

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА ПРИКЛАДНОЇ ФІЗИКИ

«До захисту допущено»

Завідувач кафедри

Г. Є. Монастирський

(підпис)

(ініціали, прізвище)

Дипломна робота
на здобуття ступеня бакалавра

зі спеціальності 105 «Прикладна фізика та наноматеріали»

(код і назва)

на тему: **Особливості формування зображень МІМО-радаром статичних цілей в середовищі з перешкодами**

Виконав: студент 4 курсу, групи ФФ-11

Корчевой Ігор Валерійович

(прізвище, ім'я, по батькові)

(підпис)

Науковий керівник ст. викл. Бех С. В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант 1,3,4 доцент, д.ф.-м.н. Монастирський Г. Є.

(номер розділу)

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант 1,2 ст. викл. Степаненко В. М.

(номер розділу)

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Рецензент доцент, к.т.н. Репа Ф. М.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

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

Студент

(підпис)

Національний Технічний Університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий Фізико-технічний інститут
Кафедра прикладної фізики

Рівень вищої освіти — перший (бакалаврський)
Спеціальність 105 «Прикладна фізика та наноматеріали»

«ЗАТВЕРДЖЕНО»

Завідувач кафедри

Монастирський Г. Є.

(підпис)

(ініціали, прізвище)

« ____ » _____ 2025 р.

ЗАВДАННЯ
на бакалаврську роботу студенту

Корчевому Ігорю Валерійовичу

(прізвище, ім'я, по батькові)

(підпис)

1. Тема роботи: Особливості формування зображень МІМО-радаром статичних цілей в середовищі з перешкодами,
науковий керівник роботи _____ ст. викл. Бех С. В. _____,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від « 26 » травня 2025 р № 2028-с

2. Термін подання студентом роботи « 18 » червня 2025 р.

3. Вихідні дані до роботи: .bin файли необроблених даних з радару, фотографії розташування цілей, технічні характеристики радару

4. Зміст роботи: Виготовлення кутового відбивача, перевірка якості калібрування, застосування алгоритму Річардсона-Люсі

5. Перелік завдань, які потрібно розробити: Опрацювати наявну літературу за темою калібрування та цифрової обробки для МІМО радарів; Вивчення технічних характеристик радару; Опанування програмне забезпечення, що обслуговує роботу МІМО радару; Проектування та виготовлення кутового відбивача методом електронно-променевої зварки; Проведення калібрування; Проведення експериментів із цілями з різною ефективною площею розсіювання в умовах відкритої місцевості; Проведення експериментів із цілями з різною ЕПР в середовищі з перешкодами; Обробка експериментальних результатів; Розробка коду ітеративного алгоритму деконволюції Річардсона-Люсі.

6. Орієнтовний перелік ілюстративного матеріалу: презентація — 22 аркушів
А4

7 Орієнтовний перелік публікацій: XXI Всеукраїнська науково-практична конференція студентів, аспірантів та молодих вчених

8 Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	ст. викл. Степаненко В. М.	05.01.2025	11.06.2025
2	ст. викл. Степаненко В. М.	05.01.2025	11.06.2025
3	доцент, д.ф.-м.н. Монастирський Г. Є.	05.01.2025	11.06.2025
4	доцент, д.ф.-м.н. Монастирський Г. Є.	05.01.2025	11.06.2025

9 Дата видачі завдання « 28 » квітня 2025 р.

Календарний план

№ з/п	Назва етапів виконання бакалаврської роботи	Термін виконання етапів проекту	Примітка
1.	Видача індивідуального завдання	28.04.2025	виконано
2.	Постановка мети та задач дослідження	05.01.2025 — 10.02.2025	виконано
3.	Оволодіння програмним забезпеченням від ТІ	05.01.2025 — 14.02.2025	виконано
4.	Написання літературного огляду	10.12.2024 — 15.03.2025	виконано
5.	Проектування та виготовлення кутового відбивача	20.03.2025 — 05.05.2025	виконано
6.	Проведення калібрування радару	13.05.2025 — 13.05.2025	виконано
7.	Проведення експериментів	15.02.2025 — 29.05.2025	виконано
8.	Обробка даних	15.02.2025 — 02.06.2025	виконано
9.	Розробка коду реалізації алгоритму Річардсона-Люсі	13.05.2025 — 02.06.2025	виконано
10.	Написання висновків	13.05.2025 — 02.06.2025	виконано
11.	Підготовка тез до конференції	13.02.2025 — 14.05.2025	виконано
12.	Виступ на конференції	14.05.2025	виконано
13.	Подання роботи на рецензування	15.02.2025 — 13.06.2025	виконано
14.	Нормоконтроль та рецензування	15.02.2025 — 13.06.2025	виконано
15.	Підготовка доповіді та презентації	01.06.2025 — 13.06.2025	виконано
16.	Попередній захист дипломної роботи	11.06.2025	виконано
17.	Подання роботи до захисту	18.06.2025	виконано
18.	Захист	23.06.2025	виконано

Студент

Науковий керівник роботи

І. В. Корчевой

С. В. Бех

(підпис)

(ініціали, прізвище)

(підпис)

(ініціали, прізвище)

РЕФЕРАТ

Пояснювальна записка дипломної роботи за обсягом становить 85 сторінок та містить 35 рисунків. Для дослідження було використано 27 бібліографічних найменувань.

Метою роботи є оцінка можливості застосування МІМО радару для виявлення об'єктів в умовах зашумленого середовища. Для досягнення мети було опрацьовано методику проведення та оцінки якості калібрування системи без спеціальної безлунної камери, розроблено і випробувано програмний пакет для адаптації ПЗ від ТІ цифрової обробки в середовищі MATLAB для застосування алгоритму покращення роздільної здатності, оцінено вплив перешкод на формування зображень в міліметровому діапазоні електромагнітних хвиль.

Предметом дослідження є методика калібрування та цифрова обробка даних МІМО радару для формування зображень в міліметровому діапазоні.

Об'єктом дослідження є каскадний FMCW МІМО радар з діапазоном робочих частот 77-81 ГГц.

Робота складається з чотирьох основних розділів та висновків. Перший розділ присвячено історії розвитку технології РАДАР, застосуванням та потенційним перевагам МІМО радарів, а також алгоритму Річардсона-Люсі.

В другому розділі описано теоретичну частину функціонування FMCW МІМО радару, визначення основних характеристик та їх роздільні здатності. Обговорено математичний аспект калібрування та цифрової обробки. В третьому розділі обговорюється експериментальне устаткування та основні частини програмної реалізації цифрової обробки в MATLAB. Результати проведених дослідів описано у четвертому розділі. Робота містить додатки з програмним кодом.

Ключові слова: *МІМО радар, бічні пелюстки діаграми спрямованості, функція розсіяння точки, віртуальний масив, калібрування.*

SUMMARY

The explanatory note of the thesis consists of 85 pages and includes 35 figures. A total of 27 bibliographic sources were used for the research.

The aim of the work is to evaluate the feasibility of using a MIMO radar for object detection in a noisy environment. To achieve this goal, a calibration methodology without a specialized anechoic chamber was developed and evaluated, a software package was created and tested to adapt TI's digital signal processing software for use in MATLAB, implementing a resolution enhancement algorithm, and the impact of interference on image formation in the millimeter-wave range was assessed.

The subject of the study is the calibration methodology and digital signal processing of MIMO radar data for image formation in the millimeter-wave range.

The object of the study is a cascaded FMCW MIMO radar operating in the frequency range of 77–81 GHz.

The work consists of four main chapters and conclusions. The first chapter is devoted to the history of RADAR technology development, the applications and potential advantages of MIMO radars, as well as the Richardson-Lucy algorithm.

The second chapter presents the theoretical aspects of FMCW MIMO radar operation, the determination of key parameters and their resolution. The mathematical aspects of calibration and digital processing are discussed. The third chapter focuses on the experimental setup and the main components of the digital processing implementation in MATLAB. The results of the conducted experiments are described in the fourth chapter. The thesis includes appendices with source code.

Keywords: *MIMO radar, sidelobes of the radiation pattern, point spread function, virtual array, calibration.*

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.	5
ВСТУП	6
РОЗДІЛ 1. Літературний огляд	9
1.1. Коротка історія розвитку технології РАДАР	9
1.2. Переваги МІМО радарів	12
1.3. Алгоритми супер-роздільної здатності.	14
1.4. Застосування МІМО радарів.	18
РОЗДІЛ 2. Теоретичні основи функціонування МІМО радару	21
2.1. Принцип FMCW радарів	22
2.2. Застосування вагових функцій	25
2.3. Принцип МІМО радарів	27
2.4. Калібрування.	32
РОЗДІЛ 3. Експериментальне устаткування	35
3.1. Радар	35
3.2. Методика та обладнання для калібрування	38
3.3. Цифрова обробка в середовищі MATLAB	42
РОЗДІЛ 4. Результати	44
4.1. Якість та вплив калібрування.	44
4.2. Зображення на тепловій карті точкового об'єкт з високим ЕПР .	47
4.3. Функція розсіювання точки МІМО радару та алгоритм деконволюції	
Річардсона-Люсі.	48
4.4. Модельні об'єкти	51
4.4.1. Модельні об'єкти в прямому полі зору	52
4.4.2. Зображення об'єктів замасковані завадами	57
4.5. Узагальнення результатів дослідів модельних об'єктів	59

ВИСНОВКИ.	61
ПЕРЕЛІК ПОСИЛАНЬ	62
ДОДАТКИ.	65
Додаток А. Код файлу головної програми.	65
Додаток Б. Код файлу програми симуляції функції розсіювання точки.	73
Додаток В. Код файлу програми оцінки якості калібрування	78

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

PSF, ФРТ — функція розсіяння точки (Point Spread Function)

SNR — співвідношення сигнал/шум (Signal-to-Noise Ratio), SNR

ULA — однорідний одновимірний масив (uniform linear array)

IF — проміжна частота (intermediate frequency)

ЧПІ — частота повтору імпульсів, PRF

MIMO — багато-вхідний багато-вихідний (Multiple-input multiple-output)

FMCW — неперервно випромінюючий частотно модульований

ЛЧМ — лінійна частотна модуляція (chirp)

CF — ялова частота (Carrier Frequency), f_c

SR — частота дискретизації (Sampling Rate), f_s

FTT, ШПФ — швидке перетворення Фур'є (Fast Fourier Transform)

ВСТУП

Технологія радару одразу після свого винайдення викликала зацікавленість як дослідників так і військових. Від простого визначення відстані до цілі, до більш складних застосувань, як от: визначення кутового положення, швидкості, форми або навіть матеріалу цілі, побудови траєкторії, ця технологія дає невичерпні можливості для забезпечення безпеки або автоматизації різноманітних задач. Здатність працювати за різних погодних умов і в умовах обмеженої видимості додатково збільшує кількість можливих застосувань.

Був пройдений довгий шлях розвитку цієї технології. Різні джерела по-різному встановлюють ключові досягнення в цій галузі. Для цієї роботи корисно виділити наступні етапи: поява моностатичних радарів, бістатичної конфігурації, логічне розвинення у вигляді фазових ґраток та МІМО (Multiple input - multiple output) радару. Головною ідеологією в цьому розвитку є покращення роздільної здатності та протидія маскуванню цілі або впливу середовища. Тому вивчення можливостей застосування МІМО радару для високоякісних зображень в міліметровому діапазоні є актуальною задачею.

Аналіз літературних джерел показує, що більшість досліджень МІМО-радарів зосереджена на виявленні об'єктів із значною ефективною площею розсіювання у відносно не зашумленому середовищі або рухомих об'єктів з малою ЕПР, які можуть бути ідентифіковані за рахунок доплерівського ефекту. Водночас питання формування якісних зображень статичних цілей у середовищі з перешкодами залишається недостатньо вивченим, що визначило мету даного дослідження – оцінити можливості застосування МІМО радару для виявлення об'єктів в умовах середовища з перешкодами.

Збільшення кількості передавальних і приймальних антен призводить до формування додаткових шляхів поширення сигналу, які називаються каналами передачі. Це вимагає належної калібровки системи, без якої недосконалість

пов'язані із процесом виробництва плати радару можуть суттєво погіршити робочі характеристики системи [1].

Досягнення поставленої мети потребувало вирішення комплексу взаємопов'язаних задач. Насамперед було необхідно опрацювати методику проведення та оцінки якості калібрування системи без використання безлунної камери. Наступним кроком стала розробка програмного пакету для адаптації програмного забезпечення від ТІ (Texas Instruments) в середовищі MATLAB з метою реалізації алгоритмів покращення роздільної здатності. Завершальним етапом була експериментальна оцінка впливу перешкод на формування зображень в міліметровому діапазоні електромагнітних хвиль.

Об'єктом дослідження виступає каскадний FMCW MIMO радар виробництва ТІ з діапазоном робочих частот 77-81 ГГц, що містить 12 передавачів та 16 приймачів, формуючи віртуальний масив із 134 унікальних фазових центрів. Предметом дослідження є методика калібрування та цифрова обробка даних MIMO радару для формування радіолокаційних зображень у міліметровому діапазоні.

Методологія дослідження базується на поєднанні експериментальних вимірювань з використанням радарної системи та методів цифрової обробки сигналів. Зокрема, застосовувалися швидке перетворення Фур'є для формування дальнісно-доплерівського портрету, вагові функції для зменшення бічних пелюсток діаграми спрямованості, а також ітеративний алгоритм деконволюції Річардсона-Люсі з регуляризацією методом повної варіації для покращення якості зображень.

Наукова новизна отриманих результатів полягає в систематичному дослідженні особливостей формування зображень MIMO-радаром статичних цілей у середовищі з перешкодами, а також у розробці оригінальної методики калібрування системи без використання безлунної камери.

Практичне значення роботи визначається можливістю застосування розробленої методики калібрування в польових умовах без спеціалізованого обладна-

ння, що розширює сферу використання МІМО-радарів. Створений програмний пакет забезпечує ефективну обробку радарних даних з можливістю адаптивного налаштування параметрів алгоритмів залежно від умов спостереження. Отримані результати можуть бути використані при розробці систем виявлення об'єктів для потреб безпеки, моніторингу територій та інших застосувань.

Основні результати дослідження пройшли апробацію на ХХІ Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених, що відбулася 14 травня 2025 року, де отримали позитивну оцінку наукової спільноти. За матеріалами роботи підготовлено публікацію у збірнику матеріалів конференції.

Структура роботи відображає логіку дослідження та складається з вступу, чотирьох розділів, висновків, переліку посилань із 28 джерел та додатків з програмним кодом. Перший розділ присвячено огляду літератури з історії розвитку радарних технологій, переваг МІМО-архітектури та алгоритмів супер-роздільної здатності. У другому розділі викладено теоретичні основи функціонування FMCW МІМО радарів та математичний апарат калібрування. Третій розділ описує експериментальне обладнання та програмну реалізацію цифрової обробки. Четвертий розділ містить результати експериментальних досліджень з аналізом ефективності запропонованих методів. Загальний обсяг роботи становить 85 сторінок, включаючи 35 рисунків та 3 таблиці.

РОЗДІЛ 1.

ЛІТЕРАТУРНИЙ ОГЛЯД

1.1. Коротка історія розвитку технології РАДАР

РАДАР (Radio detection and ranging) - пристрій для виявлення та визначення відстані до цілі. Роботу радару можна описати так: випромінення ЕМ хвилі, перевипромінювання сигналу від цілі в різні напрямки, отримання передавачем відбитого сигналу (відлуння), підсилення відлуння, подальша цифрова обробка.

Першими були моностатичні радари, тобто такі, в яких функцію передавача та приймача виконує одна антена.



Рис. 1.1. Схема типового моностатичного радару. За [2]

На рис. 1.1 показані ключові елементи. На верхній частині показано як влаштований передавач: генерує сигнал відомої форми та модуляції, після чого підсилює його. Дуплексер перемикає антену з режиму передавача в режим приймача. Перед обробкою відлуння підсилюється. Важливо, щоб підсилювач мав малі внутрішні шуми, аби зменшити хибну детекцію. Нижня частина схеми ілюструє конструкцію супергетеродинного приймача. Змішуючи вхідний

сигнал з локальним осцилятором, спеціально налаштованим на смугу частот радару, генерується сигнал на проміжній частоті (ПЧ). Цей сигнал, залежно від конструкції радару може підсилюватись, або проходити через узгоджений фільтр, або обробляти обидвома блоками. Квадратичний детектор фактично демодулює сигнал, переводячи його в низькочастотний спектр. Цей сигнал підсилюється і виводиться на дисплей, наприклад осцилограф.

На заміну імпульсним моностатичним радарам прийшли бістатичні радари, тобто такі, в яких передавачі та приймачі просторово рознесені. На відміну від перших, які використовують часове розділення для усунення інтерференції між переданим сигналом та відлунням, бістатичні радари можуть ефективно використовувати неперервне випромінювання завдяки просторовому розділенню передавача та приймача. Це дозволяє збільшити середню випромінювану потужність, що може покращити співвідношення сигнал/шум (SNR) та максимальну відстань роботи радару.

Однією з переваг цієї конфігурації є протидія технології «Stealth» рис. 1.2. Об'єкти, створені із застосуванням цієї технології, мають геометрію та покриття, які спрямовують більшість відбитої енергії в напрямках, відмінних від напрямку джерела опромінення. Оскільки передавач і приймач в загальному випадку орієнтовані по-різному, то стає можливим отримання енергії відбитої хвилі поза напрямком опромінення, зокрема від бічних пелюсток розсіювання цілі.

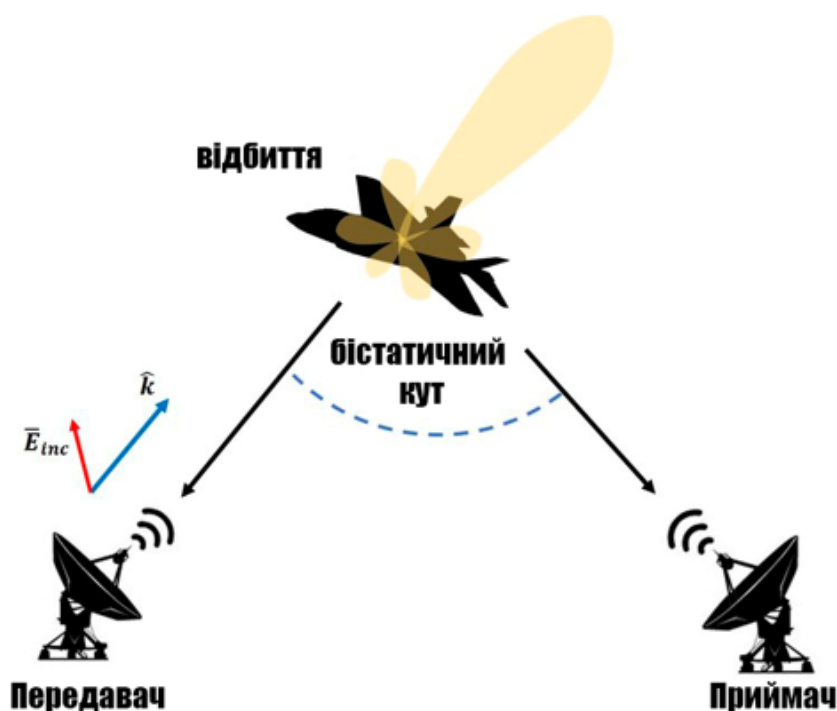


Рис. 1.2. Приклад протидії технології «Stealth»

Рознесена радіолокація активно застосовується у військовій сфері [3]. Зокрема, в зенітних ракетних системах з напівактивним самонаведенням. В даному випадку, передавач, в складі системи наведення, знаходиться на землі, а приймач встановлений на ракеті приймає сигнал, відбитий від цілі. Водночас, він також отримує сигнали від передавача. На основі цієї інформації ракета коригує свій курс. До таких зенітних комплексів відносяться, наприклад, комплекс «HAWK» та комплекс «Бук» (SA-11).

Очевидно, що чим більш направленою є діаграма спрямованості передавальної (приймальної) антени, тим більше електромагнітної енергії можна передати в певному напрямку. Такі антени потребують сканування простору шляхом безперервного її механічного повертання. Так звані фазові ґратки дозволяють усунути цей недолік. Фазова ґратка – це антена у вигляді масиву випромінювальних елементів, у якій напрям променя керується за допомогою зміни відносних фаз збуджувальних струмів, прикладених до окремих елементів. Часто, окрім фази сигналу регулюється ще й амплітуда.

До появи фазової ґратки, кутове положення цілі визначалось шляхом механічного повороту джерела випромінення, або у випадку бістатичного радару, приймача. Після їх появи, стало можливим електронно керувати напрямком променя. Як наслідок, стало можливе багатоцільове стеження. Окрім цього, зміна амплітуди або фази дозволяє реалізувати адаптивне пригнічення завад, налаштовуючи діаграму спрямованості радару.

Концепція фазової ґратки була відома ще за часів Першої світової війни, але практично була реалізована тільки під час Другої світової війни [4]. До-сі такі системи є дорого вартісними та потребують великої обчислювальної здатності для обробки.

1.2. Переваги МІМО радарів

Роздільна здатність фазової ґратки обмежена частотою та апертурою радару. Також додаткові обмеження накладені на відстань між елементами. Для однорідного лінійного масиву (ULA) відстань більша за $\lambda/2$ призводить до появи сильних періодичних бічних пелюсток (grating lobes) [4]. Отже, для досягнення високої роздільної здатності система має бути великою. Компактним та відносно недорогим рішенням є МІМО радар.

Слід окремо зазначити, що зі зростанням роздільної здатності, яка досягається шляхом звуження головної пелюстки діаграми спрямованості, зростає і загальний час, необхідний для сканування всього поля зору. Цей час визначається як добуток часу спостереження (dwell time) на кількість положень головної пелюстки, необхідних для покриття всього поля зору [5]. На відміну від цього, МІМО-радар здатен обробляти інформацію з усього поля зору протягом одного інтервалу спостереження, що суттєво підвищує ефективність сканування.

MIMO (Multiple-input multiple-output) радар, подібно до фазової ґратки, також являє собою масив окремих антен. В роботі [6], показано, що фазова ґратка, з точки зору енергії, відбитої від цілі, є еквівалентною до системи з одним джерелом випромінення. MIMO-радар, на відміну від цього, має кілька незалежних передавальних каналів. Для того щоб застосувати переваги цієї технології, радар має відрізнити ці сигнали одне від одного (вони мають бути ортогональними). Такий підхід формує віртуальний масив передавачів, набагато більший за реальний. Більш детально про віртуальний масив обговорено в другому розділі.

Ортогональність не є обов'язковою вимогою для MIMO-радарів [7]. Вона лише забезпечує некорельованість сигналів, що спрощує їх розділення на приймальній стороні. У випадках, коли відома структура передавальних сигналів (зокрема модуляція, форма імпульсів, спільна смуга частот тощо) та каналні характеристики (часові затримки, фазові зсуви, амплітудні коефіцієнти розповсюдження тощо), можливе застосування методів, таких як канална деконволюція, що дозволяє ефективно реалізовувати багатоканальні переваги MIMO-архітектур навіть за відсутності ортогональності між сигналами.

Загалом, MIMO радари можна розділити на дві великі групи: когерентні та статистичні. Когерентними називають радари в яких антени розташовані достатньо щільно, так, щоб при спостереженні цілі в дальньому полі ($R \geq \frac{2D^2}{\lambda}$, де D – розмір масиву, а λ – довжина хвилі) можна вважати, що сигнали надходять з узгодженими фазовими характеристиками. Статистичними називають радари з широко рознесеними антенами, так, щоб при спостереженні всі прийняті сигнали були незалежними по фазі (хоча можуть мати однакові статистичні характеристики, амплітуди).

Для обох груп справедливо, що MIMO радар може визначити в M_t разів більше цілей, ніж фазова ґратка [7]. Під M_t позначено кількість передавачів. Це пов'язано з додатковими степенями вільності, які надає ортогональність сигналів.

Для когерентних радарів перевагою є покращене SNR. За умови ортогональності переданих сигналів та когерентної обробки, коли сигнали від цілі відрізняються лише малою затримкою, підсумовування енергії забезпечує виграш у SNR, пропорційний кількості передавачів M_t .

Для статистичних радарів перевагою є підвищена стійкість до сцинтиляції ЕПР, тобто сильних флуктуацій інтенсивності відбитого сигналу залежно від орієнтації цілі. Зокрема, навіть незначна зміна кута нахилу цілі — наприклад, на одну тисячну радіана відносно горизонтальної площини — може зумовити варіацію амплітуди відбитого сигналу до 10 дБ [5]. Це може значно погіршувати SNR, або ускладнювати стеження за цілю. Для багатьох радарів це є критичною перешкодою, яка потребує застосування змін, таких, наприклад, як адаптивна зміна вагових коефіцієнтів, щоб відсікати напрямки, де сцинтиляції викликають найбільші збурення. Статистичні MIMO-системи, завдяки незалежності фазових спостережень з різних позицій, забезпечують диверсифікаційний ефект, що зменшує негативний вплив сцинтиляцій ЕПР та підвищує імовірність виявлення. Так, наприклад, втрата сигналу по одному каналу, може компенсуватись сигналом з іншого каналу.

Статистичні MIMO зазвичай є суттєво дорожчими і використовуються для бездротового зв'язку [8]. В даній роботі вони розглядатись не будуть.

1.3. Алгоритми супер-роздільної здатності

Не зважаючи на високу роздільну здатність, яку демонструють MIMO радары, якість зображення може значно погіршуватись через бічні пелюстки діаграми спрямованості. Це може проявлятися: в маскуванні слабкого сигналу цілі в головній пелюстці більш сильним відбиттям в бічній; зменшенням динамічного діапазону (здатності одночасно розрізняти цілі з різною ЕПР); збільшенні імовірності хибної тривоги.

Існує низка методів боротьби з бічними пелюстками, які застосовуються як на етапі проектування, так і під час обробки сигналів. На етапі проектування це, зокрема, вибір типу антени (наприклад, контрольоване викривлення геометрії дозволяє зосередити більше енергії в зоні головної пелюстки) та використання системи придушення бічних пелюсток (Sidelobe Blanking System) (рис. 1.3), оптимізація топології розташування приймачів та передавачів [9]. На етапі обробки сигналів застосовуються віконні функції, фільтр Вінера, а також урахування когерентного фактора, що дозволяє зменшити побічні компоненти у спектрі.



Рис. 1.3. Приклад системи придушення бічних пелюсток за [3]

Рис. 1.3 ілюструє одразу два методи придушення бічних пелюсток. Обирається дві антени, одна основна, інша, так звана допоміжна, в якій рівень бічних пелюсток вищий за рівень в першій. Порівнюючи відношення сигналів на приймачах з заданим порогом [10], приймається рішення або занулити сигнал (якщо співвідношення сигналів менше порогу), або подальше опрацювання головного сигналу (в протилежному випадку). Проблема цього методу полягає в тому, що для цілей з малою ЕПР, накладання завад може зменшувати співвідношення сигналів, як наслідок не досягати заданий поріг, призводячи до втрати корисної інформації. Оскільки не було можливості, якимось змінювати досліджуваний радар, то подальше обговорення рішень, які потребують модифікацію системи, не будуть обговорюватись.

Бічні пелюстки з'являються для всіх радарів скінченного розміру. Причини, які їх викликають є як фундаментальні обмеження (скінчена апертура, скінчена смуга частот), так і обмеження, які можна подолати (помилки в визначенні амплітуди/фази прийнятого сигналу, взаємний вплив між антенними елементами (*mutual coupling*), квантування сигналу при цифровій обробці). Якщо вплив з більшості перерахованих технічних обмежень можна мінімізувати через належне калібрування, та дотримання теореми Найквіста-Шеннона, то фундаментальні обмеження залишаються непереборними. Однак алгоритми супер-роздільної здатності дозволяють максимально ефективно використати доступну інформацію в межах цих фізичних обмежень, зокрема:

- наблизитися до теоретичної межі роздільної здатності
- зменшити рівень бічних пелюсток, що виникають внаслідок скінченної апертури
- покращити розділення близько розташованих цілей за рахунок використання апріорної інформації

Хоча алгоритми супер-роздільної здатності в першу чергу направлені на підвищення роздільної здатності, вони можуть опосередковано зменшувати вплив бічних пелюсток. Оскільки антенний патерн асоційований з дифракційним патерном [11], це наводить на думку застосувати функцію розсіювання точки (ФРТ). Яскравим представником такого підходу є нелінійний ітеративний алгоритм Річардсона-Люсі. Його головна ідея полягає в тому, аби послідовно застосовувати операцію зворотної згортки отриманого зображення та ФРТ. Оскільки цей алгоритм може спотворювати зображення після великої кількості ітерації [12], потрібні додаткові метрики оцінки якості, щоб вчасно його зупинити.

До інших поширених підходів відносять метод когерентного фактору [13] (Coherence Factor, CF), що дозволяє пригнічувати некогерентні компоненти

сигналу та покращувати контрастність зображення. Метод полягає в оцінці узгодженості фази сигналів від різних антен для кожної точки фокусування, що дозволяє підвищити селективність та пригнічувати паразитні відображення.

Також слід згадати методи адаптивного формування діаграми спрямованості — зокрема, алгоритм мінімізації середньої потужності з обмеженням на головний пелюсток (minimum variance distortionless response, MVDR) та алгоритм Capon. Вони дозволяють зменшити рівень бічних пелюсток за рахунок обчислення оптимальних вагових коефіцієнтів для кожного напрямку, адаптуючи фільтр до конкретної конфігурації сигналу та шуму. У роботі [14] продемонстровано реалізацію MVDR для ширококутового МІМО-радару з частотно-інваріантною діаграмою спрямованості, що дозволяє зменшити обчислювальну складність без втрати роздільної здатності.

Інший перспективний підхід — спарс-реконструкція (sparse reconstruction), зокрема методи Basis Pursuit або LASSO, які ґрунтуються на припущенні, що сцена містить лише обмежену кількість активних цілей (тобто значущих відгуків), розташованих у великому пошуковому просторі можливих положень. Інакше кажучи, сигнал, сформований радаром, можна подати як лінійну комбінацію базисних функцій (наприклад, відгуків від точкових джерел у всіх можливих положеннях), з яких лише мала частина має ненульові коефіцієнти.

Завдяки цьому підходу вдається точно реконструювати положення кількох близько розташованих цілей навіть за умов, коли класичні методи, засновані на швидкому перетворенні Фур'є (ШПФ) або адаптивному формуванні діаграми спрямованості, втрачають здатність розрізняти об'єкти через обмеження ширини головного пелюстка. Приклад успішного застосування методу представлено в роботі [15], де розріджена модель дозволила значно зменшити кількість базисних функцій при збереженні точності ідентифікації параметрів системи.

У сукупності, перелічені алгоритми дозволяють значно підвищити ефективність радарної системи без фізичного розширення апертури або збільшення частотного діапазону.

В загальному випадку зображення може бути представлено, як:

$$y(\tau, \theta) = \iint h(\tau - \tau', \theta - \theta') \cdot Y(\tau', \theta') d\tau' d\theta' + n(\tau, \theta) \quad (1.1)$$

де $Y(\tau, \theta)$, $y(\tau, \theta)$ – отримане та істинне зображення відповідно, $h(\tau, \theta)$ – ФРТ, $n(\tau, \theta)$ – шум, τ – часова затримка, θ – кутове положення.

Алгоритм Річардсона-Люсі можна подати як:

$$Y(\tau, \theta)_{k+1} = Y(\tau, \theta)_k \cdot \left(\frac{y(\tau, \theta)}{Y(\tau, \theta)_k \otimes h(\tau, \theta)} \otimes h^*(\tau, \theta) \right) \quad (1.2)$$

де $\otimes$ – операція згортки, $(\cdot)^*$ – операція комплексного спряження.

Більш детально цей алгоритм, та модельний вигляд для досліджувального радару розглянуто в розділі 4.3.

1.4. Застосування МІМО радарів

Оскільки МІМО радар розділяє випромінювальну потужність по каналам передачі, це призводить до зменшення можливої максимальної дальності. Це критично насамперед для когерентних МІМО. Тому більшість застосувань таких радарів зводяться до зйомки зображень на малій та середніх відстанях [16].

Також у роботі [16] представлено першу експериментальну реалізацію тривимірного МІМО-SAR, що базується на каскадному багаточиповому mmWave-радарі. Зображення формуються шляхом переміщення платформи по заданій траєкторії, що дозволяє отримати синтетичну апертуру. Завдяки поєднанню віртуальної просторової розгортки МІМО та механічного переміщення досягнуто бокової (поперечної) роздільної здатності приблизно 1 мм, що суттєво перевищує типові показники mmWave-систем.

Бокова роздільна здатність (lateral resolution) — це здатність системи розрізняти два близько розташованих об'єкти в напрямку, перпендикулярному

до променя радару (тобто в площині сцени). Вона визначається ефективною шириною апертури та критично важлива для якісного формування зображень.

В літературі можна знайти безліч застосувань:

детекція цілей з малою та надмалою ЕПР [17; 18], Зокрема, у [17] показано, що технологія МІМО дозволяє надійно виявляти БПЛА з низькою ЕПР, навіть у випадках маневрування та часткового перекриття. Це досягається завдяки високій кутовій роздільній здатності та гнучкому цифровому формуванню діаграми спрямованості.

в медицині для виявлення пухлин [19]. Оскільки МІМО підвищує число незалежних каналів, а отже й кількість ступенів вільності при реконструкції, МІМО-радар може бути застосований для виявлення аномалій у тканинах, де розсіювання сильно варіюється залежно від глибини та геометрії.

для автомобільного круїз-контролю. В роботі [20] демонструють, що навіть недорогі реалізації на одному друкованому модулі здатні на основі МІМО-принципів розрізняти складні сцени (пішоходи, транспорт, інфраструктура) завдяки покращеному кутовому профілю.

Також більшість застосувань є в міліметровому діапазоні, але цим не обмежується. У [20] вперше реалізовано високоточне двовимірне формування зображень з використанням МІМО сонару шляхом 2D-деконволюції, що дозволило досягти високої роздільної здатності за кутом і дальністю навіть при компактному розміщенні об'єктів.

Як видно із наведеного, переважно МІМО радары використовуються для виявлення та ідентифікації об'єктів із значним ЕПР (автомобілі) у відносно незашумленому середовищі або рухомих об'єктів з малим ЕПР (БПЛА), що можуть бути ідентифіковані за рахунок доплер ефекту. Водночас становить інтерес перевірка можливості виявлення металевих предметів на фоні землі, або замаскованих перешкодами. Очевидно, що в цьому випадку питання покращення якості зображень, що дає радар із допомогою алгоритмів деконволюції стає визначальним. Це і визначило основну мету дослідження, а саме: оцін-

ку можливості застосування MIMO радару для виявлення об'єктів в умовах зашумленого середовища. Для цього було поставлено завдання

- опрацювати методику проведення та оцінки якості калібрування системи каскадного FMCW MIMO радару від ТІ без спеціальної безлунної камери
- розробити і випробувати програмний пакет для адаптації ПЗ від ТІ цифрової обробки в середовищі MATLAB для застосування алгоритму покращення роздільної здатності
- оцінити вплив перешкод на формування зображень в міліметровому діапазоні електромагнітних хвиль.

РОЗДІЛ 2.

ТЕОРЕТИЧНІ ОСНОВИ ФУНКЦІОНУВАННЯ МІМО РАДАРУ

Для простого імпульсного моностатичного радару очевидно, що відстань до цілі визначається, як добуток швидкості світла на половину часу затримки, оскільки сигнал проходить шлях від радару до цілі та назад:

$$R = \frac{c\tau}{2} \quad (2.1)$$

Якщо посилати декілька імпульсів, то може виникнути невизначеність в розрахунку відстані, тобто не буде зрозуміло який по рахунку імпульс отриманий. Оптимальний варіант посилати наступний імпульс, після отримання сигналу від попереднього. В такому випадку максимальна відстань обмежена частотою повтору імпульсів (ЧПІ). Тривалістю переданого імпульсу знехтуємо.

$$R_{max} = \frac{c}{2PRF} \quad (2.2)$$

Для оцінки розділової здатності такого радару, тривалістю імпульсу T_p , нехтувати не можна.

$$R_{res} = \frac{cT_p}{2} \quad (2.3)$$

Отже, короткі імпульси забезпечують кращу роздільну здатність, але обмежують дальність через малу енергію. Рішенням цієї проблеми є внутрішньоімпульсна модуляція, яка дозволяє використовувати довгі імпульси з великою енергією, зберігаючи високу роздільну здатність. У таких радарх роздільна здатність визначається шириною смуги частот модуляції, а не тривалістю імпульсу. Завдяки високій піковій потужності вони краще підходять для великих відстаней.

Альтернативним рішенням є використання неперервно випромінюючих частотно модульованих (FMCW) радарів, які також використовують широку смугу частот для досягнення високої роздільної здатності, але працюють з неперервним випромінюванням. Хоча їх пікова потужність нижча, середня випромінювана потужність може бути порівнянною, що робить їх ефективними для малих та середніх відстаней при меншому енергоспоживанні.

2.1. Принцип FMCW радарів

Розглянемо випадок лінійної частотної модуляції. Загальна схема такого радару наведена на рис. 2.1.

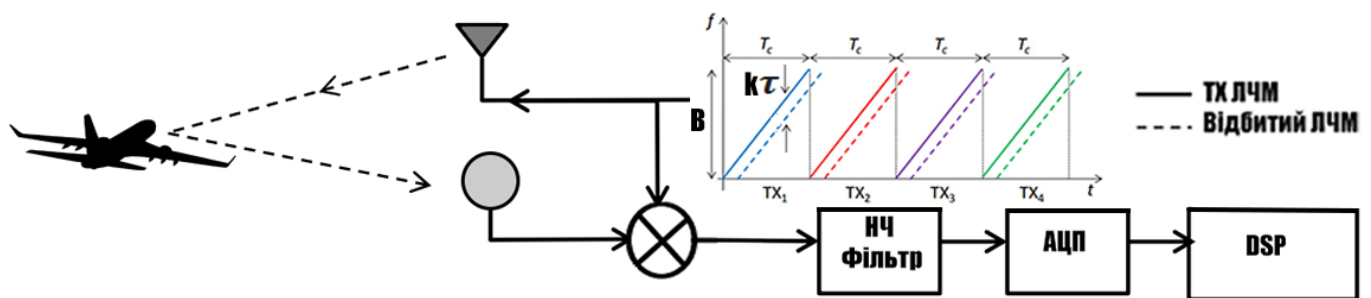


Рис. 2.1. Загальна схема FMCW радару за [21]

Як видно, схема є доволі простою в реалізації, що також є перевагою.

Передавач випромінює лінійно частотно модульований (ЛЧМ) сигнали (Tx ЛЧМ на рис. 2.1) зі швидкістю наростання/спадання частоти k . Після відбиття сигнал з приймача змішується із сигналом з передавача на гетеродині, в наслідок чого отримується проміжна частота (ПЧ). Фільтр низьких частот прибирає високочастотні компоненти ПЧ та шуми, дозволяючи оцифровувати лише низькочастотні складові ПЧ.

Розглянемо, яку корисну інформацію можна отримати з ЛЧМ сигналу. Математично його можна записати:

$$s(t) = \exp \left[2\pi j \left(f_c t + \frac{k}{2} t^2 \right) \right]; \quad t \in [-T_c/2; T_c/2] \quad (2.4)$$

де $s(t)$ – сигнал передавача, f_c – ялова частота.

Відбитий сигнал буде послабленою і зсунутою в часі версією переданого:

$$r(t) = A \cdot s(t - \tau) \quad (2.5)$$

де A – константа, яка враховує комбінацію втрат при розповсюдженні та коефіцієнт підсилення антени.

Після процесу демодуляції, фаза отриманої ПЧ записується:

$$\varphi_{\text{ПЧ}} = \varphi_t(t) - \varphi_r(t) = \varphi_r(t) - \varphi_r(t - \tau) \quad (2.6a)$$

$$= 2\pi \left(f_c t + \frac{kt^2}{2} \right) - 2\pi \left(f_c(t - \tau) + \frac{k(t - \tau)^2}{2} \right) \quad (2.6б)$$

$$= 2\pi \left[f_c t + \frac{kt^2}{2} - f_c t + f_c \tau - \frac{kt^2}{2} + \frac{k\tau^2}{2} - k\tau t \right] \quad (2.6в)$$

$$\approx 2\pi \left[f_c \frac{2R}{c} - \frac{k}{2} \left(\frac{2R}{c} \right)^2 - 2k \frac{R}{c} t \right] \quad (2.6г)$$

$$\approx 2\pi \left[2f_c \frac{R}{c} + 2 \frac{B}{T_c} \frac{R}{c} t \right] \quad (2.6д)$$

де B – ширина смуги частот.

Ми нехтуємо другим доданком в 2.6г, оскільки від обернено пропорційний c^2 . З рівняння 2.6д ми можемо знайти відстань, як:

$$R = \frac{f_{\text{ПЧ}} T_c c}{2B} \quad (2.7)$$

Ще одна перевага FMCW радарів полягає в тому, що немає потреби в квантуванні всієї ширини смуги частот, а лише проміжні частоти. Це дозволяє використовувати більшу ширину смуги, тобто мати кращу роздільну здатність.

Отже, для визначення відстані отриманий сигнал треба перевести з часового простору в частотний, піддавши перетворенню Фур'є. З теореми невизна-

ченості, витікає, що за час спостереження T_c можна розрізнити сигнали, які відрізняються на частоту $1/T_c$. Виконуючи нескладні перетворення, отримуємо:

$$\Delta f = \frac{1}{T_c} = \frac{2kR_{res}}{c} \Rightarrow R_{res} = \frac{c}{2kT_c} = \frac{c}{2B} \quad (2.8)$$

Додавання лінійної модуляції дозволяє підвищити роздільну здатність в BT разів [22]. Це видно, якщо знайти співвідношення між рівнянням 2.3 та 2.8, за умови $T_c = T_p$.

Оскільки ширина смуги ПЧ для максимальної відстані обмежена частотою квантування рис.2.2, то максимальну відстань можна подати як:

$$R_{max} = \frac{cf_{samp}T_c}{B} = \frac{cf_{samp}}{k} \quad (2.9)$$

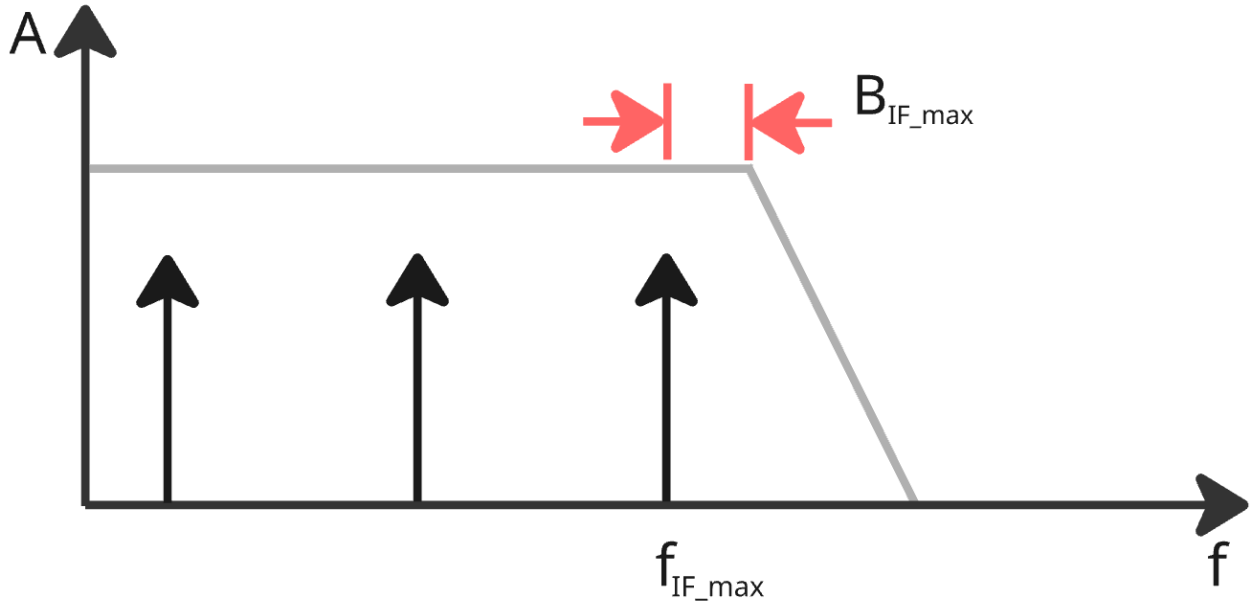


Рис. 2.2. Ілюстрація обмеження ширини смуги для максимальної ПЧ

Фазовий зсув в рівнянні 2.6 використовується для знаходження кутового положення цілі. Визначення цієї характеристики більш детально розглянемо в розділі 2.3, застосувавши парадигму МІМО радару до виведених формул.

2.2. Застосування вагових функцій

Як показано в [22] та на рис.2.3, автокореляційна функція ЛЧМ сигналу має високі бічні пелюстки. Оскільки, за теоремою Вінера-Хінчена автокореляційна функція пов'язана зі спектральною густиною через Фур'є перетворення, то ці бічні пелюстки проявляються як небажані спектральні компоненти в перетворенні Фур'є ПЧ сигналу FMCW радару, що погіршує якість отриманого дальнісного профілю. Такі бічні пелюстки називають дальнісними пелюстками (range sidelobes).

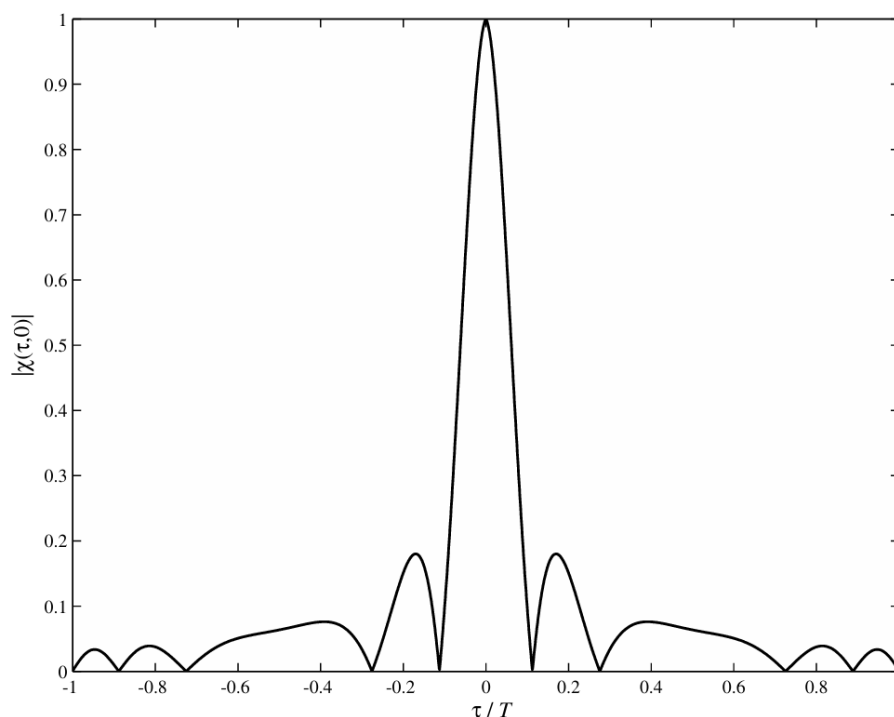


Рис. 2.3. Автокореляційна функція ЛЧМ сигналу для статичних цілей. $BT = 10$. [22]

Рішенням цієї проблеми є зміна форми спектру ПЧ, з прямокутної на більш згладжену по краях. Це може бути досягнуто двома способами – амплітудним або частотним зважуванням. Головна ідея останнього полягає в нелінійній частотній модуляції, а саме в тому, що більше часу радар випромінює в інтервалі біля одних частот, та менше часу біля інших, що призводить до зміни спектральної густини.

Амплітудне зважування навпаки, застосовне лише для ЛЧМ. Воно використовує лінійну залежність між миттєвою частотою і часом $f(t) = f_0 + (B/T)t$. Коригуючи амплітуду в певний момент, можна як наслідок коригувати спектральну щільність для частоти, яка в цей момент випромінюється. Отже, все, що потрібно зробити, – це задати форму сигналу по амплітуді.

Якщо застосовувати вагові функції лише до сигналу приймача, то погіршується SNR, оскільки фільтр перестає бути узгодженим. Для цього застосовують квадратний корінь вагової функції як для приймача, так і для передавача. Обидва методи забезпечують суттєве зниження бокових пелюсток, але застосування функції лише на приймачах демонструє трохи гірші результати [22].

Очевидно, згладжування спектру буде призводити до більшої концентрації енергії поблизу центра, що як наслідок призведе до розширення головного пелюстку та втрат у роздільній здатності.

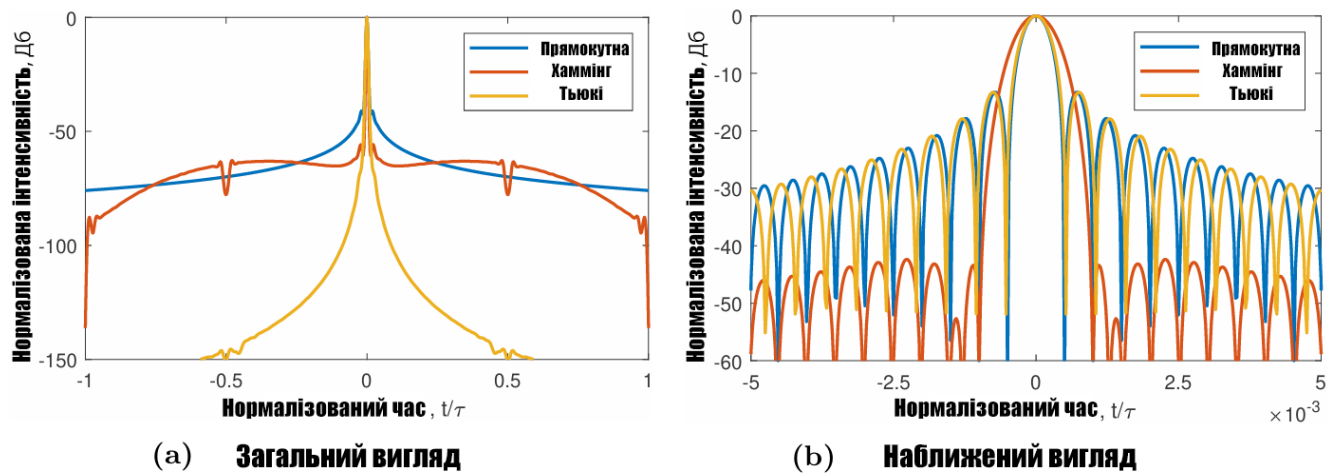


Рис. 2.4. Приклади застосування різних зважувальних функцій. $BT = 2000$ [1]

Вагова функція Хаммінга є однією із найпоширеніших у цифровій обробці сигналів завдяки гарному компромісу між шириною головної пелюстки та пригніченням бічних. Вона має вигляд:

$$w_{Ham}(n) = 0.54 - 0.46 \cos \left(\frac{2\pi n}{N-1} \right) \quad (2.10)$$

де N – кількість відцифрованих зразків.

Вагова функція Тьюкі, також відоме як косинусне вікно з параметром, є узагальненням сімейства віконних функцій на основі косинусоїд. Воно включає в себе як часткові випадки прямокутне вікно (при параметрі $\psi = 0$), так і вікно Ганна (при $\psi = 1$), що дозволяє гнучко налаштовувати компроміс між роздільною здатністю та пригніченням бічних пелюсток спектра. Вона має вигляд:

$$w_{Tukey}(n) = \begin{cases} \frac{1}{2} \left(1 + \cos \left(\frac{2\pi}{\psi} \left[\frac{n}{N-1} - \psi/2 \right] \right) \right) & , 0 \leq n \leq (N-1)\frac{\psi}{2} \\ 1 & , (N-1)\frac{\psi}{2} \leq n \leq (1 - \frac{\psi}{2})(N-1) \\ \frac{1}{2} \left(1 + \cos \left(\frac{2\pi}{\psi} \left[\frac{n}{N-1} - 1 + \psi/2 \right] \right) \right) & , (1 - \frac{\psi}{2})(N-1) \leq n \leq N-1 \end{cases}$$

де ψ – параметр, який визначає частку довжини вікна, на якій застосовується косинусне згладжування. На рис. 2.4, $\psi = 0.08$, що означає, лише 4% з кожного боку ширини смуги відцифрованих зразків було згладжено. Для випадку зображеному на рисунку, застосування прямокутної функції дає бічні пелюстки на рівні -13.2 дБ, в той час як функція Хаммінга зменшує їх до рівня -42.5 дБ.

2.3. Принцип МІМО радарів

Одне з ключових понять, необхідних для розуміння МІМО радару, – це віртуальний масив. Щоб краще його зрозуміти розглянемо простий приклад.

Нехай в нас є ULA-масив з трьох антен, розташованих з міжантеною відстанню d в напрямку $\mathbf{d}$, які одночасно виконують функцію як приймача, так і передавача. Нехай також у нас є ціль, яка знаходиться за координатами (R, φ) , де перша координата відповідає відстані до цілі, а друга – азимуту. Початок координат знаходиться в центрі масиву. Позначимо для зручності:

$$\eta = k\mathbf{u} \cdot \frac{\mathbf{d}}{\|\mathbf{d}\|} \quad (2.11)$$

де k – хвильовий вектор, $\mathbf{u}$ – напрямок спостереження, $\|\cdot\|$ – евклідова норма.

Тоді матрицю фазових коефіцієнтів, які відповідають фазовим зсувам від цілі для кожної пари передавач-приймач, можна записати у вигляді:

$$\begin{pmatrix} e^{j\eta 2d} & e^{j\eta d} & e^{j\eta 0} \\ e^{j\eta d} & e^{j\eta 0} & e^{-j\eta d} \\ e^{j\eta 0} & e^{-j\eta d} & e^{-j\eta 2d} \end{pmatrix} \quad (2.12)$$

Як видно зі структури 2.12 найбільші фазові затримки $e^{j\eta 2d}$ та $e^{-j\eta 2d}$ відповідають ефективним позиціям, що виходять за межі фізичного масиву, тобто віртуальні антени розташовані поза межами реальних антен. Це і є головна ідея віртуального масиву.

Також зі структури 2.12 видно повтор деяких елементів, тобто віртуальні центри накладаються. Це більш чітко зрозуміло з рис.2.5

В загальному випадку масив з n елементів може генерувати $2n - 1$ віртуальних фазових центрів. Для збільшення їх кількості можна варіювати відстань між антенами, але навіть при правильній топології розташування антен, через симетричність матриці 2.12 обмежено числом $n(n + 1)/2$. Вирішенням проблеми накладання, – є розділення приймачів і передавачів. В такому випадку кількість віртуальних фазових центрів може досягати $M_t \cdot M_r$, де M_t – кількість передавачів, M_r – кількість приймачів.

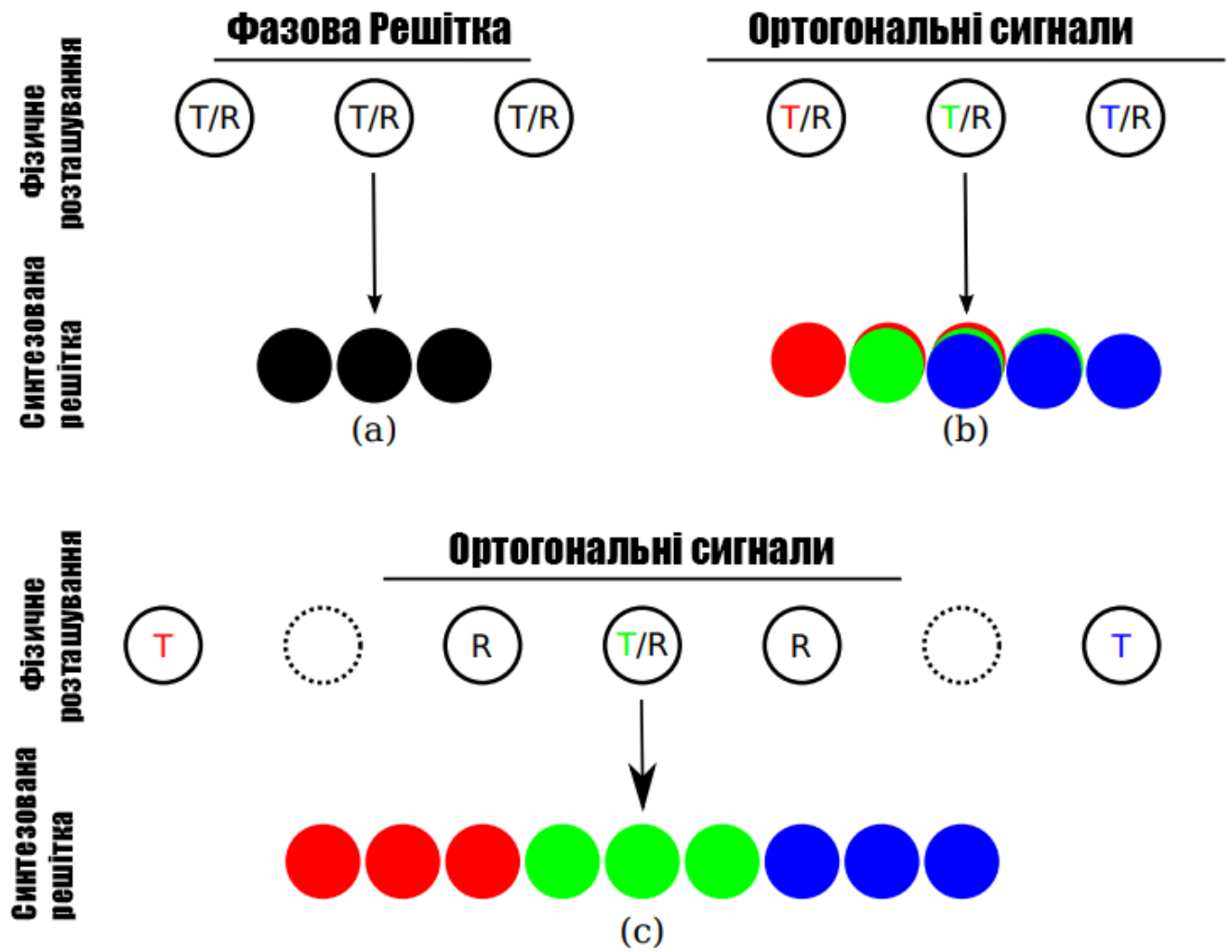


Рис. 2.5. Порівняння розташування віртуальних елементів для різних конфігурацій за [23]. (a) – фазові центри для фазової ґратки, (b) – для МІМО з моностатичними антенами, (c) – для МІМО, одна моностатична антена, дві бістатичні пари. Правильне розташування антен прибирає накладання елементів віртуального масиву.

Для довільного розташування антен, не залежно чи є вони моностатичними, чи функції передавача та приймача розділені, фазові зсуви визначаються як:

$$e^{j\eta(\mathbf{x}_m + \mathbf{y}_n)} \quad (2.13)$$

де $\mathbf{x}_m$ та $\mathbf{y}_n$ – 3-вимірні координати m -того передавача та n -того приймача. Розташування віртуальних центрів (вони є моностатичними) визначається як:

$$x_{mn} = \frac{\mathbf{x}_m + \mathbf{y}_n}{2} \quad (2.14)$$

Кутова роздільна здатність повністю визначається конфігурацією антен МІМО радару. Розглянемо як знаходиться кутове положення цілі.

Як зазначалось в розділі 2.1, для визначення кутового положення виконується шляхом аналізу частотних зсувів отриманого сигналу. Фазовий зсув в рівнянні 2.6 використовується для визначення швидкості цілі. Оскільки нас цікавлять статичні цілі, то детально визначення швидкості розглядатись не буде.

Розглянемо приклад визначення кутового положення для ULA з одним передавачем та M_r приймачами (рис. 2.6).

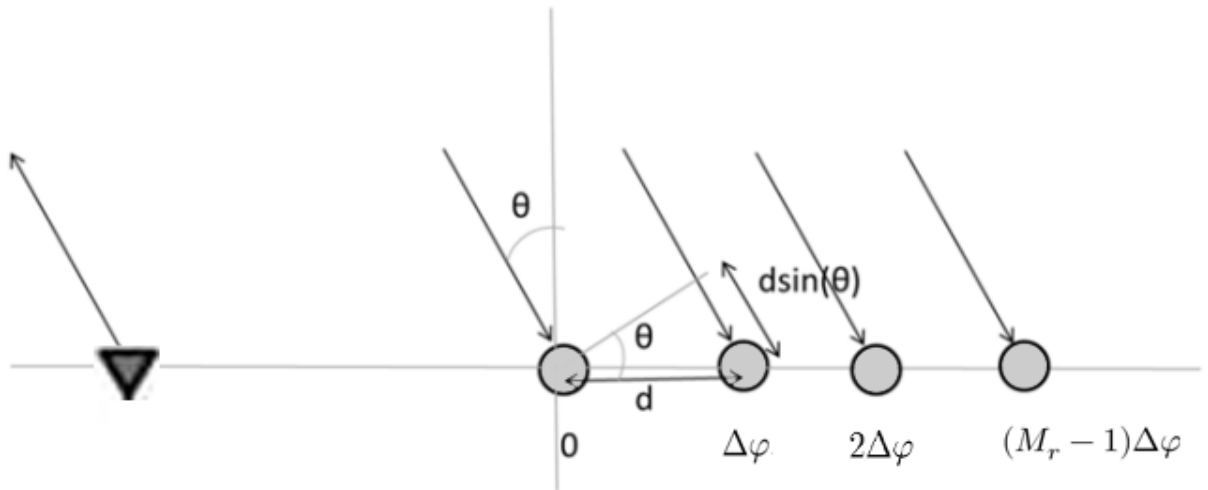


Рис. 2.6. Знаходження кутового положення за [24]. Трикутник – передавач, кружечки – приймачі

Різниця фаз між кожними передавачами є сталою, отримуємо масив:

$$[0, \Delta\varphi, 2\Delta\varphi, \dots, (M_r - 1)\Delta\varphi] \quad (2.15)$$

З якого отримуємо:

$$\theta = \arcsin\left(\frac{\Delta\varphi\lambda}{2\pi d}\right) \quad (2.16)$$

Однозначне визначення різниці фаз лежить в межах $[-\pi, \pi]$, отже з 2.16 видно, що максимально поле зору в $\pm 90^\circ$ досягається при між антенній відстані $\lambda/2$.

Очевидно, рівняння 2.16 не застосовне, в випадку коли дві цілі знаходяться на однаковій відстані та мають однакову швидкість. В загальному випадку, проводиться два швидких перетворення Фур'є (ШПФ), перше по часу (для визначення відстані), друге по ЛЧМ (slow time). Після 2-д Фур'є перетворення отримується двовимірний далекомірно-доплерівський портрет. Для кожного приймача буде пік на одній і тій самій частоті, але з різною фазою. Оскільки вони потрапляють в одну й ту саму далекомірно-доплерівську комірку, отримана різниця фаз є сумою фазових зсувів обох цілей. В такому випадку, застосовують перетворення Фур'є для послідовності фазорів, отриманих з далекомірно-доплерівського портрету, обраної комірки. Це зветься кутове ШПФ.

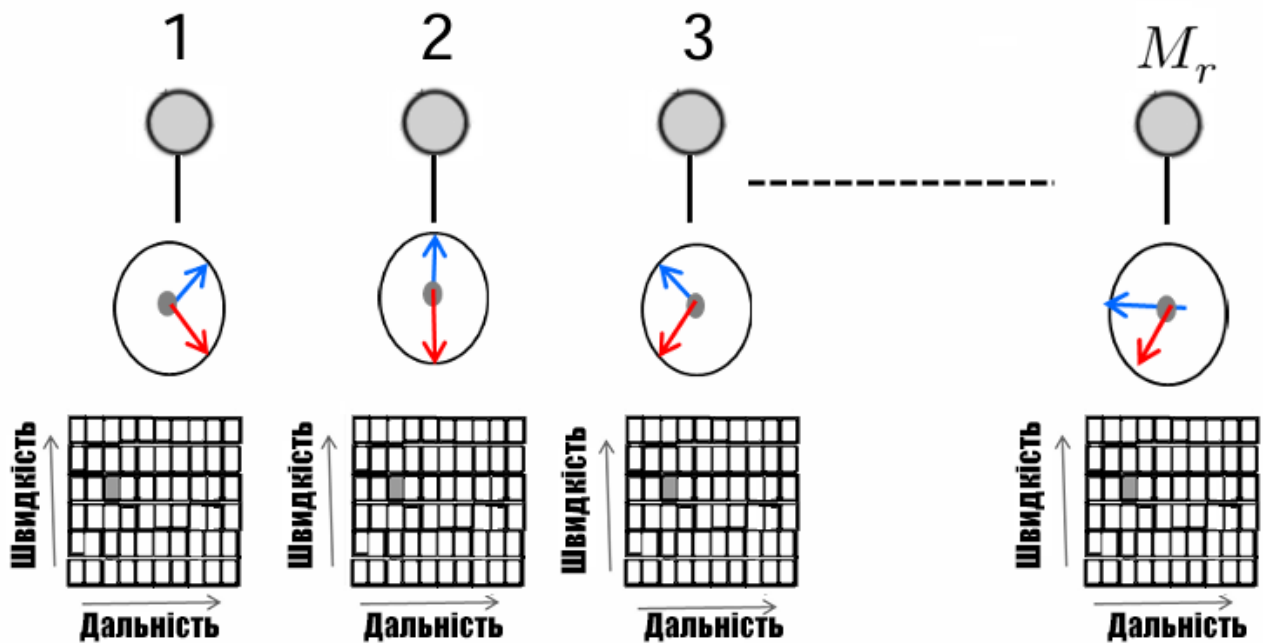


Рис. 2.7. Кутове ШПФ за [21]. Обирається одна комірка далекомірно-доплерівський портрету та по послідовності цієї комірки з різних приймачів проводиться ШПФ. Допомогає розділити фазові зсуви двох цілей (синя та червона стрілки)

Для знаходження кутової роздільної здатності, накладемо на зсув фаз обмеження, – вона має бути більша ніж $2\pi/M_r$ (розмір однієї комірки в радіа-нах). Тоді:

$$\Delta\varphi = \frac{2\pi d}{\lambda} \cos(\theta) \Delta\theta > \frac{2\pi}{N} \Rightarrow \theta_{res} = \frac{\lambda}{M_r d \cos(\theta)} \quad (2.17)$$

2.4. Калібрування

Необхідно враховувати недосконалості, які можуть проявлятися у вигляді дисбалансу каналів або взаємного зв'язування. Під першим мається на увазі небажані зміни амплітуди та фази між каналами антенної решітки. Друге, у свою чергу, виникає тоді, коли окремі канали електромагнітно впливають один на одного. Оскільки фізичні передавальні та приймальні канали формують спільну віртуальну решітку, похибки окремих каналів накопичуються та повторюються у віртуальній структурі [1].

Першочерговою цілю калібрування є усунення недосконалості, які неминуче виникають при виробничому процесі. Наприклад, в наслідок небажаної девіації доріжок Printed Circuit Board (PCB). В каскадних системах калібрування ускладнюється, адже важливим є синхронізація декількох чіпів.

Компанія ТІ надає програмне забезпечення для реалізації калібрування, що і було використане. Суть метода полягає у встановленні нахилоного під кутом в 45° кутового відбивача на відомій відстані, на висоті центру каскаду. Компанія рекомендує проводити калібрування в безлунній камері, що на практиці не завжди може бути зробленою. Тому однією із задач роботи було розробити ефективний спосіб калібрування без використання такої камери. В першу чергу запропоноване калібрування націлене на покращені кутового визначення цілі [25].

На вхід програми подаються необроблені експериментальні дані. Програма виконує детекцію піків ШПФ у межах ± 1 м від заданої еталонної відстані та формує матрицю номерів комірок (бінів) дальнісного профілю, яким відповідають ці піки. Отримана матриця має розмірність $M_t \times M_r$ (у нашому випадку 12×16) та використовується для частотного калібрування. Паралельно формується ще одна матриця такого ж розміру, що містить комплексні значення амплітуд детектованих піків і застосовується для калібрування амплітуди та фази.

Номер комірки для піку ШПФ пари Tx_1-Rx_1 обирається як референсний. На основі цього обчислюється матриця зсувів пікових комірок відносно референсної:

$$\Delta P_{ij} = P_{ij} - P_{\text{ref}}, \quad \Delta \mathbf{P} \in \mathbb{Z}^{M_t \times M_r}$$

Частотне калібрування виконується шляхом розрахунку фазових поправок:

$$F_{ij} = 2\pi \cdot \Delta P_{ij} \cdot \frac{\frac{f_{\text{samp}}^{\text{calib}}}{f_{\text{samp}}^{\text{chirp}}} \cdot \frac{k_{\text{chirp}}}{k_{\text{calib}}}}{N \cdot I_{\text{inter}}}$$

де:

- $f_{\text{samp}}^{\text{chirp}}$ — частота квантування при записі поточного сигналу;
- $f_{\text{samp}}^{\text{calib}}$ — частота квантування при калібруванні;
- k_{chirp} — швидкість наростання частоти ЛЧМ для поточного сигналу;
- k_{calib} — швидкість наростання частоти ЛЧМ, використана при калібруванні;
- N — кількість оцифрованих відліків (довжина сигналу);
- I_{inter} — коефіцієнт інтерполяції.

Калібрування по амплітуді та фазі :

$$C_{ij}^{calib} = \frac{C_{ref}}{C_{ij}} \quad (2.18)$$

де C_{ref} – комплексне значення піку референсного каналу, C_{ij} – комплексне значення піку пари і-того передавача та j-того приймача

Калібрування сигналів здійснюється множенням на фазовий множник по осі відліків $n = 0, \dots, N - 1$:

$$s_{ij}^{calib}[n] = s_{ij}[n] \cdot \exp(-j \cdot F_{ij} \cdot n) \cdot C_{ij}^{calib}$$

де $s_{ij}[n]$ — комплексне значення сигналу для пари Tx_i–Rx_j.

РОЗДІЛ 3.

ЕКСПЕРИМЕНТАЛЬНЕ УСТАТКУВАННЯ

3.1. Радар

В роботі використовувався каскадний FMCW MIMO радар від компанії Texas Instruments, вигляд якого зображено на рис. 3.1.

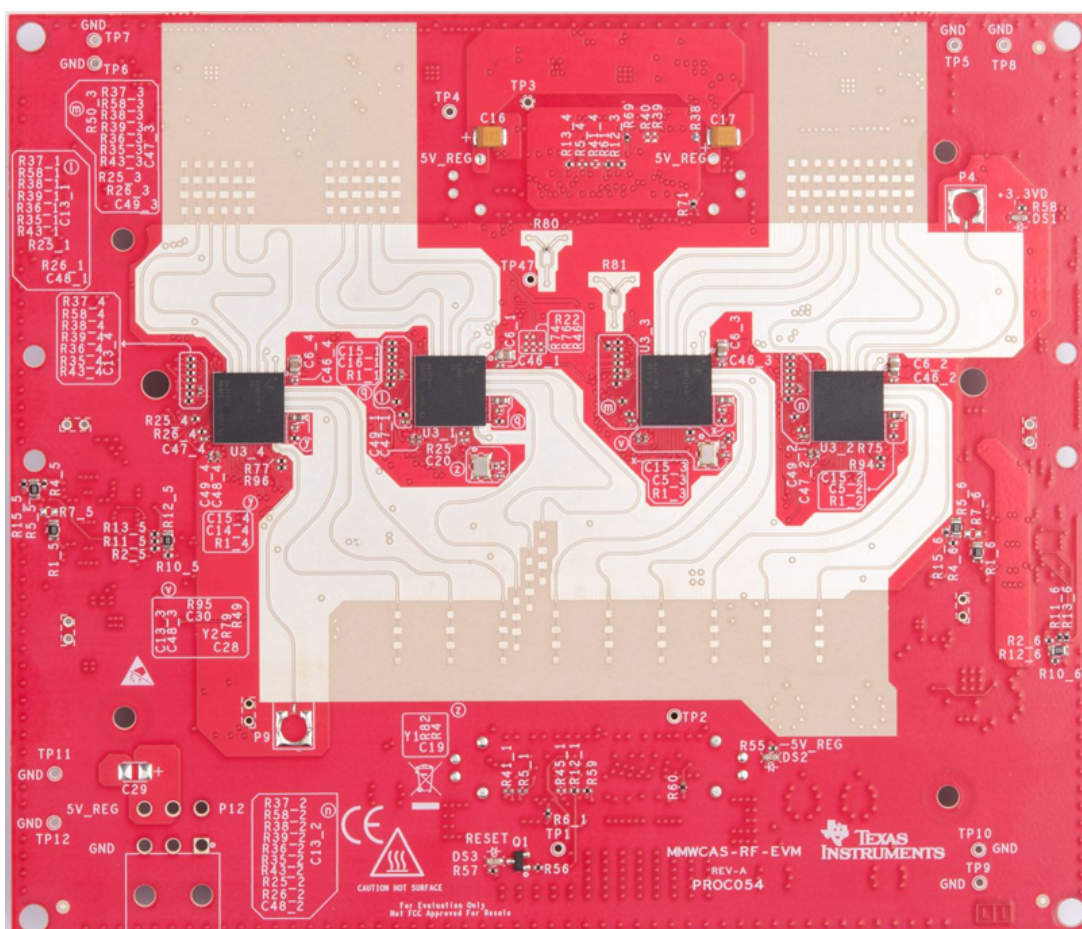


Рис. 3.1. Вигляд радару. Другий чіп зліва є головним, всі інші підлеглими

Каскад складається з головного та трьох підлеглих чіпів рис. 3.1. Має 12 передавачів та 16 приймачів схематичне розташування яких показано на рис. 3.2.

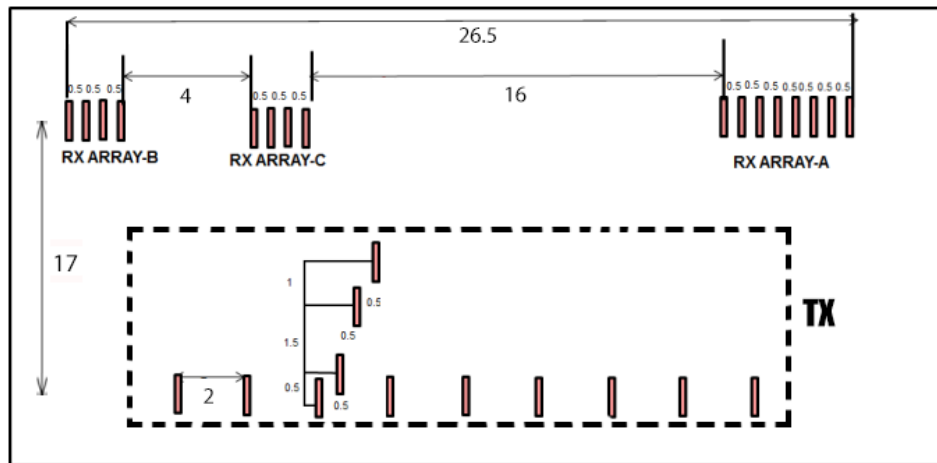


Рис. 3.2. Схема розташування передавальних та приймальних антен за [26]. Відстані подані в довжинах хвиль.

Передавальні антени головного чіпу розташовані з вертикальним відступами, щоб забезпечити розділення цілей по куту підйому. Відступи вибрані методом масиву мінімальної надмірності (Minimum Redundancy Array), для того, аби максимально зменшити чисто фазових центрів, які накладаються.

Реальному розташуванню антен відповідає віртуальний масив фазових центрів зображений на рис. 3.3

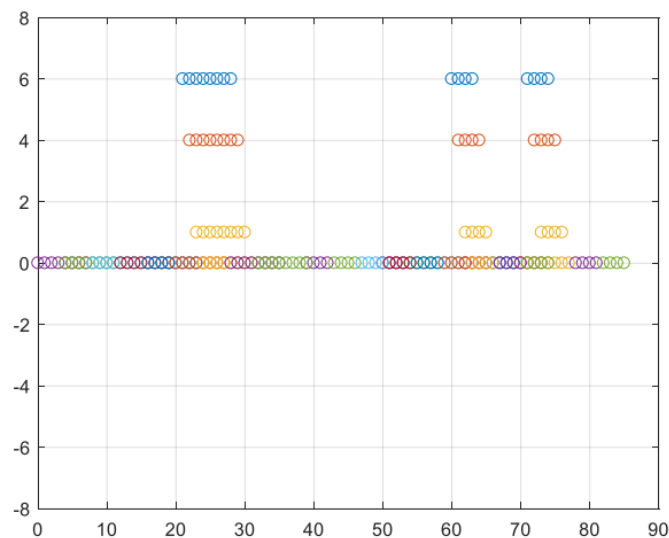
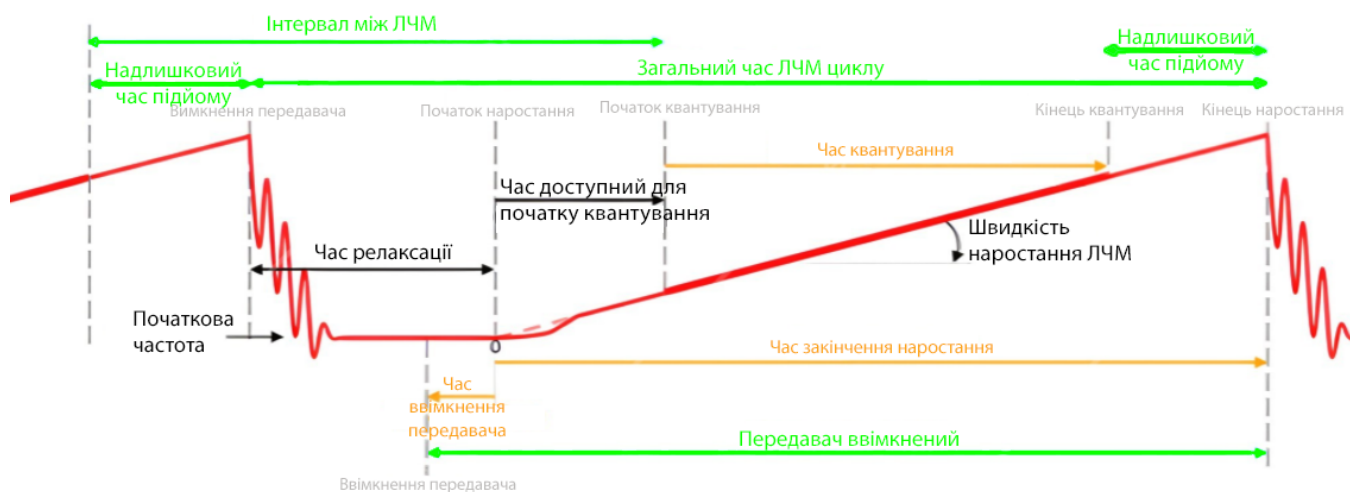


Рис. 3.3. Схема розташування віртуального масиву фазових центрів. Фазові центри що накладаються не відображені. Відстані подані в половиних довжин хвиль

Всього 134 унікальних канали: 86 по азимуту та 48 по куту підйому. Така конфігурація забезпечує кутову роздільну здатність в 1.4° по азимуту та 18°

по куту підйому. Для цілей з ЕПР близько 10 м^2 максимальна детектувальна відстань складає 150 м.

Розглянемо питання формування ЛЧМ сигналу. Рис.3.4 ілюструє основні етапи цього процесу:



ЗЕЛЕНЕ: не реєструється. Лише для ілюстрації

ЧОРНЕ: повністю налаштовується для кожного ЛЧМ (через оперативну пам'ять конфігурації ЛЧМ)

ПОМАРАНЧЕВЕ: для кожного ЛЧМ можна вибрати одне з 4 значень, відповідно до профілю ЛЧМ

Рис. 3.4. Формування сигналу ЛЧМ

У сучасних FMCW MIMO радарях організація передавання та приймання сигналів відбувається у вигляді фреймів, кожен з яких містить упорядковану у часі послідовність ЛЧМ. Фрейм є елементарною часовою одиницею спостереження, що повторюється з визначеною частотою, і в межах якої відбувається накопичення сигналів, необхідних для просторово-часової обробки, включаючи побудову дальнісного-доплерівського портрету та формування апертури.

У випадку часового мультиплексування TDM (Time Division Multiplexing), яке застосовувалося в цій роботі, до формування віртуальної апертури у MIMO-системах, кожен передавач активується послідовно у межах фрейму. Для цього ЛЧМ з різних передавачів розташовуються у часі один за одним — тобто формують послідовність із TDM-структурою.

На рис.3.4 зображено формування цієї послідовності. Задається початкова частота, час релаксації, початку та кінця квантування, наростання ЛЧМ,

ввімкнення передавача та швидкість наростання. Помаранчевим показані характеристики, які можна групувати в профілі ЛЧМ для використання різних профілів в межах одного фрейму. Це більш просунутий метод, який дозволяє адаптивно отримувати дані залежно від цілі спостереження. В роботі розглядається формування зображень тільки з використанням одного, основного, профілю.

3.2. Методика та обладнання для калібрування

Калібрування потребує високої якості кутового відбивача, адже, шорсткість поверхні, невідповідність кутів між гранями можуть призводити до спотворень частотних та фазових характеристик отриманого сигналу. Для створення кутового відбивача було задіяне обладнання та ресурси науково-дослідницького підрозділу компанії ТОВ "PlasmaTec".

Було розроблено 3-д модель в програмі AutoCAD рис. 3.5.

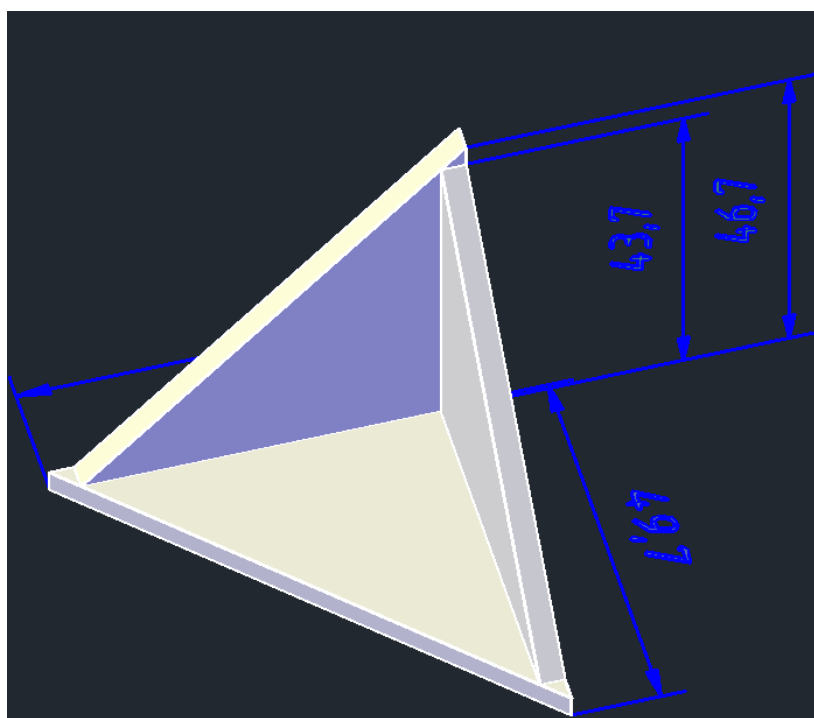


Рис. 3.5. Модель кутового відбивача, яка враховує метод зварювання та товщину металу.
Розміри в мм

Ефективна площа розсіювання визначається формулою [3]:

$$\sigma = \frac{4\pi}{3\lambda^2} a^4 \quad (3.1)$$

де a – довжина ребра. Відповідно до нашої довжини хвилі (3.89 мм) було вироблено два кутових відбивача з ЕПР 1 м² з довжиною ребра 43,7 мм та один кутовий відбивач з ЕПР 8 м² з довжиною ребра 73,4 мм. Точність в ± 0.1 мм була досягнута за допомогою установки для лазерного різання металу з числовим програмним керуванням (ЧПК).

Оскільки зварювання було з нахльостом, то для встановлення внутрішніх розмірів, всі три грані мали бути різних розмірів (a , $a + h$ та $a + 2h$, де h – товщина металу). В нашому випадку використовувалась нержавіюча сталь товщиною 3 мм.

Для дотримання куту при зварюванні було розроблено спеціальну оснастку рис. 3.6. Яка складалась з металевого куба та тримачів.

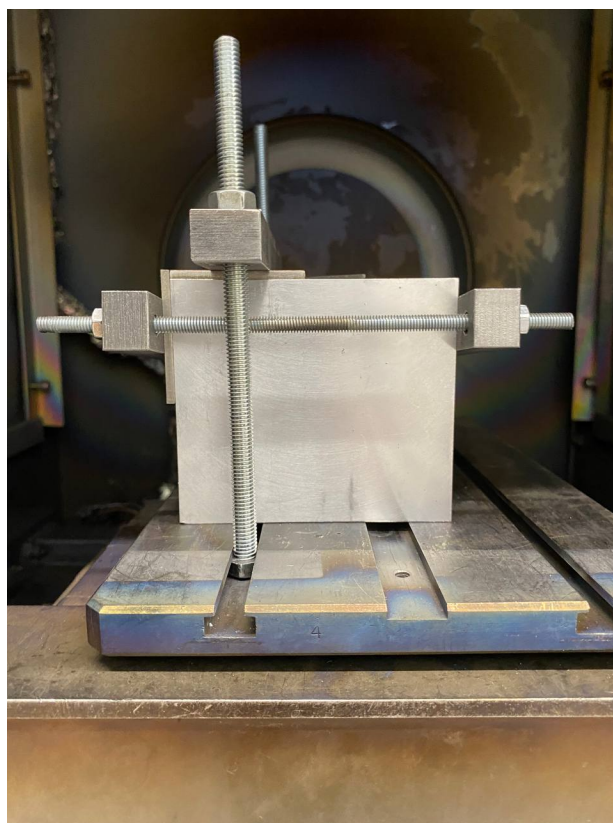


Рис. 3.6. Оснастка в камері електронно-променевої зварювальної (ЕПЗ) установки

Перед зварюванням проводилось полірування вирізаних деталей за допомогою шліфувальної машинки з абразивним диском. Послідовно використовувались диски з зернистістю 500, 1500, 2500.

Після зварювання методом ЕПЗ, було проведено чисельний вимір відхилення кутів граней від 90° . Для цього відполірований кутовий відбивач встановлювався у ємність з водою та опромінювався лазером. Далі, графічним редактором встановлювались відхилення колінеарності падаючого та відбитого променя.

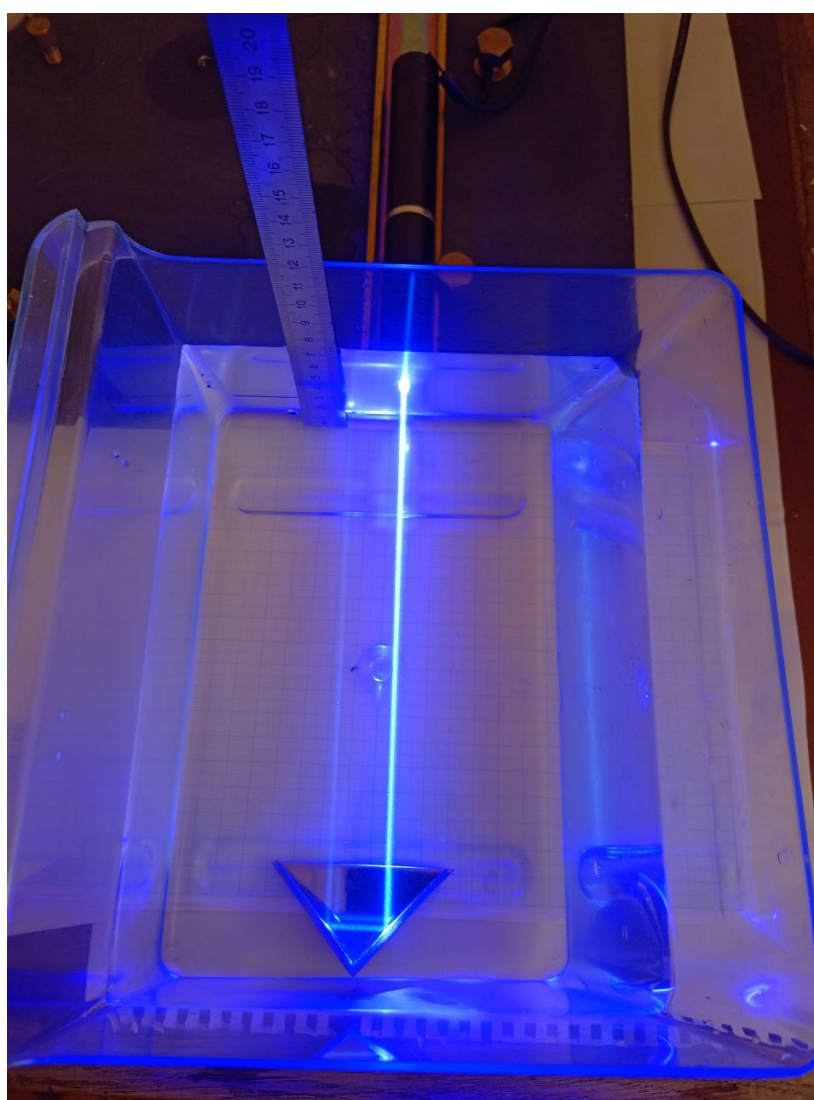


Рис. 3.7. Перевірка відхилення кутів граней кутового відбивача від 90°

Найкращі результати склали $\{0.7^\circ, -0.1^\circ, -0.6^\circ\}$, саме це кутовий відбивач і використовувався в подальшому для калібрування.

Оскільки, калібрування потребує відсутності поблизу кутового відбивача інших сильних відбивачів, без доступу до безвідлунної камери, було прийнято рішення використовувати квадратний алюмінієвий профіль, повернутий ребром до радару, в якості тримача. Таким чином зайве випромінювання розсіювалось та не поверталось до радару.

Для мінімізації відбиття від землі радар та кутовий відбивач було встановлено на висоту 1.8 метрів. В такому випадку мінімальна довжина шляху променя від землі складало близько 6 метрів, а ці дані при калібруванні відсікаються.



Рис. 3.8. Тримач

3.3. Цифрова обробка в середовищі MATLAB

Компанія TI надає програмне забезпечення для цифрової обробки отриманих даних з радару в середовищі MATLAB. Приклад обробки наведено на рис. 3.9

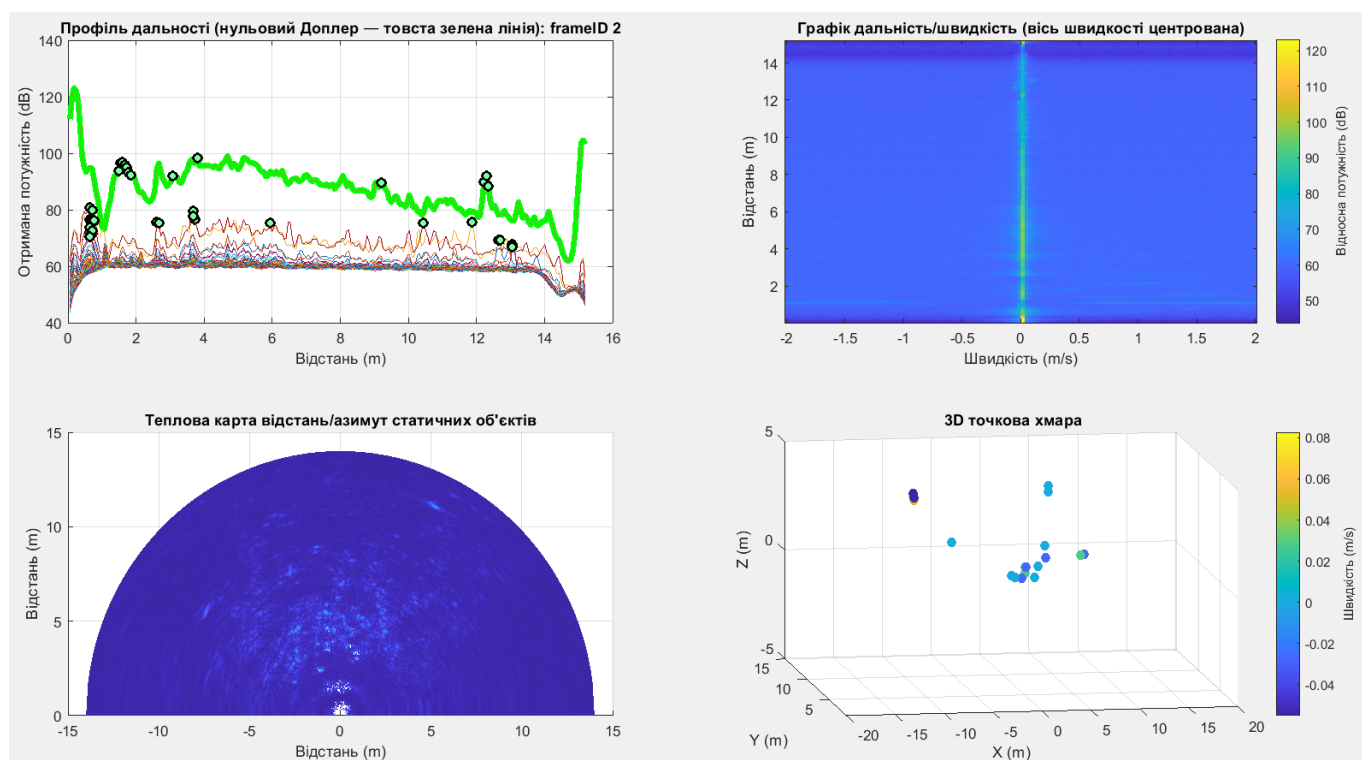


Рис. 3.9. Приклад обробки

На графіку у верхньому лівому куті наведено розподіл інтенсивності сигналу від дальності. Зелена крива позначає сумарну інтенсивність по всіх каналах для статичних цілей, криві різного кольору окремо для кожного приймального каналу.

На графіку у верхньому правому куті наведено теплову карту дальність/швидкість, логарифмічний масштаб якої, дозволяє з легкістю бачити на якій відстані є рухомі цілі та з якою швидкістю вони рухаються.

На графіку у нижньому лівому куті наведено теплову карту відстань/азимут, яка переведена в декартові координати для полегшення розуміння

фактичної позиції цілі. Може показувати як абсолютні значення, так і в показниковому масштабі ~ 0.4 .

На графіку у правому нижньому куті наведено 3-д візуалізацію детектованих цілей. Поріг детекції встановлюється алгоритмом CFAR (constant false alarm rate), параметри якого можуть бути змінені.

Програма також реалізує вагову функцію Ханна:

$$w_{Hann}(n) = 0.5 \left(1 - \cos \left(2\pi \frac{n}{N} \right) \right), 0 \leq n \leq N \quad (3.2)$$

Оскільки цілю роботи є дослідження формування зображення, надалі буде зображуватись лише теплова карта відстань/азимут.

Для роботи було збережено більшість оригінального коду від ТІ: парсинг отриманих даних в MATLAB, застосування калібрування, формування далекомірно-доплерівського портрету та інше. На основі цього було написано програму для оцінки якості калібрування та застосування алгоритму Річардсона-Люсі.

РОЗДІЛ 4.

РЕЗУЛЬТАТИ

4.1. Якість та вплив калібрування

Для оцінки калібрування підраховувались такі параметри: відношення енергії головної пелюстки до енергії найбільшої бічної – азимутальне PSLR (Peak Sidelobe Ratio), відношення енергії головної пелюстки до сумарної енергії всіх бічних пелюсток – азимутальне ISLR (Integrated Sidelobe Ratio), амплітудні A_{imb} та фазові φ_{imb} неузгодженості.

Азимутальне PSLR вираховувалось за формулою:

$$\text{PSLR} = 20 \cdot \log \left(\frac{\max |MLL|}{\max |SLL|} \right) \quad (4.1)$$

де MLL – інтенсивність головної пелюстки, SLL – інтенсивність бічної пелюстки.

азимутальне ISLR:

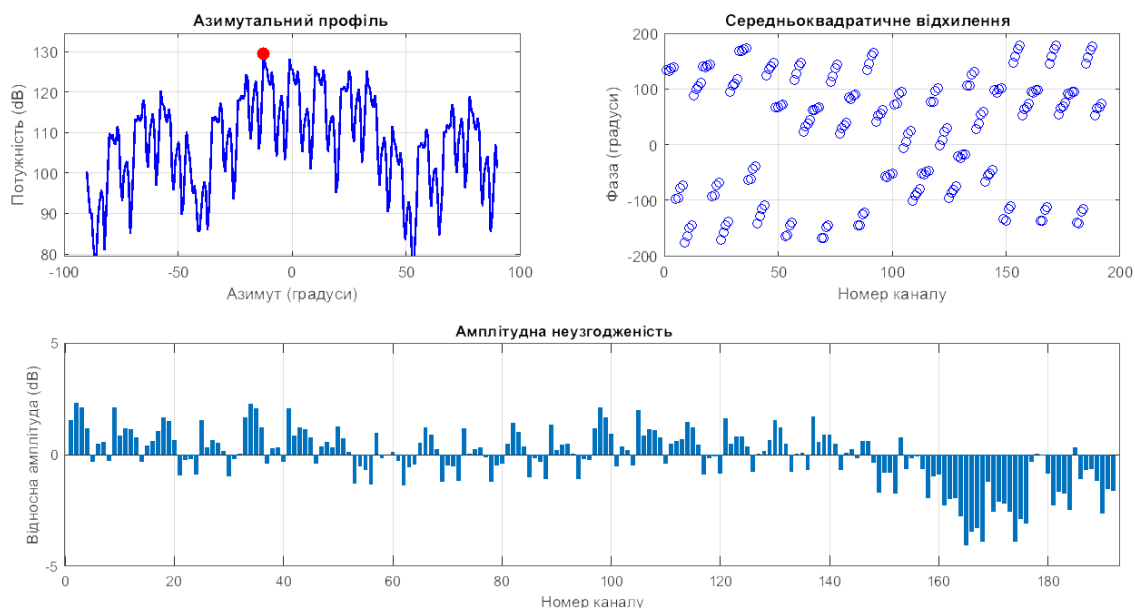
$$\text{ISLR} = 10 \cdot \left(\frac{\max |MLL|^2}{\sum |SLL|^2} \right) \quad (4.2)$$

амплітудна неузгодженість:

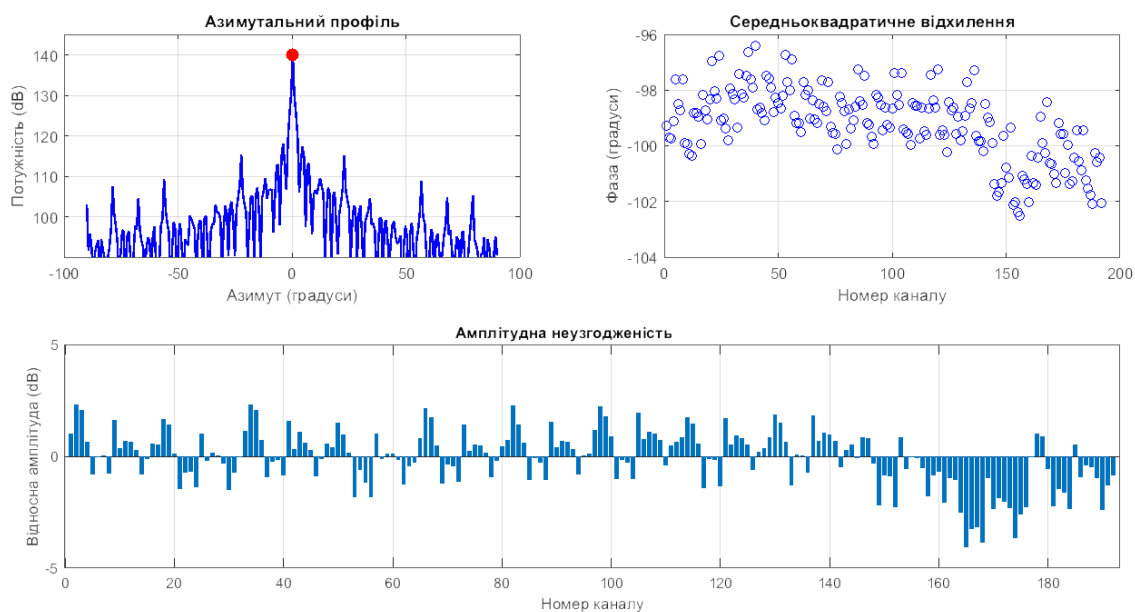
$$A_{imb} = 20 \cdot \log \left(\frac{\max |A|}{\min |A|} \right) \quad (4.3)$$

Фазова неузгодженість визначалась як середньоквадратичне відхилення значень фази у вибірці за номерами каналів.

Рис. 4.1 ілюструє відмінності в азимутальному профілі, розподілу фаз та відносних амплітуд по каналам для каліброваного та не каліброваного радару. Перший канал є референсним для відносних амплітуд.



(а) Без калібрування



(б) З калібруванням

Рис. 4.1. Розподіл інтенсивності від азимуту; розподіли фаз та відносних амплітуд від номеру каналу.

На рис.4.1 звертає на себе увагу поява регулярності азимуту та зменшення розкиду фазових неузгодженостей при калібруванні.

Калібрування потребує виставлення кутового відбивача напроти радару, що не завжди можливо зробити з високою точністю. Було проведено два ка-

лібрування з навмисно знехтуваними вимогами. Перше калібрування (K1) – кутовий відбивач встановлювався рівно навпроти центру радару, але був орієнтований під кутом 29.6° відносно вертикальної площини. Друге калібрування (K2) – правильно орієнтований кутовий відбивач відставлений в бік на 0.5 м відносно осі радару.

Таблиця 4.1

Показники PSLR, ISLR, амплітудної та фазових неузгодженостей за різних умов

	Без калібрування	Належне калібруванням	K1	K2
PSLR, dB	1.3	20.1	2.0	20.1
ISLR, dB	-10.6	10.9	-6.2	10.8
A_{imb} , dB	6.4	6.4	7.1	5.2
φ_{imb} , °	97.3	1.3	112.0	1.6

Як видно з табл. 4.1, калібрування зменшує бокові пелюстки по азимуту, а також фазові неузгодженості, підвищуючи точність детектування по куту. Проте калібрування не враховує амплітудну неузгодженість, а їх відмінності в табл. 4.1 радше мають випадковий характер.

Хоч на перший погляд може здатися що правильна методика калібрування майже не залежить від відхилення кутового відбивача від осі радару, проте таке відхилення спричиняє обертання теплової карти. Це ілюструє рис.4.2 :

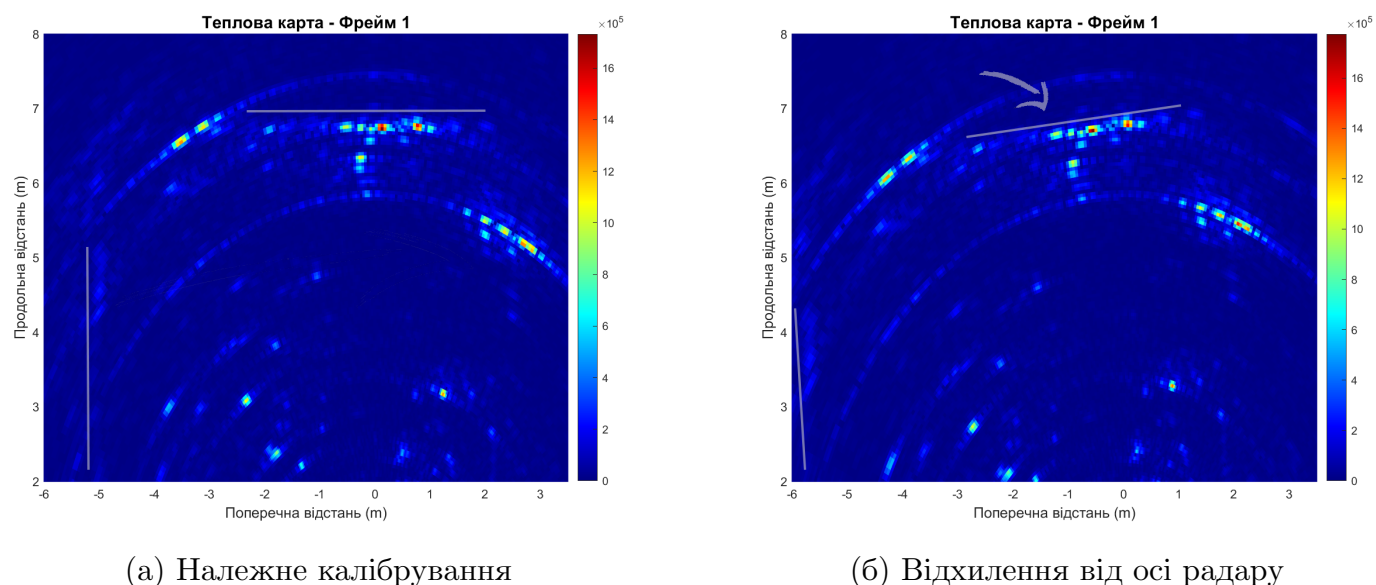


Рис. 4.2. Порівняння належного калібрування та калібрування K2

Збір даних рис.4.2 проводилось в закритому приміщенні. Зображення стіни, показане стрілкою, не відповідає дійсності.

4.2. Зображення на тепловій карті точкового об'єкт з високим ЕПР

Після проведення калібрування, наступною задачею було перевірити технічні можливості радару, а саме роздільні здатності по дальності та азимуту.

Виробник заявляє, що роздільна здатність по азимуту не залежить від конфігурації лінійно частотно модульованого (ЛЧМ) сигналу, і рівна 1.4° [25]. Роздільна здатність по дальності залежить від конфігурації ЛЧМ та визначається за формулою 2.8.

Але в радарі ширина смуги частот не задається безпосередньо див. рис. 3.4 .

Для визначенню роздільних здатностей було використано наступні характеристики:

Таблиця 4.2

Конфігурація ЛЧМ	
Початкова частота $f_{\text{поч}}$, ГГц	77
Швидкість наростання k , МГц/мкс	20
Час релаксації T_{idle} , мкс	5
Початок квантування $T_{\text{adc_st}}$, мкс	6
Кількість зразків N_{adc}	256
Частота квантування f_{samp} , МГц	8
Кінець наростання $T_{\text{ramp_end}}$, мкс	40

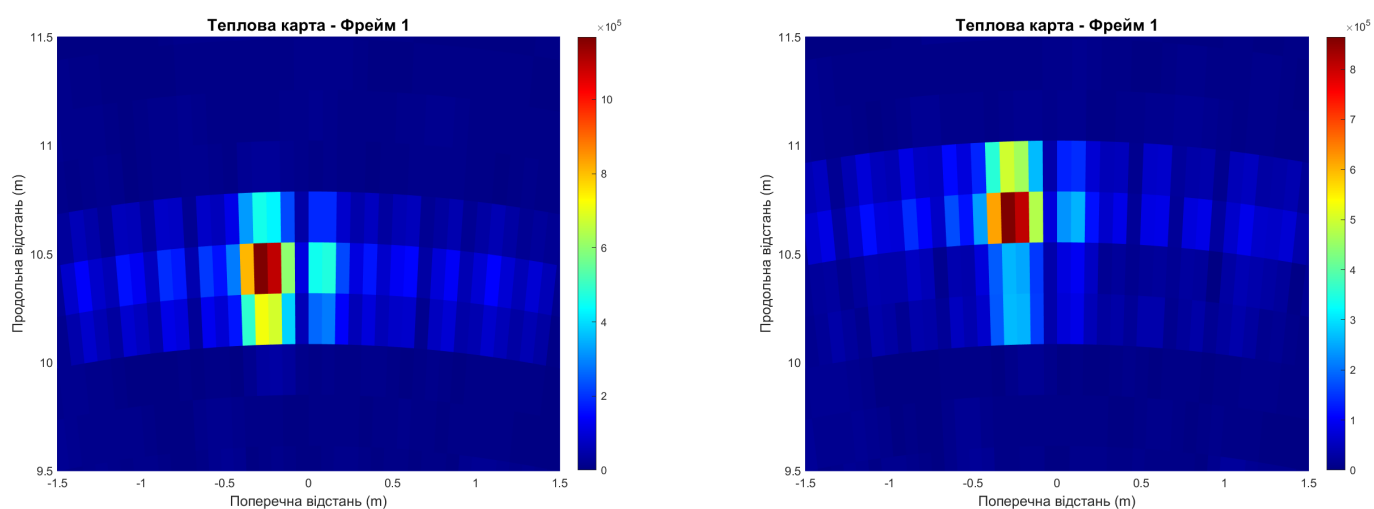
При такій конфігурації ширина смуги частот складає 640 МГц, відповідно теоретична роздільна здатність по дальності дорівнює 23.44 см.

З метою експериментального визначення азимутальної роздільної здатності на відстані 10 метрів від радару, на поверхні землі було встановлено два

кутових відбивачі з ефективною площею розсіювання (ЕПР) відповідно 1 та 8 м^2 , розміщених на відстані 25 см один від одного в напрямку, перпендикулярному до лінії зору.

Аналогічно для виміру роздільної здатності по дальності, кутові відбивачі встановлювались на тій же відстані, з таким же розділенням в 25 см, один за одним.

На рис.4.3 зображено теплову карту двох кутових відбивачів, в зазначених вище конфігураціях.



(а) кутові відбивачі розділені по азимуту

(б) кутові відбивачі розділені по дальності

Рис. 4.3. Теплова карта кутових відбивачів встановлених: а) одне біля одного, б) один за одним

Відстань 25 см на відстані 10 м еквівалентно кутовому розділенню в 1.43° . Отже, належне калібрування дозволяє в повній мірі використовувати можливості радару.

4.3. Функція розсіювання точки МІМО радару та алгоритм деконволюції Річардсона-Люсі

Як зазначалось в 1.3, зображення отримане від радару, можна представити у вигляді згортки істинного зображення та функції розсіювання точки

(ФРТ). Алгоритм Річардсона-Люсі дозволяє провести деконволюцію (зворотну згортку), відтворюючи істинне зображення та, як наслідок, покращити роздільну здатність системи. Була поставлена задача теоретично змодельовати ФРТ та застосувати за її допомогою алгоритм Річардсона-Люсі.

Функція розсіювання точки будується із урахуванням просторової конфігурації антен передавача (TX) і приймача (RX). В основі лежить побудова віртуальної антенної ґратки, що генерується шляхом векторного додавання координат передавальних і приймальних елементів:

$$\mathbf{r}_{ij} = \mathbf{r}_{tx,i} + \mathbf{r}_{rx,j} \quad (4.4)$$

де $\mathbf{r}_{tx,i}$ та $\mathbf{r}_{rx,j}$ – положення i -того передавача та j -того приймача, відповідно.

Для кожної точки простору вираховується фазовий зсув, що спричиняється віртуальними елементами, в обраному напрямку:

$$AF(\theta) = \sum_{n=1}^N \exp [jk(x_n \sin \theta + y_n \cos \theta)] \quad (4.5)$$

де $AF(\theta)$ – фазова діаграма (array factor), k – хвильове число, (x_n, y_n) – координати елементів ґратки.

Імпульсний відгук системи по дальності моделюється Фур'є перетворенням вагової функції, а саме функції Ханна. Це еквівалентно згортці в часовій області з вікном, яке має розширену головну пелюстку та ефективно пригнічені побічні пелюстки в частотній області. Дискретне перетворення Фур'є для функції Хана має вигляд:

$$\begin{aligned} \mathcal{F}\{w_{\text{Hann}}[n]\} = e^{-i\pi f N} & \left[\frac{1}{2} \cdot \frac{\sin(\pi(N+1)f)}{\sin(\pi f)} + \frac{1}{4} \cdot \frac{\sin(\pi(N+1)(f - \frac{1}{N}))}{\sin(\pi(f - \frac{1}{N}))} \right. \\ & \left. + \frac{1}{4} \cdot \frac{\sin(\pi(N+1)(f + \frac{1}{N}))}{\sin(\pi(f + \frac{1}{N}))} \right] \end{aligned} \quad (4.6a)$$

$$h(r) = |\mathcal{F}\{w_{\text{Hann}}[n]\}|^2 \quad (4.66)$$

де $h(r)$ – імпульсний відгук по відстані, N – кількість зразків.

Функція розсіювання точки є добутком відгуку по азимуту та відстані:

$$h(r, \theta) = |AF(\theta)|^2 \cdot h(r) \quad (4.7)$$

Для того щоб отримати істинне зображення Алгоритм реалізується за формулою 1.2. Оскільки проблема зворотної згортки є некоректно поставленою задачею (ill-posed) [27], тобто такою, розв’язок якої нелінійно залежить від девіації початкової умови. Тому необхідна регуляризація розв’язку. В даній роботі використовувався метод повної варіації (Total Variation).

Для зображення I , регуляризоване зображення I_{reg} обчислюється як:

$$I_{reg} = I + \lambda \cdot \Delta t \cdot \text{div} \left(\frac{\nabla I}{|\nabla I| + \varepsilon} \right) \quad (4.8)$$

де λ – коефіцієнт регуляризації, Δt – умовний крок часу, що задає стабільність і масштаб змін, ε – малий додаток для уникнення поділу на нуль, ∇ – оператор

Таким чином, реалізація дозволяє варіативно регулювати згладжування на основі обраного коефіцієнту регуляризації, зберігаючи баланс між точністю реконструкції та стабільністю у присутності шуму.

Для наочного порівняння правильності моделі алгоритм застосовувався як для змодельованої ФРТ (рівняння 4.8), так і для експериментально визначеної ФРТ, отриманої із зображення кутового відбивача (рис.4.4)

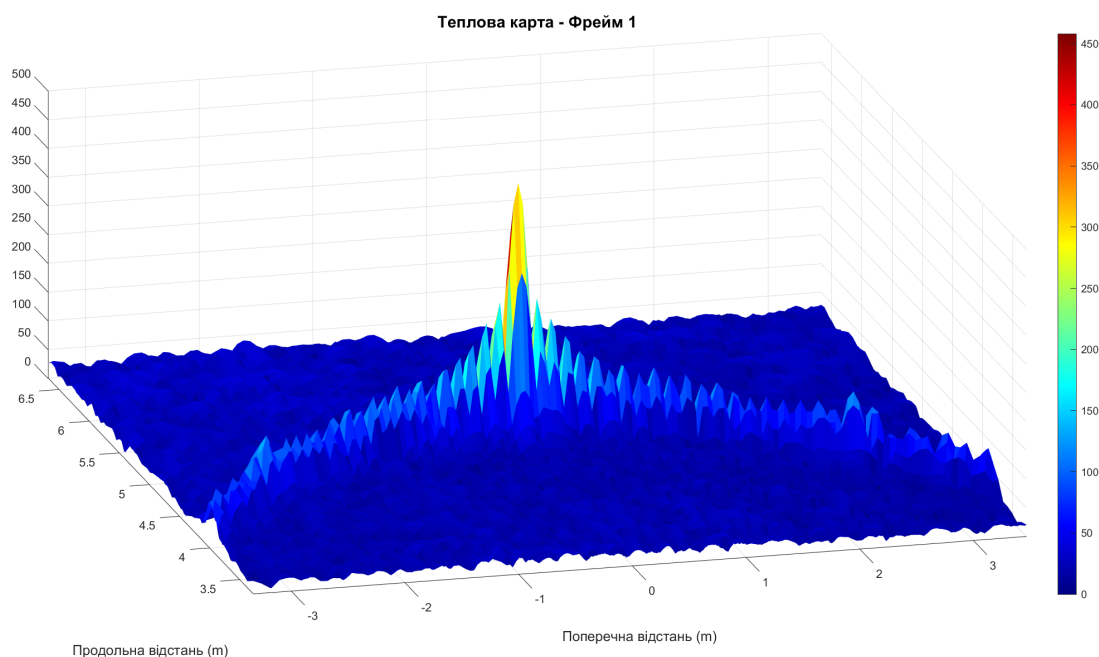


Рис. 4.4. ФРТ отримана від кутового відбивача. Масштаб показниковий

4.4. Модельні об'єкти

Було досліджено відображення об'єктів різної геометрії та з різною ЕПР. Розглянуто два випадки. В першому об'єкти знаходились в прямому полі зору на рівній поверхні. В другому об'єкти були замасковані, частина рельєфу була сильно викривленою.

Для обох випадків використовувались однакові параметри ЛЧМ:

Таблиця 4.3

Конфігурація ЛЧМ	
Початкова частота $f_{\text{поч}}$, ГГц	77
Швидкість наростання k , МГц/мкс	79
Час релаксації T_{idle} , мкс	5
Початок квантування $T_{\text{adc_st}}$, мкс	6
Кількість зразків N_{adc}	256
Частота квантування f_{samp} , МГц	8
Кінець наростання $T_{\text{ramp_end}}$, мкс	40

Така конфігурація забезпечувала поле зору в 15 м з роздільною здатністю 5 см.

4.4.1. Модельні об'єкти в прямому полі зору

Для першої серії експериментів було обрано три округлих тіла з ЕПР в декілька квадратних дециметрів, та 4 видовжених тіла типу стержня (дві лижні палки та два шампури) з ЕПР в декілька квадратних сантиметрів.

Для першого експерименту об'єкти розташовувались на відстані близько 5 метрів в конфігурації показаній на рис.4.5



Рис. 4.5. Локалізація об'єктів дослідження для експерименту №1

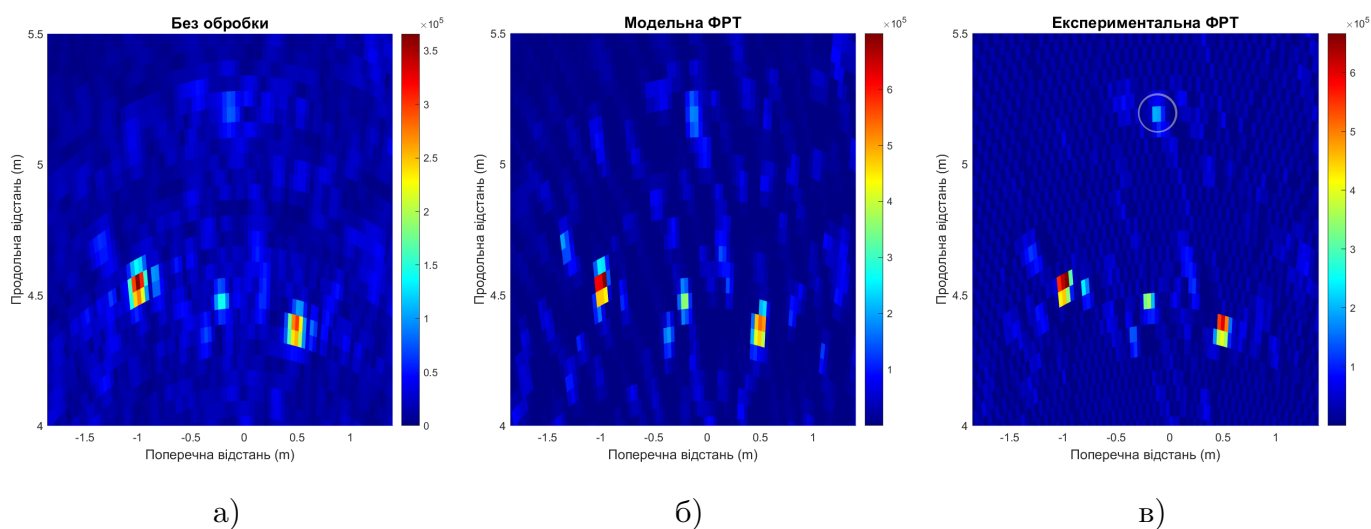


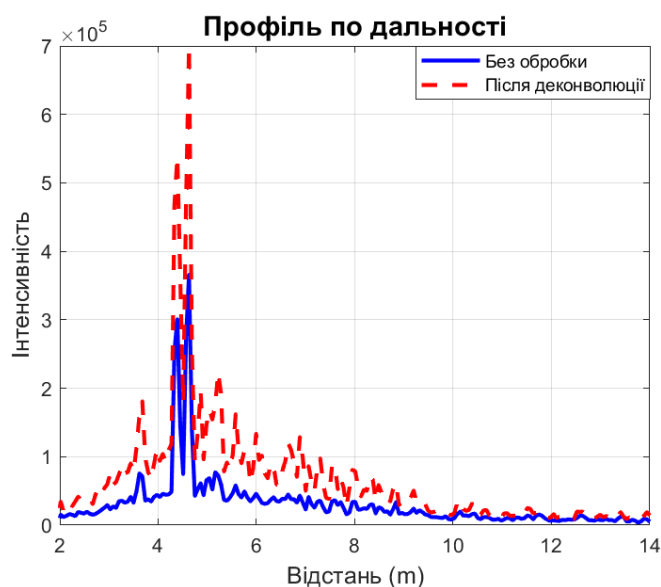
Рис. 4.6. Теплова карта, яка відповідає 4.5. а) Без обробки, б) Після обробки модельною ФРТ, в) Після обробки експериментальною ФРТ

На рис.4.6 в) колом позначені шампури, які стало можливо чітко детектувати після обробки експериментальною ФРТ.

Для застосування алгоритму коефіцієнт регуляризації становив $5 \cdot 10^4$, кількість ітерацій 125. Для чисельної оцінки якості алгоритму використовувались дві величини, а саме посилення піку PI (Peak Improvement) та загальне покращення AI (Average Improvement). Перша величина показує наскільки підсилилась інтенсивність головного піку, друга відношення середньої інтенсивності обробленого зображення до необробленого.

Для модельної ФРТ покращення склали: PI = 5.63 дБ, AI = 1.38 дБ. Для експериментальної: PI = 5.22 дБ, AI = 7.64 дБ.

Також, для більш повного розуміння перерозподілу інтенсивності, наводяться профіль по дальності та у поперечному напрямку на відстані головного піку рис. 4.7-4.8.

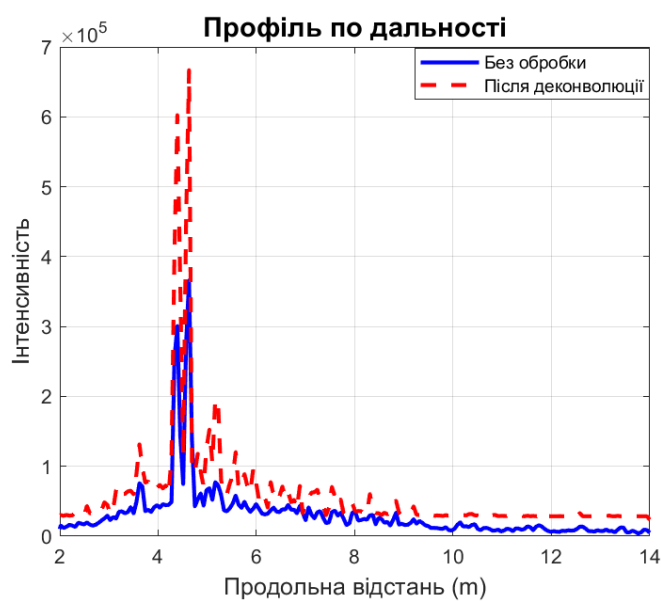


а) Профіль по дальності



б) Профіль у поперечному напрямку

Рис. 4.7. Розподіл інтенсивності для модельної ФРТ



а) Профіль по дальності



б) Профіль у поперечному напрямку

Рис. 4.8. Розподіл інтенсивності для експериментальної ФРТ

Для другого експерименту брались ті самі об'єкти, що були розташовані на відстані 10 метрів, та зміщені відносно осі радару на два метри рис.4.9. Оскільки цілі погано видно, для контрастності були додані стрілки.



Рис. 4.9. Локалізація об'єктів дослідження для експерименту №2

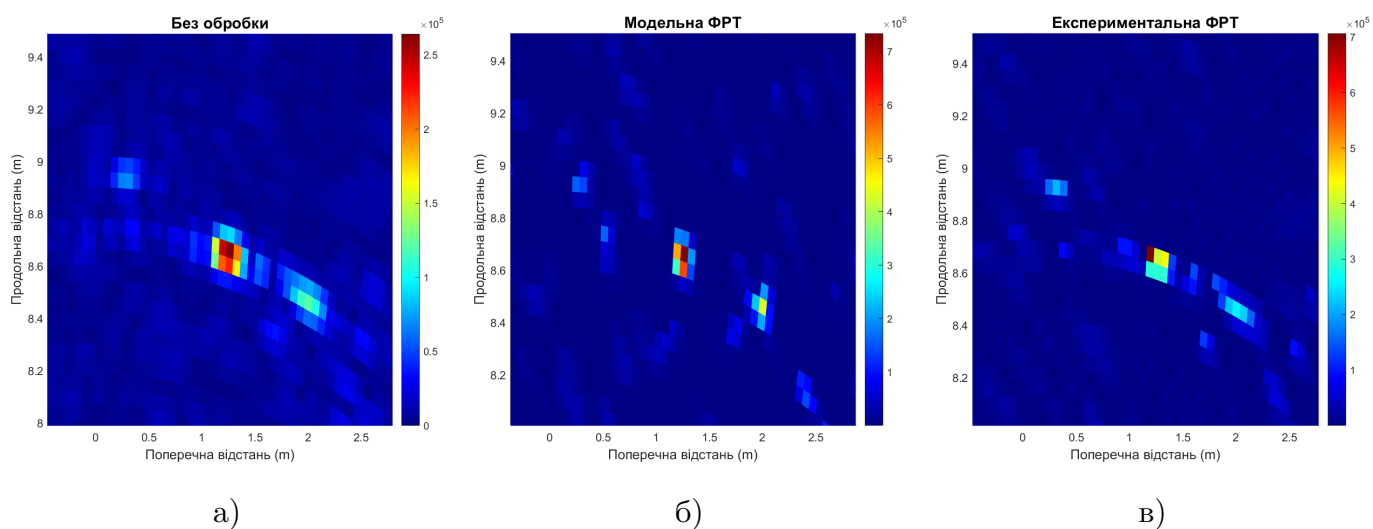
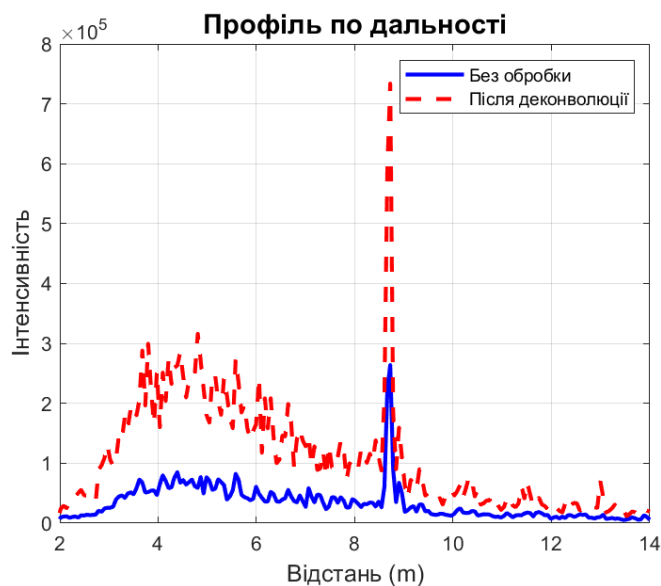


Рис. 4.10. Теплова карта, яка відповідає 4.9. а) Без обробки, б) Після обробки модельною ФРТ, в) Після обробки експериментальною ФРТ

Для застосування алгоритму коефіцієнт регуляризації становив $1 \cdot 10^4$, кількість ітерацій 150. Для модельної ФРТ покращення склали: $PI = 8.88$ дБ, $AI = 0.14$ дБ. Для експериментальної: $PI = 8.55$ дБ, $AI = 0.77$ дБ. На рис.4.10а) спостерігаються азимутальні дифузні тяжі, що зникають (або суттєво послаблюються) після застосування алгоритму Річардсона-Люсі.

Далі, на рис. 4.11-4.12 наводяться перерозподіли інтенсивності, а саме: профіль по дальності та у поперечному напрямку на відстані головного піку.

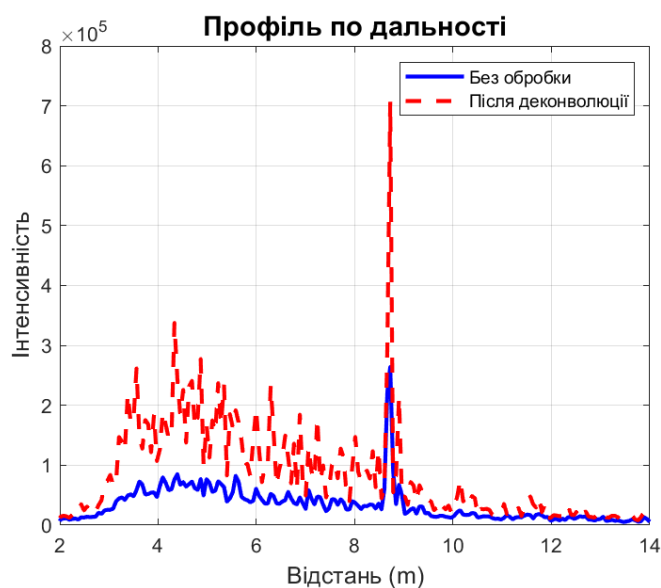


а) Профіль по дальності



б) Профіль у поперечному напрямку

Рис. 4.11. Розподіл інтенсивності для модельної ФРТ



а) Профіль по дальності



б) Профіль у поперечному напрямку

Рис. 4.12. Розподіл інтенсивності для експериментальної ФРТ

4.4.2. Зображення об'єктів замасковані завадами

Для моделювання ситуації замаскованих об'єктів, була обрана галявина засаджена деревами та викривленим рельєфом. Об'єкти розміщувались в траві, деякі були прикопані.

Об'єкти обрані для третього експерименту та їх розташування показані на рис. 4.13.

Кутові відбивачі мали ЕПР 1,2 та 8 м², інші об'єкти мали значно менший ЕПР. Об'єкти округлої форми, плита, бронежилет, лопата мали ЕПР в декілька дм². Один снаряд мав ЕПР в декілька см², інший до 1 м².



Рис. 4.13. Локалізація об'єктів дослідження для експерименту №3.
Номера 1, 5, 6 – кутові відбивачі; 2, 3, 8 – об'єкти округлої форми; 4 – пластина; 7, 9 – снаряди;
10 – бронежилет; 11 – лопата

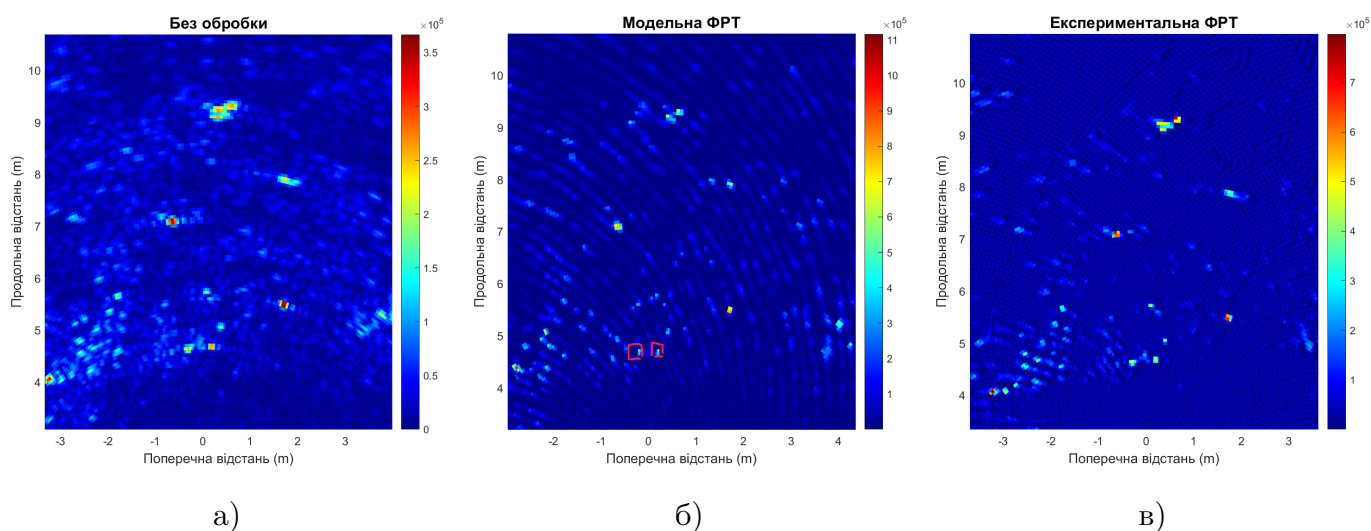
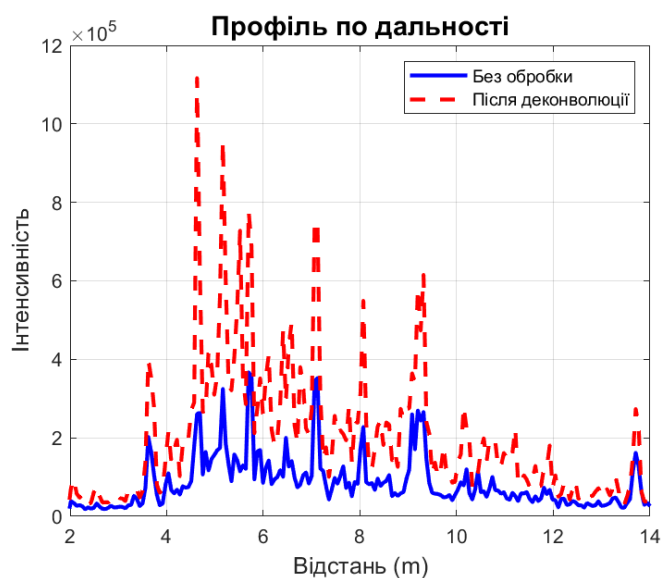


Рис. 4.14. Теплова карта, яка відповідає 4.5. а) Без обробки, б) Після обробки модельною ФРТ, в) Після обробки експериментальною ФРТ

Для застосування алгоритму коефіцієнт регуляризації становив $9 \cdot 10^4$, кількість ітерацій 125. На рис. 4.14 б) маркером позначено піки, від кутового відбивача та пластини, але вони надто тонкі, через що їх не видно на рисунку.

Для модельної ФРТ покращення склали: $PI = 9.67$ дБ, $AI = 0.7$ дБ. Для експериментальної: $PI = 6.75$ дБ, $AI = 5.55$ дБ. Як бачимо застосування алгоритму дозволяє чітко виділити відбитки досліджуваних об'єктів, проте однозначна ідентифікація залишається складною задачею.

Далі, на рис. 4.15-4.16 наводяться перерозподіли інтенсивності, а саме: профіль по дальності та у поперечному напрямку на відстані головного піку.

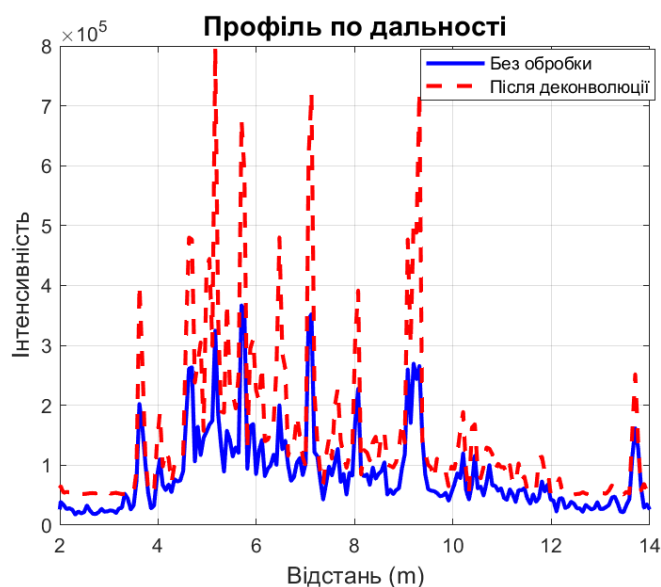


(а) Профіль по дальності



(б) Профіль у поперечному напрямку

Рис. 4.15. Розподіл інтенсивності для модельної ФРТ



а) Профіль по дальності



б) Профіль у поперечному напрямку

Рис. 4.16. Розподіл інтенсивності для експериментальної ФРТ

4.5. Узагальнення результатів дослідів модельних об'єктів

З рис.4.6, рис.4.10 та рис.4.14 видно, що загалом, після обробки алгоритмом, зберігаються піки, а також контури модельних об'єктів. Це додатково підтверджується рис. 4.7-4.8, 4.11-4.12, 4.15-4.16, де розподіл інтенсивності

після обробки наслідуює розподіл необроблених даних. На рис. 4.11-4.12 видно посилення перешкод на відстані до 8 метрів, але не зважаючи на це, біля розташувань цілей контури графіку необроблених даних наслідуються.

Загалом експериментальна ФРТ дає кращі результати, зокрема на рис. 4.6 видно, що експериментальна ФРТ дає можливість більш чітко виокремити об'єкти лінійної геометрії. Також, для всіх експериментів значення АІ є більшим, при меншому РІ, що вказує на кращий перерозподіл інтенсивностей, як наслідок більша контрастність зображення та зменшення впливу завад.

Код головної програми, а також для застосування алгоритму Річардсона-Люсі та оцінки якості калібрування наведено в додатках А-В.

ВИСНОВКИ

Результатами роботи є:

1) Огляд сучасного стану проблеми формування зображень за допомогою МІМО-радарів, переваг МІМО-архітектури, методів підвищення просторової та радіолокаційної роздільної здатності показав, що питання формування зображень у зашумленому середовищі систематично не досліджувалось.

2) Методом електронно-променевого зварювання виготовлено високоякісні кутові відбивачі з ЕПР 1 та 8 м² для радіохвиль з частотою 77 ГГц. Максимальне відхилення від перпендикулярності сторін відбивача становило 0.8°. Розроблено і виготовлено тримач для кутового відбивача, який має нехтовно малий ЕПР.

3) За спеціально розробленою методикою проведено калібрування радарної системи без використання безлунної камери. Проаналізовано вплив відхилень від стандартної методики калібрування на точність вимірювань.

4) Встановлено, що експериментально визначена роздільна здатність за дальністю та азимутом відповідає заявленою виробником, за умови належно зробленої калібровки.

5) За допомогою точкового центру розсіяння (кутового відбивача з ЕПР 1 м²) отримано експериментальну функцію розсіювання точки для досліджуваної радарної системи.

6) Розроблено програму реалізації алгоритму Річардсона—Люсі для де-конволюції зображення радару. Алгоритм застосовано як для змодельованої так і експериментально визначеної функції розсіювання точки. Показано, що застосування алгоритму Річардсона—Люсі до радарних зображень дозволяє суттєво покращити видимість центрів розсіяння замаскованих сторонніми завадами, хоча їх ідентифікація є неоднозначною.

ПЕРЕЛІК ПОСИЛАНЬ

1. Grove R. MIMO radar Systems and Algorithms - Imperfections and Calibration : дис. ... док. / Grove Ricard Llado. — 2022.
2. Skolnik M. Radar Handbook, Third Edition. — McGraw-Hill Education, 2008. — (Electronics electrical engineering). — ISBN 9780071485470.
3. Edde B. RADAR principles, technology, application. — Englewood Cliffs, N.J. : Prentice Hall, 1993. — ISBN 0-13-752346-7.
4. Van Trees H. L. Optimum array processing: Part IV of detection, estimation, and modulation theory. — Nashville, TN : John Wiley & Sons, 2002.
5. MIMO radar theory and experimental results / F. Robey [та ін.] // Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004. T. 1. — 2004. — 300—304 Vol.1. — DOI: [10.1109/ACSSC.2004.1399141](https://doi.org/10.1109/ACSSC.2004.1399141).
6. MIMO radar: an idea whose time has come / E. Fishler [та ін.] // Proceedings of the 2004 IEEE Radar Conference (IEEE Cat. No.04CH37509). — 2004. — C. 71—78. — DOI: [10.1109/NRC.2004.1316398](https://doi.org/10.1109/NRC.2004.1316398).
7. Li J., Stoica P. MIMO Radar Signal Processing. — John Wiley & Sons, Inc., 2008. — ISBN 9780470178980.
8. Tse D., Viswanath P. Fundamentals of Wireless Communication //. — Cambridge University Press, 2005. — C. 1—9.
9. Syeda R. Z., Beurden M. C. van, Smolders A. B. [Sparse virtual array synthesis for MIMO radar imaging systems](https://doi.org/10.1049/mia2.12129) // IET Microwaves, Antennas & Propagation. — 2021. — T. 15, № 11. — C. 1458—1472. — DOI: <https://doi.org/10.1049/mia2.12129>. — eprint: <https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/mia2.12129>.

10. Wahlgren S. [Sidelobe blanking in the presence of noise-like interference](#) : Master's thesis / Wahlgren Stina. — Gothenburg, Sweden : Chalmers University of Technology, 2018.
11. Tait P. [Introduction to Radar Target Recognition](#). — The Institution of Engineering, Technology, 2005. — DOI: [10.1049/PBRA018E](#). — eprint: [http://digital-library.theiet.org/doi/pdf/10.1049/PBRA018E](#).
12. [A sidelobe suppression algorithm for 77 GHz MIMO radars](#) / L. Yang [та ил.] // Measurement. — 2022. — Т. 190. — С. 110691. — ISSN 0263-2241. — DOI: [https://doi.org/10.1016/j.measurement.2021.110691](#).
13. 2-D Coherence Factor for Sidelobe and GhostSuppressions in Radar Imaging / S. Li [та ил.] // IEEE Transactions on Antennas and Propagation. — 2020. — Т. 68, № 2. — С. 1204—1209. — DOI: [10.1109/TAP.2019.2938581](#).
14. Pal P., Vaidyanathan P. P. Frequency invariant MVDR beamforming without filters and implementation using MIMO radar // 2009 IEEE International Conference on Acoustics, Speech and Signal Processing. — IEEE. 2009. — С. 2089—2092. — DOI: [10.1109/ICASSP.2009.4959912](#).
15. Combating the Dimensionality of Nonlinear MIMO Amplifier Predistortion by Basis Pursuit / E. Zenteno [та ил.] // Proceedings of the 44th European Microwave Conference (EuMC). — EuMA, 2014. — С. 833—836.
16. Yanik M. E., Wang D., Torlak M. 3-D MIMO-SAR Imaging Using Multi-Chip Cascaded Millimeter-Wave Sensors // 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP). — 2019. — С. 1—5. — DOI: [10.1109/GlobalSIP45357.2019.8969133](#).
17. Rojhani N., Shaker G. [Comprehensive Review: Effectiveness of MIMO and Beamforming Technologies in Detecting Low RCS UAVs](#) // Remote Sensing. — 2024. — Т. 16, № 6. — ISSN 2072-4292. — DOI: [10.3390/rs16061016](#).

18. DoA Estimation for Micro and Nano UAV Targets using AWR2243 Cascaded Imaging Radar / T. Kavya [та ін.] // . — 12.2022. — С. 528—531. — DOI: [10.1109/iSES54909.2022.00115](https://doi.org/10.1109/iSES54909.2022.00115).
19. Bliss D., Forsythe K. Multiple-input multiple-output (MIMO) radar and imaging: degrees of freedom and resolution // The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. Т. 1. — 2003. — 54—59 Vol.1. — DOI: [10.1109/ACSSC.2003.1291865](https://doi.org/10.1109/ACSSC.2003.1291865).
20. A Compact Single-Board Solution for Commercializing Cost-Effective 77 GHz Automotive Front Radar / W. Zhang [та ін.] // 2020 IEEE Asia-Pacific Microwave Conference (APMC). — 2020. — С. 1098—1100. — DOI: [10.1109/APMC47863.2020.9331712](https://doi.org/10.1109/APMC47863.2020.9331712).
21. Texas Instruments. [mmWave Sensing: FMCW Radar](#) : tex. збіт. / Texas Instruments. — n.d.
22. Levanon N., Mozeson E. [Radar Signals](#). — Wiley, 2004. — (IEEE Press). — ISBN 9780471663072.
23. Davis M. S. MIMO Radar Signal Processing, Waveform Design, and Application to Synthetic Aperture Imaging : Ph.D. Thesis / Davis Michael S. — 2015.
24. Texas Instruments. [Overview of Cascade Radar System](#) : tex. збіт. / Texas Instruments. — 2018. — SWRA554A.
25. Texas Instruments. [Cascade Radar Interface: User's Guide](#). — 10.2018. — Document Number: TIDUEN5A.
26. Texas Instruments. [Signal Processing User's Guide: Cascade Radar with mmWave Sensors](#). — 2020.
27. Bertero M., Boccacci P., Mol C. D. [Introduction to Inverse Problems in Imaging](#). — 2-е вид. — CRC Press, 2021. — DOI: [10.1201/9781003032755](https://doi.org/10.1201/9781003032755).

ДОДАТКИ

Додаток А. Код файлу головної програми

Лістинг 1: Код файлу програми cascade_MIMO_signalProcessing.m

```

1  %% cascade_MIMO_signalProcessing.m
2
3  clearvars; close all; clc;
4
5  %% ===== PROCESSING OPTIONS =====
6  DECONV_ON = 1; % Enable Richardson-Lucy deconvolution
7  PLOT_COMPARISON = 1; % Plot original and processed side-by-side
8  SAVE_RESULTS = 0; % Save results to .mat file
9  CALIB_ANALYSE = 0;
10
11 %% ===== PSF OPTIONS =====
12 USE_EXPERIMENTAL_PSF = 1; % 1: use measured PSF, 0: use model
13 PSF_FRAME_IDX = 1; % Frame index to use for PSF extraction
14
15 %% ===== DISPLAY OPTIONS =====
16 LOG_SCALE = 0; % Use logarithmic scale for display ( $x^{0.4}$ )
17 USE_STATIC = 1; % Process static (zero-Doppler) or dynamic targets
18 COLORMAP_TYPE = 'jet'; % Options: 'jet', 'hot', 'turbo', 'viridis'
19
20 %% ===== FRAME SELECTION =====
21 frameIdx = 2; % Frame to process
22 numFramesToAverage = 1; % Number of frames to average (1 = no averaging)
23
24 %% ===== PROCESSING PARAMETERS =====
25 % Range processing
26 minRangeBinKeep = 5; % Minimum range bin to keep
27 rightRangeBinDiscard = 20; % Range bins to discard from right
28 minDistanceThreshold = 2.0; % Minimum distance threshold in meters
29
30 % Deconvolution parameters
31 deconvIterations = 150; % Richardson-Lucy iterations (reduced from 30)
32 deconvRegularization = 1e4; % Regularization parameter (increased from 0.02)
33
34 %% Antenna Configuration (TI Cascade)
35 % Define antenna positions in wavelengths
36 rx_pos = [0, 0; 0.5, 0; 1, 0; 1.5, 0; 5.5, 0; 6, 0; 6.5, 0; 7, 0; ...
37           23, 0; 23.5, 0; 24, 0; 24.5, 0; 25, 0; 25.5, 0; 26, 0; 26.5, 0];
38
39 tx_pos = [7, -17; 9, -17; 11, -17; 11.5, -16.5; 12, -15; 12.5, -14; 13, -17; ...
40           15, -17; 17, -17; 19, -17; 21, -17; 23, -17];
41
42 % Calculate wavelength
43 c = 3e8;
44 center_freq = 77e9; % 77 GHz for TI cascade
45 wavelength = c / center_freq;
46
47 % Create antenna configuration structure
48 antenna_config = struct();
49 antenna_config.rx_pos = rx_pos;
50 antenna_config.tx_pos = tx_pos;
51 antenna_config.wavelength = wavelength;
52 antenna_config.num_rx = size(rx_pos, 1);
53 antenna_config.num_tx = size(tx_pos, 1);
54
55 fprintf('Antenna Configuration:\n');
56 fprintf('Number of RX: %d\n', antenna_config.num_rx);
57 fprintf('Number of TX: %d\n', antenna_config.num_tx);

```

```

58 fprintf('VirtualArraySize:%d\n', antenna_config.num_rx * antenna_config.num_tx);
59 fprintf('Wavelength:%.3fmm\n', wavelength * 1000);
60
61
62 %% Initialize paths
63 pro_path = getenv('CASCADE_SIGNAL_PROCESSING_CHAIN_MIMO');
64 input_path = strcat(pro_path, '\main\cascade\input\');
65
66 % Load test configuration
67 testList = strcat(input_path, 'testList.txt');
68 fidList = fopen(testList, 'r');
69
70 % Get test paths
71 dataFolder_test = fgetl(fidList);
72 dataFolder_calib = fgetl(fidList);
73 module_param_file = fgetl(fidList);
74 PSF_DATA_PATH = fgetl(fidList);
75 fclose(fidList);
76
77 % Generate data parameter file
78 pathGenParaFile = [input_path, 'test1_param.m'];
79 clear(pathGenParaFile);
80 parameter_file_gen_json(dataFolder_test, dataFolder_calib, module_param_file, pathGenParaFile, 'TDA2');
81
82 %% Load calibration data
83 load(dataFolder_calib);
84
85 %% Initialize processing modules
86 fprintf('Initializing processing modules...\n');
87 simTopObj = simTopCascade('pfile', pathGenParaFile);
88 calibrationObj = calibrationCascade('pfile', pathGenParaFile, 'calibrationfilePath', dataFolder_calib);
89 rangeFFTObj = rangeProcCascade('pfile', pathGenParaFile);
90 DopplerFFTObj = DopplerProcClutterRemove('pfile', pathGenParaFile);
91 detectionObj = CFAR_CASO('pfile', pathGenParaFile);
92 DOAObj = DOACascade('pfile', pathGenParaFile);
93
94 %% Get data files
95 [fileIdx_unique] = getUniqueFileIdx(dataFolder_test);
96 [fileNameStruct] = getBinFileNames_withIdx(dataFolder_test, fileIdx_unique{1});
97 calibrationObj.binfilePath = fileNameStruct;
98
99 %% Process radar data
100 fprintf('Processing frame%d...\n', frameIdx);
101
102 % Initialize accumulator for frame averaging
103 if numFramesToAverage > 1
104     accumulator = [];
105 end
106
107 for avgIdx = 0:(numFramesToAverage-1)
108     currentFrame = frameIdx + avgIdx;
109
110     % Read and calibrate ADC data
111     calibrationObj.frameIdx = currentFrame;
112     adcData = datapath(calibrationObj);
113
114     % RX channel re-ordering
115     adcData = adcData(:, :, calibrationObj.RxForMIMOProcess, :);
116
117     % Perform 2D FFT processing
118     rangeFFTOut = [];
119     DopplerFFTOut = [];
120
121     for i_tx = 1:size(adcData, 4)
122         rangeFFTOut(:, :, i_tx) = datapath(rangeFFTObj, adcData(:, :, :, i_tx));
123         DopplerFFTOut(:, :, i_tx) = datapath(DopplerFFTObj, rangeFFTOut(:, :, i_tx));
124     end
125
126 % Reshape for MIMO processing

```

```

127 DopplerFFTOut = reshape(DopplerFFTOut, size(DopplerFFTOut,1), size(DopplerFFTOut,2), ...
128                         size(DopplerFFTOut,3)*size(DopplerFFTOut,4));
129
130 % Accumulate for averaging
131 if numFramesToAverage > 1
132     if isempty(accumulator)
133         accumulator = DopplerFFTOut;
134     else
135         accumulator = accumulator + DopplerFFTOut;
136     end
137 end
138 end
139
140 % Average if needed
141 if numFramesToAverage > 1
142     DopplerFFTOut = accumulator / numFramesToAverage;
143     fprintf('Averaged %d frames\n', numFramesToAverage);
144 end
145
146 if CALIB_ANALYSE
147     %% Assess Calibration Quality
148     fprintf('\nAssessing calibration quality...\n');
149
150     % Create proper range axis vector
151     range_axis = (0:size(DopplerFFTOut,1)-1) * detectionObj.rangeBinSize;
152
153     % Assess calibration using the processed data
154     quality_metrics = assess_calibration_quality(DopplerFFTOut, antenna_config, ...
155         range_axis, ...
156         'PlotResults', true, ...
157         'TargetRangeWindow', [4, 6]); % Adjust range window based on your calibration target
158
159     % Store metrics for later use
160     calibration_quality = quality_metrics;
161 end
162
163 %% Generate range-azimuth heatmap
164 fprintf('Generating range-azimuth heatmap...\n');
165
166 % Get antenna positions for azimuth processing
167 antenna_azimuthonly = detectionObj.azimuthonly;
168
169 % Generate heatmap
170 [mag_data_static, mag_data_dynamic, y_axis, x_axis] = generate_range_azimuth_heatmap(...
171     DopplerFFTOut, detectionObj.rangeBinSize, ...
172     length(calibrationObj.IdTxForMIMOProcess), length(calibrationObj.RxForMIMOProcess), ...
173     antenna_azimuthonly, minRangeBinKeep, rightRangeBinDiscard, minDistanceThreshold);
174
175 % Select static or dynamic data
176 if USE_STATIC
177     mag_data = mag_data_static;
178     data_type = 'Static';
179 else
180     mag_data = mag_data_dynamic;
181     data_type = 'Dynamic';
182 end
183
184 %% Apply selected processing method
185 if DECONV_ON
186     fprintf('Applying Richardson-Lucy deconvolution...\n');
187     processed_data = apply_richardson_lucy_deconv(x_axis, y_axis, mag_data, ...
188         deconvIterations, deconvRegularization);
189     method_name = 'Richardson-Lucy Deconvolution';
190 else
191     processed_data = mag_data;
192     method_name = 'No Processing';
193 end
194
195 %% Visualization

```

```

196 fprintf('Creating visualization...\n');
197
198 % Set colormap
199 switch COLORMAP_TYPE
200     case 'viridis'
201         cmap = viridis(256);
202     case 'turbo'
203         cmap = turbo(256);
204     case 'hot'
205         cmap = hot(256);
206     otherwise
207         cmap = jet(256);
208 end
209
210 if PLOT_COMPARISON && (DECONV_ON)
211     % Side-by-side comparison
212     figure('Position', [100, 100, 1400, 600], 'Name', 'MMO_RadarProcessingComparison');
213
214     % Original heatmap
215     subplot(1,2,1);
216     if LOG_SCALE
217         surf(y_axis, x_axis, mag_data.^0.4, 'EdgeColor', 'none');
218     else
219         surf(y_axis, x_axis, mag_data, 'EdgeColor', 'none');
220     end
221     view(2);
222     xlabel('Поперечна відстань (м)', 'FontSize', 12);
223     ylabel('Продольна відстань (м)', 'FontSize', 12);
224     title(sprintf('Безобробки %s', data_type), 'FontSize', 14);
225     colormap(gca, cmap);
226     colorbar;
227     grid on;
228     xlim([-1.85 1.4]);
229     ylim([4 5.5]);
230
231     % Processed heatmap
232     subplot(1,2,2);
233     if LOG_SCALE
234         surf(y_axis, x_axis, processed_data.^0.4, 'EdgeColor', 'none');
235     else
236         surf(y_axis, x_axis, processed_data, 'EdgeColor', 'none');
237     end
238     view(2);
239     xlabel('Поперечна відстань (м)', 'FontSize', 12);
240     ylabel('Продольна відстань (м)', 'FontSize', 12);
241     title(sprintf('%s Результат', method_name), 'FontSize', 14);
242     colormap(gca, cmap);
243     colorbar;
244     grid on;
245     xlim([-1.85 1.4]);
246     ylim([4 5.5]);
247
248     % Add super title
249     sgtitle(sprintf('Frame %d-%s', frameIdx, datestr(now)), 'FontSize', 16);
250 else
251     % Single plot
252     figure('Position', [100, 100, 800, 600], 'Name', 'Теплова карта MMO-радару');
253     if LOG_SCALE
254         surf(y_axis, x_axis, processed_data.^0.4, 'EdgeColor', 'none');
255     else
256         surf(y_axis, x_axis, processed_data, 'EdgeColor', 'none');
257     end
258     view(2);
259     xlabel('Поперечна відстань (м)', 'FontSize', 12);
260     ylabel('Продольна відстань (м)', 'FontSize', 12);
261     title(sprintf('%s %s Теплова карта-Фрейм %d', data_type, method_name, frameIdx), 'FontSize', 14);
262     colormap(cmap);
263     colorbar;
264     grid on;

```

```

265     xlim([-1.85 1.4]);
266     ylim([4 5.5]);
267 end
268
269 %% Additional analysis plots
270 figure('Position', [100, 100, 1200, 400], 'Name', 'Profile_Analysis');
271
272 % Range profile comparison
273 subplot(1,2,1);
274 range_profile_orig = max(mag_data, [], 1);
275 range_profile_proc = max(processed_data, [], 1);
276 plot(y_axis(1,:), range_profile_orig, 'b-', 'LineWidth', 2, 'DisplayName', 'Без_обробки'); hold on;
277 plot(y_axis(1,:), range_profile_proc, 'r--', 'LineWidth', 2, 'DisplayName', 'Після_деконволюції');
278 xlabel('Відстань_м', 'FontSize', 12);
279 ylabel('Інтенсивність', 'FontSize', 12);
280 title('Профіль_по_дальності', 'FontSize', 14);
281 legend('Location', 'northeast');
282 grid on;
283 xlim([minDistanceThreshold, max(y_axis(1,:))]);
284
285 % Cross-range profile comparison
286 subplot(1,2,2);
287 [~, peak_range_idx] = max(range_profile_orig);
288 cross_range_orig = mag_data(:, peak_range_idx);
289 cross_range_proc = processed_data(:, peak_range_idx);
290 plot(y_axis(:,1), cross_range_orig, 'b-', 'LineWidth', 2, 'DisplayName', 'Без_обробки'); hold on;
291 plot(y_axis(:,1), cross_range_proc, 'r--', 'LineWidth', 2, 'DisplayName', 'Після_деконволюції');
292 xlabel('Поперечна_відстань_м', 'FontSize', 12);
293 ylabel('Інтенсивність', 'FontSize', 12);
294 title(sprintf('Профіль_у_поперечному_напрямку_на_відстані_%.1f_м', y_axis(1, peak_range_idx)), 'FontSize', 14);
295 legend('Location', 'northeast');
296 grid on;
297
298 %% Performance metrics
299 fprintf('\n=====Performance_Metrics=====\n');
300 fprintf('Peak_improvement: %.2f_dB\n', 20*log10(max(processed_data(:))/max(mag_data(:))));
301 fprintf('Average_improvement: %.2f_dB\n', 20*log10(mean(processed_data(:))/mean(mag_data(:))));
302
303 non_zero_mask = mag_data > 0;
304 if any(non_zero_mask(:))
305     avg_orig = mean(mag_data(non_zero_mask));
306     avg_proc = mean(processed_data(non_zero_mask));
307     avg_improvement = 20*log10(avg_proc/avg_orig);
308     fprintf('Average_improvement_(non-zero): %.2f_dB\n', avg_improvement);
309 end
310
311 % Calculate meaningful dynamic range using percentiles
312 non_zero_orig = mag_data(mag_data > 0);
313 non_zero_proc = processed_data(processed_data > 0);
314
315 if ~isempty(non_zero_orig)
316     noise_floor_orig = prctile(non_zero_orig(:), 10); % 10th percentile as noise estimate
317     noise_floor_proc = prctile(non_zero_proc(:), 10);
318     dr_orig = 20*log10(max(mag_data(:))/noise_floor_orig);
319     dr_proc = 20*log10(max(processed_data(:))/noise_floor_proc);
320     fprintf('Dynamic_range_Original: %.1f_dB, Processed: %.1f_dB, Change: %.1f_dB\n', ...
321         dr_orig, dr_proc, dr_proc - dr_orig);
322 else
323     fprintf('Dynamic_range_calculation_skipped_insufficient_non-zero_data\n');
324 end
325
326 % Calculate resolution improvement
327 [~, peak_idx] = max(mag_data(:));
328 [peak_row, peak_col] = ind2sub(size(mag_data), peak_idx);
329 orig_width = calculate_target_width(mag_data, peak_row, peak_col);
330 proc_width = calculate_target_width(processed_data, peak_row, peak_col);
331 fprintf('Target_width_improvement: %.1f%%_(%.2f_→_%.2f_bins)\n', ...
332     100*(orig_width-proc_width)/orig_width, orig_width, proc_width);
333

```

```

334 %% Save results if requested
335 if SAVE_RESULTS
336     timestamp = datestr(now, 'yyyymmdd_HHMMSS');
337     filename = sprintf('radar_visualization_%s_%s.mat', method_name, timestamp);
338
339     results.original_data = mag_data;
340     results.processed_data = processed_data;
341     results.x_axis = x_axis;
342     results.y_axis = y_axis;
343     results.method = method_name;
344     results.parameters = struct('deconv_on', DECONV_ON, ...
345         'iterations', deconvIterations, 'regularization', deconvRegularization, ...
346         'cf_window', cfWindowSize, 'cf_threshold', cfThreshold);
347     results.frame_info = struct('frame_idx', frameIdx, 'num_averaged', numFramesToAverage, ...
348         'data_type', data_type);
349
350     save(filename, 'results');
351     fprintf('\nResults saved to: %s\n', filename);
352 end
353
354 fprintf('\nProcessing complete!\n');
355
356 %% Helper Functions
357
358 function width = calculate_target_width(data, peak_row, peak_col)
359     % Calculate 3dB width of a target
360     peak_val = data(peak_row, peak_col);
361     threshold = peak_val / 2; % 3dB point
362
363     % Range direction
364     range_profile = data(peak_row, :);
365     above_threshold = find(range_profile > threshold);
366     if ~isempty(above_threshold)
367         range_width = above_threshold(end) - above_threshold(1) + 1;
368     else
369         range_width = 1;
370     end
371
372     % Cross-range direction
373     cross_profile = data(:, peak_col);
374     above_threshold = find(cross_profile > threshold);
375     if ~isempty(above_threshold)
376         cross_width = above_threshold(end) - above_threshold(1) + 1;
377     else
378         cross_width = 1;
379     end
380
381     % Return geometric mean
382     width = sqrt(range_width * cross_width);
383 end
384
385 function [mag_static, mag_dynamic, y_axis, x_axis] = generate_range_azimuth_heatmap(...
386     radar_data, range_resolution, num_tx, num_rx, antenna_positions, ...
387     min_range_bin, right_discard, min_distance)
388
389 % FFT parameters
390 angle_fft_size = 256;
391 [range_fft_size, doppler_fft_size, ~] = size(radar_data);
392
393 % Angle FFT
394 radar_data_angle = fft(radar_data(:, :, antenna_positions), angle_fft_size, 3);
395
396 % Extract static and dynamic components
397 static_idx = doppler_fft_size/2 + 1;
398
399 % Find dynamic Doppler bins
400 doppler_power = squeeze(sum(mean(abs(radar_data), 3), 1));
401 doppler_power_no_dc = doppler_power([1: doppler_fft_size/2-1, doppler_fft_size/2+3:end]);
402 [peak_val, ~] = max(doppler_power_no_dc);

```

```

403     threshold = peak_val * 0.5;
404     dynamic_bins = find(doppler_power_no_dc > threshold);
405
406     % Adjust indices
407     for i = 1:length(dynamic_bins)
408         if dynamic_bins(i) > doppler_fft_size/2-1
409             dynamic_bins(i) = dynamic_bins(i) + 3;
410         end
411     end
412
413     % Extract data
414     if isempty(dynamic_bins)
415         radar_dynamic = zeros(size(radar_data_angle(:,1,:)));
416     else
417         radar_dynamic = squeeze(sum(abs(radar_data_angle(:,dynamic_bins,:)), 2));
418     end
419     radar_static = squeeze(abs(radar_data_angle(:,static_idx,:)));
420
421     % FFT shift
422     radar_dynamic = fftshift(radar_dynamic, 2);
423     radar_static = fftshift(radar_static, 2);
424
425     % Create coordinate system
426     range_indices = (min_range_bin:range_fft_size-right_discard);
427     sine_theta = -2*((-angle_fft_size/2:angle_fft_size/2)/angle_fft_size);
428     cos_theta = sqrt(1 - sine_theta.^2);
429
430     [R_mat, sine_theta_mat] = meshgrid(range_indices*range_resolution, sine_theta);
431     [~, cos_theta_mat] = meshgrid(range_indices, cos_theta);
432
433     x_axis = R_mat .* cos_theta_mat;
434     y_axis = R_mat .* sine_theta_mat;
435
436     % Extract and transpose data
437     mag_dynamic = radar_dynamic(range_indices+1, [1:end 1])';
438     mag_static = radar_static(range_indices+1, [1:end 1])';
439
440     % Flip for correct orientation
441     mag_dynamic = flipud(mag_dynamic);
442     mag_static = flipud(mag_static);
443
444     % Apply minimum distance filter
445     distances = sqrt(x_axis.^2 + y_axis.^2);
446     distance_mask = distances >= min_distance;
447     mag_dynamic(~distance_mask) = 0;
448     mag_static(~distance_mask) = 0;
449 end
450
451 function deconv_result = apply_richardson_lucy_deconv(y_axis, x_axis, data, iterations, regularization)
452     % Apply advanced Richardson-Lucy deconvolution for FMCW MIMO radar
453
454     % Get radar parameters from the data
455     range_res = mean(diff(x_axis(1,:))); % Estimate range resolution
456
457     % Use the antenna configuration defined earlier
458     evalin('base', 'antenna_config_temp=antenna_config;');
459     antenna_config = evalin('base', 'antenna_config_temp');
460
461     % Check if experimental PSF should be used
462     use_experimental = evalin('base', 'USE_EXPERIMENTAL_PSF');
463
464     if use_experimental
465         % Use experimental PSF
466         fprintf('Using experimental PSF for deconvolution...\n');
467
468         % Get PSF data path and parameters
469         psf_path = evalin('base', 'PSF_DATA_PATH');
470         psf_frame = evalin('base', 'PSF_FRAME_IDX');
471         psf_param_file = evalin('base', 'module_param_file');

```

```

472     psf_calib_path = evalin('base', 'dataFolder_calib');
473     input_path = evalin('base', 'input_path');
474
475     % Generate PSF parameter file
476     pathGenParaFile_PSF = [input_path, 'psf_param.m'];
477     clear(pathGenParaFile_PSF);
478     parameter_file_gen_json(psf_path, psf_calib_path, psf_param_file, ...
479         pathGenParaFile_PSF, 'TDA2');
480
481     % Initialize processing modules for PSF data
482     calibrationObj_PSF = calibrationCascade('pfile', pathGenParaFile_PSF, ...
483         'calibrationfilePath', psf_calib_path);
484     rangeFFTObj_PSF = rangeProcCascade('pfile', pathGenParaFile_PSF);
485     DopplerFFTObj_PSF = DopplerProcClutterRemove('pfile', pathGenParaFile_PSF);
486
487     % Extract experimental PSF with its own processing chain
488     [psf_experimental, psf_info] = extract_experimental_psf(psf_path, psf_frame, ...
489         calibrationObj_PSF, DopplerFFTObj_PSF, rangeFFTObj_PSF, ...
490         'PSFSize', [21, 21], ...
491         'TargetRangeWindow', [4, 6], ... % Adjust based on setup
492         'PlotPSF', true);
493
494     % Store PSF info for later use
495     assignin('base', 'psf_info', psf_info);
496
497     % Call deconvolution with experimental PSF
498     [result, psf_used, conv_info] = chirp_radar_richardson_lucy_deconv_experimental(...
499         x_axis, y_axis, data, psf_experimental, ...
500         'Iterations', iterations, ...
501         'Regularization', regularization, ...
502         'RegularizationType', 'TV', ...
503         'Verbose', true);
504
505 else
506     % Use model-based PSF
507     fprintf('Using model-based PSF for deconvolution...\n');
508
509     % Adaptive PSF size based on data
510     psf_range_size = round(5 * range_res / range_res); % ~5 range cells
511     psf_cross_size = round(size(data, 1) / 10); % ~10% of cross-range
512     psf_size = [min(41, 2*floor(psf_cross_size/2)+1), ...
513         min(41, 2*floor(psf_range_size/2)+1)];
514
515     % Call advanced MIMO deconvolution function
516     [result, psf_used, conv_info] = chirp_radar_richardson_lucy_deconv_mimo(...
517         x_axis, y_axis, data, antenna_config, ...
518         'PSFType', 'mimo_array', ...
519         'PSFSize', psf_size, ...
520         'PSFOversample', 2, ...
521         'ChirpBandwidth', 2.5e9, ...
522         'CenterFrequency', 77e9, ...
523         'RangeResolution', range_res, ...
524         'Iterations', iterations, ...
525         'Regularization', regularization, ...
526         'RegularizationType', 'TV', ...
527         'AccelerationFactor', 1.0, ...
528         'Verbose', true, ...
529         'PlotPSF', true, ...
530         'PlotProgress', false);
531 end
532
533 deconv_result = result.Z_deconv;
534
535 % Print improvement statistics
536 fprintf('Deconvolution improvement factor: %.2f\n', result.improvement_factor);
537 fprintf('Converged: %s after %d iterations\n', ...
538     string(conv_info.converged), conv_info.iterations);
539
540 if ~use_experimental

```

```

541     % Display virtual array statistics
542     virtual_array_size = antenna_config.num_tx * antenna_config.num_rx;
543     fprintf('Virtual array size used: %d elements\n', virtual_array_size);
544     else
545         fprintf('Experimental PSF from range: %2f m\n', psf_info.peak_range_m);
546     end
547 end

```

Додаток Б. Код файлу програми симуляції функції розсіювання ТОЧКИ

Лістинг 2: Код файлу програми chirp_radar_richardson_lucy_deconv_mimo.m

```

1  function [result, psf_used, convergence_info] = chirp_radar_richardson_lucy_deconv_mimo(...)
2      x_axis, y_axis, mag_data, antenna_config, varargin)
3
4  %% Parse input parameters
5  p = inputParser;
6  addRequired(p, 'x_axis', @isnumeric);
7  addRequired(p, 'y_axis', @isnumeric);
8  addRequired(p, 'mag_data', @isnumeric);
9  addRequired(p, 'antenna_config', @isstruct);
10
11  % PSF parameters
12  addParameter(p, 'PSFType', 'mimo_array', @(x) ismember(x, ...
13      {'mimo_array'}));
14  addParameter(p, 'PSFSize', [41, 41], @(x) isnumeric(x) && length(x) == 2);
15  addParameter(p, 'PSFOversample', 4, @isnumeric); % Oversampling for PSF calculation
16
17  % FMCW parameters
18  addParameter(p, 'ChirpBandwidth', 2.5e9, @isnumeric);
19  addParameter(p, 'ChirpDuration', 40e-6, @isnumeric);
20  addParameter(p, 'CenterFrequency', 77e9, @isnumeric);
21  addParameter(p, 'RangeResolution', [], @isnumeric);
22
23  % Algorithm parameters
24  addParameter(p, 'Iterations', 50, @(x) isnumeric(x) && x > 0);
25  addParameter(p, 'Regularization', 0.01, @isnumeric);
26  addParameter(p, 'RegularizationType', 'TV', @(x) ismember(x, {'TV', 'none'}));
27  addParameter(p, 'ConvergenceThreshold', 1e-4, @isnumeric);
28  addParameter(p, 'AccelerationFactor', 1.0, @isnumeric);
29
30  % Display options
31  addParameter(p, 'Verbose', true, @islogical);
32  addParameter(p, 'PlotProgress', false, @islogical);
33  addParameter(p, 'PlotPSF', false, @islogical);
34
35  parse(p, x_axis, y_axis, mag_data, antenna_config, varargin{:});
36  params = p.Results;
37
38  %% Validate antenna configuration
39  if ~isfield(antenna_config, 'tx_pos') || ~isfield(antenna_config, 'rx_pos') || ~isfield(antenna_config, 'wavelength')
40      error('antenna_config must contain tx_pos, rx_pos, and wavelength fields');
41  end
42
43  %% Calculate range resolution if not provided
44  c = 3e8; % Speed of light
45  if isempty(params.RangeResolution)
46      params.RangeResolution = c / (2 * params.ChirpBandwidth);
47  end
48
49  %% Ensure positive data
50  epsilon = 1e-10;

```

```

51  observed_data = max(mag_data, epsilon);
52
53  %% Create PSF based on MIMO array configuration
54  if params.Verbose
55      fprintf('Creating PSF for MIMO array with %d TX and %d RX antennas...\n', ...
56          size(antenna_config.tx_pos, 1), size(antenna_config.rx_pos, 1));
57  end
58
59  psf_used = create_mimo_array_psf(params, antenna_config);
60
61  % Ensure PSF is normalized
62  psf_used = psf_used / sum(psf_used(:));
63
64  % Plot PSF if requested
65  if params.PlotPSF
66      figure('Name', 'MIMOArrayPSF');
67      subplot(2,2,1);
68      imagesc(psf_used); axis image; colorbar;
69      title('2D PSF');
70
71      subplot(2,2,2);
72      plot(psf_used(ceil(end/2), :)); grid on;
73      title('PSF Range Cut');
74      xlabel('Range bins');
75
76      subplot(2,2,3);
77      plot(psf_used(:, ceil(end/2))); grid on;
78      title('PSF Cross-range Cut');
79      xlabel('Cross-range bins');
80
81      subplot(2,2,4);
82      surf(psf_used); shading interp;
83      title('3D PSF View');
84  end
85
86  %% Initialize Richardson-Lucy algorithm
87  estimate = ones(size(observed_data)) * mean(observed_data(:));
88  psf_flipped = rot90(psf_used, 2); % Flip PSF for correlation
89
90  % Convergence tracking
91  convergence_info.residuals = zeros(params.Iterations, 1);
92  convergence_info.relative_changes = zeros(params.Iterations, 1);
93  convergence_info.iterations = 0;
94  convergence_info.converged = false;
95
96  %% Pre-compute regularization operator if needed
97  if ~strcmp(params.RegularizationType, 'none')
98      reg_operator = create_regularization_operator(params.RegularizationType);
99  end
100
101  %% Main Richardson-Lucy iteration loop
102  if params.Verbose
103      fprintf('Starting Richardson-Lucy deconvolution...\n');
104      fprintf('PSF size: %dx%d, Iterations: %d\n', ...
105          params.PSFSize(1), params.PSFSize(2), params.Iterations);
106  end
107
108  prev_estimate = estimate;
109  momentum = zeros(size(estimate));
110
111  for iter = 1:params.Iterations
112      % Store previous estimate
113      prev_prev_estimate = prev_estimate;
114      prev_estimate = estimate;
115
116      %% Forward model: convolve estimate with PSF
117      convolved = conv2(estimate, psf_used, 'same');
118      convolved = max(convolved, epsilon);
119

```

```

120 %% Calculate ratio (observed / model)
121 ratio = observed_data ./ convolved;
122
123 %% Back-project the ratio
124 correction = conv2(ratio, psf_flipped, 'same');
125
126 %% Richardson-Lucy update with acceleration
127 if params.AccelerationFactor > 1 && iter > 1
128     % Nesterov acceleration
129     y = estimate + momentum;
130     new_estimate = y .* correction;
131     momentum = params.AccelerationFactor * (new_estimate - estimate);
132     estimate = new_estimate;
133 else
134     % Standard update
135     estimate = estimate .* correction;
136 end
137
138 %% Apply regularization
139 if ~strcmp(params.RegularizationType, 'none') && params.Regularization > 0
140     estimate = apply_regularization(estimate, params.Regularization, ...
141         params.RegularizationType, reg_operator);
142 end
143
144 % Ensure positivity and prevent extreme values
145 estimate = max(estimate, epsilon);
146
147 % Prevent runaway values - clip to reasonable range
148 max_allowed = max(observed_data(:)) * 10; % Don't allow values > 10x input
149 estimate = min(estimate, max_allowed);
150
151 %% Calculate convergence metrics
152 residual = norm(convolved(:) - observed_data(:), 2);
153 relative_change = norm(estimate(:) - prev_estimate(:), 2) / norm(estimate(:), 2);
154
155 convergence_info.residuals(iter) = residual;
156 convergence_info.relative_changes(iter) = relative_change;
157
158 %% Display progress
159 if params.Verbose && (mod(iter, 10) == 0 || iter == 1)
160     fprintf('Iteration %3d: Residual = %.3e, Relative change = %.3e\n', ...
161         iter, residual, relative_change);
162 end
163
164 %% Plot progress if requested
165 if params.PlotProgress && mod(iter, 5) == 0
166     plot_progress(estimate, observed_data, iter, params.Iterations);
167 end
168
169 %% Check convergence
170 if relative_change < params.ConvergenceThreshold
171     convergence_info.converged = true;
172     convergence_info.iterations = iter;
173     if params.Verbose
174         fprintf('Converged after %d iterations\n', iter);
175     end
176     break;
177 end
178 end
179
180 %% Finalize convergence info
181 if ~convergence_info.converged
182     convergence_info.iterations = params.Iterations;
183     if params.Verbose
184         fprintf('Maximum iterations reached without convergence\n');
185     end
186 end
187
188 % Trim arrays

```

```

189 convergence_info.residuals = convergence_info.residuals(1:convergence_info.iterations);
190 convergence_info.relative_changes = convergence_info.relative_changes(1:convergence_info.iterations);
191
192 %% Prepare output structure
193 result.X = x_axis;
194 result.Y = y_axis;
195 result.Z_original = mag_data;
196 result.Z_deconv = estimate;
197 result.improvement_factor = max(estimate(:)) / max(observed_data(:));
198 result.params = params;
199 result.antenna_config = antenna_config;
200
201 end
202
203 %% Helper function: Create MIMO array PSF
204 function psf = create_mimo_array_psf(params, antenna_config)
205     psf_size = params.PSFSize;
206     psf_type = params.PSFType;
207     oversample = params.PSFOverSample;
208
209     % Extract antenna positions (convert from wavelengths to meters)
210     tx_pos_m = antenna_config.tx_pos * antenna_config.wavelength;
211     rx_pos_m = antenna_config.rx_pos * antenna_config.wavelength;
212
213     % Calculate virtual array positions
214     num_tx = size(tx_pos_m, 1);
215     num_rx = size(rx_pos_m, 1);
216     virtual_array = zeros(num_tx * num_rx, 2);
217
218     idx = 1;
219     for tx_idx = 1:num_tx
220         for rx_idx = 1:num_rx
221             % Virtual element position is TX + RX position
222             virtual_array(idx, :) = tx_pos_m(tx_idx, :) + rx_pos_m(rx_idx, :);
223             idx = idx + 1;
224         end
225     end
226
227     % Create oversampled PSF grid
228     psf_size_os = psf_size * oversample;
229     [X, Y] = meshgrid(1:psf_size_os(2), 1:psf_size_os(1));
230     center_x = (psf_size_os(2) + 1) / 2;
231     center_y = (psf_size_os(1) + 1) / 2;
232
233     % Map PSF coordinates to physical space
234     % Determine spatial extent based on array aperture
235     array_extent_x = max(virtual_array(:, 1)) - min(virtual_array(:, 1));
236     array_extent_y = max(virtual_array(:, 2)) - min(virtual_array(:, 2));
237
238     % PSF spatial extent (in meters)
239     psf_extent_range = params.RangeResolution * 10; % Cover ±10 range bins
240     psf_extent_cross = array_extent_x / 10; % Related to array aperture
241
242     % Convert PSF grid to physical coordinates
243     range_coords = (X - center_x) / (psf_size_os(2) / 2) * psf_extent_range;
244     cross_range_coords = (Y - center_y) / (psf_size_os(1) / 2) * psf_extent_cross;
245
246     switch psf_type
247         case 'mimo_array'
248             % Full MIMO array pattern PSF
249             psf_array = zeros(psf_size_os);
250
251             % Calculate array factor at each PSF point
252             for i = 1:psf_size_os(1)
253                 for j = 1:psf_size_os(2)
254                     % Direction angles
255                     theta = atan2(cross_range_coords(i,j), range_coords(i,j));
256
257                     % Array factor

```

```

258         af = 0;
259         for v = 1:size(virtual_array, 1)
260             % Phase shift for this virtual element
261             k = 2 * pi / antenna_config.wavelength;
262             phase = k * (virtual_array(v, 1) * sin(theta) + ...
263                 virtual_array(v, 2) * cos(theta));
264             af = af + exp(1j * phase);
265         end
266
267         psf_array(i,j) = abs(af)^2;
268     end
269 end
270
271 % Normalize array pattern
272 psf_array = psf_array / max(psf_array(:));
273
274 % Add range response (sinc function)
275 range_normalized = range_coords / params.RangeResolution;
276
277 % Three sinc components of Hanning window transform
278 main_lobe = 0.5 * sinc(range_normalized);
279 left_lobe = 0.25 * sinc(range_normalized + 1);
280 right_lobe = 0.25 * sinc(range_normalized - 1);
281
282 % Combined Hanning window response
283 range_psf = abs(main_lobe + left_lobe + right_lobe).^2;
284
285 % Combined PSF
286 psf = psf_array .* range_psf;
287
288 otherwise
289     % Fallback to simple PSF
290     sigma = min(psf_size_os) / (6 * oversample);
291     psf = exp(-(X - center_x).^2 + (Y - center_y).^2) / (2 * sigma^2));
292 end
293
294 % Downsample to final PSF size
295 psf = downsample_2d(psf, oversample);
296
297 % Ensure PSF is positive and normalized
298 psf = max(psf, 0);
299 psf = psf / sum(psf(:));
300 psf = max(psf, 1e-10);
301 end
302
303 %% Helper function: Downsample 2D array
304 function out = downsample_2d(in, factor)
305     % Simple averaging downsampler
306     [m, n] = size(in);
307     out_m = m / factor;
308     out_n = n / factor;
309     out = zeros(out_m, out_n);
310
311     for i = 1:out_m
312         for j = 1:out_n
313             block = in((i-1)*factor+1:i*factor, (j-1)*factor+1:j*factor);
314             out(i,j) = mean(block(:));
315         end
316     end
317 end
318
319 %% Helper function: Create regularization operator
320 function op = create_regularization_operator(type)
321     switch type
322     case 'TV'
323         % Total variation operator (gradient)
324         op.dx = [0 0 0; -1 1 0; 0 0 0];
325         op.dy = [0 -1 0; 0 1 0; 0 0 0];
326     case 'Tikhonov'

```

```

327         % Laplacian operator
328         op.laplacian = [0 1 0; 1 -4 1; 0 1 0] / 4;
329     end
330 end
331
332 %% Helper function: Apply regularization
333 function regularized = apply_regularization(image, lambda, type, operator)
334     switch type
335     case 'TV'
336         % Total variation regularization
337         dx = conv2(image, operator.dx, 'same');
338         dy = conv2(image, operator.dy, 'same');
339         grad_mag = sqrt(dx.^2 + dy.^2 + 1e-10);
340
341         % TV denoising step
342         dt = 0.05; % Time step
343         div_x = conv2(dx ./ grad_mag, -operator.dx, 'same');
344         div_y = conv2(dy ./ grad_mag, -operator.dy, 'same');
345         regularized = image + lambda * dt * (div_x + div_y);
346     end
347
348     % Ensure positivity
349     regularized = max(regularized, 1e-10);
350 end
351
352 %% Helper function: Plot progress
353 function plot_progress(estimate, observed, iter, max_iter)
354     persistent fig_handle
355
356     if isempty(fig_handle) || ~isvalid(fig_handle)
357         fig_handle = figure('Position', [100, 100, 1200, 400]);
358     else
359         figure(fig_handle);
360     end
361
362     subplot(1,3,1);
363     imagesc(observed);
364     axis image; colorbar;
365     title('Observed Data');
366
367     subplot(1,3,2);
368     imagesc(estimate);
369     axis image; colorbar;
370     title(sprintf('Estimate (Iter %d/%d)', iter, max_iter));
371
372     subplot(1,3,3);
373     imagesc(estimate - observed);
374     axis image; colorbar;
375     title('Difference');
376
377     drawnow;
378 end

```

Додаток В. Код файлу програми оцінки якості калібрування

Лістинг 3: Код файлу програми `assess_calibration_quality.m`

```

1 function [quality_metrics] = assess_calibration_quality(radar_data, antenna_config, range_axis, varargin)
2
3 %% Parse inputs
4 p = inputParser;
5 addRequired(p, 'radar_data', @isnumeric);
6 addRequired(p, 'antenna_config', @isstruct);
7 addRequired(p, 'range_axis', @isnumeric);

```

```

8   addParameter(p, 'PlotResults', true, @islogical);
9   addParameter(p, 'TargetRangeWindow', [], @isnumeric);
10  addParameter(p, 'DopplerBin', [], @isnumeric);
11  parse(p, radar_data, antenna_config, range_axis, varargin{:});
12  params = p.Results;
13
14  %% Initialize
15  [num_range, num_doppler, num_antennas] = size(radar_data);
16
17  % Select Doppler bin (center for static targets)
18  if isempty(params.DopplerBin)
19      doppler_bin = floor(num_doppler/2) + 1;
20  else
21      doppler_bin = params.DopplerBin;
22  end
23
24  % Extract data for analysis
25  range_doppler_slice = squeeze(radar_data(:, doppler_bin, :));
26
27  %% Find calibration target
28  if isempty(params.TargetRangeWindow)
29      % Search entire range
30      range_power = mean(abs(range_doppler_slice).^2, 2);
31      [target_power, target_range_idx] = max(range_power);
32  else
33      % Search within specified window
34      range_mask = (range_axis >= params.TargetRangeWindow(1)) & ...
35          (range_axis <= params.TargetRangeWindow(2));
36      range_indices = find(range_mask);
37      if isempty(range_indices)
38          error('No valid range bins in specified window');
39      end
40      [target_power, rel_idx] = max(mean(abs(range_doppler_slice(range_indices, :)).^2, 2));
41      target_range_idx = range_indices(rel_idx);
42  end
43
44  target_range = range_axis(target_range_idx);
45  fprintf('Calibration target found at range: %.2f m (bin %d)\n', target_range, target_range_idx);
46
47  %% 1. Azimuth Profile Analysis
48  % Perform FFT across antennas for azimuth profile
49  azimuth_fft_size = 256;
50  azimuth_profile_complex = fft(range_doppler_slice(target_range_idx, :), azimuth_fft_size);
51  azimuth_profile = abs(azimuth_profile_complex).^2;
52  azimuth_profile = fftshift(azimuth_profile);
53  azimuth_profile_db = 10*log10(azimuth_profile);
54
55  % Find azimuth mainlobe
56  [az_peak_val, az_peak_idx] = max(azimuth_profile);
57  az_peak_val_db = 10*log10(az_peak_val);
58
59  % Auto-detect azimuth mainlobe width
60  az_mainlobe_mask = azimuth_profile_db >= (az_peak_val_db - 3);
61  az_mainlobe_indices = find(az_mainlobe_mask);
62  az_peak_group = find_contiguous_group(az_mainlobe_indices, az_peak_idx);
63  az_mainlobe_start = min(az_peak_group);
64  az_mainlobe_end = max(az_peak_group);
65  az_mainlobe_width = az_mainlobe_end - az_mainlobe_start + 1;
66
67  % Define azimuth sidelobe regions
68  az_sidelobe_mask = true(azimuth_fft_size, 1);
69  az_sidelobe_mask(max(1, az_mainlobe_start-2):min(azimuth_fft_size, az_mainlobe_end+2)) = false;
70
71  % Calculate azimuth PSLR
72  if any(az_sidelobe_mask)
73      az_peak_sidelobe = max(azimuth_profile(az_sidelobe_mask));
74      azimuth_pslr = az_peak_val / az_peak_sidelobe;
75      azimuth_pslr_db = 10*log10(azimuth_pslr);
76  else

```

```

77     azimuth_pslr_db = Inf;
78 end
79
80 % Calculate azimuth ISLR
81 if any(az_sidelobe_mask)
82     az_mainlobe_power = sum(azimuth_profile(az_mainlobe_start:az_mainlobe_end));
83     az_sidelobe_power = sum(azimuth_profile(az_sidelobe_mask));
84     azimuth_islr = az_mainlobe_power / az_sidelobe_power;
85     azimuth_islr_db = 10*log10(azimuth_islr);
86 else
87     azimuth_islr_db = Inf;
88 end
89
90 %% 2. Phase and Amplitude Balance Analysis
91 target_complex_values = range_doppler_slice(target_range_idx, :);
92
93 % Phase analysis
94 phases = angle(target_complex_values);
95 phase_unwrapped = unwrap(phases);
96 phase_std_rad = std(phase_unwrapped);
97 phase_std_deg = phase_std_rad * 180/pi;
98
99 % Amplitude analysis
100 amplitudes = abs(target_complex_values);
101 amplitude_mean = mean(amplitudes);
102 amplitude_std = std(amplitudes);
103 amplitude_imbalance_db = 20*log10(max(amplitudes)/min(amplitudes));
104
105 %% 3. Calculate Overall Quality Score
106 % Scoring criteria (higher is better) - Now only 4 metrics
107 scores = [];
108
109 % Azimuth PSLR score
110 if isfinite(azimuth_pslr_db)
111     scores(end+1) = min(100, max(0, (azimuth_pslr_db - 10) * 5));
112 else
113     scores(end+1) = 100;
114 end
115
116 % Azimuth ISLR score
117 if isfinite(azimuth_islr_db)
118     scores(end+1) = min(100, max(0, (azimuth_islr_db - 5) * 10));
119 else
120     scores(end+1) = 100;
121 end
122
123 % Phase stability score (good: <10deg, excellent: <5deg)
124 scores(end+1) = min(100, max(0, 100 - phase_std_deg * 10));
125
126 % Amplitude balance score (good: <3dB, excellent: <1dB)
127 scores(end+1) = min(100, max(0, 100 - amplitude_imbalance_db * 20));
128
129 overall_score = mean(scores);
130
131 %% Store metrics
132 quality_metrics.azimuth_pslr_db = azimuth_pslr_db;
133 quality_metrics.azimuth_islr_db = azimuth_islr_db;
134 quality_metrics.phase_std_deg = phase_std_deg;
135 quality_metrics.amplitude_imbalance_db = amplitude_imbalance_db;
136 quality_metrics.overall_score = overall_score;
137 quality_metrics.target_range = target_range;
138 quality_metrics.mainlobe_width_azimuth = az_mainlobe_width;
139
140 %% Display results
141 fprintf('\n====CalibrationQualityMetrics====\n');
142 fprintf('TargetRange: %2f_m\n', target_range);
143 fprintf('\nAzimuthDomain: \n');
144 fprintf('PSLR: %1f_dB\n', azimuth_pslr_db);
145 fprintf('ISLR: %1f_dB\n', azimuth_islr_db);

```

```

146 fprintf('MainlobeWidth:%d bins\n', az_mainlobe_width);
147 fprintf('\nChannelBalance:\n');
148 fprintf('PhaseSTD:%.1f degrees\n', phase_std_deg);
149 fprintf('AmplitudeImbalance:%.1f dB\n', amplitude_imbalance_db);
150 fprintf('\nOverallQualityScore:%.1f/100\n', overall_score);
151
152 %% Plot results if requested
153 if params.PlotResults
154     plot_calibration_analysis(azimuth_profile_db, ...
155         az_peak_idx, az_mainlobe_start, az_mainlobe_end, ...
156         phases, amplitudes, quality_metrics);
157 end
158
159 end
160
161 %% Helper function: Find contiguous group containing target index
162 function group = find_contiguous_group(indices, target_idx)
163     if isempty(indices)
164         group = target_idx;
165         return;
166     end
167
168     % Find which group contains target
169     group_start = find(indices == target_idx);
170     if isempty(group_start)
171         group = target_idx;
172         return;
173     end
174
175     % Expand to find contiguous region
176     group = target_idx;
177
178     % Expand left
179     for i = group_start-1:-1:1
180         if indices(i) == indices(i+1) - 1
181             group = [indices(i), group];
182         else
183             break;
184         end
185     end
186
187     % Expand right
188     for i = group_start+1:length(indices)
189         if indices(i) == indices(i-1) + 1
190             group = [group, indices(i)];
191         else
192             break;
193         end
194     end
195 end
196
197 %% Helper function: Plot calibration analysis
198 function plot_calibration_analysis(azimuth_profile_db, ...
199     az_peak_idx, az_mainlobe_start, az_mainlobe_end, ...
200     phases, amplitudes, quality_metrics)
201
202     figure('Position', [50, 50, 1200, 600], 'Name', 'CalibrationQualityAnalysis');
203
204     % Azimuth profile
205     subplot(2,2,1);
206     az_angles = linspace(-90, 90, length(azimuth_profile_db));
207     plot(az_angles, azimuth_profile_db, 'b-', 'LineWidth', 1.5);
208     hold on;
209     plot(az_angles(az_peak_idx), azimuth_profile_db(az_peak_idx), 'ro', ...
210         'MarkerSize', 8, 'MarkerFaceColor', 'r');
211
212     % Mark mainlobe
213     if az_mainlobe_start <= az_mainlobe_end
214         xregion = az_angles([az_mainlobe_start, az_mainlobe_end]);

```

```

215         patch([xregion(1) xregion(2) xregion(2) xregion(1)], ...
216             [min(azimuth_profile_db)-5 min(azimuth_profile_db)-5 max(azimuth_profile_db)+5 max(azimuth_profile_db)+5], ...
217             'g', 'FaceAlpha', 0.2, 'EdgeColor', 'none');
218     end
219
220     xlabel('Азимут_градуси');
221     ylabel('Потужність_дБ');
222     title(sprintf('Азимутальний_профіль_PSLR:_%1f_дБ, ISLR:_%1f_дБ', ...
223         quality_metrics.azimuth_pslr_db, quality_metrics.azimuth_islr_db));
224     grid on;
225     ylim([max(azimuth_profile_db)-50, max(azimuth_profile_db)+5]);
226
227     % Phase distribution
228     subplot(2,2,2);
229     antenna_indices = 1:length(phases);
230     plot(antenna_indices, phases*180/pi, 'bo');
231     xlabel('Номер_каналу');
232     ylabel('Фаза_градуси');
233     title(sprintf('Середньоквадратичне_відхилення:_%1f°', quality_metrics.phase_std_deg));
234     grid on;
235
236     % Amplitude distribution
237     subplot(2,2,[3,4]);
238     bar(antenna_indices, 20*log10(amplitudes/mean(amplitudes)));
239     xlabel('Номер_каналу');
240     ylabel('Відносна_амплітуда_дБ');
241     title(sprintf('Амплітудна_неузгодженість:_%1f_дБ', ...
242         quality_metrics.amplitude_imbalance_db));
243     grid on;
244     ylim([-5, 5]);
245
246     % Quality score breakdow
247     categories = {'Азимутальне_PSLR', 'Азимутальне_ISLR', ...
248         'Фазова_неузгодженість', 'Амплітудна_неузгодженість'};
249     scores = [
250         min(100, max(0, (quality_metrics.azimuth_pslr_db - 10) * 5));
251         min(100, max(0, (quality_metrics.azimuth_islr_db - 5) * 10));
252         min(100, max(0, 100 - quality_metrics.phase_std_deg * 10));
253         min(100, max(0, 100 - quality_metrics.amplitude_imbalance_db * 20))
254     ];
255
256     sgtitle('Оцінка_калібрування_ММО_радару', 'FontSize', 16);
257 end

```