

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

“На правах рукопису”

УДК _____

“До захисту допущено”

Завідувач кафедри ЦТЕ
_____ **Наталія АУШЕВА**

“__” _____ 2023р.

Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою
“Цифрові технології в енергетиці”
зі спеціальності 122 **“Комп’ютерні науки”**
на тему: Засоби інформаційної підтримки ідентифікації повітряних об’єктів на
індикаторах кругового огляду

Виконав: студент 2 курсу, групи ТР-22 мп
ОГНЯНИК Дмитро Миколайович

(прізвище, ім’я, по батькові)

(підпис)

Науковий керівник: *доцент каф. цифрових технологій в енергетиці*
доц., к.в.н., Онисько Андрій Ілліч

(посада, науковий ступінь, вчене звання, прізвище, ім’я, по батькові)

(підпис)

Рецензент: _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім’я, по батькові)

(підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2023

Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ
Кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ
Рівень вищої освіти – другий (магістерський)
Спеціальність 122 “Комп’ютерні науки”
Освітньо-професійна програма “Цифрові технології в енергетиці”

ЗАТВЕРДЖУЮ
Завідувач кафедри ЦТЕ
Наталія АУШЕВА

«__»_____2023 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Огнянику Дмитру Миколайовичу

(прізвище, ім’я, по батькові)

1. Тема дисертації Засоби інформаційної підтримки ідентифікації повітряних об’єктів на індикаторах кругового огляду

Науковий керівник дисертації **Онисько Андрій Ілліч, к.в.н., доцент**

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “06” листопада 2023 року №5152-с

2. Термін подання студентом дисертації 18 грудня 2023р

3. Вихідні дані: операційна система Windows, мова програмування Python, середовище розробки PyCharm, графічний інтерфейс MongoDB Compass.

4. Перелік завдань, які потрібно розробити: проаналізувати принципи роботи радіолокаторів, здійснити пошук методів обробки сигналів з РЛС, продемонструвати обрані методи, розробити програмне забезпечення для обробки сигналів з радіолокаторів, представити архітектуру програмного забезпечення та принципи його роботи, надати приклад роботи програмного забезпечення, представити інструкцію по встановленню програмного забезпечення, провести аналіз програми як стартап-проекту.

5. Орієнтовний перелік графічного (ілюстративного) матеріалу: схеми архітектури системи, діаграми прецедентів, діаграми послідовності, зразки розробленого інтерфейсу, зразки роботи програми.

6. Орієнтовний перелік публікацій: II Міжнародна науково-практична інтернет-конференція «Education via Distance Learning and other Pedagogical Challenges» 21-22

вересня 2023, м. Дніпро

7. Дата видачі завдання «24» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Отримання завдання	01.08.2023	
2.	Вивчення та аналіз задачі	01.08.2023–14.08.2023	
3.	Збір інформації	15.08.2023–31.08.2023	
4.	Аналіз вимог завдання, розробка методів і засобів розв'язання оставленої задачі	01.09.2023–30.09.2023	
5.	Розробка та тестування програмного продукту	01.10.2023–14.10.2023	
6.	Підготовка матеріалів магістерської роботи	15.10.2023–20.11.2023	
7.	Захист програмного продукту	24.10.2023	
8.	Передзахист	05.12.2023	
9.	Захист		

Студент

(підпис)

Дмитро ОГНЯНИК

(ім'я, ПРІЗВИЩЕ)

Науковий керівник

(підпис)

Андрій ОНИСЬКО

(ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Актуальність теми. Розвиток технологій у сучасному світі супроводжується зростанням потреб у забезпеченні безпеки та ефективного використання повітряного простору. Розробка програмного забезпечення для аналізу даних з радарів та ідентифікації повітряних об'єктів має важливе значення в контексті забезпечення національної безпеки та військових операцій в країні.

Метою роботи є розробка програмного забезпечення, яке забезпечить аналіз та обробку даних, одержаних від радарів, яке являє собою систему ідентифікації повітряних об'єктів для поліпшення управління повітряним простором та забезпечення оперативної інформації операторам.

Завдання дослідження:

- проаналізувати принципи роботи радіолокаторів;
- здійснити пошук методів обробки сигналів з РЛС;
- продемонструвати обрані методи;
- розробити програмне забезпечення для обробки сигналів з радіолокаторів;
- представити архітектуру програмного забезпечення та принципи його роботи;
- надати приклад роботи програмного забезпечення;
- представити інструкцію по встановленню програмного забезпечення;
- провести аналіз програми як стартап-проекту.

Об'єкт дослідження – інформаційні методи обробки сигналів із радіолокаторів.

Предмет дослідження – метод інформаційної підтримки задля ідентифікації повітряних об'єктів на індикаторах кругового огляду.

Практична цінність розробленого програмного забезпечення має високий рівень для сфери військової та цивільної безпеки. Воно дозволяє оперативно та ефективно аналізувати повітряний простір, ідентифікувати потенційно небезпечні об'єкти та сприяє збереженню безпеки в повітряному просторі.

Апробація результатів дисертації. Основні положення даної роботи доповідались та обговорювались на:

II Міжнародній науково-практичній інтернет-конференції «Education via Distance Learning and other Pedagogical Challenges»

Дисертація складається з вступу, п'яти розділів та висновків. Повний обсяг дисертації складає 130 сторінок, в тому числі 120 сторінок основного тексту, 28 таблиць, 65 рисунків, 4 сторінки списку використаних джерел у кількості 45 найменувань.

Ключові слова: радіолокаційна станція, симулятор, ефективна площа розсіювання, AES-шифрування, постріляційні бази даних.

ABSTRACT

Actuality of theme. The development of technologies in the modern world is accompanied by an increase in the need to ensure safety and efficient use of air space. The development of software for the analysis of data from radars and identification of aerial objects is important in the context of ensuring national security and military operations both in peacetime and during hostilities in the country.

The purpose of the work is to develop software that will provide analysis and processing of data received from radars, which is a system of identification of aerial objects to improve airspace management and provide operational information to operators.

Objectives of the study:

- analyze the principles of operation of radars;
- search for radar signal processing methods;
- demonstrate the chosen methods;
- develop software for processing radar signals;
- to present the architecture of the software and the principles of its operation;
- provide an example of how the software works;
- present instructions for installing the software;
- analyze the program as a startup project.

The object of research is informational methods of processing radar signals.

The subject of research is the method of information support for the identification of aerial objects on the circular view indicators.

The practical value of the developed software has a high level for the field of military and civil security. It allows for quick and effective analysis of airspace, identification of potentially dangerous objects and contributes to maintaining safety in airspace.

Approbation of the results of the dissertation. The main provisions of this work were reported and discussed at:

II International Scientific and Practical Internet Conference "Education via Distance Learning and other Pedagogical Challenges"

The dissertation consists of an introduction, five chapters and conclusions. The full volume of the dissertation is 130 pages, including 120 pages of the main text, 28 tables, 65 figures, 4 pages of the list of used sources in the amount of 45 names.

Keywords: radar station, simulator, effective scattering area, AES-encryption, post-radiation databases.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ	11
ВСТУП.....	12
1 АКТУАЛЬНІСТЬ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ ІНДИКАТОРІВ КРУГОВОГО ОГЛЯДУ	14
1.1 Поява та розвиток радіолокації	14
1.1.1 Загальні відомості про радіолокацію.....	14
1.1.2 Перші радіолокаційні станції та їх розвиток.....	18
1.1.3 Різновиди РЛС	20
1.2 Використання РЛС в Збройних силах України	22
Висновки до розділу 1	31
2 МЕТОДИ ОТРИМАННЯ Й ОБРОБКИ ДАНИХ З РАДІОЛОКАТОРІВ	32
2.1 Аналогічна українська система безпеки – Delta.....	32
2.2 Методика збору інформації із радару, за допомогою плати Arduino	36
2.2.1 Про мікроконтролер Arduino	36
2.2.2 Конструкція радару на платах Arduino.....	37
2.2.3 Особливості роботи радару на базі Arduino.....	39
2.3 Математичні методи обрахунку даних з РЛС.....	42
2.3.1 Розрахунок ефективної площі розсіювання об'єкту.....	42
2.3.2 Розрахунок радіальної швидкості об'єкту	45
2.3.3 Розрахунок відстані до об'єкту.....	49
2.3.4 Розрахунок азимуту об'єкту	51
2.4 Інструменти програмної реалізації.....	54
2.4.1 Система керування базами даних MongoDB	54

2.4.2	Графічний інтерфейс MongoDB Compass	56
2.4.3	Мова програмування Python	58
2.4.4	Середовище PyCharm	59
2.4.5	Бібліотека tkinter	61
2.4.6	Бібліотека json.....	62
2.4.7	Бібліотека numpy	63
2.4.8	Бібліотека tomlib.....	64
2.4.9	Бібліотека random	66
2.4.10	Бібліотека hashlib.....	66
2.4.11	Бібліотека pymongo	67
	Висновки до розділу 2.....	69
3	ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ЗАСТОСУНКУ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ ДЛЯ ІДЕНТИФІКАЦІЇ ПОВІТРЯНИХ ОБ'ЄКТІВ	70
3.1	Архітектура програмного забезпечення.....	70
3.2	Принципи роботи складових частин програмного забезпечення	72
3.2.1	Опис роботи бази даних у системі.....	72
3.2.2	Модуль симулятору радіолокаційної станції та генератору об'єктів	76
3.2.3	Модуль алгоритму порівняння даних.....	77
3.2.4	Модуль шифрування та дешифрування даних	78
3.3	Діаграма послідовності розробленої програми	80
3.4	Діаграма прецедентів розробленої програми.....	81
	Висновки до розділу 3.....	83
4	ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	84
4.1	Робота користувача за допомогою створеної програми.....	84
4.2	Робота користувача із базою даних	88

4.3 Інсталяція програми	91
4.3.1 Інсталяція Python 3.11	92
4.3.2 Інсталяція й налаштування MongoDB	95
4.3.2 Інсталяція і запуск програми через консоль	97
4.4 Шляхи покращення розробленого програмного продукту	98
Висновки до розділу 4	101
5 Розроблення стартап-проекту	102
5.1 Опис ідеї стартап-проекту	102
5.2 Технологічний аудит ідеї проекту	103
5.3 Моніторинг ринкових можливостей запуску стартап-проекту	104
5.4 Розроблення ринкової стратегії продукту	112
5.5 Розроблення маркетингової програми проекту	115
Висновки до розділу 5	118
ВИСНОВКИ	119
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	121
ДОДАТОК А	125

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

РЛС – радіолокаційна система

ЕПР – ефективна площа розсіювання

АВАКС – система дальнього радіолокаційного виявлення і управління

ППО – протиповітряна оборона

SHORAD – система попередження та наведення зброї протиповітряної оборони малої дальності

ЕСМ – радіоелектронна протидія

ARM – протирадіолокаційна ракета

ЗСУ – Збройні сили України

НАТО – Організація Північноатлантичного договору

ЕМХ – електромагнітна хвиля

JSON – JavaScript Object Notation

IDE – Integrated Development Environment

ХСА – характеристики спрямованості антени

ВСТУП

Із відкриттям різних видів радіохвиль людство отримало величезну кількість можливостей щодо їх застосування. Це наприклад радіозв'язок, телебачення, а також радіолокація. Радіолокація - це технологія для виявлення об'єктів (цілей) та визначення їх місцезнаходження та характеристик руху за допомогою радіотехнічних засобів і методів. Цей процес відомий як радіолокаційне спостереження, а пристрої, які його реалізують, називаються радіолокаційними станціями (РЛС) [1] або радіолокаторами. Оскільки майже всі нові винаходи людства швидко знаходять своє застосування у військовій сфері, то на початку Другої світової війни передові світові держави вперше оснастили свої військові літаки радарми. Цей прилад дозволяв помічати інші повітряні цілі в таких умовах або на таких відстанях, коли людським зором це було неможливо зробити. Через кілька років після дебюту РЛС почали з'являтися перші спроби вже навести в оману інші РЛС, за допомогою дипольних протирадарних відбивачів [2], котрі отримали назву Window та показали себе ефективним засобом боротьби з сигналами РЛС.

В сучасному воєнному контексті використання радіолокаційних станцій (РЛС) стає вирішальною у забезпеченні безпеки та ефективності оборонно-військових операцій. РЛС є ключовими елементами систем виявлення та відстеження повітряних об'єктів, від літаків і гелікоптерів до крилатих та балістичних ракет. Їх роль у виявленні ворожих цілей, визначенні їх характеристик, а також в організації управління озброєнням надає РЛС ключове значення в сучасних військових конфліктах.

Однак з виникненням нових технологій та розвитком методів ведення війни, сучасні РЛС потребують постійного вдосконалення та адаптації. Оператори РЛС зіткаються з великим обсягом даних, які необхідно швидко та точно обробляти, щоб приймати стратегічно важливі рішення в реальному часі. Людський фактор, такий як втома та можливі помилки, може впливати на точність та ефективність цього процесу.

На даний момент кінець 2023 року, повномасштабна україно-російська війна триває другий рік і звісно що в цій війні застосовуються передові зразки озброєння. Це такі об'єкти як військові літаки різного призначення, гелікоптери, крилаті ракети та балістичні ракети. Всі ці об'єкти потенційно є дуже небезпечними як для солдат на бойових постах, так і для цивільних жителів. Для стеження за цими цілями в тому числі застосовують різноманітні радіолокатори, що мають різні розміри, характеристики і тд. [3] Операторам таких станцій слід проходити навчання, щоб приступити до роботи з ними та розумітися в особливостях роботи конкретної РЛС. Також, слід врахувати фактори людської похибки, під час роботи з РЛС, а також втоми оператора під час довготривалої роботи. Всі ці фактори можуть впливати на якість аналізу побаченого на екранах РЛС та на висновки і подальші дії, котрі зробить оператор станції [4].

Моя робота має на меті розробку програмного забезпечення, яке автоматизує та полегшує процес роботи з РЛС. Це програмне забезпечення не лише допомагатиме в оперативному виявленні та аналізі повітряних об'єктів, але й враховуватиме фактори втоми та мінімізуватиме можливі людські помилки.

У даній роботі будуть розглянуті ключові аспекти роботи з РЛС, особливості їх функціонування та вирішені завдання автоматизації та оптимізації процесів, пов'язаних з веденням нагляду та управління повітряним простором. Розроблене програмне забезпечення сприятиме підвищенню ефективності використання РЛС, забезпечуючи операторам потужний та точний інструмент для подальшого забезпечення безпеки та успішності оборонно-військових операцій.

Підсумовуючи все вищесказане можна сказати, що актуальність створення такої програми, що дещо автоматизує процеси моніторингу повітряного простору за допомогою радіолокаторів, є безумовною. Саме це зможе допомогти операторам станцій та можливо врятує більше життів, в умовах війни.

1 АКТУАЛЬНІСТЬ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ ІНДИКАТОРІВ КРУГОВОГО ОГЛЯДУ

Даний розділ пояснює актуальність досліджуваної теми, розглядаються описує принципи роботи радарів, їх види, наводиться перелік радарів на озброєнні Збройних сил України. Описується призначення систем автоматизованого керування індикаторами кругового огляду, розробка яких є метою даної роботи.

1.1 Поява та розвиток радіолокації

Радіолокація (від лат. *radius* — промінь і *locatio* — розміщення, положення) — галузь науки і техніки, предметом якої є спостереження радіотехнічними методами різних об'єктів (цілей), їхнє виявлення, розпізнавання, вимірювання координат і визначення характеристик за допомогою радіохвиль [4].

1.1.1 Загальні відомості про радіолокацію

Під радіолокацією також розуміють процес радіолокаційного спостереження (локації) об'єктів. Радіонавігація дуже тісно пов'язана із радіолокацією тісно; часто їхні методи та апаратуру практично не розрізняють. Завдання радіолокації вирішуються за допомогою окремих радіолокаційних станцій (РЛС) і складних радіолокаційних систем. За наявності декількох об'єктів радіолокація повинна забезпечувати необхідне роздільне спостереження [5].

Принцип дії радіолокації засновано на фізичних явищах:

- радіохвилі розсіюються на електричних неоднорідностях (об'єктах з іншими електричними властивостями, які відмінні від властивостей середовища поширення), що трапляються на шляху розповсюдження хвиль;

— на великих відстанях від РЛС можна вважати, що радіохвилі поширюються прямолінійно і з постійною швидкістю, що надає можливість вимірювати дальність і кутові координати цілі;

— частота прийнятого сигналу відрізняється від частоти випромінених коливань при переміщенні цілі щодо РЛС (ефект Доплера), що дозволяє вимірювати радіальні швидкості руху цілі відносно РЛС;

— під час пасивної радіолокації використовують випромінювання електромагнітних хвиль об'єктами спостереження (це може бути теплове випромінювання, властиве всім об'єктам, активне випромінювання, створюване технічними засобами об'єкта, або побічне випромінювання, створюване будь-якими об'єктами із працюючими електричними пристроями).

Для огляду навколишнього простору використовують різні способи радіолокації за рахунок переміщення діаграми спрямованості антени РЛС: круговий, секторний, огляд по гвинтовій лінії, конічний, по спіралі, «V» — огляд, лінійний — наприклад, на літаках Ан-71 (Україна), А-50 (РФ), літаках із системою АВАКС (США). Радіолокацію розподіляють також за використанням діапазоном радіохвиль, за видом зондувального сигналу, кількістю застосовуваних каналів, числом і видом вимірюваних координат, місцем установки РЛС тощо.

Інформація, яку отримують під час радіолокаційного спостереження, носить назву радіолокаційної інформації. Пристрої та системи, які використовуються для радіолокаційного спостереження, отримали найменування радіолокаційних станцій (РЛС) або радіолокаторів. Об'єкти, які піддаються радіолокаційному спостереженню, відомі як радіолокаційні цілі або просто цілі. У випадках використання відбитих радіохвиль, радіолокаційними цілями можуть бути будь-які неоднорідності в електричних параметрах навколишнього середовища (діелектрична та магнітна проникність, провідність), де поширюється початкова радіохвиля. Серед них можна визначити літальні апарати (літаки, вертольоти, метеорологічні зонди і т. д.), гідрометеоутворення (дощ, сніг, град, хмари і т. д.), річкові та морські судна, наземні об'єкти (дерева, будинки, автомобілі та інші), військові об'єкти та інше [6].

Радіолокаційний сигнал служить джерелом радіолокаційної інформації, і види радіолокаційного спостереження визначаються залежно від методів отримання радіолокаційних сигналів.

Розрізняють радіолокацію:

1. Пасивну (заснована на прийомі власного випромінювання об'єкта) на рисунку 1.1;

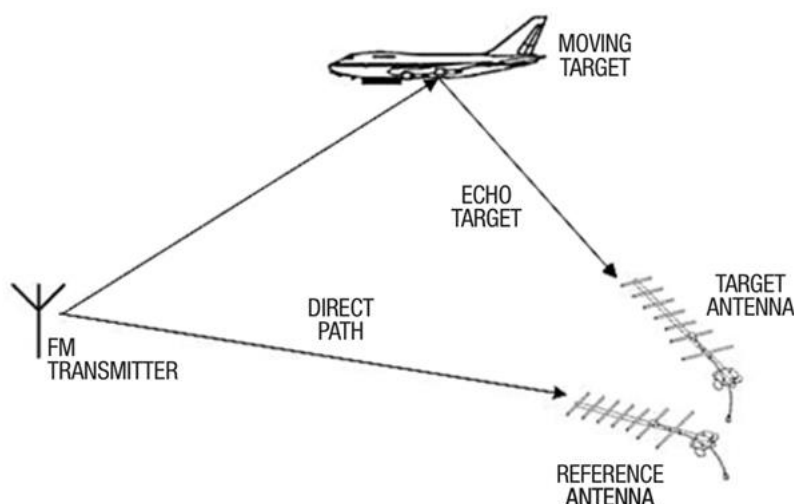


Рисунок 1.1 – Пасивна радіолокація

Радіолокація з пасивною відповіддю, базується на тому, що коливання, випромінювані радіолокаційною станцією (зондуючий сигнал), відбиваються від цілі та потрапляють у приймач РЛС у вигляді відбитого сигналу, також відомого як ехо-сигнал [7].

Цей вид спостереження часто називають активною радіолокацією з пасивною відповіддю, і він є найбільш поширеним. Важливою умовою для цілей у цьому випадку є їхні властивості, які відрізняються від властивостей навколишнього середовища, оскільки відбитий сигнал містить інформацію про цілі, визначену радіолокаційним контрастом.

2. Активну (за якої РЛС випромінює свій власний зондувальний імпульс і приймає його відбитим від об'єкта та визначає характеристики об'єкта залежно від параметрів прийнятого сигналу) на рисунку 1.2.

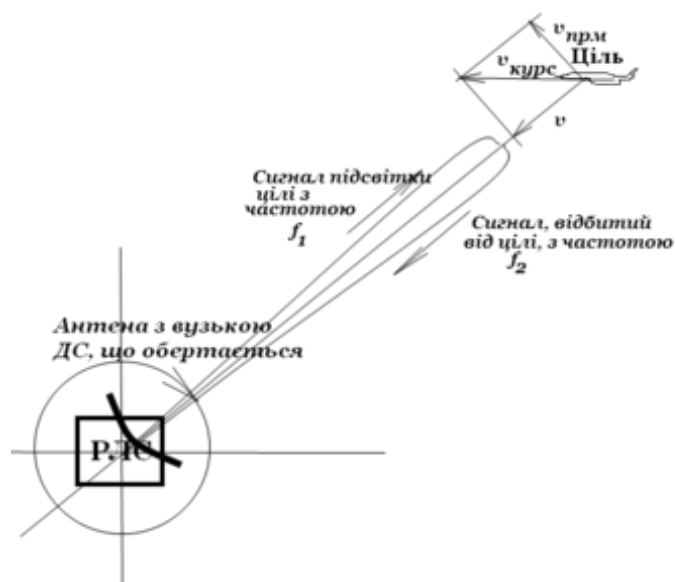


Рисунок 1.2 – Активна радіолокація

Активна радіолокація буває двох видів — з активною і пасивною відповіддю. За активної радіолокації з активною відповіддю на об'єкті передбачено наявність радіопередавача, що випромінює радіохвилі як відповідь на прийнятий сигнал. Активну відповідь застосовують для розпізнавання об'єктів («свій — чужий»), дистанційного управління, а також для одержання від них додаткової інформації (кількість палива, тип об'єкта тощо). За активної радіолокації з пасивною відповіддю зондуєчий сигнал відбивається від об'єкта і приймається РЛС як відповідь [8].

Активна радіолокація, також відома як радіолокація з активною відповіддю, характеризується тим, що сигнал, який повертається, не є просто відбитим, але є перевипроміненим з використанням спеціального відповідача, такого як ретранслятор. Це підвищує дальність радіолокаційного спостереження, роблячи його ефективним для використання у сферах, таких як спостереження за штучними супутниками Землі та визначення державної належності літаків за допомогою спеціальних кодів.

У цивільній авіації метод активної відповіді широко використовується, оскільки сигнал може надавати багато додаткової корисної інформації, такої як висота польоту, обчислена більш точним бортовим висотоміром, ніж можливо

наземними РЛС. Використання автоматизованих систем контролю повітряного руху є особливо важливим у цьому контексті.

Пасивна радіолокація, ґрунтується на прийомі власного радіовипромінювання цілей. У відміну від попередніх випадків, де зондуючий сигнал може бути використаний як опорний для вимірювання дальності і швидкості, у цьому випадку така можливість відсутня. Метод пасивної радіолокації застосовується для спостереження за станом посівів, визначення вологості ґрунту, виявлення лісових і підземних пожеж, а також для вирішення навігаційних завдань, таких як вимірювання шляхової швидкості. Відомі також військові застосування, пов'язані з виявленням штучних об'єктів, а також спостереження за грозами в довгохвильовому діапазоні.

В умовах зростаючої інтенсивності повітряного руху та переходу до автоматизації літаководіння та управління повітряним рухом роль радіолокації стає ще більш важливою для забезпечення регулярних і безпечних польотів. Не кажучи вже про території де ведуться бойові дії із використанням авіації, що потребує постійно відслідковувати переміщення як ворожих, так і союзних повітряних об'єктів, задля планування операцій.

1.1.2 Перші радіолокаційні станції та їх розвиток

Радіолокаційна станція (РЛС, радар (англ. radar від (ra)dio (d)etection (a)nd (r)anging) — це пристрій для виявлення повітряних, космічних, морських і наземних об'єктів, а також для визначення їх положення відносно станції, геометричних розмірів та параметрів руху: висоти, напрямку, швидкості, визначення «свій-чужий».

Термін «радар» започатковано в 1941 році у США. Історія радарної технології почалась 1887 з експериментів Г. Герца, який помітив, що радіохвилі вільно проходять через одні тіла й затримуються іншими. У 1900 Н. Тесла передбачив використання радіохвиль для пошуку суден у відкритому морі. [1]

Уперше ідея радару була реалізована винахідником К. Хюльсмайєром (1881-1957, Німеччина), рисунок 1.3. 1904 він запатентував прилад, названий

телемобілоскопом — перший пристрій, що був здатний виявляти присутність віддалених предметів за допомогою радіохвиль (хоча й не міг безпосередньо вимірювати відстань до цілі). Попри недосконалість, апарат містив основні елементи майбутнього радіолокатора. Апарат складався з іскрового генератора радіохвиль, двох обертових антен та радіоприймача з індикатором.



Рисунок 1.3 – Крістіан Хюльсмайєр

У 1933 Л. Хайланд (1897-1989), А. Тейлор (1899-1961) і Л. Янг (1891-1981), усі – США, отримали патент на ідею використання приймача й передавача коротких хвиль для виявлення місцезнаходження літака. В 1935 році у Великій Британії було створено й запатентовано першу імпульсну РЛС дальнього виявлення — результат досліджень групи на чолі з Р. А. Вотсоном-Воттом (1892-1973). Уперше її застосували на початку Другої світової війни для захисту від нальотів німецької авіації. 1941 почалось серійне виробництво панорамних РЛС кругового огляду з антенами, що обертаються, які пеленгували об'єкти в усіх напрямках. [9]

Першою серійною РЛС у Радянському союзі стала РУС-2, на рисунку 1.4, що розробляли в 1936-1940х роках.

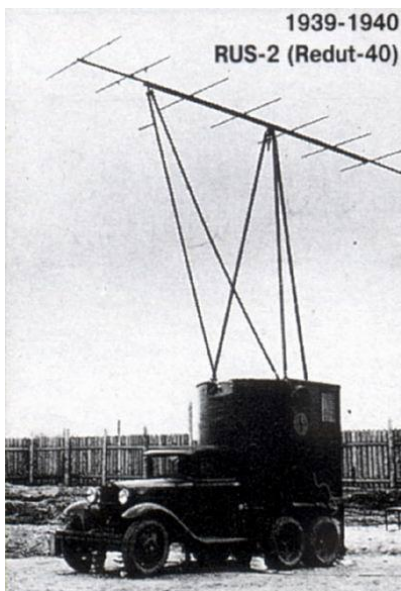


Рисунок 1.4 – РУС-2

У радарі використовувалися окремі передавальні та приймальні антени, обидві з яких оберталися синхронно з відповідними кабінами на автомобільному шасі ГАЗ-3А. Антени були антени Яги з 7 елементами. Відображення прийнятих імпульсів здійснювалося у приймальній кабіні на А-скопі. Пізніші версії (з 1942 року) використовували більшу кабіну на шасі ЗІЗ-6, в якій оберталися лише антени. Старіші версії тоді називалися «Редут-40». 30 таких радарів використовувалися у протиповітряній обороні Ленінграда та Москви.

1.1.3 Різновиди РЛС

РЛС відрізняються переважно залежно від конкретних завдань, які вони виконують, рисунок 1.5. Це може включати РЛС для систем управління повітряним рухом, РЛС для виявлення чи наведення зенітних керованих ракет, РЛС для пошуку космічних літальних апаратів і їх зближення, а також літакові РЛС з круговим або бічним оглядом і багато інших.

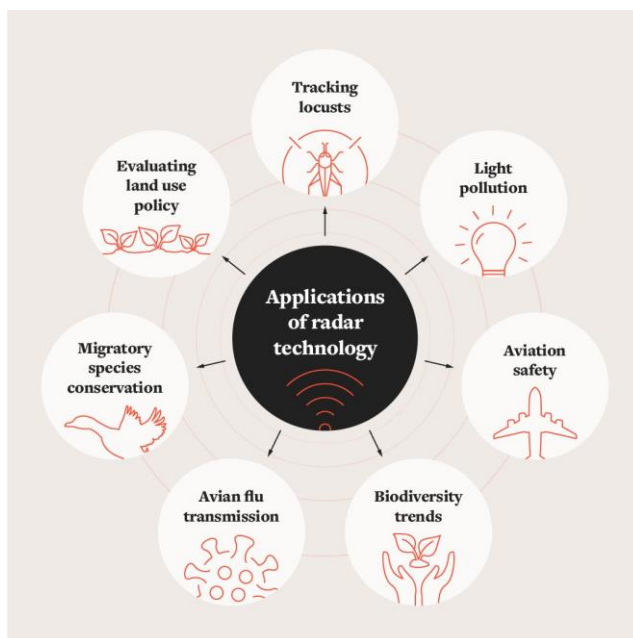


Рисунок 1.5 – Сфери застосування радарів

У зв'язку з різноманітністю поставлених завдань і широким спектром функцій, існує велика розмаїтість типів РЛС, які відрізняються за такими параметрами, як:

- сфера застосування (військові чи цивільні);
- призначення (виявлення, керування та спостереження, панорамні, бічного огляду, метеорологічні, цілевказівки та наведення, огляд обстановки);
- розташування (бортові, наземні та морські);
- тип дії (пасивні, активні з пасивною відповіддю (первинні), активні з активною відповіддю (вторинні) та моноімпульсні);
- діапазон частот (метровий, дециметровий, сантиметровий і міліметровий);
- кількість каналів (одно- та багатоканальні);
- методи і режими роботи (імпульсні, з безперервним випромінюванням, когерентні (імпульсно-доплерівські) та некогерентні);
- вид випромінюваного сигналу (простий або складний);
- параметри основних вузлів (передавача, приймача, антени і системи обробки прийнятих сигналів і т.д) [10].

1.2 Використання РЛС в Збройних силах України

Сучасні реалії такі, що збройний конфлікт в Україні діє вже протягом майже 10 років, і весь цей час обома сторонами конфлікту активно застосовувалося і застосовується озброєння повітряного типу: літаки, гелікоптери, ракети тощо. Через це активно використовуються радіолокаційні станції задля стеження за цими повітряними об'єктами, стеження за переміщенням військ, коригуванням артилерійського вогню (контрбатарейна РЛС), розпізнавання «свій-чужий» і тд.

Декілька поширених моделей РЛС в Збройних силах України радянського та сучасного зразка:

1. 1Л220У «Зоопарк-2» рисунок 1.6 — радіолокаційний артилерійський розвідувальний комплекс, призначений для ведення розвідки позицій вогневих засобів противника, розрахунку траєкторії снарядів і ракет, коригування вогню, стеження за повітряним простором, а також контролю за безпілотними літальними апаратами.



Рисунок 1.6 – РЛС 1Л220УК «Зоопарк-3»

Розробка комплексу 1Л220 "Зоопарк-2" розпочалася у 1981 році, і виконавцем цього проекту було НВК "Іскра" з міста Запоріжжя в УРСР. Процес розробки "Зоопарк-2" відбувався приблизно одночасно із розробкою його конкурента - комплексу 1Л219, який створювався у НДІ "Стріла" у місті Тула,

РРФСР. В порівнянні з попереднім комплексом 1Л219 "Зоопарк-1", український комплекс "Зоопарк-2" відрізнявся покращеною дальністю.

Після розпаду Радянського Союзу у 1992 році розробники опинилися в різних країнах і продовжили роботу незалежно один від одного. НВК "Іскра" завершила свою роботу, представивши систему 1Л220У "Зоопарк-2", де літера "У" в назві вказує на Україну. Цей комплекс був прийнятий на озброєння української армії у 2003 році.

23 квітня 2020 року КП "НВК „Іскра“" розпочало виробництво першого серійного контрбатареєного радару 1Л220УК "Зоопарк-3", який був прийнятий на озброєння Збройних Сил України в 2019 році [11]. Дальність розвідки/контролю:

- артилерії: 23/28 км;
- мінометів: 20/25 км;
- РСЗВ: 35-38/40-45 км;
- тактичних ракет: 55/80 км.

Сектор електронного огляду:

- за азимутом: $60^{\circ}(\pm 30^{\circ})$;
- за кутом місця: $40^{\circ} (\pm 25^{\circ}/-15^{\circ})$.

2. Станція наземної радіолокації міліметрового діапазону 112L1 "Борсук", що на рисунку 1.7, розроблена ВАТ "ХК „Укрспецтехніка“" і призначена для виявлення людей, наземних і надводних транспортних засобів.



Рисунок 1.7 – РЛС 112L1 "Борсук"

Ці радары типу "Борсук" використовуються в різноманітних сценаріях, таких як пошук порушників, терористів і диверсантів. Інформація про рух об'єктів відображається на екрані індикатора у вигляді амплітуди сигналу, що також передається через звуковий канал до навушників оператора.

Важливою особливістю цієї РЛС є використання передавача із дуже низькою потужністю випромінювання, що ускладнює її виявлення. Застосування кодової маніпуляції робить її стійкою до перешкод, при цьому не викликаючи їх для інших радіотехнічних засобів. РЛС може бути повністю розгорнута і готова до роботи менше, ніж за одну хвилину. Разом із акумулятором живлення вага "Борсука" становить 5-6 кг [12].

Параметри виявлення об'єктів:

- техніка: до 1600 метрів;
- людина: 600—800 метрів.

3. "Кольчуга" рисунок 1.8, представляє собою українську автоматизовану станцію радіотехнічної розвідки, яка діє в пасивному радіусі, що означає відсутність активних випромінювань і невиявлення іншими радіозасобами розвідки. Початково розроблена донецьким заводом "Топаз", виробництво було перенесене до НВК "Іскра" у Запоріжжі через російсько-українську війну в 2014 році.



Рисунок 1.8 – Радіолокатор «Кольчуга»

Презентована на виставці "Sofex-2000" в Йорданії у 2000 році, "Кольчуга-М" вперше зробила враження широкому загалу. У 2001 році взяла участь у Всесвітній виставці озброєнь в Абу-Дабі, а в тому ж році потрапила до Збройних Сил України.

Мобільна станція дальньої радіотехнічної розвідки "Кольчуга" має здатність виявляти та розпізнавати майже всі відомі активні радіотехнічні засоби, які встановлюються на різних об'єктах — наземних, морських і повітряних, включаючи ті, що базуються на "стелс-технологіях".

Основний принцип функціонування цієї пасивної, мобільної станції, що розташована на землі, полягає в "пасивній" радіолокації. Вона здатна виявляти та аналізувати електричні сигнали, що випромінюють різні об'єкти, незалежно від їхнього розташування — на землі, воді чи в повітрі. Наприклад, будь-який зенітно-ракетний комплекс, корабель чи літак-невидимка "Стелс" мають однаковий сигнал, який використовує "Кольчуга" для виявлення, залишаючись при цьому непомітною і не ризикуючи потрапити під вогонь засобів ураження. У комплект "Кольчуги" входять три розташовані на великій відстані одна від одної станції для повного визначення параметрів цілі, таких як відстань, висота, напрямок та швидкість [13].

Розташовані на значній відстані одна від одної, ці станції працюють синхронно і спільно визначають точні координати цілі. Отримані дані можуть бути передані засобом протиповітряної оборони. Застосування комплексу, що складається з трьох станцій, дозволяє визначати координати наземних і надводних цілей, а також визначати їхні маршрути руху на території до 600 км вглиб і до 1000 км вздовж фронту (для повітряних цілей на висоті 10 км - до 800 км). Це дає змогу забезпечити раннє попередження систем протиповітряної оборони країни. У складі станції існує п'ять антенних систем для метрового, дециметрового і сантиметрового діапазонів, що забезпечують чутливість радіо-тракту в широкій смузі панорамного огляду від -110 до -155 дБ/Вт, в залежності від частоти. 36-канальний приймач виявлення працює паралельно, що дозволяє швидко виявляти, аналізувати та класифікувати сигнали джерел радіовипромінювання без обмеження щільності вхідного потоку у всьому діапазоні частот від 130 до 18 000 МГц.

Станція виконує всі процеси виявлення та розпізнавання джерел радіовипромінювання повністю автоматично. У цьому процесі потужний бортовий комп'ютер відповідає за аналіз, числову обробку та розпізнавання виявлених цілей. Це відбувається шляхом порівняння їх параметрів з інформацією у банку даних, а результати виводяться на монітор з картою місцевості. Оснащені спеціальними перешкоджаючими селекторами, станції можуть виключати до 24 заважаючих сигналів з обробки, та селектори супроводження одночасно виділяють і супроводжують сигнали від 32 цілей.

Для виконання всіх основних завдань у стандартному режимі достатньо одного оператора (хоча у складі екіпажу можуть бути ще два, які забезпечують режим цілодобової роботи). Цей оператор керує роботою станції в діалоговому режимі за допомогою персонального комп'ютера. Станції «Кольчуга» монтується на автомобілях підвищеної прохідності (в українському варіанті - КраАЗи), обладнані автономними джерелами живлення, системами кондиціонування та вентиляції, що дозволяє їм працювати в діапазоні температур від -50 до +50 градусів за Цельсієм. [14]

4. "Мельхіор", на рисунку 1.9, представляє собою автоматизовану станцію радіотехнічної розвідки, що працює на основі принципів пасивного радару. Завдяки відсутності активних випромінювань, "Мельхіор" непомітний для радіоелектронної розвідки.

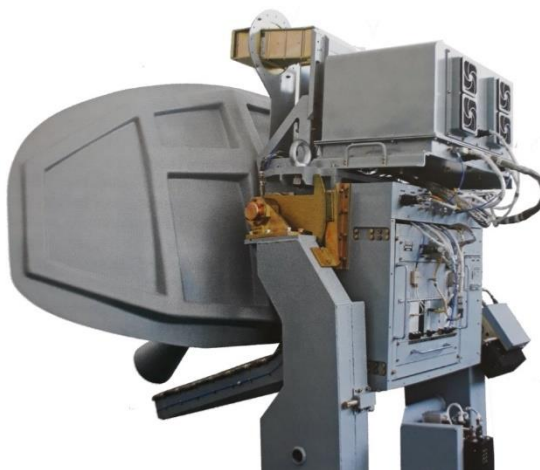


Рисунок 1.9 – РЛС «Мельхіор»

Розробку та виробництво станції "Мельхіор" виконали на підприємстві "Квант-Радіолокація" за короткий термін - 1,5 року.

Дотепер технології пасивної радіолокації в Україні асоціювались головним чином із відомою станцією "Кольчуга". Після захоплення Росією підприємства "Топаз" у Донецьку та його фактичного знищення, Україна не відновлювала виробництво "Кольчуг". Однак в області створення сучасних систем радіотехнічної розвідки у нас є великий потенціал для заповнення цієї прогалини. Розробники вважають, що станція "Мельхіор" представляє собою нове покоління таких систем і відповідає світовим стандартам в цій галузі, зокрема, рівню станцій від компанії "Thales".

На відміну від активних радарів, станція "Мельхіор" не випромінює електромагнітні імпульси для отримання відбитого від цілі сигналу. Замість цього "Мельхіор" визначає ціль за випромінюванням бортових радіо- та електротехнічних приладів, встановлених на цілі.

Максимальна дальність виявлення цілей у системі радіотехнічної розвідки "Мельхіор" становить до 450 км і суттєво перевищує радіогоризонт. Це досягається завдяки використанню ефектів суперрефракції, що полягає в відбитті радіохвиль від різних шарів атмосфери або, в певних умовах, огинанню радіохвиль приморською поверхнею.

Крім вражаючої дальності, "Мельхіор" може одночасно відстежувати до 200 цілей, що значно перевершує показник серійної "Кольчуги", яка здатна була відстежувати лише 20 цілей. Більше того, у той час як "Кольчуга" спрямовувалася переважно на повітряні цілі, "Мельхіор" може відстежувати об'єкти як на воді, так і в повітрі, включаючи ті, що розташовані на узбережжі.

Ще однією розробкою, яку впроваджено у "Мельхіорі", є широкий безшовний діапазон виявлення роботи джерел випромінювання від 1 до 18 ГГц. Це було важливим аспектом, оскільки деякі сучасні РЛС, встановлені на літаках, працюють у діапазоні 12-18 ГГц. [15]

5. Х1-М «Око», на рисунку 1.10 — українська радіолокаційна станція охорони периметру. Була представлена на виставці «Зброя та безпека 2017».

Призначена для виявлення (мало) рухомих наземних об'єктів на фоні місцевості; виявлення малорозмірних ЛА в приземному просторі; визначення координат цілей (азимуту та дальності), ЕПР, радіальної швидкості та ширини доплерівського спектру. Для визначення висоти повітряних цілей РЛС опціонально може оснащуватися додатковим модулем. Передбачена інтеграція радарних даних зі стандартними географічними інформаційними системами. РЛС може експлуатуватися як при розміщенні на стаціонарній точці, так і з мобільної платформи (автомобіль, причіп тощо). [16]



Рисунок 1.10 – РЛС X1-M «Око»

Технічні характеристики радіолокатора X1-M:

- інструментальна дальність, км 0,3...30;
- тип антени хвильоводно-щілинна;
- імпульсна потужність передавача, Вт до 30;
- сектор огляду по куту місця 30° ;
- азимутальний сектор огляду 360° ;
- максимальна швидкість сканування антени $90^\circ/\text{сек}$;
- точність вим. азимута цілі 1° ;
- точність вим. дальності, м 5;
- точність вим. радіальної швидкості, м/с 0,1;

- мінімальна вим. швидкість, м/с 0,2;
- максимальна швидкість цілей, м/с 60;
- система координат WGS-84 або РЛС;
- габарити, м 1,2х0,5х0,75;
- вага, кг до 65;
- споживана потужність, Вт 300.

6. Також Україна отримала деякі західні зразки РЛС в якості військової допомоги від її союзників. Одна з них це AN/TPQ-36, на рисунку 1.11.



Рисунок 1.11 – РЛС AN/TPQ-36

AN/TPQ-36 - це мобільна РЛС системи контрбатарейної боротьби, виготовлена компаніями Northrop Grumman та ThalesRaytheonSystems. Цей комплекс призначений для виявлення та відстеження артилерійських пострілів та ракетних пусків, забезпечуючи контрбатарейний вогонь.

AN/TPQ-36 має дистанційне управління радаром, що означає, що сама антена залишається нерухомою, хоча може бути вручну зміщена. Система також може взаємодіяти з власною артилерією для точного наведення при контрбатарейній боротьбі або коригування вогню.

Ця установка може виявляти місця розташування мінометів, артилерії, ракетно-артилерійських систем та пускових установок. Вона може працювати з 10 видами зброї одночасно та виявляти цілі під час першого проходу.

AN/TPQ-36 володіє вражаючою максимальною дальністю в 24 кілометри, що дозволяє відстежувати ракети на цій відстані, і артилерійські снаряди відстежуються на відстані 18 кілометрів. Сектор азимуту складає 90° , а антена охоплює діапазон X і включає 32 частоти.

Електричні характеристики включають 115/200 В змінного струму, 400 Гц, 3 фази, 8 кВт. Пік переданої потужності становить 23 кВт/год. Система здатна відстежувати одночасно до 99 цілей, має режим польових навчань та цифровий інтерфейс передачі даних. [17]

7. Ще один радіолокатор на озброєнні України, це AN/MPQ-64 Sentinel, рисунок 1.12.



Рисунок 1.12 – РЛС AN/MPQ-64 Sentinel

AN/MPQ-64 Sentinel — це імпульсно-доплерівська тривимірна радіолокаційна система X-діапазону з електронним керуванням, яка використовується для попередження та наведення зброї протиповітряної оборони малої дальності (SHORAD) до місць розташування ворожих цілей, які наближаються до їхніх сил на передовій. Зараз його виробляє Raytheon Missiles & Defense.

Антена використовує фазово-частотну технологію електронного сканування, утворюючи гострі 3D-промені олівця, що охоплюють великий обсяг відеоспостереження та треків. Він використовує обертову платформу з високою

швидкістю сканування (30 об/хв), щоб забезпечити охоплення на 360 градусів. Радар розроблений з високою стійкістю до радіоелектронної протидії (ECM) і протирадіаційних ракет (ARM).

Система автоматично знаходить, відстежує, класифікує, ідентифікує та повідомляє про високо- та низьковисотні цілі, включаючи крилаті ракети, безпілотні літальні апарати, гвинтокрили та літаки. Ідентифікації дружнього літака допомагає ідентифікаційний опитування свого чи ворога (IFF).

Систему можна розгорнути за допомогою HMMWV, оснащеного бортовим генератором електроенергії 10 кВт 400 Гц 115/200 В постійного струму. Він може працювати автономно і зв'язуватися з центром управління вогнем через широкосмуговий оптоволоконний зв'язок. Він також може поширювати свої дані через радіомережу SINCGARS. [18]

Висновки до розділу 1

В даному розділі розглянуто актуальність поставленої задачі. Було розглянуто проаналізовану літературу, що містить в собі інформацію про роботу радіолокаторів та приведено необхідні для подальшої розробки програми дані про принципи роботи радіолокаторів. Також представлено деякі радари на озброєнні Збройних сил України, здійснено короткий опис їх розробки та принципів роботи, а також зазначено їх характеристики. Сформульовано підґрунтя інформації, яка необхідна для реалізації системи ідентифікації даних із радіолокаторів.

2 МЕТОДИ ОТРИМАННЯ Й ОБРОБКИ ДАНИХ З РАДІОЛОКАТОРІВ

В даному розділі будуть описані вже існуючі рішення реалізованих систем для обробки сигналів з радарів, а також надані всі математичні дані й формули, які необхідні для реалізації MVP подібного проекту. Наведено табличні показники, котрі використовувалися під час розробки й налаштування програмного забезпечення. Наведено всі програмні засоби, які були використані під час розробки програмного продукту.

2.1 Аналогічна українська система безпеки – Delta

Наразі в Україні є один аналог подібного програмного забезпечення, котрий використовують Сили безпеки і оборони України — система «Дельта» рисунок 2.1. Ця система побудована задля ведення мережево-центричної війни за стандартами НАТО. [19]

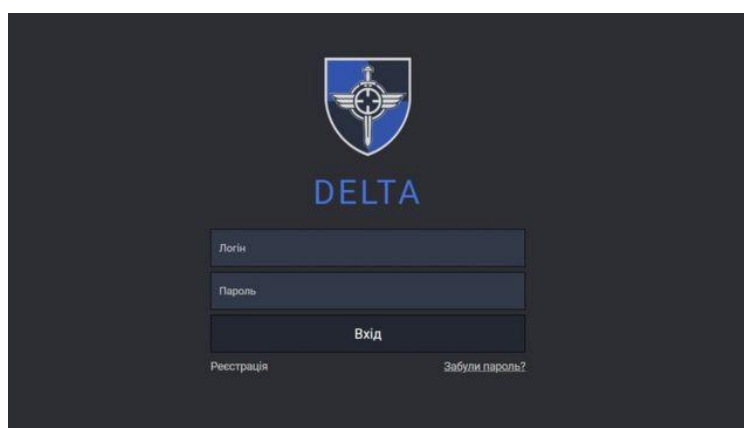


Рисунок 2.1 – Вхід в систему «Дельта»

Управління військами на основі єдиної автоматизованої системи, доступ до якої мають всі командири на полі бою, - це сучасний підхід НАТО. Головна ідея -

бачити поле бою в режимі реального часу, обмінюватися інформацією в межах підрозділу, бригади (якщо говоримо про ЗСУ), угруповання, а при потребі - і з союзниками.

За таких умов командир відділення чи роти в окопах просто зараз бачить, що відбувається у нього на фронті, яка ситуація на флангах, де йде бій, куди потрібна підтримка, а де ситуація більш стабільна. А комбриг розуміє ситуацію на довірених йому ділянці фронту у 20-30 км і суміжних з ним територіях. Це дозволяє командирам, які беруть безпосередню участь у боях, об'єктивно усвідомлювати, наскільки критичною є ситуація у їхніх підрозділах, де потрібна підтримка, а де є потенціал для контрнаступу.

Якщо говорити дуже спрощено, то такі системи багато в чому нагадують комп'ютерні ігри-стратегії, де користувач «зверху» бачить арену бойових дій, рисунок 2.2, розвиток ситуації на ній та може переміщувати підрозділи відповідно до ходу битви.

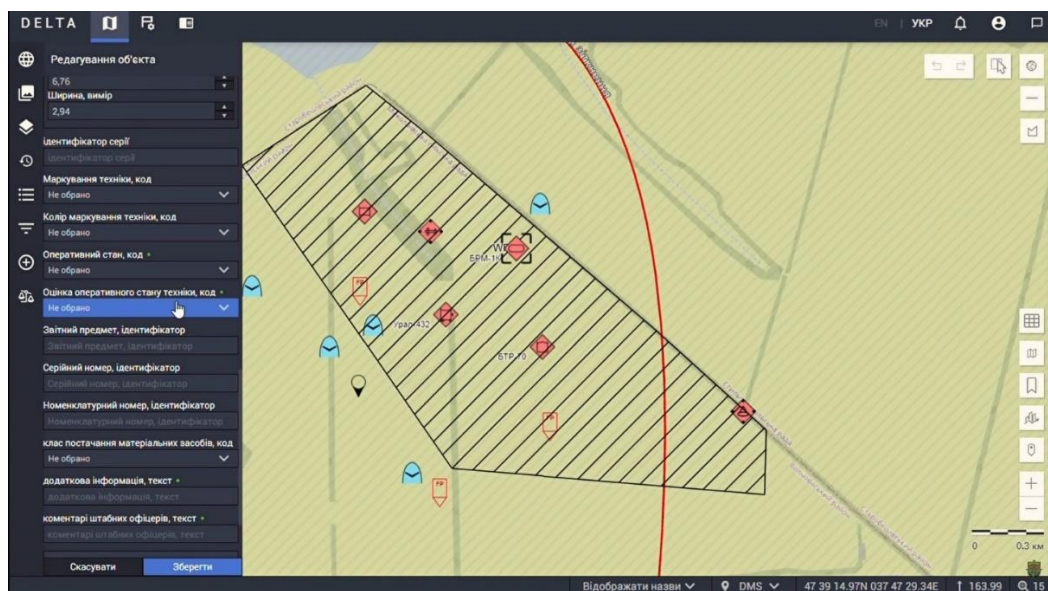


Рисунок 2.2 – Інтерфейс системи «Дельта»

В Україні таку систему намагалися втілити з 2015 року. Тоді серед кількох можливих рішень міноборони обрало систему «Дзвін-АС». Утім, пізніше виявилось, що "Дзвін" має проблеми з безпекою комунікацій. А також з

імплементацією з союзниками загалом. Тому потім повернулися до проєкту «Дельта», який весь цей час розроблявся паралельно волонтерами з групи «Аеророзвідка».

На сайті ГУР опублікований такий відгук про «Дельту»: під час виконання завдань в зоні проведення ООС добре себе зарекомендувала система відображення оперативної обстановки «Дельта», яка дозволяє в масштабі реального часу відслідковувати положення своїх військ, а також оперативно обліковувати виявлені об'єкти противника для їх подальшого вогневого ураження.

Система «Дельта» з'явилася ще у 2017 році та була випробувана на міжнародних навчання Sea Breeze та Rapid Trident. І отримала можливість вести спільні бойові дії з арміями країн НАТО, демонструвати свої позиції, бачити позиції союзників, обмінюватися цілями за стандартами НАТО. І успішна оборона української столиці, великою мірою була забезпечена «Дельтою».

Система «Дельта» інтегрує інформацію про розташування ворожих сил та ресурсів, надаючи змогу в реальному часі відстежувати місцезнаходження військ противника і швидко враховувати виявлені об'єкти для подальшого призначення їм ударів. Система об'єднує дані про противника на цифровій карті, здобуті з різних джерел, таких як супутникові знімки, радары, сенсори, GPS-трекери та радіоперехоплення. Користувачі можуть бачити події на суходолі, воді, повітрі, космосі та кіберпросторі через будь-який пристрій, включаючи ноутбуки, планшети та мобільні телефони.

Систему використовують для планування військових операцій та координації дій між підрозділами за допомогою захищеного месенджера ELEMENT, який є складовою «Дельти». Платформа та сервіси «Дельта» відповідають стандартам НАТО, підтримують специфікацію MIP (Multilateral Interoperability Programme) [5] і сприяють мережевому взаємодії. Система сумісна з аналогічними рішеннями, які використовуються арміями країн-членів НАТО. За словами співзасновника «Аеророзвідки» Ярослава Гончара, всі сучасні західні технології, які надають союзники України для протидії російській агресії, автоматично інтегруються в систему «Дельта».

На 12 березня 2022 року до системи був доданий чат-бот «Ворог», розроблений Міністерством цифрової трансформації. За допомогою цього чат-бота українці могли надсилати фотографії та відеозаписи з місцезнаходженням російських військ і техніки. Після обробки ці дані потрапляють в систему «Дельта». За дев'ять місяців роботи чат-бота ним скористалися 433 тисячі українців, згідно з повідомленням Міністерства цифрової трансформації. Пізніше був інтегрований чат-бот «STOP Russian War», розроблений Службою безпеки України.

Самі розробники «Дельти» визнають, що система вразлива. Російські хакери намагалися проникнути в неї від самого початку, але пробити ІТ-систему «в лоб» не вдалося, тому вони вдалися до фішингу - розсилки фейкових листів для людей з доступом, які б спонукали їх розкрити логіни та паролі через підроблені сторінки. І це дало свій результат. Двоє користувачів перейшли за цими посиланнями. Аккаунт одного з цих користувачів заблокували одразу, але другий зміг пробути в системі 12 хвилин. Люди, які зайшли з-під його профілю, змогли побачити все, що дозволяв доступ на рівні того аккаунту. Це було під час наступу ЗСУ на Харківщині восени. За 12 хвилин цей аккаунт було ідентифіковано і видалено. І наразі інформація в системі була змінена.

Утім, загроза зламу «Дельта» залишається, і команда підтримки слідкує за аномальною активністю. Крім того, кожен користувач має проходити верифікацію - перевіряють, чи справді ця людина служить в тому чи іншому підрозділі.

Українська «Дельта», як стверджують українські військові, вже змогла показати себе в реальних бойових діях і тому отримала довіру від бійців та командирів, які перебувають безпосередньо на полі бою.

Перевагою «Дельта» є те, що вона використовує хмарні технології і вже не залежить від конкретних серверів у конкретній локації. Тобто, їх не можна зруйнувати серверне приміщення і так знищити інформацію - вона вже в мережі і доступ до неї залишиться в будь-якому разі. І головне в системі - онлайн-мапа.

За словами розробників, користувачів системи "Дельта" в українській армії вже кілька десятків тисяч. Зокрема повністю інтегровані в неї Сили спецоперацій

та Військово-морські сили ЗСУ, а також все більше переходять підрозділи механізованих бригад, ДШВ та тероборони.

Але конкретно ця система є загальною системою керування військами, а не налаштована суто на роботу із сигналами радіолокаційних станцій у війську, тому можна зробити висновок, що розроблене програмне забезпечення не має аналогів серед представлених на ринку в Україні. [20]

2.2 Методика збору інформації із радару, за допомогою плати Arduino

Із появою Arduino стала реальною можливість реалізувати навіть в себе вдома повноцінний радар, що буде працювати, використовуючи ультразвукові хвилі. Тобто в результаті буде отримано аналог сонару.

2.2.1 Про мікроконтролер Arduino

Мікроплата Arduino - це відкритий, гнучкий набір апаратних та програмних інструментів для створення простих до складних електронних проектів. Ця платформа дозволяє ентузіастам та професіоналам створювати різноманітні електронні системи без великих зусиль.

Основою Arduino є мікроконтролер, який може бути легко програмований за допомогою мови програмування C++. Цей мікроконтролер зазвичай вбудований у малий мікрокомп'ютер, який має введення-виведення (наприклад, порти для підключення сенсорів, світлодіодів, дисплеїв тощо) та може взаємодіяти з різними датчиками та пристроями.

Плата Arduino поширилася в світі технічних хобі та електроніки завдяки своїй відкритій архітектурі, простоті використання та величезному співтовариству користувачів та розробників. Вона забезпечує можливості для розробки різноманітних проектів, таких як домашні автоматизовані системи, робототехніка, інтерактивні мистецькі об'єкти та інше. [21]

2.2.2 Конструкція радару на платах Arduino

Щоб розробити подібний радар, необхідне таке обладнання:

- плата Arduino Uno;
- сервомотори SM-S4303R;
- джерело живлення 5 В DC;
- модуль HC-SR04 – ультразвуковий датчик відстані;
- металевий конструктор – для механічної частини проекту;
- макетна плата та провідники;
- комп'ютер з USB кабелем для передачі даних.

Є дуже простий метод обміну даними між Arduino та персональним комп'ютером через USB кабель. Підключення реалізується по серійному порту, який дає можливість передавати та приймати дані. У нашому випадку Arduino передає дані на персональний комп'ютер, на якому запущено програму візуалізації показань радіолокації.

Основна частина програми для візуалізації даних із сонара написана мовою програмування Scala, рисунок 2.3. На Java написано невелику частину коду, яка відповідає за обмін даними з Arduino. У програмі доданий циклічний буфер для зберігання даних, які отримані Arduino без ризику переповнення буфера. У скетчі Arduino використовується функція `serialEvent` для обробки даних, що надходять. Цей метод набагато зручніший, ніж використовувати серійний порт для невеликого пакету даних на кожній ітерації циклу `loop`.

Програма запускає та зупиняє роботу ехолотатора, посилаючи рядки “CONT” та “STOP” на плату Arduino. Як тільки Arduino отримує рядок “CONT”, надсилається сигнал на сервомотори для початку обертання.

Коли серводвигун працює, Arduino подає команди на ультразвуковий датчик відстані, який випромінює ультразвукові хвилі та розраховує час до прийому відбитої хвилі. Для зняття показань з ультразвукового датчика відстані використовується бібліотека `NewPing library`.

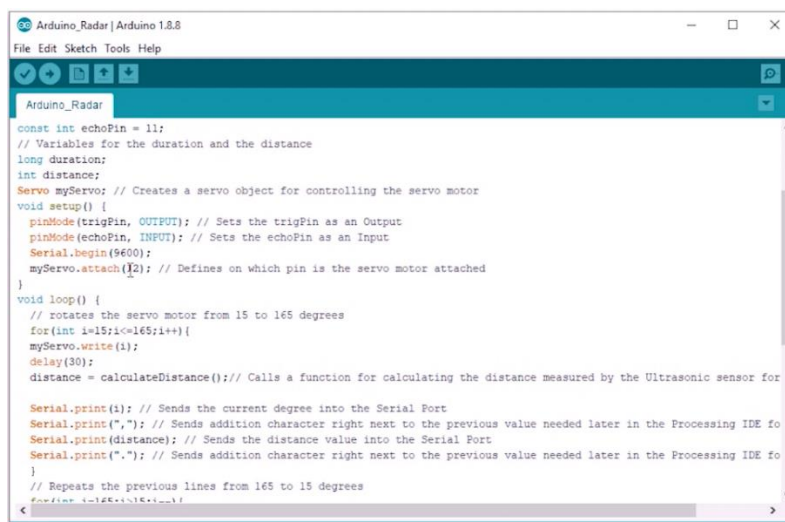


Рисунок 2.3 – Частина коду РЛС на основі Arduino

Процес вимірювання можна побачити на осцилограмі на рисунку 2.4. Варто звернути увагу на рівні напруги high і low на виході сенсора.

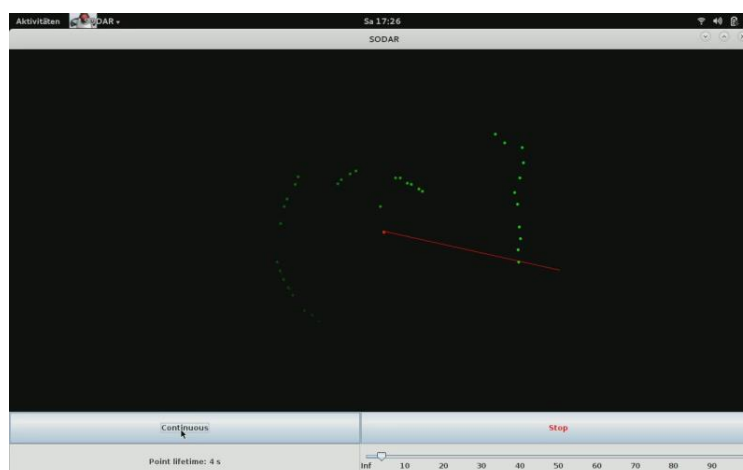


Рисунок 2.4 – Програмна демонстрація роботи розробленого радару

Щоразу, коли Arduino набуває значення відстані, контролер перевіряє час, що минув з моменту останньої зміни напрямку обертання ротора Arduino двигуна і конвертує цей час у кут повороту. Цей кут характеризує кут повороту ехолотатора в системі полярної координат. Після цього кут і відстань надсилаються на персональний комп'ютер по серійному порту. Програма на ПК зчитує отриману

пару даних і поміщає в масив, де також фіксується поточний час. Завдяки цьому можна визначити і час, коли було проведено вимір. На підставі часу в програмі реалізовано ефект загасання, який характерний для старих аналогових моніторів на радарах. [22]

2.2.3 Особливості роботи радару на базі Arduino

Алгоритм функціонує наступним чином: спочатку система сканує простір і фіксує кут та відстань до об'єкта під час першого проходу радару. Під час наступних проходів алгоритм визначає різницю для кожного кута, що дозволяє виявляти нові об'єкти. Для відстеження об'єкта розраховується кількість точок, які знаходяться поруч, у графіку це представлено областю, яку сканує локаційний пристрій.

Для визначення цілі враховується область, що перевищує певний розмір, фільтруючи шуми та перешкоди. Пристрій здатний розпізнавати велику область, враховуючи кут початку та кут кінця. Після визначення середини області радар спрямовується в цю область, фіксуючи положення об'єкта як точку спостереження. Система неперервно вимірює відстань від радару до об'єкта, і якщо точка спостереження змінюється, система повертається в режим пошуку цілі.

Для вдосконалення можливостей пристрою в разі втрати цілі, може бути доданий другий випромінювач зондуючого сигналу. Це дозволяє системі визначати напрямок руху цілі та продовжувати спостереження без втрати часу, підвищуючи швидкодію пристрою вдвічі.

Використання ультразвуку в проекті замість електромагнітних коливань полегшує реалізацію макету, зберігаючи його функціональність та швидкодію. Ультразвук, як акустичні коливання, є оптимальним вибором для електронних систем виявлення та супроводження об'єктів, забезпечуючи ефективність та вартісні переваги.

Ультразвукові датчики накладають певні обмеження через принцип своєї роботи, рисунок . Замість того, щоб виявляти всі відбиті сигнали, вимірюється час повернення звукової хвилі при її відображенні від першої перешкоди. Тобто ми не

дізнаємося відстань до об'єкта, який розташований на більшій відстані від датчика, ніж перший у межах його кута огляду. Наприклад, у другому випадку на рисунку 2.5, ми отримаємо відстань до червоного трикутника, а зірочка та квадрат вислизнуть від нашого саморобного радара.

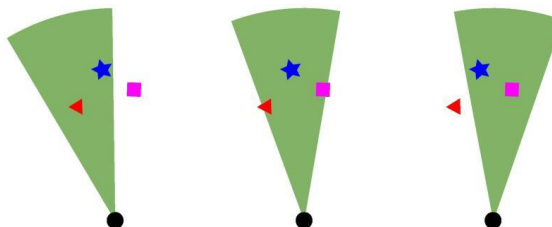


Рисунок 2.5 – Демонстрація принципів роботи радара

Чорна точка – це наш ультразвуковий датчик відстані, зелена площа – кут огляду при різних кутах повороту приводу. Синя зірочка – це об'єкт, який ми ніколи не зможемо побачити, оскільки він завжди знаходиться далі ніж червоний трикутник або рожевий квадрат.

Таку проблему можна вирішити за допомогою відеокамери. Під час використання відеокамери з Arduino можна скористатися функцією розподілу точки (ФРТ) (point spread function). Камери не є ідеальними. Завжди певний рівень шуму, який залежить від відстані до об'єкта. ФРТ – це спосіб описати ці шуми. Уявіть, що ви робите малюнок маленької чорної точки на білому тлі. На рисунку 2.6 нижче зліва показана маленька чорна точка на білому тлі. Справа – фото тієї ж точки, яка зроблена камерою з великим рівнем шумів.

Якщо фотографія робиться поганою камерою, точка виглядатиме як показано праворуч. По-перше, її розміри здаються більшими, а межі розмиті.

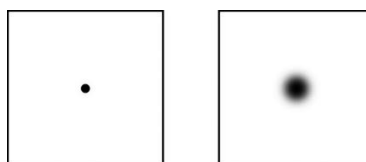


Рисунок 2.6 – Функція розподілу точки

Можна провести аналогію, що ФРТ – це зображення дуже маленької (менше одного пікселя) точки. Використовуючи найкрупішу камеру, ви ніколи не отримаєте зображення, ідентичне оригіналу. Тобто шуми будуть завжди. Якщо ми знаємо *spread* функції нашої камери, можна коригувати зображення за допомогою методу зворотного згортки. Це дозволяє отримати чіткіше зображення після його обробки. До речі, ця методика використовувалася перші три роки на Hubble Space Telescope's. Вимірюючи ФРТ телескопа, астрономи могли коригувати зображення Землі (через три роки було внесено деякі корективи у роботу телескопа і необхідність у такій обробці відпала).

На жаль, подібна методика абсолютно не застосовується у разі використання ультразвукового датчика відстані, оскільки є концепт під назвою суперпозиція, рисунок 2.7.

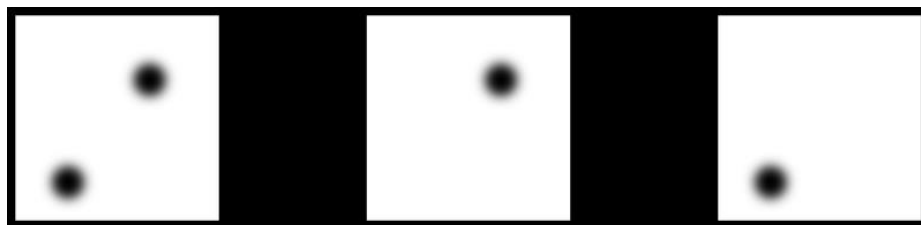


Рисунок 2.7 – Принцип суперпозиції

На знімку буде представлено дві точки, обидві трохи розмиті, проте вони абсолютно незалежні одна від одної. З іншого боку, фотографія є зображенням однієї точки, доповненою зображенням іншої точки. Іншими словами, об'єднане зображення фотографій двох точок називається лінійною суперпозицією двох індивідуальних фотографій.

При розгляді ультразвукового датчика відстані зрозуміло, що формування зображення не може бути поясненим принципом суперпозиції. Приклад вище демонструє, що "фотографії" блакитної зірочки, рожевого трикутника та рожевого квадрата не підпадають під принципи суперпозиції фотографій трьох окремих об'єктів. Замість цього датчик відстані виявляє лише найближчий об'єкт і ігнорує

всі інші. Систему такого типу неможливо описати за допомогою принципів лінійної суперпозиції та відновити об'єкти, яких бракує, шляхом подальшої обробки. Отже, використання ультразвукового датчика, Arduino та наданого програмного забезпечення в якості повноцінного 3D сканера в медицині не є можливим.

Таким чином можна реалізувати простий ультразвуковий радар, за допомогою мікро-плати Arduino. Даний радар має змогу ідентифікувати об'єкти на невеликих відстанях, тобто в перспективі може бути використаний як радіолокаційна станція охорони периметру. Інформація по даному проекту дає змогу зрозуміти принципи роботи повноцінного радару, що працює за допомогою радіохвиль. [22]

2.3 Математичні методи обрахунку даних з РЛС

Звісно що для реалізації подібної магістерської роботи необхідне проведення розрахунків у програмі й використання реальних даних про повітряні об'єкти, для можливості запровадити в майбутньому дану систему до практичного застосування. Основні обрахунки ведуться по таким параметрам повітряного об'єкта:

- ефективна площа розсіювання (ЕПР);
- радіальна швидкість;
- відстань;
- азимут.

2.3.1 Розрахунок ефективної площі розсіювання об'єкту

Здатність радіолокаційної цілі (РЛЦ) відбивати електромагнітну енергію, що падає на неї, характеризується ефективною площею розсіювання (σ). Одиницею виміру цього параметра є квадратний метр (м^2). Далі використовуватимемо термін ефективна площа розсіювання (ЕПР).

Під ЕПР розуміють площу еквівалентного ізотропного розсіювача, який, будучи поміщений у точку знаходження цілі, створює на розкриві приймальної антени таку ж густину потоку потужності, як і реальна ціль, рисунок 2.8. Таким чином, ЕПР є абстракцією, моделлю, що дає, проте можливість оцінити відбивну здатність РЛЦ. Очевидно, що ЕПР може використовуватися і для оцінки помітності радіолокації цілі. [23]

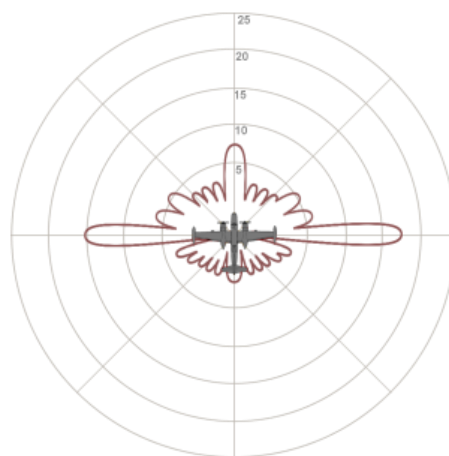


Рисунок 2.8 – Демонстрація розсіювання радіохвиль об'єктом

На величину ЕПР впливає безліч факторів, до основних з яких належать такі:

- розміри та геометрична форма цілі;
- ракурс спостереження цілі, що визначає, яка саме частина поверхні цілі опромінюється зондувальною електромагнітною хвилею (ЕМВ);
- робоча частота радіолокатора, а точніше, співвідношення між довжиною хвилі локатора та характерними розмірами цілі;
- електричні властивості матеріалу, з якого виконано конструкцію цілі.

Вплив перерахованих факторів є комплексним і тому враховувати їх потрібно також разом.

Аналітичні висловлювання до розрахунку значень ЕПР можуть бути отримані лише обмеженого набору цілей, мають просту форму поверхні. Більшість РЛЦ мають складну геометричну форму поверхні та для визначення їх ЕПР

застосовуються натурні вимірювання, а також методи фізичного чи математичного моделювання.

Стандартна формула (1) розрахунку ЕПР цілі наступна:

$$\sigma = \frac{4\pi r^2 * S_r}{S_t} \quad (1)$$

де r – радіус еквівалентного розсіювача,

S_r – щільність потоку потужності падаючої хвилі в точці знаходження мети,

S_t – щільність потоку потужності розсіяної хвилі біля антени радіолокатора.

Геометричні розміри РЛЦ більшості типів не перевищують розмірів імпульсного об'єму радіолокатора, призначеного для виявлення. Цілі, що мають такі розміри, називають точковими. ЕПР таких цілей визначається взаємодією ЕМВ, відбитих від про «блискучих» точок. «Блискучими» точками називають елементи поверхні мети, які за заданих умов спостереження (довжина хвилі радіолокатора, ракурс зондування) роблять найбільший внесок у розсіяне об'єктом поле, а значить і в ЕПР. Залежно від взаємного розташування «блискучих» крапок, і навіть напрями спостереження, відбиті ними хвилі може мати різні фазові співвідношення: від синфазного (тоді інтенсивність результуючого відбиття зростає) до протифазного (інтенсивність відбиття зменшується). Саме цей ефект визначає осцилюючий характер ЕПР залежно від ракурсу спостереження, причому кругова діаграма ЕПР має порізаний характер.

Слід зазначити, що швидкість осциляції ЕПР залежно від кута спостереження визначається співвідношенням між довжиною хвилі радіолокатора і характерними розмірами мети: що менше довжина хвилі проти розмірами мети, то сильніше осциляція ЕПР.

Враховуючи значні коливання величини ЕПР, у деяких випадках виявляється зручним представляти її значення в логарифмічному масштабі, наприклад, децибелах (дБ) щодо одиничної площі (1 м^2).

Далі буде наведено таблиці 2.1, в якій представлено використані у роботі значення ЕПР різних об'єктів [24]:

Таблиця 2.1 – ЕПР повітряних об'єктів

Об'єкт	ЕПР (м ²)
Дипольний відбивач	20
Винищувач	3-12
Фронтний бомбардувальник	7-10
Стратегічний бомбардувальник	13-20
Транспортний літак	40-70
Крилата ракета	0.1
ОТРК	0.15-1.6
Гелікоптер	0.5-2

Таким чином можемо мати усереднені дані про ефективну площу розсіювання різних повітряних об'єктів і застосовувати їх у програмі.

2.3.2 Розрахунок радіальної швидкості об'єкту

Рух об'єкта із заданою швидкістю характеризується напрямом, отже, його (рух) можна описати вектором швидкості, рисунок 2.9.

По відношенню до напрямку спостереження мети вектор швидкості цілі може бути розкладений на дві складові:

- вектор, спрямований уздовж лінії візування цілі, що називається радіальною швидкістю цілі (може бути спрямований як до точки спостереження, так і в протилежному напрямку);
- вектор, перпендикулярний лінії візування цілі, званий тангенціальною швидкістю або тангенціальної складової вектора швидкості цілі.

Діагональ прямокутника, побудованого на цих двох векторах, як на сторонах, і, отже, що представляє собою їхню суму, є повним вектором швидкості цілі або вектором швидкості цілі.

Повний вектор швидкості цілі показано на рисунку стрілкою зеленого кольору. Довжина вектора швидкості цілі дорівнює значенню швидкості цілі (чим

довше вектор, тим швидше рухається ціль), а його напрям збігається з напрямом руху (курсом) цілі.

Все, що «бачить» радіолокатор у конкретний момент часу, – це частина вектора швидкості цілі, спрямована вздовж радіусу, проведеного від антени до цілі (на рисунку 2.9 ця складова зображена у вигляді червоної стрілки). У загальному випадку дана складова не відповідає реальному курсу руху цілі, а описує тільки ту його (руху) частину, яка спрямована у бік радіолокатора або від нього (залежно від того, чи наближається чи видаляється ціль).

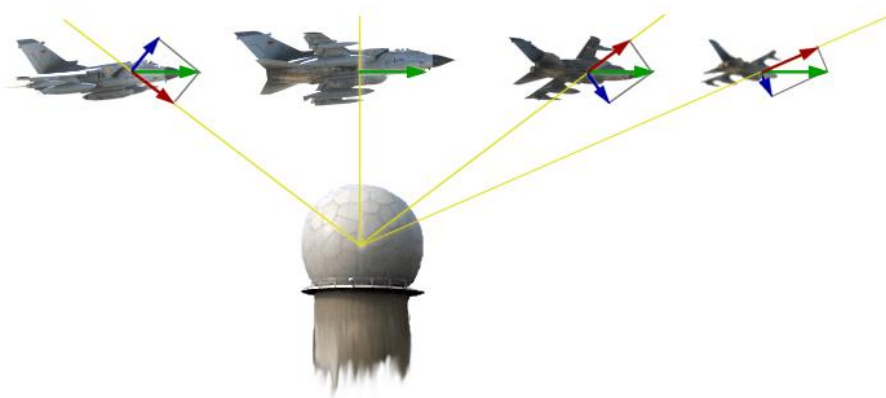


Рисунок 2.9 – Визначення радіальної швидкості повітряного об'єкта

Радіальна швидкість цілі – це складова повного вектора швидкості цілі, яка спрямована до радіолокатора або від нього.

Перпендикулярно вектору радіальної швидкості спрямований вектор тангенціальної швидкості (синя стрілка на рисунку 2.9). Розмір цього вектора впливає функціонування системи управління антеною в радіолокаторах супроводу цілі.

Коли літальний апарат пролітає повз радіолокатор по прямій лінії, тобто з постійним курсом, то на траєкторії його руху є точка, при знаходженні його в якій радіальна швидкість дорівнюватиме нулю. При цьому тангенціальна складова співпадатиме з повним вектором швидкості цілі. Як відомо, ефект Доплера виникає лише тоді, коли радіальна швидкість цілі не дорівнює нулю. Тому в момент знаходження літального апарату в цій точці його швидкість не може бути виміряна

радіолокатором і, отже, він не може бути відмінний від пасивної перешкоди, породженої нерухомим об'єктом.

При подальшому русі літака по прямій після проходження зазначеної точки радіальна швидкість змінює свій напрямок на протилежний і буде спрямована у бік радіолокатора. При цьому також виникає ефект Доплера і частота відбитого сигналу відрізнятиметься від несучої частоти сигналу зонduючого на величину, пропорційну величині радіальної швидкості. Різниця буде тільки в тому, що коли радіальна швидкість спрямована до радіолокатора (ціль наближається), то частота ехо-сигналу збільшується на величину доплерівської складової, а коли радіальна швидкість спрямована від радіолокатора (ціль віддаляється), то частота ехо-сигналу зменшується на цю ж величину.

Якщо ж ціль рухається навколо позиції радіолокатора по колу з постійним радіусом, то її радіальна швидкість у всіх точках траєкторії дорівнюватиме нулю. У такому разі ефект Доплера не виникає і доплерівська частота дорівнюватиме нулю. Ехо-сигнал такої цілі буде пригнічений у фільтрі системи селекції цілей, що рухаються, як пасивна перешкода і на індикаторі радіолокатора така ціль відобразатися не буде. [25]

Виявлення повітряних цілей у типових відбувається в такий спосіб. Станція має антену, що обертається в горизонтальній площині з частотою $\Omega_{об}$. Конструкція антени така, що характеристика спрямованості має велику ширину у вертикальній площині та малу горизонтальній площині. Передавач станції радіолокації (РЛС) формує послідовність високочастотних імпульсів, яка за допомогою антени випромінюється в простір. При контакті характеристики спрямованості антени (ХСА) із ціллю з'являється відбитий повітряною ціллю сигнал, який приймається цією ж антеною, обробляється приймальною системою РЛС та направляється до інших систем РЛС. З появою позначки від цілі на екрані індикатора робиться висновок про ціль. Азимут цілі приймається рівним азимуту антени в момент прийому максимального відбитого сигналу, а похила дальність до цілі визначається за часом затримки відбитого імпульсу щодо випромінюваного. При подальшому обертанні антени її промінь, відповідно і ХСА залишає місце розташування цілі,

тому відбитий сигнал зникає. Отже, інформація від цілі як і відмітка від цілі на індикаторі зникає і з'являється тільки через повний оборот антени в момент наступного опромінення повітряної мети. Час, протягом якого інформація від цілі відсутня, визначається періодом обертання $T_{об}$ антени (2):

$$T_{об} = \frac{2\pi}{\Omega_{об}} \quad (2)$$

де $\Omega_{об}$ – кутова швидкість обертання антени по азимуту (одиниці обертів за секунду).

При стеженні за повітряною ціллю по дальності виникає помилка, яка визначається радіальною швидкістю V_p польоту цілі, формула (3)

$$\delta D = V_p t \quad (3)$$

де t – час від моменту опромінення до поточного моменту часу t .

Таку помилку називають динамічною помилкою стеження чи відстеження за дальністю (помилкою виміру дальності). Для зменшення або повного усунення динамічної помилки необхідно якомога точніше вимірювати радіальну швидкість цілі та враховувати її при вимірі дальності у наступні контрольні моменти часу t_k за формулою (4):

$$D_{ц}(t_k) = D_{обл} + V_p t_k \quad (4)$$

де $D_{обл}$ – дальність, виміряна РЛС у момент опромінення мети (контакту ХСА з метою).

Таким чином, при підстеженні за повітряною ціллю за допомогою радару актуальним завданням є вимірювання її радіальної швидкості, яка за певних умов не може бути імпульсною РЛС обчислена правильно, що пов'язано з неоднозначністю вимірювань в імпульсних радіолокаційних системах. [26]

Далі буде наведено таблиці 2.2, в якій представлено використані у роботі значення швидкостей різних об'єктів:

Таблиця 2.2 – Швидкості повітряних об'єктів

Тип об'єкта	Швидкість (м/с)
Дипольний відбивач	0-100

Кінець таблиці 2.2

Винищувач	416-694
Фронтовий бомбардувальник	250-416
Стратегічний бомбардувальник	222-305
Транспортний літак	139-250
Крилата ракета	180-240
ОТРК	278-556
Гелікоптер	42-111

Ці дані швидкостей повітряних об'єктів також необхідні для подальшої реалізації програмного забезпечення.

2.3.3 Розрахунок відстані до об'єкту

Радіолокатор випромінює в простір короткий радіоімпульс дуже високої потужності, званий зондуючим імпульсом. Зондуючий імпульс фокусується у певному напрямку за допомогою спрямованої антени та поширюється у просторі зі швидкістю світла.

Якщо на шляху поширення зондуючого імпульсу виникає перешкода, наприклад, літак, то частина енергії імпульсу буде відбита у всіх напрямках. Деяка, дуже мала її частина буде також відбита у зворотному напрямку, тобто у напрямку антени радіолокатора.

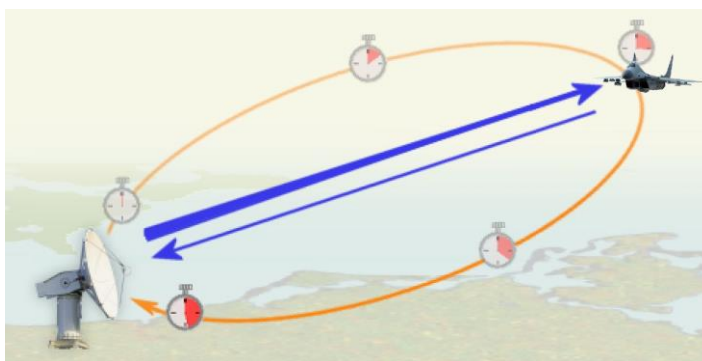


Рисунок 2.10 – Визначення відстані до цілі радаром

Тут вона може бути прийнята та використана для подальшої обробки. Дальність до цілі може бути виміряна за допомогою звичайного осцилографа. Принцип такого виміру пояснюється на рисунку 2.10. Крапка на осцилограмі, що світиться, переміщається синхронно з зондувальним імпульсом, залишаючи слід. Рух точки починається одночасно з випромінюванням зондувального імпульсу в простір. У той момент, коли антена прийме відбитий від цілі сигнал на екрані осцилографа з'являється імпульс. Тимчасова затримка між двома імпульсами, зондуючим та прийнятим, пропорційна дальності до цілі.

Так як поширення радіохвиль відбувається з постійною відомою швидкістю (рівною швидкості світла, c_0), то дальність може бути визначена за часом затримки прийнятого сигналу щодо зондуючого. Виміряна дальність від радіолокатора до цілі називається похилою дальністю цілі. Похила дальність цілі - це відстань по прямій (по лінії візування) від антени радіолокатора до об'єкта, що опромінюється. Тоді як горизонтальна дальність (проекція похилої дальності на земну поверхню), для обчислення вимагає інформації про висоту польоту чи куті місця цілі. Так як при виявленні цілі електромагнітна хвиля проходить шлях до цілі і назад, час проходження сигналу ділиться на два, таким чином ми визначаємо час руху імпульсу в один бік. Виходячи з вищесказаного, формула визначення похилої дальності:

$$R = \frac{c_0 * t}{2} \quad (5)$$

де c_0 – швидкість світла = $3 * 10^8$ (м/с)

t – виміряний час проходження імпульсу (с)

R – відстань від антени до цілі (м)

За допомогою вимірювання відстані до цілі, визначається її висота польоту, котру теж потрібно враховувати, визначаючи тип цієї цілі. Тому що різні повітряні об'єкти мають різні льотні характеристики й відповідно перебувають у повітрі на різних висотах польоту. В таблиці 2.3 наведено типові висоту польоту для наших цілей. [27]

Таблиця 2.3 – Висоти польоту повітряних об’єктів

Тип об’єкта	Висота (м)
Дипольний відбивач	100-15000
Винищувач	100-15000
Фронтовий бомбардувальник	100-12000
Стратегічний бомбардувальник	100-15000
Транспортний літак	100-12000
Крилата ракета	20-150
ОТРК	10000-50000
Гелікоптер	100-5000

Інформація про висоти польоту різних повітряних об’єктів була взята із відкритих джерел про використовувані у сучасності літальні апарати та ракети.

2.3.4 Розрахунок азимуту об’єкту

Азимутом цілі називається кут між напрямком на північ і горизонтальною проекцією напрямку на ціль. Цей кут відраховується в горизонтальній площині від напрямку на північ за годинниковою стрілкою рисунок 2.11. Також азимут може відраховуватись від центральної осі літака, корабля, з якого ведеться вимір (так званий відносний азимут).

Азимут цілі вимірюється у напрямку головного променя антени в момент приходу відбитого від мети сигналу. Спрямованість антени (характеризується значенням коефіцієнта спрямованої дії) – це здатність антени фокусувати енергію у заданому напрямі.

Антенна з високим коефіцієнтом спрямованої дії називається спрямованою. За положенням спрямованої антени в момент прийому відбитого імпульсу, можна визначити азимут і кут місця цілі. Точність виміру кутових координат залежить від спрямованих властивостей антени.

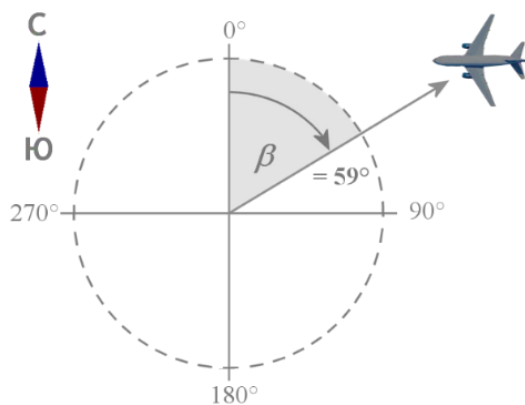


Рисунок 2.11 – Визначення азимуту цілі

Радіолокатори зазвичай використовують для роботи область високих частот. Для цього є низка причин:

- квазіоптичне поширення радіохвиль (по прямій);
- висока роздільна здатність (чим менша довжина хвилі, тим менший розмір об'єктів, які можна виявити);
- що вище частота, то менше розміри антени у тому посиленні.

Антени більшості радарів проектуються таким чином, щоб сформувати промінь діаграми спрямованості у заданому напрямку. Таким чином, змінюючи кутове положення антени, можна переміщати промінь у просторі. Як можна побачити рисунку 2.12, форма променя така, що амплітуда ехо-сигналу змінюється під час проходження променя за ціллю.

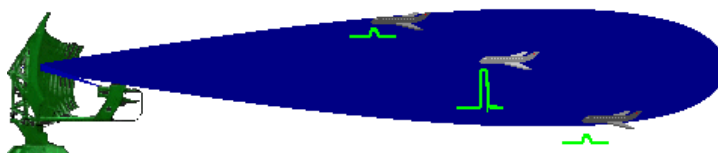


Рисунок 2.12 – Зміна амплітуди ехо-сигналу

У реальних радарах, антена рухається постійно, а точка максимуму відбитого сигналу відстежується спеціальним обладнанням або візуально оператором - це і буде момент, коли промінь "вказує" прямо на ціль. Радіолокатори стеження або

управління озброєнням зазвичай вирішують завдання пошуку максимальної амплітуди ехо-сигналу, після чого виконується утримання променя в цьому напрямку вручну або автоматичними системами стеження.

Для вимірювання азимуту цілі необхідно мати інформацію про направлення на північ в точці установки антени радіолокатора. Для отримання такої інформації виконується операція, яка називається орієнтуванням радіолокаційної станції. У випадку зі старими радіолокаторами для цього використовували компас або орієнтири на місцевості, координати яких відомі. Сучасні системи використовують із цією метою інформацію від супутників GPS.

Швидка та точна передача інформації про направлення антени між місцем її встановлення та місцем встановлення дисплея радару може бути виконана шляхом використання:

- слідкуючих систем;
- шляхом відліку імпульсів зміни азимуту.

Слідкуючі (аналогові) системи використовуються в старих станціях радіолокації і ракетних установках і будуються на основі таких пристроїв, як сельсини. У сучасних радарх використовується система Масштабних Азимутальних Імпульсів (МАІ), яка будується на основі цифрових пристроїв – енкодерів. Вісь енкодера механічно пов'язана з віссю обертання антени радіолокатора. При обертанні антени енкодер генерує імпульси, кількість яких відповідає поточному кутовому положенню антени. Також є радари, які не використовують механічний рух антени. У них застосовується електронна перебудова положення променя антени по азимуту та куту місця (фазовані антенні грати). У таких радарх кутове положення променя антени визначається за значенням управляючих сигналів на фазообертачі або інших елементах, що забезпечують переміщення променя антени. [28]

Азимут необхідно визначати, щоб розуміти географічне місце появи того чи іншого повітряного об'єкту та розуміти його напрямок польоту, задля тої чи іншої реакції на цей об'єкт.

2.4 Інструменти програмної реалізації

Мовою програмування в моєму проекті було обрано об'єктно-орієнтовану мову Python, через її зручність у веденні обрахунків, завдяки великій кількості готових бібліотек.

В якості середовища розробки програми було обрано IDE PyCharm, що має зручний інтерфейс та власний компілятор.

В якості бази даних було обрано постреляційну БД MongoDB, яка виділяється легкістю оперування над нею та зручністю передавання колекцій на інші пристрої, за допомогою JSON-файлів.

2.4.1 Система керування базами даних MongoDB

MongoDB - це система керування базами даних з відкритим вихідним кодом, спрямована на роботу з документами, і не вимагає визначення структури таблиць. MongoDB займає посередню позицію між швидкими та масштабованими системами, які використовують формат ключ-значення, і реляційними системами керування базами даних, які визначені своєю логічною структурою і можливостями для складних запитів [29].

Код MongoDB написаний мовою програмування C++ [30] і розповсюджується на умовах ліцензії AGPLv3.

MongoDB підтримує зберігання даних у форматі, подібному до JSON [31], надаючи гнучкий інструментарій для формування запитів. Вона також дозволяє створювати індекси для різних атрибутів документів, забезпечує ефективне зберігання великих бінарних об'єктів, включає журналювання операцій зміни та додавання даних до бази даних. MongoDB може працювати за парадигмою Map/Reduce, підтримує реплікацію і можливість побудови відмовостійких конфігурацій. У MongoDB існують вбудовані засоби для шардінгу, які дозволяють розподілити дані по серверах за певним ключем. В поєднанні з реплікацією це дозволяє побудувати масштабовані кластери зберігання даних, де відсутня єдина точка відмови, і де автоматично відновлюється працездатність після збоїв.

Розширення кластера або перетворення одного сервера на кластер можливе без зупинки роботи бази даних, просто додавши нові машини до нього.

Мова запитів MongoDB дозволяє виражати запити для пошуку та фільтрації даних, враховуючи складні структури, такі як масиви та вкладені документи.

MongoDB має розширені можливості для індексації, що покращує швидкодію виконання запитів. Вона також підтримує горизонтальне масштабування, що дозволяє розширювати систему для обробки великих обсягів даних.

Застосування MongoDB різноманітне: від розробки веб-застосунків до систем керування контентом та аналізу даних. MongoDB надає ряд інструментів для роботи з даними, включаючи командний рядок, графічний інтерфейс та драйвери для різних мов програмування.

Для розроблюваної мною системи MongoDB має ряд переваг над іншими СКБД, такими як:

- система працюватиме з різноманітними видами даних, пов'язаних із повітряними об'єктами. MongoDB дозволяє легко зберігати та обробляти дані без жорстко визначеної схеми, що важливо у сфері розпізнавання повітряних об'єктів, де може змінюватися формат та структура даних;

- система MongoDB має потужні можливості для роботи з геоданими. Це може бути корисно для ідентифікації та відстеження повітряних об'єктів на індикаторах кругового огляду, де важливо визначати їхнє місцезположення та траєкторію;

- оскільки розроблювана система повинна працювати в режимі реального часу та враховувати великі об'єми даних, масштабованість MongoDB дозволить ефективно обробляти та зберігати великий потік даних від повітряних об'єктів;

- система MongoDB може допомогти забезпечити швидкий доступ до даних, що важливо, оскільки потрібно оперативно аналізувати інформацію та взаємодіяти з ідентифікаторами на індикаторах кругового огляду;

- система MongoDB має ряд драйверів та інструментів для різних мов програмування, що полегшує інтеграцію з вашим програмним забезпеченням.

Загалом, MongoDB може забезпечити потрібний рівень гнучкості, продуктивності та легкості в роботі з геоданими для ефективної інформаційної підтримки ідентифікації повітряних об'єктів на індикаторах кругового огляду.

2.4.2 Графічний інтерфейс MongoDB Compass

MongoDB Compass - це графічний інтерфейс користувача (GUI), який надає зручні та візуально привабливі можливості для роботи з базами даних MongoDB. Цей інструмент призначений для адміністрування та взаємодії з MongoDB, дозволяючи користувачам виконувати різноманітні завдання без використання командного рядка.

MongoDB Compass пропонує візуальне підключення до бази даних та відслідковування структури документів у колекціях. Користувач може легко встановити з'єднання з базою даних MongoDB, використовуючи інтерфейс введення необхідних параметрів, таких як адреса сервера та порт.

Інтерактивний конструктор запитів дозволяє користувачам будувати запити та агрегаційні операції, спираючись на інтуїтивний візуальний інтерфейс. Крім того, MongoDB Compass підтримує геопросторові запити та візуалізації, що дозволяє працювати з геоданими та здійснювати географічний аналіз.

Інструмент включає в себе засоби профілювання запитів та аналізу використання індексів для оптимізації продуктивності операцій. Також MongoDB Compass інтегрується з MongoDB Atlas, надаючи зручні інструменти для адміністрування бази даних у хмарі.

MongoDB Compass виступає як потужний інструмент для адміністрування та розробки MongoDB-проектів, завдяки своїм інтуїтивно зрозумілим та візуальним можливостям, спрощуючи взаємодію з базою даних для розробників та адміністраторів.

Цей інтуїтивно зрозумілий графічний інтерфейс вражає своєю легкістю використання, а представлення даних у вигляді графіки може полегшити їх сприйняття та зробити роботу більш зручною. Для основних операцій, таких як

створення, видалення колекцій, а також додавання, перегляд, редагування та видалення документів, інтерфейс містить відповідні елементи на рисунку 2.13.

При обранні конкретної колекції і перегляді її даних, наявні кнопки для додавання нових даних, включаючи імпорт з зовнішнього файлу. Для кожного документа представлені опції перегляду, редагування та видалення.

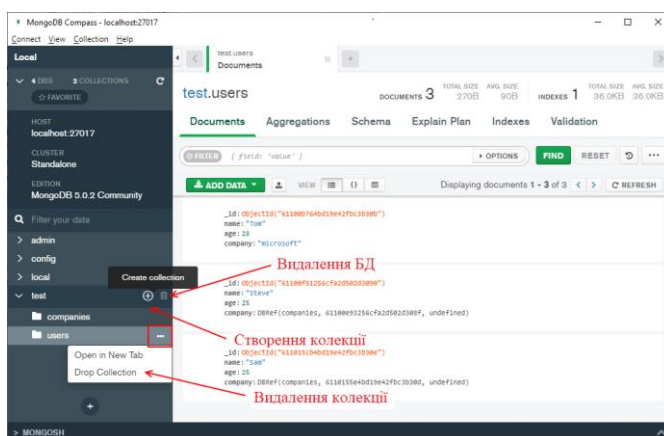


Рисунок 2.13 – Інструменти MongoDB Compass

Незважаючи на зручність графічного інтерфейсу для управління даними, він може виявитися обмеженим для більш складних завдань. Тут на допомогу приходить вбудований консольний клієнт Mongosh у MongoDB Compass, що на рисунку 2.14, де можна вводити майже ті ж команди, що й у звичайній консольній оболонці mongosh.

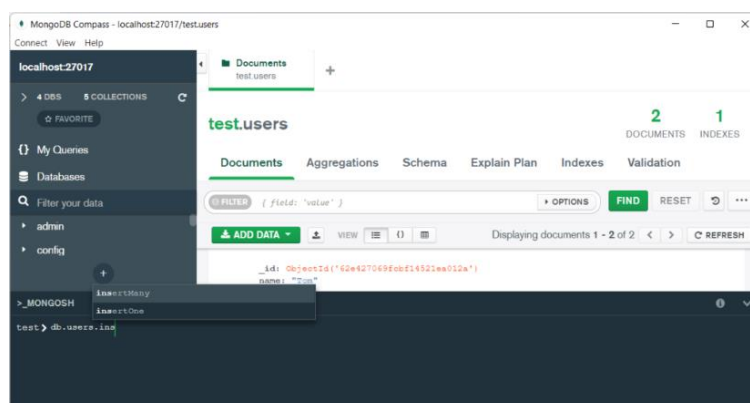


Рисунок 2.14 – Консоль в MongoDB Compass

Важливо відзначити, що консольний клієнт Compass спрощує написання запитів, забезпечуючи функцію автодоповнення, яка пропонує варіанти доступних функцій. [32]

2.4.3 Мова програмування Python

Для розробки програмного забезпечення було обрано мову програмування Python. Python був створений у 1991 році з метою надати легку для розуміння та зручну у використанні мову. Інтерпретатор Python і його стандартні бібліотеки доступні у вихідній і скомпільованій формі на всіх основних платформах. Python є мовою, яка підтримує кілька парадигм програмування, включаючи об'єктно-орієнтований, процедурний, функціональний та аспектно-орієнтований підходи. Мова вирізняється простим та зрозумілим синтаксисом, що сприяє швидкій розробці різноманітних додатків. В даний час Python використовується для розробки веб-застосунків (зокрема, на серверній стороні), наукового обчислення та написання системних сценаріїв. Більшість дослідницьких та розробницьких проектів в галузі машинного навчання використовують мову Python для проведення своїх експериментів. Однак слабкістю цієї мови є її обмежена швидкодія порівняно з скомпільованими аналогами.

Основні риси та особливості мови Python визначають її популярність та широке застосування в галузі програмування:

- мова Python відома своєю простотою та читабельністю коду. Синтаксис мови дозволяє виражати ідеї у декілька рядків коду, що сприяє легкому розумінню програмного коду, особливо для новачків;
- мова Python має вбудовані високорівневі структури даних, такі як списки, словники та кортежі, що робить роботу з даними ефективною та зручною;
- мова підтримує об'єктно-орієнтоване програмування (ООП), що дозволяє створювати та використовувати класи та об'єкти, полегшуючи розробку та обслуговування коду;

- мова Python має велику кількість розширюваних бібліотек та модулів, які полегшують розробку і дозволяють використовувати готові рішення для різноманітних задач;
- програми, написані на Python, можуть працювати на різних операційних системах без змін, що робить мову кросплатформенною;
- мова Python застосовується у багатьох галузях, таких як веб-розробка, наука про дані, штучний інтелект, автоматизація та інші;
- мова Python має велику та активну спільноту розробників, яка внесла значний вклад у розвиток мови та підтримку її користувачів.

Для розробки мого програмного продукту мова Python ідеально підходить. Це спричинено такими її особливостями як простота та читабельність коду, наявністю значної кількості готових бібліотек для проведення обчислень, таких як NumPy. Це стає вельми корисним у роботі з індикаторами кругового огляду, де обробка та аналіз великих обсягів даних є необхідною.

2.4.4 Середовище PyCharm

Середовище розробки PyCharm, що на рисунку 2.15 — це інтегроване середовище розробки (IDE) для мови програмування Python, яке надає широкий спектр інструментів для ефективної розробки та управління проектами. Давайте розглянемо основні характеристики та переваги PyCharm без використання списків.

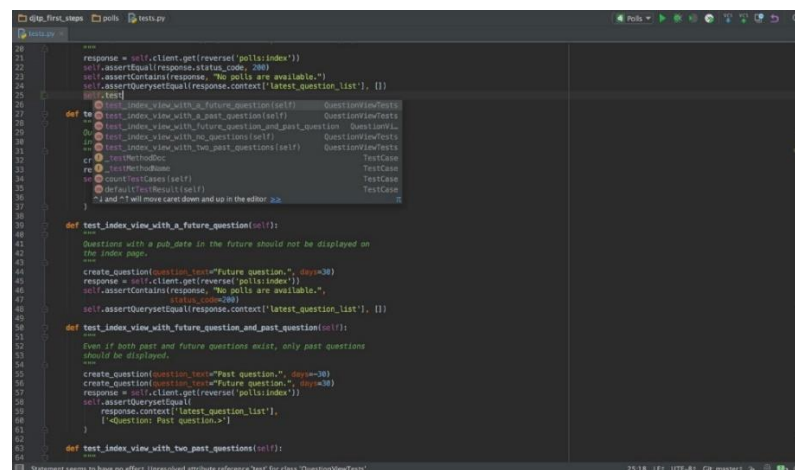


Рисунок 2.15 – Середовище розробки PyCharm

PyCharm вражає своїм інтуїтивно зрозумілим інтерфейсом, який дозволяє розробникам швидко орієнтуватися. Зручний використовуваний текстовий редактор і вбудовані інструменти значно полегшують роботу над проектами будь-якої складності. PyCharm пропонує потужні функції автоматичного доповнення коду, які полегшують введення та зменшують кількість помилок. Також, вбудована система перевірки коду попереджає про можливих проблемах та надає рекомендації для їх виправлення. Засіб для управління віртуальними середовищами та залежностями є невід'ємною частиною PyCharm. Це дозволяє розробникам ефективно використовувати ізольовані середовища для своїх проектів та управляти залежностями.

Це середовище розробки надає високий рівень інтеграції з системами контролю версій, зокрема Git. Це дозволяє командам розробників ефективно працювати над спільними проектами та використовувати всі переваги систем контролю версій.

Середовище розробки PyCharm підтримує популярні фреймворки Python, такі як Django, Flask, Pyramid та інші. Інтегровані інструменти для роботи з фреймворками роблять розробку веб-додатків більш ефективною.

Інструменти рефакторингу дозволяють змінювати структуру коду без втрати його функціональності. Це особливо важливо при роботі з великими та складними проектами.

Середовище PyCharm підтримує такі технології, як HTML, CSS, JavaScript, що дозволяє розробникам створювати комплексні веб-додатки, використовуючи повний стек технологій.

PyCharm включає в себе інструменти для написання та виконання тестів, що дозволяє розробникам створювати стійкі та надійні програми.

Завдяки можливостям роботи з віддаленими проектами та інтеграції з віддаленими середовищами розробки, PyCharm дозволяє командам працювати разом, навіть якщо вони знаходяться в різних місцях.

Узагальнюючи, PyCharm надає розробникам високоефективне середовище для роботи з мовою програмування Python, спрощуючи багато аспектів розробки та забезпечуючи потужні інструменти для підтримки проектів будь-якої складності. [33]

2.4.5 Бібліотека tkinter

Для розробки віконного інтерфейсу в моїй програмі було використану бібліотеку tkinter. Бібліотека tkinter (Tcl/Tk) - це стандартна бібліотека для створення графічного інтерфейсу користувача (ГІК) в Python []. Приклад демонстрації можливостей бібліотеки на рисунку 2.16.

Ось декілька основних особливостей бібліотеки tkinter:

- бібліотека tkinter є частиною стандартної бібліотеки Python, тому не потрібно встановлювати додаткові пакети чи бібліотеки. Вона доступна "з коробки" і вже встановлена з Python;

Srl.No	Subject	Grade	Sub Credit	Credit obtained
1	CS 201		4	
2	CS 202		4	
3	MA 201		3	
4	EC 201		4	

submit

Total credit
SGPA

Рисунок 2.16 – Вікно розроблене інструментами tkinter

- бібліотека tkinter доступний на багатьох платформах, включаючи Windows, macOS і різні дистрибутиви Linux. Це означає, що інтерфейс, створений з використанням tkinter, буде працювати на різних операційних системах без змін;

- бібліотека tkinter надає досить простий інтерфейс для створення вікон, кнопок, міток, полів введення і інших елементів інтерфейсу. Можна використовувати функції та методи для створення та налаштування цих елементів;

- бібліотека tkinter дозволяє налаштовувати різні атрибути і стилі для елементів інтерфейсу, такі як кольори, шрифти, розміри тощо;

- бібліотека tkinter має спеціальні елементи для відображення тексту та графіки, включаючи мітки, тексти, зображення, канваси для малювання тощо;

- бібліотека `tkinter` надає можливість обробки подій, таких як натискання кнопок миші, клавіш клавіатури, зміна розміру вікна тощо. Можна прив'язувати функції до цих подій для обробки їх у вашому додатку;

- бібліотека `tkinter` має добре документованій API, а також існують багато прикладів і ресурсів для навчання та розвитку. Ви зможете знайти інструкції та приклади в офіційній документації та на інших ресурсах.

Бібліотека `tkinter` є популярним вибором для створення простих та середніх за розміром десктопних додатків з графічним інтерфейсом в Python, і вона добре підходить для багатьох завдань. [34]

2.4.6 Бібліотека `json`

Бібліотека `json` у мові програмування Python є вбудованим інструментом для роботи з форматом даних JSON (JavaScript Object Notation). JSON є легким текстовим форматом обміну даними, який широко використовується для передачі даних між сервером та клієнтом, а також для зберігання структурованих даних. У моїй роботі, де база даних розроблена в MongoDB, використання бібліотеки `json` може бути доцільним з кількох причин:

- бібліотека `json` дозволяє конвертувати дані з формату Python у формат JSON для збереження їх у базі даних MongoDB, а також зчитувати JSON-дані з бази для подальшої обробки в Python. Це особливо корисно, коли потрібно взаємодіяти з базою даних MongoDB через мову Python;

- формат JSON підтримує базові типи даних, такі як рядки, числа, масиви, об'єкти, булеві значення та `null`. Бібліотека `json` у Python автоматично перетворює об'єкти Python у валідний JSON та навпаки. Це забезпечує універсальність при зберіганні та отриманні даних з MongoDB;

- формат JSON є популярним форматом обміну даними, і використання бібліотеки `json` полегшує взаємодію між різними компонентами системи, особливо якщо вони написані на різних мовах програмування;

- база даних MongoDB – схемно менша, але іноді важливо здійснювати контроль над тим, які дані зберігаються у базі. Бібліотека json уможливорює зручний спосіб перетворювати об'єкти Python на JSON-формат з обраною структурою;

- якщо застосунок взаємодіє із зовнішніми сервісами, які використовують JSON для передачі даних, використання бібліотеки json полегшить обробку цих даних в мові Python;

- при роботі із запитами до MongoDB можна використовувати бібліотеку json для форматування та передачі параметрів запитів у вигляді JSON-рядків.

Використання бібліотеки json в поєднанні з MongoDB надає зручність та гнучкість для роботи з даними та дозволяє забезпечити ефективний обмін і збереження даних в вашому застосунку. [35]

2.4.7 Бібліотека numpy

NumPy (Numerical Python) – це потужна бібліотека для мови програмування Python, призначена для роботи з числовими даними та виконання наукових обчислень. Ось детальний огляд її основних можливостей та переваг для моєї роботи:

- бібліотека NumPy надає об'єкт масиву (numpy.array), який є основним засобом для представлення та маніпулювання даними. Масиви NumPy ефективно використовують пам'ять та дозволяють виконувати різноманітні операції над числовими даними;

- одна з ключових переваг - це векторизація операцій. Замість використання циклів для обробки елементів масиву, можна використовувати операції на рівні масивів, що поліпшує швидкість виконання коду та робить його більш зрозумілим;

- бібліотека надає багато вбудованих математичних функцій, таких як тригонометричні, логарифмічні, статистичні тощо. Це робить легким виконання складних математичних операцій без необхідності власноручно втручатися у деталі реалізації;

- бібліотека NumPy підтримує багатовимірні масиви, що особливо корисно для обробки та аналізу даних високої розмірності, які можуть виникнути у роботі;

- для робіт, пов'язаних із зображенням та аналізом сигналів, важливо мати доступ до функцій лінійної алгебри. NumPy містить в собі багато відповідних функцій;

- бібліотека NumPy часто використовується в поєднанні з іншими бібліотеками для наукових обчислень, такими як SciPy, Matplotlib та іншими. Вона є основною складовою стеку для наукових обчислень у Python;

- бібліотека NumPy дозволяє ефективно використовувати пам'ять, що важливо для роботи з великими об'ємами даних.

Загалом, NumPy є потужним інструментом для роботи з числовими даними та виконання наукових обчислень в Python. Вона допомагає виконувати операції над даними, що може бути корисним у моїй роботі з обробки сигналів. [36]

2.4.8 Бібліотека tomllib

Бібліотека tomllib — це новий модуль у стандартній бібліотеці Python 3.11 для синтаксичного аналізу файлів TOML (Tom's Obvious Minimal Language). TOML — це простий, зрозумілий формат файлів конфігурації. Він стає все більш популярним у спільноті Python, оскільки його використовують багато популярних інструментів і фреймворків, наприклад Poetry. Бібліотека tomllib надає кілька переваг перед використанням сторонніх парсерів TOML:

- є частиною стандартної бібліотеки Python, тому немає необхідності встановлювати додаткові пакети;

- працює швидко та ефективно;

- добре протестована і має підтримку.

Бібліотека tomllib корисна для будь-якого проекту, якому потрібно читати файли конфігурації TOML. Ось кілька конкретних прикладів того, як можна використовувати tomllib:

- щоб проаналізувати файл конфігурації (наприклад, pyproject.toml) для проекту Python;

- для пакета Python, проаналізувати файл конфігурації (наприклад, setup.cfg або pyproject.toml);

- щоб проаналізувати файл конфігурації для інструменту або фреймворку Python (наприклад, Poetry);
- для аналізу файлів даних у форматі TOML.

Деякі інші аналізатори TOML, доступні для Python, включають `tomli` та `pytoml`. Бібліотека `tomllib` є найновішим синтаксичним аналізатором TOML, доданим до стандартної бібліотеки Python, і вважається найнадійнішим та найефективнішим парсером TOML. Ось «Таблиця 2.4» де порівнюються всі парсери TOML:

Таблиця 2.4 – Порівняння `tomllib` з іншими популярними парсерами TOML

<code>tomllib</code>	Входить до стандартної бібліотеки Python, швидка і ефективна, добре протестована, підтримується та сумісна зі специфікацією TOML 1.0	Немає функцій для запису нових файлів TOML
<code>tomli</code>	Підтримує запис файлів TOML, є швидким і ефективним і має велику й активну спільноту	Немає в стандартній бібліотеці Python
<code>pytoml</code>	Підтримує запис файлів TOML і має простий і легкий у використанні API	Не така швидка і ефективна, як <code>tomllib</code> або <code>tomli</code> , і не має такої великої та активної спільноти.

Бібліотека `tomllib` є потужним і надійним аналізатором TOML, але вона має кілька обмежень:

- `tomllib` не підтримує запис файлів TOML, тобто його не можна використовувати для створення нових файлів TOML;
- `tomllib` не підтримує всі функції специфікації TOML 1.1. Наприклад, `tomllib` не підтримує вбудовані таблиці TOML.

Бібліотека `tomllib` є відносно новим аналізатором TOML, але вона швидко стає найпопулярнішим аналізатором TOML для Python. Команда `tomllib` активно розробляє аналізатор і планує додати підтримку додаткових функцій у майбутньому. Загалом `tomllib` є чудовим вибором для аналізу файлів TOML у Python. [37]

2.4.9 Бібліотека `random`

Бібліотека `random` в мові програмування Python використовується для генерації випадкових чисел та інших елементів у програмному коді.

Ця бібліотека дозволяє генерувати випадкові числа за різними закономірностями. Вона надає функції для отримання випадкового цілого числа в заданому діапазоні, випадкового числа з плаваючою комою, та інші.

Загалом, `random` є потужним інструментом для роботи з випадковими елементами в програмах, де потрібна випадковість чи вибір елементів з деякого набору. У моїй роботі з обробкою сигналів ця бібліотека необхідна для симуляції та генерації тестових даних в моєму симуляторі РЛС. [38]

2.4.10 Бібліотека `hashlib`

Бібліотека `hashlib` в мові програмування Python надає інтерфейс для реалізації хеш-функцій, які використовуються для створення унікальних фіксованих довжинних значень (хешів) з будь-яких вхідних даних. Це може бути корисним в контексті роботи з шифруванням та забезпеченням цілісності даних. В моїй роботі використовується симетричне AES-шифрування, тому ось детальний огляд бібліотеки `hashlib` та її доцільності в контексті використання методу AES-шифрування:

– під час використання методу AES-шифрування для захисту конфіденційних даних, важливо також переконатися, що ці дані не були змінені. Можна використовувати `hashlib` для створення хеш-значень вхідних даних і зберігати їх разом із зашифрованими даними. Після розшифрування можна обчислити хеш для розшифрованих даних і порівняти його зі збереженим хешем, щоб перевірити їх цілісність;

– хеш-функції можна використовувати для створення унікального ідентифікатора для набору даних. Це корисно при збереженні та відстеженні зашифрованих об'єктів у базі даних;

– при використанні паролів в застосунку, `hashlib` може бути використана для безпечного зберігання паролів у вигляді хеш-значень, замість зберігання їх у чистому вигляді;

– хеш-значення можна використовувати для перевірки цифрових підписів. Після використання методу AES для шифрування і підписання даних, можна обчислити хеш для розшифрованих даних і порівняти його з підписом для підтвердження автентичності даних.

Загалом, використання `hashlib` в контексті роботи з методом AES-шифрування дозволяє забезпечити цілісність, ідентифікацію та безпеку зашифрованих даних. [39]

2.4.11 Бібліотека `pymongo`

Бібліотека `PyMongo` є інтерфейсом для взаємодії мови програмування `Python` з базою даних `MongoDB` []. У моєму проєкті, де я розробляю програму для аналізу даних з радару та зберігання їх в базі даних `MongoDB`, `PyMongo` може бути надзвичайно корисною. Ось приклади використання `PyMongo` у програмах:

Підключення до `MongoDB` на рисунку 2.17: Використання `PyMongo` для підключення до бази даних `MongoDB`. Можна вказати параметри підключення, такі як хост та порт бази даних.

```
from pymongo import MongoClient

# Підключення до сервера MongoDB
client = MongoClient('localhost', 27017)
```

Рисунок 2.17 – Приклад підключення до MongoDB

Вибір колекцій на рисунку 2.18: У MongoDB дані зазвичай зберігаються у колекціях, які відповідають таблицям у традиційних реляційних базах даних. Можна вибрати колекцію, з якою будемо працювати.

```
# Вибір колекції для зберігання даних з радару
db = client['радар_дані']
collection = db['дані_з_радару']
```

Рисунок 2.18 – Приклад вибору колекцій

Додавання даних в базу даних на рисунку 2.19: Можна використовувати PyMongo для додавання отриманих даних з радару до бази даних.

```
# Додавання запису до колекції
data = {
    'timestamp': '2023-10-09 14:30:00',
    'object_type': 'літак',
    'height': 10000,
    'speed': 300,
    'size': 'середній'
}
collection.insert_one(data)
```

Рисунок 2.19 – Приклад додавання даних в БД

Запити до бази даних на рисунку 2.20: Можна використовувати PyMongo для виконання запитів до бази даних та отримання інформації.

```
# Запит для вибору всіх записів про літаки
airplane_records = collection.find({'object_type': 'літак'})
for record in airplane_records:
    print(record)
```

Рисунок 2.20 – Приклад запиту до БД

Оновлення та видалення даних на рисунку 2.21: PyMongo також надає можливість оновлення та видалення даних у базі даних за потреби.

```
# Оновлення даних для певного запису
collection.update_one({'object_type': 'літак'}, {'$set': {'height': 12000}})

# Видалення запису
collection.delete_one({'object_type': 'літак'})
```

Рисунок 2.21 – Приклад оновлення та видалення даних з БД

За допомогою PyMongo можна здійснювати взаємодію з базою даних MongoDB, зберігати та витягати дані для аналізу, а також вести записи про роботу з радаром. PyMongo допоможе легко і ефективно інтегрувати базу даних MongoDB у мій проект. [40]

Висновки до розділу 2

У даному розділі було розглянуто існуючі рішення систем обробки сигналів з радарів та представлено математичні дані і формули, необхідні для успішної реалізації MVP проекту. Також були представлені табличні показники, які використовувалися під час розробки та налаштування програмного забезпечення, що стануть важливим базисом для подальших кроків у розробці та вдосконаленні системи обробки сигналів з радарів, допомагаючи забезпечити ефективність та надійність майбутнього продукту. В останньому підрозділі було представлено всі використані засоби для розробки програмного продукту.

З ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ЗАСТОСУНКУ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ ДЛЯ ІДЕНТИФІКАЦІЇ ПОВІТРЯНИХ ОБ'ЄКТІВ

Для реалізації застосунку, що буде проводити інформаційну підтримку ідентифікаторам кругового огляду, необхідно розробити програмне забезпечення яке дозволить взаємодіяти з цими ідентифікаторами та аналізувати їх сигнали. Дані процеси повинні відбуватися у прямому ефірі перед користувачем системи й працювати безперебійно, через велику важливість в справності роботи подібної системи. Також важлива захищеність подібної системи від стороннього використання.

3.1 Архітектура програмного забезпечення

Мета яка стояла за створенням програмного застосунку – надання користувачеві простої можливості користування та керування даною системою. Мінімально життєздатний продукт представлено у віконному вигляді, підготовлено до роботи на даний момент із модулем симулятора РЛС, що налаштовано за прикладом реального РЛС П-18 «Малахит», має підключення до бази даних, яка містить в собі таблицю еталонних параметрів деяких повітряних об'єктів, а також таблицю для збереження отриманих даних в результаті роботи системи. Всі базові маніпуляції над системою користувач здатний виконати із використанням розробленого віконного інтерфейсу.

Задачі, які були окресленні при створенні програмного продукту:

- користувач має змогу привести систему в дію, підключивши до симулятора радару;

- користувач може отримати дані про поточний стан віртуального повітряного простору на радарі, а також переглянути минулі записи бази даних про повітряні об'єкти;
- застосунок використовує алгоритм порівняння, котрий порівнює повітряні об'єкти в реальному часі із даними в БД та відповідно надає результати аналізу на екрані й записує їх до БД;
- застосунок використовує шифрування збережених даних задля їх безпеки та цілісності;
- програмний застосунок містить в собі також алгоритми порівняння даних, шифрування даних та дешифрування даних.

Задля демонстрації роботи програми створено симулятор радару, що являє собою генератор об'єктів у повітряному просторі та котрий підключено до мого застосунку. Генератор має змогу створювати в повітряному просторі об'єкти різних типів, котрі вже фіксує наш радар та передає ці дані до алгоритму порівняння, задля визначення типу цілі і запису її до бази даних.

Нижче, на рисунку 3.1 наведена схема взаємодії програмних модулів, задачі яких були описані в попередніх пунктах.

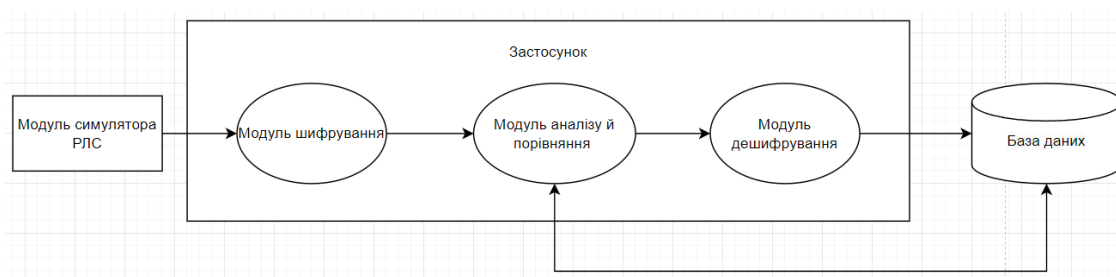


Рисунок 3.1 – Схема взаємодії модулів програмного продукту.

Система розроблена об'єктно-орієнтованою мовою програмування, тому вона звісно має класи.

Система складається із таких класів:

- class Generator - він бере дані із бази даних Mongo та відносно цих даних генерує об'єкти. Для зв'язку з БД в моїй програмі слугує клас db_mongo;

- class db_mongo - цей клас слугує "мостом" між програмою та базою даних і виконує функції читання та запису;
- class Identification - цей клас бере дані, які згенерував клас Generator, а також еталонні дані із класу db_mongo, порівнює їх і записує результати в БД. Також цей клас керується через клас інтерфейсу class Application;
- class Application - цей клас представляє із себе користувацький інтерфейс програми. Він, при натисканні певних кнопок користувачем, безпосередньо подає команду на активацію генератора, а також роботу класу Identification, що порівнює згенеровані дані.

3.2 Принципи роботи складових частин програмного забезпечення

Розроблений програмний застосунок складається із трьох основних частин: самого програмного застосунку (який включає в себе віконний користувацький інтерфейс, модуль алгоритму порівняння та модуль шифрування даних), базу даних, що зберігає в собі еталонні дані про повітряні об'єкти, а також реальні повітряні об'єкти, що були зафіксовані радаром, та модуль симулятору радару, котрий задля демонстрації роботи застосунку генерує повітряні об'єкти. Перша та третя частини розроблені мовою програмування Python, а база даних розроблена використовуючи постреляційну СКДБ MongoDB.

3.2.1 Опис роботи бази даних у системі

Розроблена за допомогою СКБД MongoDB база даних виконує дві головні функції: збереження даних про еталонні параметри повітряних цілей та збереження даних про ідентифіковані реальні повітряні цілі. Тому на даний момент база даних програми складається всього з двох таблиць.

Перше таблиця бази даних зберігає у собі еталонні дані про повітряні об'єкти та містить в собі такі поля як у таблиці 3.1.

Таблиця 3.1 – Структура таблиці catalog в базі даних

Назва поля	Тип поля	Опис
id	ObjectId	Унікальний ідентифікатор
ЕПР (min)	Int32	Мінімально можлива ЕПР цього об'єкту при нормальних умовах
ЕПР (max)	Int32	Максимально можлива ЕПР цього об'єкту при нормальних умовах
Швидкість (min) м/с	Int32	Мінімально можлива швидкість польоту цього об'єкту при нормальних умовах
Швидкість (max) м/с	Int32	Максимально можлива швидкість польоту цього об'єкту при нормальних умовах
Висота польоту (min) м	Int32	Мінімально можлива висота польоту цього об'єкту при нормальних умовах
Висота польоту (max) м	Int32	Максимально можлива висота польоту цього об'єкту при нормальних умовах
Тип цілі	String	Класифікація цього об'єкту

Друга таблиця бази даних зберігає дані про вже знайдені та проаналізовані повітряні об'єкти, за допомогою алгоритму порівняння. Ця таблиця бази даних містить наступні поля, що зображено в таблиці 3.2.

Таблиця 3.2 – Структура таблиці output_data в базі даних

Назва поля	Тип поля	Опис
id	ObjectId	Унікальний ідентифікатор
Азимут цілі	Int32	Азимут відносно горизонту на зафіксований об'єкт
Відстань до цілі	Double	Відстань від РЛС до об'єкта
ЕПР	Double	Визначена ЕПР цього об'єкта
Швидкість цілі	Int32	Визначена швидкість цього об'єкта
Час фіксації	String	Час, за яким зафіксовано об'єкт
Тип цілі	String	Визначений алгоритмом тип об'єкта

Більш детально опишемо параметри цих полів:

– азимут цілі: це поле вказує на напрямок до цілі від початкового пункту, вимірюваний в градусах від 1 до 360. Він вказує на те, в якому напрямку знаходиться ціль відносно радару або початкового положення, допомагаючи визначити, куди слід спрямовувати антену радару для виявлення цілі;

– відстань до цілі: це поле вказує на відстань від радару до цілі. Вимірюється у кілометрах. Визначення відстані допомагає з'ясувати, наскільки близько або

далеко знаходиться ціль від радару, а також при повторному скануванні однієї і тієї ж цілі чи наближається вона або віддаляється відносно радіолокатора;

- ефективна площа розсіювання цілі: це поле вказує на величину, яка характеризує, наскільки ціль розсіює радіосигнали, зазвичай вимірюється в квадратних метрах. Ця величина вказує на здатність цілі бути виявленою радаром, де менші значення означають меншу видимість перед радаром. Це основний параметр, котрий порівнюється задля ідентифікації типу цілі;

- швидкість цілі: це поле вказує на швидкість руху цілі. Вимірюється у відповідних одиницях виміру, таких як кілометри на годину чи милі на годину. Визначення швидкості цілі допомагає прогнозувати її майбутнє положення, а також допомагає ідентифікувати тип цієї цілі, знаючи середні швидкості повітряних об'єктів;

- час фіксації цілі: це поле вказує на момент часу, коли ціль була виявлена радаром або коли була оновлена інформація про ціль. Цей параметр допомагає визначити, наскільки актуальною є інформація про ціль;

- визначений тип цілі: це поле вказує на класифікацію цілі відповідно до її характеристик. Включає в себе такі типи, як "винищувач", "гелікоптер", "ракета" і т. д. Цей параметр допомагає визначити, з якими типами об'єктів у повітрі ви маєте справу.

Кожне з цих полів є важливим для зберігання та аналізу даних про повітряні цілі та робить вашу систему зберігання інформації ефективною та корисною для операторів радару.

Таблиці бази даних містять поля з типами:

- тип ObjectID: унікальний ідентифікатор, який використовується для ідентифікації документів в колекціях MongoDB. Унікальність гарантується за рахунок комбінації timestamp, ідентифікатора машини, ідентифікатора процесу та випадкового числа;

- тип Int32: 32-бітне ціле число, необхідне для зберігання цілих чисел;

- тип Double: 64-бітне число з плаваючою комою. Використовується для зберігання чисел з плаваючою комою, таких як десяткові дробки;

– тип String: Рядок Unicode. Використовується для зберігання текстової інформації.

3.2.2 Модуль симулятору радіолокаційної станції та генератору об'єктів

Мій модуль симулятору РЛС, який є генератором об'єктів, є важливою частиною мого проекту. Основна мета цього симулятора – створення віртуальних повітряних об'єктів із визначеними характеристиками для подальшого аналізу моєю програмою. Ось детальний опис, як працює цей симулятор:

– визначення діапазонів параметрів: спершу ми визначаємо діапазони параметрів для повітряних об'єктів, які ми хочемо створювати. Це включає параметри, такі як азимут на ціль, відстань до цілі, швидкість цілі, ефективна площа розсіювання та інші характеристики. Ці параметри записані у таблиці catalog моєї бази даних;

– генерація даних об'єктів: симулятор створює віртуальні об'єкти, генеруючи для кожного об'єкта значення параметрів в межах визначених діапазонів. Наприклад, для створення віртуального винищувача, симулятор може вибирати випадкові значення азимуту, відстані, швидкості і інших параметрів в межах визначених діапазонів;

– подача даних до алгоритму порівняння: отримані дані про віртуальні об'єкти, включаючи їхні параметри та типи, передаються до моєї програми для подальшого аналізу;

– аналіз та обробка даних: програма аналізує отримані дані, порівнюючи їх з еталонними даними та визначаючи типи цілей. Результати аналізу відображаються у програмному застосунку та записуються в базу даних у таблицю output_data.

Симулятор РЛС потрібен щоб створити тестовий набір даних для аналізу та перевірки моєї програми, демонстрації її роботи без використання реального обладнання, а також дозволяє розробляти та вдосконалювати алгоритми на основі віртуальних даних перед їхнім реальним використанням.

3.2.3 Модуль алгоритму порівняння даних

Однією з головних частин програми є модуль алгоритму порівняння даних. Робота РЛС це досить непроста система процесів, що частково схематично відображено на рисунку 3.2.

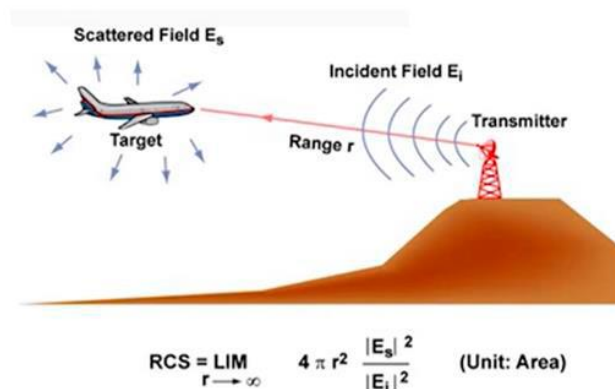


Рисунок 3.2 – Демонстрація роботи РЛС

Даний алгоритм виконує функцію порівняння даних і працює наступним чином:

- вибір даних для порівняння: алгоритм розпочинає роботу з отриманих даних від симулятора РЛС та еталонних даних з бази даних. Це такі дані як азимут на ціль, відстань до цілі, швидкість цілі, ефективна площа розсіювання цілі та час її фіксації;
- порівняння даних: алгоритм порівнює отримані дані з еталонними даними, що зберігаються у таблиці бази даних. Порівняння включає визначення різниць між отриманими та еталонними значеннями параметрів;
- визначення ймовірного типу цілі: алгоритм враховує результати порівняння та визначає, до якого типу цілі найбільше підходять отримані дані. Він враховує вже задані у БД діапазони значень параметрів, щоб визначити тип цілі. Через особливості відбивання сигналу РЛС різними типами об'єктів, в програмі надано перевагу при визначенні типу цілі на її швидкість, а не ефективну площу розсіювання;

- збереження інформації про ціль: після визначення ймовірного типу цілі, алгоритм створює запис з інформацією про цю ціль у іншій таблиці бази даних. Цей запис включає тип цілі, параметри цілі, час фіксації та іншу важливу інформацію;
- відображення результатів: програмний застосунок відображає результати порівняння на екрані користувача, надаючи інформацію про визначений тип цілі та інші характеристики цілі.

Цей алгоритм допомагає визначити типи цілей на основі отриманих даних з радару та еталонних значень, зіставляючи їх, і забезпечує збереження цієї інформації для подальшого використання та аналізу.

3.2.4 Модуль шифрування та дешифрування даних

Оскільки моя програма обробляє дані, котрі можна вважати важливими з точки зору безпеки, то було прийнято рішення додати механізм шифрування певних даних в моїй програмі, а саме даних від симулятора РЛС, котрі в майбутньому можуть бути замінені даними від реально існуючої радіолокаційної станції та також бути зашифрованими. Методом шифрування було обрано один із найпопулярніших шифрів в сьогоденні – Advanced Encryption Standard [41].

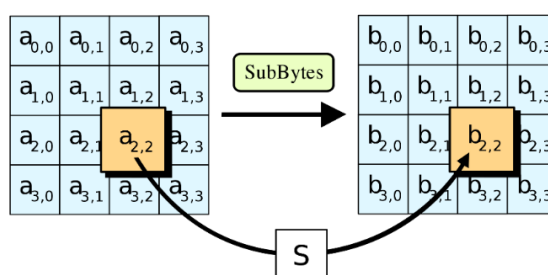


Рисунок 3.3 – Демонстрація роботи AES шифрування

AES (Advanced Encryption Standard), що на рисунку 3.3 – це симетричний блоковий шифр, який може використовуватися для шифрування даних перед їхнім передаванням та розшифрування після отримання [42].

Цей процес допомагає захистити конфіденційні дані у програмі. Розглянемо його детально шифрування даних (від симулятора до програми):

- вибір ключа шифрування: спершу обирається секретний ключ шифрування, який використовуватиметься для шифрування та розшифрування даних. Ключ повинен бути довгим і складним для того, щоб забезпечити безпеку;

- розбиття даних на блоки: дані, які відправляються від симулятора, розбиваються на блоки певної фіксованої довжини (в моєму випадку, 16 байтів - 128 біт);

- додавання вектора ініціалізації (IV): до кожного блоку даних додається вектор ініціалізації. Це унікальне значення, яке забезпечує додатковий рівень безпеки шифрування. IV повинен бути випадковим та унікальним для кожного блоку даних;

- шифрування блоків даних: застосовується алгоритм AES до кожного блоку даних, використовуючи вибраний ключ шифрування. Результатом є зашифрований блок даних;

- відправка зашифрованих даних: зашифровані блоки даних передаються до моєї програми.

Розшифрування даних (перед записом до бази даних):

- отримання зашифрованих даних: моя програма отримує зашифровані дані від симулятора;

- використання ключа шифрування та IV: програма використовує відомий ключ шифрування та IV для розшифрування кожного блоку даних;

- розшифрування блоків даних: застосовується алгоритм розшифрування AES до кожного зашифрованого блоку даних з використанням ключа шифрування і IV;

- відновлення оригінальних даних: результатом розшифрування є оригінальні дані, які можна обробити програмою для подальшого аналізу;

- запис до бази даних: розшифровані дані, які вже були оброблені та аналізовані, можуть бути записані до бази даних для зберігання і подальшого використання.

За допомогою AES-шифрування можна забезпечити конфіденційність та цілісність даних під час їхнього передавання та зберігання.

В моєму застосунку, задля його демонстративності, шифрування відбувається в моменти від отримання даних і до їх запису у таблицю бази даних. В робочій програмі шифрування буде реалізовано на всьому шляху передачі даних, в тому числі і в самій базі даних, задля убезпечення системи від витoku інформації.

3.3 Діаграма послідовності розробленої програми

Діаграма послідовності є одним із видів UML-діаграм і використовується для моделювання взаємодії між об'єктами в рамках конкретного сценарію або процесу. Ця діаграма відображає порядок виконання різних етапів або дій в системі. Основна мета - зображення послідовності повідомлень, які передаються між об'єктами в певному контексті.

Основні елементи діаграми послідовності включають:

- об'єкти (participants): представляють учасників системи, які взаємодіють між собою. Об'єкти зазвичай представляють класи або екземпляри класів;
- лінії життя (lifelines): визначають тривалість існування об'єкта під час виконання сценарію. Лінії життя позначаються вертикальними лініями, або "життєвими лініями", із зазначенням імені об'єкта чи класу;
- послідовність стрілок (sequence arrows): відображають напрямок та порядок передачі повідомлень між об'єктами. Стрілки показують, яка дія спричиняє виклик методу або обмін повідомленнями між об'єктами;
- фрейми (frames): групують повідомлення, що належать до певного етапу чи операції в системі. Фрейми можуть використовуватися для логічного розділення послідовності на більш зрозумілі частини;
- повідомлення (messages): показують взаємодію між об'єктами. Існують два типи повідомлень: синхронні (зображені стрілкою з блоком на кінці) та асинхронні (зображені стрілкою без блока на кінці).

Процес побудови діаграми послідовності зазвичай розпочинається з ідентифікації учасників системи, визначення їх ліній життя та встановлення послідовності взаємодії між ними. Діаграма дозволяє візуалізувати та розуміти порядок виконання дій в системі, а також ідентифікувати можливість виникнення проблем чи нестійкої взаємодії між об'єктами. [43]

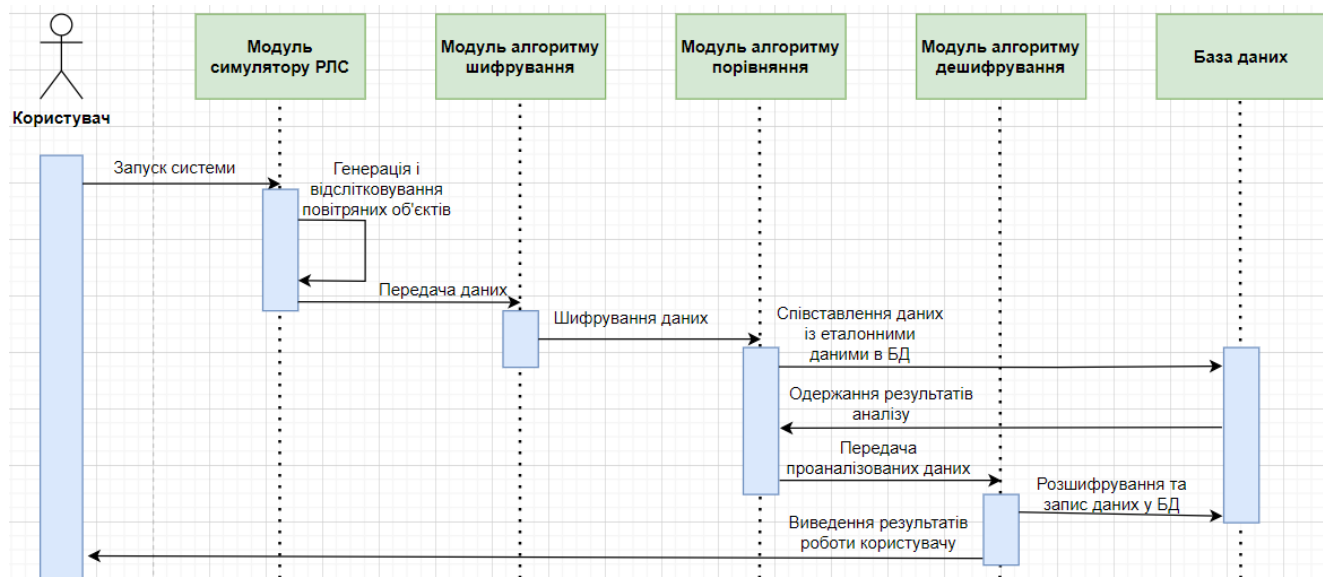


Рисунок 3.4 – Діаграма послідовності розробленої системи

Діаграма послідовності, яка відображає схему роботи моєї системи представлена далі на рисунку 3.4.

Цей інструмент є корисним для розробників програмного забезпечення, аналітиків та інших учасників процесу розробки для уточнення та вдосконалення архітектури системи.

3.4 Діаграма прецедентів розробленої програми

Діаграма прецедентів є одним із видів UML-діаграм та використовується для моделювання взаємодії між системою та її зовнішніми елементами, такими як користувачі або інші системи. Головна мета цієї діаграми - ідентифікація та

відображення функціональності системи через її взаємодію з акторами (actors) та прецедентами (use cases).

Основні елементи діаграми прецедентів включають:

- актори (Actors): актори представляють зовнішніх учасників або системи, які взаємодіють з системою. Це можуть бути користувачі, зовнішні програми, або будь-які інші сутності, які взаємодіють з системою;

- прецеденти (Use Cases): прецеденти представляють конкретні функціональні можливості системи. Кожен прецедент описує конкретний сценарій взаємодії між акторами та системою. Прецеденти можуть бути, наприклад, "Реєстрація користувача", "Оформлення замовлення" тощо;

- лінії асоціацій (Association Lines): вони визначають взаємозв'язок між акторами та прецедентами. Лінії асоціацій показують, як актор взаємодіє з конкретним прецедентом;

- система (System): головна сутність, яка взаємодіє з акторами через різні прецеденти. У деяких випадках ця сутність може бути позначена як «система» або інше ім'я, що вказує на саму систему;

- включення та розширення (Include and Extend Relationships): ці відносини допомагають вказати включення або розширення функціональності між різними прецедентами;

- сценарії (Scenarios): деякі діаграми прецедентів можуть містити додаткову інформацію у вигляді сценаріїв, які описують послідовність подій та взаємодію між акторами та прецедентами.

Діаграма прецедентів дозволяє команді проекту чітко визначити, як система взаємодіє з її користувачами або іншими системами, і визначити основні функціональні можливості, які має надавати система. Це корисний інструмент для аналізу та проектування систем, а також для спілкування з зацікавленими сторонами щодо вимог до системи. [44]

Діаграма прецедентів моєї системи наведена на рисунку 3.5 і включає в себе лише одного актора (користувача).

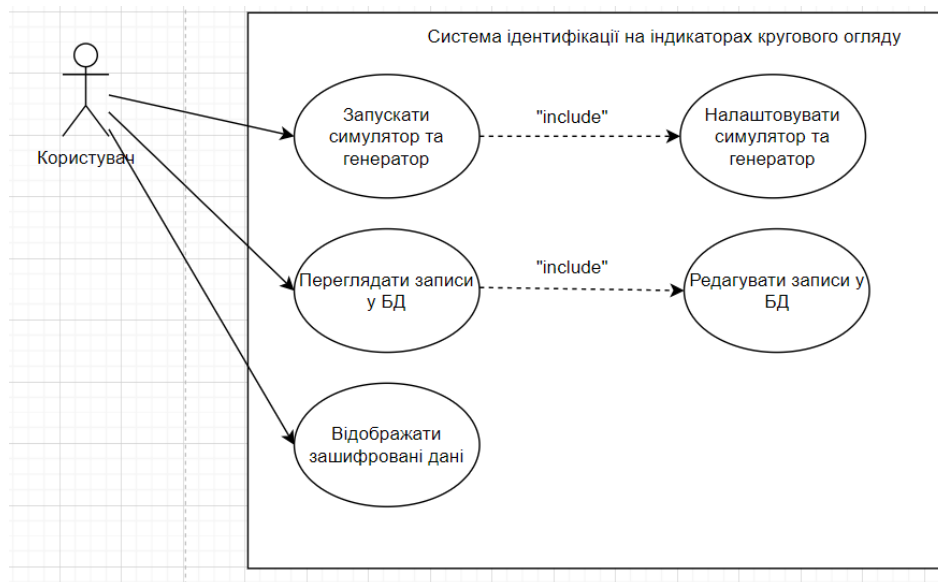


Рисунок 3.5 – Діаграма прецедентів розробленої системи

Як можна побачити, користувач може запускати й налаштовувати симулятор радару, переглядати отримані дані у вигляді відображення записів із бази даних, також через саму базу даних є можливість їх редагувати. На останок є можливість вивести дані у зашифрованому вигляді.

Висновки до розділу 3

У ході виконання завдання було успішно розроблено програмне забезпечення, призначене для взаємодії з радіолокаційними станціями та аналізу їх сигналів. Розроблене програмне забезпечення забезпечує проведення цих процесів у режимі реального часу перед користувачем системи, гарантуючи при цьому безперебійну роботу системи. Для забезпечення її ефективності системи відпрацьована її справність роботи. Для захисту системи від неправомірного використання та безпечної її експлуатації додано механізми шифрування інформації, у вигляді симетричного шифрування. Також були представлені діаграми послідовності та прецедентів, задля демонстративності принципів роботи розробленої системи.

4 ОПИС ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Створене програмне забезпечення включає в себе систему обробки сигналів із власноруч створеного симулятора РЛС та збереження необхідної інформації до бази даних, із шифруванням у процесі передавання даних.

4.1 Робота користувача за допомогою створеної програми

Програмний інтерфейс являє собою простий віконний вигляд, розроблений бібліотекою tkinter. Після відкриття програми нас зустрічають три кнопки: «Панель запуску», «Перегляд записів» та «Еталонні значення» на рисунку 4.1.

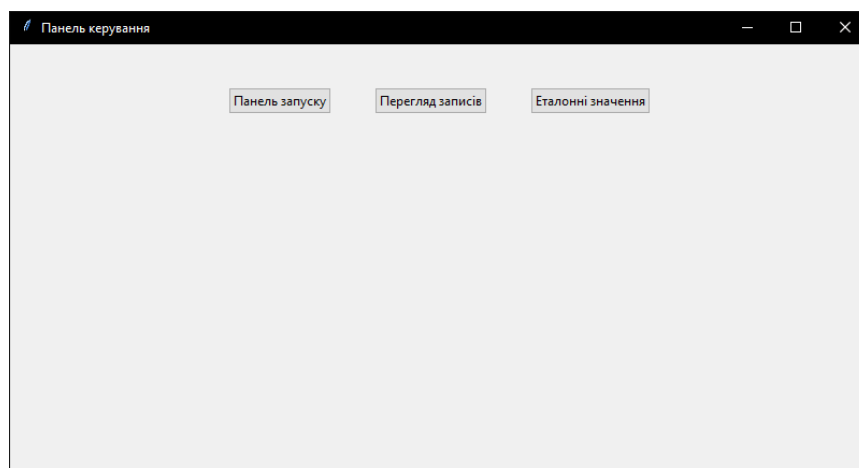


Рисунок 4.1 – Початкове меню програми

Щоб розпочати роботу з програмою, необхідно натиснути на будь-яку з цих кнопок, в залежності від потреби користувача в тій чи іншій дії. Якщо натиснути кнопку «Панель запуску», користувачу відкриються налаштування для роботи із симулятором радару, а також аналізатором сигналів, які слід налаштувати перед здійсненням запуску користувачеві. Вигляд панелі запуску, яка відкривається користувачеві можна переглянути на рисунку 4.2.

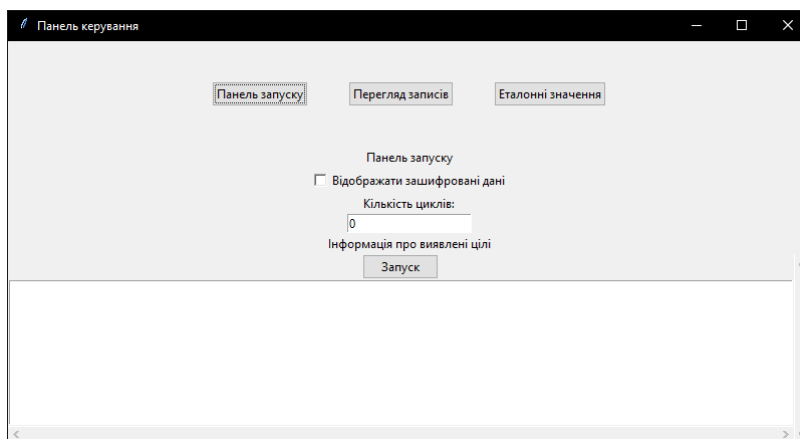


Рисунок 4.2 – Керування налаштуваннями програми перед запуском

В цьому вікні користувачеві по чергові стають доступні такі елементи інтерфейсу:

– «відображати зашифровані дані»: цей перемикач необхідний для демонстративності роботи механізму шифрування даних і, якщо перемикач увімкнено, то в текстовому полі внизу вікна під час роботи програми будуть дублюватися дані в зашифрованому вигляді. Приклад відображення таких даних на рисунку 4.3;

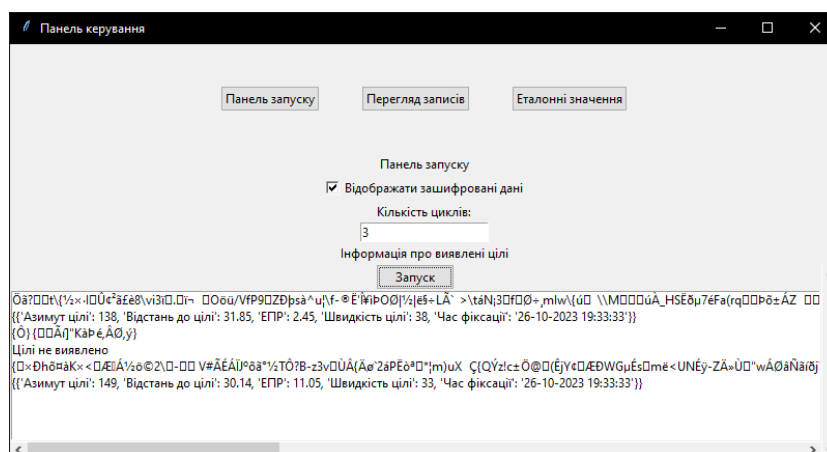


Рисунок 4.3 – Отримана інформація після запуску з відображенням шифрованих даних

– «кількість циклів»: оскільки робота програми на даному етапі відбувається із власним симулятором радіолокаційної станції, то задля налаштування його

робото-спроможності необхідно вказувати кількість циклів його роботи (за один цикл мається на увазі повне обертання тарілки радару на 360°). Кожен цикл відбувається запуск симулятора РЛС, під час дії якого можуть бути створені повітряні об'єкти, або ж ні. Задля демонстративності роботи програмного застосунку шанси на появу повітряних об'єктів значно збільшені відносно реальної ситуації у повітряному просторі. Задавати можна тільки цілі значення від 1 та до нескінченності;

– «запуск»: Останнім елементом на панелі запуску є сама кнопка запуску. Після натискання на неї починається робота програми враховуючи всі її попередні налаштування. Стається запуск генератора РЛС, відбувається шифрування даних, згодом їх обробка і порівняння з еталонними даними з бази даних, потім розшифрування та запис до таблиці в базі даних. Також здійснюється відображення даних про кожен цикл роботи аналізатора в текстовому вигляді в полі нижче цієї кнопки. В разі виявлення об'єктів у повітряному просторі виводиться спливаюче повідомлення в окремому вікні із звуковим сигналом, що виглядає як на рисунку . Якщо ж об'єктів у повітряному просторі не було знайдено, то користувач отримає повідомлення як на рисунку 4.4.

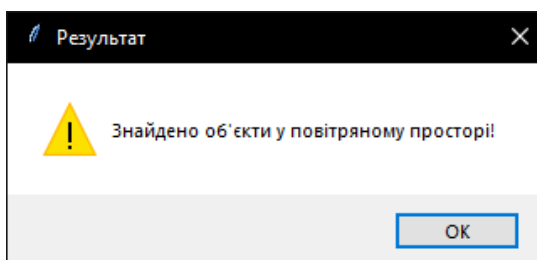


Рисунок 4.4 – Демонстрація повідомлення про знайдені об'єкти

Натиснувши «Перегляд записів» можна побачити усі записи за весь час роботи програми, як на рисунку 4.5, котрі зберігаються у окремій таблиці бази даних, які являють собою всі зафіксовані та проаналізовані раніше повітряні об'єкти. Програма відображає такі дані про ціль як:

– тип цілі;

- азимут цілі;
- відстань до цілі;
- ефективна площа розсіювання цілі;
- швидкість цілі;
- час фіксації цієї цілі.

Також можна побачити всі ці ж записи в MongoDB Compass та звідси їх можливо редагувати чи видаляти.

Панель запуску Перегляд записів Еталонні значення						
Інформація про виявлені цілі						
Тип цілі	Азимут цілі	Відстань до цілі	ЕПР	Швидкість цілі	Час фіксації	
Винищувач	35	65.9	10.79	681	19-10-2023 00:11:48	
Гелікоптер	159	92.47	2.46	110	19-10-2023 00:21:21	
Крилата ракета	279	9.96	3.15	227	19-10-2023 00:24:09	
Винищувач	90	35.35	7.06	426	19-10-2023 00:24:09	
Винищувач	331	67.8	16.75	557	19-10-2023 00:29:50	
ОТРК	89	11.31	1.25	500	19-10-2023 00:29:50	
Фронтовий бомбардувальник	238	65.69	49.16	317	19-10-2023 00:29:50	
Транспортний літак	29	10.43	66.62	132	19-10-2023 00:29:50	
Винищувач	227	70.7	7.94	420	19-10-2023 00:34:13	
Дипольний відбивач	298	81.51	65.76	99	19-10-2023 00:47:18	
Фронтовий бомбардувальник	133	23.94	21.34	413	19-10-2023 09:42:03	
Гелікоптер	265	41.15	16.98	108	19-10-2023 10:04:46	
Фронтовий бомбардувальник	335	50.62	11.68	370	22-10-2023 20:19:35	
Винищувач	123	18.23	61.96	471	22-10-2023 20:20:08	
Дипольний відбивач	258	91.75	19.55	91	24-10-2023 01:04:04	
Винищувач	119	81.22	10.38	533	24-10-2023 01:07:05	
Гелікоптер	231	64.88	9.5	92	24-10-2023 01:17:02	
Гелікоптер	210	84.76	8.32	59	24-10-2023 01:17:18	
Винищувач	43	20.25	2.66	520	24-10-2023 01:18:01	
Дипольний відбивач	32	95.76	14.93	79	24-10-2023 01:19:51	
Транспортний літак	299	35.04	21.28	172	24-10-2023 01:22:16	
Крилата ракета	3	92.96	8.27	201	24-10-2023 01:22:21	
Транспортний літак	182	39.07	61.31	247	24-10-2023 01:27:00	
Винищувач	125	51.62	12.35	461	24-10-2023 01:27:03	
Винищувач	311	96.4	19.44	452	24-10-2023 09:34:25	
Дипольний відбивач	95	1.7	14.06	80	24-10-2023 10:07:02	
Гелікоптер	269	58.7	0.46	101	24-10-2023 10:07:22	
Дипольний відбивач	103	7.16	16.69	79	25-10-2023 02:37:20	
Дипольний відбивач	138	31.85	2.45	38	26-10-2023 19:33:33	
Дипольний відбивач	149	30.14	11.05	33	26-10-2023 19:33:33	

Рисунок 4.5 – Демонстрація записів із бази даних у програмі

Натиснувши на останню кнопку користувачеві відкриється табличне відображення всіх еталонних даних про повітряні об’єкти, як на рисунку 4.6, що зберігаються в окремій таблиці бази даних. Саме враховуючи ці дані розроблений алгоритм здійснює свій аналіз і визначає дані про тип повітряного об’єкту й демонструє це користувачеві. Дані про об’єкти являють собою середні діапазони даних про подібні об’єкти із відкритих джерел. Це необхідно задля демонстративності оператору налаштувань у програмі, задля вирішення можливих спірних ситуацій із повітряним об’єктом, що програма визначила по одному, а оператор визначає по іншому.

Еталонні дані з бази даних MongoDB						
Тип цілі	ЕПР (min)	ЕПР (max)	Швидкість (min) м/с	Швидкість (max) м/с	Висота польоту (min) м	Висота польоту (max) м
Дипольний відбивач	20	20	0	100	100	15000
Винищувач	3	12	416	694	100	15000
Фронтний бомбардувальник	7	10	250	416	100	12000
Стратегічний бомбардувальник	13	20	222	305	100	15000
Транспортний літак	40	70	139	250	100	12000
Крилата ракета	0.1	0.1	180	240	20	150
ОТРК	0.15	1.6	278	556	10000	50000
Гелікоптер	0.5	2	42	111	100	5000

Рисунок 4.6 – Демонстрація еталонних значень із бази даних

Використовуючи інтерфейс бази даних є можливість додати нові еталонні об'єкти в базу даних, тим самим розширивши кількість повітряних об'єктів, що будуть враховуватися при аналізі програмою.

4.2 Робота користувача із базою даних

Оскільки основна частина роботи в розробленій програмі будується на аналізі і порівнянні записів з бази даних із інформацією від симулятора РЛС, то користувачу важливо мати також і повний доступ до бази даних. Це можна зробити зручним чином за допомогою застосунку MongoDB Compass.

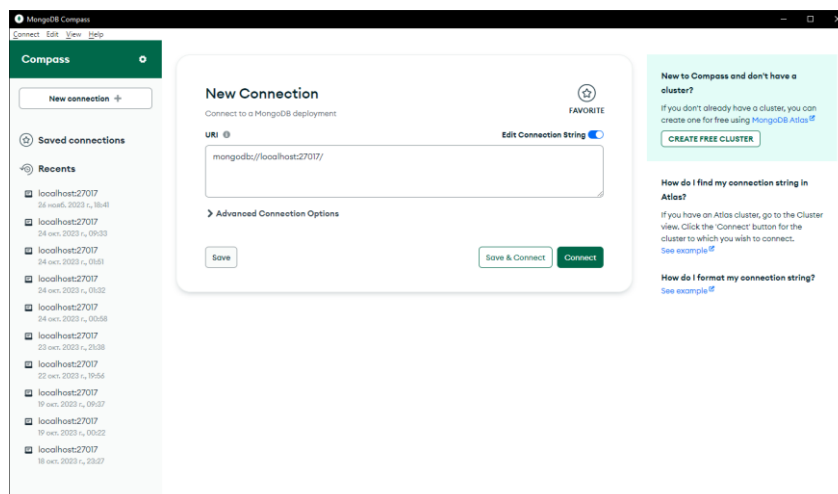


Рисунок 4.7 – Стартове вікно MongoDB Compass

Відкривши застосунок користувачу необхідно приєднатися до локального хоста як на рисунку 4.7.

Після чого користувачу відкриється структура всіх баз даних користувача, на рисунку 4.8.

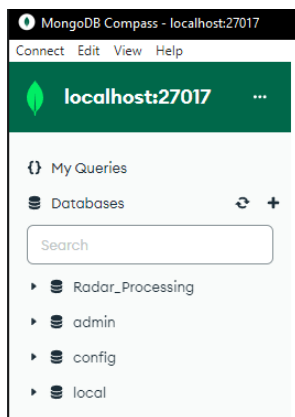


Рисунок 4.8– Структура баз даних користувача

Необхідно розкрити базу даних Radar_Processing, рисунок 4.9, котра і є базою даних до розробленого застосунку. В цій базі даних містяться дві колекції, що продемонстровано на рисунку : catalog та output_data.

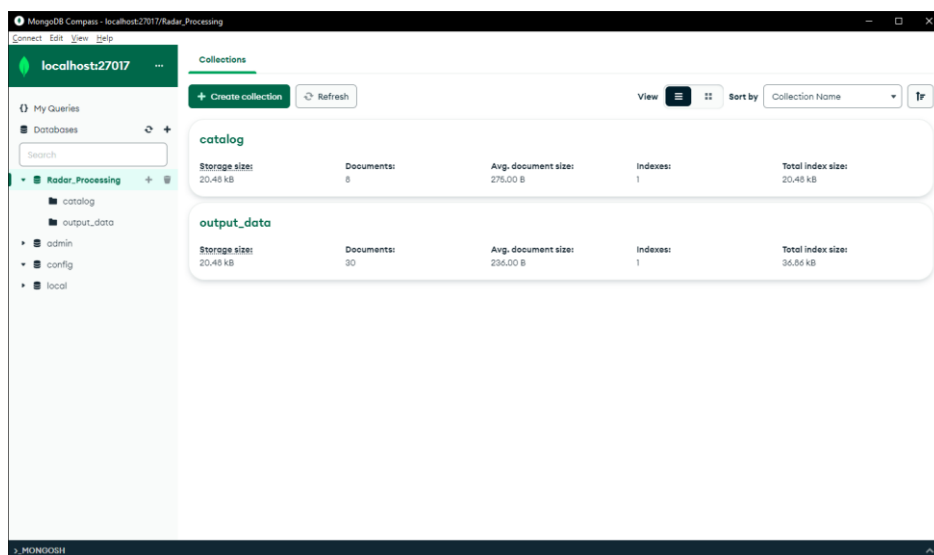


Рисунок 4.9 – База даних Radar_Processing

В колекції catalog містяться всі еталонні дані про повітряні об'єкти, із якими відбувається порівняння, під час ідентифікування типу об'єкту програмою. Загальний вигляд колекції на рисунку 4.10.

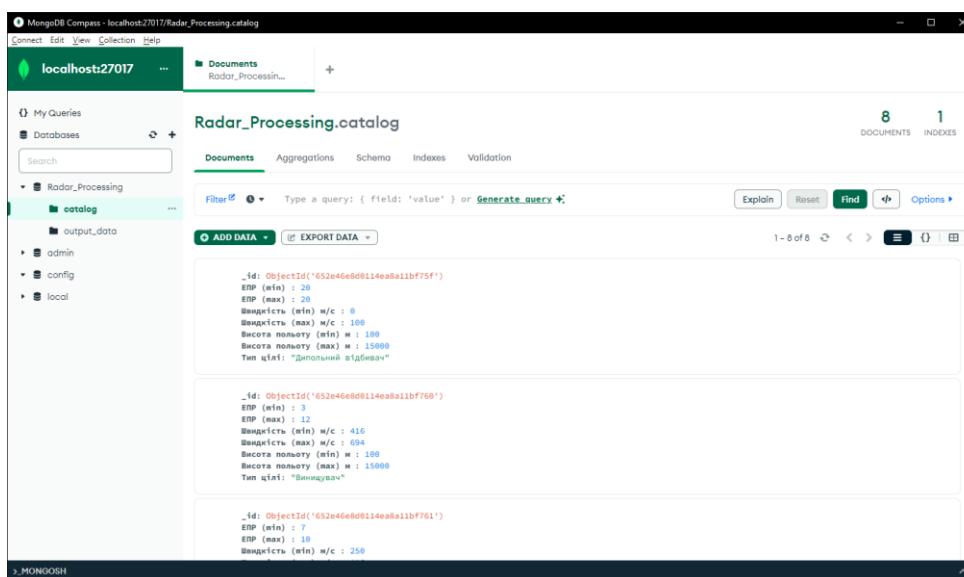


Рисунок 4.10 – Колекція catalog

Поля цієї колекції зберігають діапазони ефективної площі розсіювання, швидкостей польоту, висот польоту, а також тип цієї цілі у вигляді строки. Кожне поле цієї колекції можна відредагувати, скопіювати, клонувати чи видалити. Це може бути потрібним, щоб редагувати дані про загальновідомі повітряні об'єкти або ж додавати нові. Інструменти модифікування представлені на рисунку 4.11. Користувачу не обов'язково знати скрипти мови Mongo, щоб зрозуміти як оперувати записами у БД, що є дуже практичним і зручним рішенням.

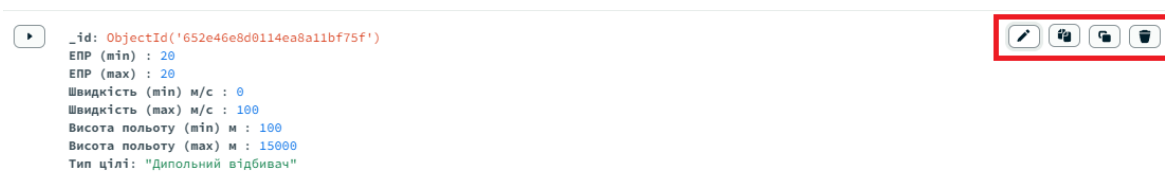


Рисунок 4.11 – Редагування поля в БД

Також маємо колекцію `output_data`, котра зберігає всі дані про помічені й проаналізовані програмою повітряні об'єкти за весь час її функціонування на рисунку 4.12. Із записами цієї колекції також можна оперувати як із записами у попередній колекції. Також, функціонал MongoDB Compass дозволяє здійснювати пошук записів за якимось параметром, змінювати спосіб їх відображення, сортувати та інше.

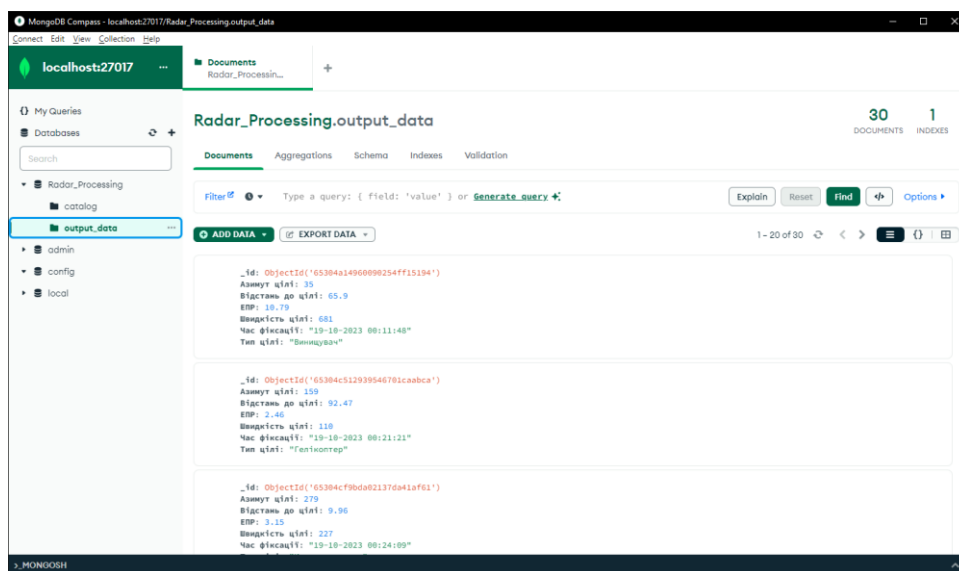


Рисунок 4.12 – Колекція `output_data`

Кожен запис в цій таблиці звісно має свій унікальний ідентифікатор, й тому може бути знайденим за необхідністю по ньому.

4.3 Інсталяція програми

Інсталяцію розробленого застосунку включає в себе налаштування простору в Python, а також розгортання робочої бази даних. Було обрано використовувати Python версії 3.11. Для роботи з базою даних обрано застосунок MongoDB версії 1.40.4. Установка включає в себе 3 етапи.

4.3.1 Інсталяція Python 3.11

Для початку необхідно перейти на офіційний веб-сайт Python за посиланням <https://www.python.org> та натиснути кнопку Download. Офіційний сайт виглядає так, як представлено на рисунку 4.13.

В розділі завантажень необхідно обрати версію 3.11.6 та завантажити файл python-3.11.6.exe, після чого його слід запустити, щоб почати встановлення. Під час інсталяції обов'язково оберіть опцію "Add Python 3.11 to PATH".

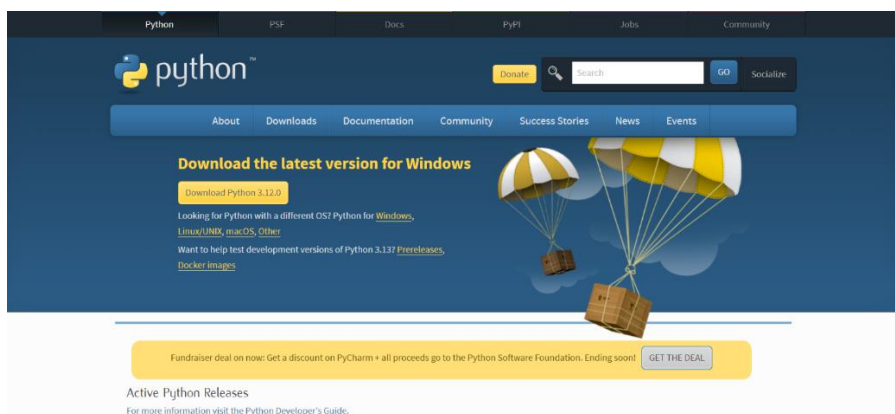


Рисунок 4.13 – Офіційний сайт Python

Це спростить роботу з Python у командному рядку, що нам необхідно буде далі. Потім слід обрати місце встановлення та натиснути Install now. По завершенню інсталяції слід відкрити командний рядок, щоб перевірити встановлену версію Python. Для цього слід ввести команду `python --version` або `python -V`. В разі успішного виведення версії Python, як на рисунку 4.14, його встановлення можна вважати завершеним.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.3693]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

F:\Documents\Stupid_Radar>python --version
Python 3.11.0

F:\Documents\Stupid_Radar>
```

Рисунок 4.14 – Перевірка встановлення Python у консолі

В разі, якщо версія Python не викликається і відображається помилка «Python Was Not Found Run Without Arguments to Install From the Microsoft Store Error» слід виконати наступні дії:

1. Перейти до шляху встановлення Python 3.11, як на рисунку 4.15, та скопіювати цей шлях.

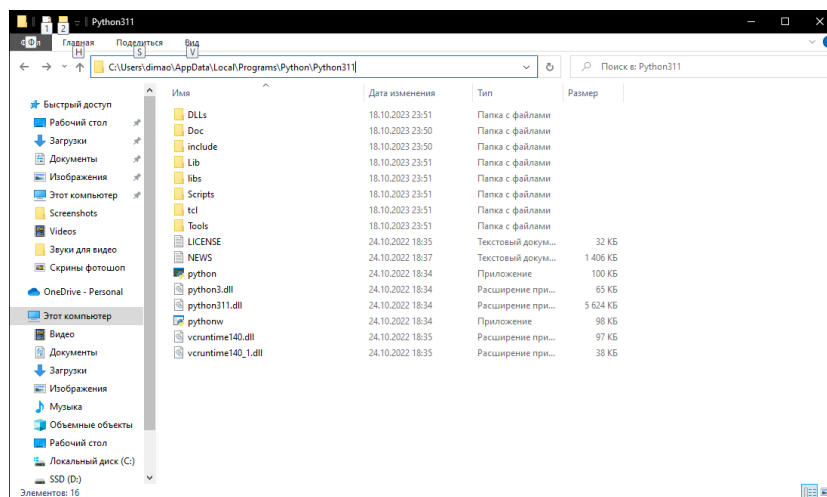


Рисунок 4.15 – Шлях встановлення Python

2. Відкрити налаштування Windows, далі «Система», «Про програму» і тут знайти пункт «Додаткові параметри системи», що на рисунку 4.16.

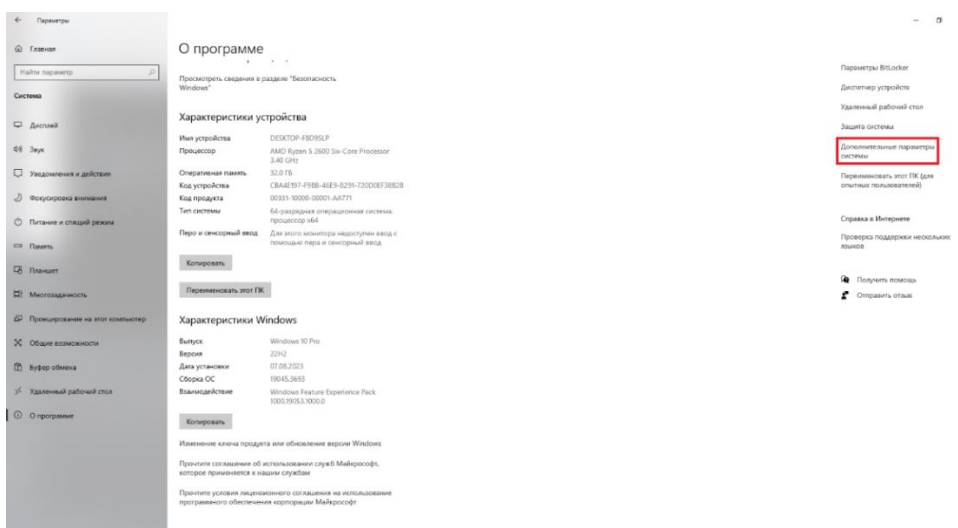


Рисунок 4.16 – Шлях в налаштуваннях до додаткових параметрів системи

3. Відкрити додаткові параметри системи та в розділі «Додатково» натиснути «Змінні середовища».

4. В системних змінних знаходимо змінну Path, виділяємо її та натискаємо кнопку «Змінити». З'явиться нове вікно, що на рисунку 4.17.

5. У відкритому вікні потрібно натиснути кнопку «Створити» та вставити шлях до місця встановлення Python 3.11 на вашому комп'ютері. Також слід додати ще один шлях – до папки Scripts. Після підтвердження всіх змін можна перевіряти робото-спроможність Python у консолі.

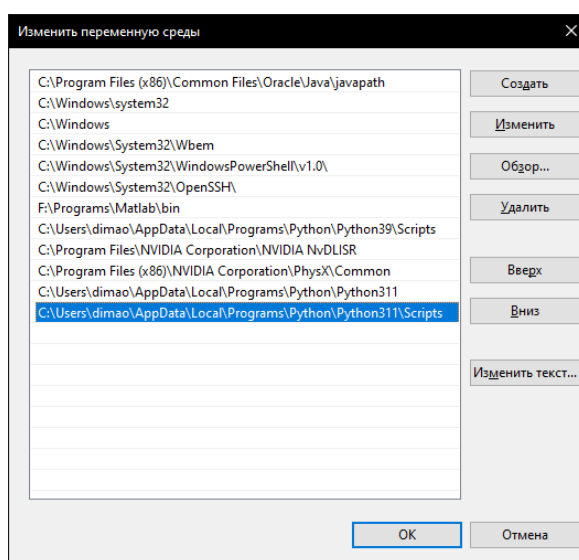


Рисунок 4.17 – Заміна параметрів змінної Path

Щоб запускати програми, розроблені мовою Python, можна створювати та запускати Python-скрипти у обраному середовищі розробки або ж використовувати IDLE, що на рисунку 4.18, яке автоматично інсталується разом з Python.

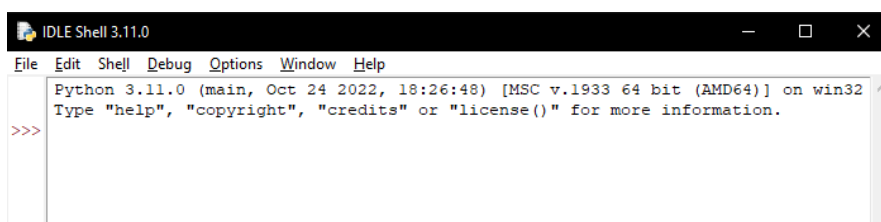


Рисунок 4.18 – IDLE до мови програмування Python

4.3.2 Інсталяція й налаштування MongoDB

Щоб встановити СКБД MongoDB на комп'ютер під керуванням ОС Windows слід виконати наступні кроки:

1. Перейти на офіційний сайт MongoDB за посиланням <https://www.mongodb.com/try/download/community>, вигляд якого представлено на рисунку 4.19, та завантажити Community Server для Windows.

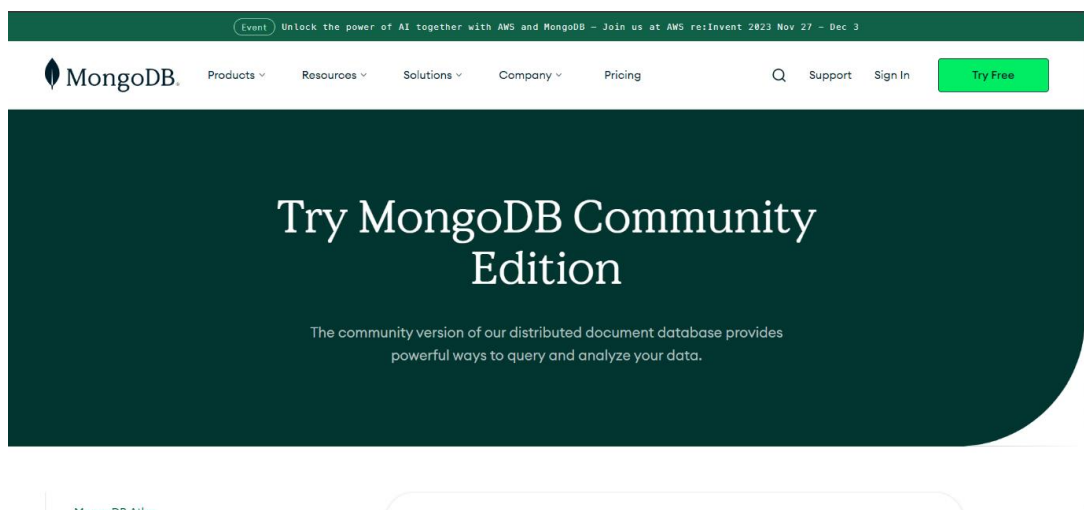


Рисунок 4.19 – Офіційний сайт MongoDB

2. Далі слід запустити завантажений інсталятор. В процесі установки слід обрати опцію Complete та включити опцію Install MongoDB as a service. Після чого слід вказати директорію встановлення програми та дочекатися завершення установки.

3. Останнім потрібно налаштувати MongoDB. Створіть новий каталог, наприклад, C:\data\db. MongoDB використовує цей каталог для зберігання даних. Додайте MongoDB в системний шлях: Додайте шлях до mongod.exe в системну змінну PATH. Зазвичай це шлях до інсталяції MongoDB, наприклад, C:\Program Files\MongoDB\Server\{version}\bin.

4. Запустіть MongoDB: Відкрийте командний рядок та введіть команду mongod, як на рисунку 4.20. MongoDB повинна успішно запуснитися.

```

C:\Users\91749>mongod -version
db version v6.0.1
Build Info: {
  "version": "6.0.1",
  "gitVersion": "32f0f9c88dc44a2c8073a5bd47cf779d4bfdee6b",
  "modules": [],
  "allocator": "tcmalloc",
  "environment": {
    "distmod": "windows",
    "distarch": "x86_64",
    "target_arch": "x86_64"
  }
}

```

Рисунок 4.20 – Перевірка командою mongod

Наступним необхідно встановити GUI MongoDB Compass для зручності оперування з базою даних. Виконайте наступні кроки:

1. Перейдіть на офіційний сайт MongoDB Compass, на рисунку 4.21 за посиланням <https://www.mongodb.com/try/download/compass> і завантажте версію для Windows.

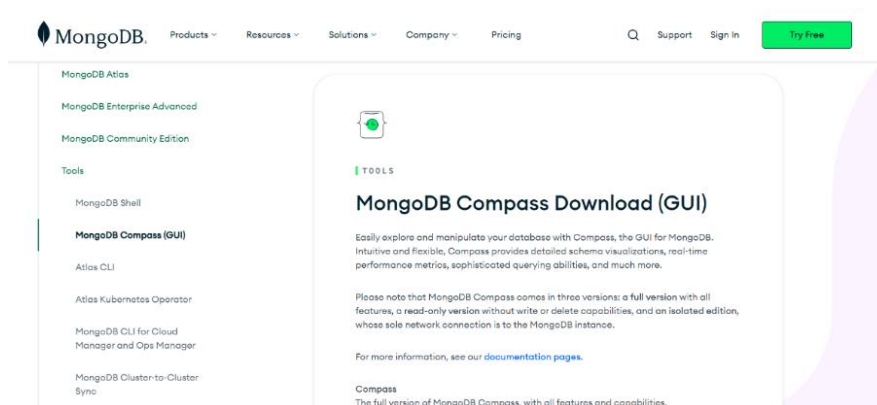


Рисунок 4.21 – Офіційний сайт MongoDB Compass

2. Запустіть інсталятор MongoDB Compass, який ви завантажили. Слідуйте інструкціям інсталятора, обираючи потрібні опції. Продовжуйте інсталяцію та завершіть її.

Тепер MongoDB та MongoDB Compass повинні бути встановлені на вашому комп'ютері. Можна використовувати MongoDB Compass для візуального адміністрування та роботи з базами даних MongoDB.

Залишається лише створити базу даних використовуючи MongoDB Compass та експортувати колекцію еталонних значень повітряних об'єктів, яку можна завантажити за посиланням https://drive.google.com/file/d/1aRxxzsd1h_ePYNkc6B8QHkTsmTcJdnm3/view . Після виконання всіх дій, що вказані в інструкції в цьому підпункті, користувач зможе працювати із розробленим програмним забезпеченням.

4.3.2 Інсталяція і запуск програми через консоль

Щоб встановити та запустити розроблену програму, для обробки сигналів з індикаторів кругового огляду, слід завантажити архів за посиланням: https://drive.google.com/file/d/1WeYTZE92t9rMLzWgBW_yQtpV89TpdBIf/view?usp=sharing . Далі потрібно розархівувати його у зручному для вас місці. Після цього слід відкрити консоль із папки, де ви розархівували програму, як на рисунку 4.22.

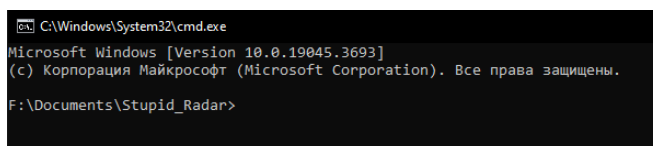


Рисунок 4.22 – Відкриття консолі в папці з проектом

Далі для запуску програми потрібно викликати пакетний файл Launch_program.bat [45], що показано на рисунку 4.23.

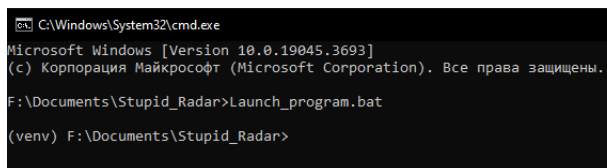


Рисунок 4.23 – Виклик пакетного файлу Launch_program.bat

Тепер можна прописати назву скрипту main.py і програму буде запущено, рисунок 4.24.

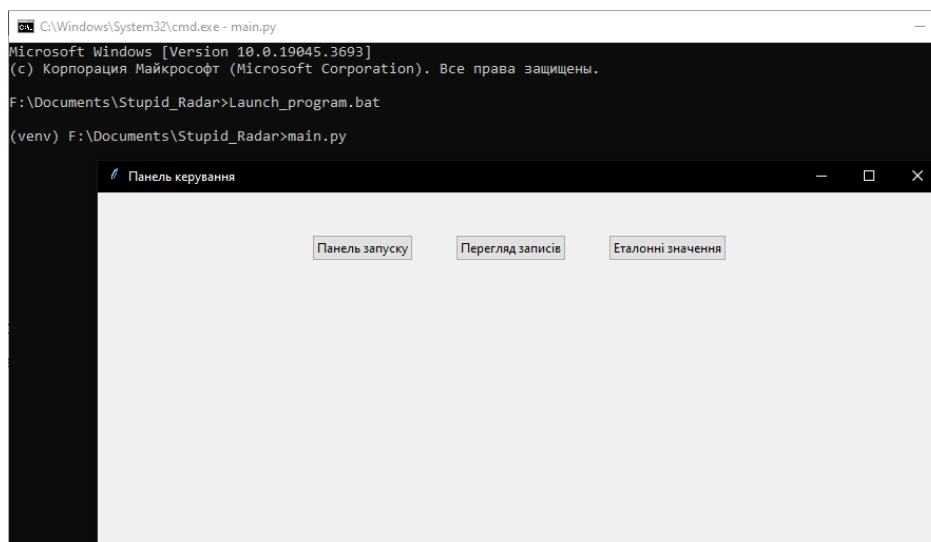


Рисунок 4.24 – Успішний запуск програми через консоль

4.4 Шляхи покращення розробленого програмного продукту

Розроблений програмний продукт має значні перспективи щодо його покращення. В поточній версії він представлений віконним застосунком без використання комп'ютерної графіки. За рахунок докладання додаткових зусиль та часу можна провести розширення функціоналу розробленого програмного забезпечення наступним чином:

1. Розробка в базі даних переліку радіолокаторів, що використовуються збройними силами України. На даний момент симулятор РЛС, що використовується у програмному забезпеченні за своїми характеристиками є приблизно подібним до реального радару П-18 «Малахит», що на рисунку 4.25, що був розроблений у СРСР і використовується збройними силами. Необхідно розширити перелік доступних РЛС відповідно до сучасних реалій, адже за 2022-2023 роки Україна отримала чимало західних зразків радіолокаторів.



Рисунок 4.25 – Радіолокаційна станція П-18

2. Додати елементи комп'ютерної графіки до розробленого програмного продукту. А саме в першу чергу додати відображення карт з простим та супутниковим шаром. На цих картах надати можливість відмічати візуально дислокації інших радіолокаційних станцій, з вказанням їх типу та області роботи, задля того щоб сформувати цілу систему радіолокації на території України. Додати візуальне відображення на картах помічених об'єктів у повітряному просторі, із вказанням детальної інформації про ці об'єкти (висота, швидкість, напрямок польоту, тип). Приклади демонстрації цієї ідеї представлені на рисунках 4.26 та 4.27.

3. Розробити інструменти для синхронізації мережі комп'ютерів, що використовують це програмне забезпечення, задля оновлення в режимі реального часу даних про стан повітряного простору, що необхідно для своєчасного реагування операторів цієї системи.

4. Оскільки подібне програмне забезпечення потребує великого рівня захищеності, то необхідно додати механізм авторизації у систему, щоб уникнути підключень від сторонніх осіб. Також, додати механізми шифрування даних на всьому шляху їх обігу. Зберігати в базі даних всю інформацію в зашифрованому вигляді й видавати розшифровані дані лише користувачу, який має авторизований доступ до них.

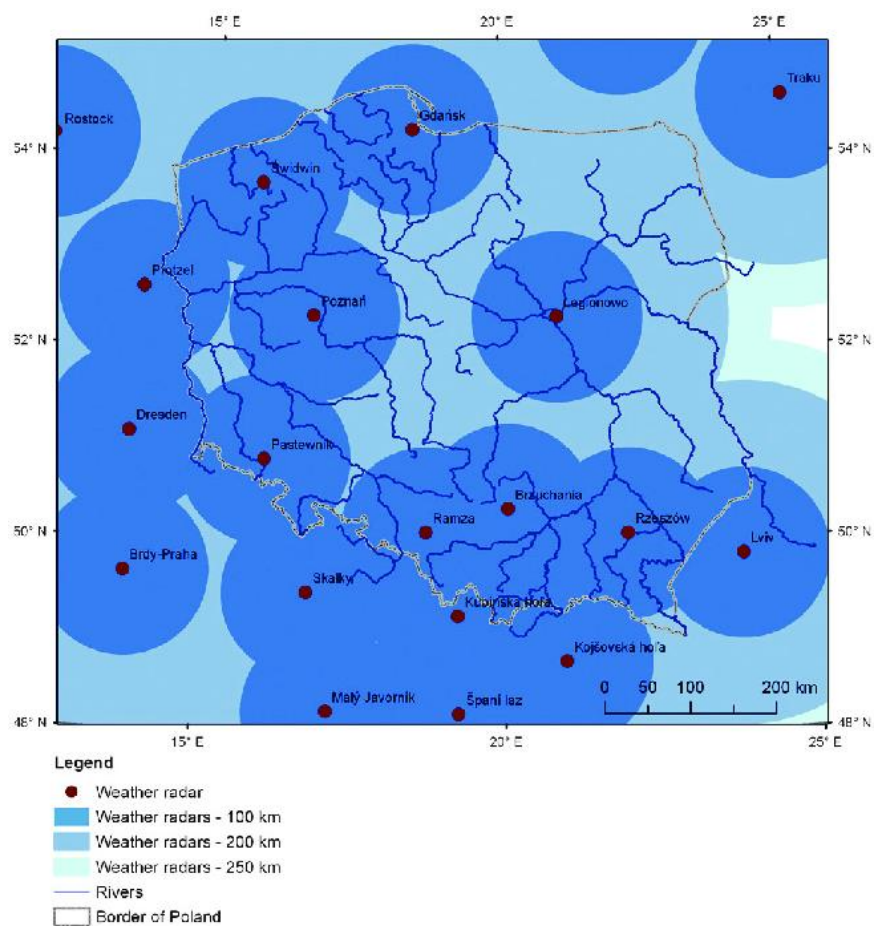


Рисунок 4.26 – Демонстрація комплексу радарів у графічній системі 1

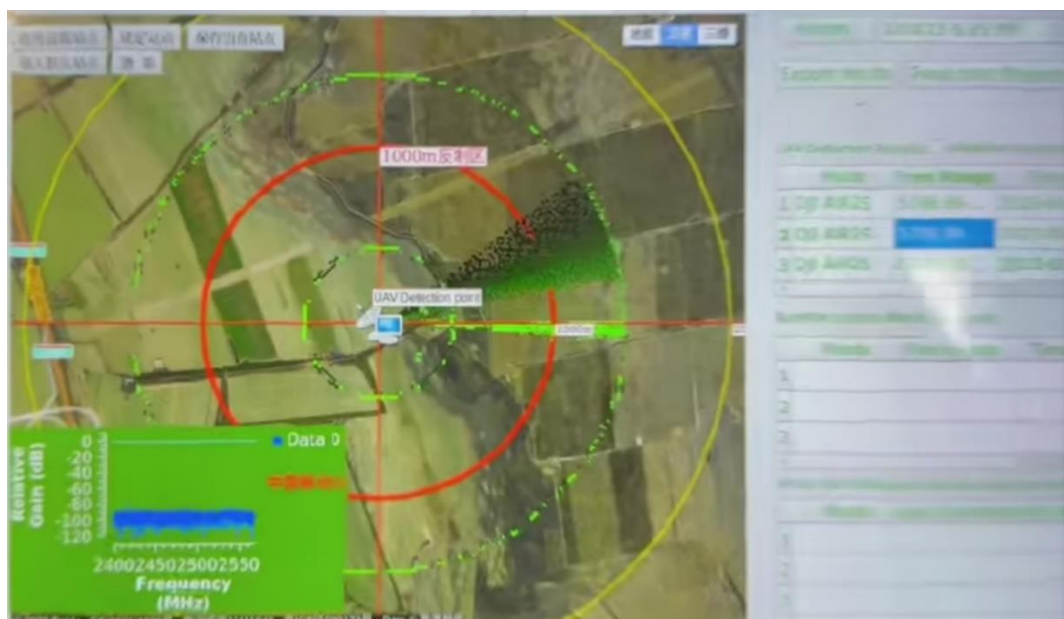


Рисунок 4.27 – Демонстрація комплексу радарів у графічній системі 2

Висновки до розділу 4

У даному розділі продемонстровано можливості розробленого програмного забезпечення, розписані вказівки щодо користування її інтерфейсом. Продемонстровано увесь користувацький інтерфейс програми та розписані дії кожної із кнопок та прапорців. Надано рекомендації щодо користування програмою для користувача. Також була представлена детальна інструкція із ілюстраціями по встановленню цього програмного забезпечення та всіх додаткових програмних засобів, що потрібні для правильного функціонування розробленого програмного забезпечення.. В кінці розділу вказане власне бачення способів задля покращення та удосконалення поточного програмного продукту й отримання у майбутньому більшого функціоналу подібної системи.

5 Розроблення стартап-проекту

Для створення стартапу необхідно провести аналіз маркетингових аспектів, задля подальшого впровадження створеного проекту. Для цього слід представити ряд таблиць із розбором характеристик стартап проекту.

5.1 Опис ідеї стартап-проекту

В цьому пункті було проаналізовано:

- зміст запропонованої ідеї;
- напрями її застосування;
- користувацькі вигоди.

Ці позиції подано у вигляді таблиці 5.1

Таблиця 5.1 — Загальні дані стартап-проекту

Зміст запропонованої ідеї	Напрямки її застосування	Користувацькі вигоди
Створення єдиної системи для моніторингу стану повітряного простору всіма наявними РЛС.	1. Проектування плану розташування РЛС по території.	1. Зручність в плануванні операцій та розміщенні подальших елементів.
	2. Забезпечення онлайн моніторингу всієї обстановки повітряного простору.	2. Цілодобова усвідомленість обстановки, забезпечення безпеки населення.

В таблиці 5.2 проводиться порівняльний аналіз переваг ідеї, порівняно із пропозиціями конкурентів.

Таблиця 5.2 — Визначення характеристик ідеї проекту

№ з/п	Економічні характеристик и	Концепції конкурентів		Слабкі сторони	Нейтральні сторони	Сильні сторони
		Власний проект	Delta			
1	Швидкодія	Так	Так		Так	
2	Інтегрованість	Можна інтегрувати до будь-якого радару у майбутньому	Точні дані засекречно			Так
3	Інтерфейс	Взаємодія відбувається в віконному застосунку	Додаток має елементи комп'ютерної графіки		Так	
4	Спеціалізація системи	Застосунок спеціалізується на роботі з РЛС	Застосунок є загально-військовою онлайн-картою			Так

Даний перелік властивостей та характеристик цього товару вважається підґрунтям щодо формування його конкурентоспроможності.

5.2 Технологічний аудит ідеї проекту

Цей розділ представляє аудит технологій, які необхідні для реалізації ідеї проекту. Для початку потрібно проаналізувати технологічну здійсненність проекту. Технологічна здійсненність ідеї проекту представлена на таблиці 5.3.

Таблиця 5.3 — Технологічна здійсненність ідеї проекту

№ з/п	Ідея проекту	Технології реалізації	Наявність технологій	Доступність технологій
1	Інтерфейс користувача	Мова програмування Python	Наявна	Умовна безкоштовно
2	Розширювана база даних	MongoDB	Наявна	Умовна безкоштовно
3	Взаємодія додатку з базою даних	Мова програмування Python, JSON-файли	Наявна	Умовна безкоштовно

Після аналізу таблиці можна підвести висновок, що є можливість технологічної реалізації проекту. Для цього було вибрано технології: pymongo, tkinter, numpy, hashlib, JSON.

5.3 Моніторинг ринкових можливостей запуску стартап-проекту

Для реалізації й впровадження проекту на ринку необхідно ідентифікувати можливі загрози, які можуть виникати під час впровадження цього проекту, що дозволить стратегічно спланувати розвиток проекту з урахуванням аналізу ринкового середовища, потреб можливих клієнтів та конкурентних пропозицій інших проектів. Аналізування ринкових можливостей проекту являє собою багатоетапний процес. Перший етап охоплює в себе аналіз попиту, що включає в себе оцінку наявності попиту, його обсягу та динаміки розвитку ринку. Ці показники представлено в таблиці 5.4.

Таблиця 5.4 — Попередня характеристика потенційного ринку стартап-проекту

№ з/п	Показники стану ринку (найменування)	Характеристика
1	Ринкова сегментація	Авіаційна промисловість, облікові служби, рятувальні органи та інші сегменти, які можуть використовувати систему ідентифікації повітряних об'єктів.
2	Ринкова динаміка	Зростання завантаженості повітряного простору нові тенденції в галузі авіації та розвиток технологій ідентифікації об'єктів в повітрі.
3	Слабкі сторони	Можливі труднощі з визначенням стандартів індустрії, конкуренція з боку вже встановлених гравців.
4	Потенційні клієнти	Збройні сили України, приватні авіаційні компанії
5	Стратегія маркетингу	Вивчення та рекламування переваг продукту, партнерства з виробниками РЛС, участь у виставках та заходах індустрії.

В результаті аналізу таблиці робимо висновок, що проект має перспективи виходу на ринок.

На другому етапі слід ідентифікувати потенційні групи клієнтів. Результати цього етапу представлені у таблиці 5.5. Результати цього етапу формують перелік вимог до продукту.

Таблиця 5.5 — Характеристика можливих клієнтів стартап-проекту

№ з/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності між поведінкою різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Створення системи для обробки сигналів з різноманітних РЛС	РЛС-системи	Різні клієнти мають різне обладнання, з яким система повинна працювати	Висока стабільність роботи; Програма навчання персоналу; Регулярні оновлення системи.

Після визначення потенційних груп клієнтів, потрібно провести аналіз ринкового середовища, щоб мати уявлення про підводні камені на шляху впровадження проекту. До нього аналізу входять: фактори загроз і фактори можливостей.

Фактори подаються у вигляді двох таблиць з колонками:

- фактор;
- зміст загроз/можливостей;
- можлива реакція компанії.

Фактори загроз та можливостей представлено в таблицях 5.6 і 5.7 відповідно.

Таблиця 5.6 — Фактори загроз

№ з/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Відсутність графіки	Недостатність інтерфейсу застосунку, через відсутність реалізації інструментів комп'ютерної графіки	Проведення додаткової роботи на покращення застосунку

Таблиця 5.7 – Фактори можливостей

№ з/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Інтеграція з системами безпеки	Зростання використання систем безпеки, де важлива роль відведена точній ідентифікації літаків, гелікоптерів та крилатих ракет.	Розширення можливостей ідентифікації для взаємодії з різними системами безпеки та підвищення ефективності військових операцій.
2	Збільшення військового застосування радарів	Збільшення потреб у військовому застосуванні радарів для виявлення та ідентифікації повітряних об'єктів.	Акцент на розробці та вдосконаленні алгоритмів ідентифікації

Далі необхідно провести оцінку пропозиції, загальних рис конкурентного середовища проекту на ринку, що визначено у таблиці 5.8. Задля проведення цього аналізу використовувалися два методи: ступеневий аналіз та аналіз конкуренції в галузі (згідно з моделі М. Портера). Аналіз за методологією М. Портера потрібен для забезпечення більш детального вивчення умов конкуренції цього проекту.

Результати ступеневого аналізу конкурентного середовища представлені у таблиці 5.8.

Таблиця 5.8 — Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії щоб бути конкурентоспроможними)
1. Вказати тип конкуренції монополія/олігополія/ монополістична/чиста	олігополія	Підтримка якості продукту, постійні нововведення, вдосконалення
2. За рівнем конкурентної боротьби: локальний/національний	національний	Адаптація продукту під міжнародні стандарти
3. За галузевою ознакою — міжгалузева/внутрішньогалузева	внутрішньогалузева	Продукт має конкретно спеціалізацію
4. Конкуренція за видами товарів: — товарно-родова — товарно-видова — між бажаннями	товарно-видова	Програмний продукт має враховувати можливості конкурентів

Кінець таблиці 5.8

5. За характером конкурентних переваг: цінова / нецінова	нецінова	Використання більш доступних технологій у порівнянні з тими, які застосовують конкуренти, за умови, що ці технології відповідають необхідним стандартам якості.
6. За інтенсивністю: марочна/не марочна	марочна	Надання функцій, які не надають конкуренти

Нижче приведено аналіз конкуренції за Портером у таблиці 5.9.

Таблиця 5.9 — Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	-	Delta	Визначити фактори сили в постачальників	Контроль якості	Фактори загроз з боку замінників
Висновки:	Прямі конкуренти відсутні	Є можливості виходу на ринок, бо існуючі рішення не надають потрібних переваг	Постачальники підлаштовують ся під ринок	Клієнти вказують побажання щодо функціоналу програмного забезпечення	Особливі х обмежень немає

Враховавши висновки, що були отримані в результаті ступеневого аналізу та аналізу за М. Портером, можна зробити висновок що є можливість входу на ринок, якщо враховувати конкурентну обстановку. На наступному етапі аналізу проводиться визначення факторів конкурентоспроможності.

На основі детального аналізу конкурентної ситуації, врахування особливостей ідеї проекту, вимог споживачів до продукту та факторів маркетингового середовища визначається перелік ключових факторів, що визначають конкурентоспроможність. Аналіз цих факторів представлено в таблиці 5.10.

Таблиця 5.10 — Обґрунтування факторів конкурентоспроможності

№ з/п	Фактор конкурентоспроможності	Обґрунтування фактору
1	Технологічна перевага	Застосування передових технологій у системі дозволяє точніше та швидше ідентифікувати повітряні об'єкти. Технологічна перевага створює високий рівень ефективності та відповідності вимогам клієнтів, що є ключовим фактором у конкурентному середовищі.
2	Інтеграція з існуючими системами безпеки	Здатність легко інтегруватися з існуючими системами безпеки робить продукт привабливим для клієнтів, які вже використовують інші рішення. Це може зменшити витрати на оновлення та поліпшення систем у клієнтів.

За визначеними факторами конкурентоспроможності проводиться аналіз сильних та слабких сторін стартап-проекту. Аналіз проводиться у вигляді виставлення балів кожному фактору. Дані представлено у таблиці 5.11.

Таблиця 5.11 — Порівняльний аналіз сильних та слабких сторін проекту

№ з/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів конкурентів у порівнянні з даним продуктом						
			-3	-2	-1	0	1	2	3
1	Використання сучасних технологій	16					+		
2	Пропозиція функцій, відсутніх у конкурентів	18		+					

Наостанок, для оцінки ринкових можливостей для впровадження проекту. потрібне проведення SWOT-аналізу, що є матрицею аналізу сильних (Strengths) та слабких (Weaknesses) сторін, а також загроз (Threats) та можливостей (Opportunities) на основі виявлених ринкових загроз та можливостей, а також сильних і слабких сторін проекту.

Аналіз SWOT стартап-проекту проведено на таблиці 5.12

Таблиця 5.12 — SWOT-аналіз стартап-проекту

Сильні сторони: Здатність обробки сигналів з РЛС та аналізу дозволяє мати технологічну перевагу у виявленні та ідентифікації об'єктів у повітрі.	Слабкі сторони: Обробка сигналів з РЛС та розробка високотехнологічної системи може вимагати значних витрат на дослідження та розробку.
Можливості: Співпраця з виробниками РЛС може забезпечити доступ до нових технологій та ринкових можливостей.	Загрози: Зміни у технологіях або відмова від застарілих РЛС можуть потребувати постійних оновлень та адаптацій.

Після завершення проведення SWOT-аналізу для стартап-проекту вже формуються альтернативні стратегії ринкової поведінки, щоб ввести його на ринок. Також здійснюється оцінка приблизного часу для ринкової реалізації, враховуючи

проекти конкурентів, які також можуть бути введені на ринок. Обрані стратегії аналізуються в контексті їхніх строків та ймовірності успішної реалізації.

Таблиця 5.13 — Альтернативи ринкового впровадження стартап-проекту

№ з/п	Альтернатива (орієнтовний комплекс заходів) поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Самостійний вихід на ринок	Середня, через значні витрат на маркетинг та рекламу.	Високі, оскільки потрібен час на побудову свідомості про бренд та завоювання ринкової позиції.
2	Розширення асортименту продукції для існуючих клієнтів	Висока, оскільки це може бути менш витратним способом розширення.	Середні, залежать від часу на дослідження та внедрення нових продуктів.
3	Фокус на регіональному впровадженні	Висока, оскільки область фокусу може зменшити витрати.	Середні, оскільки впровадження в обмеженому регіоні може бути швидшим.

Найбільше нам підходить альтернатива №2, через високі шанси на отримання ресурсів, за рахунок покращення функціоналу продукції за середній час.

5.4 Розроблення ринкової стратегії продукту

Процес створення ринкової стратегії продукту включає декілька етапів, з першим кроком, який полягає в уточненні стратегії охоплення ринку. Результатом

цього етапу є створення таблиці, в якій міститься опис всіх цільових груп споживачів та їх критерії.

Інформація, отримана на першому етапі розробки ринкової стратегії продукту, представлена у таблиці 5.14.

Таблиця 5.14 — Вибір цільових груп потенційних споживачів

№ з/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачі в сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу в сегмент
1	Потенційними споживачами є збройні сили та ПВК	Висока	Високий	Низька	Незначна кількість конкурентів в Україні
Які цільові групи обрано: військові, що обслуговують РЛС					

По завершенню аналізу потенційних груп споживачів були визначені цільові сегменти, які саме отримають пропозицію щодо даного товару, та було встановлено стратегію охоплення ринку, а саме стратегію диференційованого маркетингу, що передбачає взаємодію компанії з кількома сегментами.

На наступному етапі розроблено базову стратегію розвитку, а інформацію про цей етап занесено до таблиці 5.15.

Таблиця 5.15 — Визначення базової стратегії розвитку

№ з/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
-------	--------------------------------------	---------------------------	--	---------------------------

Кінець таблиці 5.15

1	Орієнтація поточної моделі на військові потреби	Диференціації	Можливість налаштувати систему під потреби користувачів	Спеціалізації
---	---	---------------	---	---------------

Наступним кроком необхідно обрати стратегію конкурентної поведінки. Її результати представлено в таблиці 5.16.

Таблиця 5.16 — Визначення базової стратегії конкурентної поведінки

№ з/п	Чи є проект “першопрохідцем” на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Обидва шляхи	Ні	Стратегія зняття конкурентної ніші

Використовуючи вибрані сегменти для постачальника та продукту, а також врахувавши обрану базову стратегію розвитку, а також стратегії конкурентної поведінки, необхідно розробити стратегію позиціонування. Ця стратегія включає в себе визначення ринкової позиції, яка може асоціюватися користувачами із торгівельною маркою. Відомості про цей етап деталізовано представлені в таблиці 5.17.

Таблиця 5.17 — Визначення стратегії позиціонування

№ з/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Висока стабільність та безперебійність роботи, захищеність інформації	Стратегія диференціації	Позиція на основі унікальності власного продукту	Технологічна інноваційність, ефективність та оптимізація, клієнтська орієнтовність

В результатах здійсненої роботи в даному підрозділу було сформовано систему стратегічних вирішень щодо поведінки компанії на ринку. Саме ця система визначає напрямки діяльності в ринковому середовищі у стартап-компанії.

5.5 Розроблення маркетингової програми проекту

Першим етапом у розробці маркетингової програми є формування концепції товару. Результатом цього процесу є таблиця 5.18, яка узагальнює висновки з аналізу конкурентоспроможності продукту.

Таблиця 5.18 — Визначення ключових переваг концепції потенційного товару

№ з/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Швидка інтеграція системи	Новаторський підхід до забезпечення безпеки повітряного простору країни	Відсутність прямих конкурентів

Визначивши основні переваги, розробляється трирівнева маркетингова модель продукту. В цю модель входить уточнення ідеї продукту, його фізичні компоненти та характеристики. Ці відомості представлені у таблиці 6.19.

Таблиця 5.19 — Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Програмне забезпечення для аналізу сигналів радіолокаторів		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	1. Розмір програмного забезпечення	Нм	Тх
	2. Інтерфейс користувача	Нм	Тх
	3. Модульність	Нм	Тх
	Якість: стандарти, нормативи, тестування, тощо		
	Пакування: відсутнє		
	Марка: Future Radar		
III. Товар із підкріпленням	До продажу: відсутнє Після продажу: постійна підтримка розробником		

Наступним необхідно визначити цільові межі, якими необхідно керуватись при встановленні ціни на потенційний товар. Дані представлені у таблиці 5.20.

Таблиця 5.20 — Визначення меж встановлення ціни

№ з/п	Ціни товарів-замінників	Ціни товарів-аналогів	Доходи групи споживачів	Найменша та найнижча ціна на товар/послугу
1	500000-1500000	Високий	Високий	250000-700000

Потім проводиться визначення системи оптимального збуту, в рамках якої прийнято рішення, що відображено у таблиці 5.21.

Таблиця 5.21 — Формування системи збуту

№ з/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Клієнт відразу має повний доступ	Активна рекламна компанія	Прямий канал збуту	Адаптація до потреб цільової аудиторії

Останнім етапом маркетингової програми проекту є розробка концепції маркетингових комунікацій, яка виникає з вибраного основного напрямку позиціонування, що представлено у таблиці 5.22.

Таблиця 5.22 — Концепція маркетингових комунікацій

№ з/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
-------	---------------------------------------	--	--	----------------------------------	--------------------------------

Кінець таблиці 5.22

1	Прямий маркетинг	особистий продаж в онлайн-ресурсах	Військові виставки, семінари	Дослідження ринку, Виявлення потенційних клієнтів	Пропозиція
---	------------------	------------------------------------	------------------------------	---	------------

У результаті роботи був реалізований комплексний маркетинговий план, який включає стратегії для просування товару, реалізації та попередній аналіз можливостей у ціноутворенні. Цей план ґрунтується на врахуванні цінностей та потреб потенційних клієнтів, конкурентних переваг ідеї проекту, а також на оцінці стану і тенденцій ринкового середовища, де реалізується проект. У межах цього плану визначена відповідна альтернатива ринкової стратегії.

Висновки до розділу 5

В останньому розділі було проведено аналіз програмного продукту у якості стартап-проекту. Аналіз детально розписано у 22 таблицях із описом кожного їх рядка та стовпчика. Результатом є підготовлений бізнес план щодо реалізації та впровадження на ринку цього стартап проекту. Використання подібної системи аналізу сигналів з РЛС є дуже актуальним питанням для сучасної України, тому очікується попит на подібну систему, після доведення її до завершеного стану розробки.

ВИСНОВКИ

В ході роботи над магістерською дисертацією було виконано всі поставлені задачі й розроблено функціонуючий застосунок аналізу сигналів із радіолокаторів. Під час його реалізації було виправдано необхідність існування подібної системи та впровадження її у використання у Збройних силах.

В першому розділі роботи було описано актуальність задачі, розписана стисла історія радіолокації та радіолокаторів, продемонстровано види радіолокаторів та їх відмінності їх роботи, також проведено перелік сучасних радіолокаторів у використанні Збройних сил України. Сформульоване загальні теоретичні поняття радіолокації, що необхідні у реалізації подібної системи.

В другому розділі було проведено пошук існуючих аналогічних програм та проаналізовано методи, за якими вони функціонують. Аналогічним програмним забезпеченням вважаємо систему Дельта, української розробки, що наразі впроваджується як інструмент Збройних сил й має переваги відносно інших систем, в тому числі іноземних. Також в цьому розділі представлено всі необхідні математичні формули та розрахунки, що необхідні для створення системи, а також представлено табличні характеристики повітряних об'єктів, які були використані при розробці програмного забезпечення. На останок було описано усі програмні засоби розробки продукту, а саме мову програмування, систему керування базами даних, а також середовища для роботи із цими мовами.

В третьому розділі представлена архітектура розробленого програмного забезпечення, котра складається із декількох модулів, а саме: модуль симулятору РЛС, модуль порівняння даних та модуль шифрування й дешифрування даних. Виконано опис кожного з цих модулів системи та наведено принципи їх роботи. Продемонстровано діаграму послідовності та діаграму прецедентів до розробленої системи.

В четвертому розділі надані детальні інструкції щодо користування розробленим програмним забезпеченням. Здійснено опис і призначення всіх елементів інтерфейсу, а також розписана деталізована інструкція по встановленню

системи та всього необхідного додаткового програмного забезпечення. На останок приведена коротка інформація про способи для покращення функціоналу розробленої системи.

В п'ятому розділі був проведений аналіз програмного продукту як стартап-проекту та продемонстровано усі аспекти в табличному вигляді. Результатом цього розділу є сформульований бізнес план щодо реалізації та впровадження на ринку цього стартап-проекту.

Створений програмний продукт дозволяє кінцевому користувачеві здійснювати підключення до симулятору радіолокатора, налаштовувати його, запускати й отримувати результати обробки сигналів цього радіолокатора, які автоматично будуть збережені у базі даних, задля можливості переглядати дані у теперішньому та майбутньому в самому програмному забезпеченні, а також в інтерфейсі бази даних. Користувачу надана можливість вносити додаткові дані про повітряні об'єкти у систему, задля розширення функціоналу роботи модуля аналізу й уточнення результатів роботи програми.

Розроблена система здатна виконувати головне поставлене завдання – аналіз типів повітряних об'єктів від даних з індикаторів кругового огляду.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Радіолокаційна станція — ВУЕ [Електронний ресурс] — URL: <http://surl.li/nwlck>
2. Radar Countermeasure Chaff [Електронний ресурс] — URL: <https://www.armtecdefense.com/Countermeasures/Radar-Chaff>
3. Огняник Д.М., Онисько А.І. Засоби інформаційної підтримки ідентифікації повітряних об'єктів на індикаторах кругового огляду: зб. матеріалів доп. учасн. зб. матеріалів доп. учасн. II Міжнародна науково-практична інтернет-конференція «Education via Distance Learning and other Pedagogical Challenges». Дніпро, 2023. С. 165-167.
4. Радіолокація — ВУЕ [Електронний ресурс] — URL: <http://surl.li/nwlco>
5. Радіонавігація — ВУЕ [Електронний ресурс] — URL: <http://surl.li/nwlcpr>
6. Радіотехніка / За ред. Ю. Л. Мазора, Є. А. Мачуського, В. І. Правди. Київ : Вища школа, 1999. 838 с.
7. Пасивна радіолокація [Електронний ресурс] — URL: <http://surl.li/nwlcw>
8. Активна радіолокація [Електронний ресурс] — URL: <http://surl.li/nwlcz>
9. Основи радіолокації — Історичний огляд [Електронний ресурс] — URL: <https://www.radartutorial.eu/04.history/hi04.uk.html>
10. Основи радіолокації — Класифікація радіолокаційних систем залежно від застосовуваних технологій [Електронний ресурс] — URL: <https://www.radartutorial.eu/02.basics/rp04.uk.html>
11. Вітчизняна контрбатарейна РЛС: початок виробництва дослідного зразка [Електронний ресурс] — URL: <https://web.archive.org/web/20210115144300/https://mil.in.ua/uk/vitchyznyana-kontrbatareyna-rls-pochatok-vyrobnyctva-doslidnoho-zrazka/>
12. Засіб наземної радіолокації РЛС 112L1 «Барсук» [Електронний ресурс] — URL: <https://web.archive.org/web/20160407130148/http://na.mil.gov.ua/31835-zasib-nazemno%D1%97-radiolokaci%D1%97-rls-112l1-barsuk>

13. Цикл виробництва нової станції «Кольчуги» складає від 9 до 12 місяців [Електронний ресурс] — URL: <https://mil.in.ua/uk/news/tsykl-vyrobnytstva-novoyi-stantsiyi-kolchugy-skladaye-vid-9-do-12-misyatsiv/>

14. «КОЛЬЧУГА-М» ТА «ПОЛОЖЕНІС-2» У ПРИОРИТЕТАХ ДК «УКРОБОРОНПРОМ» [Електронний ресурс] — URL: <https://old.defence-ua.com/index.php/home-page/285-kolchuha-m-ta-polozheniye-2-u-priorytetakh-dk-ukroboronprom>

15. Новий розвідник ВМСУ отримає станцію радіотехнічної розвідки «Мельхіор» [Електронний ресурс] — URL: https://www.naval-pages.com/2019/10/melhior.html?fbclid=IwAR1NbKS011OIiQQU4jXlgHGLiExgj_TE270jNR8B6iZSkJz-Sle3Z7tZqFM

16. РЛС Х1-М «ОКО»: ВОРОЖОМУ «БЕСПІЛОТНИКУ» ЖОДНОГО ШАНСУ [Електронний ресурс] — URL: <https://web.archive.org/web/20180125074708/https://defence-ua.com/index.php/statti/3974-rls-kh1-m-oko-vorozhomu-bespilotnyku-zhodnoho-shansu>

17. РЛС контрбатареїної боротьби AN/TPQ-36 [Електронний ресурс] — URL: <https://www.ukrmilitary.com/2016/05/an-tpq-36-tth.html>

18. AN/MPQ-64 „Sentinel” [Електронний ресурс] — URL: <https://www.radartutorial.eu/19.kartei/04.battle/karte011.en.html>

19. Що таке DELTA? [Електронний ресурс] — URL: <https://delta.mil.gov.ua/wiki/info/>

20. Delta для ЗСУ: Що відомо про новітню систему управління української армії [Електронний ресурс] — URL: <https://www.bbc.com/ukrainian/features-64585182>

21. What is Arduino? [Електронний ресурс] — URL: <https://www.arduino.cc/en/Guide/Introduction>

22. Розробка радіолокатора з використанням HC-SR04 та Arduino [Електронний ресурс] — URL: <https://diy-arduino.com/proekty-na-arduino/rozrobka-radiolokatora-z-vykorystanniam-hc-sr04-ta-arduino/>

23. Ефективна площа вторинного випромінення [Електронний ресурс] — URL: <https://www.radartutorial.eu/01.basics/rb56.uk.html>
24. Поперечное сечение радара - Radar cross-section [Електронний ресурс] — URL: https://ru.wikibrief.org/wiki/Radar_cross-section
25. Радиальная скорость [Електронний ресурс] — URL: <https://www.radartutorial.eu/11.coherent/co52.ru.html>
26. Способ однозначного измерения радиальной скорости цели в когерентно-импульсной радиолокационной станции [Електронний ресурс] — URL: <https://patents.google.com/patent/RU2574079C1/ru>
27. Вимірювання дальності за допомогою радіолокатора [Електронний ресурс] — URL: <https://www.radartutorial.eu/01.basics/rb07.uk.html>
28. Вимірювання азимуту цілі [Електронний ресурс] — URL: <https://www.radartutorial.eu/01.basics/rb08.uk.html>
29. What is MongoDB Atlas? [Електронний ресурс] — URL: <https://www.mongodb.com/docs/atlas/>
30. What is C++? | Definition from TechTarget [Електронний ресурс] — URL: <https://www.techtarget.com/searchdatamanagement/definition/C>
31. Що таке JSON | Навіщо потрібен цей формат [Електронний ресурс] — URL: <https://apix-drive.com/ua/blog/useful/scho-take-json>
32. Робота з даними в MongoDB Compass [Електронний ресурс] — URL: <https://krypton.com.ua/rozdil-2-robota-z-bazoyu-danyh/robota-z-danymy-v-mongodb-compass/>
33. PyCharm Features [Електронний ресурс] — URL: <https://www.jetbrains.com/pycharm/features/>
34. tkinter — Інтерфейс Python до Tcl/Tk [Електронний ресурс] — URL: <https://docs.python.org/uk/3/library/tkinter.html>
35. JSON encoder and decoder — Python 3.12.0 documentation [Електронний ресурс] — URL: <https://docs.python.org/3/library/json.html>
36. NumPy documentation [Електронний ресурс] — URL: <https://numpy.org/doc/stable/>

37. How does the tomllib library work [Електронний ресурс] — URL: <https://www.educative.io/answers/how-does-the-tomllib-library-work>
38. random — Генерація псевдовипадкових чисел [Електронний ресурс] — URL: <https://docs.python.org/uk/3/library/random.html>
39. hashlib — Захищені хеші та дайджести повідомлень [Електронний ресурс] — URL: <https://docs.python.org/uk/3/library/hashlib.html>
40. PyMongo 4.6.1 documentation [Електронний ресурс] — URL: <https://pymongo.readthedocs.io/>
41. Advanced Encryption Standard (AES) [Електронний ресурс] — URL: <https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard>
42. AES Crypt - Advanced File Encryption [Електронний ресурс] — URL: <https://www.aescrypt.com/>
43. Діаграма послідовності (Sequence Diagrams) [Електронний ресурс] — URL: <https://www.maxzosim.com/sequence-diagrams/>
44. Діаграми Прецедентів (Use Case UML Diagram) [Електронний ресурс] — URL: <http://lvivqaclub.blogspot.com/2008/10/use-case-uml-diagram.html>
45. BAT - формат пакетного файлу - File Format Docs [Електронний ресурс] — URL: <https://docs.fileformat.com/uk/executable/bat/>

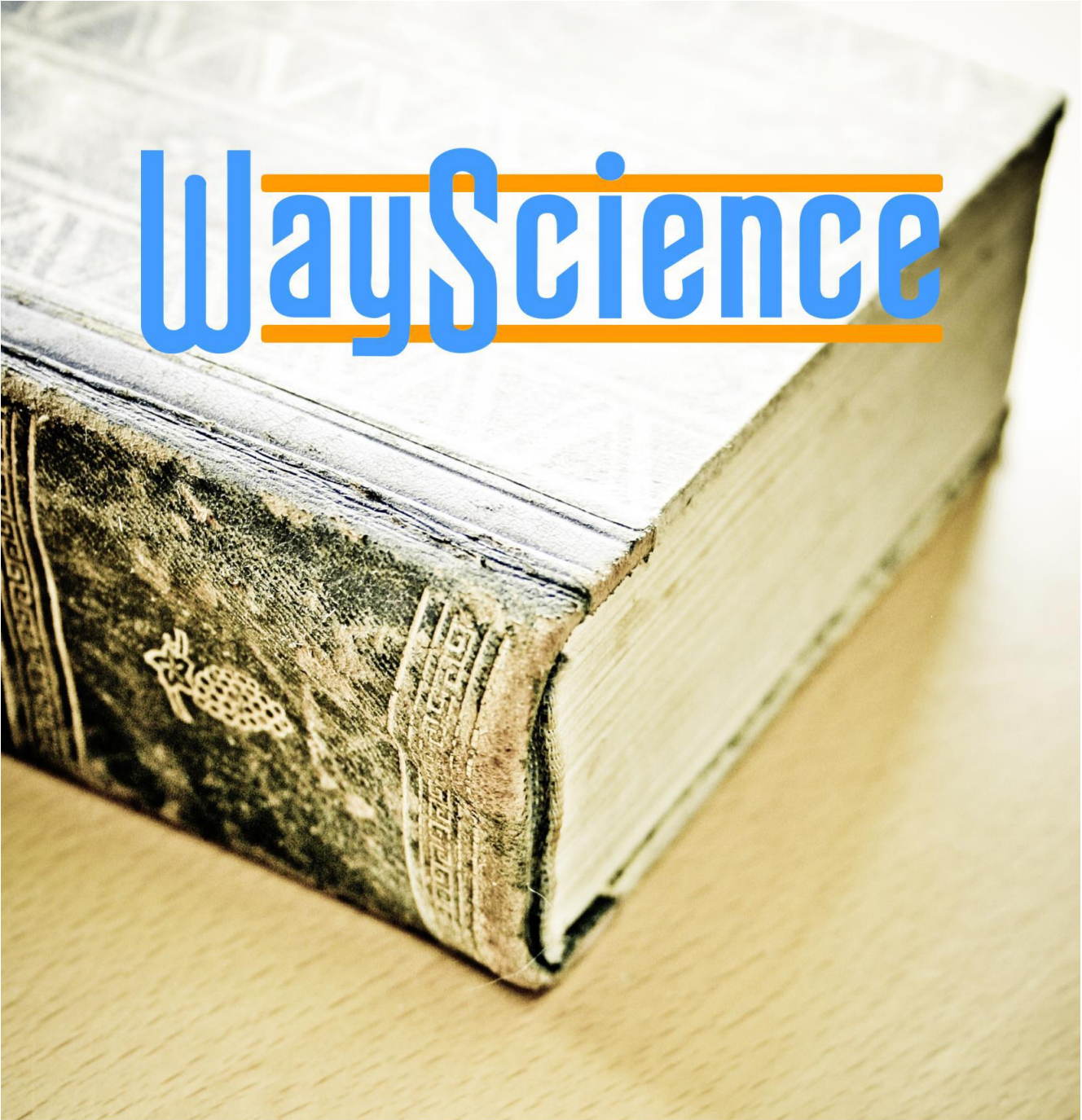
ДОДАТОК А

ЗАСОБИ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ ІДЕНТИФІКАЦІЇ ПОВІТРЯНИХ ОБ'ЄКТІВ НА ІНДИКАТОРАХ КРУГОВОГО ОГЛЯДУ

Апробація

II Міжнародна науково-практична інтернет-конференція «Education via
Distance Learning and other Pedagogical Challenges»

Аркушів - 6

The background of the cover is a photograph of an old, thick book with a worn, dark green leather binding. The book is lying flat on a light-colored wooden surface. The pages are aged and yellowed. The 'WayScience' logo is superimposed over the upper half of the book. The logo consists of the word 'Way' in a blue, sans-serif font, followed by 'Science' in a similar blue font. A thick orange horizontal bar runs behind the text, with the 'y' in 'Way' and the 'S' in 'Science' partially overlapping it.

WayScience

2nd International Scientific
and Practical Internet Conference

«Education via Distance Learning and
other Pedagogical Challenges»

ISBN 978-617-8293-08-6

Лук'янчук Ж.В., Атаманчук П.С. ІНТЕРАКТИВНІ МЕТОДИ НАВЧАННЯ ТА ЇХ РОЛЬ У ПІДГОТОВЦІ МАЙБУТНЬОГО ВЧИТЕЛЯ ФІЗИКИ	138
Мамедова А.І., Рось С.В. ДИСТАНЦІЙНЕ НАВЧАННЯ: ТЕСТОВІ ЗАВДАННЯ З ГРАМАТИКИ ТА ЇХ ФОРМИ ПРИ ВИВЧЕННІ ІНОЗЕМНИХ МОВ	141
Маріна Ю.М. ПРОБЛЕМА ЗАБЕЗПЕЧЕННЯ ПРАВ І СВОБОД ГРОМАДЯН В ПЕРІОД ВОЄННОГО СТАНУ	143
Марков А.М., Опушко Н.Р. ШТУЧНИЙ ІНТЕЛЕКТ ТА ПРОБЛЕМА АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ	146
Мартинюк І., Мартинюк Г. ВИКОРИСТАННЯ СУЧАСНИХ ІННОВАЦІЙНИХ ПЕДАГОГІЧНИХ ТЕХНОЛОГІЙ НА УРОКАХ ХІМІЇ	150
Микуляк Д.О. ЕКОНОМІКА МАЙБУТНЬОГО: ВПЛИВ ШТУЧНОГО ІНТЕЛЕКТУ НА РОБОЧУ СИЛУ ТА УПРАВЛІННЯ ДАНИМИ	152
Могильницька Л.А., Онищук Т.П. ВИКОРИСТАННЯ КЕЙС-МЕТОДУ ЯК ФОРМИ ДИСТАНЦІЙНОГО НАВЧАННЯ НА КАФЕДРІ ТЕРАПЕВТИЧНИХ ДИСЦИПЛІН ФАКУЛЬТЕТУ ПІСЛЯДИПЛОМНОЇ ОСВІТИ	154
Мороз О.Л. ШЛЯХИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ФОРМУВАННЯ КОМУНІКАТИВНИХ НАВИЧОК МАЙБУТНІХ СУДНОВОДІВ В УМОВАХ ДИСТАНЦІЙНОГО НАВЧАННЯ	156
Намачинська Г.Я. ДИСТАНЦІЙНЕ НАВЧАННЯ ТА ЙОГО ВИДИ В УМОВАХ ВОЄННОГО СТАНУ	159
Ніколаєнко С.О., Тарасова Т.Б., Теслик О.М. СУТЕСТИВНІ МЕТОДИ РОЗВИТКУ КУЛЬТУРИ ВНУТРІШНЬООРГАНІЗАЦІЙНОГО ПРОФЕСІЙНОГО СПІЛКУВАННЯ ПЕРСОНАЛУ ПРОЄКТНИХ ОРГАНІЗАЦІЙ	161
Новосельчук Н.Є. ВИКОРИСТАННЯ МОЖЛИВОСТЕЙ ШТУЧНОГО ІНТЕЛЕКТУ В АРХІТЕКТУРНІЙ ОСВІТІ	164
Огняник Д.М., Онисько А.І. ЗАСОБИ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ ІДЕНТИФІКАЦІЇ ПОВІТРЯНИХ ОБ'ЄКТІВ НА ІНДИКАТОРАХ КРУГОВОГО ОГЛЯДУ	165
Павленко М.І. ВИРІШЕННЯ ПРОБЛЕМ ПІДВИЩЕННЯ ЯКОСТІ ПРАКТИЧНОЇ ПІДГОТОВКИ МАЙБУТНІХ СПЕЦІАЛІСТІВ В ГАЛУЗІ АВТОМАТИЗАЦІЇ ХАРЧОВИХ ТА ПЕРЕРОБНИХ ВИРОБНИЦТВ	168
Попенко Т.В., Литвиненко І.С. ПСИХОСОЦІАЛЬНІ ПРОБЛЕМИ ДІТЕЙ МОЛОДШОГО ШКІЛЬНОГО ВІКУ – ВИМУШЕНИХ ПЕРЕСЕЛЕНЦІВ	170
Радкевич О.П. ПЕРЕВАГИ ЕЛЕКТРОННИХ ЗАСОБІВ НАВЧАННЯ ТА ОЦІНЮВАННЯ	173
Саєнко Н.В. ФОРМУВАННЯ ІНШОМОВНОЇ ФОНЕТИЧНОЇ КОМПЕТЕНЦІЇ З ВИКОРИСТАННЯМ МОБІЛЬНИХ ЗАСОБІВ	177
Смаглюк А.О. ІННОВАЦІЙНІ ТЕХНОЛОГІЇ В ПЕДАГОГІЧНІЙ ДІЯЛЬНОСТІ	179
Сокульська Н.Б., Гузик Н.М., Ліщинська Х.І. ДИСТАНЦІЙНЕ НАВЧАННЯ У ВИЩИХ ВІЙСЬКОВИХ НАВЧАЛЬНИХ ЗАКЛАДАХ: ПЕРЕВАГИ, НЕДОЛІКИ, ВИКЛИКИ	181
Сокульська Н.Б., Ковальчук Р.А., Кмін В.Ф. ШЛЯХИ РЕАЛІЗАЦІЇ ЦІЛЕЙ НАВЧАННЯ ДОРΟΣЛИХ У ДИСТАНЦІЙНОМУ НАВЧАЛЬНОМУ ПРОЦЕСІ	183
Станкова М.Д. ОРГАНІЗАЦІЯ ДИСТАНЦІЙНОЇ ФОРМИ ОСВІТИ В УМОВАХ ВОЄННОГО СТАНУ	185
Степанець Г.М. УКРАЇНСЬКА НАРОДНА КАЗКА ЯК ЗАСІБ ФОРМУВАННЯ ПАТРІОТИЧНИХ ПОЧУТТІВ ДІТЕЙ ДОШКІЛЬНОГО ВІКУ	186
Строїлова Д.В. ВІДДАЛЕНА ОСВІТА: ВИКЛИКИ І ПЕРЕВАГИ В УМОВАХ ВІЙНИ	189

ЗАСОБИ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ ІДЕНТИФІКАЦІЇ ПОВІТРЯНИХ ОБ'ЄКТІВ НА ІНДИКАТОРАХ КРУГОВОГО ОГЛЯДУ

Огняник Дмитро Миколайович
студент

Онисько Андрій Ілліч
кандидат військових наук, доцент
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
м. Київ, Україна

Анотація. Стаття присвячена обґрунтуванню необхідності створення засобів комп'ютерної підтримки роботи індикаторів кругового огляду, таких як радіолокаційні станції (РЛС). Автор пропонує варіант реалізації програмного забезпечення для автоматизованого аналізу одержаної інформації індикаторами кругового огляду, задля спрощення людської праці на подібних приладах. В статті розглядаються основні та додаткові можливості створюваної автором програми.

Ключові слова: радіолокаційні станції (РЛС), індикатор кругового огляду, система бойового управління Delta, ефективна площа розсіювання (ЕПР)

Постановка проблеми. Зараз осінь 2023 року, півтора року триває повномасштабна фаза україно-російської війни в якій застосовуються всі доступні передові технології людства. Використовуються балістичні та крилаті ракети, літаки, гелікоптери, фальш цілі тощо. Для фіксування подібних об'єктів у повітрі в минулому столітті були створені радіолокаційні станції. РЛС [3] бувають різних габаритів та різняться за призначенням. В умовах війни вони працюють цілодобово, обстежуючи повітряний простір небо України, щоб вчасно передавати дані про ворожі та союзні цілі у повітрі для військових, а також щоб своєчасно повідомляти населенню про повітряну тривогу в тому чи іншому регіоні країни. Питання розробки програмного забезпечення, котре буде автоматично відслідковувати сигнали, які отримує РЛС, задля зменшення ризиків пов'язаних із людським фактором (таким як неувважність та стомленість), наразі є актуальним.

Аналіз останніх досліджень і публікацій. Влітку цього року була протестована українська система бойового управління та обробки інформації Delta [5]. Один із головних протоколів, який вона вдало протестувала цього року, - це Link 16. Цей протокол надає можливість Дельті отримувати дані від винищувачів F-16, що є важливим для України.

Ця система інтегрує інформацію про розташування сил та засобів противника і надає можливість в реальному часі відстежувати положення військ супротивника та швидко враховувати виявлені об'єкти для їх подальшого поразки. Інформацію про противника система інтегрує на цифровій карті, використовуючи дані з різних джерел, таких як супутникові знімки, радары, сенсори, GPS-трекери та системи радіоперехоплення.

Виклад основного матеріалу. Для того, щоб розробити подібну систему спочатку необхідно знайти та проаналізувати дані про роботу РЛС. Для цього була використана мережа Інтернет та пошук проводився по відкритим для всіх ресурсах. Було проаналізовано типи РЛС, особливості специфіки їх роботи, підготовано алгоритм дій у розроблюваному програмному забезпеченні та розпочато створення самої програми. Радіолокаційні станції при обстеженні повітряного простору отримують різноманітні дані. Такими даними (сигналами) можуть виступати літаки, гелікоптери, хибні цілі, ракети, а також хмари та інші об'єкти. Також окрім корисних сигналів радар одержує так званий шум, який слід приглушити. Тому слід розрізняти в отриманих сигналах який саме об'єкт було помічено в

повітрі. Для цього проводиться аналіз таких параметрів, за одержаним сигналом: тип цілі, швидкість цілі, висота цілі, дальність до РЛС, азимут, приблизні габарити цілі.

Система отримує дані із симулятора радіолокаційної станції із регульованими параметрами, такими як тривалість імпульсу, ширина смуги пропускання імпульсу, робоча частота РЛС, відстань до цілі, розміри, швидкість та напрям цілі. Систему можна переналаштовувати під радар з різними робочими характеристиками.

Важливим показником є ефективна площа розсіювання ЕПР (Рисунок 1) [5]. На основі отриманої оцінки сигналу, може бути визначена ЕПР цілі, що дозволяє розпізнавати її клас.

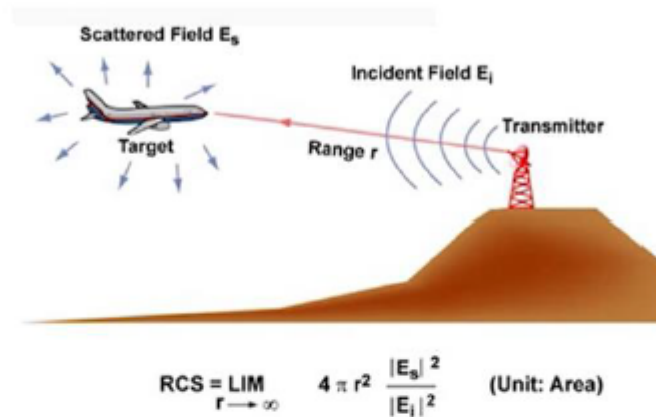


Рисунок 1. Демонстрація роботи РЛС

Він використовується для виявлення повітряних цілей у широкому діапазоні відстаней. Наприклад, літак-невидимка (який розроблений з урахуванням низької помітності) буде мати конструктивні особливості, що забезпечують низьку ЕПР (наприклад, поглинаюча фарба, плоскі поверхні, поверхні, спеціально розташовані під кутом відбивати сигнал в іншому місці, крім джерела), на відміну від пасажирського авіалайнера, у якого буде високою ЕПР (голий метал, закруглені поверхні, що ефективно гарантовано відображають деякий сигнал назад до джерела, багато виступів, такі як двигуни, антени та ін.). ЕПР є невід'ємною частиною розробки радарів стелс-технологій, особливо у додатках, пов'язаних із літаками та балістичними ракетами. Дані ЕПР для найсучасніших військових літаків є найбільш засекреченими.

Отже, програма розроблена об'єктно-орієнтованою мовою програмування Python [2], також до неї підключено й налаштовано базу даних MongoDB [1], в якій зберігаються відомі дані про типи об'єктів (літак, гелікоптер, хибна ціль тощо). При одержанні даних радаром наш алгоритм порівнює ці дані із тими, що зберігаються в базі даних (такі як швидкості цілей, їх висота та ЕПР), та враховуючи найбільшу кількість збігів даних видає інформацію який саме об'єкт знаходиться у повітрі. Також система зберігає вже оброблені параметри виявленого об'єкту в базу даних (тип об'єкту, швидкість, дальність до РЛС, азимут та час їх появи на радарі).

Висновок. Розроблювана система є незамінним інструментом в умовах, коли в повітрі перебувають гелікоптери, ракети, військові літаки, фальш цілі та інше. На фоні подій, що розгортаються, важливою функцією цієї системи стає забезпечення оперативного та точного сповіщення військових, а отже і населення про можливі обстріли.

Система дозволяє виявляти та ідентифікувати повітряні об'єкти та надавати цю інформацію операторам, що відповідають за безпеку. Ця система допоможе операторам відслідковувати небо України на кшталт появи небезпечних об'єктів, що потребує оголошення сигналів повітряних тривог для населення в конкретних регіонах країни.

Інтеграція з існуючими системами безпеки та інформаційного сповіщення гарантує надійний захист населення в умовах загрози з повітря, допомагаючи уникнути небезпек та мінімізувати можливі ризики.

Список літератури:

1. What is MongoDB Atlas? [Електронний ресурс] — URL: <https://www.mongodb.com/docs/atlas/>
2. Python 3.11.5 documentation [Електронний ресурс] — URL: <https://docs.python.org/3/index.html>
3. Радіолокаційна станція — ВУЕ [Електронний ресурс] — URL: https://vue.gov.ua/%D0%A0%D0%B0%D0%B4%D1%96%D0%BE%D0%BB%D0%BE%D0%BA%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D1%81%D1%82%D0%B0%D0%BD%D1%86%D1%96%D1%8F
4. Українська система бойового управління та обробки інформації Delta [Електронний ресурс] — URL: <https://rubryka.com/2023/07/11/ukrayinska-systema-bojovogo-upravlinnya-ta-obrobky-informatsiyi-delta-uspishno-projshla-testuvannya-nato-fedorov/>
5. Target Radar Cross Section - MIT Lincoln Laboratory [Електронний ресурс] — URL: <https://www.ll.mit.edu/media/6956#:~:text=Definition%20of%20Radar%20Cross%20Section,received%20power%20in%20the%20radar.>