

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
Кафедра Програмної інженерії

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

другий (магістерський)
(рівень вищої освіти)

Дослідження моделей і алгоритмів побудови
системи TACAN

Виконав:

студент 2 курсу групи ІПЗм-21-3

Деренський М.М.
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного
забезпечення

Тип програми Освітньо-наукова

Керівник проф. Смеляков С.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав кафедри _____

З.В. Дудар

2023 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
 Кафедра Програмної інженерії
 Рівень вищої освіти другий (магістерський)
 Спеціальність 121 – Інженерія програмного забезпечення
 (код і повна назва)
 Тип програми освітньо-наукова програма
 Освітня програма Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:

Зав. Кафедри _____

(підпис)

«__» _____ 202_р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студента Деренському Максиму Миколайовичу
 (прізвище, ім'я, по батькові)

1. Тема роботи «Дослідження моделей і алгоритмів побудови системи TACAN»

затверджена наказом університету від « 03 » 04 2023 р. № 83Стз

2. Термін подання студентом роботи до екзаменаційної комісії «23» 05 2023 р.

3. Вихідні дані до роботи радіонавігація, радіолокаційна станція, радіонавігаційна система TACAN, веб-програмування, пояснювальна записка.

4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз сучасного стану питання, розробка методів та алгоритмів, розробка програмного забезпечення симулятора, вивчення можливостей її використання.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання на виконання кваліфікаційної роботи	01.03.23	
2	Аналіз завдання та теоретичних матеріалів	01.03 – 15.03.23	
3	Розробка методів та алгоритмів	16.03 – 22.03.23	
4	Розробка вимог до програмного забезпечення	23.03 – 24.03.23	
5	Структурне проектування	25.03.23	
6	Розробка та тестування програми	26.03 – 12.04.23	
7	Подання статті на наукову конференцію	12.04.23	
8	Наукова апробація роботи	21.04.23	
9	Оформлення пояснювальної записки	12.04 – 12.05.23	
10	Подання роботи на перевірку інтернет-сервісом «Unіcheck» та нормоконтроль	12.05.23	
11	Представлення роботи на рецензування	17.05.23	
12	Подання кваліфікаційної роботи до ЕК	22.05.23	

Дата видачі завдання 03 квітня 2023_р.

Студент Деренський М. М.
(підпис)

Керівник роботи _____
(підпис) проф. Смеляков С. В.
(посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 75 с., 24 рис., 2 табл., 40 джер.

БАЗА ДАНИХ, БІЗНЕС ЛОГІКА, МОДЕЛЮВАННЯ, ПРОГРАМУВАННЯ, РАДІОЛОКАЦІЙНА СИСТЕМА, РАДІОНАВІГАЦІЯ, СИМУЛЯТОР, ТАБЛИЦЯ, ТАКАН, POSTGRESSDB, TYPESCRIPT.

Об'єктом дослідження є методи та моделі роботи системи ТАКАН.

Метою роботи є впровадження системи TACAN в нашій країні.

Методи розробки базуються на таких технологіях, як JavaScript/Typescript, клієнт-сервер, мікросервіси, веб-сокети.

В результаті роботи було досліджено моделі та алгоритми роботи системи ТАКАН та розроблено програмне забезпечення, що є симулятором роботи радіонавігаційної системи ТАКАН, яке симулює роботу кожного окремого компоненту та всі процеси в цілому для цілей проведення комп'ютерних експериментів та навчання персоналу, що працює з даною системою. Програмне забезпечення базується на мікросервісній архітектурі з відокремленим користувацьким інтерфейсом і відображенням даних у режимі реального часу.

DATABASE, BUSINESS LOGIC, RADIOLOCATION SYSTEM, TACAN, RADIONAVIGATION, PROGRAMMING, MODELING, SIMULATOR, TABLE, DATABASE, POSTGRESSDB, TYPESCRIPT.

The object of research is the methods and models of the TACAN system.

The purpose of the work is to implement the TACAN system in our country.

Development methods are based on such technologies as JavaScript/Typescript, client-server, microservices, websockets.

As a result of the work, models and algorithms of the TACAN system were studied and software was developed, which is simulator of the TACAN radio navigation system, which simulates the operation of each individual component and all processes

as a whole for the purpose of conducting computer experiments and training personnel working with the system. The software is based on microservices architecture with a separated user interface and real-time data display.

Умови публікації пояснювальної записки

Я, _____ Деренський Максим Миколайович _____
(прізвище, ім'я, по батькові)

студент(ка) групи _____ ПЗМ-21-3 _____ здобувач вищої освіти на другому
(магістерському) рівні

кафедра _____ програмної інженерії _____,
(повна назва кафедри)

Заявляю: моя кваліфікаційна робота на тему

_____ Дослідження методів і алгоритмів побудови системи TACAN _____,
(назва роботи)

Що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу EIArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений (а) з діючими положеннями «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ.....	7
1 Огляд сучасного стану питання.....	10
1.1 Опис предметної області.....	10
1.2 Огляд наукової літератури.....	12
2 Розробка моделей, алгоритмів та методів.....	15
2.1 Принцип роботи радіолокаційної системи.....	15
2.2 Алгоритми визначення параметрів об'єктів в РСБН.....	17
2.3 Принципи та алгоритми роботи РСБН ТАКАН.....	18
2.4 Вимоги до програмного продукту.....	20
3 Розробка прототипу інформаційної технології.....	24
3.1 Вибір архітектури програмного забезпечення.....	24
3.2 Опис архітектури програмної реалізації РСБН ТАКАН.....	30
3.3 Архітектура Backend сервісу «Станція».....	31
3.4 Архітектура Backend сервісу «Літак».....	34
3.5 Архітектура Backend сервісу «Веб-сокет».....	36
3.6 Архітектура Frontend додатку.....	38
3.7 Взаємодія клієнтської частини додатку з сервером.....	42
3.8 Опис можливостей клієнтської частини.....	45
4 Експериментальний розділ.....	49
4.1 Підготовка експерименту.....	49
4.2 Проведення експериментів.....	51
5 Обробка результатів.....	54
Висновки.....	57
Перелік джерел посилання.....	58
Додаток А.....	64
Додаток Б.....	65
Додаток В.....	74
Додаток Г.....	75

ВСТУП

В наш час авіація займає значущу частину у життєдіяльності людства. Авіація допомагає у вирішенні значущих проблем сучасного суспільства такі як: доставка людини з місце на місце на ближні відстані або на тисячі кілометрів у цивільній авіації, доставка цивільних або військових чи спеціалізованих вантажів у віддалені куточки світу.

Сучасні літаки є автоматизованими та комп'ютеризованими, надають великий обсяг інформації пілотам під час польоту. Пілоти у режимі реального часу слідкують за показниками бортової системи літака, що надає інформацію про висоту польоту, швидкість руху, курс, рівень пального тощо. Одне із головних місць в бортовій системі будь якого літака цивільного чи військового займає аеронавігаційна система.

Вся аеронавігація базується на радіонавігації. Радіонавігація – сукупність операцій по забезпеченню водіння рухомих об'єктів (літальних апаратів, суден і ін.), а також по наведенню керованих об'єктів за допомогою радіотехнічних засобів [1]. Вона забезпечує польоти, слідкує за повітряним простором та надає пілотам інформацію про координати літака у поточний час, а авіадиспетчерам або операторам радіолокаційних станцій інформацію про літаки у повітряному просторі, їх напрям, ідентифікацію «свій-чужий».

Користуючись радіонавігаційними засобами пілоти слідкують за маршрутом, злітають з аеродромів або заходять на посадку за зазначеними маршрутами. У разі відсутності зв'язку із супутником більш прості наземні засоби продовжують надання інформації про поточні координати.

Також активно розвивається сектор ринку інформаційних технологій, присвячений розробці та використанню симуляторів широкого спектру призначення та використання. Найбільш відомі симулятори управління транспортними засобами та медичні симулятори. Основні цілі розробки таких симуляторів полягають у тому, щоб надати можливість максимально реально заглибитися у процес, що досліджуються або спостерігається, отримати цілісне

уявлення про нього, навчитися ефективно контролювати об'єкт чи процес, наприклад управління літаком.

Наразі актуальною є розробка сучасних комп'ютерних симуляторів радіотехнічних систем ближньої навігації (РСБН), в тому числі РСБН TACAN, головним чином, в цілях оперативного навчання роботи з РСБН.

При цьому симулятор РСБН повинен цілісно інтегрувати процес взаємодії літака та маяка на реальній місцевості (карті), в реальному масштабі часу. Симулятор повинен надавати якісний інтерфейс візуалізації і необхідний функціонал для моделювання широкого спектру ситуацій, для порівняно швидкого розуміння суті процесу, освоєння основних механізмів взаємодії літака та маяка.

Об'єктом дослідження є радіотехнічна система ближньої навігації в цілому та різновиду її реалізації, системи TACAN.

Предметом дослідження є моделі та алгоритми побудови системи TACAN.

Для досягнення таких цілей у вирішуються такі основні завдання:

- аналіз сучасного стану питання в предметній області;
- дослідження алгоритмів роботи РСБН TACAN;
- вимоги до сучасного симулятора РСБН, основні методи, алгоритми та технології реалізації, функціональні можливості, вимоги до інтерфейсу та візуалізації, до використання сценаріїв і багато іншого;
- опис запропонованого симулятора, експериментів з ним, результатів експериментів, їх інтерпретація, рекомендації та оцінки ефективності.

Під час роботи були застосовані такі методи дослідження:

- аналіз – було вивчено основну теорію за темою та проаналізовано кожний окремий елемент;
- синтез – було розроблено програмну реалізацію кожного окремого елемента та пов'язано за допомогою програмних засобів зв'язування між собою у єдину працюючу систему;
- експеримент – було проведено експерименти з готовим програмним забезпеченням, що показало його відповідність до завдання роботи.

Наукова новизна роботи полягає в адаптації вже існуючого рішення для використання в нашій країні, адже такий вид радіонавігації раніше тут не застосовувався. Тому написання симулятора РСБН TACAN є інноваційним завданням.

Практичним застосуванням результатів дослідження є розроблена у процесі роботи програмна система, що є повноцінним симулятором роботи аеронавігації в системі TACAN, що імітує взаємодію бортової системи літака та радіолокаційних станцій на землі з візуалізацією процесу та параметрів на користувацькому інтерфейсу та можливістю створювати різний спектр сценаріїв для проведення комп'ютерних експериментів, що позбавляє необхідності від проведення натурних експериментів. Може застосовуватись у навчальних цілях або у цілях попереднього розташування об'єктів на віртуальній місцевості та дослідження взаємодії з ними.

Було проведено апробацію наукових досліджень шляхом подачі наукової статті на конференцію COLINS та участю у доповіді. Стаття отримала гарні відгуки і за думкою комісії вважається сучасною.

1 ОГЛЯД СУЧАСНОГО СТАНУ ПИТАННЯ

1.1 Опис предметної області дослідження

Системи навігації відіграють велику роль у авіаційній сфері діяльності людини. Завдяки навігації пілоти знають та слідкують за маршрутом польотів, а також отримують необхідні підказки та показники такі як висота, поточні координати, напрямок тощо.

Є декілька видів систем, що забезпечують літаки навігацією, це системи, що базуються на GPS і вимагають зв'язку з відповідними супутниками та системи наземного та (або) морського базування. Під час польоту бортова система літака може бути налаштована на один із цих видів навігації і, за потреби, пілоти можуть переключатися з одного виду навігації на інший. Наприклад, під час польоту на великій висоті для отримання точних поточних координат використовуються супутники GPS, а при посадці використовуються наземні станції, що розташовуються на території аеропортів.

Аеронавігація має два напрямки – цивільна, що використовується у цивільній авіації та тактична, яка використовується військовими.

Цивільна навігація зазвичай використовує цивільні станції радіолокації і загалом є відкритою для бажаючих виробників літаків у плані використання. Наприклад, на території аеропорту може стояти радіолокаційна станція, яка показана на рисунку 1.1. Такі станції також називають «маяки».

Радіолокаційна станція (РЛС), радар – система для виявлення повітряних, морських та наземних об'єктів, а також для визначення їх дальності, швидкості та геометричних параметрів. Використовує метод радіолокації, що ґрунтується на випромінюванні р та реєстрації їх відбиття від об'єктів [2].

Радіолокаційні станції мають такі характеристики:

- висока оперативність отриманих даних до яких входять координати та додаткова інформація;
- висока ступінь об'єктивності отриманих даних;
- повнота інформації про повітряну обстановку у зоні контролю;
- наочність інформації, що показується;

– висока точність та надійність.

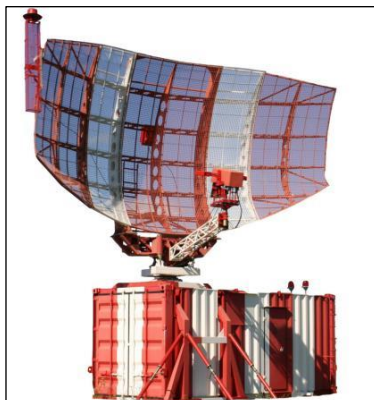


Рисунок 1.1 – Цивільна радіолокаційна станція

Наземні станції мають високий пріоритет у тактичній військовій навігації адже вони дозволяють працювати у несприятливих умовах, зменшити кліматичні впливи на апаратуру станції, збільшити площу антени для посилення радіолокації, використовувати резервування даних, а також легші в обслуговуванні порівняно із повітряними чи космічними засобами.

Наприклад, у країнах, що входять до блоку НАТО, поширене застосування системи TACAN (Tactical Air Navigation System) на рівні із GPS навігацією. Військова тактична навігація використовується при польотах військових літаків, вертольотів, наведення безпілотних літальних апаратів та ракет. Вигляд станції TACAN представлено на рисунку 1.2.



Рисунок 1.2 – Станція TACAN

Система тактичної радіонавігації TACAN поширена у країнах-членах блоку НАТО. В Україні ця система представлена цивільним аналогом VOR/DME. Ця система в майбутньому може бути використана в нашій країні.

Перед тим, як її використовувати буде відбуватися довгий процес налаштування та іспитів системи «на землі» для отримання експериментальних результатів та коригування параметрів. Для того, щоб зменшити витрати пропонується створення програмного засобу-емулятора, що імітує процеси, які відбуваються всередині системи TACAN на рівні комунікації між станціями «маяками» та бортовою системою літаків. Програмний засіб має ефективно імітувати роботу всіх компонентів системи та наглядно виводити інформацію під час проведення комп'ютерних експериментів. Таким чином дана робота є актуальною у наш час.

1.2 Огляд наукової літератури

Після аналізу останніх оглядів [1-3] можна побачити, що в останні роки стабільно швидко розвиваються технології пов'язані з побудовою та візуалізації симуляторів у робототехніці, медицині та транспорті. Використання сучасних інформаційних технологій та комп'ютерної техніки дозволяє максимально точно змодельовати реальні ситуації у предметній області при створенні симуляторів. Забезпечуючи таким чином максимальний ефект навчання при використанні сучасних симуляторів. В цьому відношенні актуальним є розвиток і трансфер технологій у сферу розробки комп'ютерних симуляторів РСБН [4-6].

У цьому відношенні в роботі [4] запропонована модель підвищення ефективності позиціонування за рахунок спільного позиціонування в частковому середовищі GNSS в мережах спеціального призначення. У роботі [6] представлено опис моделі та технологій моделювання сценаріїв, а також генерації кількох цілей з індивідуальними діапазонами та швидкостями для радарів безперервного сигналу з частотно-модульованою послідовністю ЛЧМ. У роботі [7] представлені результати досліджень у галузі позиціонування на основі магнітної локації – представлений новий інструмент моделювання магнітного

поля з урахуванням специфікації та галузі локалізації. У роботі [8] представлено інтелектуальні технології навігації малого радіусу дії на основі IoT. У роботі [9] пропонується новий алгоритм для локалізації транспортного засобу ближнього радіусу дії для того, щоб компенсувати розрив відстані по GPS-приладу. У цьому вирішується проблема низького ставлення сигналу шуму. У роботі [10] вирішується проблема використання стаціонарного калібрувального пристрою систем навігації для перевірки функцій прийому DME та тестера DME ATC-1400A. У роботі [11] дається розв'язання задачі використання нового програмного забезпечення для ефективної обробки радіонавігаційних сигналів, зменшення впливу перешкод, збільшення ефективності спектральних сигналів.

Пропонується спектр комп'ютерних моделей, описуються результати лабораторних досліджень конструкцій FBMC. У роботі [12] вирішується завдання побудови лінійного регулятора польоту з параметром посилення, що змінюється, для підвищення ефективності польотних симуляторів (IFS) в умовах великої параметризації. У роботі [13] запропоновано ефективні алгоритми та симулятор для моделювання циклу навігації по дорожніх точках на карті місцевості. У роботі [14] наведена модель динамічного симулятора польоту літака, що низько летить, з декількома ступенями свободи. У роботі [15] запропоновано нову прогностичну модель на основі методу зворотного введення. У роботі [16] запропоновано нову технологію використання багатоядерних платформ для автоматизації бортової обробки даних, а також відповідну тестову платформу для оцінки ефективності використання системи автоматичного пілотування. У роботі [17] наведено алгоритми та моделі, на основі яких побудовано симулятор перевірки системи попередження про зліт літака, включаючи візуалізацію панелі управління. У роботі [18] наведено результати використання генетичного алгоритму для оптимізації деяких процесів усередині симуляторів авіації. У роботі [19] представлені результати імітаційного моделювання елементів системи навігації на часовій шкалі для оцінки ефективності їхньої спільної роботи та взаємного впливу один на одного.

Аналіз сучасного стану питання показує значний прогрес у галузі побудови різноманітних симуляторів та актуальність застосування розроблених у них моделей, алгоритмів та технологій, у тому числі у галузі візуалізації, для побудови симуляторів радіотехнічних систем ближньої навігації з урахуванням вимог часу. У тому числі для радіотехнічної системи ближньої навігації TACAN [20, 21].

Ключові вимоги до такого симулятора - узгоджений облік всього комплексу факторів впливу, високий рівень візуалізації, робота в реальному часі на реальних картах місцевості, можливість оперативного коригування ситуації, роботи з польотними сценаріями [20-22].

У плані розвитку таких симуляторів найбільш актуальними є аспекти впровадження: сучасних систем обробки зображень та обчислювального інтелекту [23, 24] для покращення сприйняття, навчання та цілісності обробки потоків даних; останніх рішень у галузі мережевої оптимізації [25] та апаратних технологій [26, 27] для високої ефективності мережної комунікації; сучасних методів та засобів реалізації інтерфейсу та візуалізації симулятора для забезпечення високого рівня навчання персоналу в системах e-learning відповідно до сучасних критеріїв оцінювання [28, 29].

Основною задачею дослідження є дослідження внутрішніх процесів роботи радіонавігаційної системи ближньої навігації, порівняння існуючих алгоритмів розрахунку параметрів повітряних цілей на радіолокаційних станціях та розробка моделей та алгоритмів роботи елементів РСБН TACAN для роботи у програмному симуляторі, що робить візуалізацію роботи всіх її елементів.

2 РОЗРОБКА МОДЕЛЕЙ, АЛГОРИТМІВ ТА МЕТОДІВ

У розділі наведено принципи організації та функціонування радіотехнічної системи ближньої навігації TACAN. Системи ближньої навігації призначені для визначення навігаційних параметрів, що характеризують положення літака в полярній системі координат (азимут, дальність) щодо радіонавігаційної точки (радіомаяку), положення якої на місцевості точно відоме [30].

Основою системи є мережа незалежних наземних радіомаяків, якими визначаються навігаційні параметри. Розрізняють азимутальні, далекомірні та азимутально-далекомірні радіомаяки, які розміщують на аеродромах та в точках, характерних для ділянок повітряних трас. Для роботи виділено ділянки діапазону метрових та дециметрових хвиль. Тому їхня дальність дії обмежена дальністю прямої видимості [30, 31].

Основне призначення системи ближньої навігації – приведення літака до аеродрому посадки та забезпечення навігації при польоті маршрутом над континентальною частиною земної поверхні.

За класифікацією радіотехнічні системи ближньої навігації відносять до кутомірно-дальномірних радіонавігаційних систем, в яких місцезнаходження літака визначають на борту перетину ліній положення, рівних його азимуту і дальності щодо наземного радіомаяка. До складу таких систем входять азимутальний (кутомірний) канал, далекомірний канал та канал наземної індикації. Всі три канали використовуються як наземні, так і бортові пристрої (комплекс наземних та бортових пристроїв) [32].

2.1 Принцип роботи радіолокаційної системи

У будь якій радіолокаційній системі використовуються схожі принципи та методи роботи радіолокаційних систем як на апаратному рівні так і на рівні розрахункового алгоритмічного апарату. Принцип роботи радіолокатора схожий на принцип відбиття звукових хвиль. На приклад, якщо ми знаємо швидкість розповсюдження звуку та крикнемо у напрямку об'єкта, що може відбивати звук, ми можемо визначити напрямок та дальність до об'єкта.

Радіолокатор використовує електромагнітні імпульси. Радіолокатор випромінює деякий потужний сигнал, його називають «зондувальний», який відбивається від об'єкта та невелика частка від сигналу повертається назад до радіолокатора. Для обробки відбитого сигналу використовується приймач високої чутливості. Далі внутрішня система розраховує дальність та напрямок і видає інформацію на екран. В залежності від самого радіолокатора та устаткування, можливо визначати такі параметри як висоту, азимут, курс, швидкість та дальність повітряних об'єктів. Схема роботи радіолокаційної системи показана на рисунку 2.1.

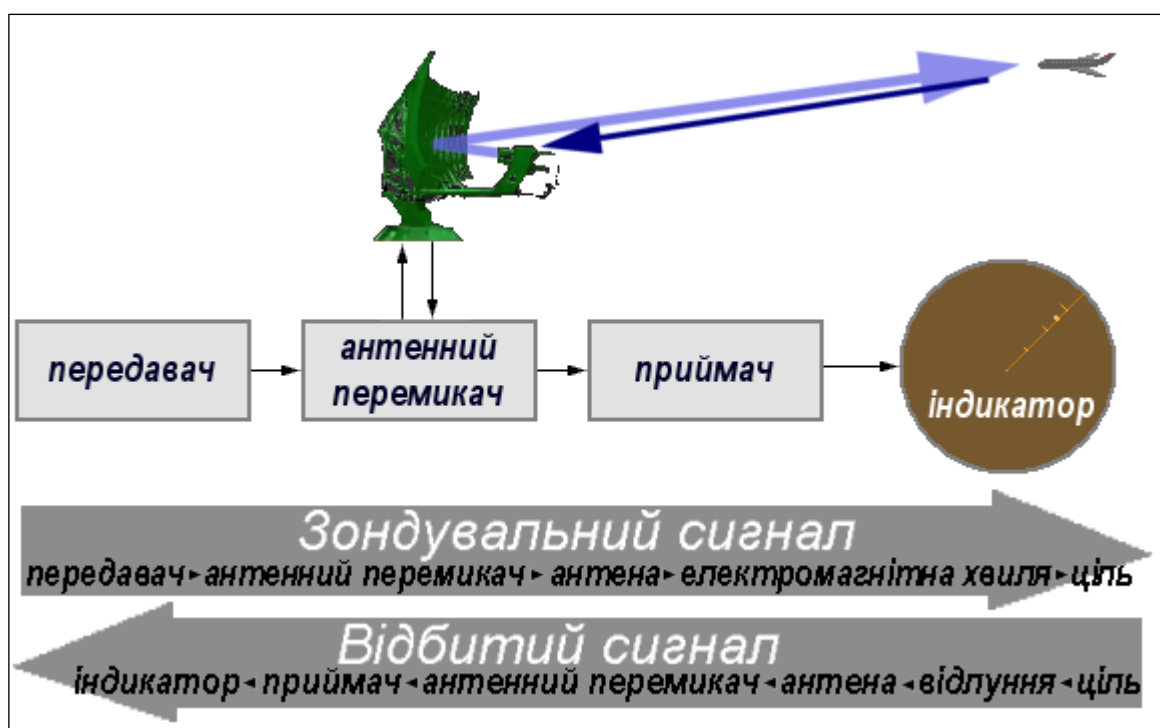


Рисунок 2.1 – Схема роботи радіолокаційної станції

Таким чином стандартний радіолокатор складається з таких елементів устаткування:

- передавач – генерує короткий радіоімпульс великої потужності, що випромінюється антеною у простір;
- дуплексер – підключає до антени передавач або приймач;
- приймач – приймає та виконує демодуляцію отриманих радіочастотних сигналів;

- антена – випромінює енергію передавача у простір з заданою ефективністю та розподіленням. Таким же чином приймає відбитий сигнал;
- індикатор – надає спостерігачу зручне та безперервне графічне зображення об'єктів. Радіолокаційний дисплей відображає відмітки в місці прийому відбитих сигналів. Чим більше часу пройшло між випроміненим та отриманим сигналом, тим далі від центру екрану буде відмітка;

2.2 Алгоритми визначення параметрів об'єктів в РСБН

Так як поширення радіохвиль у просторі відбувається з постійною швидкістю, що дорівнює швидкості світла c_0 , то дальність може бути визначена за часом затримки прийнятого сигналу по відношенню до сигналу, що випромінюється, зондуючого. Дальність, що вимірюється від раіолокатору до повітряного об'єкту, називається похилою дальністю до цілі. Для визначення потребує інформації про висоту польоту цілі або куту місцевості. Так як електромагнітна хвиля проходить шлях до цілі та назад, то час ділиться навпіл. Так формула розрахунку похилої дальності до цілі показана нижче (формула 2.1), а схема показана на рисунку 2.2.

$$R = \frac{c_0 * t}{2} \quad (2.1)$$

де $c_0 = 3 * 10^8$ м/с.

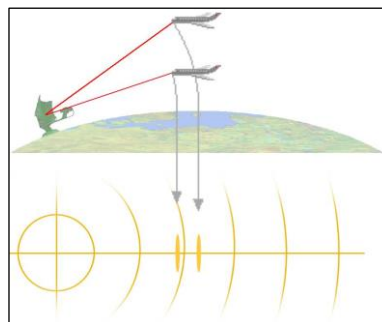


Рисунок 2.2 – Схема розрахунку похилої дальності цілі

Азимутом цілі називають кут між напрямом на північ та горизонтальною проекцією напрямку на ціль. Цей кут розраховується за хвилиною стрілкою від направленості на північ. Також азимут може визначатися від центральної вісі літака або місця, з якого ведеться вимір, тоді він буде відносним. Азимут цілі вимірюється за направленням головного проміння антени в момент приходу відбитого від цілі сигналу. Спрямованість антени (характеризується значенням коефіцієнта спрямованої дії) – це здатність антени фокусувати енергію у заданому напрямку. Антена з високим коефіцієнтом спрямованої дії називається спрямованою. За положенням спрямованої антени в момент прийому відбитого імпульсу, можна визначити азимут і кут місця мети. Точність виміру кутових координат залежить від спрямованих властивостей антени (див.рис.2.3).

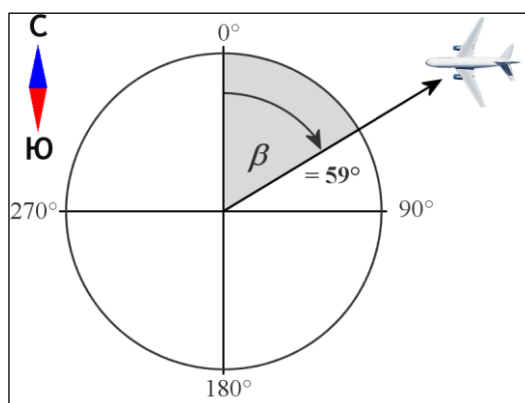


Рисунок 2.3 – Схема розрахунку азимуту цілі

2.3 Принципи та алгоритми роботи РСБН TACAN

Система ближньої навігації TACAN має два канали далекомірний та азимутальний [21]. Конфігурація системи TACAN показана на рисунку 2.4.

Дальномірний канал системи TACAN складається з запитача, встановленого на борту повітряного судна, і радіомаяка, що передає відповіді. Залежно від рівня, що генерується, еквівалентної ізотропно-випромінюваної потужності та конструкції запитувача може бути виміряна похила дальність до 400 морських миль (740 км), але на практиці діапазон вимірювань обмежується максимальною дальністю прямої видимості радіозасобів (RLOS). Передавач повітряного судна

випромінює регулярні пари імпульсів, звані імпульси запиту, прийняті установками наземного базування (радіомаяками).

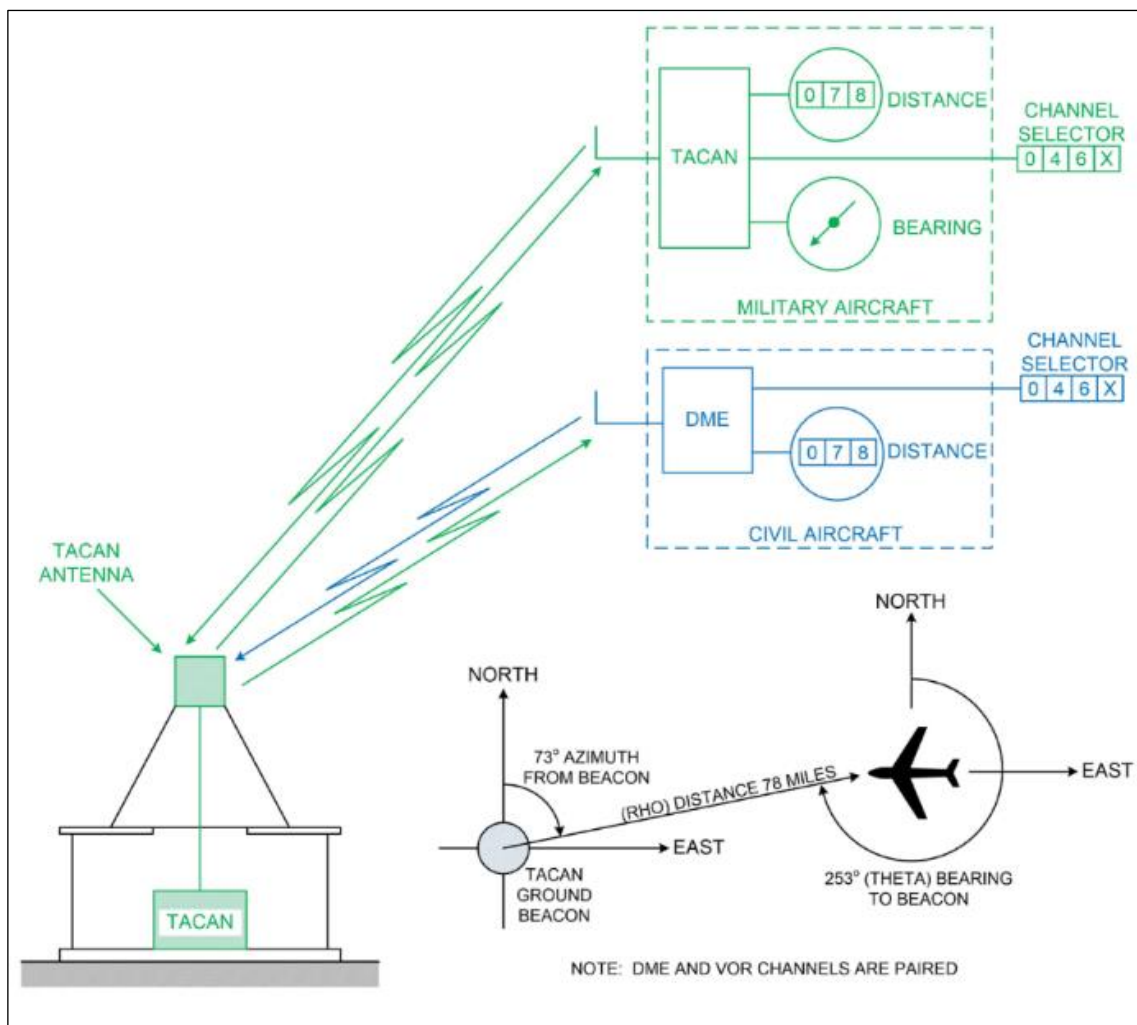


Рисунок 2.4 – Радіотехнічна система ближньої навігації TACAN [33]

Імпульси системи TACAN мають тривалість 3,5 мкс при 50-відсотковому рівні амплітуди. Тимчасове рознесення між парою імпульсів запиту становить 12 мкс (канал X) або 36 мкс (канал Y). Після прийому пари імпульсів запиту наземна станція перевіряє форму імпульсів та інтервал між ними. Якщо ці параметри перебувають у допустимих межах, вона передає відповідь після фіксованої затримки зі зміщенням частоти ± 63 МГц за частотою запиту залежно від вибраного каналу імпульсного коду. Інтервал між імпульсами відповіді радіомаяка становить 12 мкс (канал X) та 30 мкс (канал Y). Після прийому відповіді запитувач розраховує миттєву дальність похилої радіомаяка за часом,

що пройшов між прийомом пари імпульсів запиту та отриманням пари імпульсів відповіді.

Азимутальний канал системи TACAN надає інформацію про азимутальне положення літака. Азимутальна інформація забезпечується за рахунок застосування амплітудної модуляції переданих імпульсів наземним радіомаяком. Така амплітудно-імпульсна модуляція (AIM) створюється за допомогою використання антени радіомаяка з механічним або електронним скануванням. Зміна азимутальної діаграми спрямованості у вигляді пелюсток шириною 15 – 135 Гц при максимально допустимому коефіцієнті модуляції 55 відсотків знижує рівень сигналу імпульсів відповіді на величину до 10,7 дБ нижче максимального рівня еквівалентної ізотропно-випромінюваної потужності імпульсів. Для того, щоб запитувач міг декодувати дані з AIM про орієнтацію діаграми спрямованості антени у напрямку на північ, радіомаяк передає ще 900 імпульсних пар, що складаються з групи імпульсів напрямку на північ (NRPG) та додаткових еталонних точних груп імпульсів (RPG). Для отримання точної азимутальної інформації, і щоб мати можливість відповідати не менше 100 повітряним суднам з 70% ефективністю відповідей, необхідно постійно передавати щонайменше 3600 імпульсних пар.

Інформація TACAN про дальність/азимут використовується для підтримки операцій на маршруті, прибуття та вильоту, а також операцій заходу на посадку, посадки та зближення. Для цих застосувань TACAN можна використовувати в поєднанні з іншими навігаційними засобами, такими як RADAR точного заходу на посадку та система посадки приладів (ILS). Крім того, пілоти використовують інформацію про азимут TACAN для опублікованих схем польоту та маршрутів як за приладами, так і за візуальних метеорологічних умов (IMC та VMC відповідно).

2.4 Вимоги до програмного симулятора

Призначення симулятора, що розробляється – імітація сигналів радіомаяка і літака, емуляція обладнання, що генерує і приймає ці сигнали, а також

візуалізація навігаційного обладнання на стороні радіомаяка і літака в прив'язці до карти місцевості.

Стоїть задача в створенні програмного забезпечення, що має реалізовувати процеси взаємодії у середині системи радіонавігації TACAN. Програмне забезпечення має наочно показувати процеси, що відбуваються та параметри в кожному мить часу. Для наочності треба відобразити літак та станції на мапі. Літак може літати за задалегіть зазначеними шаблонними маршрутами та взаємодіяти зі станціями. Станції можуть додаватися чи видалятися користувачем, що проводить експерименти, а також допускається зміна параметрів станцій. Користувач також має змогу додавати нові маршрути та зберігати їх.

Планується багатоцільове використання програмного симулятора:

- побудова сценаріїв польоту літака;
- модулювання радіосигналів, що проходять в дальномірному та азимутальному каналах;
- візуалізація устаткування, що приймає та передає, що виконує формування та виділення складових сигналу;
- візуалізація навігаційних приладів, що контролюють радіомаяк та відображають положення літака щодо маяка: азимут (пеленг та відстань);
- верифікація реальних сигналів та обладнання радіотехнічної системи ближньої навігації TACAN.

Програмне забезпечення має бути модульним та складатися з таких модулів:

- модуль «Літак»;
- модуль «Станція».

Модуль «Літак» представляє собою довершений незалежний модуль, у якому зберігаються шаблони польотів літаків. Він дозволяє створювати нові шаблони і зберігати їх, а також видаляти наявні шаблони. Дозволяє запуск шаблонів, що ініціює сценаріїв переміщення літака по заданому маршруту. Має в кожному момент часу ідентифікувати положення літака і надавати цю інформацію.

Положення літака за цим модулем означається координатами довготи та широти (lat та lng відповідно).

Модуль «Станція» представляє собою довершений незалежний модуль, який зберігає розташування станцій. Розташування станцій задається координатами довготи та широти (lat та lng відповідно). Має дозволяти розстановку нових станцій, видалення та редагування параметрів вже існуючих станцій. Має відображати характеристики станції при запиті. Під час польоту літака деяким чином має визначати положення цього літака відносно кожної станції та передавати цю інформацію на бортову систему літака.

Вимоги до симулятора розроблені відповідно до виконання завдань щодо призначення TACAN (забезпечення вимірювання на борту літака його азимуту та похилої дальності щодо місця встановлення наземного обладнання (радіомаяку), який має ідентифікуватися на літаку), необхідності забезпечення сумісності бортового та наземного обладнання РСБН TACAN, аналізу вимог ICAO [34] до радіонавігаційних засобів авіаційного електрозв'язку, сертифікаційних вимог до наземних засобів радіотехнічного забезпечення в цивільній авіації, вимог стандарту MIL-STD-291C [35], рекомендації МСЕ ITU-R M.2013 [36] та аналіз тактико-технічних характеристик сучасних транспондерів (приймально-відповідачів) TACAN [37].

Симулятор наземного обладнання має імітувати роботу в діапазоні частот від 962 до 1213 МГц. Модель далекомірного каналу систем РСБН типу TACAN забезпечує визначення похилої дальності до ПВС щодо радіомаяка залежно від висоти польоту літака у зоні. Азимутальний канал РСБН типу TACAN забезпечує визначення азимуту літака щодо радіомаяка. У горизонтальній площині симулятор наземного обладнання повинен мати зону дії 360 °. При визначенні ширини зони дії по куту місця цілком прийнятним є здійснення огляду вертикальної площини сектора кутів місця від 0° до 45°. Розмір зони огляду за дальністю визначається теоретично можливою дальністю радіозв'язку, яка визначається як дальність радіогоризонту для висот польоту літака. Для висоти 20 км. дальність радіогоризонту становить 550 км.

Точність вимірювання азимуту та похилої дальності визначається бортовим обладнанням та відповідністю сигналів наземного обладнання TACAN вимогам їх характеристик. Симулятор наземного обладнання повинен забезпечити прийом, формування та випромінювання сигналів. При цьому відхилення величини затримки сигналів відповідності дальності в наземному транспондері (радіомаяку) не повинно перевищувати $\pm 0,1$ мкс, що обумовлює внесення помилки ± 15 м у загальну помилку вимірювання дальності на борту літака.

Точність вимірювання азимуту має становити $1/3$ градуси. Точність виміру похилої дальності повинна становити 926 м ($\pm 0,5$ милі) або 3% від дальності, залежно від того, що більше.

Для роботи наземного обладнання необхідно відвести 252 частотні канали в діапазоні від 962 до 1213 МГц з кроком 1 МГц. Це дозволяє реалізувати 126 частотно-кодових каналів X і 126 частотно-кодових каналів Y.

3 РОЗРОБКА ПРОТОТИПУ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

У даному розділі надається опис архітектури програмного забезпечення симулятора РСБН системи ТАСАН, його функціональних можливостей та модулів візуалізації; описується планування та проведення експериментів із застосуванням запропонованого симулятора в різних режимах роботи. Результати експериментів та їх інтерпретація наводяться у двох розділах.

3.1 Вибір архітектури програмного забезпечення

Для розробки програмного забезпечення для виконання у веб середовищі найбільш поширені у використанні два архітектурних підходи такі як монолітні програмні додатки та додатки, розділені на сервіси. Нижче наведено опис та порівняння цих підходів з виділенням переваг та недоліків підходів.

Монолітна архітектура – це традиційна модель програмного забезпечення, яка є єдиним модулем, що працює автономно і незалежно від інших додатків. Моноліт часто називають щось велике і неповоротке, і ці два слова добре описують монолітну архітектуру для проектування ПЗ. Монолітна архітектура – це окрема велика обчислювальна мережа з єдиною базою коду, де об'єднані всі бізнес-завдання. Щоб внести зміни в таку програму, необхідно оновити весь стек через базу коду, а також створити та розгорнути оновлену версію інтерфейсу, що знаходиться на стороні служби. Це обмежує роботу з оновленнями та потребує багато часу. Приклад схеми монолітної архітектури показано на рисунку 3.1.

Можна отримати вигоду як з монолітної архітектури, так і з мікросервісної в залежності від різних факторів. При використанні монолітної архітектури зручно створювати програми на основі однієї бази коду, тому її основна перевага полягає у швидкості розробки.

Моноліти зручно використовувати на початкових етапах проектів, щоб полегшити розгортання та не витратити надто багато розумових зусиль при керуванні кодом. Це дозволяє одразу випускати все, що є в монолітному додатку.

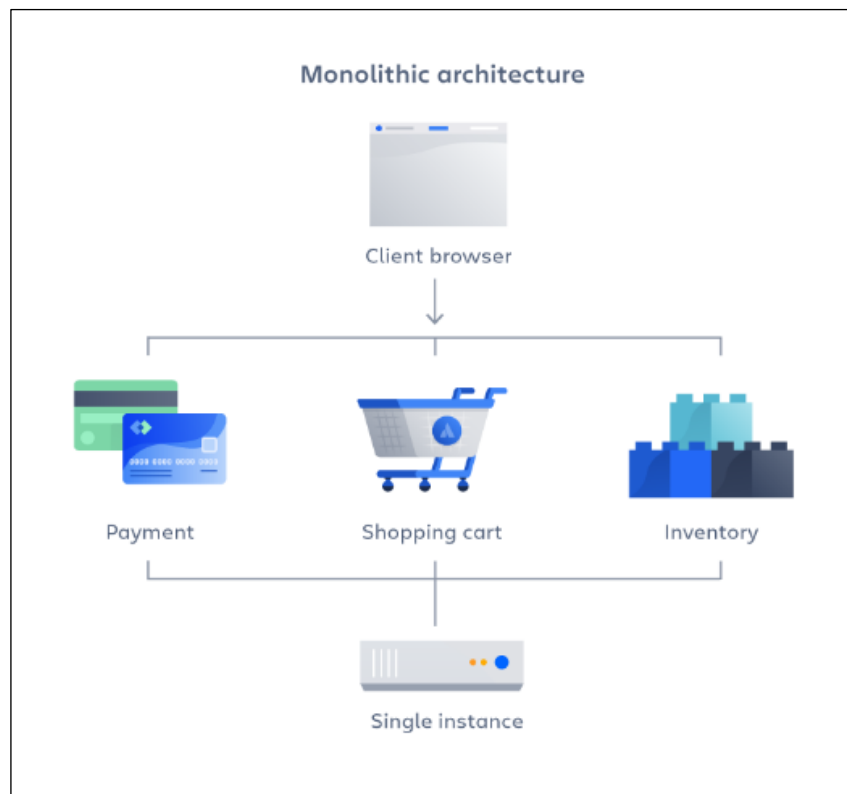


Рисунок 3.1 – Схема монолітної архітектури ПЗ

До переваг монолітної архітектури можна віднести такі особливості:

Просте розгортання. Використання одного файлу або каталогу, що виконується, спрощує розгортання:

- розробка. Програма легше розробляти, коли вона створена з використанням однієї бази коду;
- продуктивність. У централізованій базі коду та репозиторії один інтерфейс API часто може виконувати ту функцію, яку під час роботи з мікросервісами виконують численні API;
- спрощене тестування. Монолітне додаток є єдиним централізованим модулем, тому наскрізне тестування можна проводити швидше, ніж при використанні розподіленої програми;
- зручне налагодження. Весь код знаходиться в одному місці, завдяки чому стає легше виконувати запити та знаходити проблеми.

Монолітні програми працюють досить ефективно до тих пір, поки вони не стають занадто великими і не викликають проблем із масштабуванням. Щоб

внести невелику зміну в одну функцію, необхідно виконати компіляцію та тестування всієї платформи, що суперечить agile-підходу, якому віддають перевагу сучасні розробники.

До недоліків монолітної архітектури можна віднести такі особливості:

- зниження швидкості розробки. Великий монолітний додаток ускладнює та уповільнює розробку;
- масштабованість. Неможливо масштабувати окремі компоненти;
- надійність. Помилка в одному модулі може вплинути на доступність програми;
- перешкоди запровадження технологій. Будь-які зміни в інфраструктурі чи мові розробки впливають на додаток цілком, що часто призводить до збільшення вартості та тимчасових витрат;
- недостатня гнучкість. Можливості монолітних додатків обмежені технологіями, що використовуються;
- розгортання. При внесенні невеликої зміни буде потрібно повторне розгортання всього монолітного додатку.

Мікросервісна архітектура (або просто «мікросервіси») являє собою метод організації архітектури, заснований на ряді служб, що незалежно розгортаються.

Ці служби мають власну бізнес-логіку та базу даних з конкретною метою. Оновлення, тестування, розгортання та масштабування виконуються всередині кожної служби. Мікросервіси розбивають великі завдання, характерні для конкретного бізнесу, на кілька незалежних баз коду. Мікросервіси не знижують складність, але вони роблять будь-яку складність видимою і більш керованою, поділяючи завдання на дрібніші процеси, які функціонують незалежно один від одного і роблять внесок у загальне ціле. Приклад схеми архітектури з використанням мікросервісів показано на рисунку 3.2.

Використання мікросервісів часто тісно пов'язане з DevOps, оскільки вони лежать в основі методики безперервного постачання, яка дозволяє командам швидко адаптуватися до вимог користувачів.

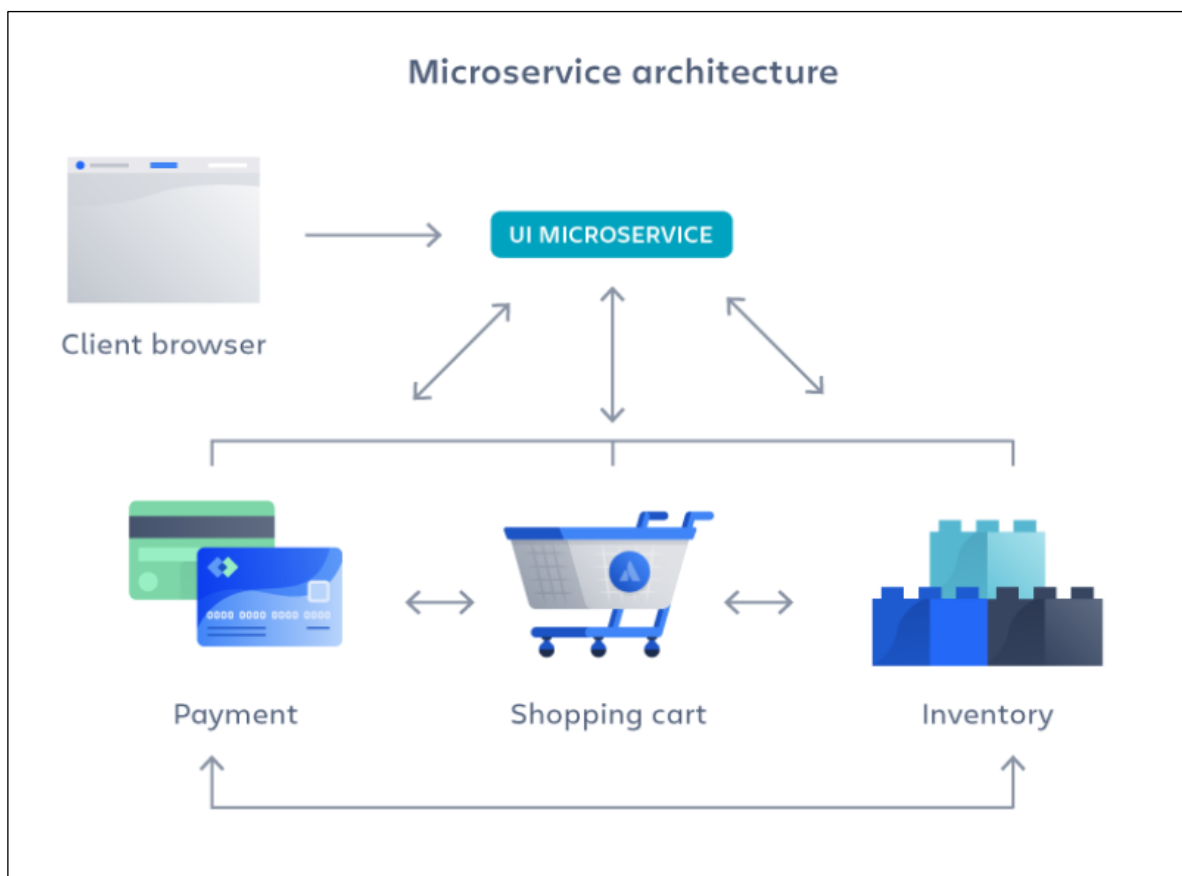


Рисунок 3.2 – Схема архітектури ПЗ з використанням мікросервісів

Мікросервіси не є чарівною паличкою, але вони вирішують низку проблем, з якими стикаються компанії, що ростуть, при розвитку ПЗ. Оскільки архітектура мікросервісів складається з незалежно працюючих модулів, кожену службу можна розробляти, оновлювати, розгортати та масштабувати окремо від інших. Оновлення можна виконувати частіше, підвищуючи надійність, час безперебійної роботи та продуктивність програмного забезпечення.

При такому підході можна на багато частіше випускати оновлення, що будуть дотичні тільки до тих частей системи, що оновлюються а не до всього продукту в цілому. Крім того, мікросервіси спрощують для команд оновлення коду та прискорюють цикли релізу завдяки безперервній інтеграції та безперервній поставці (CI/CD).

Команди можуть поекспериментувати з кодом і повернутись до попередньої версії, якщо щось піде не так.

Отже, мікросервіси дають такі переваги:

- гнучкість. Просувайте гнучкі методи роботи серед невеликих команд, які регулярно розгортаються;
- гнучке масштабування. Коли мікросервіс досягає граничного навантаження, можна швидко виконати розгортання нових екземплярів цієї служби у супутньому кластері та знизити навантаження. Тепер ми працюємо з кількома власниками та без збереження стану, а клієнти розподілені за різними примірниками. З таким підходом ми можемо підтримувати екземпляри значно більшого розміру;
- безперервне розгортання. Тепер у нас є регулярні та прискорені цикли релізу. Раніше ми випускали оновлення раз на тиждень, а тепер можемо робити це приблизно двічі-тричі на день;
- легкість обслуговування та тестування. Команди можуть експериментувати з новими функціями та повертатися до попередньої версії, якщо щось не працює. Це спрощує оновлення коду та прискорює випуск нових функцій на ринок. Крім того, в окремих службах легко знаходити та виправляти помилки та баги;
- незалежне розгортання. Мікросервіси є окремими модулями, тому з ними можна легко і швидко виконувати незалежне розгортання окремих функцій;
- гнучкість технологій. У разі використання архітектури мікросервісів команди можуть вибирати інструменти з урахуванням своїх переваг;
- висока надійність. Розгортаючи зміни для конкретної служби, можна не боятися, що програма вийде повністю;
- задоволені команди. Команди, що працюють з мікросервісами, набагато краще відгукуються про свою роботу завдяки автономності та можливості самостійно створювати та розгортати програми, не чекаючи схвалення запиту pull протягом кількох тижнів.

Мікросервіси можуть зробити процес розробки складнішим і призвести до його розростання – швидкого та некерованого зростання. Іноді буває складно визначити, як різні компоненти пов'язані один з одним, хто володіє конкретним

програмним компонентом або як уникнути втручання у роботу залежних компонентів.

До недоліків мікросервісів можна віднести такі особливості.

- розростання процесу розробки. Мікросервіси ускладнюють роботу порівняно з монолітною архітектурою, оскільки у різних місцях виникає дедалі більше служб, створених кількома командами. Якщо розростання не контролюється належним чином, воно призводить до уповільнення розробки та зниження операційної ефективності;
- експонентне зростання витрат на інфраструктуру. Кожен новий мікросервіс може мати свою вартість комплекту тестів, інструкцій з розгортання, інфраструктури хостингу, інструментів моніторингу тощо;
- додаткові організаційні витрати. Командам потрібний додатковий рівень комунікації та співробітництва, щоб координувати роботу над оновленнями та інтерфейсами;
- проблеми при налагодженні. Кожен мікросервіс має свій набір журналів, що ускладнює налагодження. Крім того, додаткові труднощі можуть виникати у тому випадку, коли один бізнес-процес виконується на кількох машинах;
- відсутність стандартизації. Без загальної платформи може виникнути ситуація, де розширюється список мов, стандартів ведення журналів та засобів моніторингу;
- відсутність ясності у питаннях володіння. У міру появи нових служб збільшується і кількість команд, що працюють над ними. Згодом стає складніше визначити, які служби команда може використовувати і до кого слід звертатись за підтримкою.

Поставленій задачі емуляції кожного окремого компоненту системи найбільш задовольняє архітектура, що будується на малих окремих сервісах, адже це надає найбільшій степені відокремленості кожного елементу системи, а взаємодія між ними може бути побудована з часовими затримками наче зв'язок елементів у реальних умовах у просторі.

3.2 Опис архітектури програмної реалізації РСБН TACAN

Відповідно до постановки завдання розроблено симулятор РСБН системи TACAN, який ми будемо використовувати для демонстрації та проведення експериментів. Запропонований симулятор (у вигляді спеціалізованого програмного забезпечення) здійснює програмну емуляцію роботи реального обладнання компонентів системи TACAN, таких як маяки системи TACAN і бортове обладнання літака.

Було розроблено програму з використанням Web-технологій за принципами "клієнт-серверної" архітектури. Загальна архітектура програмного забезпечення симулятора показана на рисунку 3.3.

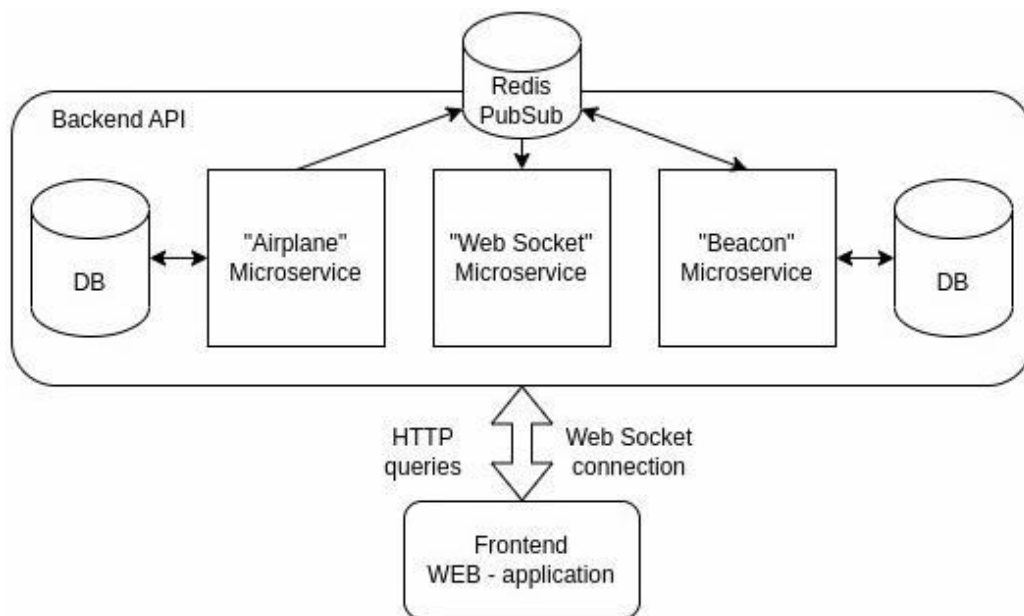


Рисунок 3.3 – Архітектура програмного забезпечення

Backend API побудована з використанням мікросервісів та поділена на логічні частини обробки літака та обробки маяків у системі TACAN. Для роботи мікросервісів використовується середовище виконання Node.js, а мікросервіси написані з використанням мови програмування TypeScript, яка є надбудовою над JavaScript і має сувору типізацію та бібліотеку express, яка надає компоненти для серверної логіки.

Мікросервіс Airplane відповідає за обробку літаків, він дозволяє створювати і зберігати сценарії польоту літаків і запускати їх на вимогу.

Мікросервіс “Beacon” дозволяє зберігати розстановку станцій, зберігаючи положення маяків на карті за координатами та їх параметри, а також подальше редагування параметрів маяків, наприклад, радіус видимості.

Всі мікросервіси пов'язані між собою подієвою шиною, що реалізується Redis PubSub функціональністю. Так, після запуску сценарію польоту, кожну секунду Redis PubSub буде відправлятися сигнал з інформацією про положення літака і його параметрами такими як швидкість, висота і напрямок польоту.

Мікросервіс "Web Socket" підписаний на інформацію про положення літака, а також інформацію з маяків і служить сполучною частиною між Backend API і Frontend Web - application для постійної передачі даних реального часу для показу на інтерфейсі користувача симулятора.

3.3 Архітектура Backend сервісу «Станція»

Для найбільшого наближення до роботи реального обладнання було вирішено робити відокремлений невеликий сервіс, що відповідає за роботу окремого елемента системи, а саме станції радіонавігаційної системи TACAN, а також емуляцію проходження сигналів у просторі за допомогою шини подій.

Сервіс представляє собою міні додаток, що побудовано за MVC (Model-View-Controller) архітектурою.

Сутність станції та її представлення у базі даних, якою оперує мікросервіс «Станція» показана на рисунку 3.4.

<i>Station</i>	<i>stations</i>
id: number	id: INTEGER, AUTOINCREMENT
title: string	title: VARCHAR(255)
coord_lng: number	coord_lng: FLOAT
coord_lat: number	coord_lat: FLOAT
frequency_hz: number	frequency_hz: INTEGER
radius_m: number	radius_m: INTEGER
created_at: Timestamp	created_at: TIMESTAMP
updated_at: Timestamp	updated_at: TIMESTAMP

Рисунок 3.4 – Сутність «Станція» та її представлення у БД

Сутність «Станція» має такі поля:

- id – унікальний ідентифікатор;
- title – назва станції, наприклад за назвою аеропорту;
- coord_lng – координата широти станції;
- coord_lat – координата довготи станції;
- frequency_hz – робоча частота станції;
- radius_m – радіус дії станції;
- created_at – дата створення станції;
- updated_at – дата оновлення станції;

Архітектура модулю «Станція» показана на рисунку 3.5. Модуль побудовано з використанням MVC архітектури і бібліотеки express.

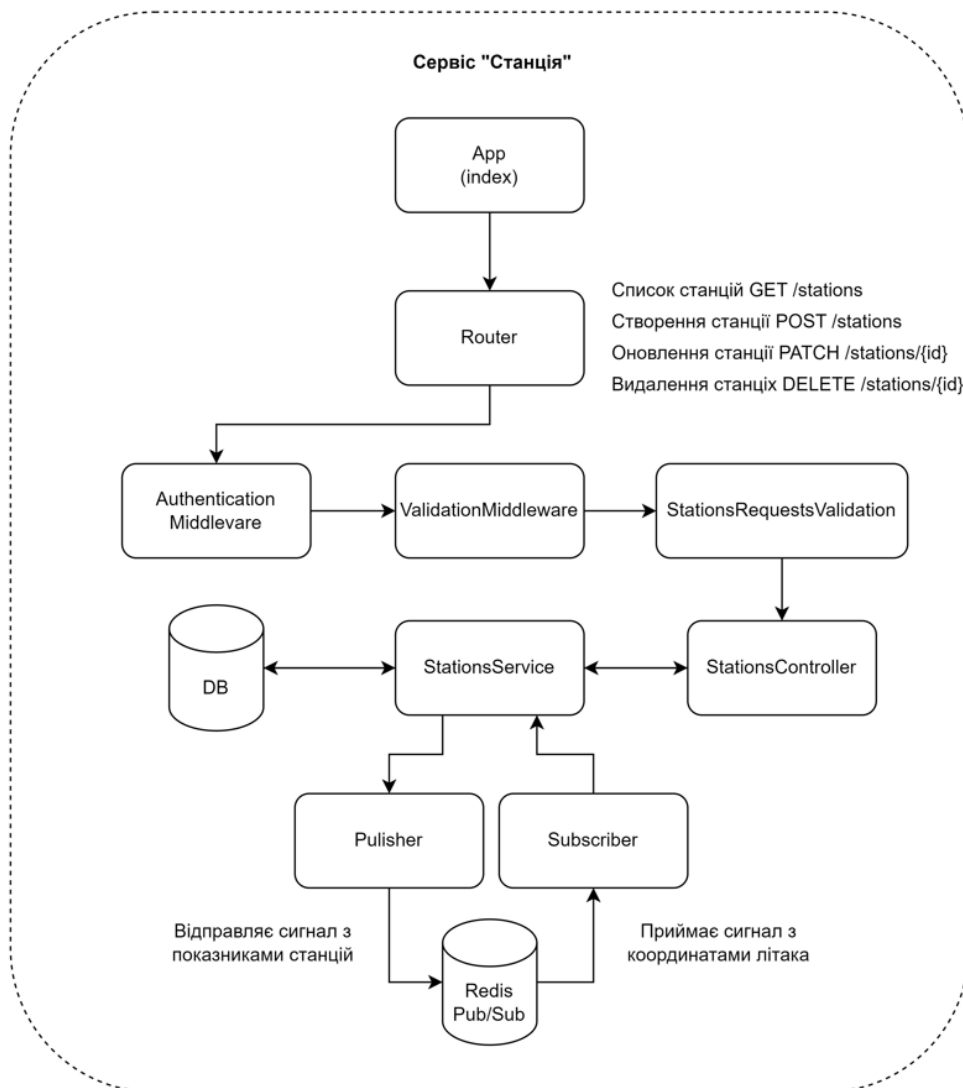


Рисунок 3.5 – Архітектура сервісу «Станція»

Початок модуль бере у файлі `App`, що є вхідною точкою ініціалізації роботи додатку, в ній запускається у роботу веб сервер та слухач запитів, створюється `Router`, який слухає запрограмовані «адреси» запиту до веб-серверу. При запиті по кожній «адресі» проводиться перевірка авторизації користувача у файлі `AuthenticationMiddleware`. Після цього йде перевірка даних, з якими клієнт робить запит, наприклад коректне заповнення полів форми при створенні нової станції у `ValidationMiddleware` та `StationsRequestsValidation`. Після успішних перевірок запит потрапляє у контроллер запитів, де проходить подальша обробка та повернення клієнту деякого результату запиту.

Клас `StationsController` має такі методи:

- `create` – створення нового станції;
- `update` – оновлення параметрів станції;
- `fetchAll` – отримання всього списку станцій;
- `deleteById` – видалення станції за ідентифікатором.

Так як контролер відповідає за отримання запиту та відправку відповіді, між цими двома діями він передає керування сервісу `StationService`, який виконує всю основну роботу обробки даних та створення сутностей. За своєю суттю він дублює контролер і має такі методи:

- `create` – створює з отриманих даних модель сутності «Станція» та записує її у БД;
- `update` – оновлює в існуючому записі в БД сутності поля, які надходять від контролера;
- `getAll` – запитує всі записи з БД;
- `deleteById` – видаляє запис з БД за ідентифікатором.

Також модуль має підписку на прослуховування подій, що надходять від шини подій `Redis Pub/Sub`, а саме, коли є запущений сценарій польоту літака, він отримує через `Subscriber` інформацію про координати літака у конкретний момент часу та ініціює перерахунок похилої дальності та азимуту від літака. Після процесу перерахунку оновлені параметри відправляються у `Publisher`, який видає їх на сторону клієнта для показу у інтерфейсі користувача.

3.4 Архітектура Backend сервісу «Літак»

Для найбільшого наближення до роботи реального обладнання було вирішено робити відокремлений невеликий сервіс, що відповідає за роботу окремого елемента системи, а саме літак та його бортове обладнання, а також емуляцію проходження сигналів у просторі за допомогою шини подій.

Сервіс представляє собою міні додаток, що побудовано за MVC (Model-View-Controller) архітектурою.

Сутність сценарію літака та її представлення у базі даних, якою оперує мікросервіс «Літак» показана на рисунку 3.6.

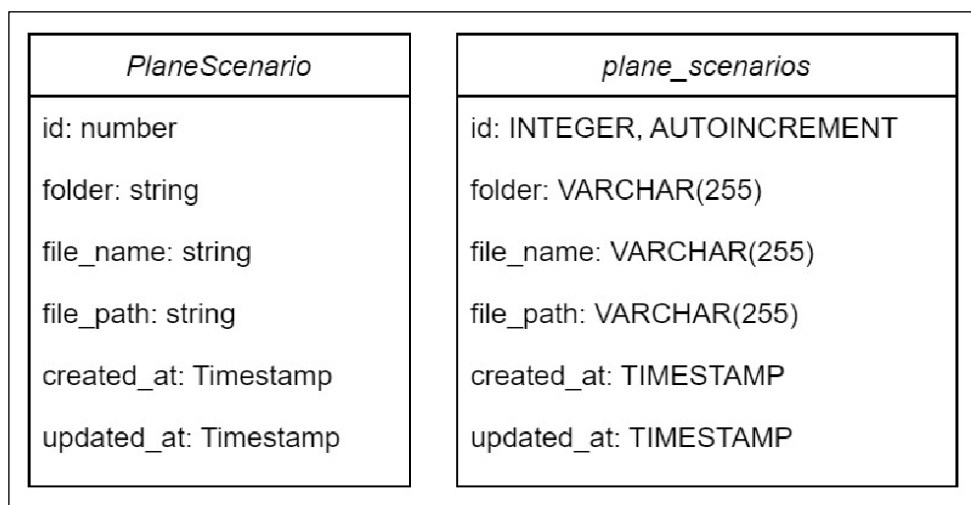


Рисунок 3.6 – Сутність «Сценарій літака» та її представлення у БД

Сутність «Сценарій літака» має такі поля:

- id – унікальний ідентифікатор;
- title – назва станції, наприклад за назвою аеропорту;
- coord_lng – координата широти станції;
- coord_lat – координата довготи станції;
- frequency_hz – робоча частота станції;
- radius_m – радіус дії станції;
- created_at – дата створення станції;
- updated_at – дата оновлення станції.

Архітектура модулю «Літак» показана на рисунку 3.7. Модуль побудовано з використанням MVC архітектури і бібліотеки express.

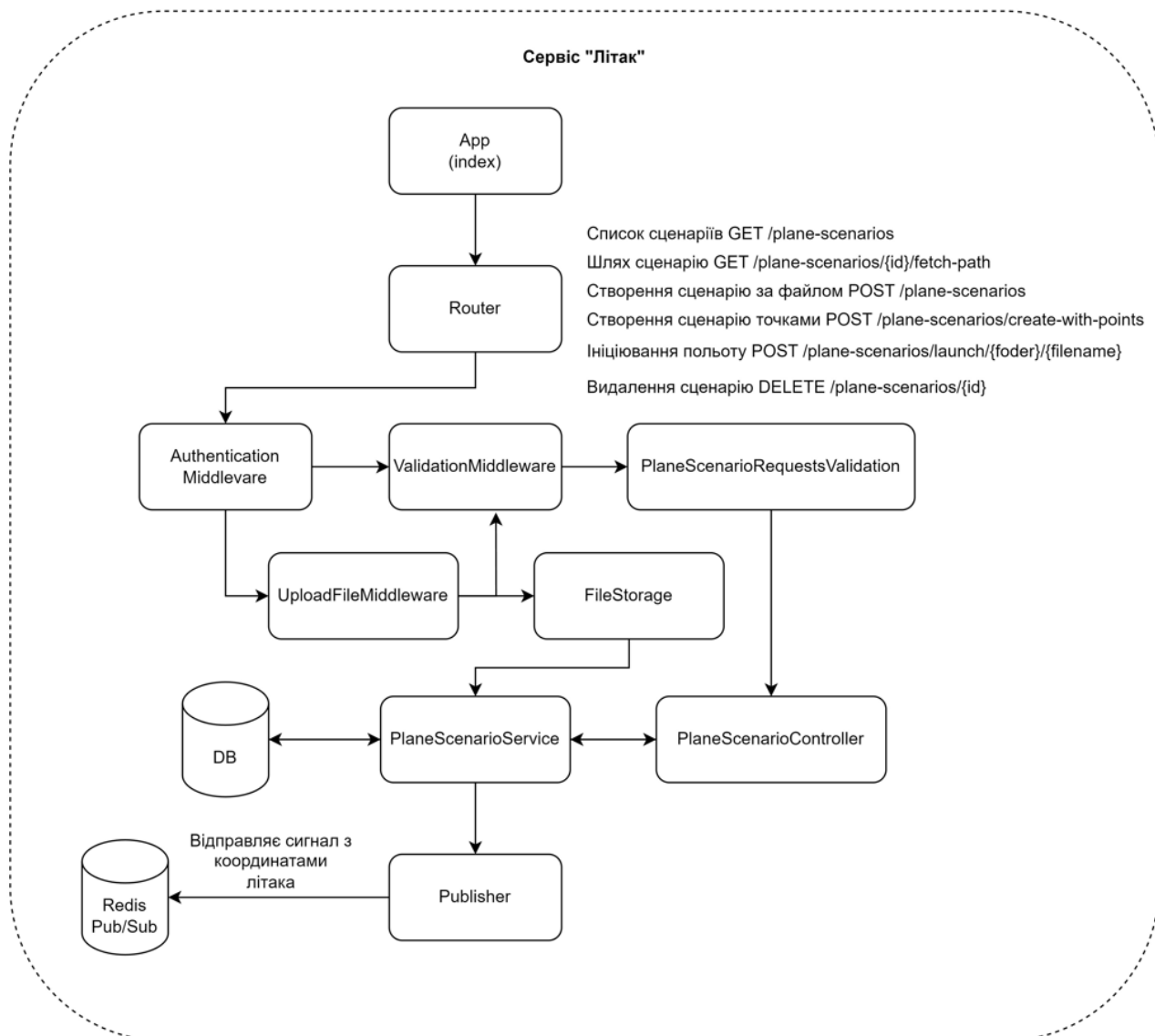


Рисунок 3.7 – Архітектура сервісу «Літак»

Початок модуль бере у файлі App, що є вхідною точкою ініціалізації роботи додатку, в ній запускається у роботу веб сервер та слухач запитів, створюється Router, який слухає запрограмовані «адреси» запиту до веб-серверу. При запиті по кожній «адресі» проводиться перевірка авторизації користувача у файлі AuthenticationMiddleware. Після цього йде перевірка даних, з якими клієнт робить запит, наприклад коректне заповнення полів форми при створенні нової станції у ValidationMiddleware та StationsRequestsValidation. Після успішних перевірок

запит потрапляє у контролер запитів, де проходить подальша обробка та повернення клієнту деякого результату запиту.

Клас `StationsController` має такі методи:

- `createWithFile` – створення нового сценарію шляхом завантаження файлу у форматі `csv` з координатами літака та його параметрами у точці;
- `createWithPoints` – створення сценарію з точок, що користувач відмічає на карті і перетворення набору кутових точок у багато точок у кожній секунді часу;
- `fetchAll` – отримання всього списку станцій;
- `deleteById` – видалення станції за ідентифікатором.

Так як контролер відповідає за отримання запиту та відправку відповіді, між цими двома діями він передає керування сервісу `StationService`, який виконує всю основну роботу обробки даних та створення сутностей. За своєю суттю він дублює контролер і має такі методи:

- `create` – створює з отриманих даних модель сутності «Станція» та записує її у БД;
- `update` – оновлює в існуючому записі в БД сутності поля, які надходять від контролера;
- `getAll` – запитує всі записи з БД;
- `deleteById` – видаляє запис з БД за ідентифікатором.

Також модуль має підписку на прослуховування подій, що надходять від шини подій `Redis Pub/Sub`, а саме, коли є запущений сценарій польоту літака, він отримує через `Subscriber` інформацію про координати літака у конкретний момент часу та ініціює перерахунок похилої дальності та азимуту від літака. Після процесу перерахунку оновлені параметри відправляються у `Publisher`, який видає їх на сторону клієнта для показу у інтерфейсі користувача.

3.5 Архітектура Backend сервісу «Веб-сокет»

Для імітації проходження сигналу у просторі з точки зору програмної реалізації було використано шину подій `Redis Pub/Sub` для комунікації між

сервісами. Для відображення результатів розрахунків та всіх сигналів, що є у додатку створено окремий невеликий сервіс, що передає всі сигнали до клієнтського додатку. Для постійної передачі інформації від серверу до клієнта використовується так звана WebSocket технологія.

WebSocket – це протокол, що призначений для обміну інформацією між браузером та вебсервером в режимі реального часу. Він забезпечує двонаправлений канал зв'язку через один TCP-сокет. Технологія спроектована для втілення у веб-браузерах та веб-серверах, але може використовуватись будь-яким клієнт-серверним застосунком. Цей протокол використовується якщо необхідно відображати інформацію у режимі реального часу.

Архітектура сервісу показана на рисунку 3.8. Цей невеликий сервіс не використовує складних бібліотек і виконує лиш одну функцію прослуховування усіх сигналів, що проходять через шину подій Redis Pub/Sub за допомогою класу Subscriber та відправку їх до клієнтського додатку через веб-сокети за допомогою класу SocketConnection.

Такий підхід дозволяє миттєво відображати усю необхідну інформацію на користувацькому інтерфейсі у процесі проходження літаком сценарію польоту та розрахунків параметрів по станціях.

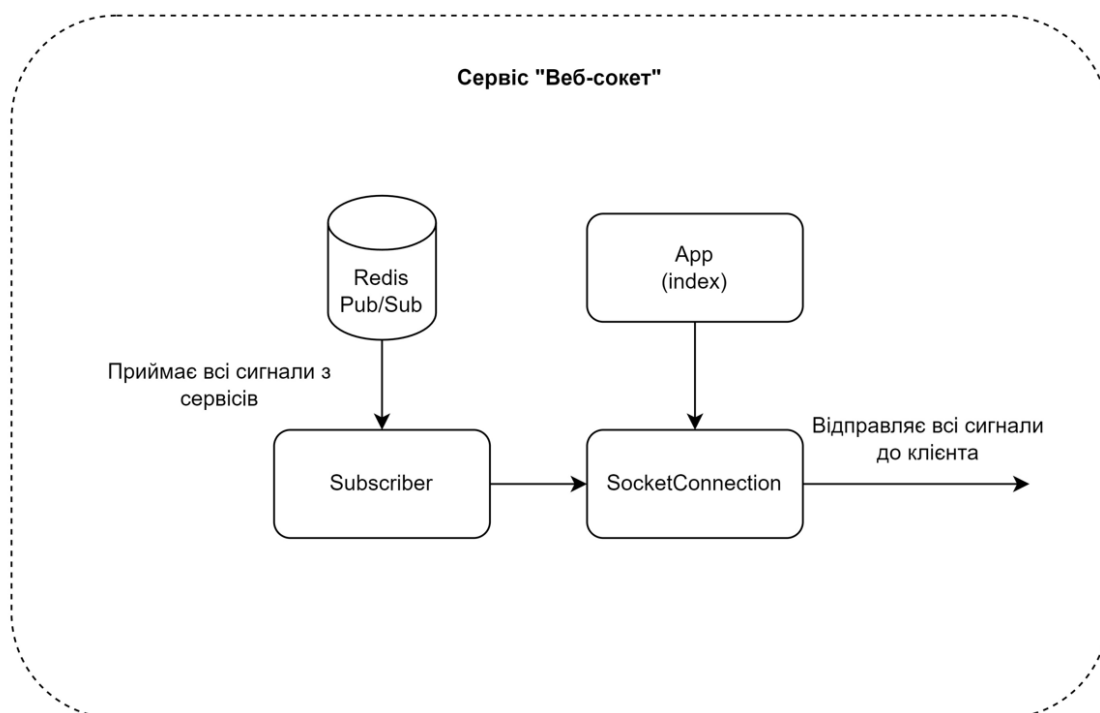


Рисунок 3.8 – Архітектура сервісу «Веб-сокет»

3.6 Архітектура Frontend додатку

Клієнтський додаток представляє собою веб-додаток з графічним інтерфейсом для того, щоб користувач взаємодіяв з Backend сервісами та отримував наочне представлення інформації, що обробляється.

Клієнтський додаток виконується у середовищі веб-браузеру та працює з використанням мови програмування JavaScript з використанням TypeScript для суворої типізації, бібліотеки для побудови складних графічних інтерфейсів React.js та бібліотеки зберігання загального стану даних додатку Redux.

Веб-додаток у плані коду та структури представляє собою моноліт, який поділено на модулі для зручної розробки та розширення функціоналу. Загальна архітектура модулю представлена на рисунку 3.9.

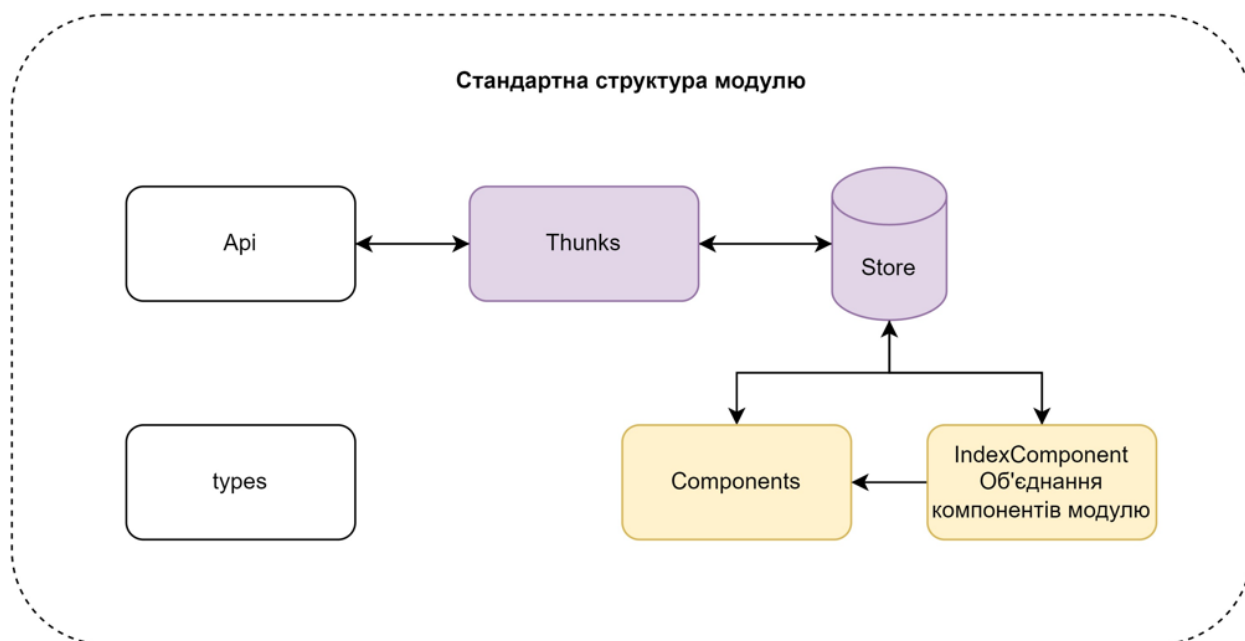


Рисунок 3.9 – Загальна архітектура модулю

Загально, модуль будується навколо сховища даних «Store», в якому зберігаються дані що отримуються від серверу та необхідні для логіки модулю. В тісному зв'язку з ним знаходиться файл «Thunks», який надає необхідні для бібліотеки асинхронні операції, наприклад запити до серверу та обробку відповідей. Файл «Api» надає набір функцій запитів до серверу з компонуванням об'єктів запиту та відповідей. Файл «types» необхідний для опису моделей та

інтерфейсів, що використовуються в модулі і позбавляє від поширених помилок у роботі з об'єктами. Файли, що об'єднані під назвою «Components» - це маленькі частинки користувацького інтерфейсу, що відображають дані, що зберігаються в сховищі та виконують різні запити до нього з діями, які виконує користувач такі як натискання на кнопки, переходи по сторінкам тощо. Файл «IndexComponent» необхідний для єдиної точки експортування компонентів модулю, що можуть використовуватися у інших модулях.

Першим модулем, що представляє ядро додатку є модуль «App». Його архітектуру показано на рисунку 3.10. Він ініціалізує глобальне сховище даних та надає доступи до нього через функції, так звані «хуки». В глобальному сховищі модулю комбінуються усі сховища з інших модулів. В компоненті App об'єднуються усі компоненти з інших модулів та утворюють цілу функціонуючу сторінку додатку.

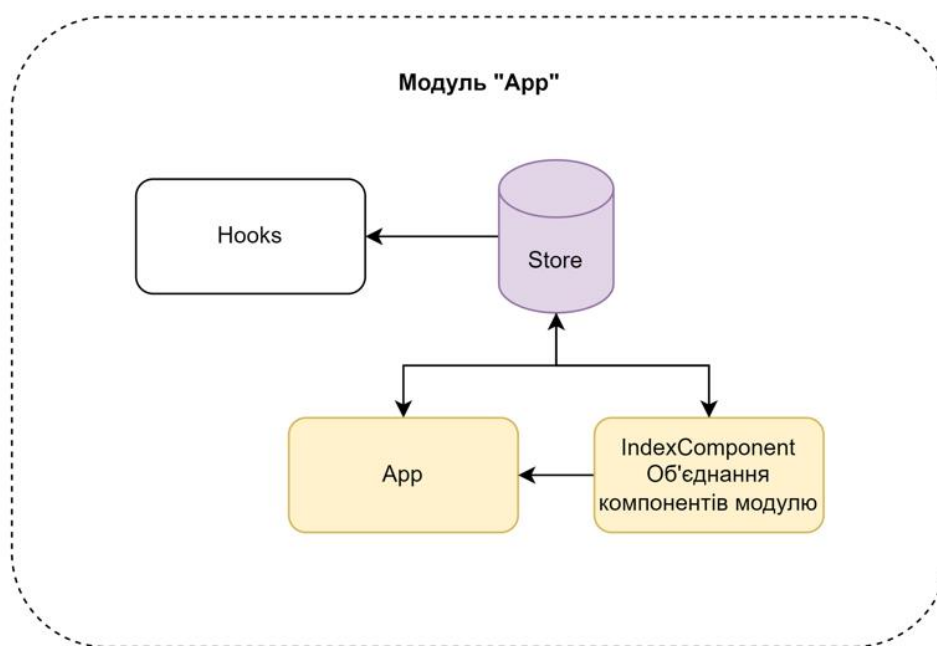


Рисунок 3.10 – Архітектура модулю «App»

Модуль «Controls» відповідає за користувацькі дії у інтерфейсі, активні режими роботи додатку, у режимі розташування станцій, прокладенні сценарію польоту літака або перегляд показників з бортової системи літака. В файлі «Store» зберігається об'єкт з вибраним режимом відображенні інтерфейсу. Компонент

«Controls» відповідає за вибір режиму відображення інтерфейсу. Архітектура модулю показана на рисунку 3.11.

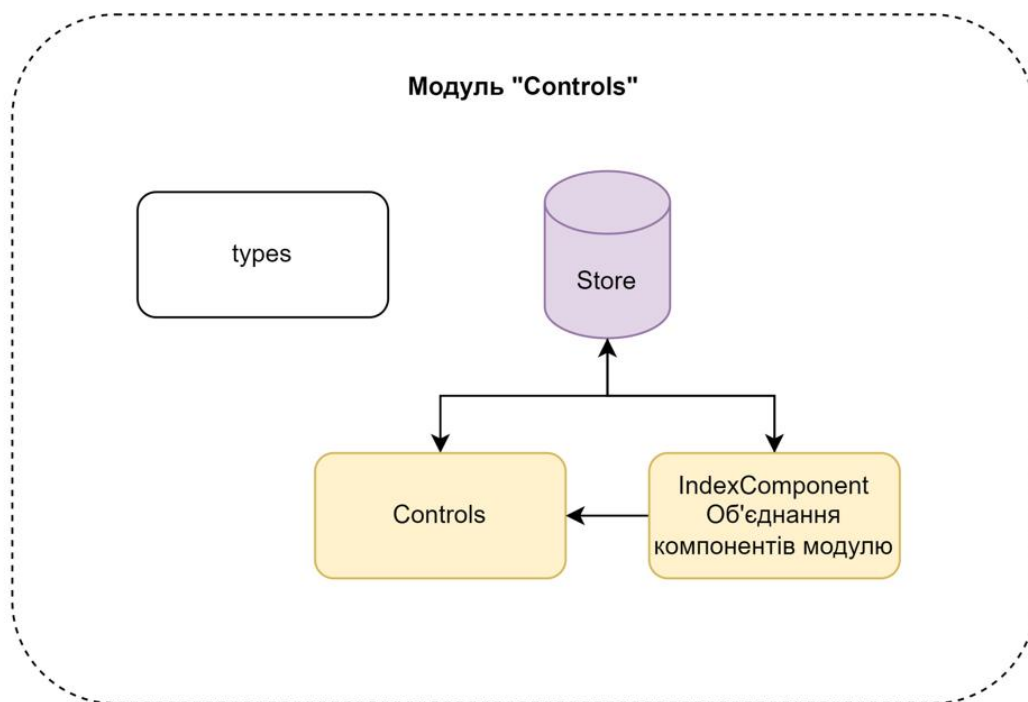


Рисунок 3.11 – Архітектура модулю «Controls»

Модуль «Мар» відповідає за ініціалізацію та базову логіку роботи карти. При завантаженні додатку компонент Мар ініціалізує бібліотеку для роботи з картами та надає їй токен, що є ключем для її роботи адже це стороння карта з деяким безкоштовним обсягом використання. Архітектура модулю показана на рисунку 3.12.

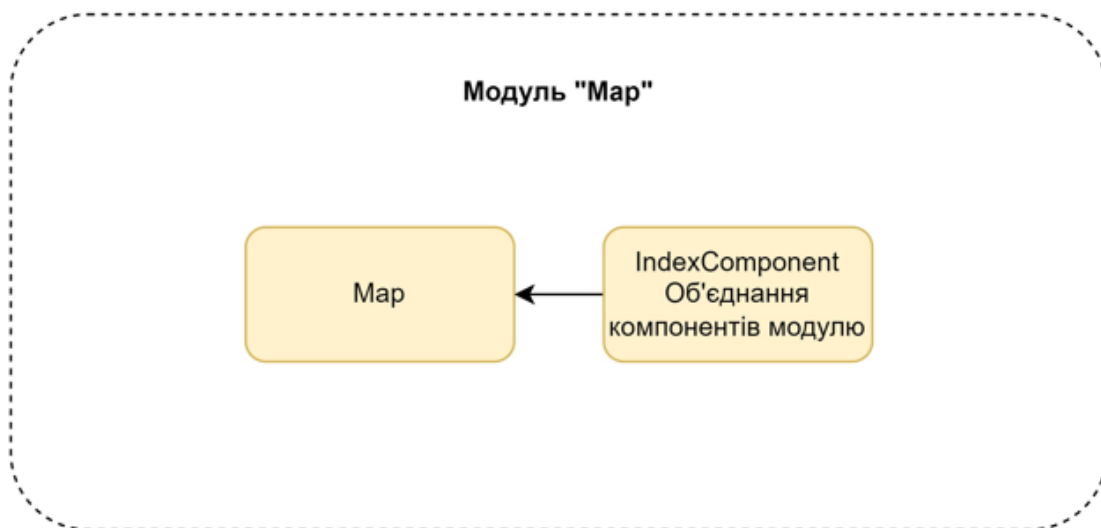


Рисунок 3.12 – Архітектура модулю «Мар»

Модуль «Stations» надає функціонал для роботи зі станціями. Архітектура модулю показана на рисунку 3.13. Після запуску додатку сховище ініціює запит до API для отримання списку існуючих станцій, які вже були створені користувачем. Після отримання відповіді, перелік станцій зберігається у сховищі. Компонент «StationsMapLayer» додає до карти новий шар з відображення стацій та задає стилі їх відображення залежно від параметрів, що вони мають. Коли в сховищі зберігаються станції, вони відображаються на карті.

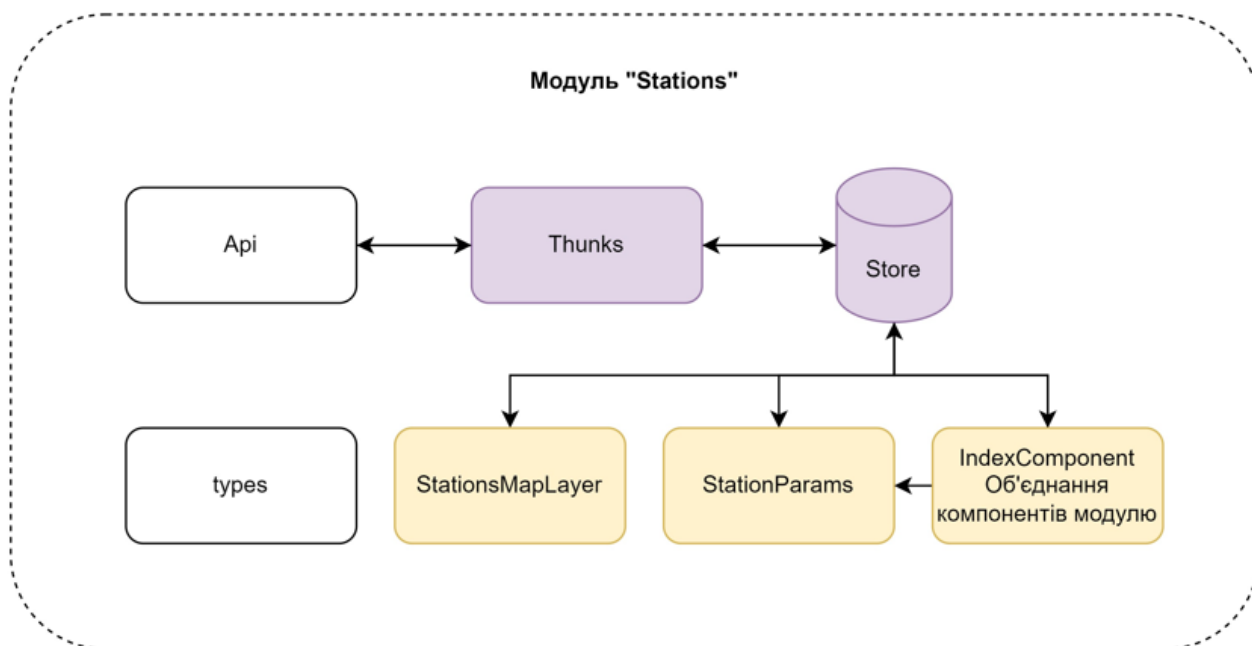


Рисунок 3.13 – Архітектура модулю «Stations»

Компонент «StationParams» надає інтерфейс форми задання параметрів станції, що користувач хоче додати до карти, а саме її частота роботи та радіус, на якому станція активна і може приймати сигнали від бортового запитувача літака.

Останнім модулем є модуль, що надає функціонал роботи з літаком, а саме із сценаріями польоту літака. Користувач може створювати нові сценарії та програвати вже існуючі.

Після запуску додатку, сховище ініціює запит до API для отримання списку існуючих сценаріїв польоту літака. Після їх отримання, вони відображаються у компоненті «PlaneScenarioSelector», що надає список, що випадає, для вибору сценарію для програвання, а також кнопку для його запуску. Архітектура модулю показана на рисунку 3.14.

Сховище модулю зберігає багато даних, що відносяться до модулю. Наприклад, під час створення нового сценарію польоту, у сховищі зберігаються кутові точки, що користувач ставить на карті. Відображення та взаємодія з картою під час створення нового сценарію польоту забезпечується у компоненті «PlaneScenarioPointMapLayer», що додає новий шар та логіку до карти.

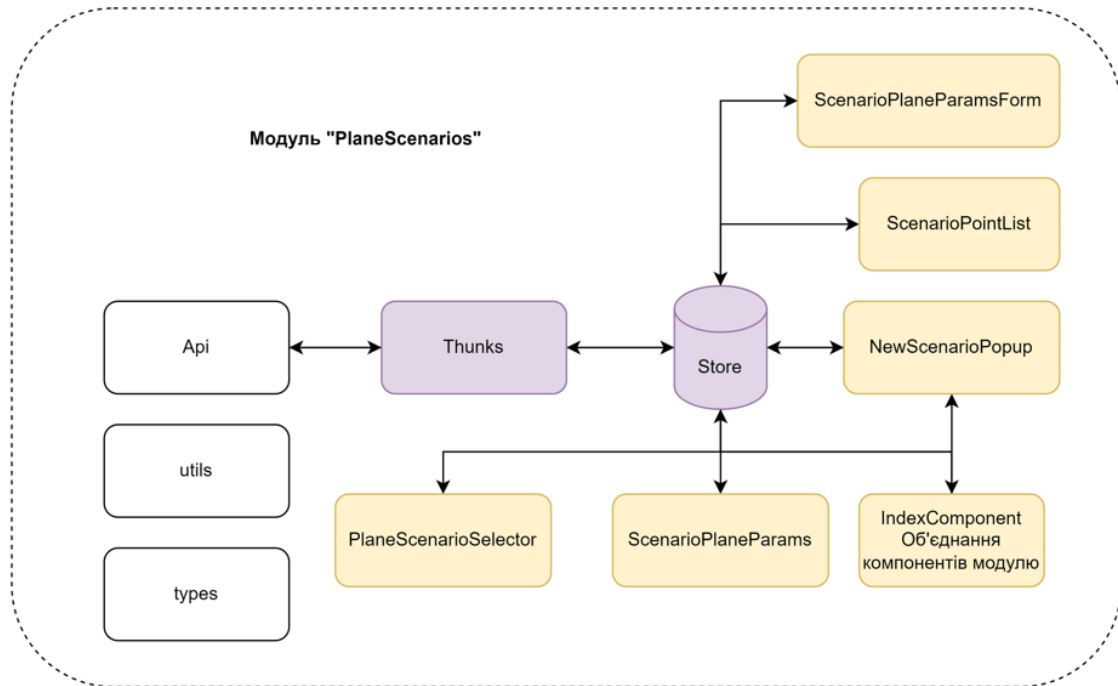


Рисунок 3.14 – Архітектура модулю «PlaneScenarios»

Компонент «ScenarioPointList» відображає точки та їх параметри під час створення маршруту. Кожна точка відображає такі параметри як

- координати літака (lat та lng);
- висоту літака (altitude);
- курс літака (bearing);
- швидкість літака (speed);

Компонент «ScenarioPlaneParams» представляє собою форму, де користувач вводить параметри висоти та швидкості, що будуть записані у наступну точку.

3.7 Взаємодія клієнтської частини додатку з сервером

Взаємодія між клієнтською частиною системи та сервером відбувається завдяки використанню HTTP запитів. Сервіс, що запущено на сервері надає

доступ у вигляді публічного API з набором методів для взаємодії з ним. В свою чергу сервіс підключено до бази даних і сервіс взаємодіє з нею. Діаграма послідовності взаємодії з сервісом «Станція» показано на рисунку 3.15. Таким чином надається доступ до такого функціоналу сервісу «Станція» як:

- створення станції;
- редагування параметрів станції;
- запит переліку станцій;
- видалення станції.



Рисунок 3.15 – Діаграма послідовності взаємодії з сервісом «Станція»

За схожою схемою відбувається взаємодія з сервісом «Літак», що зображено на рисунку 3.16. Цей сервіс використовує базу даних для зберігання записів про

наявні сценарії, а також має внутрішнє сховище, де зберігаються файли сценаріїв польоту. Таким чином сервіс надає доступ до такої функціональності як:

- створення нового сценарію за допомогою файлу;
- створення нового сценарію за допомогою точок;
- запит переліку сценаріїв;
- видалення сценарію;
- запуск сценарію польоту.

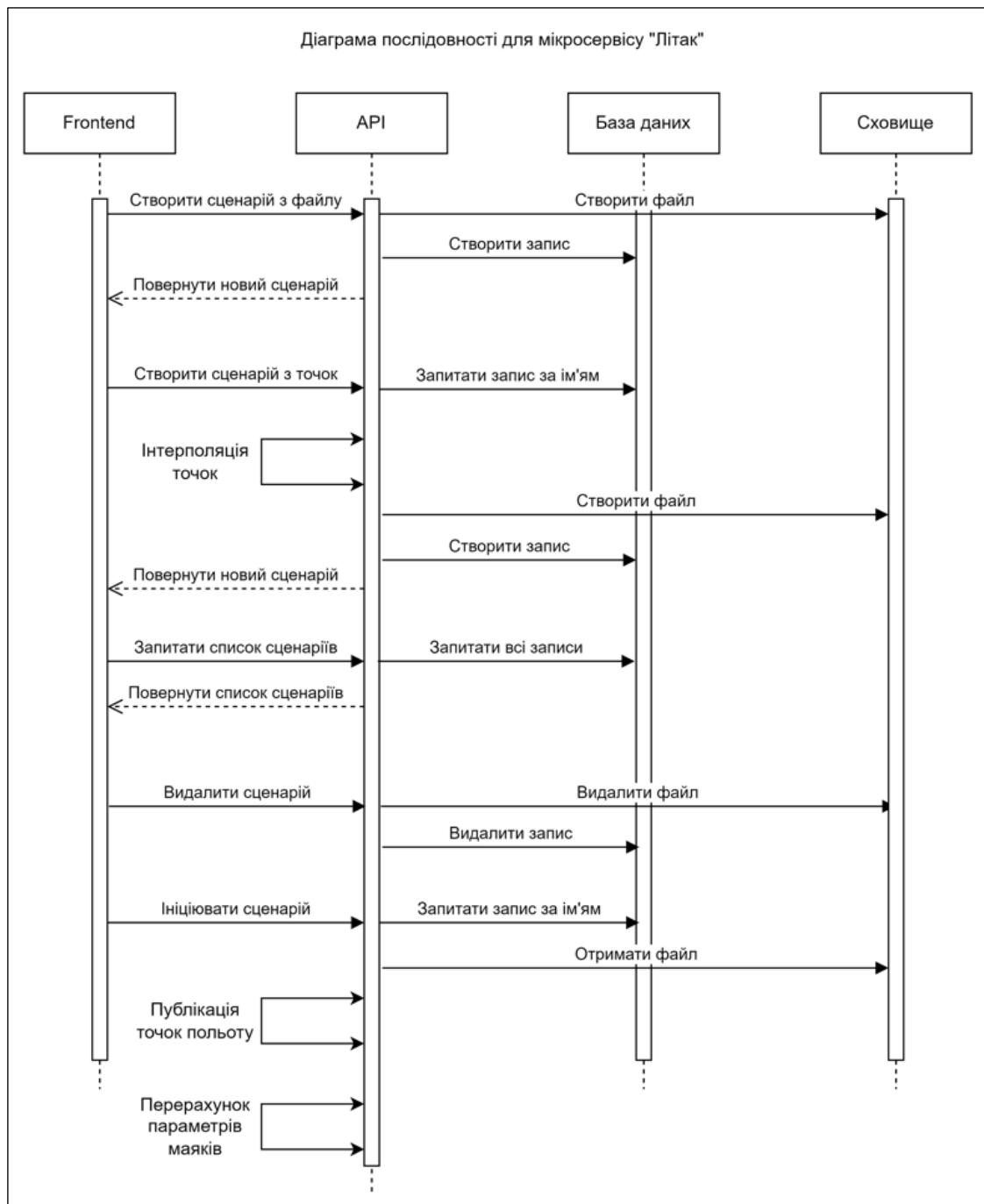


Рисунок 3.16 – Діаграма послідовності взаємодії з сервісом «Літак»

3.8 Опис можливостей клієнтської частини

У загальному вигляді інтерфейс програмного забезпечення клієнтської частини веб-додатку показаний на рисунку 3.17 і представляє карту реальної місцевості з цільовими модулями візуалізації та елементами керування.

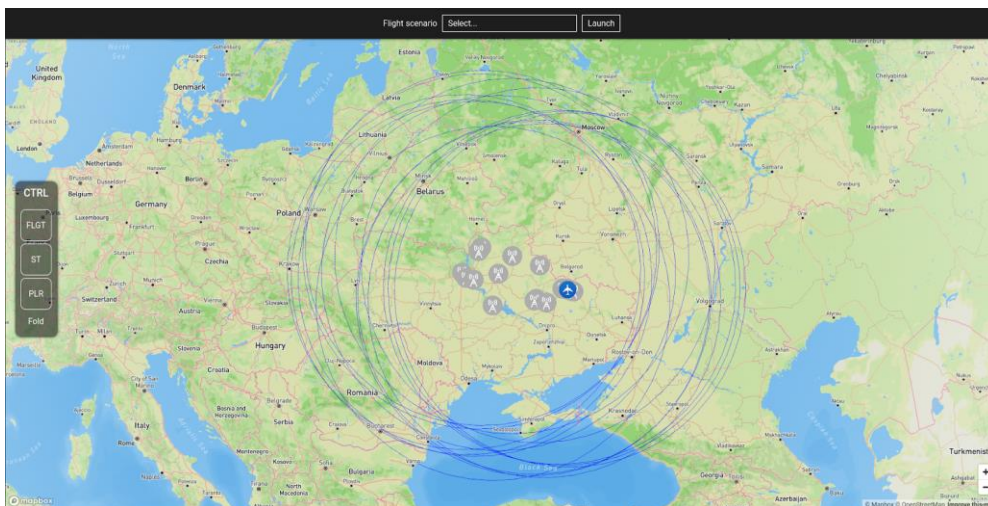


Рисунок 3.17 – Інтерфейс програмного забезпечення

За допомогою інтерфейсу користувача можна розставляти маяки системи TACAN в будь-яких точках карти, де це необхідно і задавати або редагувати їх параметри. Стандартні станції відображаються у вигляді сірих іконок “не активних” станцій з показом їх радіусу видимості у вигляді кола.

Також інтерфейс дозволяє створювати маршрути польоту літака (сценарії), як на рисунку 3.18.

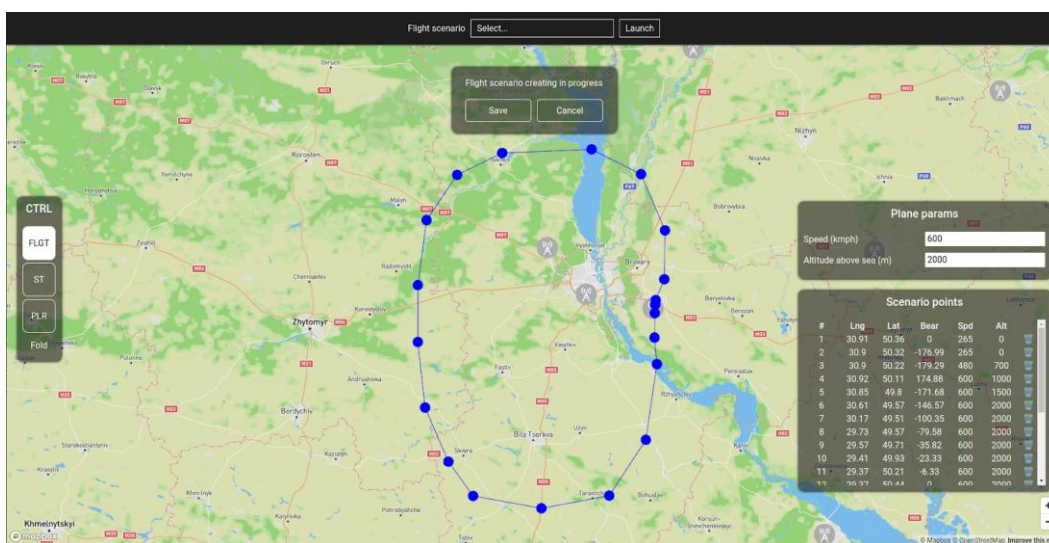


Рисунок 3.18 – Створення маршруту польоту літака

Створення маршруту польоту літака здійснюється простою розстановкою кутових точок маршруту, для яких у спеціальній формі Plane params вказуються миттєві для цієї точки параметри висоти та швидкості літака. У таблиці “Scenario Points” відображається інформація по кутових точках маршруту, яка включає координати точки, напрямок між цією і попередньою точкою, швидкість і висота літака в цій точці.

Перед збереженням сценарію польоту, для нього вказується назва для подальшого виведення в селекторі сценаріїв польоту.

Під час збереження маршруту відбувається процес інтерполяції між точками, в якому здійснюється розрахунок точок положення літака кожної секунди і параметрів літака в цих точках. Зрештою, вся інформація зберігається в csv файл, який за необхідності можна імпортувати в будь-яке інше програмне забезпечення або селектор сценаріїв польоту для іншого користувача. В csv файлі зберігаються точки кожної секунди польоту літака, що вміщують такі поля з інформацією як counter, lng, lat, bearing, speed, altitude.

За допомогою селектора заздалегідь створених сценаріїв відбувається запуск необхідного сценарію польоту, при якому створюється унікальний потік сценарію, відкривається канал Redis PubSub який кожну секунду відправляє місцезнаходження літака з подальшим відображенням його маркера на карті, як на рисунку 3.19.

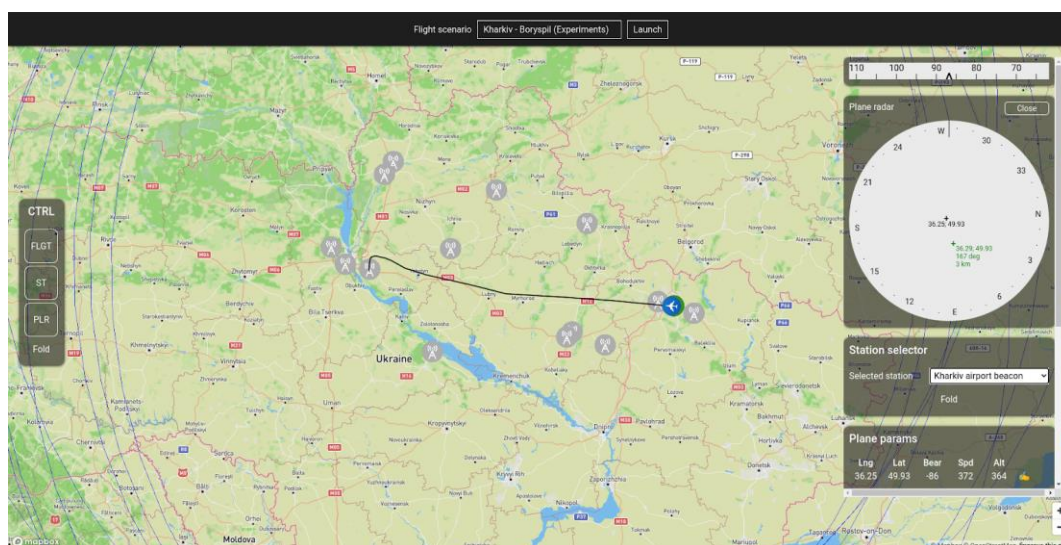


Рисунок 3.19 – Відображення увімкненого сценарію польоту

У процесі обробки кожної точки сценарію відбувається емуляція роботи бортової системи літака як на рисунку 3.20, спрямованої на розрахунок параметрів маяків у зоні видимості, таких як: похила дальність (км); напрямлення на маяк (градуси).

Програмно, кожен радар має змінний радіус дії для того, щоб проводити експерименти з маяками з різними радіусами видимості. Також, радару літака програмно можна змінювати радіус для маяків, що відображаються для зручності відображення і кращої читаності їх в інтерфейсі.

Інтерфейс користувача виконується в середовищі веб браузера і побудований з використанням TypeScript і бібліотеки React, яка служить для зручної побудови складних інтерфейсів користувача. Як картка використовується бібліотека Mapbox, яка використовує дані з проекту Open Street Map для відображення карток.

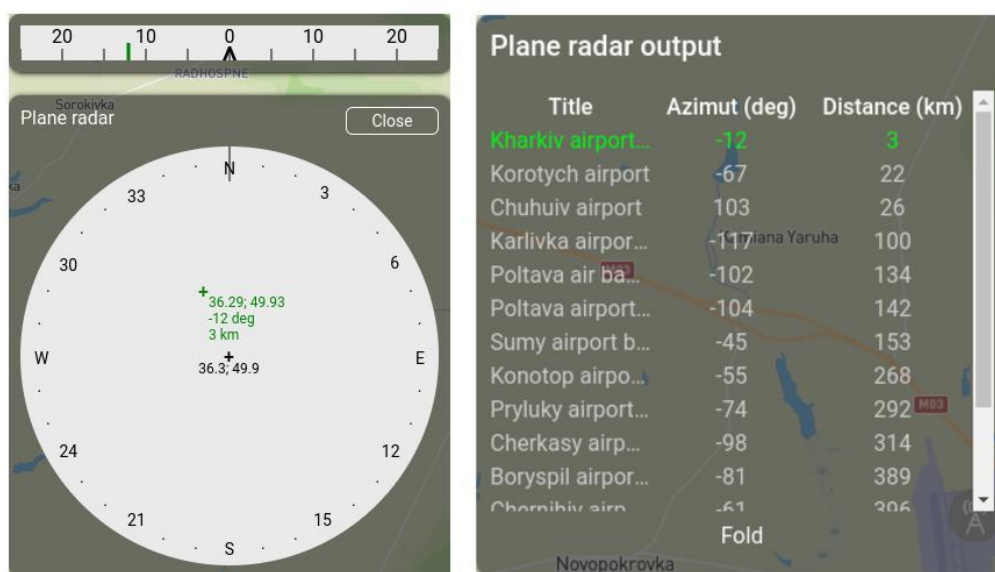


Рисунок 3.20 – Радар літака та маяки з відображенням дальності та напрямку

Особливий вид сценаріїв польоту – це сценарії, які з однієї точки. Такий вид сценаріїв необхідний тестування поведінки радара при повністю контрольованих параметрах літака. Після запуску такого сценарію та появи літака з'являється форма "Plane params", показана на рисунку 3.21, в якій можна редагувати всі необхідні параметри літака, такі як:

- положення літака на карті, координати (lng, lat);
- напрямлення літака (градуси);
- швидкість літака (км/ч);
- висота літака (м);

Разом із ручним селектором маяків TACAN у секції “Station selector” такі сценарії дозволяють проводити тестування коректної роботи всіх елементів системи.

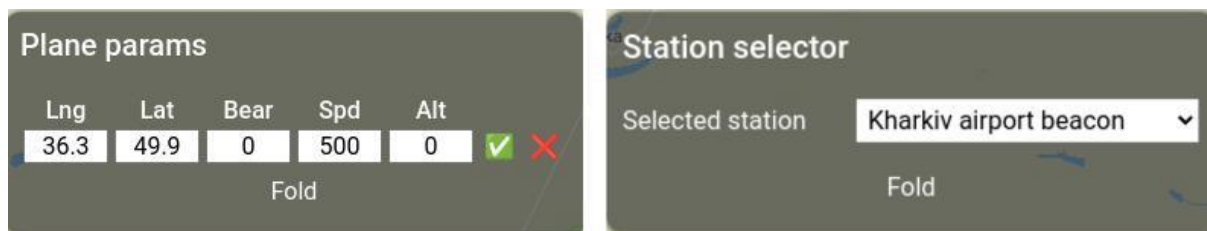


Рисунок 3.21 – Форма зміни параметрів літака та вибору маяків TACAN

4 ЕКСПЕРИМЕНТАЛЬНИЙ РОЗДІЛ

4.1 Підготовка експерименту

Основний спосіб проведення експериментів з симулятором - запуск сценаріїв польоту літака, при якому можна бачити показники, що змінюються в часі, з програмно-емульованої бортової системи літака. З оцінкою результату точності із застосуванням процедур експертного аналізу. За часом оцінка не проводиться, оскільки симулятор працює у часі.

Планується проведення кількох експериментів для демонстрації трьох режимів польоту:

- політ літака через по прямій між двома точками;
- політ з Харкова до Борисполя з використанням великої кількості точок маршруту, що наближено до реального польоту;
- режим з однією точкою для тонкого налаштування.

Перед проведенням експерименту необхідно провести два підготовчих етапи:

- розміщення маяків на карті;
- створення сценарію польоту літака.

Розставлення маяків здійснюється з меню управління “CTRL” активацією кнопки “ST” після чого необхідно розставити маяки на карті за допомогою миші, задаючи параметри маякам такі як Distance та Frequency. Результат розміщення маяків TACAN показаний рисунку 4.1.

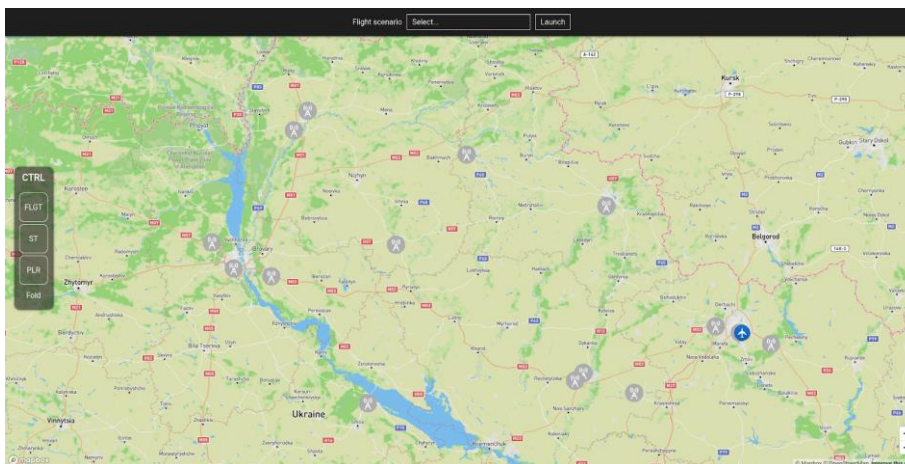


Рисунок 4.1 – Відображення маяків на карті

Для вже створених маяків можна змінювати параметри їх Distance та Frequency за допомогою форми з рисунку 4.2, в якій відображено параметри за замовчуванням.

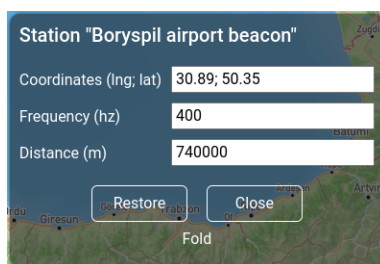


Рисунок 4.2 – Редагування параметрів маяка

Перехід у створення маршруту польоту здійснюється з меню CTRL активацією кнопки FLGT після чого необхідно на карті за допомогою миші вказати кутові точки польоту літака. Кутовими точками називаються ті точки, де відбувається зміна будь-якого параметра польоту, наприклад, зміна курсу, зміна висоти та швидкості. Доступні кілька варіантів створення маршруту:

- набір точок;
- дві точки;
- одна точка;

На рисунку 4.3 відображено результат створення маршруту з багатьох точок. Внизу у розділі “Scenario points” відображається список точок з миттєвими параметрами літака у цій точці, а “Plane params” змінюються параметри літака для наступної точки.

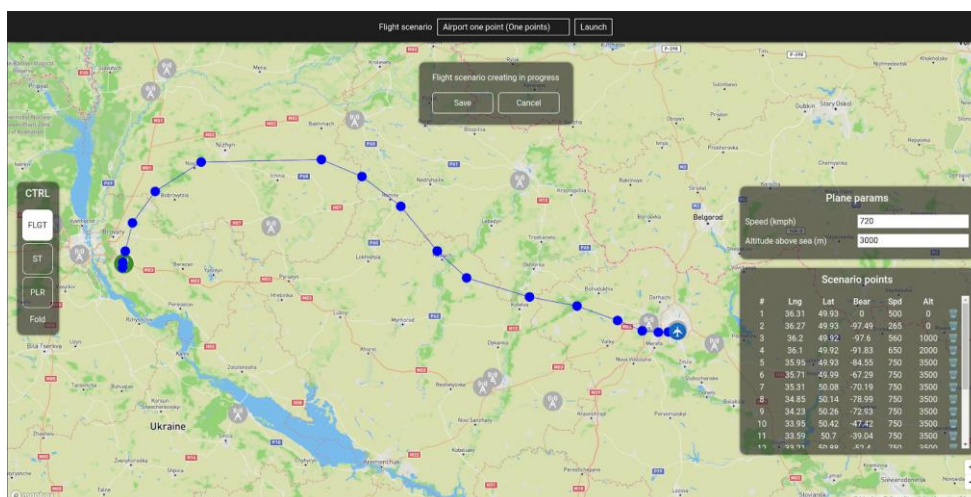


Рисунок 4.3 – Створення сценарію польоту з багатьма точками

Також був створений сценарій польоту, який складається з двох точок, з польотом від Чугуєва до Борисполя, який показано на рисунку 4.4 та сценарій з однією точкою, який використовуватиметься для тонких тестів.

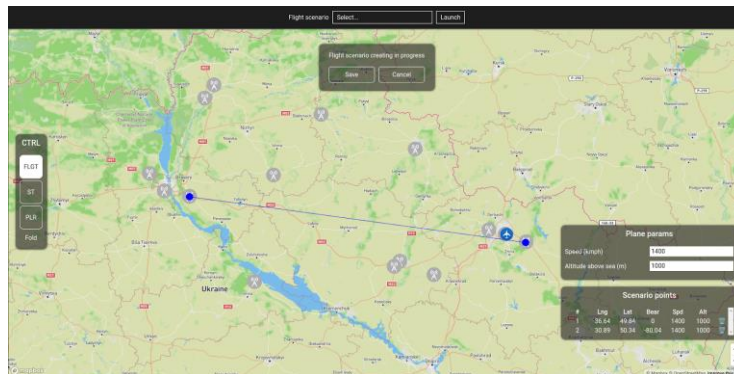


Рисунок 4.4 – Створення сценарію польоту з двома точками

4.2 Проведення експериментів

Відповідно до плану, були проведені експерименти із запуском сценаріїв польоту, зняттям параметрів у різні моменти часу та оцінюємо помилки групою експертів.

Створений сценарій польоту між двома точками був інтерпольований у файл із 1065 точками, тобто весь сценарій займає 17 хвилин та 45 секунд при швидкості польоту 1400 км/год. Інтерфейс приймає зовнішній вигляд, показаний на рисунку 4.5, де видно відображення маршруту польоту літака, станцію, з якою взаємодіє літак у вигляді маркера зеленого кольору.

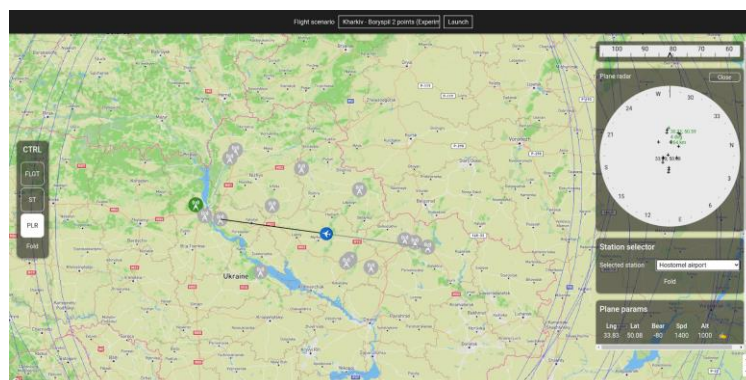


Рисунок 4.5 – Політ літака між двома точками з відображенням параметрів

На рисунку 4.6 показаний радар літака, на якому відображаються маяки в зоні видимості радару бортової системи літака, так і маяків. Маркер обраного для

взаємодії маяка забарвлюється в зелений колір і поруч з ним відображається координат маяка, напрямок на маяк і похила дальність до нього.

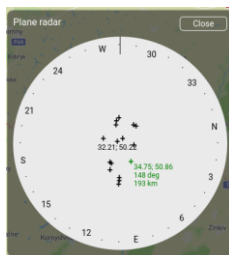


Рисунок 4.6 – Радар літака з відображенням маяків

У таблиці 4.1 показані зняті параметри в різні моменти проходження сценарію з двома точками з бортової системи літака, де показані параметри напрямку і похилої дальності по відношенню до маяка "Sumy airport beacon", який розташований на координатах 34.75,50.86.

Таблиця 4.1 – Значення з бортової системи літака у сценарії з двома точками

Положення літака (lat,lng)	Напрямок літака (градуси)	Дальність до маяка (км)	Напрямок на маяк (градуси)	Висота (м)
36.34,49.86	-80 (280)	158	35	1000
36.12,49.88	-80 (280)	146	39	1000
35.88,49.90	-80 (280)	133	43	1000
35.52,49.94	-80 (280)	116	52	1000
34.88,50.00	-80 (280)	96	77	1000
34.18,50.05	-80 (280)	98	104	1000
33.59,50.10	-80 (280)	117	124	1000
33.19,50.14	-80 (280)	137	134	1000
32.21,50.22	-80 (280)	193	148	1000
31.77,50.26	-80 (280)	221	151	1000

У таблиці 4.2 показані зняті параметри в різні моменти проходження сценарію з безліччю точок з бортової системи літака, де показані параметри

напрямку і похилої дальності по відношенню до маяка Voryspil airport beacon, який розташований на координатах 30.89,50.35.

Таблиця 4.2 – Значення з бортової системи літака в сценарії з багатьма точками

Положення літака (lat,lng)	Напрямок літака (градуси)	Дальність до маяка (км)	Напрямок на маяк (градуси)	Висота (м)
36.15,49.93	-86 (274)	378	5	1727
35.41,49.98	-84 (276)	324	3	3570
34.80,50.01	-86 (274)	281	5	4822
33.24,50.08	-76 (284)	170	-3 (357)	6000
32.48,50.20	-77 (283)	114	-4 (356)	6000
31.71,50.32	-72 (288)	59	-14 (346)	5521
31.38,50.41	-61 (299)	35	-40 (320)	3785
31.08,50.49	-79 (281)	20	-61 (299)	2000
30.92,50.48	-126 (234)	15	-47 (213)	2000
30.90,50.45	-171 (189)	12	-7 (353)	1688

Також було проведено експерименти зі сценарієм, що складається з однієї точки. У такому вигляді сценаріїв літак з'являється у певній точці і залишається нерухомим. Основними елементами управління є форма налаштування параметрів літака, де задаються координати, напрямок та висота літака, а також селектор маяків для перемикання як на рисунку 4.7. Загалом, результати експерименту подібні до двох попередніх.

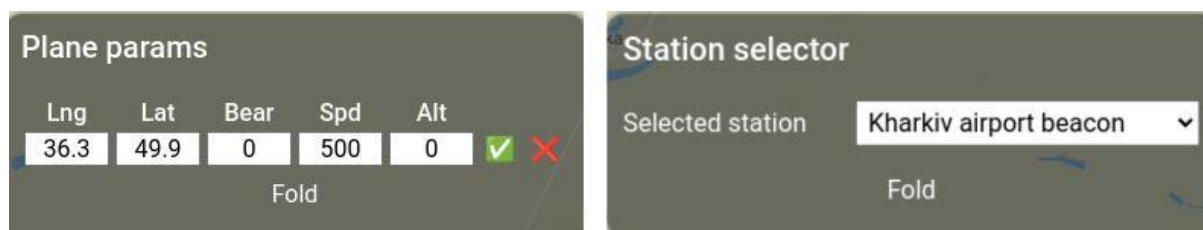


Рисунок 4.7 – Форма параметрів літака та вибір маяків

5 ОБРОБКА РЕЗУЛЬТАТІВ

Основне призначення програмного забезпечення – це проведення комп'ютерного моделювання взаємодії бортової системи літака та маяків РСБН засобами ЕОМ, що дозволить скоротити витрати та позбавить необхідності проведення натурних експериментів з реальним дорогим обладнанням, яке навіть не завжди можна отримати. Також, за допомогою симулятора можна демонструвати принципи роботи РСБН для майбутніх або діючих пілотів та обслуговуючого персоналу станцій.

Проведена робота над розробкою та проведення експериментів показали, що комп'ютерний симулятор дозволяє швидко та ефективно проводити експерименти для дослідження роботи всіх компонентів системи РСБН TACAN у загальній зв'язці, при цьому маючи можливість контролювати окремі параметри цих компонентів як літаків, так і маяків у будь-який момент часу.

Зручний і простий інтерфейс, а також широкі можливості програмного забезпечення дозволяють швидко створювати сценарії польотів і проводити розміщення маяків на карті, змінювати параметри за потреби, здійснювати моделювання в реальному часі на реальній карті місцевості.

Підтримка кількох видів сценаріїв дозволяє моделювати як політ літака, наближений до реальності за допомогою тонкого налаштування параметрів літака таких як швидкість і висота і прокладання максимально точного маршруту, так і більш експериментальні, як політ між двома точками або зовсім помістити літак у довільну крапку і змінювати його параметри, тестуючи коректність результатів обробки бортової системі літака. Проводити навчання для співробітників, які можуть взаємодіяти з програмним забезпеченням не складе труднощів.

Оцінка результатів експериментів (симуляції) експертами із застосуванням процедури паралельного оцінювання азимуту та похилої дальності показали, що симулятор видає оцінки параметрів у реальному часі з похибкою, яка не є статистично значущою. Усього було проведено понад 100 симуляцій різних варіантів польоту та взаємодії літака та маяка.

Для такого оцінювання залучалася група із 7 експертів у предметній області. Експерти однакової кваліфікації індивідуально і незалежно один від одного запускали сценарії польоту, проводили вимірювання параметрів та порівняльний аналіз із оцінками параметрів симулятора, що спостерігаються. Це вироблялося з урахуванням використання типових таблиць (типу Таблиця 1, Таблиця 2). Потім проводилося усереднення оцінок параметрів, виявлення викидів за критерієм Граббса, реалізувалась процедура голосування. Більш ніж у 99% випадків результати оцінювання та голосування експертів збіглися. Лише одному випадку було виявлено розбіжність; воно було викликане неточністю введення даних. При повторній перевірці його було скориговано. В результаті, з допустимою в інженерних розрахунках похибкою, всі оцінки експертів збіглися з оцінками симулятора.

Комплексна оцінка результатів експериментів показала як адекватність оцінок параметрів азимуту і похилої дальності, але й високу ефективність застосування симулятора загалом. Включаючи можливості побудови сценаріїв та модулі візуалізації.

Запропонований симулятор може використовуватися на практиці (в системах онлайн та офлайн навчання) для побудови сценаріїв польоту літака та його взаємодії з маяками на карті місцевості в реальному часі. При його використанні доцільно створювати та використовувати сценарії польоту у прив'язці до місцевості, а не окремі маршрути.

Подальший розвиток проекту залежить від попиту на технологію, в предметній області якої працює симулятор і включає розгортання системи у виробничому середовищі як MVP для початку реального тестування користувачами, які будуть використовувати програмне забезпечення у професійній діяльності. За допомогою збору відгуків щодо роботи системи та побажань на додаткову функціональність може бути сформований подальший список функціональних елементів, які можна додати до симулятора.

На даний момент можна запропонувати таку функціональність для розробки як:

- підтримка декількох літаків у сценарії;
- створення та зберігання окремих наборів розташування маяків;
- інтегрування нових елементів інтерфейсу, моделюючих робоче місце оператора землі, і навіть деякі нові елементи приладової дошки літака.

Окремо можна відзначити таку можливість для розвитку (у тісній взаємодії з експертами в авіаційній сфері), як реалізацію підтримки стрімінгу сигналів із сенсорів реального обладнання, які визначають положення літака, висоту, швидкість, курс та інші параметри для подальшої їх обробки всередині симулятора.

ВИСНОВКИ

У роботі проведено аналіз особливостей реалізації сучасних комп'ютерних симуляторів, включаючи алгоритми та технології роботи, функціональні можливості, реалізації інтерфейсу та візуалізації. На цій основі сформульовано вимоги та розроблено специфікацію симулятора РСБН TACAN.

Розглянуто основні принципи організації та функціонування радіотехнічної системи ближньої навігації TACAN, далекомірною та азимутального каналів радіомаяка та літака.

Описано специфікацію запропонованої реалізації програмного забезпечення симулятора РСБН TACAN, яке моделює роботу компонентів РСБН, таких як бортова система літака та маяки в реальному часі. Наведена архітектура та опис компонентів дає коротке пояснення принципів роботи програмного забезпечення та використовуваних технологій. Описано етапи планування та проведення експериментів на основі використання сценаріїв, де були показані основні можливості роботи з симулятором рядового користувача в процесі створення шаблонів польоту та розстановки маяків. Проведення експериментів супроводжувалося описом можливостей системи виводити інформацію з сенсорів бортової системи літака, а самі вимірювання були проведені з достатньою точністю, що є основним фактором ефективності симулятора як програмного забезпечення, що забезпечує емуляцію компонентів системи РСБН.

Для оцінки результатів симуляції було залучено групу експертів, яка паралельно оцінювала азимут і похилу дальність, порівнювала їх з оцінками симулятора, оцінювала інтерфейс, можливості та візуальне подання результатів симулятором. Аналіз роботи експертів показав, що симулятор видає оцінки параметрів у реальному часі з похибкою, яка є статистично значущою.

Симулятор може бути використаний на практиці (в системах онлайн та офлайн навчання) для побудови сценаріїв польоту літака та його взаємодії з маяками на карті місцевості в реальному часі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. S. Maglio, C. Park, S. Tognarelli, A. Menciassi and E. T. Roche, "High-Fidelity Physical Organ Simulators: From Artificial to Bio-Hybrid Solutions," in *IEEE Transactions on Medical Robotics and Bionics*, vol. 3, no. 2, pp. 349-361, May 2021, doi: 10.1109/TMRB.2021.3063808.
2. R. M. Vigliani et al., "Augmented Reality to Improve Surgical Simulation: Lessons Learned Towards the Design of a Hybrid Laparoscopic Simulator for Cholecystectomy," in *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 7, pp. 2091-2104, July 2019, doi: 10.1109/TBME.2018.2883816.
3. D. Jung, "A Minimally Configured Hardware-In-the-Loop Simulator of Electrical Power Steering System for Human Driver Interaction on Crosswind Effect," in *IEEE Access*, vol. 9, pp. 60470-60481, 2021, doi:10.1109/ACCESS.2021.3073989.
4. Y. Yan, I. Bajaj, R. Rabiee and W. P. Tay, "A Tightly Coupled Integration Approach for Cooperative Positioning Enhancement in DSRC Vehicular Networks," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 23278-23294, Dec. 2022, doi: 10.1109/TITS.2022.3208257.
5. P. Skryja, P. Barcik, Z. Kolka and O. Wilfert, "Autoaligning System for Short-Range Free Space Optics Links," 2022 32nd International Conference Radioelektronika (RADIOELEKTRONIKA), Kosice, Slovakia, 2022, pp. 1-5, doi: 10.1109/RADIOELEKTRONIKA54537.2022.9764918.
6. P. Schoeder, B. Schweizer, A. Grathwohl and C. Waldschmidt, "Multitarget Simulator for Automotive Radar Sensors With Unknown Chirp-Sequence Modulation," in *IEEE Microwave and Wireless Components Letters*, vol. 31, no. 9, pp. 1086-1089, Sept. 2021, doi: 10.1109/LMWC.2021.3088882.
7. L. Ferrigno et al., "Magnetic Localization System for Short-Range Positioning: A Ready-to-Use Design Tool," in *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-9, 2021, Art no. 6002309, doi: 10.1109/TIM.2020.3035397.
8. M. Fazio, A. Buzachis, A. Galletta, A. Celesti and M. Villari, "A proximity-based indoor navigation system tackling the COVID-19 social distancing measures,"

2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 2020, pp. 1-6, doi: 10.1109/ISCC50000.2020.9219634.

9. J. Zhang and Y. Zhang, "WAVE-Based Short-Range Vehicular Localization Through Vector Match," in *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 592-600, Jan. 2020, doi: 10.1109/TIE.2019.2893828.

10. H. Zhao, G. Liu, D. Liu, D. Cheng and X. Shi, "DME test equipment receiving function metrological method and device design," *CSAA/IET International Conference on Aircraft Utility Systems (AUS 2022)*, Nanchang, China, 2022, pp. 295-299, doi: 10.1049/icp.2022.1542.

11. D. W. Matolak et al., "Novel Filterbank Multicarrier Waveform for L-band Digital Aeronautical Communications: Initial Field Test Results," *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, San Antonio, TX, USA, 2021, pp. 1-10, doi: 10.1109/DASC52595.2021.9594295.

12. M. Sato, "Robust Gain-Scheduled Flight Controller for an In-Flight Simulator," in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 3, pp. 2122-2135, June 2020, doi: 10.1109/TAES.2019.2943770.

13. A. Kaviyarasu, A. Saravanakumar and M. L. Venkatesh, "Hardware in Loop Simulation of a Way Point Navigation Using Matlab/Simulink and X-Plane Simulator," *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, Palladam, India, 2019, pp. 332-335, doi: 10.1109/ISS1.2019.8908103.

14. Y. Wang, J. Zhang, P. Wang and L. Zhang, "Dynamic Flight Simulator for Low-Altitude Planes," *2019 International Conference on Virtual Reality and Visualization (ICVRV)*, Hong Kong, China, 2019, pp. 277-278, doi: 10.1109/ICVRV47840.2019.00068.

15. H. Zhou, T. Jiang, Y. Yan, Z. Liu, J. Zhang and S. Wang, "Far-field calculation of NEMP simulator near ground by transfer function," *2021 International Applied Computational Electromagnetics Society (ACES-China) Symposium*, Chengdu, China, 2021, pp. 1-2, doi: 10.23919/ACES-China52398.2021.9581424.

16. J. M. M. Junior, T. Khamvilai, L. Sutter and E. Feron, "Test platform for autopilot system embedded in a model of multi-core architecture using X-Plane flight

simulator," 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 2019, pp. 1-6, doi: 10.1109/DASC43569.2019.9081788.

17. D. A. G. Rey, G. S. I. Yajamín, R. C. B. Zurita and E. A. A. Rodríguez, "Simulation of the Boeing 737-500 Aircraft Takeoff Warning System," 2022 XII International Conference on Virtual Campus (JICV), Arequipa, Peru, 2022, pp. 1-4, doi: 10.1109/JICV56113.2022.9934375.

18. A. da Conceição Matheus, E. Villani and W. R. de Oliveira, "Washout filter parameterization of a robotic flight simulator," 2021 14th IEEE International Conference on Industry Applications (INDUSCON), São Paulo, Brazil, 2021, pp. 1023-1028, doi: 10.1109/INDUSCON51756.2021.9529739.

19. Z. Li, S. Perera, Y. Zhang, G. Zhang, R. Doviak and I. Ivić, "Evaluation of the Impacts of System Modules on Polarimetric Radar Data Quality Using a Phased Array Weather Radar System Simulator," 2019 IEEE International Symposium on Phased Array System & Technology (PAST), Waltham, MA, USA, 2019, pp. 1-5, doi: 10.1109/PAST43306.2019.9020739.

20. Radio Navigation Systems. URL: <https://avacs-avia.com/en/radionavigatsionnaya-sistema>

21. TACAN. URL: https://www.globalsecurity.org/military/library/policy/navy/nrtc/14090_ch2.pdf .

22. J. Duan, S. Yu, H. L. Tan, H. Zhu and C. Tan, "A Survey of Embodied AI: From Simulators to Research Tasks," in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 6, no. 2, pp. 230-244, April 2022, doi: 10.1109/TETCI.2022.3141105.

23. K. Smelyakov, A. Chupryna, O. Bohomolov and N. Hunko, "The Neural Network Models Effectiveness for Face Detection and Face Recognition," 2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), 2021, pp. 1-7, doi: 10.1109/eStream53087.2021.9431476.

24. K. Smelyakov, S. Smelyakov and A. Chupryna, Advances in Spatio-Temporal Segmentation of Visual Data. Chapter 1. Adaptive Edge Detection Models and

Algorithms. – Springer Nature Switzerland AG 2020, pp. 1–51. DOI:[10.1007/978-3-030-35480-0_1](https://doi.org/10.1007/978-3-030-35480-0_1).

25. S. Bieliievstvov, I. Ruban, K. Smelyakov and D. Sumtsov, “Network technology for transmission of visual information”, Selected Papers of the XVIII International Scientific and Practical Conference “Information Technologies and Security” (ITS 2018), Kyiv, Ukraine, November 27, 2018. In CEUR Workshop Proceedings, Vol-2318, 2018, pp. 160-175. <https://ceur-ws.org/Vol-2318/>.

26. Krivoulya G., Ilina I., Tokariev V., Shcherbak V. Mathematical Model for Finding Probability of Detecting Victims of Man-Made Disasters Using Distributed Computer System with Reconfigurable Structure and Programmable Logic / G. Krivoulya, V. Tokariev, I. Ilina, V. Shcherbak // IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology: (PIC S&T), 06-09 oct. 2020y. - Kharkiv, 2020. - P.573 - 576.

27. Krivoulya G., Tokariev V., Ilina I., Lebediev O., Shcherbak V. Algorithm of Iterations of Distribution of Subtasks Between «S-Bot» in One «Swarm-Bot» System // Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Systems: (COLINS 2022). CEUR Workshop Proceedings., 12-13 may. 2022 y. - Gliwice, Poland, 2022. - P. 1531-1541.

28. J. Ariza, M. Jimeno, R. Villanueva-Polanco and J. Capacho, "Provisioning Computational Resources for Cloud-Based e-Learning Platforms Using Deep Learning Techniques," in IEEE Access, vol. 9, pp. 89798-89811, 2021, doi: 10.1109/ACCESS.2021.3090366.

29. Federal Aviation Administration, "Distance Measuring Equipment (DME), Very High Frequency Omnidirectional Range (VOR), Tactical Air Navigation (TACAN) Sustainment Program (DVT Sustainment Program) Operational Safety Assessment (OSA)," February 10, 2021.

30. Federal Aviation Administration, “Preliminary Hazard Analysis for Distance Measuring Equipment used for Terminal and En Route Area Navigational Aid Facilities,” version 1.0, July 10, 2017.

31. Sauta O.I., Shatrakov A.Y., Shatrakov Y.G., Zavalishin O.I. Principles of Radio Navigation for Ground and Ship-Based Aircrafts. Springer Aerospace Technology. Springer, Singapore. 2019. - 130p. https://doi.org/10.1007/978-981-13-8293-2_4.

32. TACAN Antenna System FFHA, Version 1.2. URL: [https://imlive.s3.amazonaws.com/Federal%20Government/ID301675791526460350253803756283185162116/Tactical%20Air%20Navigation%20\(TACAN\)%20Antenna%20System%20Functional%20Fault%20Hazard%20Analysis%20\(FFHA\).pdf](https://imlive.s3.amazonaws.com/Federal%20Government/ID301675791526460350253803756283185162116/Tactical%20Air%20Navigation%20(TACAN)%20Antenna%20System%20Functional%20Fault%20Hazard%20Analysis%20(FFHA).pdf).

33. ANNEX 10 TO THE CONVENTION ON INTERNATIONAL CIVIL AVIATION. AERONAUTICAL TELECOMMUNICATIONS VOL. I. RADIO NAVIGATION AID, SEVENTH EDITION. 2018. URL: https://www.iacm.gov.mz/app/uploads/2018/12/an_10_v1_Aeronautical-Telecommunications_7ed_2018_rev.91_01.07.18.pdf.

34. MIL-STD-291C. STANDARD TACTICAL AIR NAVIGATION (TACAN) SIGNAL (1998). – Department of defense interfase standard. URL: http://everyspec.com/MIL-STD/MIL-STD-0100-0299/MIL-STD-291C_11561.

35. MM-7000MP Ruggedized Man Portable TACAN. AIRCRAFT | NAVIGATION AND SURVEILLANCE SYSTEMS. ©2018 Moog, Inc. URL: <https://www.moog.com/content/dam/moog/literature/Airc-raft/nass/Moog-MM7000MP-Brochure.pdf>.

36. MM-7000 Ruggedized TACAN System. AIRCRAFT | NAVIGATION AND SURVEILLANCE. SYSTEMS. ©2019 Moog, Inc. URL: <https://www.moog.com/content/dam/moog/literature/Air-craft/nass/Moog-MM7000TACAN-Brochure.pdf>.

37. Smelyakov, K., Bizkrovnyi, O., Sharonova, N., Smelyakov, S., Chupryna, A. Building of Regression Models for Cryptocurrency Price Prediction CEUR Workshop Proceedings, 2022, 3171, pp. 1216–1232.

38. Smelyakov K., Smelyakov S., Chupryna A. Advances in Spatio-Temporal Segmentation of Visual Data. Chapter 1. Adaptive Edge Detection Models and

Algorithms. Series Studies in Computational Intelligence (SCI), Vol. 876. – Publisher Springer, Cham, 2020. – P. 1-51. DOI <https://doi.org/10.1007/978-3-030-35480-0>.

39. Smelyakov, K., Prokopenko, O., Chupryna, A. Object-Based Image Comparison Algorithm Development for Data Storage Management Systems CEUR Workshop Proceedings, 2022, 3171, pp. 1251–1266.

40. Smelyakov, K., Chupryna, A., Darahan, D., Midina, S. Effectiveness of modern text recognition solutions and tools for common data sources CEUR Workshop Proceedings, 2021, 2870, стр. 154–165.

41. Smelyakov, K., SCOPUS, URL: <https://www.scopus.com/authid/detail.uri?authorId=57203149663>.