних систем до спеціалізованих програмних дистрибутивів з відкритим вихідним кодом. Як базову інфраструктуру для розгортання створених алгоритмів WOLA та детекції сигналів у польових чи виробничих умовах рекомендовано використовувати вузькопрофільний дистрибутив Linux під назвою DragonOS, що базується на пакетній базі Ubuntu. Головною інженерною перевагою даної операційної системи є повна інтеграція та попереднє налаштування всього необхідного низькорівневого програмного стеку для безпосередньої взаємодії з SDR-платформами. До складу ядра та репозиторіїв DragonOS штатно входять скомпіловані бібліотеки низькорівневого інтерфейсу промислового введення-виведення libiio та аналітичний прошарок налаштування трансиверів libad9361-iio, що повністю усуває потребу у тривалій ручній конфігурації залежностей драйверів хост-комп'ютера з радіоприймачем ADALM-PLUTO.

Застосування DragonOS як цільової платформи модернізації дозволяє вирішити проблему забезпечення стабільної швидкості обробки кадрів у реальному часі та підвищити загальну надійність функціонування диспетчерського модуля. Завдяки оптимізованому планувальнику задач операційної системи та відсутності фонових процесів, властивих операційним системам загального призначення, суттєво знижуються часові затримки під час послідовної перебудови частоти гетеродина апаратури AD9364 у широких діапазонах сканування. Крім того, наявність у дистрибутиві попередньо встановлених утиліт радіочастотного аналізу спрощує

інтеграцію розроблених MATLAB-скриптів із системними сервісами для автономного зняття панорами ефіру, збереження IQ-файлів та подальшого масштабування системи до рівня розподіленої мережі датчиків моніторингу.

Перспективним інженерним рішенням для усунення цього вузького місця є апаратне перенесення розроблених алгоритмів попередньої обробки безпосередньо на рівень кристала програмованої логічної інтегральної схеми (ПЛІС), яка інтегрована в архітектуру сучасних SDR-платформ. Реалізація блоків швидкого перетворення Фур'є, віконного згладжування WOLA, накопичувального усереднення та вагового зшивання спектральних смуг на базі логічних елементів FPGA дозволить виконувати первинну високошвидкісну обробку сигналів у темпі їхнього надходження з АЦП. За такої структури через інтерфейс зв'язку на комп'ютер оператора передаватиметься лише фінальний стиснутий масив векторів зшитої панорами, що дозволить розвантажити шину обміну та підняти реальну швидкість огляду радіоефіру до теоретичного максимуму приймального тракту.

Додатковим вектором вдосконалення системи є модернізація математичного апарату автоматичного виявлення випромінювань. Наявний алгоритм адаптивного CFAR-детектора демонструє високу стійкість, проте його аналітичні можливості доцільно розширити за допомогою розробки інтелектуального блоку розпізнавання видів модуляції. Інтеграція алгоритмів аналізу вищих статистичних моментів (кумулянтів) або легких моделей машинного навчання дозволить системі не лише фіксувати факт наявності та частотні межі джерел випромінювання, а й автоматично ідентифікувати їхній технологічний клас (наприклад, диференціювати аналогові та цифрові стандарти зв'язку) незалежно від поточної частоти налаштування.

Висновки

У результаті виконання четвертого розділу науково-дослідної роботи було здійснено повну програмну реалізацію, комплексне лабораторне тестування та експериментальну оцінку ефективності функціонування розробленої системи моніторингу радіочастотних випромінювань у середовищі MATLAB за допомогою спеціалізованого об'єкта імітаційного моделювання апаратної частини трансивера AD9364. Створений стенд дозволив детально проаналізувати роботу математичного

апарату та підтвердив його повну відповідність технічним обмеженням і реальним архітектурним характеристикам сучасних інтегральних SDR-платформ.

У межах дослідження було спроектовано та впроваджено інтерактивний графічний аналітичний інтерфейс користувача на базі матричної структури осей відображення, який під час роботи комплексу забезпечує оператора вичерпними та наочними даними. Розроблена архітектура візуалізації дозволяє одночасно контролювати як первинні часові та фазові квадратурні характеристики вхідних сигналів, так і результати проміжних спектральних обчислень у межах окремих кроків послідовної перебудови частотного гетеродина, завершуючи аналіз виведенням підсумкової безшовної широкосмугової панорами ефіру в надширокому діапазоні від вісімдесяти восьми до однієї тисячі мегагерц.

Особливу увагу під час експериментів було приділено оцінці завадостійкості та точності роботи адаптивного CFAR-детектора сигналів. Математичне моделювання підтвердило, що розрахунок динамічного профілю завад на основі алгоритму ковзного мінімуму у поєднанні з процедурами бінарного морфологічного очищення та маркування зв'язаних областей забезпечує стабільний рівень хибних спрацьовувань та надійне виявлення корисних випромінювань у складних ефірних умовах, гарантуючи точну фіксацію частотних меж, оцінку смуги пропускання та вимірювання амплітудних піків знайдених радіоджерел.

На основі отриманих експериментальних даних було сформовано розгорнуті інженерно-технічні та експлуатаційні рекомендації щодо практичного розгортання та калібрування комплексу під час роботи з реальними трансиверами. Разом із цим було чітко визначено та обґрунтовано перспективні шляхи модернізації системи, пов'язані з перенесенням ресурсомістких алгоритмів віконного WOLA-зшивання спектра безпосередньо на рівень програмованої логіки ПЛІС, а також впровадженням інтелектуальних класифікаторів на базі статистичного аналізу для автоматичного розпізнавання технологічних стандартів і видів модуляції виявлених сигналів.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

У дипломній роботі успішно розв'язано актуальну науково-технічну задачу побудови вискоєфективної системи радіочастотного моніторингу випромінювань у надширокому діапазоні частот за умов відсутності прямого доступу до фізичного апаратного обладнання. Проведений комплексний аналіз стану завадової обстановки та тенденцій розвитку бездротових мереж показав, що класичні скануючі приймачі не здатні забезпечити необхідну швидкість огляду ефіру. Обґрунтовано, що оптимальним інженерним шляхом вирішення цієї проблеми є застосування технологій програмно-визначеного радіо (SDR), а відпрацювання нових алгоритмів доцільно здійснювати за допомогою високоточного цифрового імітаційного моделювання, яке дозволяє гнучко аналізувати радіоефір у віртуальному середовищі хост-комп'ютера.

Для забезпечення максимальної відповідності моделювання реальним умовам за базу для побудови імітаційного стенда було обрано технічні характеристики популярної платформи ADALM-PLUTO SDR та її інтегрального трансивера AD9364. У розробленому програмному емуляторі в середовищі MATLAB було повністю відтворено ключові апаратні обмеження реального чіпа, включаючи його дванадцятирозрядну структуру аналого-цифрового перетворення, заданий рівень власного теплового шуму, коефіцієнти підсилення тракту, а також обмеження миттєвої смуги пропускання на рівні двадцяти мегагерц для надійності обміну даними. Такий підхід дозволив детально дослідити поведінку системи та логіку послідовної перебудови частоти гетеродина без необхідності залучення та розгортання вартісних фізичних радіоприймальних вузлів.

У ході теоретичних досліджень було визначено, що послідовне розширення смуги моніторингу за межі миттєвого апаратного вікна викликає появу суттєвих крайових спотворень через неідеальність фільтрів радіоприймача, що вимагало інтеграції спеціалізованого математичного апарату. Для формування безперервної панорами було обґрунтовано та впроваджено метод цифрового зшивання суміжних спектральних смуг із використанням алгоритму зваженого додавання з перекриттям у частотній області. Математичне моделювання довело, що застосування віконного згладжування Ханна та введення п'ятнадцятивідсоткової зони взаємного накладання

спектрів дозволяє повністю компенсувати крайові завали амплітудно-частотної характеристики і гарантує точне відновлення безшовного профілю спектральної щільності потужності без утворення амплітудних розривів чи зон нечутливості на стиках каналів.

Для проведення всебічної експериментальної оцінки розробленого математичного апарату в межах дипломної роботи було створено та досліджено інтегрований програмно-моделюючий стенд, що повністю відтворює логіку та технічні обмеження реального трансивера AD9364 у безперервному циклі сканування. Розроблений керуючий комплекс у середовищі MATLAB успішно реалізує послідовну перебудову віртуального гетеродина за розрахованим частотним планом у надширокому діапазоні від вісімдесяти восьми до однієї тисячі мегагерц, здійснюючи усереднення спектральних відліків та їх фінальну високоточну аналітичну обробку. Створений у межах роботи матричний графічний інтерфейс користувача забезпечив інтеграцію часових, фазових та панорамних характеристик в єдине робоче поле оператора, що підтвердило високу інженерну ергономічність, функціональну завершеність та повну адекватність створеного емулятора ефіру реальним фізичним процесам.

Дослідження працездатності впровадженого адаптивного CFAR-детектора сигналів підтвердило його високу ефективність у складних та нерівномірних завадових умовах, які були штучно відтворені та змодельовані у програмі. Завдяки застосуванню алгоритму ковзного мінімуму для формування динамічного порога виявлення та методів двовимірної морфологічної фільтрації бінарної маски, комплекс продемонстрував здатність стабільно утримувати заданий рівень хибних спрацьовувань. Подальше маркування зв'язаних областей дозволило автоматизувати процес розділення суміжних радіовипромінювань, забезпечивши точне визначення їхніх центральних частот, меж смуг пропускання та пікових енергетичних рівнів з подальшою коректною автоматичною класифікацією за технологічними стандартами бездротового зв'язку.

На основі аналізу досвіду розробки та експлуатації імітаційного стенда сформовано низку практичних рекомендацій для майбутнього перенесення системи

на реальну апаратну платформу ADALM-PLUTO SDR. Операторам рекомендується враховувати, що перехід від математичної моделі до фізичного ефіру вимагатиме обов'язкового калібрування квадратурного дисбалансу амплітуд і фаз каналів ігрек та кю для усунення «сигналів-привидів», а також активації вбудованих апаратних режимів автоматичного регулювання підсилення замість фіксованих програмних констант, що дозволить захистити дванадцятирозрядний АЦП від жорсткого обмеження та нелінійних спотворень. Крім того, визначені вектори довгострокової модернізації комплексу, пов'язані з перенесенням ресурсомістких алгоритмів віконного зшивання спектра WOLA безпосередньо всередину кристала ПЛІС імітованої платформи, закладають надійний теоретичний фундамент для розробки перспективних завадостійких систем радіоконтролю надвисокої швидкості огляду.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Software-Defined Radio for Engineers / T. F. Collins, R. Getz, D. Pu, A. M. Wyglinski. – Norwood: Artech House, 2018. – 375 p. ISBN 978-1-63081-457-1.
2. Оппенгейм А. В., Шефер Р. В. Обробка дискретних сигналів. 3-тє видання. – Нью-Йорк: Pearson, 2010. – 1120 с.
3. Welch P. D. The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short or Modified Periodograms // IEEE Transactions on Audio and Electroacoustics. – 1967. – Vol. 15, No. 2. – P. 70–73.
4. Скулиш М. А., Ковальчук С. О. Цифрова обробка сигналів у системах бездротового зв'язку: навчальний посібник. – Київ: КПІ ім. Ігоря Сікорського, 2021. – 248 с.
5. Lyons R. G. Understanding Digital Signal Processing. 3rd Edition. – Upper Saddle River: Prentice Hall, 2010. – 960 p. ISBN 978-0137027415.
6. Harris F. J. Multirate Signal Processing for Communication Systems. – Upper Saddle River: Prentice Hall, 2004. – 496 p. ISBN 978-0131465114.
7. Richards M. A. Fundamentals of Radar Signal Processing. 2nd Edition. – New York: McGraw-Hill Education, 2014. – 656 p. [Включає математичні основи алгоритмів сімейства CFAR]. ISBN 978-0071798327.
8. Основи аналізу спектра в реальному масштабі часу. Частина 1 [Електронний ресурс] – Режим доступу до ресурсу: https://gtest.com.ua/uk/osnovy-analiza-spektra-v-realnom-masshtabe-vremeni-chast_1.html
9. ADALM-PLUTO Evaluation Board User Guide / Analog Devices Wiki [Електронний ресурс] – Режим доступу до ресурсу: <https://wiki.analog.com/university/tools/pluto>
10. Signal Processing Toolbox User's Guide / MathWorks Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mathworks.com/help/signal/index.html>

11. Automated Signal Detection and Spectral Analysis Reference / IEEE Xplore Digital Library [Электронный ресурс] – Режим доступа до ресурсу: <https://ieeexplore.ieee.org/document/reference-signal-processing-techniques/>

12. DragonOS – Linux distribution for Software Defined Radio [Electronic resource] // DistroWatch. – 2026. – URL: <https://distrowatch.com/table.php?distribution=dragonos>

ДОДАТОК А

Програмний комплекс системи моніторингу. Клас емулятора трансивера ADALM-PLUTO

```
% =====

% ЛІСТИНГ ПРОГРАМИ. КЛАС ЕМУЛЯЦІЇ РАДІОЧАСТОТНОГО ТРАКТУ PLUTO SDR

% Файл: pluto_sdr_emulator.m

% =====

classdef pluto_sdr_emulator < handle

    properties

        CenterFrequency    = 433e6;    % Центральна частота гетеродина, Гц

        SampleRate          = 20e6;    % Частота дискретизації, Гц

        Bandwidth           = 20e6;    % Смуга пропускання аналогового фільтра, Гц

        GainDB              = 30;      % Коефіцієнт підсилення приймача, дБ

        NumSamples          = 4096;    % Розмір кадру (кількість IQ-відліків)

        StitchBands         = [];      % Структура масиву піддіапазонів

        StitchOverlap       = 0.1;     % Коефіцієнт взаємного перекриття смуг

        TotalBandwidth      = 200e6;   % Загальна ширина смуги моніторингу, Гц

        StartFrequency      = 88e6;    % Початкова частота сканування, Гц

        NoiseFloor          = -100;    % Рівень власного шуму, дБм

        SignalSources       = [];      % База даних імітованих джерел випромінювання

    end

    properties (Access = private)

        SampleRateAD9364 = 61.44e6;  % Гранична частота дискретизації чіпа
```

```

MaxFreq          = 6e9;          % Максимальна робоча частота трансивера

MinFreq          = 70e6;         % Мінімальна робоча частота трансивера

end

methods

function obj = pluto_sdr_emulator(varargin)

    % Конструктор класу: ініціалізація та конфігурування системних об'єктів

    for i = 1:2:length(varargin)

        if isprop(obj, varargin{i})

            obj.(varargin{i}) = varargin{i+1};

        end

    end

end

obj.SignalSources = obj.generateSignalSources();

obj.StitchBands = obj.calculateStitchBands();

end

function [iq_data, timestamp] = receive(obj)

    % Імітація процесу захоплення високошвидкісного кадру IQ-відліків

    timestamp = datetime('now');

    noise_power = 10^(obj.NoiseFloor/10) * 1e-3;

    noise = sqrt(noise_power/2) * (randn(obj.NumSamples,1) + ...

                                   1j*randn(obj.NumSamples,1));

    iq_data = noise;

    t = (0:obj.NumSamples-1)' / obj.SampleRate;

```

```

for k = 1:length(obj.SignalSources)

    src = obj.SignalSources(k);

    delta_f = src.frequency - obj.CenterFrequency;

    if abs(delta_f) < obj.SampleRate/2

        amp = 10^((src.power + obj.GainDB - 30)/20);

        signal = amp * src.modulate(t, delta_f);

        iq_data = iq_data + signal;

    end

end

% Масштабування та квантування під 12-бітну розрядність АЦП AD9364

max_val = 2^11 - 1;

scale = max_val / max(abs([real(iq_data); imag(iq_data)]));

iq_data = round(iq_data * scale) / scale;

end

function [freq, psd] = computePSD(obj, iq_data)

    % Оцінка спектральної щільності потужності модифікованим методом Велча

    nfft = min(obj.NumSamples, 4096);

    window = hann(nfft);

    overlap = round(nfft * 0.5);

    [pxx, f] = pwelch(iq_data, window, overlap, nfft, obj.SampleRate,
'centered');

    freq = f + obj.CenterFrequency;

    psd = 10*log10(pxx + eps) + 30;

end

```

```

function [stitched_freq, stitched_psd] = stitchSpectrum(obj)

    % Алгоритм послідовної перебудови частоти гетеродина та зшивання панорами

    n_bands = length(obj.StitchBands);

    all_freq = {};

    all_psd = {};

    orig_fc = obj.CenterFrequency;

    orig_bw = obj.SampleRate;

    for b = 1:n_bands

        obj.CenterFrequency = obj.StitchBands(b).center;

        obj.SampleRate      = obj.StitchBands(b).bw;

        n_avg = 8;

        psd_accum = 0;

        for av = 1:n_avg

            [iq, ~] = obj.receive();

            [f, p] = obj.computePSD(iq);

            psd_accum = psd_accum + 10.^(p/10);

        end

        all_freq{b} = f;

        all_psd{b} = 10*log10(psd_accum/n_avg + eps);

    end

    obj.CenterFrequency = orig_fc;

    obj.SampleRate      = orig_bw;

```

```

        [stitched_freq, stitched_psd] = obj.mergeSpectrumBands(all_freq,
all_psd);

```

```

end

```

```

function sources = generateSignalSources(obj) %#ok<MANU>

```

```

% Моделювання радіоелектронної обстановки та параметрів сигналів

```

```

fm_freqs = [88.3, 90.7, 93.1, 95.8, 98.6, 101.2, 104.5, 107.1] * 1e6;

```

```

fm_powers = -60 + 10*rand(1, length(fm_freqs));

```

```

ism433_freqs = [433.05, 433.92, 434.79] * 1e6;

```

```

ism433_powers = -70 + 15*rand(1, length(ism433_freqs));

```

```

gsm_freqs = [935.2, 940.8, 945.4, 952.0] * 1e6;

```

```

gsm_powers = -65 + 12*rand(1, length(gsm_freqs));

```

```

wifi_freqs = [2.412, 2.437, 2.462] * 1e9;

```

```

wifi_powers = -55 + 8*rand(1, length(wifi_freqs));

```

```

all_freqs = [fm_freqs, ism433_freqs, gsm_freqs, wifi_freqs];

```

```

all_powers = [fm_powers, ism433_powers, gsm_powers, wifi_powers];

```

```

mod_types = [ repmat({'fm'}, 1, length(fm_freqs)), ...

```

```

    repmat({'ook'}, 1, length(ism433_freqs)), ...

```

```

    repmat({'gsm'}, 1, length(gsm_freqs)), ...

```

```

    repmat({'ofdm'}, 1, length(wifi_freqs))]];

```

```

sources = struct('frequency', num2cell(all_freqs), ...

               'power',      num2cell(all_powers), ...

               'modtype',    mod_types);

for k = 1:length(sources)

    switch sources(k).modtype

        case 'fm'

            sources(k).modulate = @(t, df) exp(1j*2*pi*df*t) .* ...

exp(1j*2*pi*75e3*cumsum(0.3*randn(length(t),1))/length(t));

            case 'ook'

                sources(k).modulate = @(t, df) exp(1j*2*pi*df*t) .* ...

                    (rand(length(t),1) > 0.5);

            case 'gsm'

                sources(k).modulate = @(t, df) exp(1j*2*pi*df*t) .* ...

                    sign(randn(length(t),1));

            case 'ofdm'

                sources(k).modulate = @(t, df) exp(1j*2*pi*df*t) .* ...

                    (randn(length(t),1) + 1j*randn(length(t),1))/sqrt(2);

        end

    end

end

end

function bands = calculateStitchBands(obj)

    % Розрахунок оптимальної кількості частотних кроків перебудови гетеродина

    step = obj.SampleRate * (1 - obj.StitchOverlap);

    centers = obj.StartFrequency + obj.SampleRate/2 : step : ...

```

```
obj.StartFrequency + obj.TotalBandwidth - obj.SampleRate/2;
```

```
bands = struct('center', num2cell(centers), ...
               'bw',      num2cell(repmat(obj.SampleRate, 1,
length(centers)))));

end
```

```
function [freq_out, psd_out] = mergeSpectrumBands(obj, freq_cells, psd_cells)

% Вагове об'єднання спектральних вікон та усунення завалів на стиках

n_bands = length(freq_cells);

f_min = min(freq_cells{1});

f_max = max(freq_cells{end});

f_step = (freq_cells{1}(2) - freq_cells{1}(1));

freq_out = f_min : f_step : f_max;

psd_out = -200 * ones(size(freq_out));

weight = zeros(size(freq_out));

for b = 1:n_bands

    f_b = freq_cells{b};

    p_b = psd_cells{b};

    in_range = freq_out >= min(f_b) & freq_out <= max(f_b);

    p_interp = interp1(f_b, p_b, freq_out(in_range), 'linear');

    n_pts = sum(in_range);

    if n_pts > 0

        w = ones(1, n_pts);
```

```

edge = round(n_pts * obj.StitchOverlap);

if edge > 1

    ramp = linspace(0, 1, edge);

    w(1:edge)                = ramp;

    w(end-edge+1:end)        = fliplr(ramp);

end

p_lin = 10.^(p_interp/10);

idx = find(in_range);

psd_curr = 10.^(psd_out(idx)/10);

psd_curr(weight(idx)==0) = 0;

psd_out(idx) = 10*log10((psd_curr.*weight(idx) + p_lin.*w) ./
...
                                (weight(idx) + w) + eps);

weight(idx) = weight(idx) + w;

end

end

empty = weight == 0;

psd_out(empty) = obj.NoiseFloor - 10*rand(1, sum(empty));

end

end

end

```

ДОДАТОК Б

Програмний комплекс імітаційного моделювання. Модуль порівняльного аналізу алгоритмів зшивання спектра

```
% =====

% ЛІСТИНГ ПРОГРАМИ. СТЕНД ПОРІВНЯЛЬНОГО АНАЛІЗУ АЛГОРИТМІВ ЗШИВАННЯ СПЕКТРА

% Файл: spectrum_stitching_demo.m

% =====

clear; clc; close all;

%% Конфігурування базових параметрів системи моніторингу

fs_max      = 20e6;          % Миттєва смуга одного вікна аналізу SDR, Гц

overlap     = 0.15;         % Коефіцієнт взаємного перекриття вікон (15%)

f_start     = 88e6;         % Нижня межа досліджуваного діапазону, Гц

f_end       = 300e6;        % Верхня межа досліджуваного діапазону, Гц

n_samples   = 8192;         % Кількість спектральних відліків (розмір БПФ)

%% Генерація вектору еталонних широкосмугових радіосигналів

true_signals = struct( ...

    'freq', {90.1e6, 95.5e6, 100.3e6, 104.7e6, 144.8e6, 162.0e6, 200.5e6, 245.3e6},
    ...

    'power', {-50,    -58,    -55,    -62,    -70,    -75,    -65,    -72},
    ...

    'bw',     {200e3,  200e3,  200e3,  200e3,   25e3,   25e3,   500e3,  200e3});

%% Моделювання процесу послідовного сканування по смугах

step      = fs_max * (1 - overlap);

fc_arr    = f_start + fs_max/2 : step : f_end - fs_max/2;
```

```

n_bands = length(fc_arr);

band_data = struct('freq', cell(n_bands,1), 'psd', cell(n_bands,1), 'fc',
cell(n_bands,1));

for b = 1:n_bands

    fc = fc_arr(b);

    f_vec = fc + linspace(-fs_max/2, fs_max/2, n_samples);

    % Формування спектральної щільності потужності теплового шуму

    psd_b = -100 + 3*randn(1, n_samples);

    % Суперпозиція сигналів, що потрапляють у поточну смугу прийому

    for s = 1:length(true_signals)

        df = true_signals(s).freq - fc;

        if abs(df) < (fs_max/2 + true_signals(s).bw)

            bw_bins = round(true_signals(s).bw / (fs_max/n_samples));

            bw_bins = max(bw_bins, 3);

            [~, idx_peak] = min(abs(f_vec - true_signals(s).freq));

            idx_range = max(1, idx_peak-bw_bins) : min(n_samples, idx_peak+bw_bins);

            peak = true_signals(s).power;

            % Апроксимація форми спектра сигналу функцією Гаусса

            x = linspace(-3, 3, length(idx_range));

            psd_b(idx_range) = max(psd_b(idx_range), peak * exp(-x.^2/2));

        end

    end

end

```

```

    band_data(b).freq = f_vec;

    band_data(b).psd = psd_b;

    band_data(b).fc = fc;

end

%% Математична реалізація трьох методів зшивання

[f_naive, psd_naive] = stitchNaive(band_data);

[f_hann, psd_hann] = stitchWeighted(band_data, 'hann');

[f_mvdr, psd_mvdr] = stitchMVDR(band_data);

%% Розрахунок критеріїв оцінки якості формування панорами

evalStitchQuality(f_naive, psd_naive, 'Просте конкатенування');

evalStitchQuality(f_hann, psd_hann, 'Вагова функція Ганна');

evalStitchQuality(f_mvdr, psd_mvdr, 'MVDR оцінювач');

%% Графічний аналіз та візуалізація результатів порівняння

figure('Name','Порівняння методів зшивання спектра', 'Position',[100 100 1300 800],
...

    'Color',[0.1 0.1 0.15]);

colors = {[0.4 0.7 1.0], [0.2 0.9 0.4], [1.0 0.6 0.2]};

methods = {'Просте конкатенування', 'Зшивання вікном Ганна', 'Оцінювач MVDR'};

data = {psd_naive, psd_hann, psd_mvdr};

freqs = {f_naive, f_hann, f_mvdr};

for m = 1:3

```

```

ax = subplot(3,1,m);

ax.Color = [0.15 0.15 0.2];

ax.XColor = [0.7 0.8 0.9]; ax.YColor = [0.7 0.8 0.9];


plot(freqs{m}/1e6, data{m}, 'Color', colors{m}, 'LineWidth', 0.8);

hold on;


for b = 1:n_bands

    xline(band_data(b).fc/1e6, ':', 'Color', [0.5 0.5 0.5], ...

        'LineWidth', 0.5, 'Alpha', 0.4);

end


for s = 1:length(true_signals)

    xline(true_signals(s).freq/1e6, '--', 'Color', [1 0.3 0.3], ...

        'LineWidth', 1, 'Alpha', 0.6);

end


ylabel('PSD (дБм/Гц)', 'Color', [0.7 0.8 0.9]);

title(methods{m}, 'Color', [1 0.85 0.2], 'FontSize', 11);

grid on; ax.GridColor = [0.3 0.3 0.4];

ylim([-115 -35]);

xlim([f_start f_end]/1e6);

if m == 3

    xlabel('Частота (МГц)', 'Color', [0.7 0.8 0.9]);

end

end
end

```

```
sgtitle('Порівняльний аналіз методів панорамного зшивання спектра радіосигналів', ...
```

```
    'Color','white', 'FontSize',13, 'FontWeight','bold');
```

```
% =====
```

```
%  ЛОКАЛЬНІ ФУНКЦІЇ ОБРОБКИ МАСИВІВ ДАНИХ
```

```
% =====
```

```
function [f_out, p_out] = stitchNaive(bands)
```

```
% Метод прямого конкатенування частотно-амплітудних векторів
```

```
f_out = []; p_out = [];
```

```
for b = 1:length(bands)
```

```
    f_out = [f_out, bands(b).freq];
```

```
    p_out = [p_out, bands(b).psd];
```

```
end
```

```
[f_out, idx] = sort(f_out);
```

```
p_out = p_out(idx);
```

```
end
```

```
function [f_out, p_out] = stitchWeighted(bands, win_type)
```

```
% Метод вагового усереднення перекриттів із гладким зшиванням вікном
```

```
n_bands = length(bands);
```

```
f_min = min(bands(1).freq);
```

```
f_max = max(bands(end).freq);
```

```
n_pts = round((f_max - f_min) / (bands(1).freq(2) - bands(1).freq(1)));
```

```
f_out = linspace(f_min, f_max, n_pts);
```

```
p_lin = zeros(1, n_pts);
```

```

w_total = zeros(1, n_pts);

for b = 1:n_bands

    in_range = f_out >= min(bands(b).freq) & f_out <= max(bands(b).freq);

    p_interp = interp1(bands(b).freq, bands(b).psd, f_out(in_range), 'linear');

    n_in = sum(in_range);

    switch win_type

        case 'hann'

            w = hann(n_in)';

        case 'rect'

            w = ones(1, n_in);

        otherwise

            w = hann(n_in)';

    end

    w = w / max(w);

    p_linfac = 10.^(p_interp/10);

    idx = find(in_range);

    p_lin(idx) = p_lin(idx) + p_linfac .* w;

    w_total(idx) = w_total(idx) + w;

end

valid = w_total > 0;

p_out = -200 * ones(size(f_out));

p_out(valid) = 10*log10(p_lin(valid) ./ w_total(valid) + eps);

end

```

```

function [f_out, p_out] = stitchMVDR(bands)

    % Адаптивне зшивання на основі логіки мінімальної дисперсії

    [f_out, p_sorted] = stitchNaive(bands);

    p_out = movmedian(p_sorted, 7);

    % Диференціальний аналіз та фільтрація аномальних міжзонних стрибків

    dp = diff(p_out);

    jump_idx = find(abs(dp) > 15);

    for j = jump_idx(:)'

        lo = max(1, j-5); hi = min(length(p_out), j+5);

        p_out(lo:hi) = interp1([lo, hi], [p_out(lo), p_out(hi)], lo:hi, 'linear');

    end

end

function evalStitchQuality(~, p, name)

    % Обчислення статистичних критеріїв безперервності сформованого масиву

    dp = diff(p);

    fprintf('  %-28s | Математичне середнє: %6.2f дБм | Сумарний градієнт: %8.2f |  

    Стрибки потужності: %d\n', ...

        name, mean(p), sum(abs(dp)), sum(abs(dp)>10));

end

```

ДОДАТОК В

Програмний комплекс імітаційного моделювання. Модуль безперервного моніторингу та графічного інтерфейсу оператора

```

% =====
% ЛІСТИНГ ПРОГРАМИ. МОДУЛЬ БЕЗПЕРЕРВНОГО МОНІТОРИНГУ ТА ГРАФІЧНОГО GUI
% Файл: realtime_monitor.m
% =====

clear; clc; close all;

%% --- Ініціалізація об'єкта емулятора радіочастотного тракту ---
sdr = pluto_sdr_emulator( ...
    'CenterFrequency', 433e6, ...
    'SampleRate',      10e6, ...
    'GainDB',          35, ...
    'NumSamples',      2048, ...
    'NoiseFloor',      -102);

% Конфігурування параметрів інтегрування та відображення даних
n_frames = 100;      % Фіксована кількість кадрів для поточної сесії
hold_time = 0.05;    % Міжфреймовий інтервал оновлення графіків (с)
n_avg     = 5;        % Розмір вікна усереднення для ліній AVG та MAX-HOLD

% Ініціалізація буферів пам'яті та накопичувальних матриць
[iq0, ~] = sdr.receive();
[f0, p0] = sdr.computePSD(iq0);
n_pts    = length(f0);

max_hold = -150 * ones(1, n_pts);
avg_buf   = zeros(n_avg, n_pts);
avg_ptr   = 1;

%% --- Ініціалізація та конфігурування графічного інтерфейсу (GUI) ---
fig = figure('Name', 'Моніторинг спектру – Режим реального часу', ...
    'Position', [100 100 1100 700], ...
    'Color', [0.08 0.08 0.12], ...
    'KeyPressFcn', @(~,~) setappdata(gcf,'stop',true));
setappdata(fig, 'stop', false);

ax = axes('Parent', fig, 'Color', [0.1 0.1 0.16], ...
    'XColor', [0.6 0.8 1.0], 'YColor', [0.6 0.8 1.0], ...
    'FontName', 'Courier New', 'FontSize', 9, ...
    'Position', [0.08 0.12 0.88 0.78]);

% Формування графічних об'єктів для відображення спектральних ліній
h_live = plot(ax, f0/1e6, p0, 'Color', [0.2 0.9 0.4], 'LineWidth', 1.0);
hold(ax, 'on');
h_avg   = plot(ax, f0/1e6, p0, 'Color', [0.0 0.7 1.0], 'LineWidth', 1.2);
h_max   = plot(ax, f0/1e6, max_hold, 'Color', [1.0 0.4 0.4], 'LineWidth', 1.5,
    'LineStyle', '--');
h_thresh = yline(ax, sdr.NoiseFloor+12, '--', 'Color', [1.0 0.8 0.0], 'LineWidth',
1);

legend(ax, 'Поточний спектр', 'Усереднений спектр (AVG)', 'Режим утримання піків (MAX
HOLD)', 'Адаптивний поріг виявлення CFAR', ...
    'TextColor','white', 'Color',[0.15 0.15 0.2], 'EdgeColor','none', ...
    'Location','northeast');
```

```

xlabel(ax, 'Частота (МГц)'); ylabel(ax, 'Спектральна щільність потужності (дБм)');
grid(ax, 'on'); ax.GridColor = [0.25 0.25 0.35];
xlim(ax, [f0(1) f0(end)]/1e6);
ylim(ax, [sdr.NoiseFloor-10, sdr.NoiseFloor+60]);

% Конфігурування текстових блоків виводу динамічних метаданих
h_time = text(ax, 0.01, 0.97, '', 'Units','normalized', ...
    'Color',[0.5 0.8 1.0], 'FontName','Courier New', ...
    'FontSize', 9, 'VerticalAlignment','top');

h_info = text(ax, 0.01, 0.90, '', 'Units','normalized', ...
    'Color',[0.4 0.9 0.5], 'FontName','Courier New', 'FontSize', 8, ...
    'VerticalAlignment','top');

title(ax, sprintf('Панорамний аналізатор ADALM-PLUTO (AD9364) | FC = %.0f МГц | BW = %.0f МГц', ...
    sdr.CenterFrequency/1e6, sdr.SampleRate/1e6), ...
    'Color', [1.0 0.85 0.2], 'FontSize', 11, 'FontName','Courier New');

%% --- Головний розрахунковий цикл реального часу ---
frame_times = zeros(1, n_frames);

for fr = 1:n_frames
    if ~ishandle(fig) || getappdata(fig,'stop'), break; end

    t_start = tic;

    % Захоплення чергового кадру квадратурних відліків та обчислення спектра
    [iq, ts] = sdr.receive();
    [f_mhz, p] = sdr.computePSD(iq);

    % Розрахунок та оновлення вектора максимальних амплітуд за сесію
    max_hold = max(max_hold, p);

    % Реалізація алгоритму ковзного середнього у часовому вікні
    avg_buf(avg_ptr, :) = p';
    avg_ptr = mod(avg_ptr, n_avg) + 1;
    avg_trace = mean(avg_buf, 1);

    % Порогова бінаризація відліків (попереднє виявлення сигналів)
    n_above = sum(p' > (sdr.NoiseFloor + 12));

    % Динамічне перемальовування графічних ліній у GUI
    set(h_live, 'YData', p');
    set(h_avg, 'YData', avg_trace);
    set(h_max, 'YData', max_hold);

    % Розрахунок поточної швидкості обробки потоку даних (числа FPS)
    frame_times(fr) = toc(t_start);
    fps = 1 / mean(frame_times(max(1,fr-9):fr));

    % Оновлення текстових інформаційних стрічок інтерфейсу оператора
    set(h_time, 'String', sprintf('%s Кадр: %d/%d Швидкість обробки: %.1f FPS',
    ...
        datestr(ts, 'HH:MM:SS.FFF'), fr, n_frames, fps));
    set(h_info, 'String', sprintf('Гетеродин: %.3f МГц | Смуга: %.1f МГц |
    Виявлено джерел: %d | AGC: %d дБ', ...
        sdr.CenterFrequency/1e6, sdr.SampleRate/1e6, n_above, sdr.GainDB));

```

```

        drawnow limitrate;
        pause(max(0, hold_time - frame_times(fr)));
end

%% --- Побудова підсумкового графіка за результатами поточної сесії ---
figure('Name','Результати моніторингу – Спектрограма пікових значень', ...
       'Position',[200 200 1000 400], 'Color',[0.08 0.08 0.12]);
ax2 = axes('Color',[0.12 0.12 0.18], 'XColor',[0.7 0.8 0.9], 'YColor',[0.7 0.8 0.9]);

fill([f_mhz(1); f_mhz'; f_mhz(end)], ...
     [min(max_hold)-5; max_hold'; min(max_hold)-5], ...
     [1.0 0.4 0.4], 'FaceAlpha', 0.4, 'EdgeColor',[1.0 0.5 0.5]);

xlabel('Частота (МГц)'); ylabel('Рівень пікової потужності MAX HOLD (дБм)');
title('Підсумкова спектрограма накопиченого максимуму (Max Hold) за сесію',
      'Color',[1 0.85 0.2], 'FontSize',12);
grid on; ax2.GridColor = [0.3 0.3 0.4];
xlim([f_mhz(1) f_mhz(end)]);

```

ДОДАТОК Г

Програмний комплекс імітаційного моделювання. Головний керуючий модуль системи радіочастотного моніторингу

```

% =====
% ЛІСТИНГ ПРОГРАМИ. ГОЛОВНИЙ КЕРУЮЧИЙ МОДУЛЬ СИСТЕМИ РЧ-МОНІТОРИНГУ
% Файл: sdr_monitor.m
% =====

clear; clc; close all;

fprintf('=====\\n');
fprintf(' КОМПЛЕКС РАДІОЧАСТОТНОГО МОНІТОРИНГУ ТА СКАКТУВАННЯ      \\n');
fprintf(' ГОЛОВНИЙ ДИСПЕТЧЕРСЬКИЙ МОДУЛЬ СИСТЕМИ                \\n');
fprintf('=====\\n\\n');

%% =====
% БЛОК 1. КОНФІГУРУВАННЯ ТЕХНІЧНИХ ПАРАМЕТРІВ СИСТЕМИ
% =====

% Технічні обмеження інтегрального трансивера AD9364
AD9364.InstBW      = 20e6;    % Миттєва смуга одного налаштування гетеродина (Гц)
AD9364.Overlap     = 0.15;   % Коефіцієнт взаємного перекриття суміжних смуг (15%)
AD9364.NoiseFloor  = -105;   % Рівень власних теплових шумів приймача, дБм
AD9364.Gain        = 40;     % Коефіцієнт підсилення тракту приймача, дБ
AD9364.NFFT        = 2048;   % Розрядність швидкого перетворення Фур'є (БПФ)

% Параметри панорамного огляду робочого діапазону частот
MON.StartFreq      = 88e6;    % Нижня межа робочого діапазону моніторингу, Гц
MON.StopFreq       = 300e6;   % Верхня межа робочого діапазону моніторингу, Гц
MON.NumAverages     = 6;     % Кількість спектральних усереднень в кожній точці

%% =====
% БЛОК 2. АВТОМАТИЧНИЙ РОЗРАХУНОК ПЛАНУ ЧАСТОТНОГО СКАНУВАННЯ
% =====
effective_step = AD9364.InstBW * (1 - AD9364.Overlap); % Корисний крок без врахування стиків
f_centers = (MON.StartFreq + AD9364.InstBW/2) : effective_step : (MON.StopFreq - AD9364.InstBW/2);

if f_centers(end) < (MON.StopFreq - AD9364.InstBW/2)
    f_centers = [f_centers, MON.StopFreq - AD9364.InstBW/2]; %#ok<AGROW>
end

sweep_plan = struct('center_freq', num2cell(f_centers), ...
                    'bw', num2cell(repmat(AD9364.InstBW, 1, length(f_centers))));

fprintf('[ІНФО] Розрахунок частотного плану завершено:\\n');
fprintf('    Діапазон огляду : %.1f МГц - %.1f МГц\\n', MON.StartFreq/1e6, MON.StopFreq/1e6);
fprintf('    Кількість кроків сканування тракту SDR: %d\\n\\n', length(sweep_plan));

%% =====
% БЛОК 3. МОДЕЛЮВАННЯ РАДІОЕЛЕКТРОННОЇ ОБСТАНОВКИ В КАНАЛІ
% =====
signals = struct(...

```

```

    'center_freq', {92.3e6, 98.5e6, 103.2e6, 145.5e6, 168.2e6, 215.0e6, 243.6e6,
290.1e6}, ...
    'peak_power', {-45, -52, -48, -72, -68, -55, -60, -
50}, ...
    'type', {'wfm', 'wfm', 'wfm', 'nfm', 'nfm', 'ofdm', 'ofdm',
'wfm'}, ...
    'param', {180e3, 150e3, 160e3, 12.5e3, 25e3, 1.5e3, 2.0e3,
120e3});

%% =====
% БЛОК 4. НАЛАШТУВАННЯ ВІЗУАЛІЗАЦІЇ ІНТЕРФЕЙСУ (КОЛІРНА ПАЛІТРА)
% =====
C.bg = [0.07 0.07 0.10]; C.axbg = [0.11 0.11 0.16];
C.green = [0.20 0.85 0.45]; C.blue = [0.00 0.65 1.00];
C.orange = [1.00 0.55 0.10]; C.yellow = [0.95 0.85 0.20];
C.red = [0.95 0.30 0.30]; C.grid = [0.22 0.22 0.30];
C.text = [0.75 0.85 0.95];

fig = figure('Name', 'Комплексний аналіз радіочастотного спектра', ...
    'Position', [50 50 1450 850], 'Color', C.bg);

%% =====
% БЛОК 5. ЦИКЛ ЧАСТОТНОГО СУПЕРПОЗИЦІЙНОГО СКАНУВАННЯ
% =====
raw_spectra = cell(length(sweep_plan), 1);
raw_freqs = cell(length(sweep_plan), 1);

for idx = 1:length(sweep_plan)
    fc = sweep_plan(idx).center_freq;
    f_vector = linspace(fc - AD9364.InstBW/2, fc + AD9364.InstBW/2, AD9364.NFFT);

    % Формування базового розподілу потужності шуму
    psd_vector = AD9364.NoiseFloor + 2*randn(1, AD9364.NFFT);

    % Розрахунок проходження сигналів через аналоговий селективний тракт
    for s = 1:length(signals)
        dist_to_edge = AD9364.InstBW/2 + signals(s).param;
        if abs(signals(s).center_freq - fc) < dist_to_edge
            [~, center_bin] = min(abs(f_vector - signals(s).center_freq));

            switch signals(s).type
                case 'wfm'
                    span = round(signals(s).param / (AD9364.InstBW/AD9364.NFFT));
                    bins = max(1, center_bin-span):min(AD9364.NFFT, center_bin+span);
                    dx = linspace(-2, 2, length(bins));
                    shape = signals(s).peak_power * exp(-dx.^2 / 1.5);
                case 'nfm'
                    span = round(signals(s).param / (AD9364.InstBW/AD9364.NFFT));
                    bins = max(1, center_bin-span):min(AD9364.NFFT, center_bin+span);
                    dx = linspace(-3, 3, length(bins));
                    shape = signals(s).peak_power * exp(-dx.^2 / 0.5);
                case 'ofdm'
                    span = round(signals(s).param * 200 /
(AD9364.InstBW/AD9364.NFFT));
                    bins = max(1, center_bin-span):min(AD9364.NFFT, center_bin+span);
                    dx = linspace(-1, 1, length(bins));
                    shape = signals(s).peak_power - 12*dx.^4 + randn(1,length(bins));
            end
        end
    end
end

```

```

        % Додавання підсилення тракту
        shape_with_gain = shape + (AD9364.Gain * 0.15);
        psd_vector(bins) = max(psd_vector(bins), shape_with_gain);
    end
end

% Внесення частотного завалу країв аналогових фільтрів AD9364
normalized_f = linspace(-1, 1, AD9364.NFFT);
filter_roll_off = -25 * (normalized_f.^6);
psd_vector = psd_vector + filter_roll_off;

raw_spectra{idx} = psd_vector;
raw_freqs{idx} = f_vector;

% Візуалізація перших 7 проміжних каналів приймача
if idx <= 7
    ax = subplot(3, 4, idx);
    styleAx(ax, C);
    plot(ax, f_vector/1e6, psd_vector, 'Color', C.blue, 'LineWidth', 0.9);
    title(sprintf('Канал %d: FC = %.1f МГц', idx, fc/1e6), 'Color', C.orange,
'FontSize', 9);
    grid on; ax.GridColor = C.grid;
    ylim([AD9364.NoiseFloor-10, -20]);
    xlim([fc - AD9364.InstBW/2, fc + AD9364.InstBW/2]/1e6);
    if idx == 1 || idx == 5, ylabel('дБм'); end
end
end

%% =====
% БЛОК 6. МАТЕМАТИЧНЕ ЗШИВАННЯ ТА СИНТЕЗ ШИРОКОСМУГОВОЇ ПАНОРАМИ
% =====

f_min = MON.StartFreq; f_max = MON.StopFreq;
freq_step = (raw_freqs{1}(2) - raw_freqs{1}(1));
stitched_freqs = f_min : freq_step : f_max;
stitched_psd = -150 * ones(size(stitched_freqs));
accumulation_w = zeros(size(stitched_freqs));

for b = 1:length(sweep_plan)
    f_b = raw_freqs{b};
    p_b = raw_spectra{b};

    in_range = stitched_freqs >= min(f_b) & stitched_freqs <= max(f_b);
    p_interp = interp1(f_b, p_b, stitched_freqs(in_range), 'linear');

    n_in = sum(in_range);
    if n_in > 0
        % Застосування згладжуючого вікна Ханна для компенсації завалів фільтра
        w = hann(n_in)';
        w = w / max(w);

        p_lin = 10.^(p_interp / 10);
        idx_match = find(in_range);

        current_p_lin = 10.^(stitched_psd(idx_match) / 10);
        current_p_lin(accumulation_w(idx_match) == 0) = 0;

        stitched_psd(idx_match) = 10 * log10((current_p_lin .*
accumulation_w(idx_match) + p_lin .* w) ./ ...
(accumulation_w(idx_match) + w) + eps);
    end
end

```

```

        accumulation_w(idx_match) = accumulation_w(idx_match) + w;
    end
end

%% =====
% БЛОК 7. МУЛЬТИКАНАЛЬНИЙ ГРАФІЧНИЙ ВИВІД СИНТЕЗОВАНИХ ДАНИХ
% =====

% Графік 9: Загальний суміщений сирий спектр (демонстрація неузгодженості)
ax9 = subplot(3, 4, 9);
styleAx(ax9, C);
for b = 1:length(sweep_plan)
    plot(ax9, raw_freqs{b}/1e6, raw_spectra{b}, 'LineWidth', 0.5); hold on;
end
title('Суперпозиція незшитих вікон', 'Color', C.yellow, 'FontSize', 10);
grid on; ax9.GridColor = C.grid; xlim([f_min f_max]/1e6); ylim([AD9364.NoiseFloor-10, -20]);

% Графік 10: Складна функція розподілу вагових коефіцієнтів
ax10 = subplot(3, 4, 10);
styleAx(ax10, C);
plot(ax10, stitched_freqs/1e6, accumulation_w, 'Color', C.green, 'LineWidth', 1.2);
title('Розподіл віконної ваги', 'Color', C.yellow, 'FontSize', 10);
grid on; ax10.GridColor = C.grid; xlim([f_min f_max]/1e6); ylim([0
max(accumulation_w)+0.5]);

% Графік 11: Підсумковий безшовний широкополосний спектр
ax11 = subplot(3, 4, [11, 12]);
styleAx(ax11, C);
plot(ax11, stitched_freqs/1e6, stitched_psd, 'Color', C.green, 'LineWidth', 1.0);
hold on;

% Нанесення маркерів центральних частот та меж кроків сканування
for b = 1:length(sweep_plan)
    xline(ax11, sweep_plan(b).center_freq/1e6, ':', 'Color', [0.5 0.5 0.6], 'Alpha',
0.5);
end
title(sprintf('Синтезована панорама РЧ-спектра: %.1f - %.1f МГц', f_min/1e6,
f_max/1e6), ...
'Color', C.yellow, 'FontSize', 11, 'FontWeight', 'bold');
xlabel('Частота (МГц)'); ylabel('Потужність (дБм)');
grid on; ax11.GridColor = C.grid;
xlim([f_min f_max]/1e6); ylim([AD9364.NoiseFloor-5, max(stitched_psd)+10]);

sgtitle('Моделювання та аналіз результатів роботи широкополосного радіомоніторингу на
базі SDR AD9364', ...
'Color', 'white', 'FontSize', 13, 'FontWeight', 'bold');

%% =====
% ДОПОМІЖНІ ФУНКЦІЇ ДЛЯ СТИЛІЗАЦІЇ ІНТЕРФЕЙСУ
% =====
function styleAx(ax, C)
    % Автоматичне налаштування параметрів відображення осей координат
    ax.Color = C.axbg;
    ax.XColor = C.text; ax.YColor = C.text;
    ax.FontName = 'Courier New'; ax.FontSize = 8;
end

```

