

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

Метод та інтелектуальна кіберфізична система для навігації БПЛА в умовах дії
засобів РЕБ противника
Назва теми

Рівень вищої освіти другий (магістерський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 123 «Комп'ютерна інженерія»

Шифр, назва

Освітня програма «Комп'ютерна інженерія та програмування»

Назва

Шифр КВРКІ 240123.24.01.23 ПЗ

Виконав здобувач ІІ курсу, група КІ2м-24-1


Підпис

Владислав СЕВОСТЬЯНОВ

Ініціали, прізвище

Керівник

канд.-техн. наук, доцент
Науковий ступінь, учене звання

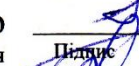

Підпис

Катерина БЕРЕЗЬКА

Ініціали, прізвище

Нормоконтролер

д. техн. наук, професор
Науковий ступінь, учене звання


Підпис

Сергій ЛИСЕНКО

Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КІС
«01» травня 2026 р.


Підпис

Ольга ПАВЛОВА

Ініціали, прізвище

дата

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ДРУГИЙ (МАГІСТЕРСЬКИЙ)

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 123 КОМП'ЮТЕРНА ІНЖЕНЕРІЯ

Освітня програма «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ ТА ПРОГРАМУВАННЯ»

ЗАТВЕРДЖУЮ
Завідувачка кафедри КІІС

Ольга ПАВЛОВА



“ 12 ” 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Севостьянову Владиславу Сергійовичу

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Метод та інтелектуальна кіберфізична система для навігації БПЛА в умовах дії засобів РЕБ противника

Керівник проекту (роботи) Березька К.М., к.т.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 12.01.2026 р. № 6

2. Термін подання здобувачем роботи на кафедру 05.05.2026 р.

3. Вихідні дані до роботи розробити метод та інтелектуальну кіберфізичну систему для навігації БПЛА в умовах дії засобів РЕБ противника

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) _____

Аналіз предметної області та наявних рішень навігації БПЛА в умовах відсутності GNSS-сигналу

Концепція інтелектуальної кіберфізичної системи для навігації БПЛА в умовах дії засобів РЕБ противника

Метод та алгоритм для вирішення задачі навігації БПЛА в умовах дії РЕБ противника

Результати роботи методу та інтелектуальної кіберфізичної системи для навігації БПЛА в умовах роботи засобів РЕБ противника

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання « 12 » 01 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) кваліфікаційної роботи магістра	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напряму дослідження та узгодження тематики КвРМ з керівником	12.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	12.01.2026	виконано
3	Робота над розділом 1 – аналіз відомих моделей, методів за темою; постановка задачі	20.01.2026	виконано
4	Робота над розділом 2 – розробка моделей для вирішення поставленої задачі	01.02.2026	виконано
5	Робота над науковою статтею	01.03.2026	виконано
6	Робота над розділом 3 – розробка методів для вирішення поставленої задачі	15.03.2026	виконано
7	Робота над розділом 4 – проектування та розробка ПЗ для вирішення поставленої задачі, експериментальна частина	01.04.2026	виконано
8	Оформлення пояснювальної записки згідно вимог	18.04.2026	виконано
9	Попередній захист ДРМ	29.04.2026	виконано
10	Захист ДРМ на засіданні ЕК	До 15.05.2026	

Здобувач


Підпис

Владислав СЕВОСТЬЯНОВ

Ім'я, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи


Підпис

Катерина БЕРЕЗЬКА

Ім'я, ПРІЗВИЩЕ

РЕФЕРАТ

Тема кваліфікаційної роботи магістра: Метод та інтелектуальна кіберфізична система для навігації БПЛА в умовах дії засобів РЕБ противника

Автор роботи: Владислав СЕВОСТЬЯНОВ

Керівник роботи: канд.-техн. наук, доцент Катерина БЕРЕЗЬКА

Пояснювальна записка: 74 с., 30 рис., 10 табл., 3 дод., 74 джерел.

ДРОН, НАВІГАЦІЯ, ОПТИЧНИЙ ПОТІК, БПЛА, ПОЗИЦІОНУВАННЯ, РОЗШИРЕНИЙ ФІЛЬТР КАЛМАНА.

Об'єктом дослідження є навігація БПЛА в умовах дії засобів РЕБ противника.

Предметом дослідження є застосування методу оптичного потоку та методу розширеної фільтрації Калмана для навігації БПЛА в умовах дії засобів РЕБ противника.

Метою кваліфікаційної роботи магістра є розробка методу та інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника.

Для розв'язання поставлених задач використовувалися методи обчислення оптичного потоку, методи математичного моделювання та метод розширеного фільтра Калмана.

Наукова новизна отриманих результатів полягає у вдосконаленні існуючих методів відносного позиціонування БПЛА за допомогою поєднання різних датчиків з датчиком оптичного потоку на основі рекурентної моделі штучного інтелекту, а також використання розширеного фільтра Калмана.

Набула подальшого розвитку інтелектуальна кіберфізична система та метод об'єднання датчиків з використанням оптичного потоку з подальшою фільтрацією за допомогою методу розширеного фільтра Калмана.

На основі проведених досліджень розроблена архітектура і компоненти інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника.

Практична значимість отриманих результатів полягає у розробці методу та інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника.

У першому розділі проведено аналіз існуючих методів позиціонування в умовах відсутності GNSS-сигналу.

В другому розділі побудовано концепцію системи позиціонування, розглянуто роботу двох поширених алгоритмів оптичного потоку та методу RAFT, описано математичну модель для використання фільтрації Калмана.

У третьому розділі було обрано експериментальним шляхом оптимальний метод та модель обчислення оптичного потоку, а також представлено метод та побудовано алгоритм роботи інтелектуальної кіберфізичної системи навігації БПЛА.

У четвертому розділі на основі розробленого методу та алгоритму роботи системи проведено експеримент для визначення точності роботи програмного забезпечення інтелектуальної кіберфізичної системи навігації БПЛА в умовах роботи засобів РЕБ противника.

ЗМІСТ

Скорочення та умовні позначки	5
Вступ.....	6
1 Аналіз предметної області та наявних рішень навігації БПЛА в умовах відсутності GNSS-сигналу.....	8
1.1 Аналіз предметної області і виявлення наявних проблем і завдань	8
1.2 Порівняльний аналіз переваг та недоліків існуючих рішень	10
1.3 Методологічні підходи до вирішення задачі за темою дослідження..	18
1.4 Висновки до першого розділу	19
2 Модель та методи вирішення поставленої задачі	20
2.1 Концепція інтелектуальної кіберфізичної системи для навігації БПЛА в умовах дії засобів РЕБ противника	20
2.2 Принцип роботи алгоритмів обчислення оптичного потоку	23
2.3 Побудова та принцип роботи розширеного фільтру Калмана	24
2.4 Висновки до другого розділу	42
3 Метод та алгоритм для вирішення задачі навігації БПЛА в умовах роботи засобів РЕБ противника.....	43
3.1 Вибір моделі обчислення оптичного потоку для вирішення задачі навігації БПЛА в умовах роботи засобів РЕБ противника	43
3.2 Метод вирішення задачі навігації БПЛА в умовах роботи засобів РЕБ противника	56
3.3 Алгоритм фільтрації та обчислення стану системи	61
3.4 Алгоритм побудови пройденого маршруту та його збереження	62
3.5 Висновки до третього розділу	64
4 Результати роботи інтелектуальної кіберфізичної системи для навігації БПЛА в умовах роботи засобів РЕБ противника.....	66
4.1 Аналіз вимог до розробки інтелектуальної кіберфізичної системи навігації БПЛА в умовах роботи засобів РЕБ противника	66

4.2 Розробка архітектури інтелектуальної кіберфізичної системи навігації БПЛА в умовах роботи засобів РЕБ противника.....	68
4.2 Проведення експерименту та аналіз результатів роботи програмного забезпечення інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника.....	69
4.3 Висновки до четвертого розділу.....	77
Висновки.....	79
Перелік джерел посилань.....	80
Додаток А Лістинг коду програмного забезпечення інтелектуальної кіберфізичної системи для навігації БПЛА в умовах дії засобів РЕБ противника	89
Додаток Б Публікація.....	103
Додаток В Презентація.....	115

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БПЛА – безпілотний літальний апарат

РЕБ – радіоелектронна боротьба

GPS – глобальна система позиціонування

GNSS – глобальна супутникова навігаційна система

ВСТУП

Стрімкий розвиток ударних безпілотних літальних апаратів (БПЛА) в умовах російсько-української війни показав необхідність та ефективність їх застосування. Технології їх протидії розвиваються як зі сторони Збройних Сил України, так і зі сторони противника. Посилене використання засобів радіоелектронної боротьби (РЕБ) противником часто унеможлиблює застосування звичних супутникових систем позиціонування. Розробка та вдосконалення альтернативних методів та систем позиціонування є актуальною проблемою сьогодення.

Важливим аспектом проблеми є не лише наявність рішень, а також їх ефективність та практичність застосування в бойових умовах. Рішення повинне бути максимально простим, надійним та компактним, щоб відповідати жорстким умовам використання та масовості.

Актуальність роботи полягає в розробці методу та інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника.

Метою кваліфікаційної роботи магістра є розробка методу та інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника.

Поставлена мета досягається шляхом розв'язання таких основних завдань:

- аналіз існуючих методів навігації та позиціонування в умовах відсутності GNSS-сигналу;
- вибір найбільш підходящого методу обчислення оптичного потоку;
- розробка математичної моделі системи з використанням методу розширеного фільтру Калмана;
- розробка методу та програмного забезпечення інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника;
- здійснення експерименту для перевірки розробленого методу навігації на основі розробленого програмного забезпечення інтелектуальної кіберфізичної системи навігації БПЛА.

Об'єктом дослідження є навігація БПЛА в умовах дії засобів РЕБ противника.

Предметом дослідження є застосування методу оптичного потоку та методу розширеної фільтрації Калмана для навігації БПЛА в умовах дії засобів РЕБ противника.

Наукова новизна отриманих полягає в розробленому методі відносного позиціонування БПЛА в умовах роботи засобів РЕБ противника, який об'єднує кілька датчиків, включно з датчиком оптичного потоку на основі рекурентної моделі та фільтрує значення методом розширеного фільтру Калмана. Такий підхід дозволяє створювати просту в застосуванні кіберфізичну систему позиціонування.

На основі проведених досліджень розроблено метод навігації БПЛА в умовах, де відсутній GNSS-сигнал. Також обрано експериментальним шляхом найкращий підхід до обчислення алгоритму оптичного потоку для поставленої задачі.

Практична значимість отриманих результатів полягає у розробленому програмному забезпеченні та інтелектуальній кіберфізичній системі навігації БПЛА в умовах роботи засобів РЕБ противника.

Для розв'язання поставлених задач використовувалися методи обчислення оптичного потоку на основі рекурентної моделі, методи математичного моделювання, а також метод фільтрації розширеного фільтру Калмана.

За темою кваліфікаційної роботи опубліковано одну публікацію [16] у міжнародному науково-технічному журналі «Вимірювальна та обчислювальна техніка в технологічних процесах». (Хмельницький – 2026). С. 28-39.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА НАЯВНИХ РІШЕНЬ НАВІГАЦІЇ БПЛА В УМОВАХ ВІДСУТНОСТІ GNSS-СИГНАЛУ

1.1 Аналіз предметної області і виявлення наявних проблем і завдань

На сьогодні система GPS (Global Positioning System) є найбільш популярною системою навігації у таких сферах як авіація, споживацькі пристрої, портативні навігаційні пристрої, автомобільна сфера та інші. Як наслідок, очікується значне збільшення ринку систем глобального позиціонування [1]. Високоточне позиціонування особливо потрібне в таких сферах як аграрна, гірнична, геодезія та автономних систем.

Попри високу точність та поширеність систем глобальної навігації, залишається проблема залежності системи позиціонування від супутникового сигналу, який постійно піддається «глушінню», особливо в умовах війни. Серед джерел завад GNSS (Global Navigation Satellite Systems) наводять наступні [2]:

- природні завади – це включає завади від атмосферних явищ;
- завади багатьох шляхів – це завади, що створені відбиттям сигналу з поверхонь будівель, рельєфу Землі, чи навіть гір, що призводить до помилок обчислення позиції;
- радіоперешкоди – завади, що створені радіосигналами з веж, радарів, чи навіть з особистих пристроїв. Такий тип завад може спотворювати, або ж повністю блокувати сигнал із супутників;
- джеммінг (jamming) – тип навмисно створених перешкод, які є потужним радіосигналом, що надсилається на тій же частоті, що й сигнал із супутників. Таким чином блокується отримуваний сигнал на пристрої приймачі;
- спуфінг (spoofing) – навмисно створені перешкоди, що підміняють сигнал із супутників. Таким чином пристрій отримувач сигналу вводиться в оману.

Дані за січень-травень 2024 року з Цюріхського університету прикладних наук та служби SkAI (рис. 1.1) показують зростання спуфінгу GPS-сигналу в різних куточках світу.

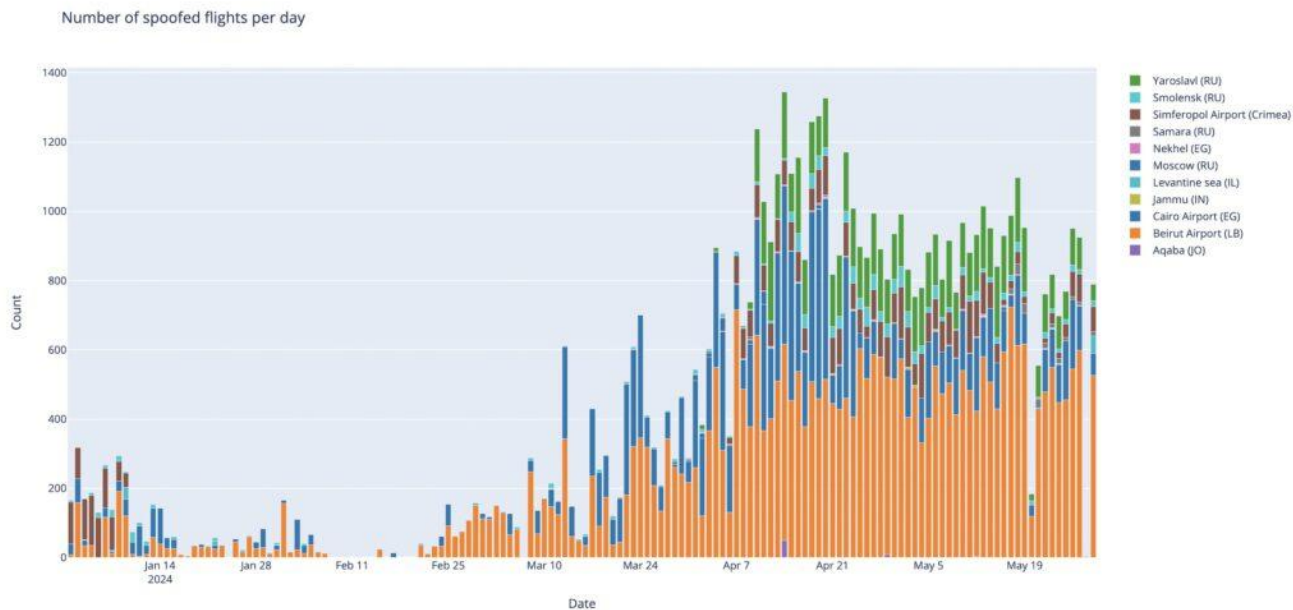


Рисунок 1.1 – Дані зафіксованих випадків спуфінгу GPS-сигналу [13]

Відповідно до даних міжнародної асоціації повітряного транспорту (IATA) кількість випадків втрат GPS сигналу під час польотів значно збільшилось у 2024 році (рис 1.2).

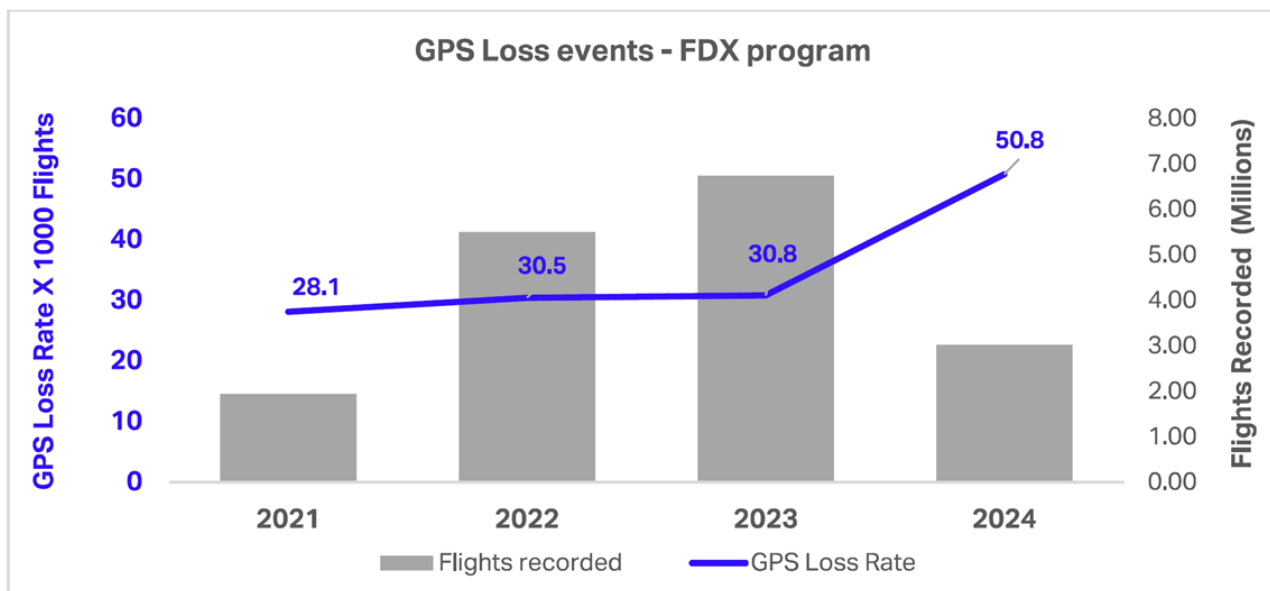


Рисунок 1.2 – Кількість випадків втрати GPS-зв’язку до років та зареєстрованих польотів [14]

При цьому значне збільшення випадків втрати GNSS-сигналу спостерігається саме в Чорному морі, близькому сході, частини території Білорусі та на заході Росії (рис. 1.3).

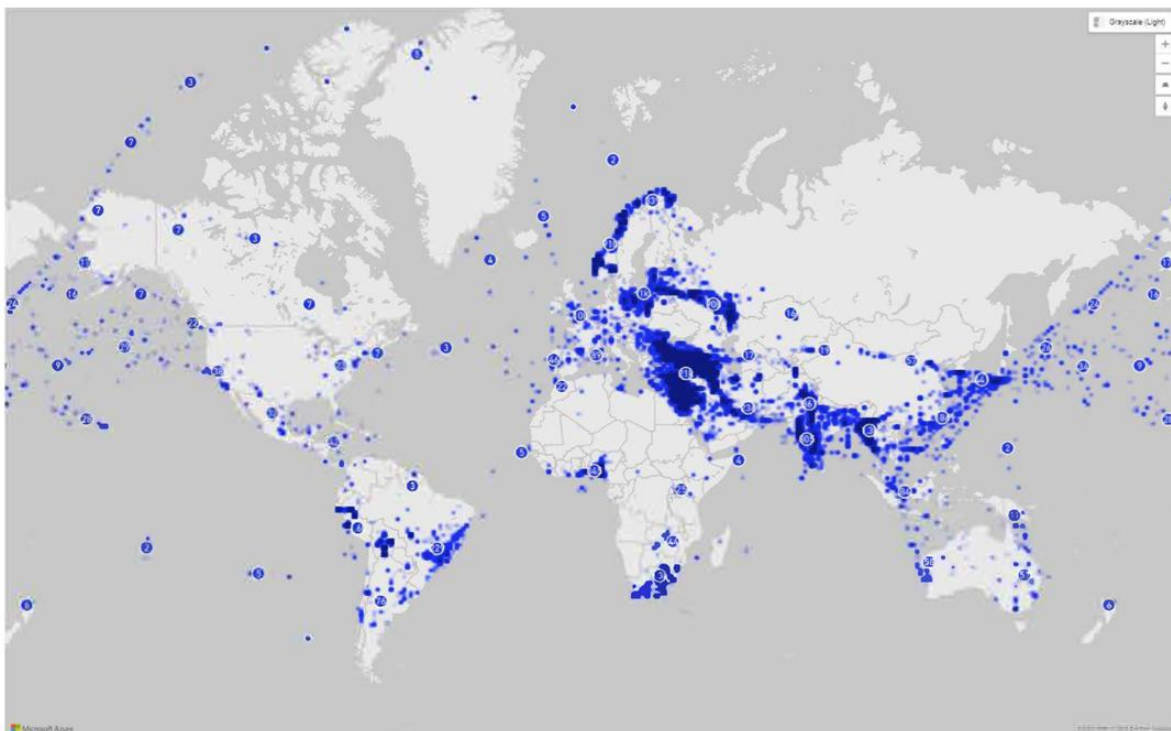


Рисунок 1.3 – Зареєстровані випадки завад GNSS-сигналу в світі [14]

В умовах російсько-української війни GNSS-сигнал піддається постійному «глушінню», тобто створенню різноманітних завад задля дезорієнтації противника [3]. Використання GPS є недоцільним для більшості задач із навігації та позиціонування в бойових умовах.

1.2 Порівняльний аналіз переваг та недоліків існуючих рішень

Навігацією називають процес відстеження та контролю руху судна, чи апарату з однієї точки в іншу. В контексті БПЛА навігація – це те, як визначає своє положення в просторі безпілотний літальний апарат.

Для визначення положення в просторі використовуються різноманітні датчики як GPS, інерційні датчики, радары, лідари та ін. Кожен з цих датчиків визначає певний аспект позиції дрона в просторі.

Кожен БПЛА має наступні складові:

1. Навігаційна система, яка визначає позицію дрона в реальному часі з отриманих даних з датчиків.
2. Контролююча система, яка отримує дані про позицію дрона з навігаційної системи та надсилає команди керування тілом дрона.
3. Датчики, які надсилають дані про стан системи в реальному часі.

На рисунку 1.4 зображено схему загальної структури функціонування БПЛА.

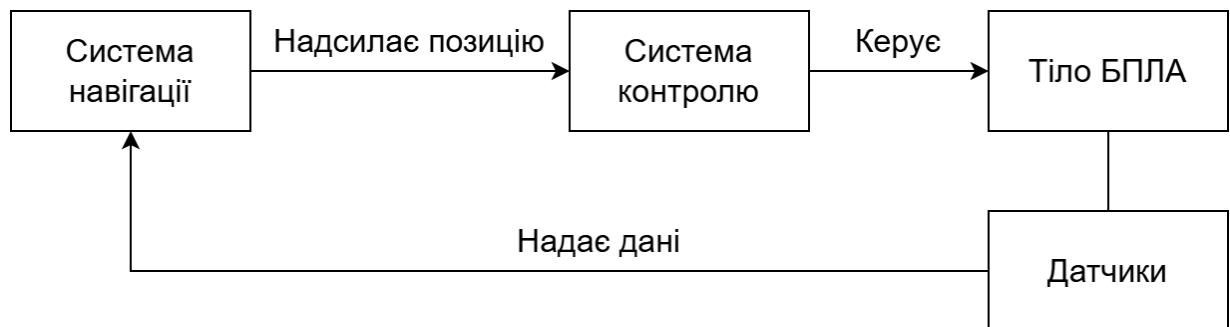


Рисунок 1.4 – Схема загальної структури функціонування БПЛА

Навігацію за способами позиціонування можна поділити на абсолютне та відносне позиціонування [4].

Абсолютне позиціонування визначає позицію відносно глобальної системи координат Землі. Окрім систем GNSS системою абсолютного позиціонування також виступають системи позиціонування за мапами місцевості та мапами висот. TRN (Terrain-Based Navigation) спирається на завчасно завантажену мапу місцевості та дані з таких датчиків, як камера, радар, або ж лірад, та порівнює отримані дані з вимірною мапою місцевості з тією, що зберігається в пам'яті пристрою [5].

Одним з прикладів абсолютного позиціонування в умовах відсутності GNSS-сигналу є системи TERCOM (Terrain contour matching), які застосовуються в

балістичних ракетах ще з 50-х років минулого сторіччя [6]. Системи TERCOM, як от найперша ATRAN (Automatic Terrain Recognition And Navigation), використовують для позиціонування радар, що постійно сканує поверхню Землі під ракетою, чи БПЛА (рис. 1.5).

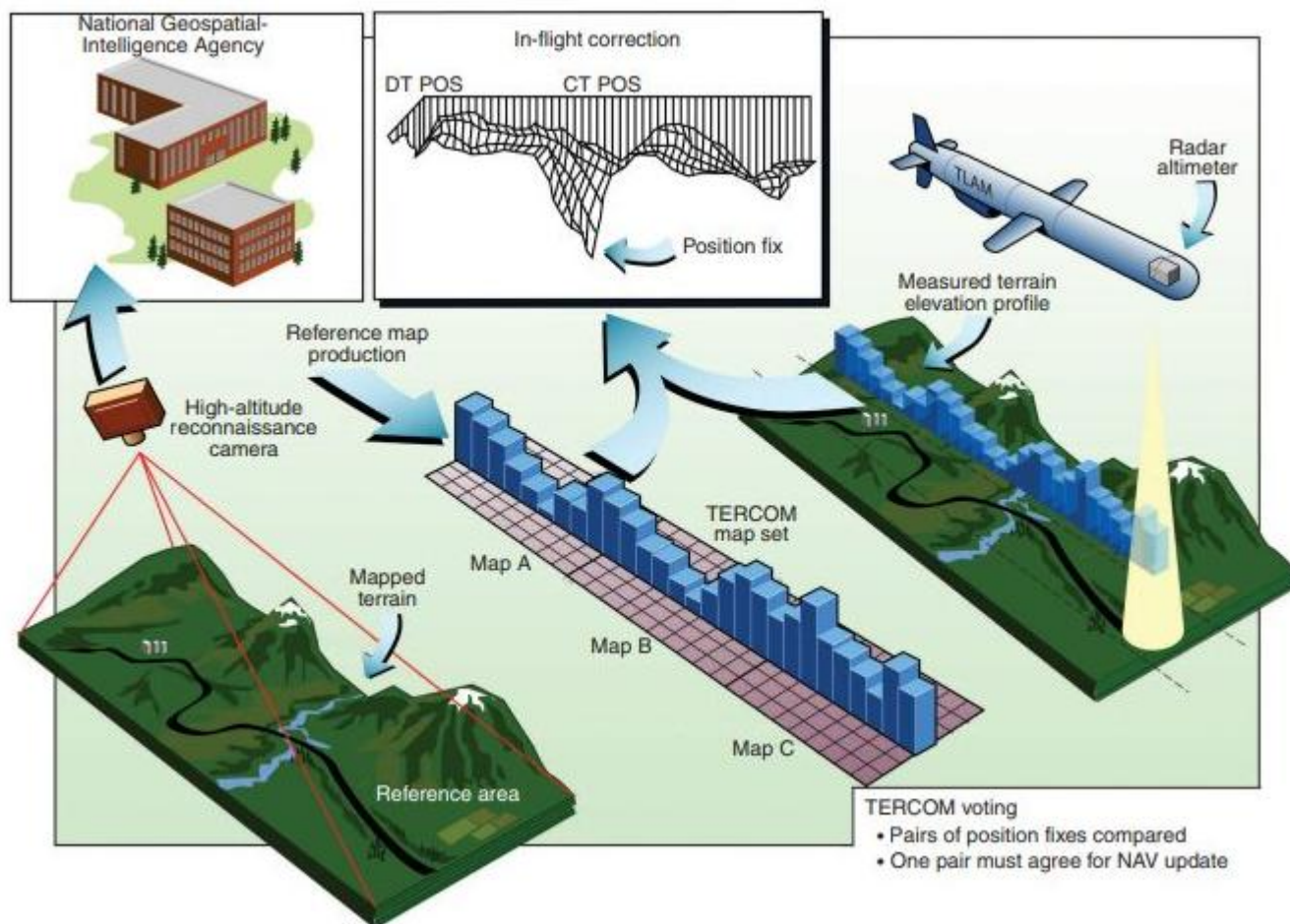


Рисунок 1.5 – Візуалізація загального принципу роботи системи TERCOM [7]

Знімки з радару порівнювались зі даними рельєфу, що були зібрані за допомогою подібного радару, який було встановлено на літальний апарат, що пролітав потрібний маршрут. Перед польотом в бортовому комп'ютері завантажувались мапа знімків з літального апарату та прокладений маршрут польоту ракети. Отриманий сигнал з радару утворював амплітудно-модульований сигнал, який демонстрував зміну рельєфу та відстань до поверхні Землі. Знімки з попереднього сканування радаром були записані на фоточутливу плівку, сканування якої утворювало схожий амплітудно-модульований сигнал. Похибка

між зчитаним сигналом з півки та з радару вела до корекцій маршруту ракети автопілотом [7].

Основним недоліком систем TERCOM є значне зменшення точності позиціонування на пласких місцевостях. Альтернативою є система SITAN, яка бере за основу принцип навігації за допомогою радару та корегує позицію за допомогою алгоритму фільтрації, як от фільтр Калмана. Це дає можливість позиціонування у складному та непередбачуваному середовищі.

Значне покращення абсолютного позиціонування пропонує система корелятор порівняння площі цифрових сцен (Digital Scene Matching Area Correlator (DSMAC)). DSMAC бере за основу відзняті монохромні знімки поверхні Землі та порівнює їх із візуальними даними з літального апарату. Такі знімки можуть збиратися місяць, чи більше до безпосередньо самого польоту апарату. Звідси можуть виникати різноманітні відмінності між віднятими сценами та зображеннями з камери апарату. Системи DSMAC дозволять працювати в умовах суттєвих змін відзнятих сцен, проте є обмеження відносно того, наскільки сильними можуть бути ці відмінності [8].

Така система найкраще себе показує в умовах середовища, де немає різких змін висот поверхні Землі, адже DSMAC покладається лише на візуальні зображення. Попри переваги та точність системи, основним недоліком залишається те, що потрібно регулярно поновлювати дані розвідки маршруту, по якому буде рухатись літальний апарат. Ще одним суттєвим недоліком є неможливість позиціонування в місцевості, попередня розвідка якої не була проведена. Даний метод добре підходить для складних та довгих місій. Апарати з подібною системою повинні бути достатньо великі, щоб вмістити її. Завчасне та детальне планування та збір даних про маршрут є обов'язковим для успішного виконання місії. Принцип роботи системи DSMAC зображено на рисунку 1.6.

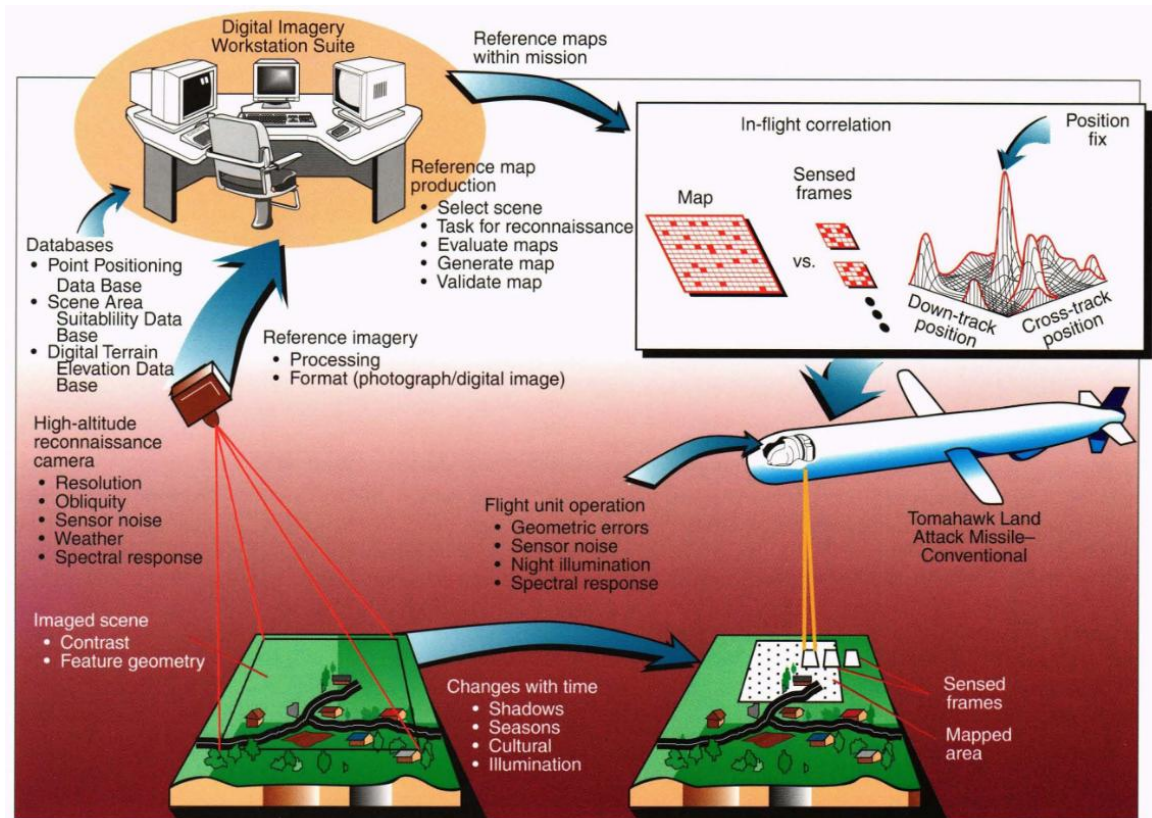


Рисунок 1.6 – Принцип роботи системи DSMAC на прикладі крилатої ракети Томагавк [8]

Серед нових підходів до абсолютного позиціонування основною тенденцією є передові технології візуалізації, алгоритми нейронних мереж та метод об'єднання датчиків. Такі системи використовують зображення поверхні Землі високої якості, дані із супутників, а також методи вилучення ознак на основі штучного інтелекту.

Прикладом такої системи є LSVL (Large-scale season-invariant visual localization). Даний метод використовує зображення високої роздільної здатності та дані із супутників для точного позиціонування, незалежно від сезону та зміни погоди [9].

Локалізоване позиціонування спирається на відносну систему координат, де, наприклад, точкою відліку координат є точка зльоту БПЛА. Такі системи зазвичай спираються лише на внутрішні датчики, як от акселерометр, гіроскоп, барометр, лідар, радар.

Одним зі способів локалізованого позиціонування є так зване числення координат (англ. dead reckoning), яке полягає у визначенні положення апарату,

спираючись на дані про попередню позицію, швидкість та напрямок руху [10]. Даний спосіб зазвичай використовує дані з внутрішніх датчиків, як от акселерометр та гіроскоп та застосовує фільтрацію для зменшення похибки, як от розширений фільтр Калмана. Попри простоту використання, даний метод не може визначати точну позицію довгий час через накопичення похибки, що спричинено недосконалістю датчиків та фільтрації.

Для покращення результатів вимірів внутрішніх датчиків використовують також методи візуальної одометрії та оптичного потоку [11]. Візуальна одометрія фокусується саме на визначенні відносної позиції апарату за допомогою камери та алгоритму оптичного потоку. Такий підхід дозволяє достатньо просто визначити напрямок та швидкість руху апарату без важкого, чи дуже складного обладнання. Попри це такі системи залежать від рівня освітлення навколишнього середовища.

Ще одним підходом локалізованого позиціонування є системи одночасного позиціонування та картографії (англ. Simultaneous localization and mapping (SLAM)). Такі системи використовують одночасно безліч датчиків, як от камери, лідари та радары для одночасної картографії місцевості навколо та позиціонування апарату в ній. SLAM дозволяє дуже точно визначити позицію в невідомому середовищі, проте потребує складної системи датчиків, що розташовані навколо апарату. Також системи SLAM є доволі важкими для обчислення, адже потребують одночасне збереження та доповнення місцевості у трьох вимірах.

Аналізуючи різні методи та підходи навігації в умовах відсутності GNSS-сигналу, було вибрано декілька статей, в яких було запропоновано та/чи реалізовано різні підходи навігації без GNSS-сигналу.

У таблиці 1.1 представлено аналіз деяких статей на тему розробки методів та підходів навігації в умовах відсутності GNSS-сигналу.

Таблиця 1.1 – Аналіз статей на тему розробки методів та підходів навігації в умовах відсутності GNSS-сигналу.

Номер	Назва методу	Опис методу
1	Інтегрований GNSS\LiDAR-SLAM фреймворк оцінки позиції	Запропонований підхід використовує взаємодоповнюваність GNSS-позиціонування та LiDAR-SLAM і розподіляє процес побудови карти залежно від стану GNSS RTK (кінематичного позиціонування в реальному часі). У зонах без GNSS застосовується алгоритм 3D LiDAR-SLAM для оцінки положення, а також алгоритм корекції для виправлення накопичених помилок (дрейфу) [46].
2	Навігація без GPS-сигналу за допомогою радіолокатора із синтезованою апертурою з використанням алгоритму доплерівського ефекту дальності	Стаття описує метод навігації без GPS для малогабаритних безпілотників із використанням SAR-радару, який забезпечує стабільну та точну оцінку положення навіть без супутникового сигналу. У цьому дослідженні розроблено метод навігації без GPS для легких літальних апаратів із використанням зображень, отриманих за допомогою радіолокатора із синтезованою апертурою (SAR). Зображення формуються за допомогою доплерівського ефекту дальності, який добре підходить для обмежених обчислювальних ресурсів легких літальних апаратів. Система інерціальної навігації та обробки радарних даних реалізована з використанням як реальних, так і змодельованих радарних зображень для оцінки стану літального апарата в умовах відсутності GPS [72].

Кінець таблиці 1.1

Номер	Назва методу	Опис методу
3	iSimLoc: Візуальна глобальна локалізація для невідомих середовищ за допомогою імітованих зображень	У цій статті представлено iSimLoc — підхід до глобальної повторної локалізації на основі машинного навчання, стійкий до змін вигляду сцени та ракурсу. Ознаки, які навчає мережа розпізнавання місць у iSimLoc, дозволяють зіставляти запитувані зображення з еталонними, навіть якщо вони належать до різних стилістичних доменів або мають різні ракурси [73].
4	Відносна навігації БПЛА літакового типу в умовах відсутності GPS-сигналу	У роботі реалізовано можливість польоту без GPS для безпілотних літальних апаратів літакового типу з урахуванням специфічних вимог до сенсорів і використанням підходу відносної навігації як основної концепції. Стаття описує систему навігації без GPS, яка використовує камеру та IMU разом із фільтром Калмана і графовою оптимізацією для точної оцінки положення [74].

Аналіз підходів показав тенденцію використання радарів, лідарів, чи камер для оцінки позиції БПЛА в просторі. Також багато методів використовують об'єднання даних з внутрішніх датчиків, як от акселерометр, чи гіроскоп, так із зовнішніх, як камери візуальної одометрії. Ще однією рисою розглянутих методів є використання алгоритмів фільтрації для уточнення позиції апарату.

Деякі методи були розроблені для застосування в малогабаритних БПЛА, що є також важливою рисою системи навігації. Практичні у застосуванні та невеликі в розмірі систем навігації мають більшу доцільність застосування, ніж громіздкі.

1.3 Методологічні підходи до вирішення задачі за темою дослідження

Система позиціонування БПАЛ з умовах дії РЕБ противника повинна бути достатньою для визначення відносної позиції апарату для можливого повернення до точки зльоту. Також система повинна бути повноцінною та працювати в незнайомій місцевості.

Для вирішення задачі позиціонування в реальному часі пропонується об'єднувати дані з різних джерел одночасно, а саме: дані з барометра та лідара для визначення висоти над поверхнею; дані з акселерометра та гіроскопа для визначення положення апарату в просторі; дані з камери оптичного потоку та компаса для ефективного визначення напрямку та швидкості руху апарату.

Пропонується також використання методу розширеного фільтру Калмана для уточнення позиції та фільтрації шумів з датчиків. Це дасть змогу зробити систему завадостійкою та попередити можливі збої роботи пропонованих датчиків.

Одним з недоліків датчиків оптичного потоку є шум зображення з камери, який негативно впливає на точність вимірюного напрямку руху та відстані. Для вирішення цієї проблеми пропонується використання додаткової фільтрації даних з датчика оптичного потоку, або ж використання рекурентної мережі для обчислення оптичного потоку. Пропонується провести порівняльний аналіз для вибору найкращого підходу обчислення оптичного потоку.

Задачею роботи є створення методів та інтелектуальної кіберфізичної системи навігації БПЛА в умовах роботи засобів РЕБ противника. Процес виконання завдання можна розбити на такі фази:

- формування об'єкту та мети поставленої задачі;
- дослідження функціональних особливостей системи навігації БПЛА в умовах роботи РЕБ противника;
- проведення теоретичного аналізу сфери системи навігації БПЛА в умовах роботи РЕБ противника;
- визначення предметної області та її структури;

- опис існуючих рішень реалізації даної задачі, визначення наявних в них проблем та запропонувати методи їх вирішення;
- визначення функціональних вимоги розроблювальної системи;
- оцінка виконання поставлених завдань;
- розробка системи навігації БПЛА в умовах роботи РЕБ противника;
- аналіз розробленої системи та методів.

1.4 Висновки до першого розділу

Зважаючи на сучасний темп російсько-української війни, де застосування БПЛА стало чи не найбільш вирішальною силою на полі бою, з'явилась потреба в постійному покращенні існуючих систем керування та позиціонування дронів на межі зіткнення. Підхід до позиціонування з використанням оптичного потоку не новий, проте може бути покращений за рахунок використання кількох додаткових датчиків одночасно, їх фільтрації, а також покращений підхід обчислення методу оптичного потоку.

Аналіз відомих методів навігації БПЛА в умовах відсутності GNSS-сигналу показав тенденцію застосування методів об'єднання внутрішніх датчиків з інерційної системи та зовнішніх датчиків, як радари, лідари, чи камери. Також широко використовуються різні методи фільтрації отриманих даних з датчиків для уточнення позиції апарату.

Для досягнення поставленої мети, а саме покращення методу навігації за БПЛА в умовах застосування засобів РЕБ противника, пропонується використання альтернативного підходу до обчислення алгоритму оптичного потоку, а також використання уточнення позиції за рахунок використання алгоритму фільтрації та додаткових датчиків.

2 МОДЕЛЬ ТА МЕТОДИ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1 Концепція інтелектуальної кіберфізичної системи для навігації БПЛА в умовах дії засобів РЕБ противника

Зважаючи на складність використання існуючих систем позиціонування в умовах відсутності GNSS-сигналу та їх дороговизну, розроблювана система має бути достатньо простою у використанні, компактною та недорогою в побудові. Проте, основними критеріями роботи системи навігації залишаються наступні:

- точність позиціонування. Система повинна мати достатню точність визначення позиції апарату для надійного утримання точки, прокладання маршруту, чи повернення до точки зльоту з визначеною похибкою;
- відмовостійкість. Система повинна бути стійкою до змін рельєфу, рівня освітленості, погодних умов та особливостей маршруту польоту апарату;
- надійність. Система повинна витримувати роботу за різних температурних режимів, а також мати способи забезпечення безпеки апарату, якщо неможливо визначити точну позицію.

Оптимальним вирішенням поставленої задачі є система, яка буде об'єднувати дані з кількох датчиків про позицію та напрямок руху дрона в певний момент часу. Обрані джерела даних про позиціонування не повинні залежати від попередньо визначених даних, як мапа місцевості, чи її рельєф. Датчики повинні бути повністю незалежними від радіосигналу, який може бути спотворений засобами РЕБ.

Окрім того, обраний алгоритм обробки даних з датчиків та їх об'єднання повинні бути достатньо надійним та ефективним у використанні. Потрібно також зважати на обмеженість обчислювальних ресурсів літальних апаратів.

Оскільки система не передбачається як повна заміна GPS, а однією з вимог є незалежність від зовнішніх попередньо зібраних даних, тому найкращим підходом є система відносного позиціонування.

Для визначення положення апарату в просторі та зменшення похибки найкращим підходом буде використання кількох датчиків. Серед таких датчиків

найкращим вибором будуть, що не потребують попередньо зібраних ресурсів для роботи. У таблиці 2.1 зазначено обрані датчики позиціонування в просторі.

Таблиця 2.1 – Обрані датчики позиціонування в просторі

Номер	Назва датчику	Дані про положенні в просторі	Опис
1	Акселерометр	Значення прискорення в трьох осях	Датчик визначає величину та вектор напрямку прискорення
2	Гіроскоп	Орієнтація тіла в просторі	Датчик визначає орієнтацію тіла в просторі
3	Магнітометр	Вказує на північний полюс Землі	Відкалібрований магнітометр може працювати в якості магнітного компаса
4	Барометр	Висота відносно точки зльоту	Датчик визначає тиск повітря, який змінюється з висотою
5	Лідар	Висота відносно поверхні Землі	Датчик вимірює відстань до об'єкту, що можна використати для визначення висоти, спрямувавши його донизу

Кінець таблиці 2.1

Номер	Назва датчику	Дані про положенні в просторі	Опис
6	Камера оптичного потоку	Напрямок та пройдена відстань відносно поверхні Землі	Датчик вимірює магнітуду та напрямок руху

Велику роль у точності позиціонування відіграє обраний метод фільтрації даних з датчиків. Найбільш поширеним та випробуваним часом є розширений фільтр Калмана, який, враховуючи похибку з датчиків та визначає коефіцієнт коригування передбачуваної оцінки стану апарату в просторі.

На основі попереднього аналізу існуючих рішень пропонується концептуальна модель апарату, основні компоненти якого описано вище. На рисунку 2.1 зображено концептуальне рішення системи відносного позиціонування дрона.

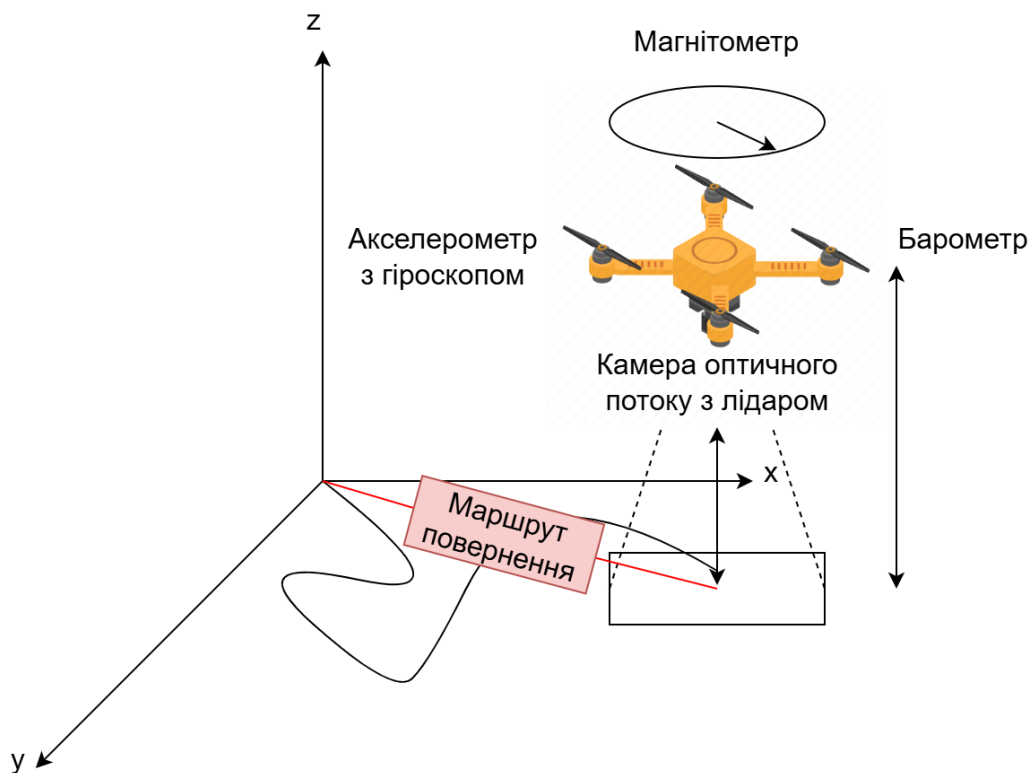


Рисунок 2.1 – Концептуальна схема системи позиціонування дрона

2.2 Принцип роботи алгоритмів обчислення оптичного потоку

Визначення 2.1. Оптичний потік – це двовимірне векторне поле, що описує видимий рух кожного пікселя, що викликано відносним рухом між камерою (спостерігачем) та об'єктами, що спостерігаються [15].

Обчислення оптичного потоку відбувається між послідовними кадрами відео. Розрізняють два основних підходи до обчислення оптичного потоку:

1. Розріджений (sparse) оптичний потік обчислює рух між обраними точками двох зображень. Точки інтересу обчислюються алгоритмами виявлення кутів, або ж країв, такими як алгоритм Сі — Томазі. Розріджений алгоритм зазвичай обчислюється швидше, проте має меншу точність в шумних зображеннях, чи зображеннях, де мало контрасту. Прикладом розрідженого алгоритму обчислення оптичного потоку є метод Лукаса — Канаде.

2. Щільний (dense) алгоритм обчислює рух між кожним пікселем двох зображень. Даний підхід дає більш точний рух, адже враховує кожен піксель, проте потребує більшої обчислювальної потужності. Прикладом щільного методу обчислення оптичного потоку є методи Горна — Шунка, чи Фарнебека.

Ще одним підходом до обчислення оптичного потоку є алгоритми на основі моделей глибокого навчання. Серед алгоритмів можна навести FlowNet, FlowNet 2.0, LiteFlowNet, PWC-Net та інші. Здебільшого такі алгоритми використовують згорткові моделі навчання. Такий підхід збільшує надійність обчислення оптичного потоку у важких умовах, коли зображення шумне, або ж об'єкти погано освітлені.

Покращеним варіантом обчислення оптичного потоку на основі глибокого навчання є модель, яка за основу бере рекурентну модель навчання. Таким підходом є RAFT (Recurrent All-Pairs Field Transforms), схема роботи якого зображено на рисунку 2.2 [12].

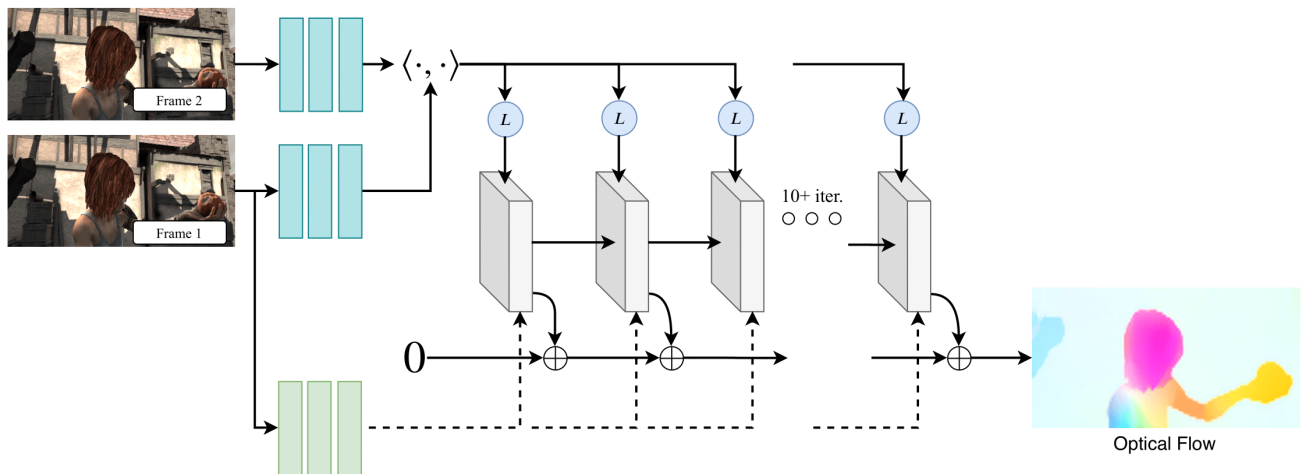


Рисунок 2.2 – Схема роботи алгоритму оптичного потоку на основі рекурентної моделі штучного інтелекту [12]

Основою пропонованої системи є алгоритм оптичного потоку, від точності роботи якого буде залежати якість навігації, а саме побудова пройденого маршруту та маршруту повернення.

Серед потенційних кандидатів є класичні методи обчислення оптичного потоку, як методи Лукаса — Канаде та Фарнебека, а також новий підхід обчислення оптичного потоку із застосуванням рекурентної нейронної мережі RAFT.

2.3 Побудова та принцип роботи розширеного фільтра Калмана

Дрон може переміщуватися в усіх напрямках з різною лінійною швидкістю. Також він може мати різний кут обертання та швидкість обертання навколо своєї осі. Стан апарату в певний момент часу визначається його просторовою моделлю. Просторова модель складається з параметрів відстежуваного об'єкту. У цьому випадку такими параметрами виступають координати в просторі та кути нахилу по кожній з осей координатного простору.

Тут поворот навколо осі X називається креном, поворот навколо осі Y називається тангажем, а поворот навколо осі Z називається роланням.

Просторова система слугує для відстеження параметрів об'єкту в просторі в будь-який момент часу. Кожна її складова відображає певний аспект відстежуваного об'єкту.

Для розробки інтелектуальної кіберфізичної системи навігації БПЛА узято просторову модель квадрокоптера зі складовими координат та кутів нахилів навколо кожної з осей.

Просторова модель існує в системі координат, як зображено на рисунку 2.3.

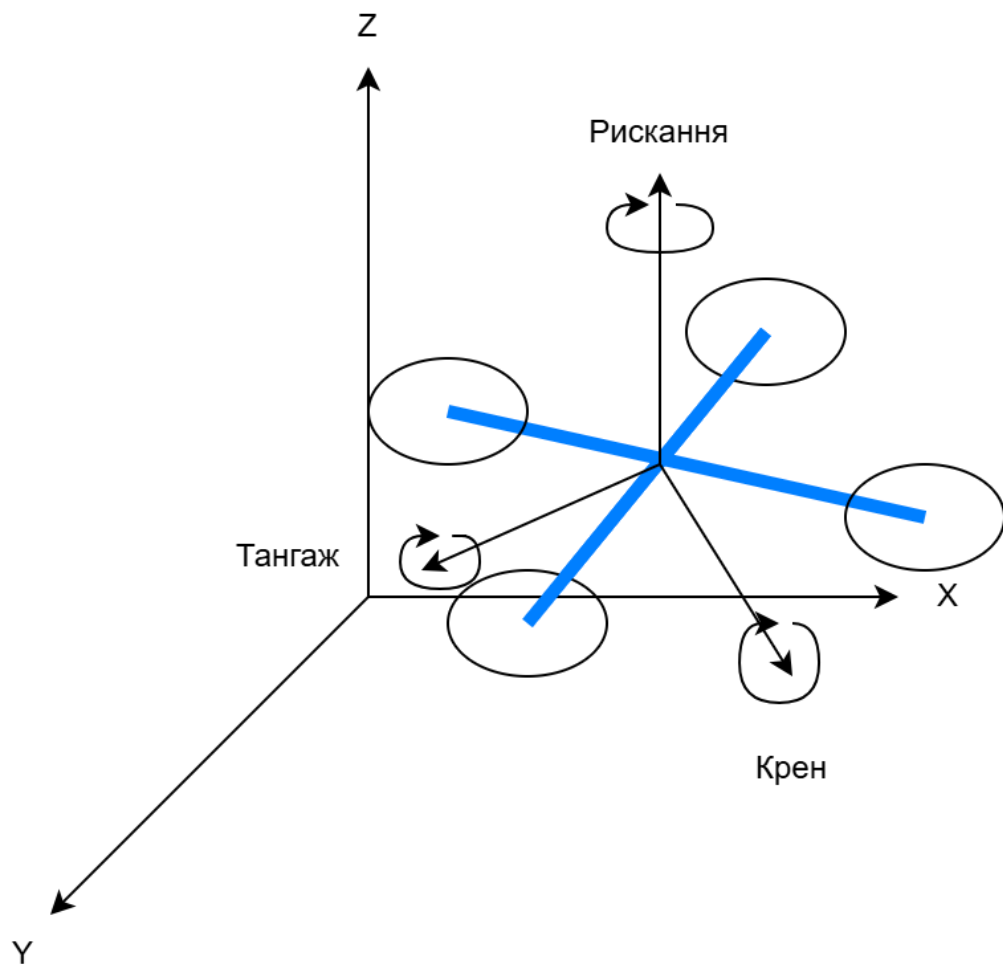


Рисунок 2.3 – Схематичне представлення просторової моделі дрона

Стан в такій системі представляється формулою:

$$S(t) = A(t)x(t) + B(t)u(t), \quad (2.1)$$

де S – вектор стану;

A – квадратна матриця стану системи;

B – вхідна матриця;

u – контролюючий вектор;

t – момент часу вимірів стану системи.

Стан дрона в певний момент часу буде представлятись вектором розміром 6, як показано у формулі:

$$S_t = \begin{bmatrix} x_t \\ y_t \\ z_t \\ \alpha_t \\ \beta_t \\ \gamma_t \end{bmatrix}, \quad (2.2)$$

де x_t – координата x в метрах в час t ;

y_t – координата y в метрах в час t ;

z_t – координата z в метрах в час t ;

α_t – кут повороту в радіанах навколо осі X в час t ;

β_t – кут повороту в радіанах навколо осі Y в час t ;

γ_t – кут повороту в радіанах навколо осі Z в час t .

Дані з датчиків представляються як вектор u_t керуючих змінних, з яких уже обчислюються складові вектора стану дрона.

Для такої системи для обчислення стану в час t дані зі швидкості з датчика оптичного потоку беруться з часу $t - 1$, оскільки положення вираховується з прискорення та швидкості одночасно.

Кожен з датчиків є складовою вектору вимірів. Вектор вимірів з датчиків у час t представлений формулою:

$$u_t = \begin{bmatrix} a_{x_t} \\ a_{y_t} \\ a_{z_t} \\ v_{\alpha_t} \\ v_{\beta_t} \\ v_{\gamma_t} \\ h_t \\ mag_t \\ v_{xof_{t-1}} \\ v_{yof_{t-1}} \\ x_{of_t} \\ y_{of_t} \end{bmatrix}, \quad (2.3)$$

де a_{x_t} – прискорення з акселерометра в метрах на секунду в квадраті по осі x дрона в час t;

a_{y_t} – прискорення з акселерометра в метрах на секунду в квадраті по осі y в час t;

a_{z_t} – прискорення з акселерометра в метрах на секунду в квадраті по осі z в час t;

v_{α_t} – кутова швидкість в радіанах на секунду навколо осі x в час t;

v_{β_t} – кутова швидкість в радіанах на секунду навколо осі y в час t;

v_{γ_t} – кутова швидкість в радіанах на секунду навколо осі z в час t;

h_t – висота з барометра та лідара в метрах;

mag_t – кут повороту в радіанах навколо осі z відносно півночі Землі;

$v_{xof_{t-1}}$ – лінійна швидкість з датчика оптичного потоку уздовж осі x дрона в час t - 1;

$v_{yof_{t-1}}$ – лінійна швидкість в метрах на секунду з датчика оптичного потоку уздовж осі y в час t - 1;

x_{of_t} – позиція борта по осі X з датчика оптичного потоку в час t;

y_{of_t} – позиція борта по осі Y з датчика оптичного потоку в час t;

Проте також потрібно враховувати, що рух дрона вимірюється в глобальній площині, хоч і у відносних координатах. Дані з датчиків акселерометра та оптичного потоку видають значення в системі координат апарату. Відповідно

потрібно перетворювати значення з датчиків акселерометра та оптичного потоку з локальної системи координат у глобальну.

Для цього існують так звані поворотні матриці, за допомогою яких відбувається перетворення векторів з одної системи координат в іншу, знаючи кути повороту по кожній з осей в Ейлеровій системі координат. Поворотна матриця в трьох осях має наступне представлення:

$$R_{xyz} = \begin{bmatrix} \cos \gamma \cos \beta & \cos \gamma \sin \beta \sin \alpha - \sin \gamma \cos \alpha & \cos \gamma \sin \beta \cos \alpha + \sin \gamma \sin \alpha \\ \sin \gamma \cos \beta & \sin \gamma \sin \beta \sin \alpha + \cos \gamma \cos \alpha & \sin \gamma \sin \beta \cos \alpha - \cos \gamma \sin \alpha \\ -\sin \beta & \cos \beta \sin \alpha & \cos \beta \cos \alpha \end{bmatrix}, \quad (2.4)$$

де R_{xyz} – поворотна матриця для осей X, Y та Z;

γ – кут повороту навколо осі Z в радіанах;

β – кут повороту навколо осі Y в радіанах;

α – кут повороту навколо осі X в радіанах.

Потрібні кути вимірюються гіроскопом, тому перетворення зробити достатньо просто, помноживши матрицю проєкцій векторів прискорення на матрицю повороту по всім осям.

Описати вектор прискорення по осям X, Y та Z в локальній системі координат можна наступною формулою:

$$a_{local} = \begin{bmatrix} a_{lx} \\ a_{ly} \\ a_{lz} \end{bmatrix}, \quad (2.5)$$

де a_{local} – вектор прискорення в локальній системі координат;

a_{lx} – прискорення в локальній системі координат по осі X;

a_{ly} – прискорення в локальній системі координат по осі Y;

a_{lz} – прискорення в локальній системі координат по осі Z.

Вектор прискорення в глобальній системі координат представлено формулою:

$$a_{global} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}, \quad (2.6)$$

де a_{global} – вектор прискорення в глобальній системі координат;

a_x – прискорення в глобальній системі координат по осі X;

a_y – прискорення в глобальній системі координат по осі Y;

a_z – прискорення в глобальній системі координат по осі Z.

Відповідно залежність глобального вектору прискорення від локального описується такою формулою:

$$a_{global} = R_{xyz}a_{local}, \quad (2.7)$$

де a_{global} – вектор прискорення в глобальній системі координат;

R_{xyz} – поворотна матриця в трьох осях;

a_{local} – вектор прискорення в локальній системі координат.

Те ж стосується швидкостей, що обчислюються модулем оптичного потоку. Вектор швидкостей оптичного потоку в локальній системі координат описаний формулою:

$$v_{local} = \begin{bmatrix} v_{lxof} \\ v_{lyof} \\ 0 \end{bmatrix}, \quad (2.8)$$

де v_{local} – вектор швидкості в локальній системі координат;

v_{lxof} – швидкість з датчика оптичного потоку по осі X в локальній системі координат;

v_{lyof} – швидкість з датчика оптичного потоку по осі Y в локальній системі координат.

В глобальній системі координат вектор швидкості представлено формулою:

$$v_{global} = \begin{bmatrix} v_{xof} \\ v_{yof} \\ 0 \end{bmatrix}, \quad (2.9)$$

де v_{global} – вектор швидкості в глобальній системі координат;

v_{xof} – швидкість з датчика оптичного потоку по осі X в глобальній системі координат;

v_{yof} – швидкість з датчика оптичного потоку по осі Y в глобальній системі координат.

Якщо у випадку прискорення по акселерометру кожен вектор діяв лише в системі координат дрона, то датчик оптичного потоку одразу вимірює швидкість руху в глобальній системі, незалежно від кутів нахилу апарату по крену та тангажу. Єдиний кут, який не є врахованим під час виміру, це кут азимуту спрямування дрона. Відповідно потрібно перетворювати вектори швидкості руху по осям X та Y відносно азимуту спрямування дрона.

Для цього існують часткові матриці повороту. Матриця повороту навколо осі Z має наступне представлення:

$$R_z = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (2.10)$$

де R_z – матриця повороту навколо осі Z ;

γ – кут повороту навколо осі Z в радіанах.

Для обчислення вектору швидкості руху в глобальній системі координат потрібно помножити матрицю повороту на вектор швидкості руху з датчика оптичного потоку, як описано у формулі:

$$v_{global} = R_z v_{local}, \quad (2.11)$$

де v_{global} – вектор швидкості в глобальній системі координат;

R_z – матриця повороту по осі Z;

v_{local} – вектор швидкості в локальній системі координат;

Координати з датчика оптичного потоку потребують аналогічного перетворення. Опис вектору позиції апарату по осям X та Y в локальній системі координат:

$$p_{local} = \begin{bmatrix} x_{lof} \\ y_{lof} \\ 0 \end{bmatrix}, \quad (2.12)$$

де p_{local} – позиція в локальній системі координат;

x_{lof} – позиція з датчика оптичного потоку по осі X в локальній системі координат;

y_{lof} – позиція з датчика оптичного потоку по осі Y в локальній системі координат.

В глобальній системі координат вектор позиції матиме наступний вигляд:

$$p_{global} = \begin{bmatrix} x_{of} \\ y_{of} \\ 0 \end{bmatrix}, \quad (2.13)$$

де p_{global} – позиція в глобальній системі координат;

x_{of} – позиція з датчика оптичного потоку по осі X в глобальній системі координат;

y_{of} – позиція з датчика оптичного потоку по осі Y в глобальній системі координат.

Для обчислення вектора швидкості руху в глобальній системі координат потрібно помножити матрицю повороту на вектор швидкості руху з датчика оптичного потоку:

$$p_{global} = R_z p_{local}, \quad (2.14)$$

де p_{global} – вектор позиції в глобальній системі координат;

R_z – матриця повороту по осі Z ;

p_{global} – вектор позиції в локальній системі координат;

Матриця A відображає те, як стан дрона змінюється, якщо ніяких контролюючих команд не дано. У випадку з квадрокоптером, є сили, що постійно діють на нього, навіть якщо ніякого керування немає. Серед цих сил: вітер та гравітація. Матриця A є матрицею часткових похідних функцій стану по складовим вектора стану.

Функції стану для кожної складової матриці стану мають наступний вигляд:

$$\begin{bmatrix} x_t \\ y_t \\ z_t \\ \alpha_t \\ \beta_t \\ \gamma_t \end{bmatrix} = \begin{bmatrix} x_{of_{t-1}} + v_{xof_{t-2}} \Delta t + \frac{1}{2} a_{x_{t-1}} \Delta t^2 \\ y_{of_{t-1}} + v_{yof_{t-2}} \Delta t + \frac{1}{2} a_{y_{t-1}} \Delta t^2 \\ h_{t-1} + v_{z_{t-2}} \Delta t + \frac{1}{2} a_{z_{t-1}} \Delta t^2 \\ \alpha_{t-1} + v_{\alpha_{t-1}} \Delta t \\ \beta_{t-1} + v_{\beta_{t-1}} \Delta t \\ mag_{t-1} + v_{\gamma_{t-1}} \Delta t \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}, \quad (2.15)$$

де f_1-f_6 – функції стану дрона;

Δt – різниця в часі між попереднім та поточним вимірюваннями з датчиків.

Матриця стану A обчислена формулою:

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_{t-1}} & \frac{\partial f_1}{\partial y_{t-1}} & \frac{\partial f_1}{\partial z_{t-1}} & \frac{\partial f_1}{\partial \alpha_{t-1}} & \frac{\partial f_1}{\partial \beta_{t-1}} & \frac{\partial f_1}{\partial \gamma_{t-1}} \\ \frac{\partial f_2}{\partial x_{t-1}} & \frac{\partial f_2}{\partial y_{t-1}} & \frac{\partial f_2}{\partial z_{t-1}} & \frac{\partial f_2}{\partial \alpha_{t-1}} & \frac{\partial f_2}{\partial \beta_{t-1}} & \frac{\partial f_2}{\partial \gamma_{t-1}} \\ \frac{\partial f_3}{\partial x_{t-1}} & \frac{\partial f_3}{\partial y_{t-1}} & \frac{\partial f_3}{\partial z_{t-1}} & \frac{\partial f_3}{\partial \alpha_{t-1}} & \frac{\partial f_3}{\partial \beta_{t-1}} & \frac{\partial f_3}{\partial \gamma_{t-1}} \\ \frac{\partial f_4}{\partial x_{t-1}} & \frac{\partial f_4}{\partial y_{t-1}} & \frac{\partial f_4}{\partial z_{t-1}} & \frac{\partial f_4}{\partial \alpha_{t-1}} & \frac{\partial f_4}{\partial \beta_{t-1}} & \frac{\partial f_4}{\partial \gamma_{t-1}} \\ \frac{\partial f_5}{\partial x_{t-1}} & \frac{\partial f_5}{\partial y_{t-1}} & \frac{\partial f_5}{\partial z_{t-1}} & \frac{\partial f_5}{\partial \alpha_{t-1}} & \frac{\partial f_5}{\partial \beta_{t-1}} & \frac{\partial f_5}{\partial \gamma_{t-1}} \\ \frac{\partial f_6}{\partial x_{t-1}} & \frac{\partial f_6}{\partial y_{t-1}} & \frac{\partial f_6}{\partial z_{t-1}} & \frac{\partial f_6}{\partial \alpha_{t-1}} & \frac{\partial f_6}{\partial \beta_{t-1}} & \frac{\partial f_6}{\partial \gamma_{t-1}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (2.16)$$

Матриця B виражає те, як буде змінюватися стан системи від контролюючих команд на дрон. Часткові похідні матриці B розраховуються по складових керуючого вектору. Обчислена матриця B матиме наступний вигляд:

$B =$

$$\begin{aligned}
 & \left[\begin{array}{cccccccccccc}
 \frac{\partial f_1}{\partial \alpha_{x_{t-1}}} & \frac{\partial f_1}{\partial a_{y_{t-1}}} & \frac{\partial f_1}{\partial a_{z_{t-1}}} & \frac{\partial f_1}{\partial v_{\alpha_{t-1}}} & \frac{\partial f_1}{\partial v_{\beta_{t-1}}} & \frac{\partial f_1}{\partial v_{\gamma_{t-1}}} & \frac{\partial f_1}{\partial h_{t-1}} & \frac{\partial f_1}{\partial \text{tag}_{t-1}} & \frac{\partial f_1}{\partial v_{x_{of_{t-2}}}} & \frac{\partial f_1}{\partial v_{y_{of_{t-2}}}} & \frac{\partial f_1}{\partial x_{of_{t-1}}} & \frac{\partial f_1}{\partial y_{of_{t-1}}} \\
 \frac{\partial f_2}{\partial \alpha_{x_{t-1}}} & \frac{\partial f_2}{\partial a_{y_{t-1}}} & \frac{\partial f_2}{\partial a_{z_{t-1}}} & \frac{\partial f_2}{\partial v_{\alpha_{t-1}}} & \frac{\partial f_2}{\partial v_{\beta_{t-1}}} & \frac{\partial f_2}{\partial v_{\gamma_{t-1}}} & \frac{\partial f_2}{\partial h_{t-1}} & \frac{\partial f_2}{\partial \text{tag}_{t-1}} & \frac{\partial f_2}{\partial v_{x_{of_{t-2}}}} & \frac{\partial f_2}{\partial v_{y_{of_{t-2}}}} & \frac{\partial f_2}{\partial x_{of_{t-1}}} & \frac{\partial f_2}{\partial y_{of_{t-1}}} \\
 \frac{\partial f_3}{\partial \alpha_{x_{t-1}}} & \frac{\partial f_3}{\partial a_{y_{t-1}}} & \frac{\partial f_3}{\partial a_{z_{t-1}}} & \frac{\partial f_3}{\partial v_{\alpha_{t-1}}} & \frac{\partial f_3}{\partial v_{\beta_{t-1}}} & \frac{\partial f_3}{\partial v_{\gamma_{t-1}}} & \frac{\partial f_3}{\partial h_{t-1}} & \frac{\partial f_3}{\partial \text{tag}_{t-1}} & \frac{\partial f_3}{\partial v_{x_{of_{t-2}}}} & \frac{\partial f_3}{\partial v_{y_{of_{t-2}}}} & \frac{\partial f_3}{\partial x_{of_{t-1}}} & \frac{\partial f_3}{\partial y_{of_{t-1}}} \\
 \frac{\partial f_4}{\partial \alpha_{x_{t-1}}} & \frac{\partial f_4}{\partial a_{y_{t-1}}} & \frac{\partial f_4}{\partial a_{z_{t-1}}} & \frac{\partial f_4}{\partial v_{\alpha_{t-1}}} & \frac{\partial f_4}{\partial v_{\beta_{t-1}}} & \frac{\partial f_4}{\partial v_{\gamma_{t-1}}} & \frac{\partial f_4}{\partial h_{t-1}} & \frac{\partial f_4}{\partial \text{tag}_{t-1}} & \frac{\partial f_4}{\partial v_{x_{of_{t-2}}}} & \frac{\partial f_4}{\partial v_{y_{of_{t-2}}}} & \frac{\partial f_4}{\partial x_{of_{t-1}}} & \frac{\partial f_4}{\partial y_{of_{t-1}}} \\
 \frac{\partial f_5}{\partial \alpha_{x_{t-1}}} & \frac{\partial f_5}{\partial a_{y_{t-1}}} & \frac{\partial f_5}{\partial a_{z_{t-1}}} & \frac{\partial f_5}{\partial v_{\alpha_{t-1}}} & \frac{\partial f_5}{\partial v_{\beta_{t-1}}} & \frac{\partial f_5}{\partial v_{\gamma_{t-1}}} & \frac{\partial f_5}{\partial h_{t-1}} & \frac{\partial f_5}{\partial \text{tag}_{t-1}} & \frac{\partial f_5}{\partial v_{x_{of_{t-2}}}} & \frac{\partial f_5}{\partial v_{y_{of_{t-2}}}} & \frac{\partial f_5}{\partial x_{of_{t-1}}} & \frac{\partial f_5}{\partial y_{of_{t-1}}} \\
 \frac{\partial f_6}{\partial \alpha_{x_{t-1}}} & \frac{\partial f_6}{\partial a_{y_{t-1}}} & \frac{\partial f_6}{\partial a_{z_{t-1}}} & \frac{\partial f_6}{\partial v_{\alpha_{t-1}}} & \frac{\partial f_6}{\partial v_{\beta_{t-1}}} & \frac{\partial f_6}{\partial v_{\gamma_{t-1}}} & \frac{\partial f_6}{\partial h_{t-1}} & \frac{\partial f_6}{\partial \text{tag}_{t-1}} & \frac{\partial f_6}{\partial v_{x_{of_{t-2}}}} & \frac{\partial f_6}{\partial v_{y_{of_{t-2}}}} & \frac{\partial f_6}{\partial x_{of_{t-1}}} & \frac{\partial f_6}{\partial y_{of_{t-1}}}
 \end{array} \right] = \\
 & \left[\begin{array}{cccccccccccc}
 \frac{\Delta t^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t & 0 & 1 & 0 \\
 0 & \frac{\Delta t^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \Delta t & 0 & 1 \\
 0 & 0 & \frac{\Delta t^2}{2} & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & \Delta t & 0 & 1 & 0 & 0 & 0
 \end{array} \right], \tag{2.17}
 \end{aligned}$$

Основою розширеного фільтра Калмана є передбачення вимірів з датчиків та порівняння отриманих значень з реальними вимірами. Для передбачення будується модель, описана формулою:

$$p_t = H_t * S_t + w_t, \tag{2.18}$$

де p_t – вектор передбачень вимірювань з датчиків у час t ;

H_t – матриця, яка перетворює передбачуваний стан системи в час t у набір передбачуваних вимірів з датчиків у час t ;

w_t – вектор сукупності похибок (шумів) з датчиків у час t .

H є матрицею часткових похідних функцій перетворення стану системи по складовим стану системи в час t .

Виведені функції перетворень стану апарату в час $t-1$ у передбачені виміри з датчиків у час t представлені формулою:

$$\begin{bmatrix} a_{x_t} \\ a_{y_t} \\ a_{z_t} \\ v_{\alpha_t} \\ v_{\beta_t} \\ v_{\gamma_t} \\ h_t \\ mag_t \\ v_{x_{of_{t-1}}} \\ v_{y_{of_{t-1}}} \\ x_{of_t} \\ y_{of_t} \end{bmatrix} = \begin{bmatrix} \frac{2(x_t - x_{t-1} - v_{x_{of_{t-2}}}\Delta t)}{\Delta t^2} \\ \frac{2(y_t - y_{t-1} - v_{y_{of_{t-2}}}\Delta t)}{\Delta t^2} \\ \frac{2(z_t - h_{t-1} - v_{z_{t-2}}\Delta t)}{\Delta t^2} \\ \frac{\alpha_t - \alpha_{t-1}}{\Delta t} \\ \frac{\beta_t - \beta_{t-1}}{\Delta t} \\ \frac{\gamma_t - \gamma_{t-1}}{\Delta t} \\ z_t - v_{z_{t-2}}\Delta t - \frac{1}{2}a_{z_{t-1}}\Delta t^2 \\ \gamma_t - v_{\gamma_{t-1}}\Delta t \\ \frac{x_t - x_{t-1} - \frac{1}{2}a_{x_{t-1}}\Delta t^2}{\Delta t} \\ \frac{y_t - y_{t-1} - \frac{1}{2}a_{y_{t-1}}\Delta t^2}{\Delta t} \\ x_t - v_{x_{of_{t-2}}}\Delta t - \frac{1}{2}a_{x_{t-1}}\Delta t^2 \\ y_t - v_{y_{of_{t-2}}}\Delta t - \frac{1}{2}a_{y_{t-1}}\Delta t^2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \\ f_{10} \\ f_{11} \\ f_{12} \end{bmatrix}, \quad (2.19)$$

де f_1-f_{12} – функції перетворення стану апарату в передбачені виміри з датчиків.

Для визначення матриці H також потрібно розрахувати матрицю Якобі часткових похідних функції перетворення передбачених вимірів з датчиків по складовим вектору стану.

Матриця H є результатом обчислення часткових похідних функцій перетворення з формули 2.20. Часткові похідні обчислюються за складовими вектору стану з формули 2.3.

Одержана матриця матиме розміри $m \times n$, де m – це розмір вектору вимірів з датчиків з формули 2.4, а n – це розмір вектору стану з формули 2.3.

Обчислена матриця H матиме наступний вигляд:

$$H = \begin{bmatrix} \frac{\partial f_1}{\partial x_t} & \frac{\partial f_1}{\partial y_t} & \frac{\partial f_1}{\partial z_t} & \frac{\partial f_1}{\partial \alpha_t} & \frac{\partial f_1}{\partial \beta_t} & \frac{\partial f_1}{\partial \gamma_t} \\ \frac{\partial f_2}{\partial x_t} & \frac{\partial f_2}{\partial y_t} & \frac{\partial f_2}{\partial z_t} & \frac{\partial f_2}{\partial \alpha_t} & \frac{\partial f_2}{\partial \beta_t} & \frac{\partial f_2}{\partial \gamma_t} \\ \frac{\partial f_3}{\partial x_t} & \frac{\partial f_3}{\partial y_t} & \frac{\partial f_3}{\partial z_t} & \frac{\partial f_3}{\partial \alpha_t} & \frac{\partial f_3}{\partial \beta_t} & \frac{\partial f_3}{\partial \gamma_t} \\ \frac{\partial f_4}{\partial x_t} & \frac{\partial f_4}{\partial y_t} & \frac{\partial f_4}{\partial z_t} & \frac{\partial f_4}{\partial \alpha_t} & \frac{\partial f_4}{\partial \beta_t} & \frac{\partial f_4}{\partial \gamma_t} \\ \frac{\partial f_5}{\partial x_t} & \frac{\partial f_5}{\partial y_t} & \frac{\partial f_5}{\partial z_t} & \frac{\partial f_5}{\partial \alpha_t} & \frac{\partial f_5}{\partial \beta_t} & \frac{\partial f_5}{\partial \gamma_t} \\ \frac{\partial f_6}{\partial x_t} & \frac{\partial f_6}{\partial y_t} & \frac{\partial f_6}{\partial z_t} & \frac{\partial f_6}{\partial \alpha_t} & \frac{\partial f_6}{\partial \beta_t} & \frac{\partial f_6}{\partial \gamma_t} \\ \frac{\partial f_7}{\partial x_t} & \frac{\partial f_7}{\partial y_t} & \frac{\partial f_7}{\partial z_t} & \frac{\partial f_7}{\partial \alpha_t} & \frac{\partial f_7}{\partial \beta_t} & \frac{\partial f_7}{\partial \gamma_t} \\ \frac{\partial f_8}{\partial x_t} & \frac{\partial f_8}{\partial y_t} & \frac{\partial f_8}{\partial z_t} & \frac{\partial f_8}{\partial \alpha_t} & \frac{\partial f_8}{\partial \beta_t} & \frac{\partial f_8}{\partial \gamma_t} \\ \frac{\partial f_9}{\partial x_t} & \frac{\partial f_9}{\partial y_t} & \frac{\partial f_9}{\partial z_t} & \frac{\partial f_9}{\partial \alpha_t} & \frac{\partial f_9}{\partial \beta_t} & \frac{\partial f_9}{\partial \gamma_t} \\ \frac{\partial f_{10}}{\partial x_t} & \frac{\partial f_{10}}{\partial y_t} & \frac{\partial f_{10}}{\partial z_t} & \frac{\partial f_{10}}{\partial \alpha_t} & \frac{\partial f_{10}}{\partial \beta_t} & \frac{\partial f_{10}}{\partial \gamma_t} \\ \frac{\partial f_{11}}{\partial x_t} & \frac{\partial f_{11}}{\partial y_t} & \frac{\partial f_{11}}{\partial z_t} & \frac{\partial f_{11}}{\partial \alpha_t} & \frac{\partial f_{11}}{\partial \beta_t} & \frac{\partial f_{11}}{\partial \gamma_t} \\ \frac{\partial f_{12}}{\partial x_t} & \frac{\partial f_{12}}{\partial y_t} & \frac{\partial f_{12}}{\partial z_t} & \frac{\partial f_{12}}{\partial \alpha_t} & \frac{\partial f_{12}}{\partial \beta_t} & \frac{\partial f_{12}}{\partial \gamma_t} \end{bmatrix} = \begin{bmatrix} \frac{2}{\Delta t^2} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{2}{\Delta t^2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2}{\Delta t^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\Delta t} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\Delta t} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{\Delta t} \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{1}{\Delta t} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\Delta t} & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad (2.20)$$

В загальному фільтр Калмана має два основних етапи:

1. Етап передбачення - відбувається передбачення стану системи в час t , маючи стан системи в час $t - 1$, а також відбувається передбачення коваріації апроксимації стану, відповідно до попередньої коваріації та шуму.

2. Етап оновлення - розраховується різниця між реальними вимірами із сенсорів та передбачуваними вимірами; розраховується коваріація відхилення; розраховується близький до оптимального передавальний коефіцієнт Калмана; розраховується оновлена оцінка стану для часу t ; оновлюється коваріація оцінки стану для часу t .

Після ініціалізації системи попередніх даних про стан системи ще не існує, тому ми вважаємо, що координати дрона рівні нулю. Вектор початкового стану дрона описано формулою:

$$S_{t-1} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \\ \alpha_{t-1} \\ \beta_{t-1} \\ \gamma_{t-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (2.21)$$

Значення керуючого вектору також будуть дорівнювати нулю. Початковий вектор вимірів описаний формулою:

$$u_{t-1} = \begin{bmatrix} a_{x_t} \\ a_{y_t} \\ a_{z_t} \\ v_{\alpha_t} \\ v_{\beta_t} \\ v_{\gamma_t} \\ h_t \\ mag_t \\ v_{xof_{t-1}} \\ v_{yof_{t-1}} \\ x_{of_t} \\ y_{of_t} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (2.22)$$

Наступним етапом є передбачення коваріації оцінки стану. На цьому етапі відбувається передбачення того, наскільки добре система оцінює свій стан, тобто наскільки передбачення оцінки стану системи відповідає зчитуваним значенням з датчиків. Передбачення коваріації стану описано формулою:

$$P_t = A_t P_{t-1} A_t^T + Q_t, \quad (2.23)$$

де P_t – квадратна матриця залежності складових моделей стану одна від одної в час t ;

Q_t – матриця коваріації шуму моделі стану в час t .

По діагоналі матриця P позначає варіації, тобто 100% відповідність, а всі інші значення є коваріацією, тобто залежністю одного значення від іншого. Матриця A є матрицею стану системи, а A^T є транспонованою матрицею A .

Матриця Q є квадратною матрицею коваріації шуму моделі стану. Розмір цієї матриці залежить від кількості змінних моделі стану. Матриця коваріації шуму представлена формулою:

$$Q_t = \begin{bmatrix} Cov(x, x) & Cov(x, y) & Cov(x, z) & Cov(x, \alpha) & Cov(x, \beta) & Cov(x, \gamma) \\ Cov(y, x) & Cov(y, y) & Cov(y, z) & Cov(y, \alpha) & Cov(y, \beta) & Cov(y, \gamma) \\ Cov(z, x) & Cov(z, y) & Cov(z, z) & Cov(z, \alpha) & Cov(z, \beta) & Cov(z, \gamma) \\ Cov(\alpha, x) & Cov(\alpha, y) & Cov(\alpha, z) & Cov(\alpha, \alpha) & Cov(\alpha, \beta) & Cov(\alpha, \gamma) \\ Cov(\beta, x) & Cov(\beta, y) & Cov(\beta, z) & Cov(\beta, \alpha) & Cov(\beta, \beta) & Cov(\beta, \gamma) \\ Cov(\gamma, x) & Cov(\gamma, y) & Cov(\gamma, z) & Cov(\gamma, \alpha) & Cov(\gamma, \beta) & Cov(\gamma, \gamma) \end{bmatrix} = \begin{bmatrix} Var(x) & Cov(x, y) & Cov(x, z) & Cov(x, \alpha) & Cov(x, \beta) & Cov(x, \gamma) \\ Cov(y, x) & Var(y) & Cov(y, z) & Cov(y, \alpha) & Cov(y, \beta) & Cov(y, \gamma) \\ Cov(z, x) & Cov(z, y) & Var(z) & Cov(z, \alpha) & Cov(z, \beta) & Cov(z, \gamma) \\ Cov(\alpha, x) & Cov(\alpha, y) & Cov(\alpha, z) & Var(\alpha) & Cov(\alpha, \beta) & Cov(\alpha, \gamma) \\ Cov(\beta, x) & Cov(\beta, y) & Cov(\beta, z) & Cov(\beta, \alpha) & Var(\beta) & Cov(\beta, \gamma) \\ Cov(\gamma, x) & Cov(\gamma, y) & Cov(\gamma, z) & Cov(\gamma, \alpha) & Cov(\gamma, \beta) & Var(\gamma) \end{bmatrix}, \quad (2.24)$$

Оскільки величини x, y, z, α, β та γ є незалежними одна від одної, їх коваріація буде рівною нулю. Також, для початкового стану ми припускаємо, що дисперсія всіх величин буде рівною 1. Відповідно матриця Q буде одиничною, як представлено у формулі:

$$Q_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.25)$$

В якості початкового значення матриці P_{t-1} може бути діагональна матриця довільних коефіцієнтів. Наприклад матрицю P_{t-1} можна прирівняти наступним чином:

$$P_{t-1} = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}, \quad (2.26)$$

Якщо підставити всі отримані значення, то отримаємо наступне представлення:

$$P_t = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.27)$$

На наступному етапі обчислюється вектор d різниці між реальними та передбачуваними показниками з датчиків. Вектор різниці передбачень та реальних вимірів представлено формулою:

$$d_t = u_t - p_t = u_t - H_t S_t + w_t, \quad (2.28)$$

де u_t - це вектор зчитаних даних з датчиків в час t ;

p_t - це вектор передбачених даних з датчиків в час t .

На наступному етапі використовується передбачена коваріація оцінки стану P_t , матрицю H_t , її транспозицію та матрицю R_t , яка позначає коваріацію шуму отриманих даних з датчиків.

Далі обчислюється коваріація оновлення C :

$$C_t = H_t P_t H_t^T + R_t, \quad (2.29)$$

де C_t – матриця коваріації оновлення в час t ;

R_t – матриця коваріації шуму отриманих даних з датчиків у час t .

Для початку матрицю коваріації шуму можна ініціалізувати як одиничну, а потім уже коригувати відповідно до похибки зчитуваних значень з датчиків. Матриця коваріації шуму представлена формулою:

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.30)$$

Далі розраховується найголовніший коефіцієнт, який буде напряму впливати на оновлення оцінки стану дрона. Близько-оптимальний коефіцієнт Калмана розраховується за формулою:

$$K_t = P_t H_t^T C_t^{-1}, \quad (2.31)$$

де K_t – близько-оптимальний коефіцієнт Калмана.

Коефіцієнт Калмана позначає те, наскільки стан та коваріація передбачуваної оцінки стану має бути скоригована. Якщо шум із сенсорів дуже великий, то коефіцієнт Калмана буде досягати нуля, а зчитані дані з датчиків будуть переважно ігноруватися.

Далі відбувається оновлення оцінки стану за формулою:

$$S_t = S_{t-1} + K_t d_t, \quad (2.32)$$

Якщо шум передбачення стану великий, тоді коефіцієнт Калмана буде досягати 1, а зчитані дані з датчиків будуть домінувати в оцінці стану системи.

На останньому етапі відбувається коваріація оцінки стану за формулою:

$$P_t = (I - K_t H_t) P_{t-1}, \quad (2.33)$$

де I – це одинична матриця.

На цьому етапі ми оновлюємо значення матриці передбачення коваріації оцінки стану.

Далі отримане значення передбачення коваріації оцінки стану системи використовується для обчислення коваріації оновлення після отримання нових даних з датчиків та обчислення відхилення вимірювання.

Обчислення стану системи із застосуванням розширеного фільтру Калмана можна розбити на 9 етапів, 4 з яких є підготовчими для ініціалізації системи, а 5 наступних виконуються циклічно.

Застосування такого методу фільтрації покращить завадостійкість та надійність системи, адже буде враховано шуми та можливі відмови датчиків.

Алгоритм обчислення фільтру Калмана для розроблюваної системи зображено на рисунку 2.4.

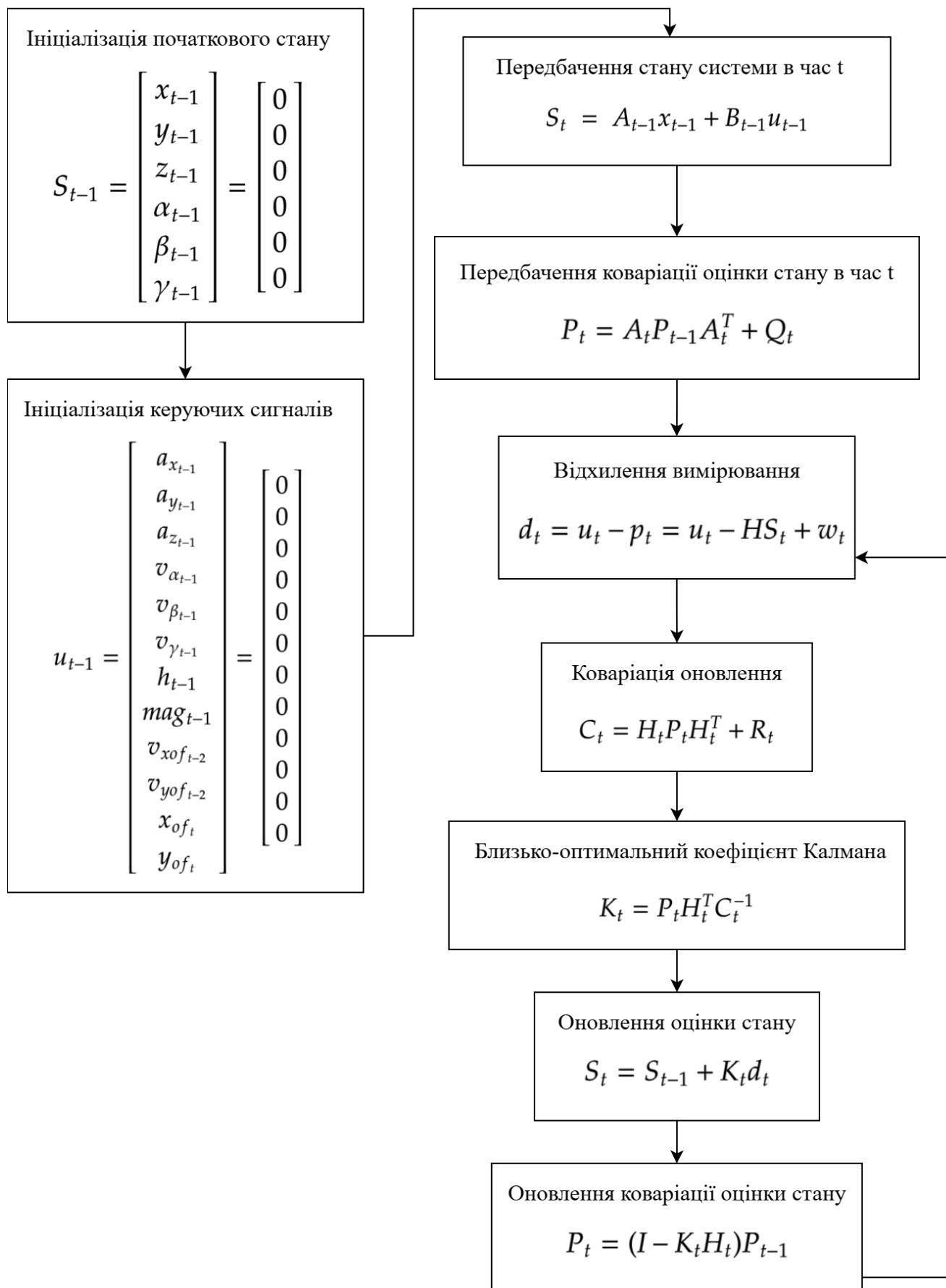


Рисунок 2.4 – Алгоритм обчислення стану системи методом розширеної фільтрації Калмана

2.4 Висновки до другого розділу

У цьому розділі було розроблено загальну концепцію методу та кіберфізичної системи навігації БПЛА в умовах роботи засобів РЕБ противника. Розроблено концепцію об'єднання датчиків акселерометра, гіроскопа, лідара, барометра та магнітометра для уточнення позиції, яка буде обчислюватись за допомогою модуля оптичного потоку.

Розглянуто принцип роботи поширених алгоритмів оптичного потоку та нового методу обчислення на основі рекурентної моделі RAFT.

Описано математичну модель за методом розширеного фільтру Калмана, використання якого дозволить зробити систему стійкою до шумів та відмов датчиків. Розроблено алгоритм обчислення математичної моделі для розроблюваної системи.

3 МЕТОД ТА АЛГОРИТМ ДЛЯ ВИРІШЕННЯ ЗАДАЧІ НАВІГАЦІЇ БПЛА В УМОВАХ РОБОТИ ЗАСОБІВ РЕБ ПРОТИВНИКА

3.1 Вибір моделі обчислення оптичного потоку для вирішення задачі навігації БПЛА в умовах роботи засобів РЕБ противника

Загальний алгоритм роботи системи позиціонування складається із передбачення наступного стану просторової моделі апарату, збору даних з датчиків та внесення коригування в просторову модель.

Для вибору оптимального алгоритму обчислення оптичного потоку для датчика оптичного потоку було проведено експерименти з порівняльним аналізом результатів.

В якості алгоритму розрідженого підходу було обрано метод Лукаса — Канаде. В якості алгоритму щільного підходу було обрано метод Фарнебека.

Усі обчислення оптичного потоку виконувались функціями бібліотеки OpenCV мовою програмування Python.

На додачу до розрідженого та щільного алгоритмів обчислення оптичного потоку було додано мінімальну фільтрацію та згладжування даних для аналізу їх впливу на точність результату. Даний метод попри свою складність є вразливим до шумів зображення, що створюють похибку та неточності обчислення оптичного потоку. Для зменшення отриманої похибки було реалізовано трирівневу фільтрацію отриманих даних.

Перший рівень фільтрує отримані значення магнітуди та кутів за значенням магнітуди.

Другий рівень згладжує значення отриманих кутів. Оскільки алгоритм обчислення оптичного потоку може видавати велику похибку, отримані кути було вирішено згладжувати методом експоненційного згладжування. Дане згладжування добре працює з часовими рядами, проте є проблема згладжування кутів, що вказують на напрямок 0° , або 360° (0 радіан, або 6,28 радіан), тому для кутів експоненційне згладжування було застосовано до їх синусів та косинусів, з яких уже було знову побудовано згладжені значення кутів.

Експоненційне згладжування у формулі 3.1 обчислює нове значення часового ряду відповідно до попереднього, поточного та коефіцієнту згладжування.

$$s_t = \begin{cases} c_1 & : t = 1 \\ s_{t-1} + a(c_t - s_{t-1}) & : t > 1 \end{cases} \quad (3.1)$$

де s_t – нове згладжене значення в час t ;

s_{t-1} – попереднє згладжене значення в час $t - 1$;

a – коефіцієнт згладжування;

c_1 – перше значення часового ряду;

c_t – значення часового ряду в час t .

Дана формула застосовується для синуса та косинуса кожного кута вектора відповідно, а з отриманих значень знаходиться новий кут за допомогою функції atan2 .

Третій рівень згладжує магнітуду за методом експоненційного згладжування.

Наступний етап включає в себе знаходження середніх значень магнітуди та кутів векторів. Середні значення кутів знаходяться також за синусом та косинусом через проблему переходу значень радіан кутів з $6,28$ на 0 та навпаки.

Отримані значення магнітуди та кутів векторів виводяться як графіки у полярних координатах та декартових координатах, які окремо рисують значення кутів та магнітуди. Також виводяться зображення з оброблюваного відеопотоку та оптичного потоку. Виводиться обчислений маршрут руху з даних алгоритму оптичного потоку, відстань та напрямок до початку руху. Обчислюється та виводиться мінімальне, максимальне та середнє значення часу виконання алгоритму оптичного потоку.

Було проведено серію тестів для вибору оптимального алгоритму обчислення оптичного потоку [16]. Перше тестування включає в себе відео польоту дрона на висоті 25 метрів та опущеною камерою донизу. Тут було протестовано вплив згладжування та фільтрації на щільний та розріджений алгоритми. Оброблювальне зображення було зменшено до розмірів 300 на 300 пікселів для пришвидшення

процесу тестування. Критеріями оцінки були подібність пройденого маршруту дроном до того, що обчислює оптичний потік. Схематично маршрут польоту дрона виглядав наступним чином, як зображено на рисунку 3.1.

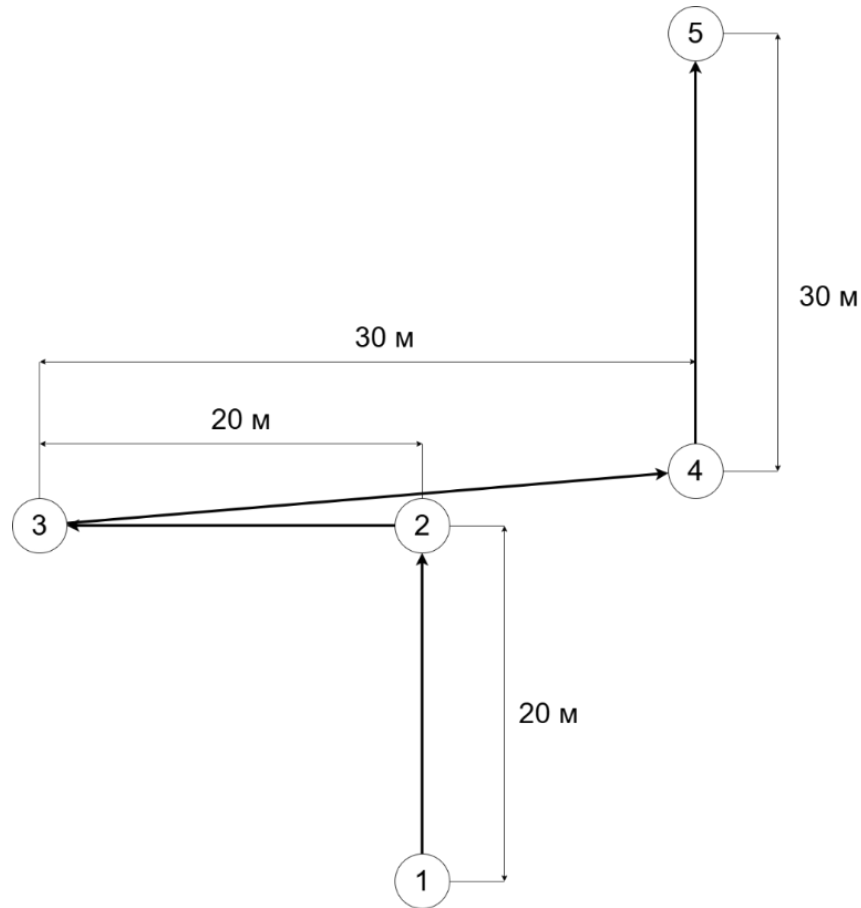


Рисунок 3.1 – Схематичне зображення пройденого дроном шляху

Для проведення тестування було розроблено програмне забезпечення, яке зчитує покадрово відео та оброблює методами оптичного потоку. Отримані магнітуда та кут вектора руху перетворюються в приблизний маршрут руху.

Також окремо обчислюється маршрут повернення апарату до точки зльоту.

На рисунку 3.2 зображено приклад виводу програми для тестування оптичного потоку.

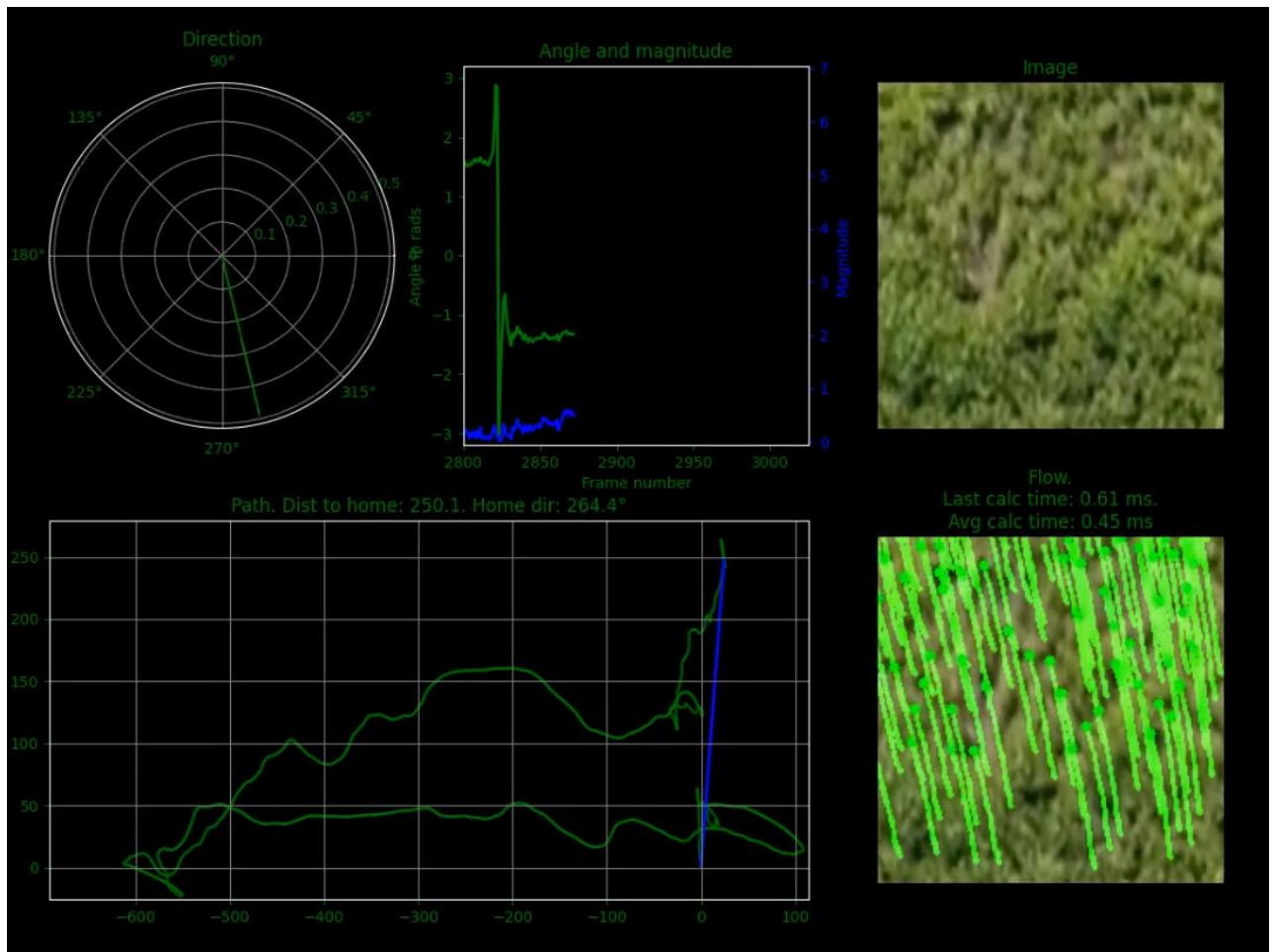


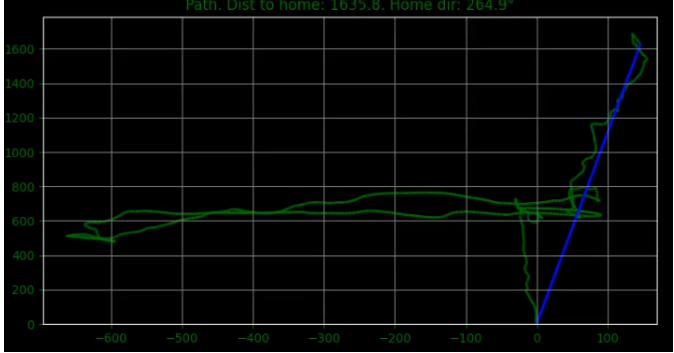

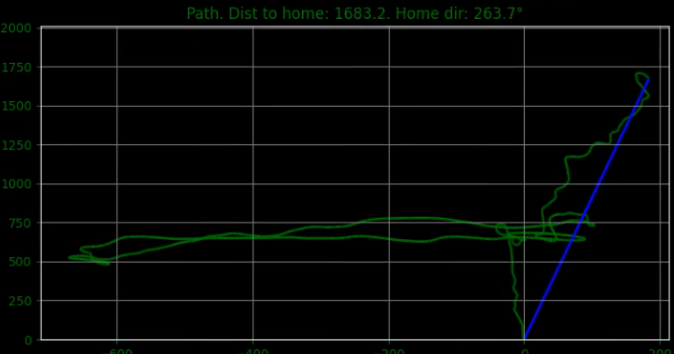
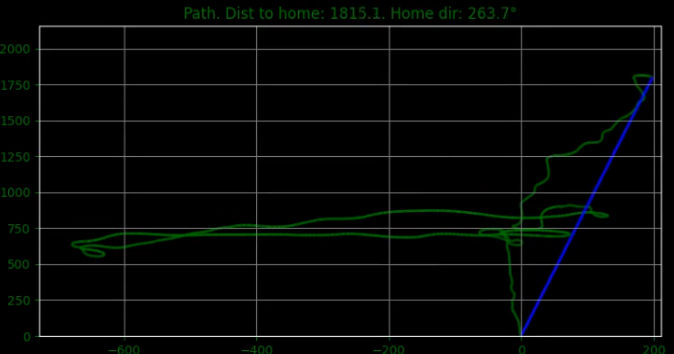
Рисунок 3.2 – Приклад виводу даних розроблено програмою для тестування оптичного потоку

Для порівняння алгоритмів обчислення оптичного потоку було порівняно відстані до точки зльоту апарату, обчислені оптичним потоком. Таке порівняння дасть розуміння того, який метод буде краще будувати пройдений маршрут відносно іншого. Порівнюватись буде загальна фігура маршруту, враховуючи різні коефіцієнти згладжування та фільтрації.

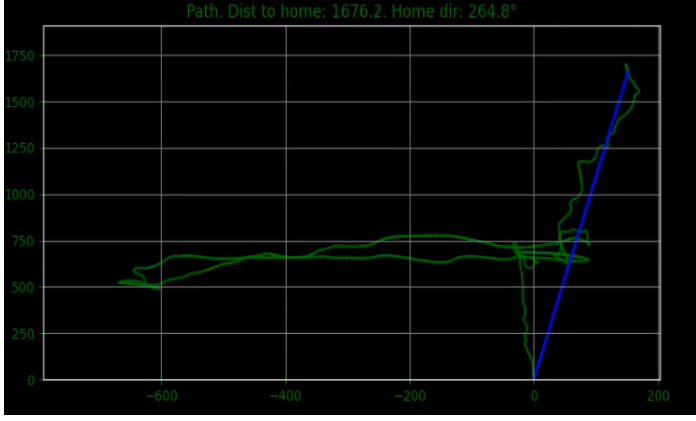
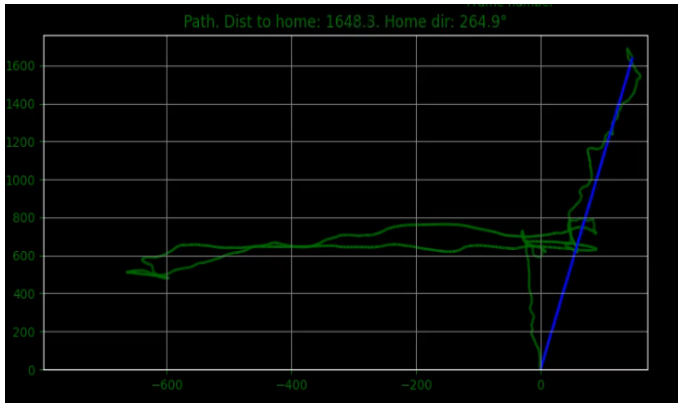
Якщо побудований маршрут є схожим до схематичного, отже, можна впевнено сказати про ефективність застосованого алгоритму обчислення оптичного потоку.

У таблиці 3.1 виведено результати згладжування та фільтрації для щільного алгоритму оптичного потоку.

Таблиця 3.1 – Результати згладжування та фільтрації для щільного алгоритму

Номер	Маршрут	Коефіцієнт фільтрації	Коефіцієнт згладжування
0		Без фільтрації	Без згладжування
1		0,2	0,2
2		0,4	0,2
3		0,1	0,1

Кінець таблиці 3.1



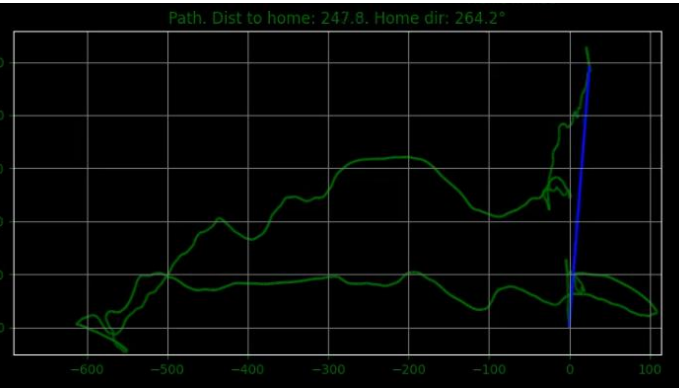
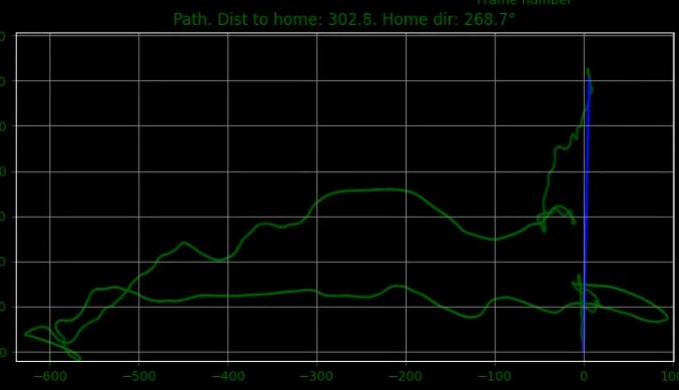
Номер	Маршрут	Коефіцієнт фільтрації	Коефіцієнт згладжування
4		0.1	0.4
5		0.1	0.6

Як можна бачити, щільний алгоритм уже достатньо точно рисує пройдений маршрут, незалежно від величини фільтрації, чи згладжування. Попри спотворені відстані, загальна форма маршруту відповідає дійсності.

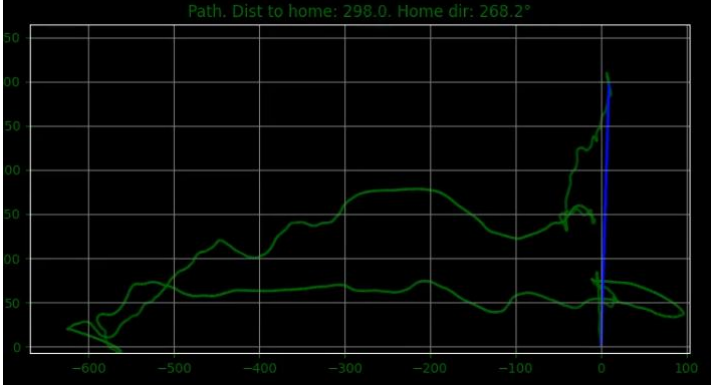


Аналогічно до тестування щільного алгоритму було проведено тестування розрідженого підходу для обчислення пройденого маршруту з різними коефіцієнтами згладжування та фільтрації.

У таблиці 3.2 виведено результати згладжування та фільтрації для розрідженого алгоритму оптичного потоку.

Таблиця 3.2 – Результати згладжування та фільтрації для розрідженого алгоритму

Номер	Маршрут	Коефіцієнт фільтрації	Коефіцієнт згладжування
0		Без фільтрації	Без згладжування
1		0,1	0,1
2		0,1	0,2
3		0,1	0,4

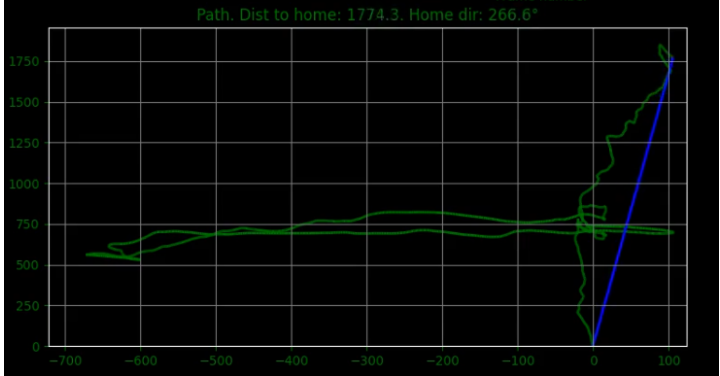
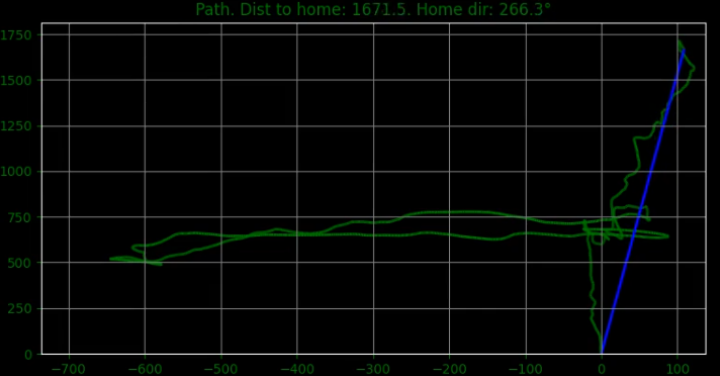
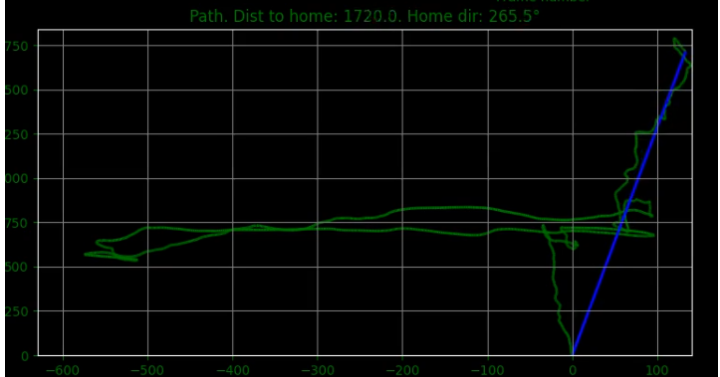
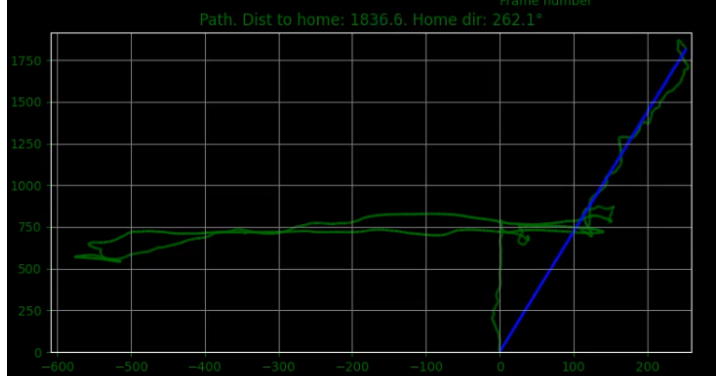
Кінець таблиці 3.2

Номер	Маршрут	Коефіцієнт фільтрації	Коефіцієнт згладжування
4		0,1	0,6
5		0,2	0,2
6		0,4	0,2

Алгоритм оптичного потоку RAFT було протестовано на 4 моделях без використання фільтрації, чи згладжування, адже очікується, що дана модель є достатньо стійкою до шуму.

У таблиці 3.3 виведено результати побудови маршруту для різних моделей методу RAFT.

Таблиця 3.3 – Результати побудови маршруту для різних моделей методу RAFT

Номер	Маршрут	Назва моделі
1	 <p>Path. Dist to home: 1774.3. Home dir: 266.6°</p> <p>The plot shows a green path starting at approximately (-650, 550) and ending at (100, 1750). The path is highly oscillatory, with many loops and turns. A blue line represents the direct path from the start to the end point. The x-axis ranges from -700 to 100, and the y-axis ranges from 0 to 1750.</p>	Kitty
2	 <p>Path. Dist to home: 1671.5. Home dir: 266.3°</p> <p>The plot shows a green path starting at approximately (-650, 550) and ending at (100, 1750). The path is highly oscillatory, with many loops and turns. A blue line represents the direct path from the start to the end point. The x-axis ranges from -700 to 100, and the y-axis ranges from 0 to 1750.</p>	Sintel
3	 <p>Path. Dist to home: 1720.0. Home dir: 265.5°</p> <p>The plot shows a green path starting at approximately (-550, 600) and ending at (100, 750). The path is highly oscillatory, with many loops and turns. A blue line represents the direct path from the start to the end point. The x-axis ranges from -600 to 100, and the y-axis ranges from 0 to 750.</p>	Small
4	 <p>Path. Dist to home: 1836.6. Home dir: 262.1°</p> <p>The plot shows a green path starting at approximately (-550, 600) and ending at (200, 1750). The path is highly oscillatory, with many loops and turns. A blue line represents the direct path from the start to the end point. The x-axis ranges from -600 to 200, and the y-axis ranges from 0 to 1750.</p>	Things

Як можна бачити, згладжування та фільтрація майже не мали вплив на побудований маршрут, проте найбільший вплив на пройдену відстань по прямій було в розрідженому алгоритмі. Такий на перший погляд випадковий розподіл відстаней можна пояснити тим, що навіть невеликі коригування в маршрут можуть потенційно мати великий вплив на пройдену відстань.

Підхід RAFT продемонстрував достатньо точний маршрут, близький до щільного алгоритму, на якому він і заснований.

Найгірший маршрут був побудований розрідженим алгоритмом. Даний алгоритм, попри фільтрацію та згладжування, видавав хаотичні рухи, які у свою чергу мали значний вплив на форму побудованого маршруту та пройдену дистанцію по прямій.

Друге тестування було проведено без фільтрації та згладжування, змінюючи параметри застосування розрідженого та щільного алгоритмів.

Бібліотека OpenCV має імплементацію щільного алгоритму Фарнебека у вигляді функції `calcOpticalFlowFarneback`, яка приймає наступні параметри:

- 1) `pyr_scale` – визначає масштаб піраміди, тобто те, у скільки буде зменшено кожне наступне зображення;
- 2) `levels` – кількість рівнів піраміди зображень;
- 3) `winsize` – розмір квадрату, для якого обчислюється оптичний потік за одну ітерацію.

Імплементація розрідженого алгоритму реалізована функцією `calcOpticalFlowPyrLK`, яка приймає наступні параметри:

- 1) `winSize` – розмір вікна пошуку на кожному рівні піраміди;
- 2) `maxLevel` - максимальний рівень піраміди;
- 3) `criteria` – параметр, що визначає критерії завершення ітераційного алгоритму пошуку (після заданої максимальної кількості ітерацій `criteria.maxCount` або коли вікно пошуку зміщується менше ніж на `criteria.epsilon`).

Друге тестування складається з відео руху камери по горизонталі на висоті 10 см. Максимальна відстань становить 99 см, кінцева відстань – 57,5 см. Оскільки алгоритми оптичного потоку не можуть видавати точні значення пройдені

відстані, порівняння будуть відбуватись як відношення максимальної відстані до кінцевої. Коефіцієнт повної відстані та відстані по закінченню: 99 см / 57,5 см \approx 1,722. На рисунку 3.3 зображено приклад виводу програмного забезпечення для другого тестування оптичного потоку.

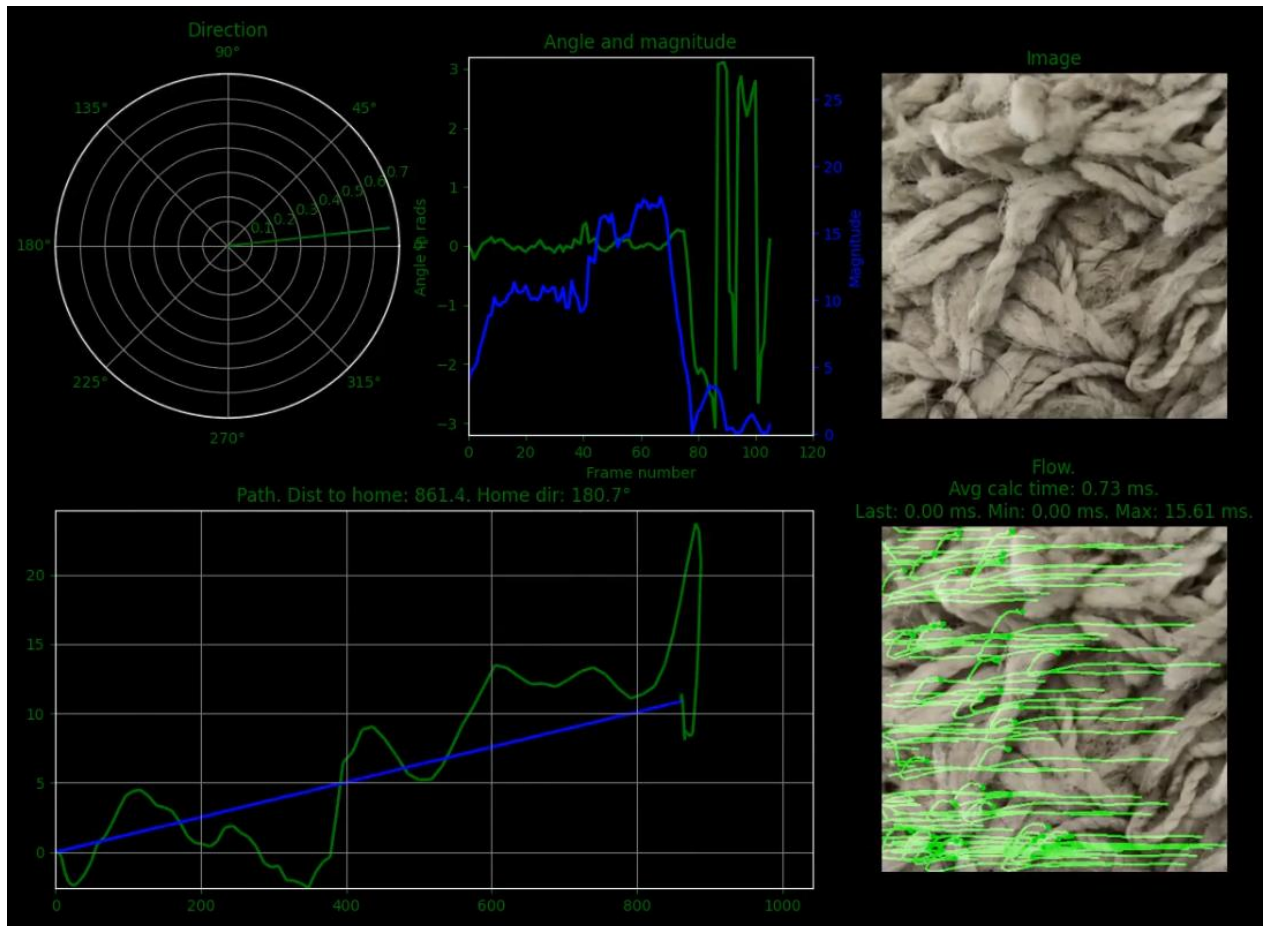


Рисунок 3.3 – Приклад виводу даних розроблено програмою для тестування оптичного потоку

У таблиці 3.4 виведено результати тестування, які включають розрахований коефіцієнт відстані, мінімальний, максимальний та середній час обчислення, а також деякі параметри функцій, що змінювались для щільного підходу обчислення оптичного потоку.

Таблиця 3.4 – Результати другого тесту для щільного алгоритму

Номер	Коефіцієнт відстані	Мін. час обчислення	Макс. час обчислення	Серед. час обчислення	pur_scale	levels	winsize
1	1,06	16,26 мс	122,78 мс	18,44 мс	0,7	1	300
2	1,04	14,51 мс	39,72 мс	20,15 мс	0,2	1	300
3	1,065	14,52 мс	56,17 мс	21,14 мс	0,5	3	300
4	1,058	14,49 мс	43,22 мс	17,34 мс	0,5	5	300
5	1,04	15,01 мс	53,02 мс	26,14 мс	0,5	1	200
6	1,066	14,51 мс	19,3 мс	19,39 мс	0,5	1	50

У таблиці 3.5 виведено результати другого тестування зі зміною розміру вікна та критеріїв розрідженого підходу обчислення оптичного потоку.

Таблиця 3.5 – Результати другого тесту для розрідженого алгоритму

№	Коефіцієнт відстані	Мін. час обчислення	Макс. час обчислення	Серед. час обчислення	win_size	maxlevel	criteria.maxCount, criteria.epsilon
1	1,68	0,009 мс	15,61 мс	0,56 мс	(30, 30)	2	30, 0,03
2	1,71	0,009 мс	6,9 мс	1,49 мс	(60, 60)	2	30, 0,03
3	1,63	0,51 мс	10,9 мс	3,33 мс	(120, 120)	2	30, 0,03

Кінець таблиці 3.5

№	Коефіцієнт відстані	Мін. час обчислення	Макс. час обчислення	Серед. час обчислення	win_size	maxlevel	criteria.maxCount, criteria.epsilon
4	1,74	0,009 мс	2,01 мс	0,27 мс	(15, 15)	0	30, 0,03
5	1,72	0,009 мс	2,03 мс	0,54 мс	(15, 15)	4	30, 0,03
6	1,72	0,009 мс	2,89 мс	0,54 мс	(15, 15)	8	30, 0,03
7	1,74	0,009 мс	2,05 мс	0,41 мс	(15, 15)	2	10, 0,03
8	1,67	0,009 мс	2,01 мс	0,41 мс	(15, 15)	2	60, 0,03
9	1,72	0,009 мс	2 мс	0,43 мс	(15, 15)	2	120, 0,03
10	1,74	0,009 мс	2,05 мс	0,41 мс	(15, 15)	2	30, 0,01
11	1,77	0,009 мс	1,51 мс	0,37 мс	(15, 15)	2	30, 0,06
12	1,76	0,009 мс	2 мс	0,41 мс	(15, 15)	2	30, 0,12

У таблиці 3.6 виведено результати другого тесту для методу обчислення оптичного потоку RAFT. Тут використано різні моделі для порівняння.

Таблиця 3.6 – Результати другого тесту для методу RAFT

Номер	Коефіцієнт відстані	Мін. час обчислення	Макс. час обчислення	Серед. час обчислення	Назва моделі
1	1,75	31,78 мс	269,99 мс	53,33 мс	Chairs
2	1,77	46,91 мс	66 мс	53,63 мс	Kitti
3	1,77	53,93 мс	46,97 мс	53,93 мс	Sintel
4	1,74	24 мс	88,21 мс	39,89 мс	Small
5	1,77	46,92 мс	130,4 мс	53,60 мс	Things

Найкраще себе показали розріджений та RAFT алгоритми. Для розрідженого алгоритму було визначено, що найбільший вплив на точність визначення пройденої відстані та часу виконання алгоритму впливає розмір вікна та збільшення кількості рівнів піраміди.

Для RAFT найкраще себе показала зменшена модель, яка в середньому в 1,3 рази швидше виконує алгоритм та показує точніший результат.

Щільний алгоритм показав найгірший результат в цьому тестуванні, пропри зміни в параметрах.

Найкращим варіантом для імплементації оптичного потоку буде саме RAFT зі зменшеною моделлю виконання. Попри відносно високий час обчислення, даний метод показав себе найбільш стабільним у двох тестах. Час виконання алгоритму можна покращити, використавши спеціальні обчислювальні модулі NPU, чи TPU.

3.2 Метод вирішення задачі навігації БПЛА в умовах роботи засобів РЕБ противника

Проаналізувавши різноманітні підходи до навігації БПЛА в умовах, де відсутній GNSS-сигнал, пропонується метод позиціонування, який базуватиметься на покращеному способі обчислення оптичного потоку на основі моделі RAFT. Очікується, що такий метод продемонструє кращі результати за наявні альтернативи позиціонування з використанням оптичного потоку щільним та

розрідженим алгоритмами. Для передбачення та оцінки стану системи в певний момент часу пропонується використання розширеного фільтру Калмана.

Метод навігації БПЛА в умовах роботи засобів РЕБ противника складається з трьох циклічно повторюваних основних етапів:

1. Збір даних з наступних датчиків: акселерометр, гіроскоп, лідар, барометр, магнітометр та датчик оптичного потоку на основі моделі штучного інтелекту RAFT.

2. Визначення поточного стану системи та передбачення наступного з допомогою розширеного фільтру Калмана. Коригування шумів з датчиків.

3. Побудова та збереження пройденого маршруту відносно точки зльоту.

Графічне представлення послідовності реалізації запропонованого методу зображено на рисунку 3.4.

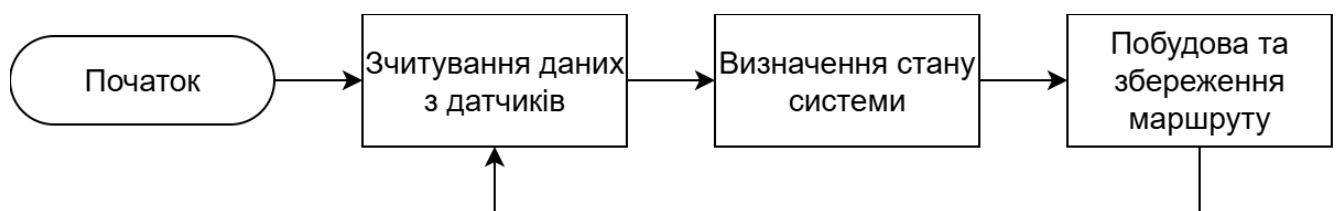


Рисунок 3.4 – Послідовність реалізації запропонованого методу позиціонування

Загальний алгоритм збору даних з датчиків складається з послідовних операцій зчитування даних з датчиків акселерометра, гіроскопа, магнітометра, барометра, лідара та датчика оптичного потоку. Оскільки побудована модель системи передбачає лише одне джерело даних про висоту апарату, основним джерелом даних про висоту буде лідар. Це пояснюється його точністю вимірювання відносно рельєфу, що важливо для датчика оптичного потоку. Барометр буде виконувати роль заміни лідара у випадках, якщо той перестане видавати коректні дані, що може бути наслідком помилкових вимірів через попадання в датчик світла різної інтенсивності, або ж підняття на висоту, вище якою не передбачено коректну роботу датчика.

На рисунку 3.5 зображено загальну блок-схему алгоритму роботи першої частини методу навігації БПЛА.

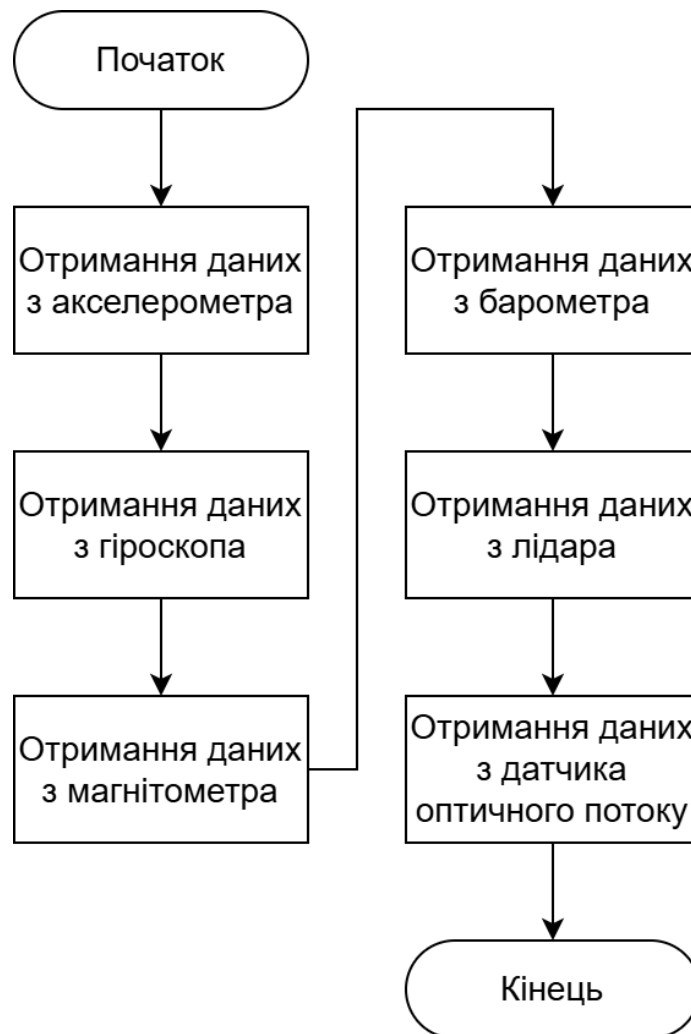


Рисунок 3.5 – Алгоритм збору даних з датчиків

Особливу увагу приділено обробці даних з датчика оптичного потоку, який реалізовано на основі моделі RAFT. На відміну від класичних методів обчислення оптичного потоку, запропонований підхід забезпечує більш точне обчислення оптичного потоку між послідовними кадрами завдяки використанню рекурентної моделі. Це дає кращу оцінку відносного переміщення БПЛА навіть в умовах поганого освітлення, або за наявності шумів у зображенні.

Отримані значення векторів переміщення оптичного потоку перетворюються в полярні координати, де кожен вектор представляється кутом та магнітудою.

Обчислюється середнє значення кутів та магнітуд векторів, яке видається як загальний напрямок відносного руху БПЛА з датчика оптичного потоку.

На рисунку 3.6 зображено алгоритм перетворення оптичного потоку в напрямок руху БПЛА.



Рисунок 3.6 – Блок-схема алгоритму перетворення оптичного потоку в напрямок руху БПЛА

Також потрібно враховувати, що з давачами можуть траплятися різного роду несправності, через що вони перестануть надавати будь-які дані. В такому випадку потрібно використовувати останні виміри з датчиків, поки не надійдуть нові.

Більш деталізований загальний алгоритм роботи збору даних для кожного давача зображено на рисунку 3.7.

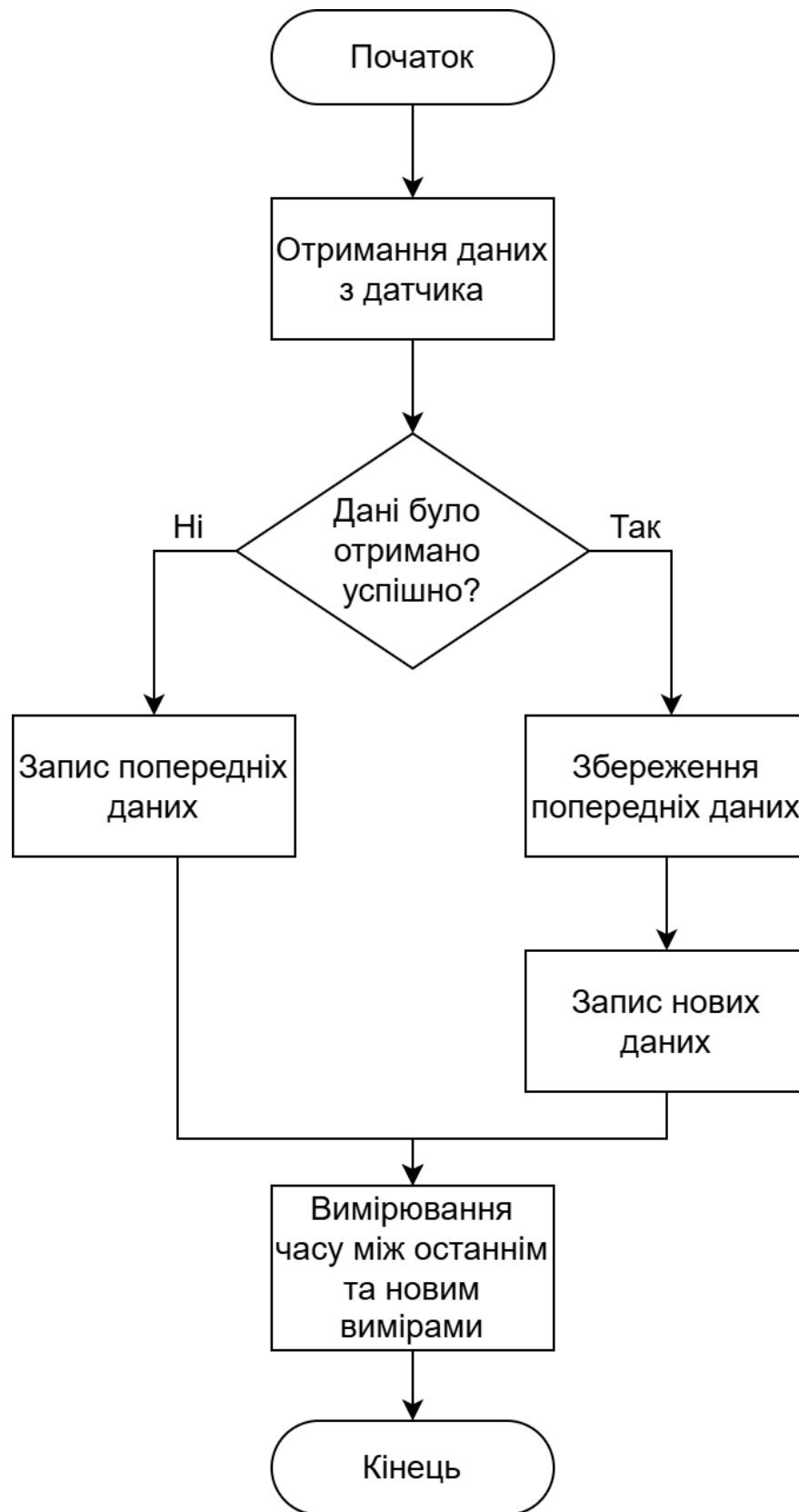


Рисунок 3.7 – Алгоритм обробки даних з давачів

3.3 Алгоритм фільтрації та обчислення стану системи

Оскільки обраним методом фільтрації даних з датчиків є розширений фільтр Калмана, потрібно розробити алгоритм для визначення поточного стану системи на основі отриманих даних з датчиків.

Першим етапом буде ініціалізація початкового стану, який із самого початку буде рівним нулю, адже ми вважаємо, що дрон завжди стартує з нульової координати.

На другому етапі відбувається перетворення векторів прискорення та швидкості з локальної системи координат у глобальну.

Третій етап передбачає стан системи в час t відповідно до вимірів з датчиків в час $t-1$. Для цього беруться дані з датчиків та вимірюється час Δt між попередніми та поточними вимірами.

На наступному етапі відбувається обчислення оцінки прогнозованої коваріації. Тобто оцінка залежності одних параметрів стану системи над іншими.

П'ятий етап обчислює відхилення між реальними та прогнозованими вимірами. Для початку обчислюються прогнозовані виміри з датчиків з спостережуваної моделі.

На шостому етапі обчислюється коваріація оновлення вимірів. Дана коваріація показує залежність передбачуваних вимірів від поточного стану системи.

Сьомий етап обчислює близько-оптимальний коефіцієнт Калмана, з якого потім обчислюється оновлена оцінка стану системи на восьмому етапі.

На дев'ятому етапі обчислюється нова оцінка прогнозованої коваріації. Дев'ятий етап не є заключним, адже усі вимірювання циклічно повторюються з другого етапу. Ініціалізація вже не потрібна, адже попередній стан системи уже обчислений.

Блок-схема алгоритму фільтрації та обчислення стану системи зображена на рисунку 3.8.

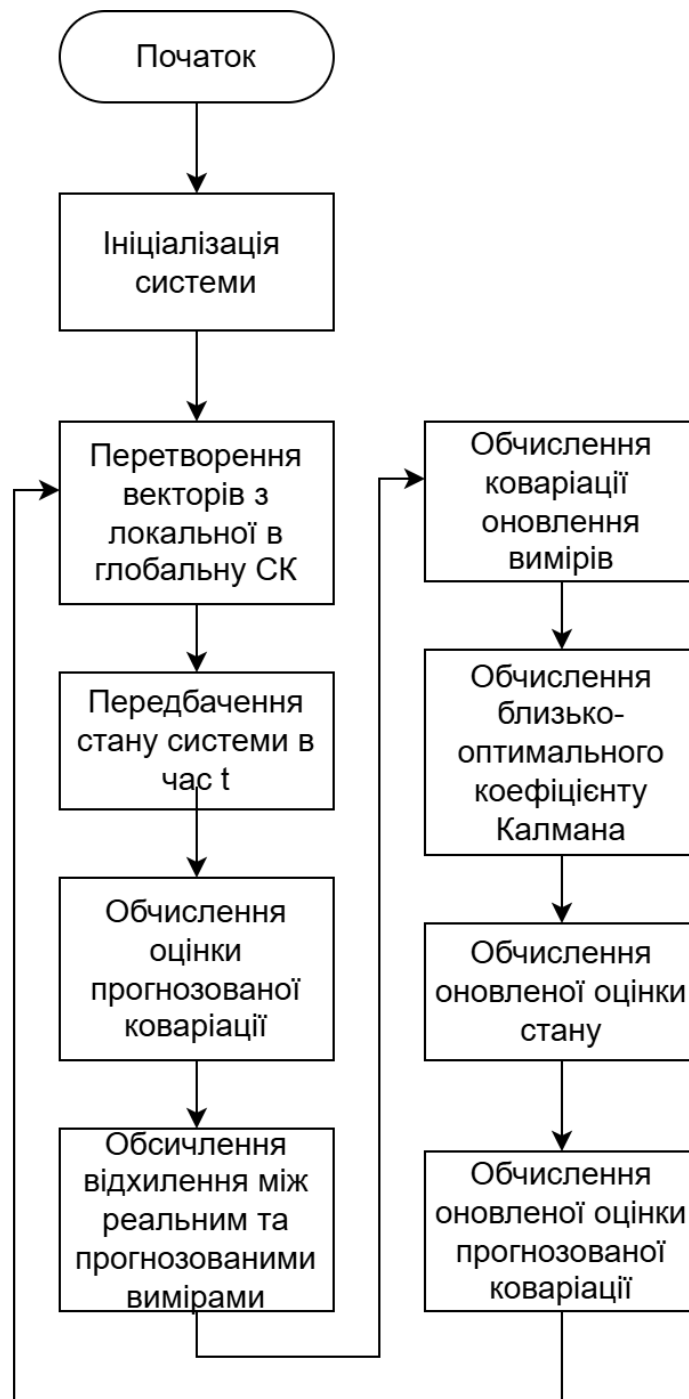


Рисунок 3.8 – Блок-схема алгоритму обчислення стану системи

3.4 Алгоритм побудови пройденого маршруту та його збереження

Отримавши дані про стан системи в час t , можна обчислити пройдений маршрут та напрямок повернення до точки зльоту. З отриманих координат будуватиметься пройдений маршрут, а також обчислюватиметься вектор

повернення до точки зльоту. Інформація про побудований маршрут та напрямок повернення буде виводити на екран у вигляді графіку.

Маршрутом повернення є вектор з початком у кінцевій точці пройденого маршруту та кінцем в точці зльоту апарату. За замовчуванням точкою зльоту апарату є точка з координатами (0,0). Отже, можна обчислити довжину та напрямок руху дрона для повернення до точки зльоту. писано Обчислення азимута повернення дрона до точки зльоту відбувається за наступною формулою:

$$a = \text{atan2}(y_n - y_0, x_n - x_0) \quad (3.2)$$

де a – азимут повернення дрона до точки зльоту;

y_0 – точка початку маршруту по осі Y;

y_n – кінцева точка маршруту по осі Y;

x_0 – точка початку маршруту по осі X;

x_n – кінцева точка маршруту по осі X.

Відстань маршруту повернення буде обчислюватись за формулою:

$$m = \sqrt{(y_n - y_0)^2 + (x_n - x_0)^2} \quad (3.3)$$

де m – відстань маршруту повернення.

Алгоритм роботи побудови пройденого маршруту та маршруту повернення дрона складається з додавання нової точки в загальний маршрут, його збереження та обчислення вектора напрямку руху до точки зльоту.

Кожна нова точка береться з попереднього алгоритму обчислення стану системи. Щоразу, як з'являється нова координата, пройдений маршрут та маршрут повернення оновлюються та зберігаються.

На рисунку 3.9 зображено блок-схему алгоритму роботи побудови пройденого маршруту дроном.

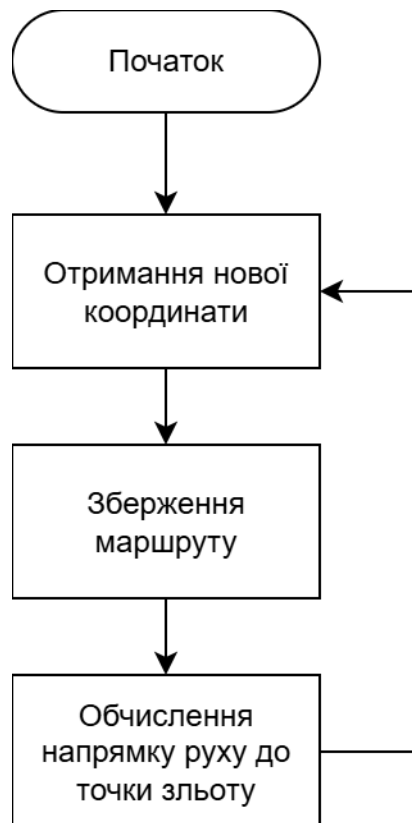


Рисунок 3.9 – Блок-схема алгоритму роботи побудови пройденого маршруту

3.5 Висновки до третього розділу

У цьому розділі було представлено розроблюваний метод навігації БПЛА в умовах застосування засобів РЕБ противника, який включає в себе збір даних за таких датчиків:

- 1) акселерометр;
- 2) гіроскоп;
- 3) барометр;
- 4) лідар;
- 5) магнітометр;
- 6) модуль оптичного потоку.

Основою методу є збір обробка зображень з камери для обчислення оптичного потоку, що дає змогу отримувати напрямок та швидкість руху апарату. Уточнення позиції відбувається за рахунок допоміжних датчиків. Фільтрація шумів

відбувається за допомогою розширеного фільтру Калмана, математична модель якого була описана в другому розділі.

Розроблено алгоритми роботи програмного забезпечення інтелектуальної кіберфізичної системи навігації БПЛА в умовах роботи засобів РЕБ противника. Описано алгоритм збору даних з датчиків, алгоритм фільтрації шумів, а також алгоритм побудови та збереження пройденого маршруту.

Описано метод збору та перетворення даних з датчика оптичного потоку, а також метод обчислення пройденого маршруту та маршруту повернення БПЛА.

Запропонований метод може бути використаний у системах автономного керування БПЛА для виконання завдань в умовах роботи радіоелектронної протидії, зокрема під час розвідки, моніторингу територій та виконання спеціальних операцій.

4 РЕЗУЛЬТАТИ РОБОТИ ІНТЕЛЕКТУАЛЬНОЇ КІБЕРФІЗИЧНОЇ СИСТЕМИ ДЛЯ НАВІГАЦІЇ БПЛА В УМОВАХ РОБОТИ ЗАСОБІВ РЕБ ПРОТИВНИКА

4.1 Аналіз вимог до розробки інтелектуальної кіберфізичної системи навігації БПЛА в умовах роботи засобів РЕБ противника

Метою розробки інтелектуальної кіберфізичної системи навігації БПЛА в умовах роботи засобів РЕБ противника є збереження повної або часткової автономності апарату, а також можливість автоматичного повернення до точки зльоту.

Дана система має виконувати наступні функції:

- обчислення пройденої відстані відносно точки зльоту;
- визначення напрямку слідування відповідно до магнітного азимута Землі;
- обчислення відносних координат відповідно до точки зльоту;
- визначення абсолютної та відносної висоти на рівнем моря та рівнем землі відповідно;
- обчислення та збереження пройденого маршруту відносно точки зльоту;
- обчислення кута спрямування та відстані по прямій до точки зльоту з будь-якої точки пройденого маршруту.

Основними компонентами системи є наступні:

- акселерометр, що використовується для визначення прискорення по осям пристрою;
- гіроскоп, що використовується для визначення положення пристрою у просторі;
- магнітометр, що використовується для визначення абсолютного спрямування пристрою відносно магнітного поля Землі, тобто азимуту;

- барометр, що використовується для визначення абсолютної висоти над рівнем моря;
- лідар, що використовується для визначення висоти датчика над поверхнею;
- камера оптичного потоку, що використовується для визначення напрямку швидкості руху пристрою по двом осям.

Кожен з датчиків допомагає будувати пройдений маршрут апаратом та визначати його відносне положення в реальному часі. Можливість системи більш точно визначати відносне положення пристрою дає кращий результат при побудові маршруту повернення до точки зльоту. Маршрут повернення до точки зльоту представляє собою пряму лінію певного напрямку та довжини, яку потрібно пройти дрону, щоб дістатися назад.

Основною вимогою системи є побудова прямого маршруту до точки зльоту в будь-який момент час польоту. Щоб перевірити дану вимогу потрібно порівняти різницю побудованого маршруту системою з реальним. Чим менша похибка, тим краще система відповідає вимозі.

Відповідно до поставленої задачі було сформульовано функційні та нефункційні вимоги системи в таблиці 4.1.

Таблиця 4.1 – Функційні та нефункційні вимоги до кіберфізичної системи навігації БПЛА в умовах роботи засобів РЕБ противника

Тип вимоги	Опис
Функційна	Побудова пройденого маршруту дрона
Функційна	Збереження пройденого маршруту дронам
Функційна	Вивід інформації про пройдений маршрут дронам
Функційна	Обчислення прямої повернення дрона до точки зльоту

Кінець таблиці 4.1

Тип вимоги	Опис
Функційна	Система повинна обчислювати маршрут борта в реальному часі
Нефункційна	Точність відносного позиціонування повинне складати не більше 5% відхилення від висоти по-горизонталі.
Нефункційна	У разі відмови одного, або кількох датчиків, система повинна запобігти дезорієнтації борта

4.2 Розробка архітектури інтелектуальної кіберфізичної системи навігації БПЛА в умовах роботи засобів РЕБ противника

Зважаючи на вимоги розроблюваної системи позиціонування пропонується архітектура, яка включає в себе набір датчиків, включно з камерою оптичного потоку, одноплатний комп'ютер для обчислення оптичного потоку та обробки даних з датчиків. Уся система матиме в основі польотний контролер з програмним забезпеченням автопілоту Arducopter версії 4.6.2. На рисунку 4.1 зображено загальну структуру розроблюваної системи.

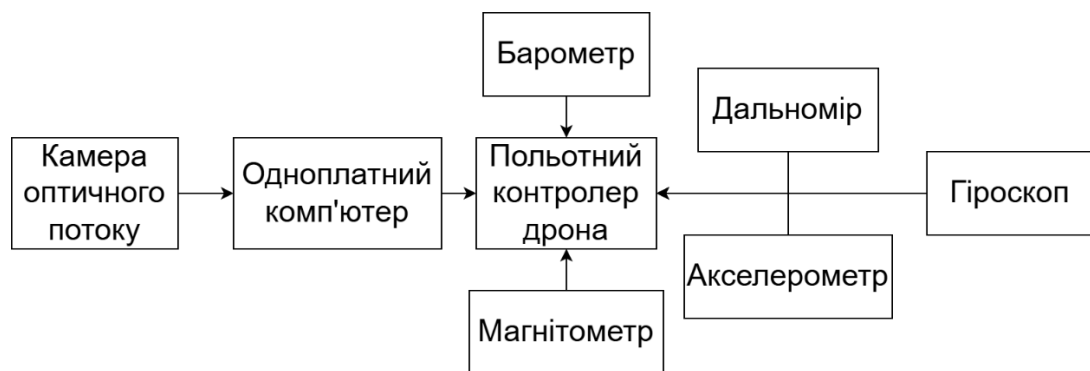


Рисунок 4.1 – Структура інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника

Розроблювана система має датчики на вході та блок оброблення даних, що включає як одноплатний комп'ютер, так і польотний контролер борта. На рисунку 4.2 зображено діаграму компонентів розроблюваної системи.

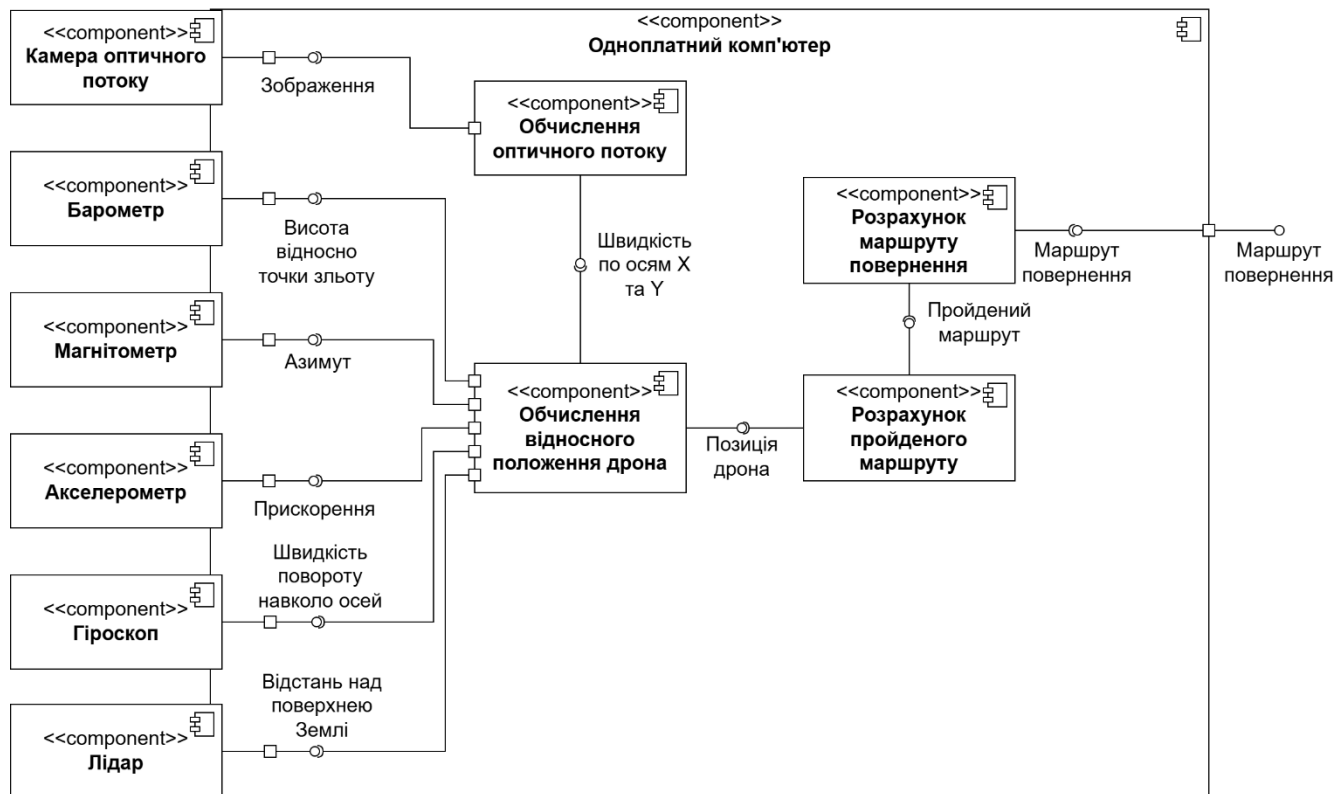


Рисунок 4.2 – Діаграма компонентів інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника

4.2 Проведення експерименту та аналіз результатів роботи програмного забезпечення інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника

Для тестування роботи розроблюваного програмного забезпечення кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника було створено та налаштовано квадрокоптер з усіма потрібними датчиками. Опис датчика та його назви наведено в таблиці 4.2.

Таблиця 4.2 – Датчики системи та їх опис

Назва датчика	Модель	Опис
Акселерометр	ICM-42605	3-осьовий акселерометр з низьким рівнем шуму та споживанням
Гіроскоп	ICM-42605	3-осовий гіроскоп з низьким рівнем шуму та споживанням
Барометр	DPS310	Високоточний барометр з низьким рівнем споживання струму
Лідар	PTGC-12X	Споживацький лідар з максимальною відстанню вимірювання 1200 м та частотою оновлення до 3 Гц
Магнітометр	PNI RM3100	Легкий та високоточний 3-осовий магнітометр
Камера оптичного потоку	DJI O4 Air Unit Pro	Легка та компактна камера з полем зору 155 градусів та максимальною роздільною здатністю 4K

Для проведення калібрування системи, а також експерименту пропонується спорядити борт GPS-модулем, який також буде окремо будувати пройдений маршрут, відносно якого буде виконуватись попереднє калібрування пройденої відстані датчиком оптичного потоку, а також будуть порівнюватись результати обох маршрутів.

Було здійснено політ борта з усіма датчиками, включно із GPS-модулем. Журнал польоту квадрокоптера буде використано як джерело даних з датчиків програмного забезпечення інтелектуальної кіберфізичної системи навігації БПЛА в умовах роботи засобів РЕБ противника.

Для аналізу та читання файлу-журналу борта використовується бібліотека `rutmavlog` мови програмування Python. Для обчислення розширеного фільтру Калмана використовується бібліотека `filterpy` мови програмування Python.

В якості борта для проведення експерименту було взято 18-дюймовий квадрокоптер виробництва ArmDrones, який зображено на рисунку 4.3.



Рисунок 4.3 – Квадрокоптер носій ArmDrones для проведення експерименту

Дрон було обладнано усіма потрібними датчиками, включно з GPS-модулем для порівняння пройденого маршруту та побудованого розробленою системою. Польотний контролер встановлено в центральній частині квадрокоптера та має вбудовані акселерометр та гіроскоп.

В якості GPS-модуля було взято Holybro M9N зі значенням кругового імовірного відхилення 1.5 м. У рамках експерименту можна знехтувати похибкою

GPS-модуля та вважати побудований ним маршрут як точний. Модуль було встановлено у верхній частині дрона, як зображено на рисунку 4.4.



Рисунок 4.4 – Модуль GPS, встановлений на квадрокоптері

Курсову камеру DJI O4 Pro встановлено на поворотному механізмі, який керується ШИМ-сигналом з польотного контролера. Камера може займати як горизонтальне, так і вертикальне положення відносно квадрокоптера. Курсову камеру зображено на рисунку 4.5.



Рисунок 4.5 – Курсова камера DJI O4 Pro

Лідар встановлено вертикально донизу в лівій передній частині квадрокоптера. Встановлений лідар зображено на рисунку 4.6.



Рисунок 4.6 – Встановлений лідар на квадрокоптері

Магнітометр встановлено у хвостовій частині квадрокоптера для зменшення електромагнітних завад з електроніки та моторів. Також допоміжний магнітометр міститься в модулі GPS. Розміщення основного магнітометра зображено на рисунку 4.7.

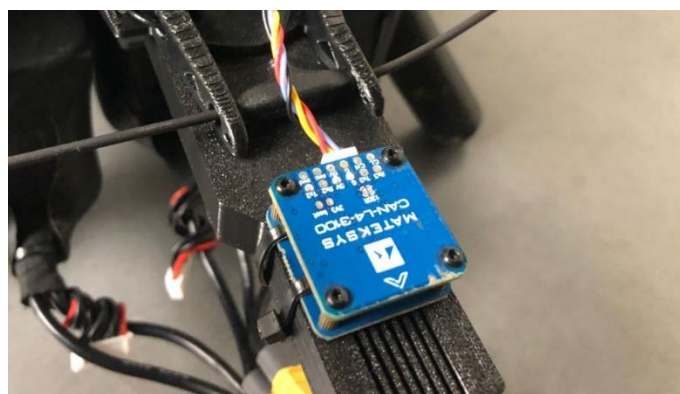


Рисунок 4.7 – Основний магнітометр, розміщений у хвостовій частині квадрокоптера

Магнітометри було калібровано засобами прошивки польотного контролера Ardupilot 4.6.2, а також проведено калібрування на завади з моторів дрона для збільшення точності роботи пристрою.

Було виконано простий маршрут польоту квадрокоптером із зависанням у точках зміни напрямку. Схему маршрут польоту квадрокоптера зображено на рисунку 4.8.

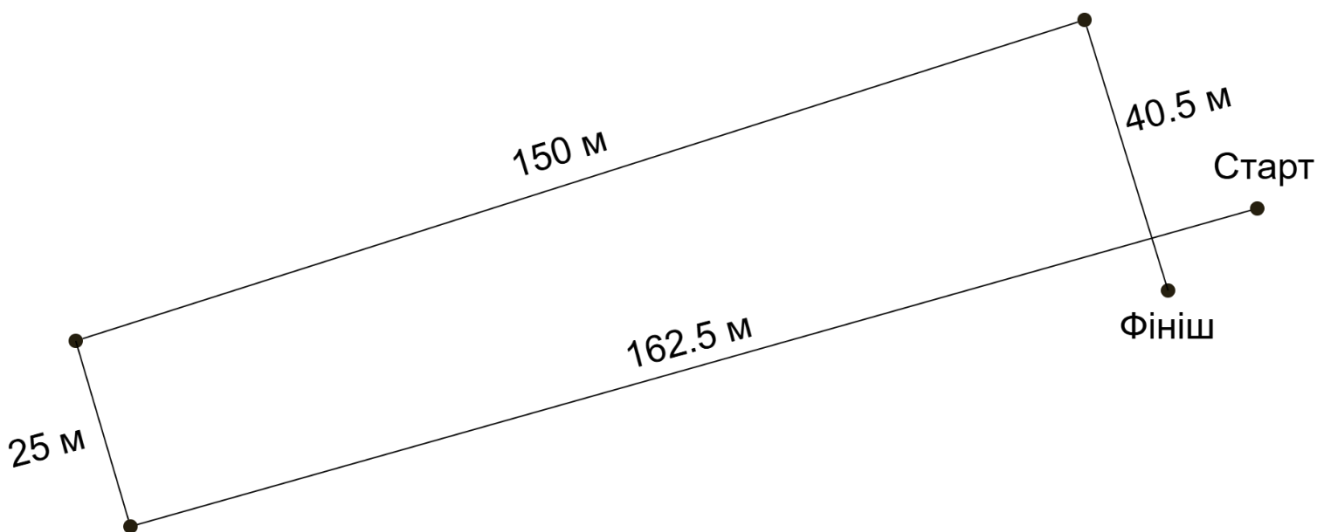


Рисунок 4.8 – Схематичне зображення пройденого маршруту

Журнал польоту та відео з курсової камери, направленої перпендикулярно донизу, було оброблено з урахуванням часових штампів усіх вимірів. Оскільки найбільшу частоту вимірювання мають саме акселерометр та гіроскоп, послідовно було зчитано значення з прив'язкою до часу роботи системи. Відносно часу зчитувань з акселерометра та гіроскопа було зчитано датчики: барометр, лідар та магнітометр. Оскільки лідар працює від 5 метрів, висоту до 6 метрів було отримано з барометра, а далі зчитування відбувались лише з лідара.

Було також проведено калібрування датчика оптичного потоку та визначено коефіцієнт висоти до пройденої відстані. Виміри з датчика оптичного потоку перетворювались за формулою 4.1.

$$m = m_{of} |h| * k, \quad (4.1)$$

де m – магнітуда вектору пройденої відстані;

m_{of} – магнітуда вектору отримана з датчика оптичного потоку;

$|h|$ – модуль висоти дрона над поверхнею Землі;

k – коефіцієнт висоти до пройденої відстані.

Визначено коефіцієнт висоти до пройденої відстані як $k = 0.00098$. Після було проведено фінальні обчислення з побудовою маршруту. Вигляд результату побудованого маршруту програмним забезпеченням зображено на рисунку 4.9.

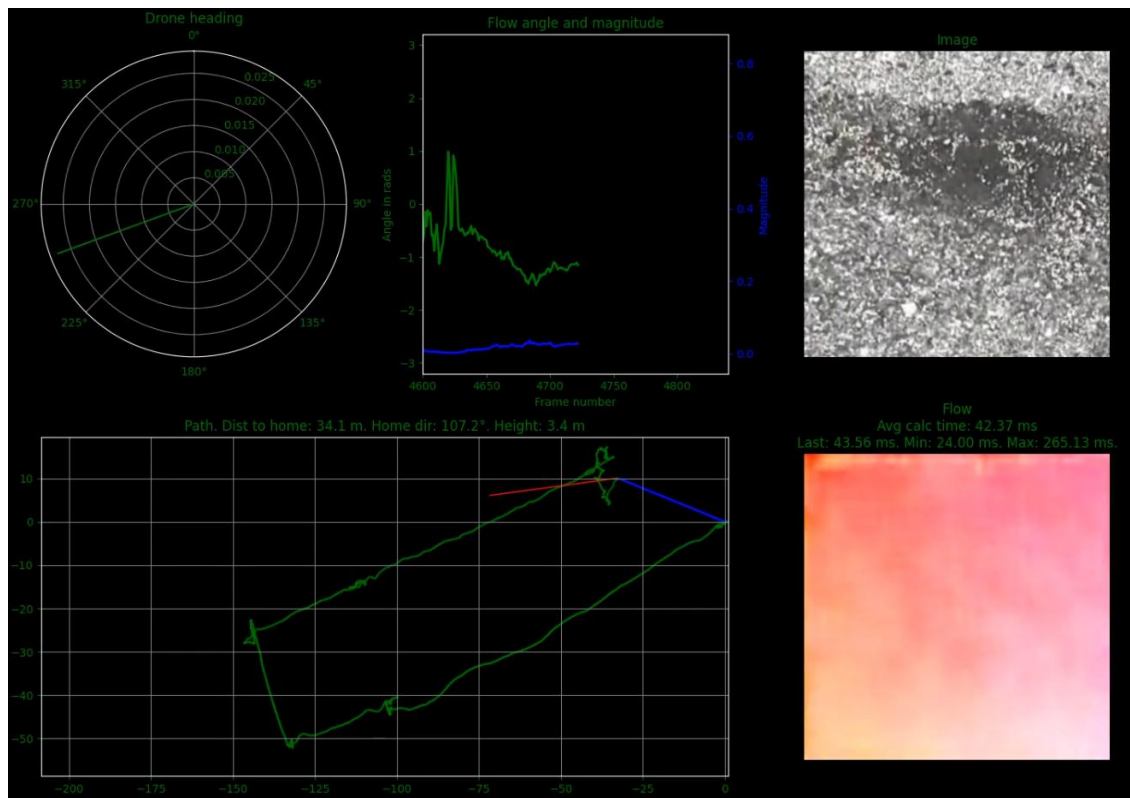


Рисунок 4.9 – Результати обчислення маршруту інтелектуальною кіберфізичною системою для навігації БПЛА в умовах дії засобів РЕБ противника

Також виведено додаткову інформацію про азимут спрямування дрона (ліворуч зверху), магнітуда та кут вектора з датчика оптичного потоку (по-центру зверху), шматок зображення з курсової камер, яке аналізувалось датчиком оптичного потоку (праворуч зверху) та зображення спектру напрямку обчисленого оптичного потоку (праворуч знизу).

Побудований маршрут кіберфізичною системою зображено на рисунку 4.10. Зеленим зображено пройдений маршрут, синій вектор позначає відстань та напрямок маршруту повернення до точки зльоту, червоним позначено азимут спрямування дрона.



Рисунок 4.10 – Побудований маршрут інтелектуальною кіберфізичною системою для навігації БПЛА в умовах дії засобів РЕБ противника

Схема побудованого маршруту інтелектуальною кіберфізичною системою навігації зображена на рисунку 4.11.

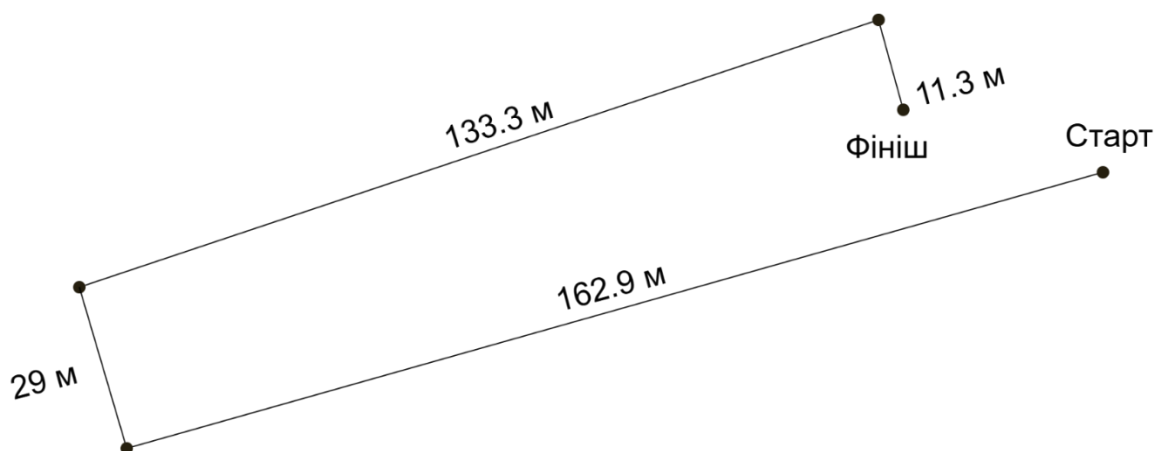


Рисунок 4.11 – Схема побудованого маршруту інтелектуальною кіберфізичною системою для навігації БПЛА в умовах дії засобів РЕБ противника

Як можна бачити, присутня певна похибка у вимірах відстані, яка може бути викликана відсутністю стабілізації камери оптичного потоку, неточним визначенням коефіцієнту пройденої відстані до висоти та похибкою самої моделі RAFT.

Відстань пройденого маршруту, побудованого GPS-модулем, становить 378 метрів. Відстань пройденого маршруту, побудованого інтелектуальною кіберфізичною системою навігації, становить 336,5 метрів.

Загальну точність пройденого маршруту можна розрахувати за формулою 4.2.

$$E = \frac{1}{d_{gps}} * |d_{gps} - d_{of}| * 100\% = 10.97\%, \quad (4.2)$$

де E – виміряна похибка системи у відсотках;

d_{gps} – відстань пройденого маршруту за модулем GPS;

d_{of} – відстань пройденого маршруту, обчисленого інтелектуальною кіберфізичною системою навігації.

За результатами проведеного експерименту значення похибки становить 10.97%, що є невеликим значенням. Це означає, що дрон з подібною навігацією схибить на приблизно 10% від пройденої відстані. Точність такої системи буде становити 89% від пройденої відстані, чого має бути достатньо для виконання простих місій із автоматичним поверненням до точки зльоту.

4.3 Висновки до четвертого розділу

У цьому розділі було проаналізовано вимоги до розробки та створення програмного забезпечення інтелектуальної кіберфізичної системи навігації для навігації БПЛА в умовах роботи засобів РЕБ противника. Проведено експеримент із використанням усіх датчиків, що входять в розроблювану систему. Експеримент виконувався шляхом читання файлу-журналу польоту борта та синхронного

аналізу відео польоту з камери оптичного потоку. За результатами проведеного експерименту було продемонстровано точність вимірної відстані пройденого маршруту в 89%. Така точність дозволяє використовувати систему в нескладних автоматичних місіях польотів БПЛА із можливістю автоматичного повернення до точки зльоту.

Розроблена система може бути адаптована під практично БПЛА будь-якого типу із потрібним набором датчиків та камерою оптичного потоку. Також можливе використання більш точного методу обчислення оптичного потоку та використання стабілізації відео, чи камери, для покращення вимірів пройденої відстані.

ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено метод та інтелектуальну кіберфізичну систему для навігації БПЛА в умовах дії засобів РЕБ противника. Подальшого розвитку набула система та метод об'єднання датчиків з використанням оптичного потоку з подальшою фільтрацією за допомогою методу розширеного фільтру Калмана.

За результатами роботи було проведено експеримент для визначення точності навігації розробленої інтелектуальної кіберфізичної системи. Визначена точність дозволяє використання даної системи для невеликих автономних місій польотів, де відсутній зв'язок GPS.

Метою кваліфікаційної роботи магістра є розробка методу та інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника.

Поставлену мету досягнуто шляхом розв'язання таких основних завдань:

- проаналізовано існуючі методи навігації та позиціонування в умовах відсутності GNSS-сигналу;
- обрано експериментальним шляхом найбільш підходящий метод обчислення оптичного потоку;
- розроблено математичну модель системи з використанням методу розширеного фільтру Калмана;
- розроблено метод та програмне забезпечення інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника;
- здійснено експеримент для перевірки розробленого методу навігації на основі розробленого програмного забезпечення інтелектуальної кіберфізичної системи навігації БПЛА. Результати експерименту показали точність вимірювання пройденого маршруту в 89% порівняно з виміряним маршрутом модулем GPS.

За темою кваліфікаційної роботи опубліковано одну публікацію [16] у міжнародному науково-технічному журналі «Вимірювальна та обчислювальна техніка в технологічних процесах». (Хмельницький – 2026). С. 28-39.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Global Positioning Systems Market (2025 - 2033). Режим доступу: <https://www.grandviewresearch.com/industry-analysis/gps-market> (дата звернення 01.03.2026).
2. FAQ on Global Navigation Satellite Systems (GNSS) Interference. Режим доступу: <https://www.itu.int/en/ITU-R/Documents/FAQs%20on%20GNSS%20Interference.pdf> (дата звернення 01.03.2026).
3. Jacobs F. GPS jamming, a weapon in hot and hybrid wars, will soon be obsolete. *Strange Maps*. 2025. 1269. Режим доступу: <https://bigthink.com/strange-maps/gnss-jamming/>.
4. Jarraya I., Al-Batati A., Kadri M.B. та ін. Gnss-denied unmanned aerial vehicle navigation: analyzing computational complexity, sensor fusion, and localization methodologies. *Satellite Navigation*. 2025. 6(9). Режим доступу: <https://doi.org/10.1186/s43020-025-00162-z>.
5. Lee J., Sung C-K., Oh J., Han K. та ін. A Pragmatic Approach to the Design of Advanced Precision Terrain-Aided Navigation for UAVs and Its Verification. *Remote Sensing*. *Remote Sens.* 2020. 12(9). 1396. Режим доступу: <https://doi.org/10.3390/rs12091396>.
6. Martin TM-76/MGM-13/CGM-13 Масе. Режим доступу: <https://www.designation-systems.net/dusrm/m-13.html> (дата звернення 03.03.2026).
7. Raković D., Simonović A., Grbović M. UAV Positioning and Navigation - Review. In: Mitrovic, N., Mladenovic, G., Mitrovic, A. (eds) Experimental and Computational Investigations in Engineering. *Springer, Cham*. 2020. С. 220-556. Режим доступу: https://doi.org/10.1007/978-3-030-58362-0_14.
8. Geoffrey B., James P. IMAGE PROCESSING FOR TOMAHAWK SCENE MATCHING. *Johns Hopkins APL Technical Digest*. 1994. 3(15). Режим доступу: <https://secwww.jhuapl.edu/techdigest/Content/techdigest/pdf/V15-N03/15-03-Irani.pdf>.

9. Jouko K., Riccardo R., Francesco V., Ville K. LSVL: Large-scale season-invariant visual localization for UAVs. *Robotics and Autonomous Systems*. 2023. 168. Режим доступу: <https://doi.org/10.1016/j.robot.2023.104497>.
10. Eduardo G., Antonio B.. Reduction of GNSS-Denied inertial navigation errors for fixed wing autonomous unmanned air vehicles. *Aerospace Science and Technology*. 2022. 120. Режим доступу: <https://doi.org/10.1016/j.ast.2021.107237>.
11. Feng W., Yang Z., Cheng Z., Joao B. та ін. UAV navigation in large-scale GPS-denied bridge environments using fiducial marker-corrected stereo visual-inertial localisation. *Automation in Construction*. 2023. 156. Режим доступу: <https://doi.org/10.1016/j.autcon.2023.105139>.
12. Deng J., Teed Z. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. *CoRR*. 2020. 2003.12039. Режим доступу: <https://doi.org/10.48550/arXiv.2003.12039>.
13. New aviation workgroup on spoofing – all invited. Режим доступу: <https://rntfnd.org/2024/07/17/aviation-workgroup-on-spoofing-all-invited/> (дата звернення 03.03.2026).
14. 65% Increase in GPS loss rate in 2024 – IATA. Режим доступу: <https://rntfnd.org/2024/12/08/65-increase-in-gps-loss-rate-in-2024-iata/> (дата звернення 03.03.2026).
15. Chiang B., Bohg J. Optical and Scene Flow. 2024. Режим доступу: https://web.stanford.edu/class/cs231a/course_notes/09-optical-flow.pdf
16. Березька К., Севостьянов В. ІНТЕЛЕКТУАЛЬНА КІБЕРФІЗИЧНА СИСТЕМА ДЛЯ НАВІГАЦІЇ БПЛА В УМОВАХ ДІЇ ЗАСОБІВ РЕБ ПРОТИВНИКА. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2026. 2. С. 28-39. Режим доступу: <https://doi.org/10.31891/2219-9365-2026-86-4>.
17. Chang Y., Cheng Y., Manzoor U., Murray J. A review of UAV autonomous navigation in GPS-denied environments. *Robotics and Autonomous Systems*. 2023. 170. 104533. Режим доступу: <https://doi.org/10.1016/j.robot.2023.104533>.

18. Jiang C., Zhou X., Chen H., Liu T. UAV Positioning Using GNSS: A Review of the Current Status. *Drones*. 2026. 10(2). 91. Режим доступа: <https://doi.org/10.3390/drones10020091>.
19. Senhaji-Mouhaddib S., Lagrat I. Advanced Autonomous Navigation Technologies for UAVs: Challenges and Opportunities. *4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*. 2024. С. 1-7. Режим доступа: <https://doi.org/10.1109/IRASET60544.2024.10548645>.
20. Mingliang Z., Xuezhi X., Ning L., Xiangdong K. Optical flow and scene flow estimation: A survey. *Pattern Recognition*. 2021. 114. Режим доступа: <https://doi.org/10.1016/j.patcog.2021.107861>.
21. Hui T. -W., Tang X., Loy C. A Lightweight Optical Flow CNN —Revisiting Data Fidelity and Regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2021. 8(43). С. 2555-2569. Режим доступа: <https://doi.org/10.1109/TPAMI.2020.2976928>.
22. Sun D., Herrmann C., Reda F., Rubinstein M. Та ін. Disentangling Architecture and Training for Optical Flow. *Springer, Cham*. 2022. 13682. Режим доступа: https://doi.org/10.1007/978-3-031-20047-2_10.
23. Jonschkowski R., Stone A., Barron J.T. та ін. Gordon A. What Matters in Unsupervised Optical Flow. *Springer, Cham*. 2020. 12347. Режим доступа: https://doi.org/10.1007/978-3-030-58536-5_33.
24. Haofei X., Jiaolong Y., Jianfei C., Juyong Z. та ін. High-Resolution Optical Flow From 1D Attention and Correlation. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021. С. 10498-10507. Режим доступа: <https://doi.org/10.48550/arXiv.2104.13918>.
25. Jisoo J., Jamie M.L., Fatih P., Nojun K. Imposing Consistency for Optical Flow Estimation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022. С. 3181-3191. Режим доступа: <https://doi.org/10.48550/arXiv.2204.07262>.
26. Hartley R., Ghaffari M., Eustice RM., Grizzle JW. Contact-aided invariant extended Kalman filtering for robot state estimation. *The International Journal of*

Robotics Research. 2020. 39(4). С. 402-430. Режим доступа: <https://doi.org/10.1177/0278364919894385>.

27. Madhukar P., Prasad L. State Estimation using Extended Kalman Filter and Unscented Kalman Filter. *2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*. 2020. С. 1-4. Режим доступа: <https://doi.org/10.1109/ICONC345789.2020.9117536>.

28. Odry Á., Kecskes I., Sarcevic P., Vizvari Z. та ін. A Novel Fuzzy-Adaptive Extended Kalman Filter for Real-Time Attitude Estimation of Mobile Robots. *Sensors*, 2020. 20(3). 803. Режим доступа: <https://doi.org/10.3390/s20030803>.

29. Yang B., Yang E., Yu L., Niu C. Adaptive Extended Kalman Filter-Based Fusion Approach for High-Precision UAV Positioning in Extremely Confined Environments. *IEEE/ASME Transactions on Mechatronics*. 2023. 1(28). С. 543-554. Режим доступа: <https://doi.org/10.1109/TMECH.2022.3203875>.

30. Tang X., Zhao F., Tang Z., Wang H. Nonlinear Extended Kalman Filter for Attitude Estimation of the Fixed-Wing UAV. *International Journal of Optics*. 2022. Режим доступа: <https://doi.org/10.1155/2022/7883851>.

31. Liu Y., Duan C., Liu L., Cao L. Discrete-Time Incremental Backstepping Control with Extended Kalman Filter for UAVs. *Electronics*. 2023. 12(14). Режим доступа: <https://doi.org/10.3390/electronics12143079>.

32. Aldosari W., Moinuddin M., Aljohani A. J., Al-Saggaf U. M. Distributed Extended Kalman Filtering Based Techniques for 3-D UAV Jamming Localization. *Sensors*. 2020. 20(22). 6405. Режим доступа: <https://doi.org/10.3390/s20226405>.

33. Xiaoyu Y., Fujun S., Zongyu Z., Rui Z., Qinghua Z. Semi-Aerodynamic Model-Aided Invariant Kalman Filtering for UAV Full-State Estimation. *IEEE Sensors Journal*. 2024. 16(24). С. 25920-25939. Режим доступа: <https://doi.org/10.1109/JSEN.2024.3414995>.

34. Dai Z., Jing L. Lightweight Extended Kalman Filter for MARG Sensors Attitude Estimation. *IEEE Sensors Journal*. 2021. 13(21). С. 14749-14758. Режим доступа: <https://doi.org/10.1109/JSEN.2021.3072887>.

35. Miao Hui W., Thompson M. Attitude Estimation of Four-Rotor UAV Based on Extended Kalman Filter. *International Journal of Engineering Continuity*. 2022. 1(2). С. 72–84. Режим доступа: <https://doi.org/10.58291/ijec.v1i2.52>.
36. Tiemann J., Schweikowski F., Wietfeld C. Design of an UWB indoor-positioning system for UAV navigation in GNSS-denied environments. *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 2015. С. 1-7. Режим доступа: <https://doi.org/10.1109/IPIN.2015.7346960>.
37. Dai H., Bian H., Wang R., Ma H. An INS/GNSS integrated navigation in GNSS denied environment using recurrent neural network. *Defence Technology*. 2020. 16(2). С. 334-340. Режим доступа: <https://doi.org/10.1016/j.dt.2019.08.011>.
38. Kassas Z. та ін. Aircraft Navigation in GNSS-Denied Environments via Radio SLAM With Terrestrial Signals of Opportunity. *IEEE Transactions on Intelligent Transportation Systems*. 2024. 10(25). С. 14164-14182. Режим доступа: <https://doi.org/10.1109/TITS.2024.3405908>.
39. Lin H.-Y., Zhan J.-R. GNSS-denied UAV indoor navigation with UWB incorporated visual inertial odometry. *Measurement*. 2023. 206.112256. Режим доступа: <https://doi.org/10.1016/j.measurement.2022.112256>.
40. Yang Y., Khalife J., Morales J. J., Kassas Z. M. UAV Waypoint Opportunistic Navigation in GNSS-Denied Environments. *IEEE Transactions on Aerospace and Electronic Systems*. 2022. 1(58). С. 663-678. Режим доступа: <https://doi.org/10.1109/TAES.2021.3103140>.
41. Luo H., Li G., Zou D., Li K., Li X., Yang Z. UAV Navigation With Monocular Visual Inertial Odometry Under GNSS-Denied Environment. *IEEE Transactions on Geoscience and Remote Sensing*. 2023. 61. С. 1-15. Режим доступа: <https://doi.org/10.1109/TGRS.2023.3323519>.
42. Li J., Yang G., Cai Q., Niu H., Li J. Cooperative navigation for UAVs in GNSS-denied area based on optimized belief propagation. *Measurement*. 2022. 192. 110797. Режим доступа: <https://doi.org/10.1016/j.measurement.2022.110797>.

43. Krystian B., Cezary S. GNSS denied navigation system for the manoeuvring flying objects. *Aircraft Engineering and Aerospace Technology*. 2023. 96(1). С. 63-72. Режим доступа: <https://doi.org/10.1108/AEAT-05-2023-0124>.
44. Hou X., Li Z., Pan Q. Autonomous navigation of a multirotor robot in GNSS-denied environments for search and rescue. *Sci China Inf Sci*. 2023. 66(3). Режим доступа: <https://doi.org/10.1007/s11432-020-3188-4>.
45. Abdelaziz N., El-Rabbany A. INS/LIDAR/Stereo SLAM Integration for Precision Navigation in GNSS-Denied Environments. *Sensors*. 2023. 23(17). 7424. Режим доступа: <https://doi.org/10.3390/s23177424>.
46. G. He, X. Yuan, Y. Zhuang, H. Hu. An Integrated GNSS/LiDAR-SLAM Pose Estimation Framework for Large-Scale Map Building in Partially GNSS-Denied Environments. *IEEE Transactions on Instrumentation and Measurement*. 2021. 70. С. 1-9. Режим доступа: <https://doi.org/10.1109/TIM.2020.3024405>.
47. Partsinevelos P., Chatziparaschis D., Trigkakis D., Tripolitsiotis A. A Novel UAV-Assisted Positioning System for GNSS-Denied Environments. *Remote Sensing*. 2020. 12(7). 1080. Режим доступа: <https://doi.org/10.3390/rs1207108>.
48. Zhang Q., Dai Y., Zhang T., Guo C., Niu X. Road Semantic-Enhanced Land Vehicle Integrated Navigation in GNSS Denied Environments. *IEEE Transactions on Intelligent Transportation Systems*. 2024. 12(25). С. 20889-20899. Режим доступа: <https://doi.org/10.1109/TITS.2024.3449892>.
49. Cui Y. та ін. A Novel Positioning Method for UAV in GNSS-Denied Environments Based on Mechanical Antenna. *IEEE Transactions on Industrial Electronics*. 2024. 10(71). С. 13461-13469. Режим доступа: <https://doi.org/10.1109/TIE.2023.3347859>.
50. Mugnai M. та ін. An Efficient Framework for Autonomous UAV Missions in Partially-Unknown GNSS-Denied Environments. *Drones*. 2023. 7(7). С. 471. Режим доступа: <https://doi.org/10.3390/drones7070471>.
51. Zhang H., та ін. A Real-Time Simulator for Navigation in GNSS-Denied Environments of UAV Swarms. *Applied Sciences*. 2023. 13(20). С. 11278. Режим доступа: <https://doi.org/10.3390/app132011278>.

52. Zhao RM. та ін. Rapid development methodology of agricultural robot navigation system working in GNSS-denied environment. *Adv. Manuf.* 2023. 11. С. 601–617. Режим доступу: <https://doi.org/10.1007/s40436-023-00438-0>.
53. Wang J. та ін. Navigation in GNSS-Denied Environment Based on Observation of RSOs and Stars. *Guidance, Navigation and Control.* 2024. 1(5). С. 101-113. Режим доступу: <https://doi.org/10.1142/S2737480724500225>.
54. H. Zhang, X. Xia, M. Nitsch, D. Abel. Continuous-Time Factor Graph Optimization for Trajectory Smoothness of GNSS/INS Navigation in Temporarily GNSS-Denied Environmentsю. *IEEE Robotics and Automation Letters.* 2022. 4(7). С. 9115-9122. Режим доступу: <https://doi.org/10.1109/LRA.2022.3189824>.
55. Akhiehiero D., Olawoye U., Das S., Gross J. Cooperative Localization for GNSS-Denied Subterranean Navigation: A UAV–UGV Team Approach. *Navigation.* 2024. 71(4). Режим доступу: <https://doi.org/10.33012/navi.677>.
56. Wang, Y. та ін. Angle Robustness Unmanned Aerial Vehicle Navigation in GNSS-Denied Scenarios. *Proceedings of the AAAI Conference on Artificial Intelligence.* 2024. 38(9). С. 10386-10394. Режим доступу: <https://doi.org/10.1609/aaai.v38i9.28906>.
57. S. Alghamdi та ін. Autonomous Navigation Systems in GPS-Denied Environments: A Review of Techniques and Applications. *2025 11th International Conference on Automation, Robotics, and Applications (ICARA).* 2025. pp. 290-299. Режим доступу: <https://doi.org/10.1109/ICARA64554.2025.10977619>.
58. S. Rezwani, W. Choi. Artificial Intelligence Approaches for UAV Navigation: Recent Advances and Future Challenges. *IEEE Access.* 2022. 10. С. 26320-26339. Режим доступу: <https://doi.org/10.1109/ACCESS.2022.3157626>.
59. Gyagenda N. та ін. A review of GNSS-independent UAV navigation techniques. *Robotics and Autonomous Systems.* 2022. 152. Режим доступу: <https://doi.org/10.1016/j.robot.2022.104069>.
60. Ye X., Song F., Zhang Z., Zeng Q. A Review of Small UAV Navigation System Based on Multisource Sensor Fusion. *IEEE Sensors Journal.* 2023. 17(23). С. 18926-18948. Режим доступу: <https://doi.org/10.1109/JSEN.2023.3292427>.

61. GUO T. UAV navigation in high dynamic environments: A deep reinforcement learning approach. *Chinese Journal of Aeronautics*. 34(2). 2021. С. 479-489. Режим доступа: <https://doi.org/10.1016/j.cja.2020.05.011>.
62. Yin Y. та ін. Autonomous UAV Navigation with Adaptive Control Based on Deep Reinforcement Learning. *Electronics*. 2024. 13(13). 2432. Режим доступа: <https://doi.org/10.3390/electronics13132432>.
63. D. Saccani, L. Cecchin, L. Fagiano. Multitrajectory Model Predictive Control for Safe UAV Navigation in an Unknown Environment. *IEEE Transactions on Control Systems Technology*. 2023. 5(31). С. 1982-1997. Режим доступа: <https://doi.org/10.1109/TCST.2022.3216989>.
64. Xue Y., Chen W. A UAV Navigation Approach Based on Deep Reinforcement Learning in Large Cluttered 3D Environments. *IEEE Transactions on Vehicular Technology*. 2023. 3(72). С. 3001-3014. Режим доступа: <https://doi.org/10.1109/TVT.2022.3218855>.
65. Boiteau S., Vanegas F., Gonzalez F. Framework for Autonomous UAV Navigation and Target Detection in Global-Navigation-Satellite-System-Denied and Visually Degraded Environments. *Remote Sensing*. 2024. 16(3). 471. Режим доступа: <https://doi.org/10.3390/rs16030471>.
66. Aldao E., González-de Santos L. M., González-Jorge H. LiDAR Based Detect and Avoid System for UAV Navigation in UAM Corridors. *Drones*. 2022. 6(8). 185. Режим доступа: <https://doi.org/10.3390/drones6080185>.
67. Arafat M. Y., Alam M. M., Moh S. Vision-Based Navigation Techniques for Unmanned Aerial Vehicles: Review and Challenges. *Drones*. 2023. 7(2). 89. Режим доступа: <https://doi.org/10.3390/drones7020089>.
68. Xue Y., Chen W. Combining Motion Planner and Deep Reinforcement Learning for UAV Navigation in Unknown Environment. *IEEE Robotics and Automation Letters*. 2024. 1(9). С. 635-642. Режим доступа: <https://doi.org/10.1109/LRA.2023.3334978>.

69. Katkuri A. та ін. Autonomous UAV navigation using deep learning-based computer vision frameworks: A systematic literature review. *Array*. 2024. 23. Режим доступу: <https://doi.org/10.1016/j.array.2024.100361>.
70. Liu S. та ін. AerialVLN: Vision-and-Language Navigation for UAVs. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2023. С. 15384-15394. Режим доступу: <https://doi.org/10.48550/arXiv.2308.06735>.
71. Xu Z. та ін. A real-time dynamic obstacle tracking and mapping system for UAV navigation and collision avoidance with an RGB-D camera. *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023. С. 10645-10651. Режим доступу: <https://doi.org/10.1109/ICRA48891.2023.10161194>.
72. Lindstrom C., Christensen R., Gunther J., Jenkins S. . Gps-denied navigation aided by synthetic aperture radar using the range-doppler algorithm. *NAVIGATION: Journal of the Institute of Navigation*. 2022. 69(3), 533. Режим доступу: <https://doi.org/10.33012/navi.533>.
73. Yin P. та ін. Isimloc: Visual global localization for previously unseen environments with simulated images. *IEEE Transactions on Robotics*. 2023. 39(3), С. 1893–1909. Режим доступу: <https://doi.org/10.1109/TRO.2023.3238201>.
74. Ellingson G., Brin, K., McLain, T. Relative navigation of fixed-wing aircraft in gps-denied environments. *NAVIGATION: Journal of the Institute of Navigation*. 2020. 67(2). С. 255–273. Режим доступу: <https://doi.org/10.1002/navi.364>.

ДОДАТОК А (обов'язковий)

Лістинг коду програмного забезпечення інтелектуальної кіберфізичної системи для навігації БПЛА в умовах дії засобів РЕБ противника

```
import sys
sys.path.append('core')

import argparse
import cv2
import numpy as np
import torch
from PIL import Image
import matplotlib.pyplot as plt
import matplotlib.animation as animation
from matplotlib.path import Path
import matplotlib.patches as patches
import math
import time

import os
import subprocess

from raft import RAFT
from utils import flow_viz

from filterpy.kalman import KalmanFilter
from pymavlog import MavLog

DEVICE = 'cuda'

crop_length_RAFT = 800
crop_offset_RAFT = 400

cap = None

plt.style.use("dark_background")

fig_width = 14
fig_height = 10

fig, axs = plt.subplots(nrows = 2, ncols = 3, figsize = (fig_width, fig_height), num = "Nvigation
OF Test")

for x in axs:
    for ax in x:
        ax.axis("off")

pos_polar = 231
```

```

ax_polar = plt.subplot(pos_polar, polar = True)

ax_angle = axs[0, 1]
ax_magnitude = ax_angle.twinx()
path_gridspec = axs[1, 0].get_gridspec()
ax_path = plt.subplot(path_gridspec[1, 0:2])
ax_image = axs[0, 2]
ax_flow = axs[1, 2]

ax_angle.axis("on")
ax_angle.set_title(label = "Flow angle and magnitude", color = "green")
ax_angle.set_xlabel(xlabel = "Frame number", color = "green")
ax_angle.set_ylim(bottom = -3.20, top = 3.20,)
ax_angle.set_ylabel(ylabel = "Angle in rads", color = "green")
ax_angle.yaxis.set_label_position(position = "left")
ax_angle.tick_params(colors = "green")

ax_polar.tick_params(colors = "green")
ax_polar.grid(which = "major", axis = "both", color = "gray")
ax_polar.set_theta_zero_location("N")
ax_polar.set_theta_direction(-1)

ax_magnitude.set_ylabel(ylabel = "Magnitude", color = "blue")
ax_magnitude.tick_params(axis = "y", colors = "blue")

ax_path.grid(which = "major", axis = "both", color = "gray")
ax_path.tick_params(colors = "green")
ax_path.set_title(label = "Path", color = "green")

ax_image.set_title(label = "Image", color = "green")
ax_flow.set_title(label = "Flow", color = "green")

line_angle, = ax_angle.plot([], [], lw = 2, color = "green")
line_magnitude, = ax_magnitude.plot([], [], lw = 2, color = "blue")

plt.tight_layout(pad = 1, w_pad = 1, h_pad = 1.0)

angle_x_data = []
angle_y_data = []

magnitude_x_data = []
magnitude_y_data = []

angle_smoothed_prev = np.zeros(0)
magnitude_smoothed_prev = np.zeros(0)

start_point = (0, 0)

codes = [Path.MOVETO, Path.LINETO]

calc_time = []

prev_frame = []

```

```

anim_figures = None
model = None

magnitude_avg = 0
angle_avg = 0

distance_height_coef = 0.00098
g = 9.81

log_path = "2026-04-21 18-49-04.bin"

mavlog = MavLog(log_path)
mavlog.parse()

measurements_imu = np.array([mavlog["IMU"]["TimeUS"][:,2],
                             mavlog["IMU"]["AccX"][:,2],
                             mavlog["IMU"]["AccY"][:,2],
                             mavlog["IMU"]["AccZ"][:,2] + g,
                             mavlog["IMU"]["GyrX"][:,2],
                             mavlog["IMU"]["GyrY"][:,2],
                             mavlog["IMU"]["GyrZ"][:,2]])

measurements_baro = np.array([mavlog["BARO"]["TimeUS"],
                              mavlog["BARO"]["Alt"]])

measurements_rangefinder = np.array([mavlog["RFND"]["TimeUS"],
                                      mavlog["RFND"]["Dist"]])

measurements_compass = np.array([mavlog["AHR2"]["TimeUS"],
                                  mavlog["AHR2"]["Yaw"]])

flight_time = np.array([mavlog["RFRH"]["TimeUS"],
                        mavlog["RFRH"]["TF"]])

measurement_index_imu = 0
measurement_index_baro = 0
measurement_index_rangefinder = 0
measurement_index_compass = 0

rangefinder_min_dist_m = 6.0
rangefinder_offset_dist_m = 5.0

of_delta_time_s = 1 / 30

prev_of_vel_x = 0
prev_of_vel_y = 0
current_of_pos_x = 0
current_of_pos_y = 0
current_of_vel_x = 0
current_of_vel_y = 0

kf = KalmanFilter(dim_x = 6, dim_z = 12)

```

```

kf.x = np.array([[0.],
                 [0.],
                 [0.],
                 [0.],
                 [0.]])

kf.F = np.array([[0., 0., 0., 0., 0., 0.],
                 [0., 0., 0., 0., 0., 0.],
                 [0., 0., 0., 0., 0., 0.],
                 [0., 0., 0., 1., 0., 0.],
                 [0., 0., 0., 0., 1., 0.],
                 [0., 0., 0., 0., 0., 0.]])

kf.P *= 2.
kf.R *= 0.01

timestamp_imu1 = measurements_imu[0, 0] #* 1e-6
timestamp_baro = measurements_baro[0, 0] #* 1e-6
timestamp_rangefinder = measurements_rangefinder[0, 0] #* 1e-6
timestamp_compass = measurements_compass[0, 0] #* 1e-6

imu_time_armed_s = measurements_imu[0, 0] * 1e-6

def clear_console():
    cmd = "cls" if os.name == "nt" else "clear"
    subprocess.call(cmd, shell=True)

def read_image_RAFT(cap, skip = 0):
    for i in range(skip):
        cap.read()

    ret, image = cap.read()

    if not ret:
        return None

    image_cropped = image[crop_offset_RAFT:crop_length_RAFT, crop_offset_RAFT:crop_length_RAFT]

    image_cropped = torch.from_numpy(image_cropped).permute(2, 0, 1).float()

    return image_cropped[None].to(DEVICE)

def visualizeRAFT_flow(image, flow):
    image = image[0].permute(1, 2, 0).cpu().numpy()
    flow_image = flow[0].permute(1, 2, 0).cpu().numpy()

    # map flow to rgb image
    flow_image = flow_viz.flow_to_image(flow_image)

    visualize_flow(image[:, :, [2,1,0]] / 255.0, flow_image[:, :, [2,1,0]] / 255.0)

```

```

def visualize_flow(image, flow):
    ax_image.clear()
    ax_image.axis("off")
    ax_image.set_title(label = "Image", color = "green")
    ax_image.imshow(image)

    ax_flow.clear()
    ax_flow.axis("off")
    ax_flow.imshow(flow)

def average_angle(angle):
    sin = np.sin(angle)
    cos = np.cos(angle)

    sin_avg, cos_avg = np.average(sin), np.average(cos)

    return np.atan2(sin_avg, cos_avg)

def convert_to_polar(x_start, y_start, x_end, y_end):
    x = x_end - x_start
    y = y_end - y_start

    magnitude = math.sqrt(x**2 + y**2)
    angle = math.atan2(y, x)

    return magnitude, angle + 2 * np.pi

def extract_bigger_than(numbers, numbers_paralel, threshold):
    filtered_numbers = []
    filtered_numbers_paralel = []

    for index, number in enumerate(numbers):
        if abs(number) > threshold:
            filtered_numbers.append(number)
            filtered_numbers_paralel.append(numbers_paralel[index])

    return np.array(filtered_numbers), np.array(filtered_numbers_paralel)

def smooth_angle(angle, angle_prev, smoothing_coef):
    sin = np.sin(angle)
    cos = np.cos(angle)

    sin_prev = np.sin(angle_prev)
    cos_prev = np.cos(angle_prev)

    sin_prev_avg, cos_prev_avg = np.average(sin_prev), np.average(cos_prev)

    sin_smoothed = sin_prev_avg + smoothing_coef * (sin - sin_prev_avg)
    cos_smoothed = cos_prev_avg + smoothing_coef * (cos - cos_prev_avg)

    return np.atan2(sin_smoothed, cos_smoothed)

def smooth_magnitude(magnitude, magnitude_prev, smoothing_coef):

```

```

return np.array(magnitude_prev + smoothing_coef * (magnitude - magnitude_prev))

def smooth_data(magnitude, angle, filter_threshold, smooth_coef):
    global angle_smoothed_prev, magnitude_smoothed_prev

    filtered_magnitude, filtered_angle = extract_bigger_than(
        np.array(magnitude).flatten(),
        np.array(angle).flatten(),
        threshold = filter_threshold)

    if filtered_magnitude.size != 0:
        if magnitude_smoothed_prev.size == 0:
            smoothed_magnitude = filtered_magnitude
        else:
            smoothed_magnitude = smooth_magnitude(filtered_magnitude,
np.average(magnitude_smoothed_prev), smoothing_coef = smooth_coef)

        magnitude_smoothed_prev = smoothed_magnitude

        magnitude_smoothed_avg = np.average(smoothed_magnitude)
    else:
        magnitude_smoothed_avg = 0

    if filtered_angle.size != 0:
        if angle_smoothed_prev.size == 0:
            smoothed_angle = filtered_angle
        else:
            smoothed_angle = smooth_angle(filtered_angle, angle_smoothed_prev, smoothing_coef =
smooth_coef)

        angle_smoothed_prev = smoothed_angle

        angle_smoothed_avg = average_angle(smoothed_angle)
    else:
        angle_smoothed_avg = 0

    return magnitude_smoothed_avg, angle_smoothed_avg

def visualize_data(x, y, yaw, height, of_magnitude, of_angle, frame_number, calc_time):
    global ax_polar, line_angle, ax_angle, ax_flow, start_point

    avg_calc_time = np.average(calc_time)
    min_calc_time = np.min(calc_time)
    max_calc_time = np.max(calc_time)

    ax_flow.set_title(f"Flow\nAvg calc time: {avg_calc_time:.2f} ms\nLast: {calc_time[-1]:.2f}
ms. Min: {min_calc_time:.2f} ms. Max: {max_calc_time:.2f} ms.", color = "green")

    ax_polar.clear()
    ax_polar.set_title(label = "Drone heading", color = "green")
    ax_polar.set_theta_zero_location("N")
    ax_polar.set_theta_direction(-1)

```

```

angle_x_data.append(frame_number)
angle_y_data.append(of_angle)

magnitude_x_data.append(frame_number)
magnitude_y_data.append(of_magnitude)

if of_magnitude == 0.0:
    ax_polar.vlines(x = yaw, ymin = 0, ymax = 0.1, colors = "green")
else:
    ax_polar.vlines(x = yaw, ymin = 0, ymax = of_magnitude, colors = "green")

xmin_angle, xmax_angle = ax_angle.get_xlim()
ymin_magnitude, ymax_magnitude = ax_magnitude.get_ylim()

frames_x_offset = 200

if angle_x_data[-1] >= xmax_angle:
    if xmax_angle < frames_x_offset:
        ax_angle.set_xlim(xmin_angle, xmax_angle * 2)
    else:
        ax_angle.set_xlim(xmin_angle + frames_x_offset, xmax_angle + frames_x_offset)

    ax_angle.figure.canvas.draw()

if magnitude_y_data[-1] >= ymax_magnitude:
    ax_magnitude.set_ylim(ymin_magnitude, 2 * ymax_magnitude)
    ax_magnitude.figure.canvas.draw()

end_point = (y, x)

xmin_path, xmax_path = ax_path.get_xlim()
ymin_path, ymax_path = ax_path.get_ylim()

heading_point = (end_point[0] + math.sin(yaw) * 0.2 * max(abs(xmin_path), abs(xmax_path)),
end_point[1] + math.cos(yaw) * 0.2 * max(abs(ymin_path), abs(ymax_path)))

if abs(end_point[0]) >= abs(heading_point[0]):
    if end_point[0] >= xmax_path:
        ax_path.set_xlim(xmin_path, end_point[0] * 1.2)
        ax_path.figure.canvas.draw()
    elif end_point[0] <= xmin_path:
        ax_path.set_xlim(end_point[0] * 1.2, xmax_path)
        ax_path.figure.canvas.draw()
else:
    if heading_point[0] >= xmax_path:
        ax_path.set_xlim(xmin_path, heading_point[0] * 1.2)
        ax_path.figure.canvas.draw()
    elif heading_point[0] <= xmin_path:
        ax_path.set_xlim(heading_point[0] * 1.2, xmax_path)
        ax_path.figure.canvas.draw()

if abs(end_point[1]) >= abs(heading_point[1]):
    if end_point[1] >= ymax_path:

```

```

        ax_path.set_ylim(ymin_path, end_point[1] * 1.2)
        ax_path.figure.canvas.draw()
    elif end_point[1] <= ymin_path:
        ax_path.set_ylim(end_point[1] * 1.2, ymax_path)
        ax_path.figure.canvas.draw()
else:
    if heading_point[1] >= ymax_path:
        ax_path.set_ylim(ymin_path, heading_point[1] * 1.2)
        ax_path.figure.canvas.draw()
    elif heading_point[1] <= ymin_path:
        ax_path.set_ylim(heading_point[1] * 1.2, ymax_path)
        ax_path.figure.canvas.draw()

verts = [ start_point, end_point ]
path = Path(verts, codes)
patch = patches.PathPatch(path, color = "green", lw = 2)

return_verts = [ (0, 0), end_point ]
return_path = Path(return_verts, codes)
return_patch = patches.PathPatch(return_path, color = "blue", lw = 2)

heading_verts = [ end_point, heading_point ]
heading_path = Path(heading_verts, codes)
heading_patch = patches.PathPatch(heading_path, color = "red", lw = 1)

if frame_number > 0:
    ax_path.patches[-1].remove()
    ax_path.patches[-1].remove()

ax_path.add_patch(patch)
ax_path.add_patch(return_patch)
ax_path.add_patch(heading_patch)

return_magnitude, return_angle = convert_to_polar(end_point[1], end_point[0], 0, 0)

home_direction = abs(360 - return_angle * (180 / np.pi))

if home_direction > 360:
    home_direction = home_direction - 360

ax_path.set_title(f"Path. Dist to home: {return_magnitude:.1f} m. Home dir:
{(home_direction):.1f}°. Height: {(height):.1f} m", color = "green")

start_point = end_point

line_angle.set_data(angle_x_data, angle_y_data)
line_magnitude.set_data(magnitude_x_data, magnitude_y_data)

def clear_axis():
    global ax_path, ax_angle, ax_magnitude, line_angle, line_magnitude
    global start_point, calc_time
    global anim_figures, model, prev_frame, frame

```

```

ax_path.clear()
ax_angle.clear()
ax_magnitude.clear()

for index, _ in enumerate(ax_path.patches):
    ax_path.patches[index].remove()

ax_path.grid(which = "major", axis = "both", color = "gray")

line_angle, = ax_angle.plot([], [], lw = 2, color = "green")
line_magnitude, = ax_magnitude.plot([], [], lw = 2, color = "blue")

magnitude_x_data.clear()
magnitude_y_data.clear()

angle_x_data.clear()
angle_y_data.clear()

ax_angle.set_title(label = "Flow angle and magnitude", color = "green")
ax_angle.set_xlabel(xlabel = "Frame number", color = "green")
ax_angle.set_ylim(bottom = -3.20, top = 3.20)
ax_angle.set_xlim(left = 0, right = 60)
ax_angle.set_ylabel(ylabel = "Angle in rads", color = "green")
ax_angle.yaxis.set_label_position(position = "left")
ax_angle.tick_params(colors = "green")

ax_magnitude.set_ylabel(ylabel = "Magnitude", color = "blue")
ax_magnitude.tick_params(axis = "y", colors = "blue")
ax_magnitude.yaxis.set_label_position(position = "right")

start_point = (0, 0)

calc_time = []
frame = []

def animate_RAFT(frame_number):
    global prev_frame, start_point, model, anim_figures, cap, calc_time, magnitude_avg, angle_avg
    global kf, prev_of_vel_x, prev_of_vel_y, current_of_vel_x, current_of_vel_y,
current_of_pos_x, current_of_pos_y
    global measurement_index_imu, measurement_index_baro, measurement_index_rangefinder,
measurement_index_compass
    global timestamp_imu1, timestamp_baro, timestamp_compass, timestamp_rangefinder,
imu_time_armed_s

    clear_console()
    print(f"Raft. Frame number: {frame_number}\t", end = '\r')

    while True:
        alpha = kf.x[3][0]
        beta = kf.x[4][0]
        gamma = kf.x[5][0]

```

```

R_xyz = np.matrix([[math.cos(gamma) * math.cos(beta), math.cos(gamma) * math.sin(beta) *
math.sin(alpha) - math.sin(gamma) * math.cos(alpha), math.cos(gamma) * math.sin(beta) * math.cos(alpha)
+ math.sin(gamma) * math.sin(alpha)],
                  [math.sin(gamma) * math.cos(beta), math.sin(gamma) * math.sin(beta) *
math.sin(alpha) + math.cos(gamma) * math.cos(alpha), math.sin(gamma) * math.sin(beta) * math.cos(alpha)
- math.cos(gamma) * math.sin(alpha)],
                  [-1. * math.sin(beta), math.cos(beta) * math.sin(alpha), math.cos(beta)
* math.cos(alpha)]])

R_z = np.matrix([[math.cos(gamma), -1. * math.sin(gamma), 0.],
                 [math.sin(gamma), math.cos(gamma), 0.],
                 [0., 0., 1.]])

measurement = np.zeros((12, 1))

for j in range(1, 3):
    measurement[j - 1] = measurements_imu[j, measurement_index_imu] * 1e-12

for j in range(4, 7):
    measurement[j - 1] = measurements_imu[j, measurement_index_imu] * 1e-6

timestamp_imu1 = measurements_imu[0, measurement_index_imu]

if measurement_index_baro < len(measurements_baro[0]):
    if rangefinder_min_dist_m <= measurements_baro[1, measurement_index_baro] and
measurement_index_rangefinder < len(measurements_rangefinder[0]):
        measurement[6][0] = measurements_rangefinder[1, measurement_index_rangefinder] -
rangefinder_offset_dist_m
    else:
        measurement[6][0] = measurements_baro[1, measurement_index_baro]

if (timestamp_baro <= timestamp_imu1) and (measurement_index_baro + 1 <
len(measurements_baro[0])):
    measurement_index_baro += 1
    timestamp_baro = measurements_baro[0, measurement_index_baro]

if (timestamp_rangefinder <= timestamp_imu1) and (measurement_index_rangefinder + 1 <
len(measurements_rangefinder[0])):
    measurement_index_rangefinder += 1
    timestamp_rangefinder = measurements_rangefinder[0, measurement_index_rangefinder]

if measurement_index_compass < len(measurements_compass[0]):
    measurement[7][0] = measurements_compass[1, measurement_index_compass] * math.pi /
180.0

if (timestamp_compass <= timestamp_imu1) and (measurement_index_compass + 1 <
len(measurements_compass[0])):
    measurement_index_compass += 1
    timestamp_compass = measurements_compass[0, measurement_index_compass]

a_local = np.matrix([[measurement[0][0]],
                    [measurement[1][0]],
                    [measurement[2][0]]])

```

```

a_global = R_xyz * a_local

measurement[0][0] = a_global[0, 0]
measurement[1][0] = a_global[1, 0]
measurement[2][0] = a_global[2, 0]

imu_time_delta_since_armed_s = measurements_imu[0, measurement_index_imu] * 1e-6 -
imu_time_armed_s

v_local = np.matrix([[prev_of_vel_x],
                    [prev_of_vel_y],
                    [0]])

v_global = R_z * v_local

pos_local = np.matrix([[current_of_pos_x],
                      [current_of_pos_y],
                      [0]])

pos_global = R_z * pos_local

measurement[8][0] = v_global[0, 0]
measurement[9][0] = v_global[1, 0]
measurement[10][0] = pos_global[0, 0]
measurement[11][0] = pos_global[1, 0]

prev_of_vel_x = current_of_vel_x
prev_of_vel_y = current_of_vel_y

converted_measurement_time_imu1 = measurements_imu[0, measurement_index_imu] -
measurements_imu[0, measurement_index_imu + 1]

kf.H = np.array([[2.0/(converted_measurement_time_imu1**2), 0., 0., 0., 0., 0.],
                [0., 2.0/(converted_measurement_time_imu1**2), 0., 0., 0., 0.],
                [0., 0., 2.0/(converted_measurement_time_imu1**2), 0., 0., 0.],
                [0., 0., 0., 1.0/converted_measurement_time_imu1, 0., 0.],
                [0., 0., 0., 0., 1.0/converted_measurement_time_imu1, 0.],
                [0., 0., 0., 0., 0., 1.0/converted_measurement_time_imu1],
                [0., 0., 1., 0., 0., 0.],
                [0., 0., 0., 0., 0., 1.],
                [1./converted_measurement_time_imu1, 0., 0., 0., 0., 0.],
                [0., 1./converted_measurement_time_imu1, 0., 0., 0., 0.],
                [1., 0., 0., 0., 0., 0.],
                [0., 1., 0., 0., 0., 0.]])

kf.B = np.array([[.5 * converted_measurement_time_imu1**2, 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.],
                [of_delta_time_s * 1e6, 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
                [0., .5 * converted_measurement_time_imu1**2, 0., 0., 0., 0., 0., 0., 0., 0., 0.],
                [of_delta_time_s * 1e6, 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
                [0., 0., .5 * converted_measurement_time_imu1**2, 0., 0., 0., 1., 0., 0., 0., 0.],
                [0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]])

```

```

        [0., 0., 0., converted_measurement_time_imu1, 0., 0., 0., 0., 0., 0., 0., 0.,
0.],
        [0., 0., 0., 0., converted_measurement_time_imu1, 0., 0., 0., 0., 0., 0., 0.,
0.],
        [0., 0., 0., 0., 0., converted_measurement_time_imu1, 0., 1., 0., 0., 0.,
0.])

kf.predict(measurement)
kf.update(measurement)

measurement_index_imu += 1

if imu_time_delta_since_armed_s > of_delta_time_s * frame_number:
    break
else:
    current_of_vel_x = 0.
    current_of_vel_y = 0.

with torch.no_grad():
    if(cap.isOpened()):
        frame = read_image_RAFT(cap, skip = 1)

        if frame is None:
            cap.release()
            cv2.destroyAllWindows()

            anim_figures.event_source.stop()

            return

        calc_time_start = time.time()

        flow_predictions = model(prev_frame, frame)

        calc_time_end = time.time()

        flow_up = flow_predictions[0]

        prev_frame = frame

        visuaillizeRAFT_flow(prev_frame, flow_up)

        flow = flow_up[0].permute(1, 2, 0).cpu().numpy()

        magnitude, angle = cv2.cartToPolar(flow[:, :, 0], flow[:, :, 1])

        angle = np.pi - angle

        calc_time.append((calc_time_end - calc_time_start) * 1000)

        magnitude_avg = np.average(magnitude) * abs(kf.x[2][0]) * distance_height_coef
        angle_avg = np.average(angle)

```

```

        current_of_pos_x += math.sin(angle_avg) * magnitude_avg
        current_of_pos_y += math.cos(angle_avg) * magnitude_avg
        current_of_vel_x = current_of_pos_x / (of_delta_time_s * 1e6)
        current_of_vel_y = current_of_pos_y / (of_delta_time_s * 1e6)
    else:
        cap.release()
        cv2.destroyAllWindows()

        anim_figures.event_source.stop()

    return

    visualize_data(kf.x[0][0], kf.x[1][0], kf.x[5][0], kf.x[2][0], magnitude_avg, angle_avg,
frame_number, calc_time)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        cap.release()
        cv2.destroyAllWindows()

        anim_figures.event_source.stop()

    return

'''
video_file_name - video file name.

raft_model - wich model to use to evaluate Raft.
-- models/raft-chairs.pth
-- models/raft-kitti.pth
-- models/raft-sintel.pth
-- models/raft-small.pth
-- models/raft-things.pth

small - make model size small or not.
-- True
-- False

mixed_precision - use mixed precision or not.
(https://docs.pytorch.org/tutorials/recipes/recipes/amp\_recipe.html)
-- True
-- False

alternate_corr - makse model slower than all-pairs, but uses significantly less GPU memory during
the forward pass.
-- True
-- False
'''

def start_raft(video_file_name, raft_model, small = False, mixed_precision = False, alternate_corr
= False, show_plot = False, frames = 2880, fps = 60):
    global prev_frame, anim_figures, model, cap

    cap = cv2.VideoCapture(video_file_name)

```

```

args = argparse.Namespace()

args.model = raft_model
args.small = small
args.mixed_precision = mixed_precision
args.alternate_corr = alternate_corr

model = torch.nn.DataParallel(RAFT(args))
model.load_state_dict(torch.load(args.model))

model = model.module
model.to(DEVICE)
model.eval()

prev_frame = read_image_RAFT(cap)

anim_figures = animation.FuncAnimation(
    fig,
    animate_RAFT,
    cache_frame_data = False,
    frames = frames,
    interval = 0,
    init_func = clear_axis,
    repeat = False)

raft_model = raft_model.replace('models/', '')

if show_plot:
    plt.show()
else:
    anim_figures.save(
        filename =
            "magister/Raft optical flow test model = " + str(raft_model) +
            ", small = " + str(small) +
            ", mixed_precision = " + str(mixed_precision) +
            ", alternate_corr = " + str(alternate_corr) +
            ", distance_height_coef = " + str(distance_height_coef) +
            ".mp4",
        fps = fps)

video_file_path = "GPS Fly.mp4"

start_raft(
    video_file_path,
    raft_model = 'models/raft-small.pth',
    show_plot = False,
    small = True,
    mixed_precision = False,
    alternate_corr = False, frames = 4765, fps = 30)

```

ДОДАТОК Б (обов'язковий)

Публікація

Міжнародний науково-технічний журнал
«Вимірювальна та обчислювальна техніка в технологічних процесах»

ISSN 2219-9365

<https://doi.org/10.31891/2219-9365-2026-86-4>

УДК 004.9

БЕРЕЗЬКА Катерина
Хмельницький національний університет
<https://orcid.org/0000-0002-9632-4004>
e-mail: km.berezka@gmail.com

СЄВОСТЬЯНОВ Владислав
Хмельницький національний університет
<https://orcid.org/0009-0000-9707-3756>
e-mail: sevost@i.ua

ІНТЕЛЕКТУАЛЬНА КІБЕРФІЗИЧНА СИСТЕМА ДЛЯ НАВИГАЦІЇ БПЛА В УМОВАХ ДІЇ ЗАСОБІВ РЕБ ПРОТИВНИКА

У сучасних умовах ведення бойових дій, зокрема в ході російсько-української війни, роль безпілотних літальних апаратів (БПЛА) стала визначальною. Однак масоване застосування засобів радіоелектронної боротьби (РЕБ) противником часто робить неможливим використання стандартних супутникових систем навігації (GNSS) через придушення або підміну сигналу. Це зумовлює гостру потребу в розробці автономних кіберфізичних систем, які здатні забезпечувати точне позиціонування БПЛА в умовах повної відсутності зовнішніх радіосигналів. Дослідження спрямоване на пошук та імплементацію ефективного програмно-апаратного рішення для відносного позиціонування БПЛА. Основна увага приділяється створенню системи, що поєднує дані з інерціальних датчиків та камер оптичного потоку для забезпечення надійного повернення апарату до точки зльоту або утримання позиції без використання GPS. У роботі запропоновано концептуальну модель системи, що базується на комплексованні даних з акселерометра, гіроскопа, магнітометра, барометра, лідара та камери оптичного потоку. Для фільтрації та об'єднання цих даних використано розширений фільтр Калмана (EKF). Проведено порівняльний аналіз трьох алгоритмів обчислення оптичного потоку: розрідженого (Lucas-Kanade), щільного (Fameback) та сучасного нейронмережевого підходу RAFT (Recurrent All-Pairs Field Transforms). Для підвищення точності розроблено тривірневу систему фільтрації та експоненційного згладжування векторів руху. Тестування алгоритмів у різних сценаріях (політ на висоті 25 м та рух на низькій висоті) показало, що розріджений алгоритм, попри високу швидкість, є найменш стабільним і схильним до хаотичних похибок. Щільний алгоритм продемонстрував низьку ефективність за певних параметрів обчислень. Найкращі результати за критеріями стабільності та точності відтворення маршруту показав алгоритм RAFT, особливо його зменшена модель, яка працює в 1,3 рази швидше за повнорозмірну версію. Встановлено, що інтелектуальна кіберфізична система на основі алгоритму RAFT та мультисенсорного об'єднання даних є найбільш перспективною для навігації в умовах дії РЕБ.

Ключові слова: БПЛА, радіоелектронна боротьба, оптичний потік, RAFT, фільтр Калмана, автономна навігація, кіберфізичні системи, GNSS-denied.

BEREZKA Kateryna, SEVOSTYANOV Vladyslav
Khmelnitskiy National University

INTELLIGENT CYBER-PHYSICAL SYSTEM FOR UAV NAVIGATION UNDER ENEMY ELECTRONIC WARFARE CONDITIONS

In modern conditions of combat operations, in particular during the Russian-Ukrainian war, the role of unmanned aerial vehicles (UAVs) has become decisive. However, the massive use of electronic warfare (EW) by the enemy often makes it impossible to use standard satellite navigation systems (GNSS) due to signal suppression or substitution. This creates an urgent need to develop autonomous cyber-physical systems that are capable of ensuring accurate positioning of UAVs in the complete absence of external radio signals. The research is aimed at finding and implementing an effective software and hardware solution for relative positioning of UAVs. The main focus is on creating a system that combines data from inertial sensors and optical flow cameras to ensure reliable return of the device to the take-off point or maintaining position without using GPS. The paper proposes a conceptual model of the system based on the integration of data from an accelerometer, gyroscope, magnetometer, barometer, lidar, and optical flow camera. The extended Kalman filter (EKF) was used to filter and combine these data. A comparative analysis of three algorithms for calculating optical flow was conducted: sparse (Lucas-Kanade), dense (Fameback), and the modern neural network approach RAFT (Recurrent All-Pairs Field Transforms). To increase accuracy, a three-level filtering system and exponential smoothing of motion vectors were developed. Testing of algorithms in different scenarios (flight at an altitude of 25 m and movement at low altitude) showed that the sparse algorithm, despite its high speed, is the least stable and prone to chaotic errors. The dense algorithm demonstrated low efficiency under certain calculation parameters. The best results in terms of stability and accuracy of route reproduction were shown by the RAFT algorithm, especially its reduced model, which works 1.3 times faster than the full-size version. It was found that an intelligent cyber-physical system based on the RAFT algorithm and multi-sensor data fusion is the most promising for navigation under EW conditions. Despite the high requirements for computing resources, the use of specialized NPU or TPU modules allows achieving the required performance in real time.

Keywords: UAV, electronic warfare, optical flow, RAFT, Kalman filter, autonomous navigation, cyber-physical systems, GNSS-denied.

Стаття надійшла до редакції / Received 14.03.2026
Прийнята до друку / Accepted 22.04.2026
Опубліковано / Published 28.05.2026



This is an Open Access article distributed under the terms of the [Creative Commons CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/)

© Березька Катерина, Севостьянов Владислав

ПОСТАНОВКА ПРОБЛЕМИ У ЗАГАЛЬНОМУ ВИГЛЯДІ ТА ЇЇ ЗВ'ЯЗОК ІЗ ВАЖЛИВИМИ НАУКОВИМИ ЧИ ПРАКТИЧНИМИ ЗАВДАННЯМИ

Стрімкий розвиток ударних безпілотних літальних апаратів (БПЛА) в умовах російсько-української війни показав необхідність та ефективність їх застосування [1]. Технології їх протидії розвиваються як з нашої сторони, так і зі сторони противника. Посилене використання радіоелектронної боротьби (РЕБ) противником часто унеможливило застосування звичних супутникових систем позиціонування [2, 3]. Саме тому розробка альтернативи є актуальною задачею сьогодення [4]. Розробка кіберфізичної системи для навігації БПЛА в умовах дії засобів РЕБ противника складається з пошуку доступного та ефективного програмно-апаратного рішення та його імплементації із системою навігації БПЛА [5, 6].

Зважаючи на складність використання існуючих систем позиціонування в умовах відсутності GNSS-сигналу та їх дороговизну, розроблювана система має бути достатньо простою у використанні, компактною та недорогою в побудові [7]. Проте, основними критеріями роботи системи навігації залишаються наступні:

- точність позиціонування. Система повинна мати достатню точність визначення позиції апарату для надійного утримання точки, прокладання маршруту, чи повернення до точки зльоту з визначеною похибкою;
- відмовостійкість. Система повинна бути стійкою до змін рельєфу, рівня освітленості, погодних умов та особливостей маршруту польоту апарату;
- надійність. Система повинна витримувати роботу за різних температурних режимів, а також мати способи забезпечення безпеки апарату, якщо неможливо визначити точну позицію.

Інтелектуальна кіберфізична система для навігації БПЛА в умовах дії засобів РЕБ противника

Оптимальним вирішенням поставленої задачі є система, яка буде об'єднувати дані з кількох датчиків про позицію та напрямок руху дрона в певний момент часу. Обрані джерела даних про позиціонування не повинні залежати від попередньо визначених даних, як мапа місцевості, чи її рельєф. Датчики повинні бути повністю незалежними від радіосигналу, який може бути спотворений засобами РЕБ. Окрім того, обраний алгоритм обробки даних з датчиків та їх об'єднання повинен бути достатньо надійним та ефективним у використанні. Потрібно також зважати на обмеженість обчислювальних ресурсів літальних апаратів.

Для визначення положення апарату в просторі та зменшення похибки найкращим підходом буде використання кількох датчиків. Серед таких датчиків найкращим вибором будуть ті, що не потребують попередньо зібраних ресурсів для роботи. У Таблиці 1 зазначено обрані датчики позиціонування в просторі.

Таблиця 1

Обрані датчики позиціонування в просторі

Назва датчика	Дані про положення в просторі	Опис
Акселерометр	Значення прискорення в трьох осях	Датчик визначає величину та вектор напрямку прискорення
Гіроскоп	Положення тіла в просторі	Датчик визначає орієнтацію тіла в просторі
Магнітометр	Вказує на північний полюс Землі	Відкалібрований магнітометр може працювати в якості магнітного компаса
Барометр	Висота відносно точки зльоту	Датчик визначає тиск повітря, який змінюється з висотою
Лідар	Висота відносно поверхні Землі	Датчик вимірює відстань до об'єкту, що можна використати для визначення висоти, спрямувавши його донизу
Камера оптичного потоку	Напрямок та пройдена відстань відносно поверхні Землі	Датчик вимірює магнітуду та вектор руху

Оскільки система не передбачається як повна заміна GPS, а однією з вимог є незалежність від зовнішніх попередньо зібраних даних, тому найкращим підходом є система відносного позиціонування.

Велику роль у точності позиціонування відіграє обраний метод фільтрації даних з датчиків. Найбільш поширеним та випробуваним часом є розширений фільтр Калмана, який, враховуючи похибку з датчиків, визначає коефіцієнт коригування передбачуваної оцінки стану апарату в просторі.

На основі попереднього аналізу існуючих рішень пропонується концептуальна модель апарату, основні компоненти якого описано вище. На Рисунку 1 зображено концептуальне рішення системи відносного позиціонування дрона.

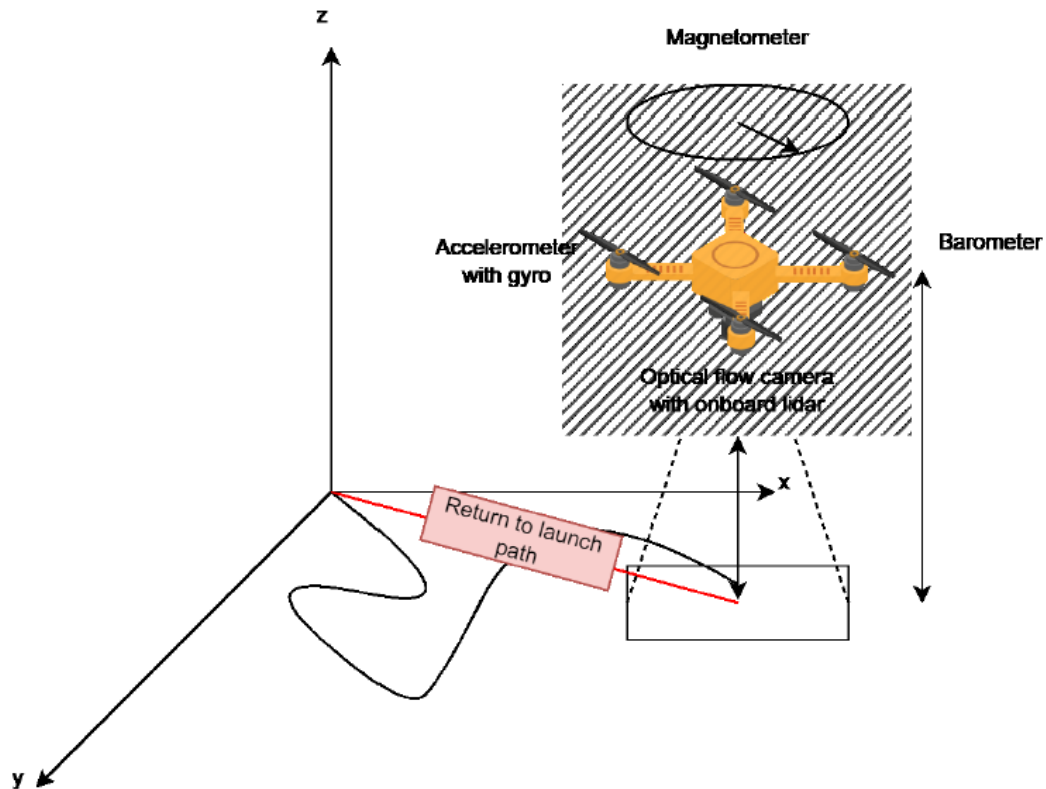


Рис. 1. Концептуальна схема системи позиціонування дрона [8-10]

Загальний алгоритм роботи системи позиціонування складається із передбачення наступного стану просторової моделі апарату, збору даних з датчиків, внесення коригування в просторову модель. Обробку та фільтрацію даних виконує розширений фільтр Калмана.

Серед існуючих алгоритмів обчислення оптичного потоку можна виділити два найбільш поширених: розріджений та щільний алгоритми. Також існує новий метод обчислення оптичного потоку на основі штучного інтелекту. Одним з кандидатів є RAFT, схема роботи якого зображено на Рисунку 2. RAFT (Recurrent All-Pairs Field Transforms) використовує рекурентну модель штучного інтелекту для обчислення оптичного потоку.

На додачу до розрідженого та щільного алгоритмів обчислення оптичного потоку було додано мінімальну фільтрацію та згладжування даних для аналізу їх впливу на точність результату. Даний метод попри свою складність є вразливим до шумів зображення, що створюють похибку та неточності обчислення оптичного потоку. Для зменшення отриманої похибки було реалізовано тривірневу фільтрацію отриманих даних.

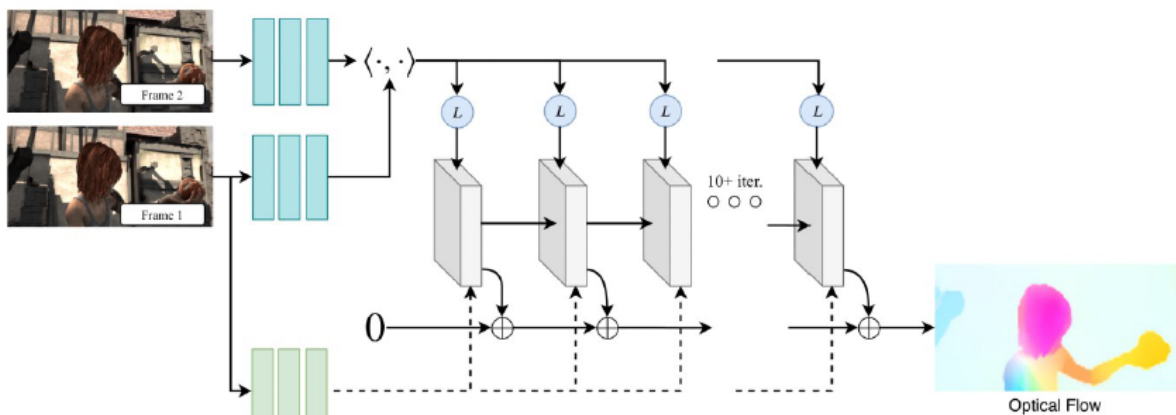


Рис. 2. Схема роботи алгоритму оптичного потоку на основі рекурентної моделі штучного інтелекту [11-13]

Перший рівень фільтрує отримані значення магнітуди та кутів за значенням магнітуди.

Другий рівень згладжує значення отриманих кутів. Оскільки алгоритм обчислення оптичного потоку може видавати велику похибку, отримані кути було вирішено згладжувати методом експоненційного згладжування. Дане згладжування добре працює з часовими рядами, проте є проблема згладжування кутів, що вказують на напрямок 0° , або 360° (0 радіан, або $6,28$ радіан), тому для кутів експоненційне згладжування було застосовано до їх синусів та косинусів, з яких уже було знову побудовано згладжені значення кутів.

Експоненційне згладжування обчислює нове значення часового ряду відповідно до попереднього, поточного та коефіцієнту згладжування.

Третій рівень згладжує магнітуду за методом експоненційного згладжування.

Наступний етап включає в себе знаходження середніх значень магнітуди та кутів векторів. Середні значення кутів знаходяться також за синусом та косинусом через проблему переходу значень радіан кутів з $6,28$ на 0 та навпаки.

Отримані значення магнітуди та кутів векторів виводяться як графіки у полярних координатах та декартових координатах, які окремо рисують значення кутів та магнітуди. Також виводяться зображення з оброблюваного відеопотоку та оптичного потоку. Виводиться обчислений маршрут руху з даних алгоритму оптичного потоку, відстань та напрямок до початку руху. Обчислюється та виводиться мінімальне, максимальне та середнє значення часу виконання алгоритму оптичного потоку.

Було проведено серію тестів для вибору оптимального алгоритму обчислення оптичного потоку.

Перше тестування включає в себе відео польоту дрона на висоті 25 метрів та опущеною камерою донизу. Тут було протестовано вплив згладжування та фільтрації на щільний та розріджений алгоритми. Оброблюване зображення було зменшено до розмірів 300 на 300 пікселів для пришвидшення процесу тестування. Критеріями оцінки були подібність пройденого маршруту дрона до того, що обчислює оптичний потік. Схематично маршрут польоту дрона виглядав наступним чином – Рисунок 3.

Приклад1 виводу даних виконано програмою для тестування оптичного потоку – Рисунок 4.

Проведемо порівняльний аналіз щільного алгоритму, розрідженого алгоритму та RAFT-алгоритму – Таблиці 2-4. Алгоритм RAFT було протестовано на 4 моделях без використання фільтрації чи згладжування (Таблиця 4).

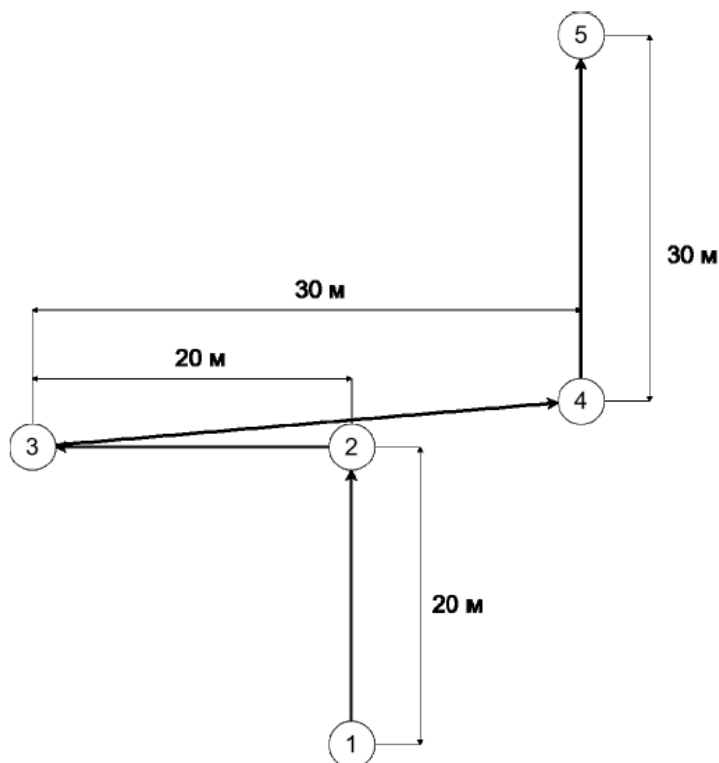


Рис. 3. Схематичне зображення пройденого дрона шляху

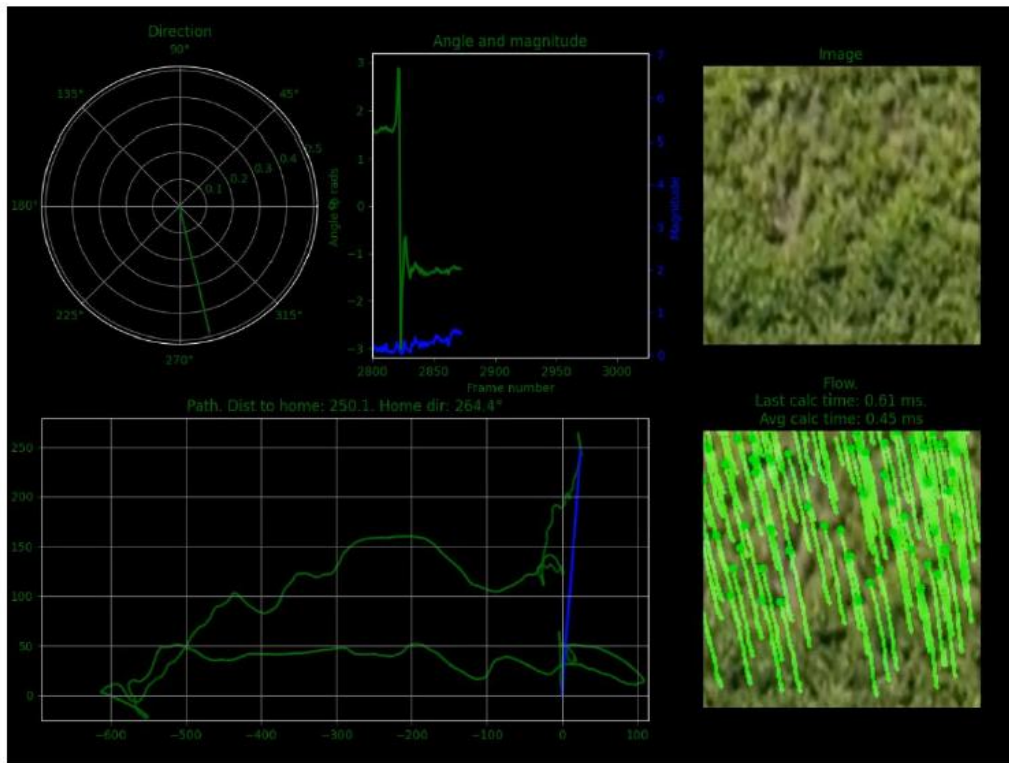


Рис. 4. Приклад виводу даних

Таблиця 2

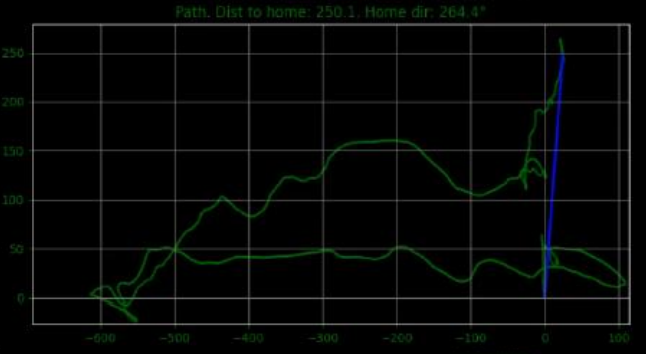
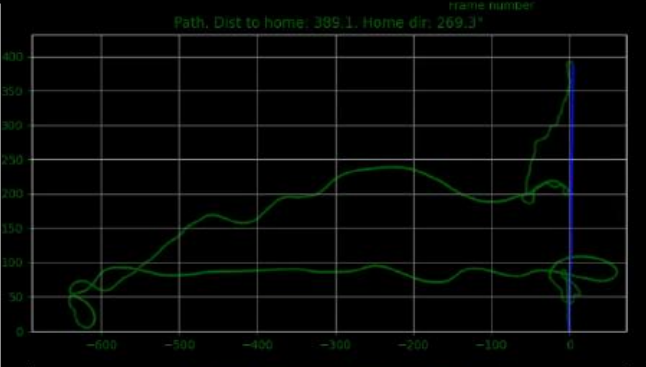
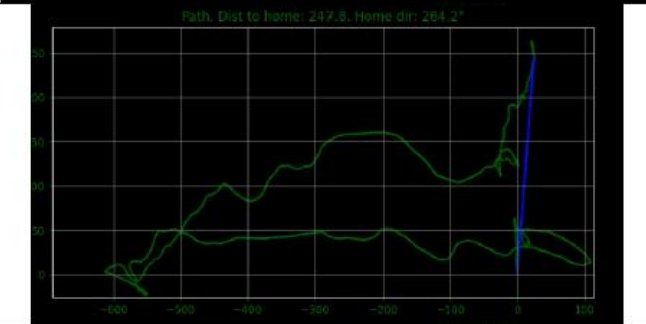
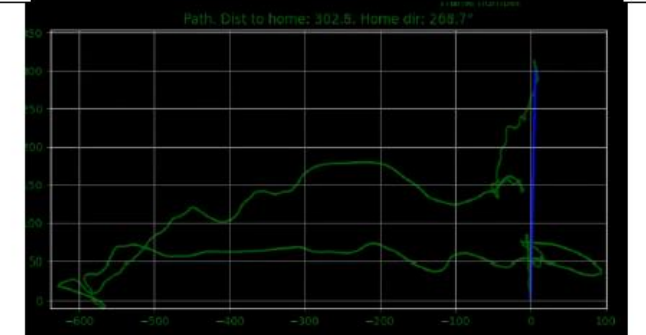
Щільний алгоритм

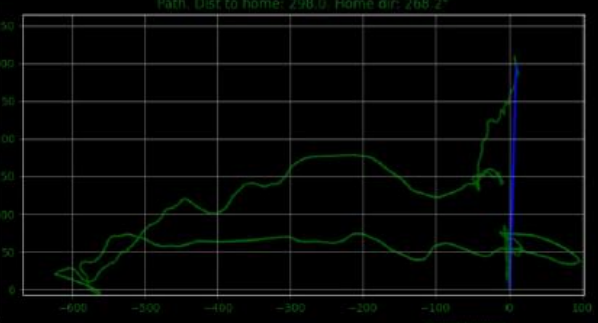
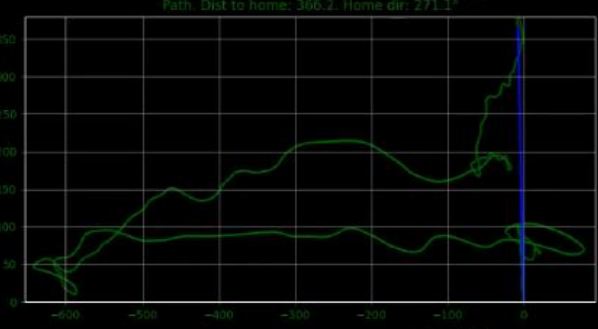
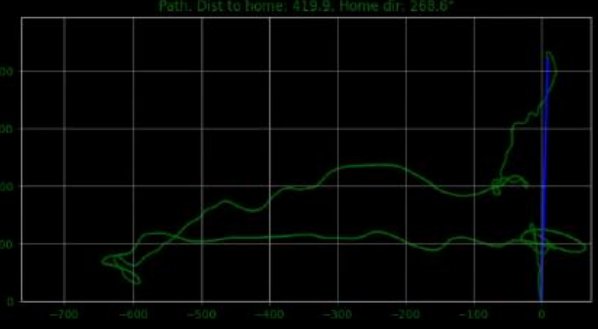
Маршрут	Коефіцієнт фільтрації	Коефіцієнт згладжування
	Без фільтрації	Без згладжування
	0,2	0,2

	<p>0,4</p>	<p>0,2</p>
	<p>0,1</p>	<p>0,1</p>
	<p>0,1</p>	<p>0,4</p>
	<p>0,1</p>	<p>0,6</p>

Таблиця 3

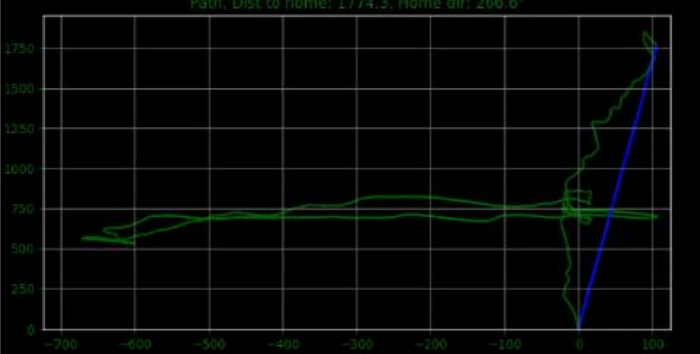
Розріджений алгоритм

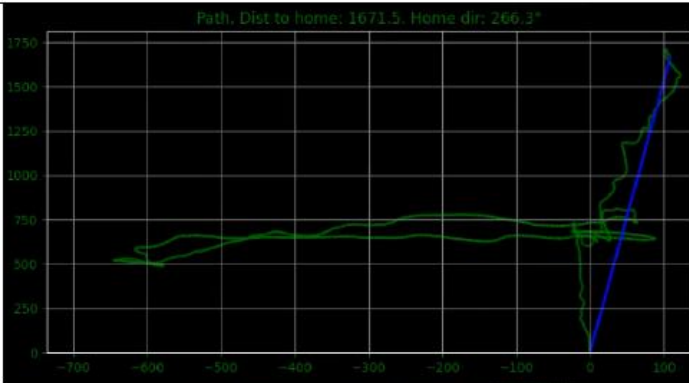

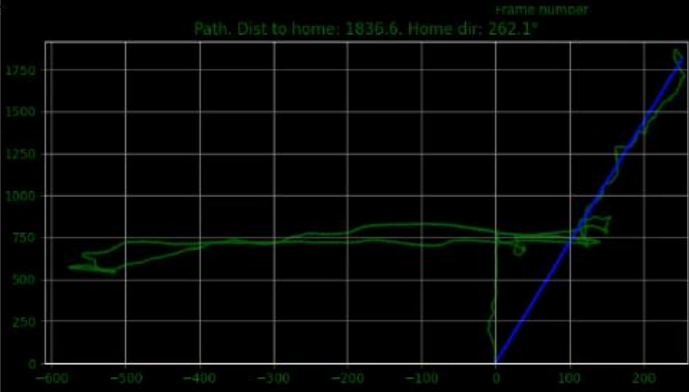
Маршрут	Коефіцієнт фільтрації	Коефіцієнт згладжування
	Без фільтрації	Без згладжування
	0,1	0,1
	0,1	0,2
	0,1	0,4

	<p>0,1</p>	<p>0,6</p>
	<p>0,2</p>	<p>0,2</p>
	<p>0,4</p>	<p>0,2</p>

Таблиця 4

RAFT алгоритм

Маршрут	Назва моделі
	<p>Kitty</p>

	Sintel
	Small
	Things

Як можна бачити, згладжування та фільтрація майже не мали вплив на побудований маршрут, проте найбільший вплив на пройдену відстань по прямій було в розрідженому алгоритмі. Такий на перший погляд випадковий розподіл відстаней можна пояснити тим, що навіть невеликі коригування в маршрут можуть потенційно мати великий вплив на пройдену відстань.

Підхід RAFT продемонстрував достатньо точний маршрут, близький до щільного алгоритму, на якому він і заснований.

Найгірший маршрут був побудований розрідженим алгоритмом. Даний алгоритм, попри фільтрацію та згладжування, видавав хаотичні рухи, які у свою чергу мали значний вплив на форму побудованого маршруту та пройдену дистанцію по прямій.

Друге тестування було проведено без фільтрації та згладжування, змінюючи параметри застосування розрідженого та щільного алгоритмів.

Бібліотека OpenCV має імплементацію щільного алгоритму Фарнебека у вигляді функції `calcOpticalFlowFarneback`, яка приймає наступні параметри:

- 1) `pyr_scale` – визначає масштаб піраміди, тобто те, у скільки буде зменшено кожне наступне зображення;
- 2) `levels` – кількість рівнів піраміди зображень;
- 3) `winsize` – розмір квадрату, для якого обчислюється оптичний потік за одну ітерацію.

Імплементація розрідженого алгоритму реалізована функцією calcOpticalFlowPyrLK, яка приймає наступні параметри:

- 1) winSize – розмір вікна пошуку на кожному рівні піраміди;
- 2) maxLevel - максимальний рівень піраміди;

3) criteria – параметр, що визначає критерії завершення ітераційного алгоритму пошуку (після заданої максимальної кількості ітерацій criteria.maxCount або коли вікно пошуку зміщується менше ніж на criteria.epsilon).

Друге тестування (Рисунок 5) складається з відео руху камери по горизонталі на висоті 10 см. Максимальна відстань становить 99 см, кінцева відстань – 57,5 см. Оскільки алгоритми оптичного потоку не можуть видавати точні значення пройдені відстані, порівняння будуть відбуватись як відношення максимальної відстані до кінцевої. Коефіцієнт повної відстані та відстані по закінченню: $99 \text{ см} / 57,5 \text{ см} \approx 1,722$.

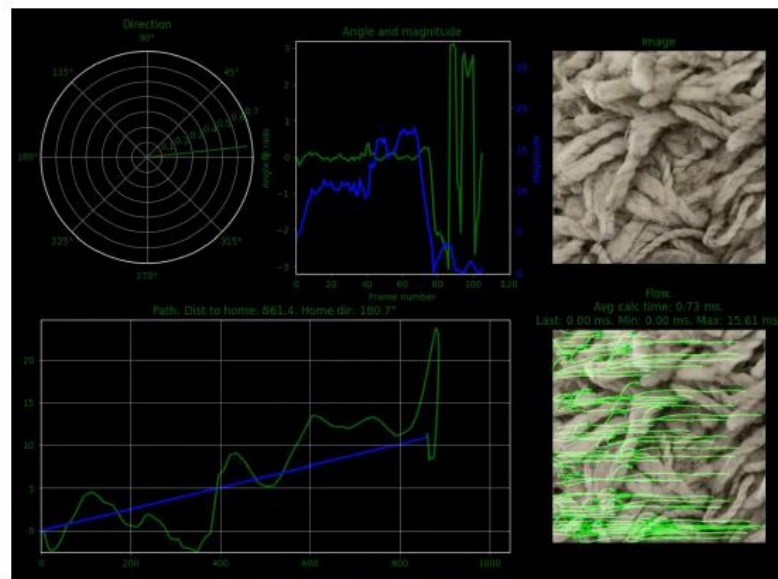


Рис. 5. Приклад2 виводу даних

Проведемо порівняльний аналіз щільного алгоритму, розрідженого алгоритму та RAFT-алгоритму – Таблиці 5-7.

Таблиця 5

Щільний алгоритм

Коефіцієнт відстані	Мін. час обчислення	Макс. час обчислення	Серед. час обчислення	pyr_scale	levels	winsize
1,06	16,26 мс	122,78 мс	18,44 мс	0,7	1	300
1,04	14,51 мс	39,72 мс	20,15 мс	0,2	1	300
1,065	14,52 мс	56,17 мс	21,14 мс	0,5	3	300
1,058	14,49 мс	43,22 мс	17,34 мс	0,5	5	300
1,04	15,01 мс	53,02 мс	26,14 мс	0,5	1	200
1,066	14,51 мс	19,3 мс	19,39 мс	0,5	1	50

Таблиця 6

Розріджений алгоритм

Коефіцієнт відстані	Мін. час обчислення	Макс. час обчислення	Серед. час обчислення	win_size	maxlevel	criteria.maxCount, criteria.epsilon
1,68	0,009 мс	15,61 мс	0,56 мс	(30, 30)	2	30, 0,03
1,71	0,009 мс	6,9 мс	1,49 мс	(60, 60)	2	30, 0,03
1,63	0,51 мс	10,9 мс	3,33 мс	(120, 120)	2	30, 0,03
1,74	0,009 мс	2,01 мс	0,27 мс	(15, 15)	0	30, 0,03
1,72	0,009 мс	2,03 мс	0,54 мс	(15, 15)	4	30, 0,03
1,72	0,009 мс	2,89 мс	0,54 мс	(15, 15)	8	30, 0,03
1,74	0,009 мс	2,05 мс	0,41 мс	(15, 15)	2	10, 0,03
1,67	0,009 мс	2,01 мс	0,41 мс	(15, 15)	2	60, 0,03
1,72	0,009 мс	2 мс	0,43 мс	(15, 15)	2	120, 0,03
1,74	0,009 мс	2,05 мс	0,41 мс	(15, 15)	2	30, 0,01
1,77	0,009 мс	1,51 мс	0,37 мс	(15, 15)	2	30, 0,06
1,76	0,009 мс	2 мс	0,41 мс	(15, 15)	2	30, 0,12

Таблиця 7

RAFT алгоритм				
Коефіцієнт відстані	Мін. час обчислення	Макс. час обчислення	Серед. час обчислення	Назва моделі
1,75	31,78 мс	269,99 мс	53,33 мс	Chairs
1,77	46,91 мс	66 мс	53,63 мс	Kitti
1,77	53,93 мс	46,97 мс	53,93 мс	Sintel
1,74	24 мс	88,21 мс	39,89 мс	Small
1,77	46,92 мс	130,4 мс	53,60 мс	Things

Найкраще себе показали розріджений та RAFT алгоритми. Для розрідженого алгоритму було визначено, що найбільший вплив на точність визначення пройденої відстані та часу виконання алгоритму впливає розмір вікна та збільшення кількості рівнів піраміди.

Для RAFT найкраще себе показала зменшена модель, яка в середньому в 1,3 рази швидше виконує алгоритм та показує точніший результат.

Щільний алгоритм показав найгірший результат в цьому тестуванні, пропри зміни в параметрах.

Найкращим варіантом для імплементації оптичного потоку буде саме RAFT зі зменшеною моделлю виконання. Попри відносно високий час обчислення, даний метод показав себе найбільш стабільним у двох тестах. Час виконання алгоритму можна покращити, використавши спеціальні обчислювальні модулі NPU, чи TPU.

ВИСНОВКИ З ДАНОГО ДОСЛІДЖЕННЯ

І ПЕРСПЕКТИВИ ПОДАЛЬШИХ РОЗВІДОК У ДАНОМУ НАПРЯМІ

1) Стрімкий розвиток засобів радіоелектронної боротьби (РЕБ) у сучасних конфліктах створює критичну потребу в альтернативних системах навігації для БПЛА, які здатні функціонувати без сигналу GNSS. Розроблена концепція інтелектуальної кіберфізичної системи базується на принципах відносного позиціонування, що об'єднує дані акселерометра, гіроскопа, магнітометра, барометра, лідара та камери оптичного потоку. Такий мультисенсорний підхід у поєднанні з розширеним фільтром Калмана дозволяє досягти необхідної точності та відмовостійкості, забезпечуючи автономність апарату від зовнішніх мап чи рельєфу місцевості.

2) Особлива увага в роботі приділена алгоритмам обчислення оптичного потоку як ключового інструменту визначення пройденої відстані та напрямку руху. Порівняльний аналіз розрідженого, щільного та нейромережевого (RAFT) методів показав, що традиційний розріджений алгоритм, попри високу швидкість, схильний до накопичення похибок через хаотичні коливання. Впровадження тривірневої системи фільтрації та експоненційного згладжування (зокрема для синусів і косинусів кутів) дозволило частково нівелювати вплив шумів зображення, проте не усунуло проблему низької стабільності маршруту для розрідженого методу.

3) Результати тестування продемонстрували, що найбільш перспективним рішенням для імплементації в системи навігації БПЛА є алгоритм RAFT. Використання його зменшеної моделі забезпечує оптимальний баланс між обчислювальною складністю та точністю, демонструючи найбільш стабільні результати відтворення маршруту в різних умовах. Хоча час обробки даних нейромережевими моделями залишається відносно високим, подальша оптимізація системи можлива завдяки використанню спеціалізованих обчислювальних модулів, таких як NPU або TPU.

Література

1. Distributed Control of Cyber Physical System on Various Domains: A Critical Review / M. Hamzah et al. Systems. 2023. Vol. 11, no. 4. P. 208. URL: <https://doi.org/10.3390/systems11040208>
2. Cooperative motion planning and control for aerial-ground autonomous systems: Methods and applications / R. Chai et al. Progress in Aerospace Sciences. 2024. Vol. 146. P. 101005. URL: <https://doi.org/10.1016/j.paerosci.2024.101005>
3. Unmanned Aerial Vehicles: Control Methods and Future Challenges / Z. Zuo et al. IEEE/CAA Journal of Automatica Sinica. 2022. Vol. 9, no. 4. P. 601–614. URL: <https://doi.org/10.1109/jas.2022.105410>
4. Devising a traffic control method for unmanned aerial vehicles with the use of gNB-IOT in 5G / H. Q. Nameer et al. Eastern-European Journal of Enterprise Technologies. 2022. Vol. 3, no. 9 (117). P. 56–59. URL: <https://doi.org/10.15587/1729-4061.2022.260084>
5. Improving Quality-of-Service in Cluster-Based UAV-Assisted Edge Networks / T. Bose et al. IEEE Transactions on Network and Service Management. 2022. P. 1. URL: <https://doi.org/10.1109/tnsm.2021.3139892>
6. Damigos G., Lindgren T., Nikolakopoulos G. Toward 5G Edge Computing for Enabling Autonomous Aerial Vehicles. IEEE Access. 2023. Vol. 11. P. 3926–3941. URL: <https://doi.org/10.1109/access.2023.3235067>
7. Sonugur G. A Review of quadrotor UAV: Control and SLAM methodologies ranging from conventional to innovative approaches. Robotics and Autonomous Systems. 2022. P. 104342. URL: <https://doi.org/10.1016/j.robot.2022.104342>

8. Peris C., Norton M., Khoo S. Y. Variations in Finite-Time Multi-Surface Sliding Mode Control for Multirotor Unmanned Aerial Vehicle Payload Delivery with Pendulum Swinging Effects. *Machines*. 2023. Vol. 11, no. 9. P. 899. URL: <https://doi.org/10.3390/machines11090899>
9. A Weight-Based Group Decision Controller for Air-Ground Collaborative Multi-unmanned Systems / H. Liu et al. *Lecture Notes in Electrical Engineering*. Singapore, 2025. P. 182–191. URL: https://doi.org/10.1007/978-981-96-2204-7_18
10. Exposing Spoofing Attack on Flocking-Based Unmanned Aerial Vehicle Cluster: A Threat to Swarm Intelligence / X. Huang et al. *Security and Communication Networks*. 2020. Vol. 2020. P. 1–15. URL: <https://doi.org/10.1155/2020/8889122>
11. Teed Z., Deng J. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. URL: https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123470392.pdf
12. Isk S., Kang S. Z., Lack Z. Optical Flow Models and Training Techniques in Data-Constrained Environment. URL: <https://cs231n.stanford.edu/reports/2022/pdfs/87p.pdf>
13. Babenko A. Hidden state refinement for optical flow forecasting. URL: <https://er.ucu.edu.ua/server/api/core/bitstreams/438f772e-0b7f-423e-9b67-ecd8ebd8d5d9/content>

References

1. Distributed Control of Cyber Physical System on Various Domains: A Critical Review / M. Hamzah et al. *Systems*. 2023. Vol. 11, no. 4. P. 208.
2. Cooperative motion planning and control for aerial-ground autonomous systems: Methods and applications / R. Chai et al. *Progress in Aerospace Sciences*. 2024. Vol. 146. P. 101005.
3. Unmanned Aerial Vehicles: Control Methods and Future Challenges / Z. Zuo et al. *IEEE/CAA Journal of Automatica Sinica*. 2022. Vol. 9, no. 4. P. 601–614.
4. Devising a traffic control method for unmanned aerial vehicles with the use of gNB-IOT in 5G / H. Q. Nameer et al. *Eastern-European Journal of Enterprise Technologies*. 2022. Vol. 3, no. 9 (117). P. 56–59.
5. Improving Quality-of-Service in Cluster-Based UAV-Assisted Edge Networks / T. Bose et al. *IEEE Transactions on Network and Service Management*. 2022. P. 1.
6. Damigos G., Lindgren T., Nikolakopoulos G. Toward 5G Edge Computing for Enabling Autonomous Aerial Vehicles. *IEEE Access*. 2023. Vol. 11. P. 3926–3941.
7. Sonugur G. A Review of quadrotor UAV: Control and SLAM methodologies ranging from conventional to innovative approaches. *Robotics and Autonomous Systems*. 2022. P. 104342.
8. Peris C., Norton M., Khoo S. Y. Variations in Finite-Time Multi-Surface Sliding Mode Control for Multirotor Unmanned Aerial Vehicle Payload Delivery with Pendulum Swinging Effects. *Machines*. 2023. Vol. 11, no. 9. P. 899.
9. A Weight-Based Group Decision Controller for Air-Ground Collaborative Multi-unmanned Systems / H. Liu et al. *Lecture Notes in Electrical Engineering*. Singapore, 2025. P. 182–191.
10. Exposing Spoofing Attack on Flocking-Based Unmanned Aerial Vehicle Cluster: A Threat to Swarm Intelligence / X. Huang et al. *Security and Communication Networks*. 2020. Vol. 2020. P. 1–15.
11. Teed Z., Deng J. RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. URL: https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123470392.pdf
12. Isk S., Kang S. Z., Lack Z. Optical Flow Models and Training Techniques in Data-Constrained Environment. URL: <https://cs231n.stanford.edu/reports/2022/pdfs/87p.pdf>
13. Babenko A. Hidden state refinement for optical flow forecasting. URL: <https://er.ucu.edu.ua/server/api/core/bitstreams/438f772e-0b7f-423e-9b67-ecd8ebd8d5d9/content>

ДОДАТОК В (обов'язковий)

Презентація

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Кафедра комп'ютерної інженерії та інформаційних систем



Метод та інтелектуальна кіберфізична система для навігації БПЛА в умовах дії засобів РЕБ противника

Здобувач: Владислав СЕВОСТЬЯНОВ

Науковий керівник: канд.-техн. наук, доцент. Катерина БЕРЕЗЬКА

Хмельницький - 2026

МЕТА ДОСЛІДЖЕННЯ

Метою кваліфікаційної роботи магістра є розробка методу та інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника.

Об'єктом дослідження є навігація БПЛА в умовах дії засобів РЕБ противника.

Предметом дослідження є застосування методу оптичного потоку та методу розширеної фільтрації Калмана для навігації БПЛА в умовах дії засобів РЕБ противника.

ЗАДАЧІ ДОСЛІДЖЕННЯ

Поставлена мета досягається розв'язанням таких основних завдань:

- аналіз існуючих методів навігації та позиціонування в умовах відсутності GNSS-сигналу;
- вибір найбільш підходящого методу обчислення оптичного потоку;
- розробка математичної моделі системи з використанням методу розширеного фільтру Калмана;
- розробка методу та програмного забезпечення інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника;
- здійснення експерименту для перевірки розробленого методу навігації на основі розробленого програмного забезпечення інтелектуальної кіберфізичної системи навігації БПЛА.

НАУКОВА НОВИЗНА ТА ПРАКТИЧНА ЦІННІСТЬ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Наукова новизна отриманих результатів:

- розроблено метод відносного позиціонування БПЛА в умовах роботи засобів РЕБ противника, який об'єднує кілька датчиків, включно з датчиком оптичного потоку на основі рекурентної моделі та фільтрує значення методом розширеного фільтру Калмана;
- набула подальшого розвитку інтелектуальна кіберфізична система навігації БПЛА в умовах відсутності GNSS-сигналу.

Практична значимість отриманих результатів полягає у розробленому програмному забезпеченні та інтелектуальної кіберфізичної системи позиціонування БПЛА в умовах роботи засобів РЕБ противника.

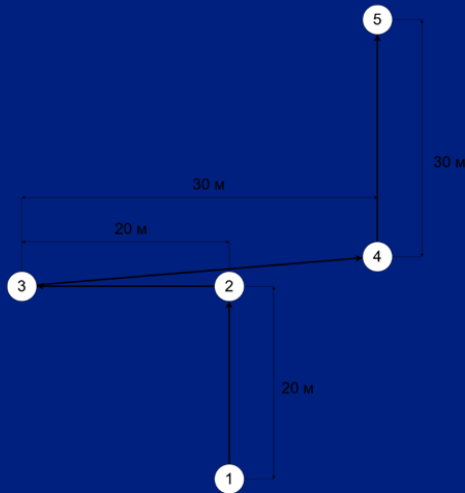
АКТУАЛЬНІСТЬ ДОСЛІДЖЕННЯ

- ❑ На сьогодні система GPS (Global Positioning System) є найбільш популярною системою навігації у таких сферах як авіація, споживацькі пристрої, портативні навігаційні пристрої, автомобільна сфера та інші. Як наслідок, очікується значне збільшення ринку систем глобального позиціонування.
- ❑ Попри високу точність та поширеність систем глобального, залишається проблема залежності системи позиціонування від супутникового сигналу, який постійно піддається «глушінню», особливо в умовах війни.
- ❑ Проблемою стає потреба використання навігаційних систем для БПЛА в умовах, де відсутній GNSS-сигнал.

АНАЛІЗ ВІДОМИХ МЕТОДІВ

- ❑ Проаналізувавши відомі методи навігації БПЛА в умовах відсутності GNSS-сигналу було визначено основні вимоги до обраного методу позиціонування, який має бути достатнім для визначення відносної позиції апарату для можливого повернення до точки зльоту.
- ❑ Для вирішення задачі позиціонування в реальному часі пропонується об'єднувати дані з різних джерел одночасно, а саме: дані з барометра та лідара для визначення висоти над поверхнею; дані з акселерометра та гіроскопа для визначення положення апарату в просторі; дані з камери оптичного потоку та компаса для ефективного визначення напрямку та швидкості руху апарату.
- ❑ Пропонується також використання методу розширеного фільтру Калмана для уточнення позиції та фільтрації шумів з датчиків. Це дасть змогу зробити систему завадостійкою та попередити можливі збої роботи пропонованих датчиків.

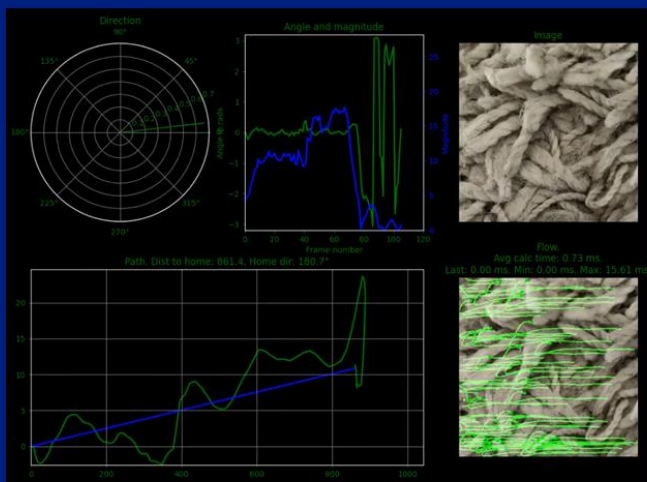
ВИБІР ОПТИМАЛЬНОГО МЕТОДУ ОБЧИСЛЕННЯ ОПТИЧНОГО ПОТОКУ



Схематичне зображення маршруту для першого тесту

Було проведено серію тестів для вибору оптимального алгоритму обчислення оптичного потоку. Перше тестування включає в себе відео польоту дрона на висоті 25 метрів та опущеною камерою донизу. Тут було протестовано вплив згладжування та фільтрації на щільний та розріджений алгоритми.

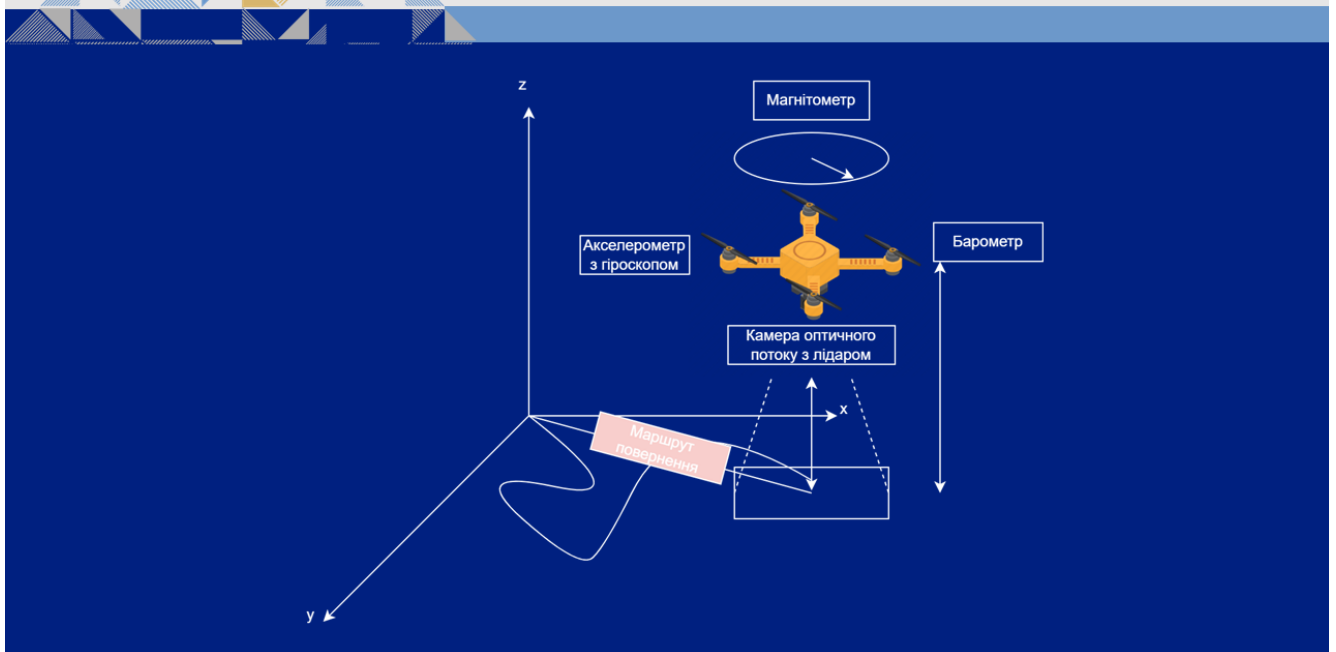
ВИБІР ОПТИМАЛЬНОГО МЕТОДУ ОБЧИСЛЕННЯ ОПТИЧНОГО ПОТОКУ



Друге тестування складалось з відео руху камери по горизонталі на висоті 10 см. Максимальна відстань становить 99 см, кінцева відстань – 57,5 см.

За результатами обох тестувань було визначено, що найкращим методом обчислення оптичного потоку розроблюваної системи є алгоритм на основі рекурентної моделі RAFT.

КОНЦЕПТУАЛЬНА СХЕМА СИСТЕМИ ПОЗИЦІОНУВАННЯ



МЕТОД НАВІГАЦІЇ БПЛА В УМОВАХ РОБОТИ ЗАСОБІВ РЕБ ПРОТИВНИКА

Метод навігації БПЛА в умовах роботи засобів РЕБ противника складається з трьох циклічно повторюваних основних етапів:

1. Збір даних з наступних датчиків: акселерометр, гіроскоп, лідар, барометр, магнітометр та датчик оптичного потоку на основі моделі штучного інтелекту RAFT.
2. Визначення поточного стану системи та передбачення наступного з допомогою розширеного фільтра Калмана. Коригування шумів з датчиків.
3. Побудова та збереження пройденого маршруту відносно точки зльоту.

МЕТОД НАВІГАЦІЇ БПЛА В УМОВАХ РОБОТИ ЗАСОБІВ РЕБ ПРОТИВНИКА

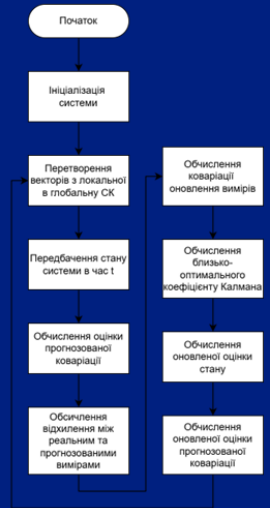
Алгоритм збору даних з давачів



Алгоритм обробки даних з давачів

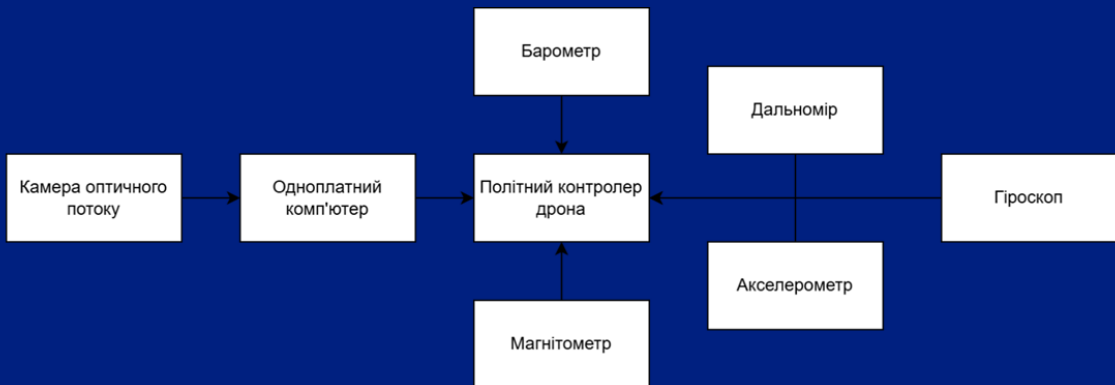


Блок-схема алгоритму обчислення стану системи



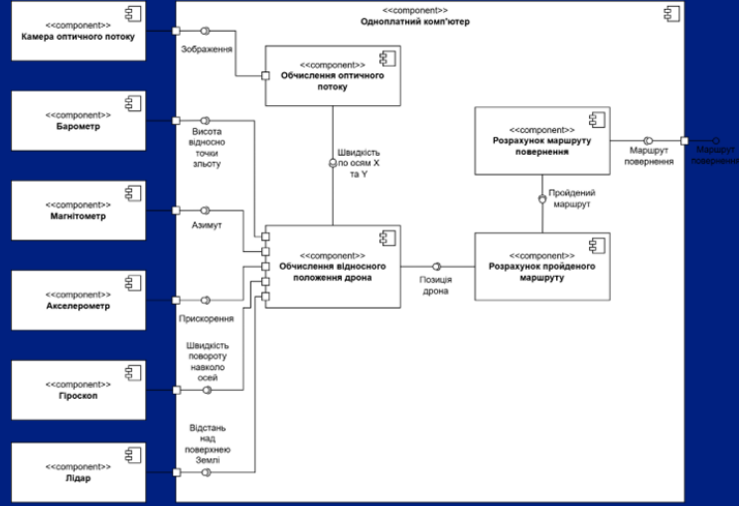
РЕАЛІЗАЦІЯ МЕТОДУ

Структура інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника



РЕАЛІЗАЦІЯ МЕТОДУ

Діаграма компонентів інтелектуальної кіберфізичної системи навігації БПЛА в умовах дії засобів РЕБ противника



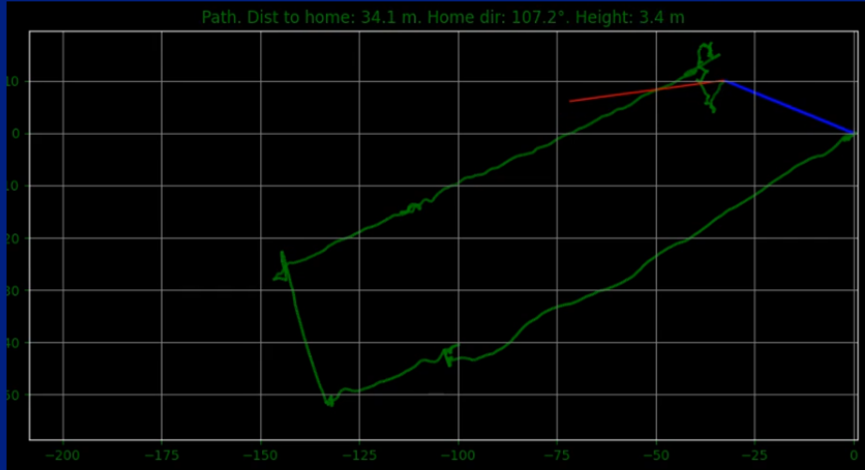
ЕКСПЕРИМЕНТ

Дрон носій ArmDrones для проведення експерименту



ЕКСПЕРИМЕНТ

Побудований маршрут інтелектуальною кіберфізичною системою для навігації БПЛА в умовах дії засобів РЕБ противника



ЕКСПЕРИМЕНТ

Схематичне зображення пройденого маршруту



Схема побудованого маршруту інтелектуальною кіберфізичною системою для навігації БПЛА в умовах дії засобів РЕБ противника



ВИСНОВКИ

У роботі за результатами виконаних теоретичних та практичних досліджень розроблено метод та інтелектуальну кіберфізичну систему для навігації БПЛА в умовах дії засобів РЕБ противника. Подальшого розвитку система та метод об'єднання датчиків з використанням оптичного потоку з подальшою фільтрацією за допомогою методу розширеного фільтру Калмана.

За результатами роботи було проведено експеримент для визначення точності навігації розробленої інтелектуальної кіберфізичної системи. Визначена точність дозволяє використання даної системи для невеликих автономних місій польотів, де відсутній зв'язок GPS. Результати експерименту показали точність вимірювання пройденого маршруту в 89% порівняно з вимірним маршрутом модулем GPS.



ПУБЛІКАЦІЇ

- Березька К., Севостьянов В. ІНТЕЛЕКТУАЛЬНА КІБЕРФІЗИЧНА СИСТЕМА ДЛЯ НАВІГАЦІЇ БПЛА В УМОВАХ ДІЇ ЗАСОБІВ РЕБ ПРОТИВНИКА. *Вимірювальна та обчислювальна техніка в технологічних процесах*. 2026. 2. С. 28-39. Режим доступу: <https://doi.org/10.31891/2219-9365-2026-86-4>.

Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Владислав СЕВОСТЬЯНОВ

Співавтор:

Назва: Метод та інтелектуальна кіберфізична система для навігації БПЛА в умовах дії засобів РЕБ противника

Експерт: Катерина БЕРЕЗЬКА

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 9.14%

Коефіцієнт подібності 2: 1.91%

Мікропробіли: 3

Заміна букв: 6

Інтервали: 0

Білі знаки: 6

Дата створення звіту: 2026-04-27 16:21:22.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

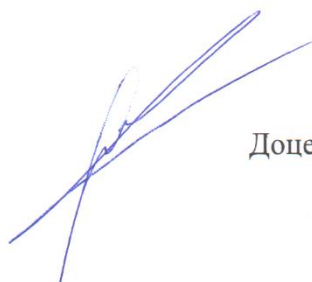
Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2026-04-27

Дата



Доцент Андрій Нічепорук

експерт

Anti-Plagiarism (<http://ap.km.ua>) v-15.701

Максимальне співпадіння з одним документом 13.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 15%

ID: 270728 Назва: МКР Метод та інтелектуальна кіберфізична система для навігації БПЛА в умовах дії засобів РЕБ противника Додано в БД: 2026-04-27 Автора: Владислав СЕВОСТЬЯНОВ Керівники: Катерина БЕРЕЗЬКА Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	72012	659	11419 (16%)	106 (16%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми
269835	Назва: Звіт з ПДП Метод та інтелектуальна кіберфізична система для навігації БПЛА в умовах дії засобів РЕБ противника Додано в БД: 2026-03-17 Автора: Сєвостьянова В. С. Керівники: Гнатчук Є.Г. Консультанти: Опоненти:	9375 (13.0%)	75 (11.0%)

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ МАГІСТРА

Здобувач: Владислав СЄВОСТЬЯНОВ

Тема: Метод та інтелектуальна кіберфізична система для навігації БПЛА в умовах дії засобів РЕБ противника

Спеціальність: 123 «Комп'ютерна інженерія»

Обсяг кваліфікаційної роботи магістра:

Кількість листів креслень —; кількість сторінок записки 74

1. Короткий зміст роботи та прийнятих рішень У роботі запропоновано метод інтелектуальної кіберфізичної системи для навігації БПЛА в умовах дії засобів РЕБ противника

2. Висновок про відповідність роботи дипломному завданню Кваліфікаційна робота магістра відповідає виданому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено аналіз існуючих методів позиціонування в умовах відсутності GNSS-сигналу. Досліджено відомі рішення та засоби в цій сфері. В другому розділі побудовано концепцію системи позиціонування, описано математичну модель для використання фільтрації Калмана, а також обрано оптимальний метод та модель обчислення оптичного потоку. У третьому розділі представлено метод та побудовано алгоритм роботи кіберфізичної системи навігації БПЛА У четвертому розділі на основі розробленого методу та алгоритму роботи системи проведено експеримент для визначення точності роботи програмного забезпечення інтелектуальної кіберфізичної системи навігації БПЛА в умовах роботи засобів РЕБ противника.

4. Позитивні сторони роботи: Запропонована система навігації БПЛА має достатню точність побудови маршруту та позиціонування для виконання нескладних автономних місій без використання GPS.

5. Негативні сторони роботи: Розроблена система має певні недоліки із стабілізацією зображення камери та похибку оптичного потоку, що спотворює отриманий маршрут БПЛА.

6. Оцінка графічного оформлення та пояснювальної записки роботи: —

7. Відгук про роботу в цілому: В загальному робота виконана на високому рівні.

8. Інші зауваження: —

9. Оцінка кваліфікаційної роботи магістра:

Розглянувши позитивні та негативні сторони представленої кваліфікаційної роботи магістра вважаю, що робота заслуговує оцінки «відмінно» 95.00 (А)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) —

Феріше Олександр Валерійович, р.т.ч., проф., зав. каф ХН ХНУ

“ 1 травня ” 2026р.



Зав. кафедри КПС
д-р. філософії Ользі ПАВЛОВІЙ

Владислав СЄВОСТЬЯНОВ

ПІБ здобувача вищої освіти

ФІТ, 2 курсу, групи КІ2М-24-1

ЗАЯВА


З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

01.05.2026

Дата



Підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Метод та інтелектуальна кіберфізична система для навігації БПЛА в умовах дії засобів РЕБ противника

Автор Владислав СЕВОСТЬЯНОВ

Освітня програма Інформаційні системи та технології

Рівень вищої освіти другий (магістерський)

Спеціальність 123 Комп'ютерна інженерія

Науковий керівник: к.т.н., доцент Катерина БЕРЕЗЬКА

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.
- 4) значна частина знайденого плагіату відноситься до списку використаних джерел

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 9,14% і адресується до 74 першоджерел; та системою Anti-Plagiarism складає 13%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

27.04.2026

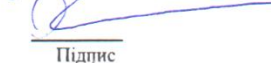
Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи



Підпис



Підпис



Підпис

Ольга ПАВЛОВА
Ім'я, ПРІЗВИЩЕ

Олег САВЕНКО
Ім'я, ПРІЗВИЩЕ

Катерина БЕРЕЗЬКА
Ім'я, ПРІЗВИЩЕ