ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «СИСТЕМА ДЛЯ ПЕРЕХОПЛЕННЯ ПОВІТРЯНИХ ЦІЛЕЙ НА БАЗІ ШТУЧНОГО ІНТЕЛЕКТУ»

на здобуття освітнього ступеня магістра
зі спеціальності 123 Комп'ютерної інженерії
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерні системи та мережі
(назва)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ *Вадим ОМЕЛЬЧУК*
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав: здобувач вищої освіти гр. КСДМ-61
Вадим ОМЕЛЬЧУК
Ім'я, ПРІЗВИЩЕ

Керівник: Кирило СОКОЛОВ
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Рецензент: _____
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут Інформаційних технологій

Кафедра 123 Комп'ютерної інженерії
Ступінь вищої освіти Магістр
Спеціальність 123 Комп'ютерної інженерії
Освітньо-професійна програма Комп'ютерні системи та мережі

ЗАТВЕРДЖУЮ
Завідувач кафедрою Наталія ЛАЩЕВСЬКА
Ім'я, ПРІЗВИЩЕ
«_____» _____ 2025р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Омельчук Вадим Васильович
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Система для перехоплення повітряних цілей на базі штучного інтелекту
керівник кваліфікаційної роботи Кирило СОКОЛОВ,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «30» 10 2025р. № 467
2. Строк подання кваліфікаційної роботи «_____» _____ 2025р.
3. Вихідні дані до кваліфікаційної роботи: Науково-технічна література, методи теорії автоматичного керування, імовірнісні методи оцінки стану, методи глибокого навчання та комп'ютерного моделювання.
Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Актуальні методи та проблематика автономного переміщення безпілотних літальних апаратів
2. Теоретико-методологічні аспекти побудови систем автономного керування безпілотними літальними апаратами
3. Проєктування інтегрованої системи автономного наведення бпла
4. Перелік ілюстративного матеріалу: *презентація*
5. Дата видачі завдання «_____» _____ 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	25.09-24.10.2025	Виконано
2	Обробка матеріалу	25.10-05.11.2025	Виконано
3	Розробка теоретичної частини	06.11-10.11.2025	Виконано
4	Розробка системи	11.11-15.11.2025	Виконано
5	Висновки за результатами аналізу	15.11-19.11.2025	Виконано
6	Оформлення пояснювальної записки	19.11-21.11.2025	Виконано
7	Розробка презентації	21.11-29.11.2025	Виконано
8	Передзахист	01.12-02.12.2025	Виконано
9	Здача в деканат	15.12-18.12.2025	Виконано

Здобувач вищої освіти _____ Вадим ОМЕЛЬЧУК
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Кирило СОКОЛОВ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 72 стор., 29 рис., 10 табл., 77 джерел.

Мета дослідження – підвищення ефективності систем протиповітряної оборони шляхом створення автономного безпілотного перехоплювача, керованого штучним інтелектом, здатного виявляти та вражати повітряні цілі без використання глобальних навігаційних супутникових систем.

Об'єкт дослідження – системи автономної навігації, технічного зору та керування рухом безпілотних літальних апаратів.

Предмет дослідження – алгоритми сенсорної фузії, нейромережеві методи детекції об'єктів та методи планування траєкторії для динамічного перехоплення.

Короткий зміст роботи: У кваліфікаційній роботі проведено детальний аналіз методів автономної орієнтації БПЛА. Обґрунтовано неефективність класичних GPS-систем в умовах РЕБ та запропоновано гібридний підхід на основі інерціальних датчиків та оптичного потоку. Розроблено математичну модель руху перехоплювача з використанням фільтра Калмана для прогнозування стану. Реалізовано модуль комп'ютерного зору на базі нейромережі YOLO, що дозволяє виконувати детекцію та супровід цілей у відеопотоці. Спроектовано архітектуру програмного забезпечення в середовищі ROS, яка об'єднує підсистеми SLAM, планування маршруту та безпосереднього керування польотом.

Результати: Створено та верифіковано у симуляторі Gazebo модель інтелектуальної системи перехоплення. Система демонструє здатність стабільно утримувати ціль у полі зору, прогнозувати її траєкторію та виконувати автономне наведення з урахуванням перешкод, що підтверджено серією віртуальних експериментів.

КЛЮЧОВІ СЛОВА: БПЛА, ПЕРЕХОПЛЕННЯ ЦІЛЕЙ, ШТУЧНИЙ ІНТЕЛЕКТ, КОМП'ЮТЕРНИЙ ЗІР, YOLO, ФІЛЬТР КАЛМАНА, ROS, SLAM, АВТОНОМНА НАВІГАЦІЯ, СЕНСОРНА ФУЗІЯ.

ABSTRACT

Textual part of the Master’s thesis: 72 pages, 29 figures, 10 tables, 77 references.

The aim of the research is to enhance the efficiency of air defense systems by creating an autonomous AI-controlled unmanned interceptor capable of detecting and engaging aerial targets without reliance on global navigation satellite systems (GNSS).

The object of the research is the systems for autonomous navigation, computer vision, and motion control of unmanned aerial vehicles (UAVs).

The subject of the research covers sensor fusion algorithms, neural network-based object detection methods, and trajectory planning methods for dynamic interception.

Content summary: The thesis conducts a detailed analysis of methods for autonomous UAV orientation. The inefficiency of classical GPS systems under Electronic Warfare (EW) conditions is substantiated, and a hybrid approach based on inertial sensors and optical flow is proposed. A mathematical model of the interceptor's motion is developed using the Kalman filter for state prediction. A computer vision module based on the YOLO neural network is implemented, enabling target detection and tracking within the video stream. The software architecture is designed in the ROS environment, integrating SLAM subsystems, path planning, and direct flight control.

Results: A model of an intelligent interception system was created and verified in the Gazebo simulator. The system demonstrates the ability to stably keep the target within the field of view, predict its trajectory, and perform autonomous guidance while accounting for obstacles, as confirmed by a series of virtual experiments.

KEYWORDS: UAV, TARGET INTERCEPTION, ARTIFICIAL INTELLIGENCE, COMPUTER VISION, YOLO, KALMAN FILTER, ROS, SLAM, AUTONOMOUS NAVIGATION, SENSOR FUSION.

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

ПОДАННЯ
ГОЛОВІ ДЕРЖАВНОЇ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня магістра

Направляється здобувач Омельчук В. В. до захисту кваліфікаційної роботи
(прізвище та ініціали)
за спеціальністю 123 Комп'ютерна інженерія
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерні системи та мережі
(назва)
на тему: «Система для перехоплення повітряних цілей не базі штучного інтелекту».
Кваліфікаційна робота і рецензія додаються.

Директор ННІ _____ Катерина НЕСТЕРЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Висновок керівника кваліфікаційної роботи

Здобувач Омельчук В. В. показав відмінну теоретичну підготовку, уміння володіти та розуміння новими комп'ютерними технологіями, користуватися навчальною і науково-технічною літературою. Працюючи над завданнями, які були поставлені керівником, проявив ініціативність, сумлінність та хист до інженерної роботи.
Все це дозволяє оцінити кваліфікаційну роботу студента Омельчука Вадима Васильовича на оцінку «відмінно», та присвоїти їй кваліфікацію «магістр з комп'ютерної інженерії».

Керівник кваліфікаційної роботи _____ Кирило СОКОЛОВ
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

«_» _____ 2025 року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач *Омельчук В. В.* допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедрою Комп'ютерної інженерії _____ Наталія ЛАЩЕВСЬКА
(назва) *(підпис)* *(Ім'я, ПРІЗВИЩЕ)*

ВІДГУК РЕЦЕНЗЕНТА
на кваліфікаційну магістерську роботу

здобувача вищої освіти Омельчук Вадим Васильович
(прізвище, ім'я, по батькові)
на тему: «Система для перехоплення повітряних цілей не базі штучного інтелекту»

Актуальність:

Кваліфікаційна магістерська робота присвячена вирішенню надзвичайно гострої проблеми сучасності — захисту повітряного простору від малорозмірних безпілотних апаратів. В умовах, коли традиційні засоби ППО є економічно невигідними для знищення дешевих дронів, а канали керування та GPS часто придушуються засобами РЕБ, розробка автономних перехоплювачів на базі штучного інтелекту набуває критичного значення. Робота пропонує комплексний підхід, що поєднує сучасні методи робототехніки та глибокого навчання для створення ефективного засобу протидії повітряним загрозам.

Позитивні сторони:
1. Автором проведено глибокий аналіз математичного апарату навігації. Використання розширеного фільтра Калмана для комплексування даних від IMU та систем візуальної одометрії свідчить про високий інженерний рівень роботи. Це дозволяє системі працювати стабільно навіть при накопиченні похибок окремих сенсорів.
2. Вдало обґрунтовано вибір програмного стека. Використання мови Python у поєднанні з бібліотеками OpenCV та фреймворками для нейромереж забезпечує гнучкість розробки. Інтеграція з ROS дозволяє створити модульну архітектуру, яку легко масштабувати або переносити на різне апаратне забезпечення.
3. Вагомим результатом є практична реалізація віртуального прототипу в симуляторі Gazebo. Автор не обмежився теоретичними викладками, а продемонстрував працездатність алгоритмів планування маршруту та уникнення перешкод у динамічному середовищі. Реалізація сценаріїв "відмови GPS" підтверджує надійність запропонованих рішень.

Недоліки:
1. У роботі основна увага приділена програмним алгоритмам та симуляції, проте недостатньо детально розглянуто питання енергетичного балансу системи. Процеси обробки відеопотоку нейромережами та розрахунки SLAM вимагають значних обчислювальних потужностей, що може суттєво вплинути на час автономного польоту реального перехоплювача.
2. Бажано було б навести детальніший аналіз аеродинамічних обмежень при перехопленні високошвидкісних цілей, оскільки модель руху в симуляції може бути дещо спрощеною порівняно з реальною фізикою польоту при різких маневрах на великих швидкостях.

Висновок: *кваліфікаційна бакалаврська робота заслуговує оцінку "відмінно", а здобувач Омельчук В. В. заслуговує присвоєння кваліфікації: магістр з комп'ютерної інженерії.*

Рецензент:
науковий ступінь, вчене звання _____ *підпис* _____ *Ім'я, ПРІЗВИЩЕ*

ЗМІСТ

ВСТУП 10
1 АКТУАЛЬНІ МЕТОДИ ТА ПРОБЛЕМАТИКА АВТОНОМНОГО ПЕРЕМІЩЕННЯ БЕЗПІЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ 12
1.1 Аналіз комплексних засобів просторової орієнтації та локалізації безпілотників 12
1.2 Алгоритмічні методи комплексування різнорідної сенсорної інформації: фільтрація Калмана та інерціальне числення 17
1.3 Інтеграція методів комп'ютерного зору та алгоритмів SLAM для просторової орієнтації 24
1.4 Роль інтелектуальних систем у забезпеченні автономності безпілотних апаратів 29
2 ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ АСПЕКТИ ПОБУДОВИ СИСТЕМ АВТОНОМНОГО КЕРУВАННЯ БЕЗПІЛОТНИМИ ЛІТАЛЬНИМИ АПАРАТАМИ 36
2.1 Формалізація та визначення ключових завдань навігації в контексті автономних безпілотних систем 36
2.2 Аналіз апаратно-програмних засобів та алгоритмічних стратегій реалізації автономного польоту 38
2.3 Обґрунтування вибору інструментальних засобів та програмного стека реалізації системи 42
2.4 Методологія віртуального прототипування та верифікації алгоритмів керування 46
2.5 Стратегії виключення оператора з контуру керування та алгоритмічне забезпечення автономності 49
3 ПРОЄКТУВАННЯ ІНТЕГРОВАНОЇ СИСТЕМИ АВТОНОМНОГО НАВЕДЕННЯ БПЛА 52
3.1 Апаратна архітектура та сенсорне забезпечення навігації 52

3.2 Математичні методи оцінки стану та картографування 57
3.3 Алгоритми планування шляху та розпізнавання об’єктів..... 59
ВИСНОВКИ..... 62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 65
Додаток А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація) 73

ВСТУП

У сучасному безпековому середовищі стрімке поширення безпілотних авіаційних систем створює нові виклики для захисту повітряного простору. Малорозмірні дрони, що використовуються для розвідки або нанесення ударів, часто залишаються непомітними для класичних радіолокаційних станцій через малу ефективну площу розсіювання та здатність рухатися на наднизьких висотах. Традиційні методи протидії, які вимагають участі людини-оператора, часто виявляються недостатньо швидкими або ефективними, особливо в умовах масованих атак або складної радіоелектронної обстановки. Це зумовлює нагальну потребу в розробці автономних перехоплювачів, здатних діяти за принципом "виявив – знищив" без постійного каналу зв'язку з наземною станцією.

Критично важливим аспектом створення таких систем є впровадження методів штучного інтелекту та комп'ютерного зору. Оскільки супутникова навігація може бути придушена засобами радіоелектронної боротьби противника, перехоплювач повинен покладатися на бортові системи орієнтації та візуального розпізнавання цілей. Використання нейромережевих алгоритмів дозволяє не лише ідентифікувати ворожий об'єкт на фоні складного ландшафту, але й прогнозувати траєкторію його руху для виконання точного маневру зближення.

Мета і завдання дослідження. Метою роботи є підвищення ефективності захисту повітряного простору шляхом розробки системи автономного керування для БПЛА-перехоплювача, що базується на методах машинного навчання, сенсорній фузії та алгоритмах комп’ютерного зору.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Провести аналіз існуючих методів автономної навігації та способів орієнтації безпілотників в умовах відсутності супутникового сигналу, розглянувши можливості інерціальних систем та візуальної одометрії.

2. Розробити математичну модель оцінки стану перехоплювача та цілі, використовуючи алгоритми фільтрації (зокрема розширений фільтр Калмана) для мінімізації шумів сенсорів та прогнозування траєкторії руху об'єктів.

3. Реалізувати підсистему технічного зору на базі згорткових нейронних мереж для детекції та класифікації повітряних цілей у реальному часі.

4. Обґрунтувати вибір апаратно-програмного стека для забезпечення швидкодії системи перехоплення.

5. Розробити алгоритми планування маршруту перехоплення (на основі A* або аналогів) з урахуванням динамічних перешкод та необхідності уникнення зіткнень з іншими об'єктами.

6. Створити віртуальний прототип системи та провести верифікацію алгоритмів керування і наведення у симуляційному середовищі Gazebo.

Об’єкт дослідження – процеси автономного керування, навігації та наведення безпілотних літальних апаратів.

Предмет дослідження – методи та алгоритми штучного інтелекту, комп’ютерного зору та сенсорної фузії для реалізації задачі перехоплення повітряних цілей.

Методи дослідження. У роботі застосовано методи теорії автоматичного керування, імовірнісні методи оцінки стану, методи глибокого навчання та комп’ютерного моделювання.

Практичне значення. Розроблена система забезпечує можливість створення відносно недорогих автономних перехоплювачів, здатних ефективно протидіяти ворожим дронам в умовах радіоелектронних перешкод, мінімізуючи ризики для особового складу та знижуючи вартість ураження цілі.

1 АКТУАЛЬНІ МЕТОДИ ТА ПРОБЛЕМАТИКА АВТОНОМНОГО ПЕРЕМІЩЕННЯ БЕЗПІЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ

1.1 Аналіз комплексних засобів просторової орієнтації та локалізації безпілотників

Критично важливою умовою для реалізації стабільного автономного польоту безпілотних літальних апаратів (БПЛА) є наявність надійних бортових систем, здатних з високою точністю визначати поточне місцезнаходження, орієнтацію та динамічні параметри руху пристрою. На сучасному етапі розвитку робототехніки базовий набір навігаційного обладнання зазвичай включає приймачі глобальних супутникових навігаційних систем (GPS), інерціальні вимірювальні модулі (IMU), магнітометричні датчики (компаси), а також просунуті алгоритми комп'ютерного зору. Кожна з цих підсистем має свої унікальні характеристики та сфери застосування, які варто розглянути детальніше.

Глобальна система позиціонування (GPS) залишається фундаментальним інструментом для отримання абсолютних географічних координат дрона. Її робота базується на принципі трилатерації сигналів від супутників, що дозволяє обчислити широту, довготу та висоту польоту. Ця технологія є незамінною при виконанні місій на відкритих просторах, де забезпечується пряма видимість супутникового сузір'я. Однак ефективність GPS суттєво знижується або стає неможливою в умовах щільної міської забудови (через ефект перевідбиття сигналу), у гірській місцевості, тунелях або всередині приміщень, де радіосигнал екранується перешкодами [1, 2]. Схематичне зображення процесу отримання навігаційних даних наведено на рисунку, рис. 1.1.

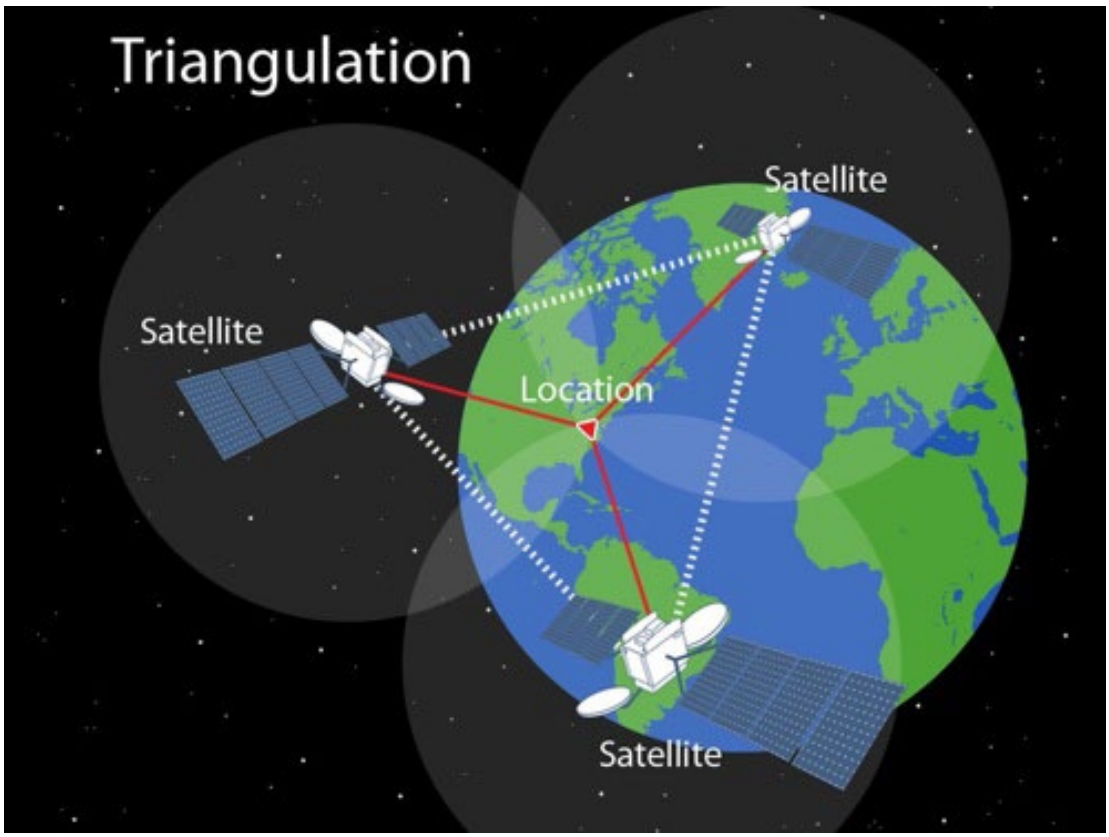


Рисунок 1.1 – Схематичне відображення принципу функціонування модуля супутникової навігації

Для автономної стабілізації та відстеження швидких змін положення використовується інерціальний вимірювальний блок (IMU). Цей вузол є комплексом мікроелектромеханічних сенсорів, що включає акселерометри для вимірювання лінійних прискорень по трьох осях, гіроскопи для фіксації кутових швидкостей обертання та, як правило, магнітометр для прив'язки до сторін світу. Обробка даних з IMU дозволяє обчислити положення дрона шляхом подвійного інтегрування прискорення. Втім, головним недоліком інерціальних систем є накопичувальний характер похибки, відомий як дрейф, що робить неможливим використання виключно IMU для тривалої навігації без зовнішньої корекції [3]. Структуру цього модуля продемонстровано на ілюстрації, рис. 1.2.

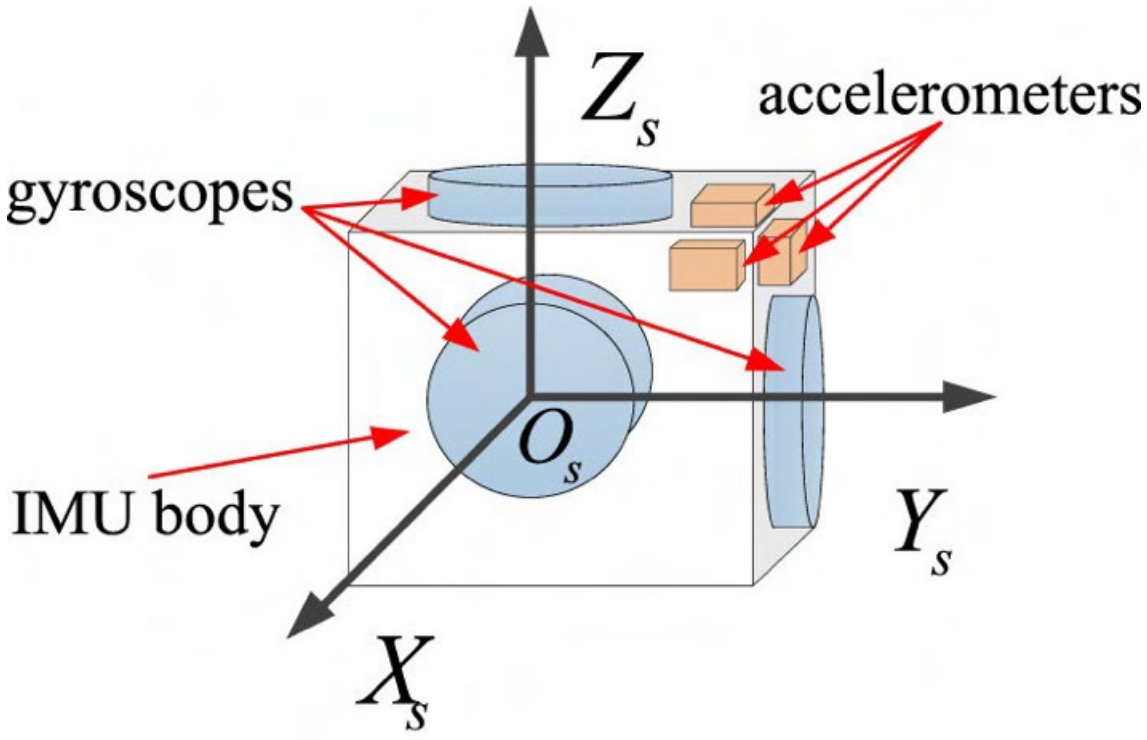


Рисунок 1.2 – Схема компонентів інерціальної вимірювальної системи

Зростаючу роль у сучасній навігації відіграють технології комп'ютерного зору, які надають дрону здатність "бачити" та аналізувати навколишнє середовище за допомогою відеокамер. Впровадження алгоритмів візуальної одометрії та методів одночасної локалізації і картографування (SLAM) дозволяє безпілотнику орієнтуватися у просторі навіть за повної відсутності супутникового сигналу. Система аналізує відеопотік, виділяє характерні ознаки об'єктів, будує карту місцевості в реальному часі та визначає власне положення відносно цієї карти [4, 5]. Розташування візуальних та інших сенсорів на корпусі апарата показано на схемі, рис. 1.3.

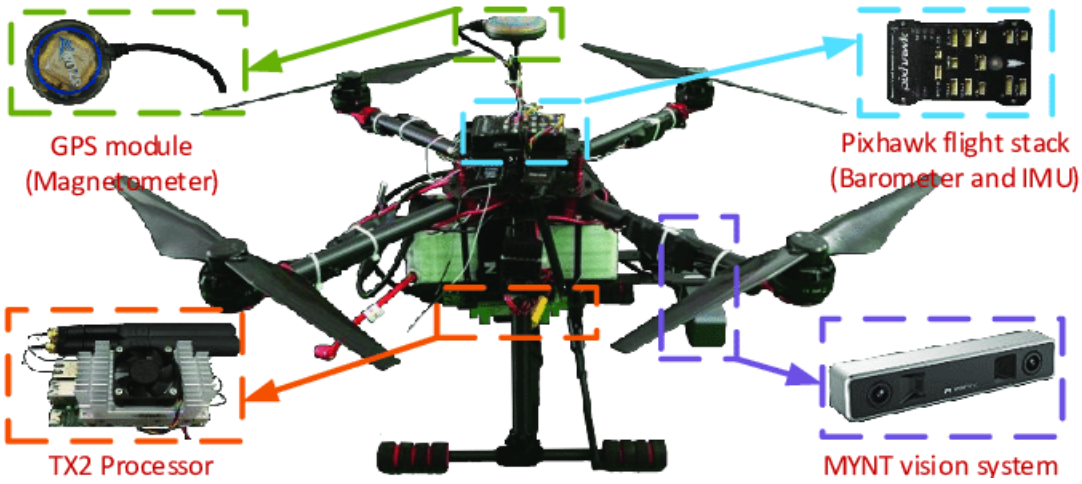


Рисунок 1.3 – Компонування основних сенсорних елементів на борту БПЛА

Окремим важливим напрямком у комп'ютерному зорі є метод оптичного потоку. Ця концепція описує видиме зміщення об'єктів у кадрі або патерни руху яскравості, що виникають внаслідок відносного переміщення камери та сцени. Оптичний потік є критично важливим для оцінки швидкості руху дрона, детектування перешкод та стабілізації положення у зависанні без використання GPS.

Фундаментальний принцип роботи алгоритмів оптичного потоку базується на припущенні про сталість яскравості відповідних точок на серії послідовних кадрів. Аналізуючи зміщення пікселів між двома або більше зображеннями, система генерує векторне поле швидкостей, де кожен вектор вказує напрямок та інтенсивність руху конкретної точки зображення. Для реалізації цього завдання застосовують різноманітні математичні підходи. Локальні методи, найвідомішим з яких є алгоритм Лукаса-Канаде, обробляють обмежені ділянки зображення, виходячи з припущення, що сусідні пікселі рухаються синхронно. Натомість глобальні методи, такі як алгоритм Хорна-Шунка, намагаються мінімізувати функцію енергії для всього кадру, що дає більш цілісну картину, але потребує значних обчислювальних потужностей. Останнім часом все частіше застосовуються згорткові нейронні мережі, які демонструють високу стійкість до візуальних шумів та складних умов освітлення [7, 8].

Інформація, отримана шляхом аналізу оптичного потоку, дозволяє точно визначати вектори руху об'єктів або самої камери відносно статичного оточення. При комплексуванні цих даних з інформацією від далекомірів або стереокамер стає можливою побудова детальних тривимірних моделей простору та реконструкція траєкторії руху в динамічному середовищі, як це візуалізовано на рисунку, рис. 1.4.

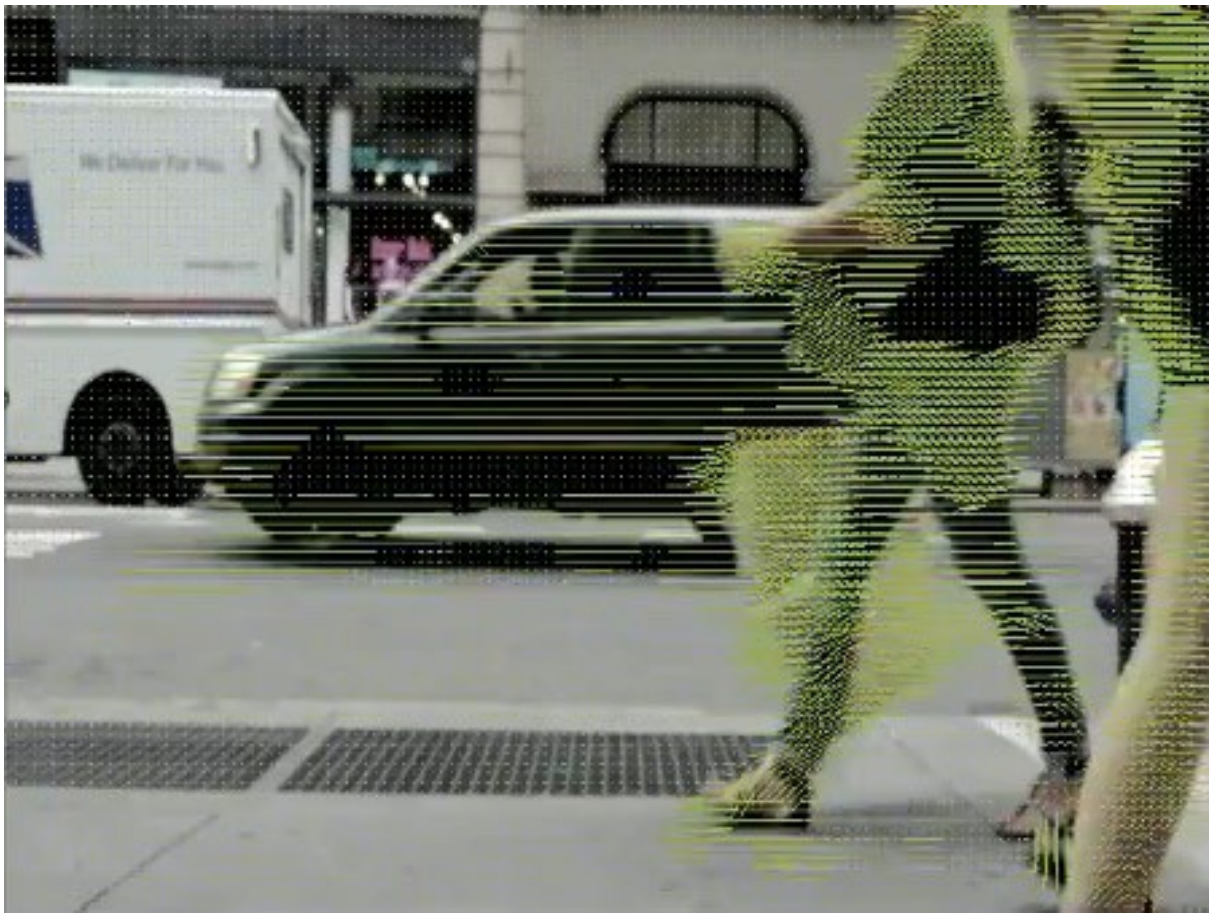


Рисунок 1.4 – Графічна візуалізація векторного поля оптичного потоку

Варто зазначити, що технологія оптичного потоку не позбавлена певних технічних обмежень. Значні труднощі для алгоритмів становлять однорідні поверхні без вираженої текстури, такі як гладкі стіни, чисте небо або водна поверхня, оскільки на них неможливо виділити контрольні точки для відстеження зміщення. Також джерелом похибок можуть стати різкі зміни освітлення, поява тіней або візуальний шум матриці камери. Крім того, при занадто швидкому русі дрона зміщення між кадрами може бути надто великим

для коректної обробки стандартними алгоритмами, що вимагає застосування багаторівневих пірамідальних методів обробки зображень [9].

Попри наявні виклики, інтеграція оптичного потоку в систему управління є необхідним кроком для створення повноцінних автономних навігаційних комплексів, особливо для роботи в умовах "GPS-denied". У сучасних безпілотних системах ця технологія виконує роль візуального датчика швидкості, допомагає ухилятися від зіткнень та забезпечує високоточне маневрування. Завдяки поєднанню з даними інерціальних датчиків та стереозору, оптичний потік значно розширює функціональні можливості робототехнічних систем [10].

1.2 Алгоритмічні методи комплексування різнорідної сенсорної інформації: фільтрація Калмана та інерціальне числення

Фундаментальною проблемою навігації автономних систем є неможливість гарантувати абсолютну точність позиціонування при використанні лише одного джерела даних. Кожен вимірювальний прилад, будь то супутниковий приймач чи акселерометр, має свої фізичні обмеження, піддається впливу специфічних шумів та зовнішніх перешкод. Саме тому критично важливим підходом у проектуванні навігаційних комплексів БПЛА є сенсорна фузія — процес об'єднання потоків даних від різнорідних датчиків. У даному підрозділі проаналізовано ключові стратегії інтеграції інформації, які дозволяють нівелювати слабкі сторони окремих компонентів та забезпечити надійну роботу системи в цілому [11].

Історично одним із найбільш значущих досягнень у цій сфері стала розробка рекурсивного фільтра, який отримав назву на честь свого творця — Рудольфа Е. Калмана. Теоретичні основи цього методу були закладені на рубежі 1950–60-х років. Ключовою подією стала публікація Рудольфом

Калманом у 1960 році наукової праці, присвяченої новому підходу до задач лінійної фільтрації та прогнозування. Вагомий внесок у популяризацію та практичну адаптацію алгоритму зробив інженер Стенлі Шмідт, завдяки якому теоретичні викладки знайшли своє застосування у реальних інженерних задачах. Ця співпраця стала каталізатором для подальшого розвитку теорії керування [13, 14].

Першими сферами, де фільтр Калмана довів свою незамінність, стали аерокосмічна індустрія та військова авіація. Вже у 1960-х роках фахівці NASA імплементували цей алгоритм у бортові комп'ютери космічних апаратів. Його застосування стало критичним фактором успіху місячної програми Apollo, де вдосконалені версії фільтра використовувалися для розрахунку траєкторій польоту та забезпечення високоточної посадки на поверхню Місяця. Це дозволило досягти безпрецедентного на той час рівня надійності систем керування [13, 14, 15, 16].

Сутність фільтра Калмана полягає у тому, що це оптимальний рекурсивний алгоритм, здатний оцінювати внутрішній стан динамічної системи, спираючись на серію зашумлених вимірювань. Завдяки своїй універсальності, він став стандартом де-факто у робототехніці та навігації. Головна сила методу полягає у здатності інтелектуально "зважувати" дані з різних джерел, спираючись на статистичні характеристики їхньої точності, та ефективно відфільтровувати випадкові флуктуації сигналу. Алгоритм працює циклічно, використовуючи математичні моделі руху об'єкта та моделі вимірювань. Він постійно уточнює оцінку поточного стану, комбінуючи попередній прогноз із новими даними, що надходять у реальному часі.

Спектр застосування цього методу надзвичайно широкий. Він використовується для згладжування "сирих" даних від сенсорів, що дозволяє прибрати високочастотний шум. Критично важливою є його роль у корекції помилок інерціальних систем: фільтр дозволяє компенсувати дрейф гіроскопів та акселерометрів, що накопичується з часом. Найбільш поширеним сценарієм є комплексування даних, коли, наприклад, високочастотні, але схильні до

дрейфу дані IMU, коригуються точними, але повільними даними GPS або систем комп'ютерного зору.

В контексті безпілотних літальних апаратів фільтр Калмана виступає ядром навігаційної системи. Він забезпечує оптимальну оцінку координат та кутів орієнтації навіть за умов значних вібрацій або тимчасової втрати сигналу GPS. Це дозволяє використовувати дрони у логістиці, геодезії та розвідці, де вимоги до стабільності польоту є надзвичайно високими. Інженерна реалізація таких систем дозволяє дрону утримувати задану позицію при сильному вітрі або виконувати складні автономні місії у зонах з високим рівнем електромагнітних перешкод. В авіації аналогічні підходи дозволяють пілотам та автопілотам отримувати коректні дані про висоту та швидкість, об'єднуючи інформацію з барометрів, радарів та інерціальних блоків.

Окрім авіації, алгоритм є фундаментом для наземної робототехніки та автомобільної промисловості. Автономні роботи-кур'єри використовують його для навігації у складному міському середовищі, інтегруючи дані лідарів та одометрії. У сучасних автомобілях системи курсової стійкості (ESP) та адаптивний круїз-контроль базуються на тих самих математичних принципах для забезпечення безпеки пасажирів [18]. Програмна реалізація цих алгоритмів зазвичай виконується мовами високого рівня, такими як C++, Python або MATLAB, що дозволяє обробляти великі масиви даних з мінімальною затримкою.

У більш екзотичних застосуваннях, таких як підводна робототехніка або супутникове зондування Землі, фільтр Калмана допомагає створювати точні карти морського дна або аналізувати кліматичні зміни, очищуючи дані від атмосферних спотворень та похибок апаратури. Принцип роботи фільтрації у контексті супутникової навігації візуалізовано на схемі (рис. 1.5).

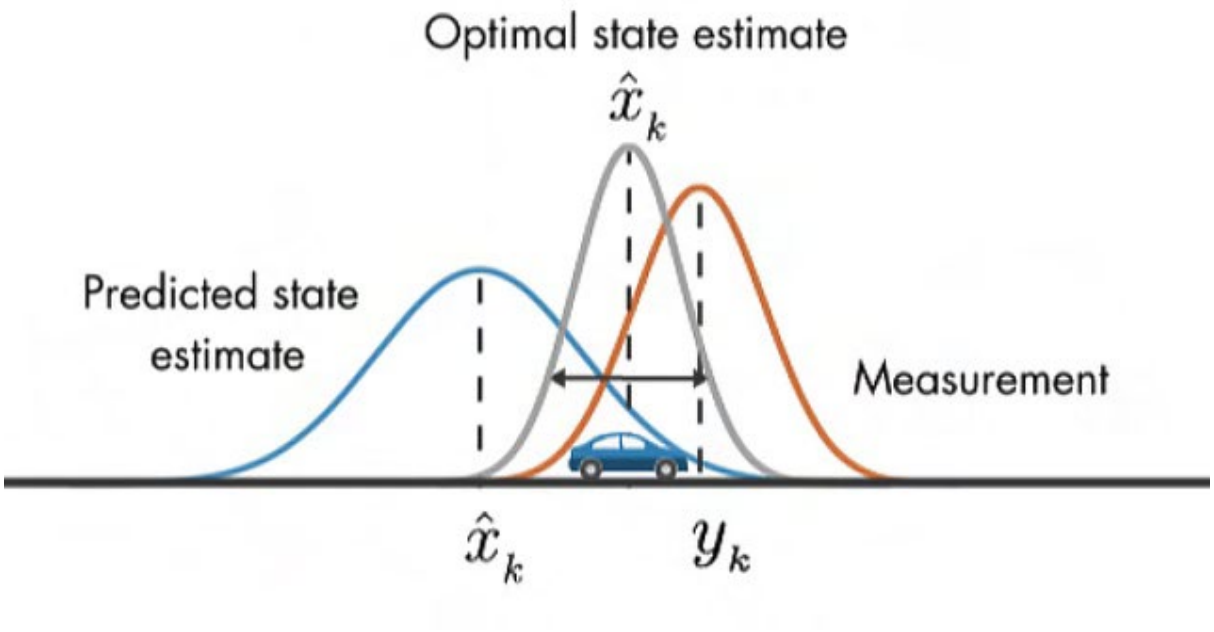


Рисунок 1.5 – Застосування алгоритму фільтрації у супутникових навігаційних системах

Робота алгоритму базується на циклічному процесі, який можна розділити на два етапи: прогнозування (Predict) та оновлення (Update). На етапі прогнозування система розраховує апріорну оцінку стану об'єкта на наступний момент часу, виходячи з його поточної швидкості та діючих керуючих сил. Паралельно прогнозується матриця коваріації помилок, яка показує ступінь невизначеності цього прогнозу. На етапі оновлення відбувається розрахунок так званого коефіцієнта підсилення Калмана. Цей коефіцієнт визначає, наскільки сильно нові вимірювання від датчиків повинні вплинути на кінцевий результат. Далі відбувається коригування прогнозованого стану з урахуванням реальних вимірювань та оновлення матриці коваріації для наступного циклу.

Математичний апарат фільтра описує динаміку системи через набір матричних рівнянь. Рівняння стану пов'язує майбутнє положення системи з поточним через матрицю переходу станів та вектор керування, враховуючи при цьому випадковий шум процесу. У свою чергу, рівняння вимірювань описує зв'язок між вектором реальних показників сенсорів та вектором стану системи, враховуючи шуми вимірювань.

Алгоритмічно процес можна описати наступною послідовністю дій. Спочатку виконується екстраполяція вектора стану на наступний крок, використовуючи фізичну модель руху. Далі розраховується апріорна коваріаційна матриця помилок. Після отримання нових даних від сенсорів обчислюється оптимальний коефіцієнт підсилення, який мінімізує середньоквадратичну помилку оцінки. За допомогою цього коефіцієнта знаходиться апостеріорна (уточнена) оцінка стану системи як зважена сума прогнозу та різниці між реальним вимірюванням і очікуваним. На фінальному етапі коригується матриця коваріації помилок, яка буде використана у наступній ітерації [20, 21]. Загальну логіку функціонування алгоритму наведено на ілюстрації (рис. 1.6).

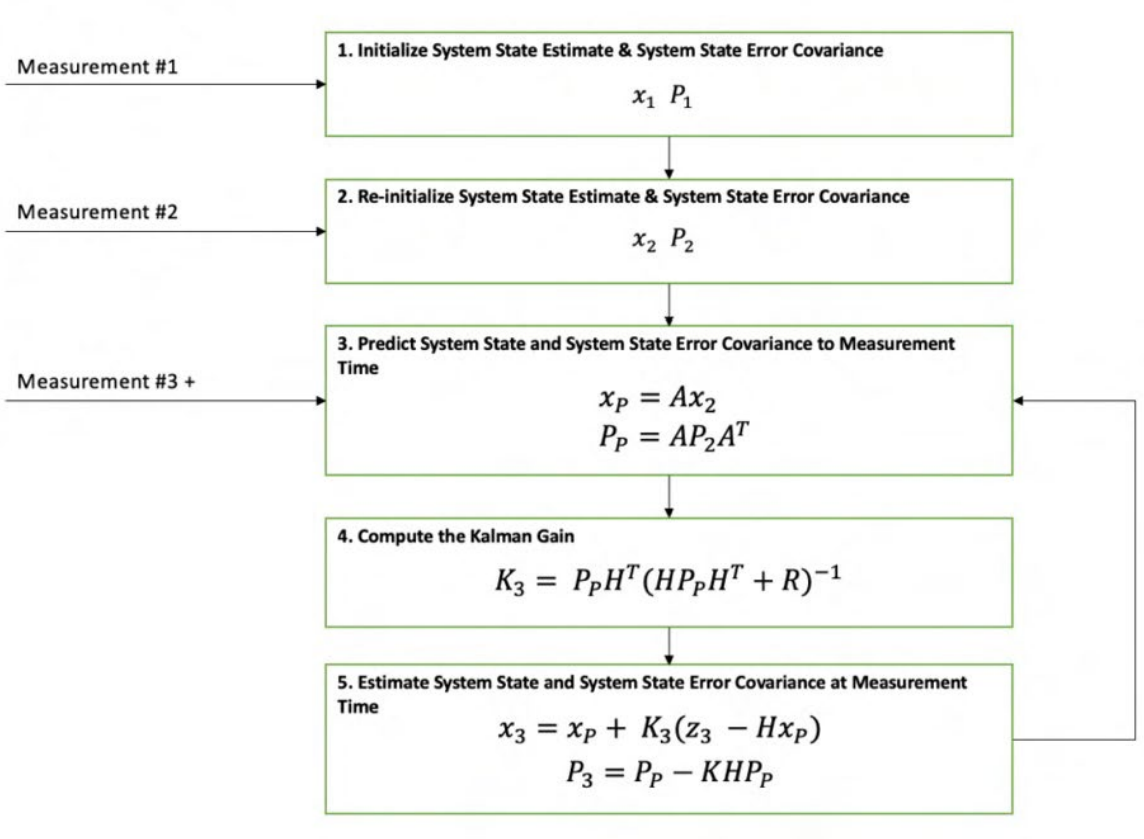


Рисунок 1.6 – Структурна схема циклу роботи рекурсивного фільтра

Важливо зазначити, що класичний лінійний фільтр Калмана не завжди підходить для реальних фізичних систем, рух яких часто є нелінійним. Для

вирішення таких задач застосовується Розширений фільтр Калмана (Extended Kalman Filter — EKF). Його особливість полягає у лінеаризації рівнянь руху та вимірювань в околі поточної оцінки за допомогою обчислення матриць Якобі (перших похідних рядів Тейлора) [23, 24].

Підсумовуючи, можна виділити ключові переваги та обмеження методу. До сильних сторін відносяться: ефективне зменшення шумів, можливість злиття даних різної фізичної природи (наприклад, оптичних та інерціальних) та висока обчислювальна ефективність завдяки рекурсивній природі, що не вимагає зберігання всієї історії вимірювань. Серед недоліків слід відзначити високу чутливість до точності математичної моделі: якщо динаміка об'єкта описана некоректно, фільтр може давати великі похибки або розходитися. Крім того, реалізація EKF для складних нелінійних систем вимагає значних математичних розрахунків та обчислювальних ресурсів процесора [23, 24, 25, 27, 28, 29].

Окремим фундаментальним методом визначення просторового положення безпілотних систем є навігаційне числення координат, відоме у міжнародній практиці як Dead Reckoning. Суть цього підходу полягає у математичному моделюванні поточної позиції об'єкта шляхом інтегрування вектора швидкості та напрямку руху відносно відомої початкової точки. На відміну від позиційних систем, які визначають координати «тут і зараз», цей метод розраховує, де повинен знаходитися дрон, базуючись на тому, як довго і в якому напрямку він рухався. Ця технологія є базовою для інерціальних навігаційних систем і набуває критичного значення в умовах, коли зовнішні навігаційні сигнали недоступні або спотворені [30].

Застосування методу числення координат має низку суттєвих переваг, серед яких ключовою є повна автономність. Система не потребує зв'язку із супутниками, наземними станціями чи наявності попередньо завантажених карт, що робить її незамінною в умовах радіоелектронної боротьби або в замкнутих просторах. Крім того, завдяки використанню виключно внутрішніх пропріоцептивних сенсорів (зокрема акселерометрів та гіроскопів у складі

IMU), обробка даних відбувається з надзвичайно високою частотою. Це забезпечує миттєву реакцію системи управління на зміни траєкторії, що є критичним для стабілізації польоту високоманеврених апаратів.

Однак, методологія Dead Reckoning має й критичні недоліки, пов'язані з фізикою процесу інтегрування. Будь-яка, навіть найменша похибка вимірювання сенсора (шум акселерометра або дрейф гіроскопа), при інтегруванні накопичується і зростає з часом у геометричній прогресії. Це явище призводить до того, що без зовнішньої корекції розраховані координати швидко відхиляються від реальних. Для забезпечення тривалої точної роботи необхідна періодична верифікація положення за допомогою абсолютних систем позиціонування, таких як GPS-приймачі або системи візуальної одометрії та SLAM, які «скидають» накопичену помилку [31].

Для наочного зіставлення розглянутих у розділі підходів до обробки навігаційних даних доцільно звернутися до порівняльної характеристики, наведеної у таблиці, табл. 1.1.

Таблиця 1.1 – Порівняльна характеристика алгоритмів сенсорної фузії та навігації.

Алгоритмічний підхід	Ключові переваги	Технічні обмеження
Фільтрація Калмана	Забезпечує ефективне згладжування сенсорних шумів та дозволяє коректно об'єднувати (фузіонувати) інформацію з різнорідних джерел (GPS, IMU, одометрія).	Вимагає наявності точної математичної моделі динаміки системи; помилки в моделюванні можуть призвести до розходження фільтра.
Навігаційне числення (Dead Reckoning)	Характеризується повною автономністю від зовнішніх сигналів та високою	Страждає від ефекту накопичення похибки з часом (дрейф), що вимагає

	швидкістю оновлення даних, що критично для контурів управління.	періодичної корекції абсолютними методами навігації.

1.3 Інтеграція методів комп'ютерного зору та алгоритмів SLAM для просторової орієнтації

В умовах, коли супутникова навігація стає недоступною або ненадійною, критичного значення набувають системи технічного зору, які дозволяють безпілотним літальним апаратам сприймати навколишній простір подібно до живих істот. На сьогоднішній день провідною концепцією у цій сфері є технологія одночасної локалізації та побудови карти, відома у міжнародній практиці як SLAM (Simultaneous Localization and Mapping). Ця методологія вирішує фундаментальний парадокс мобільної робототехніки: для того, щоб точно визначити своє положення, дрону необхідна карта місцевості, проте для створення такої карти необхідно точно знати траєкторію руху апарата [32].

Сутність алгоритмів SLAM полягає у виконанні двох нерозривно пов'язаних процесів у режимі реального часу. По-перше, система здійснює локалізацію, тобто обчислює координати та кути орієнтації робота відносно точок прив'язки у просторі. По-друге, відбувається динамічне картографування — створення цифрової моделі оточення, яка постійно оновлюється та доповнюється новими даними. Для забезпечення коректності цих процесів алгоритми повинні враховувати стохастичну природу вимірювань, компенсуючи шуми сенсорів та похибки обчислень [32, 33, 34]. Загальну логіку функціонування цього процесу візуалізовано на схемі, рис. 1.7.

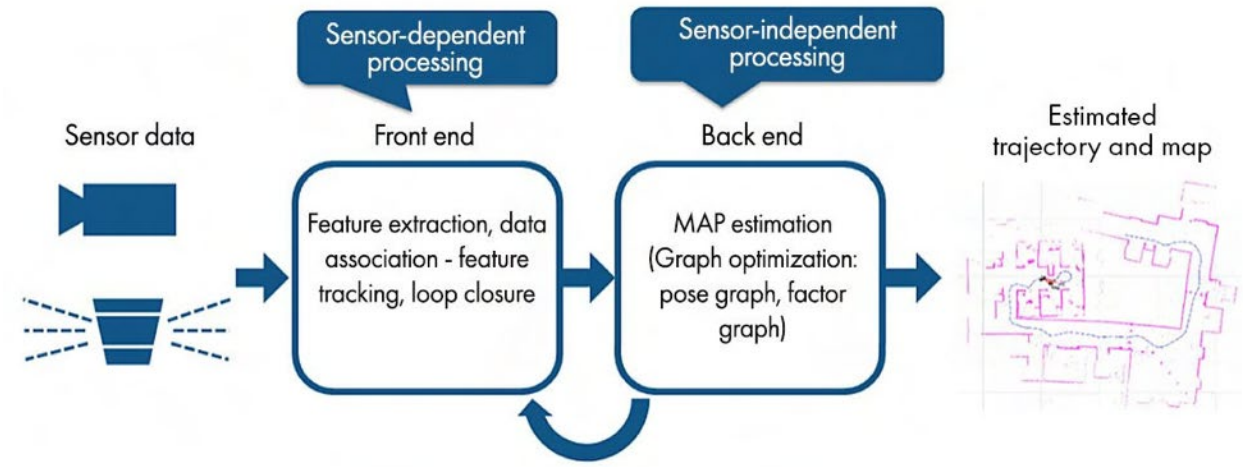


Рисунок 1.7 – Структурна схема функціонування методу SLAM

Класифікація методів SLAM здійснюється залежно від типу сенсорів, що використовуються для отримання первинної інформації. Найбільш поширеним є візуальний SLAM, який базується на аналізі відеопотоку. У рамках цього підходу виділяють монокулярні системи, що використовують одну камеру, проте вони мають суттєвий недолік у вигляді невизначеного масштабу, що ускладнює оцінку реальних дистанцій. Стереоскопічні системи, що імітують бінокулярний зір людини, дозволяють отримувати карту глибини безпосередньо з пари зображень. Окремим класом є RGB-D системи, які додатково використовують інфрачервоні проектори або ToF-сенсори (наприклад, Microsoft Kinect) для точного вимірювання відстані до кожного пікселя сцени [35].

Альтернативою оптичним методам є лазерний SLAM, який базується на використанні лідарів (LiDAR). Ця технологія забезпечує надзвичайно високу точність геометричних вимірювань та є стандартом для створення детальних тривимірних карт, хоча й вирізняється високою вартістю та значним енергоспоживанням. Також існує інерціальний підхід, який використовує дані акселерометрів та гіроскопів для прогнозування переміщення, однак зазвичай він застосовується як допоміжний елемент для компенсації різких маневрів у візуальних системах [36, 37].

Найбільш перспективним напрямком розвитку є мультисенсорний SLAM. Ця концепція передбачає комплексування даних від камер, лідарів, інерціальних блоків, ультразвукових сонарів та GNSS-приймачів. Такий підхід дозволяє нівелювати слабкі сторони окремих датчиків за рахунок сильних сторін інших. Наприклад, в умовах недостатнього освітлення або візуальної однорідності середовища (туман, дим), де камери стають неефективними, система автоматично переходить на використання даних лідара або IMU. Це забезпечує системі властивості відмовостійкості та підвищеної точності за рахунок перехресної перевірки вимірювань. Архітектура такої системи наведена нижче, рис. 1.8.

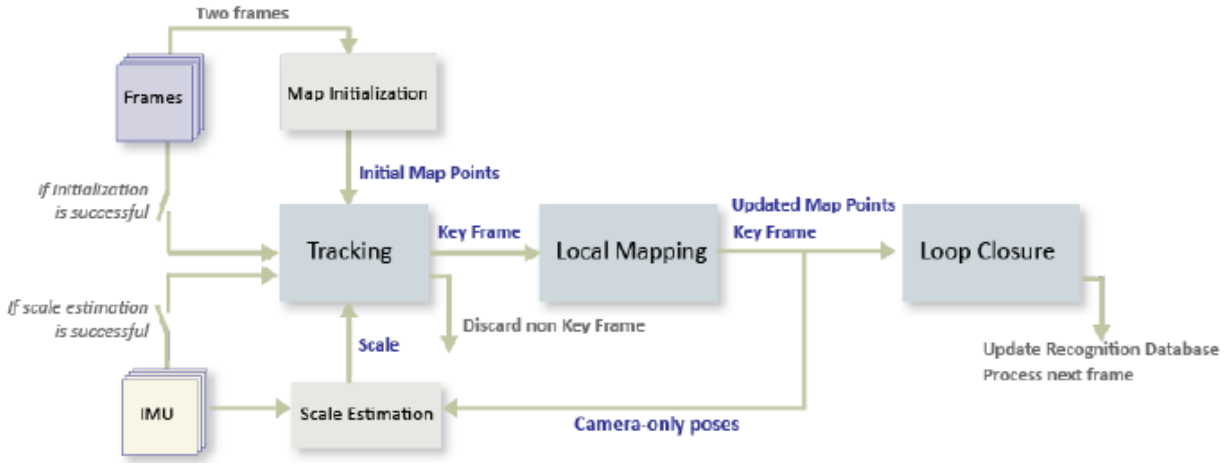


Рисунок 1.8 – Архітектура системи мультисенсорної навігації

Однак інженерна реалізація мультисенсорних систем пов'язана зі значними технічними викликами. Першочерговою проблемою є часова синхронізація потоків даних, оскільки різні сенсори мають різну частоту оновлення та затримки передачі сигналу. Не менш важливим є просторове калібрування, що дозволяє точно визначити взаємне розташування та орієнтацію сенсорів на корпусі дрона. Крім того, обробка масивів різнорідних даних вимагає значних обчислювальних потужностей, що накладає обмеження на апаратну частину бортового комп'ютера. Для вирішення задачі інтеграції

даних розробляються складні алгоритми "сенсорної фузії", де ключову роль часто відіграють розширені фільтри Калмана.

Для побудови ефективної навігаційної системи безпілотника необхідний ретельний підбір сенсорів. Камери забезпечують розпізнавання образів та текстур, лідари гарантують метричну точність карти, IMU відповідає за відстеження високочастотної динаміки, а ультразвукові датчики незамінні для виявлення прозорих перешкод або визначення точної висоти на малих дистанціях [39]. Порівняльний аналіз характеристик цих приладів наведено у таблиці, табл. 1.2.

Таблиця 1.2 – Аналіз ефективності сенсорів у складі навігаційних комплексів

Тип сенсора	Ключові переваги	Технічні обмеження та недоліки
Оптична камера	Надає багату інформацію про колір та текстуру, має високу роздільну здатність.	Критично залежить від умов освітлення, потребує складних обчислень для отримання глибини сцени.
LiDAR	Забезпечує пряме та високоточне вимірювання відстаней, працює у повній темряві.	Має високу вартість, значні габарити та високе енергоспоживання.
IMU (Інерціальний блок)	Працює з високою частотою, повністю автономний, стійкий до зовнішніх перешкод.	Схильний до швидкого накопичення інтегральної похибки (дрейфу).
Ультразвуковий сонар	Низька вартість та простота інтеграції.	Обмежена дальність дії, низька роздільна здатність, вразливість до акустичних перешкод.

Серед програмних рішень, що реалізують згадані принципи, можна виділити кілька алгоритмів, які стали стандартом в індустрії. ORB-SLAM2 відомий своєю універсальністю та здатністю працювати з різними конфігураціями камер, забезпечуючи виявлення замикання циклів. Google Cartographer, розроблений технологічним гігантом, спеціалізується на створенні 2D та 3D карт у реальному часі, ефективно використовуючи дані лідарів, табл.1.3. Алгоритм ROVIO, у свою чергу, фокусується на тісній інтеграції візуальної та інерціальної одометрії, що робить його ідеальним для високоманеврених мікро-дронів [40, 41, 42].

Таблиця 1.3 – Порівняльна характеристика сучасних SLAM-алгоритмів

Назва алгоритму	Основні джерела даних	Специфіка застосування
ORB-SLAM2	Монокулярні, стерео та RGB-D камери	Відкритий вихідний код, висока точність локалізації за ознаками.
Google Cartographer	LiDAR, IMU	Оптимізований для побудови карт великих площ, підтримує 2D/3D режими.
ROVIO	Камера, IMU	Висока швидкодія, низькі вимоги до ресурсів, базується на фільтрі Калмана.

Завершуючи огляд, варто підкреслити роль фільтра Калмана у цих системах. Він виступає математичним ядром, що об'єднує розрізнені показники в єдину узгоджену модель стану. Фільтр дозволяє прогнозувати положення дрона між кадрами відеопотоку, коригувати карту при повторному відвідуванні локацій та згладжувати траєкторію руху, відсіюючи випадкові

викиди вимірювань. Саме завдяки такій математичній обробці досягається синергія сенсорів у сучасних автономних системах [43].

1.4 Роль інтелектуальних систем у забезпеченні автономності безпілотних апаратів

Впровадження методів штучного інтелекту (ШІ) стало каталізатором революційних змін у сфері безпілотних авіаційних систем. Саме завдяки інтелектуальним алгоритмам дрон перетворюється з дистанційно керованого механізму на повноцінного автономного робота, здатного аналізувати ситуацію та приймати рішення за долі секунди. Це дозволяє вирішувати комплексні завдання: від високоточного позиціонування та побудови маршрутів до адаптації під непрогнозовані зміни зовнішнього середовища та самонавчання в процесі експлуатації [43].

У даному дослідженні пропонується архітектура навігаційної системи, що базується на глибокій інтеграції мультисенсорного SLAM, глобальної системи позиціонування (GPS) та нейромережевих моделей. Ключовим елементом, що об'єднує ці різнорідні потоки даних, виступає фільтр Калмана. Така гібридна модель дозволяє нівелювати слабкі сторони окремих технологій, забезпечуючи надійну навігацію навіть в зонах радіоелектронних перешкод або щільної міської забудови, де сигнал GPS може бути втрачений [45].

Фундаментальним викликом для будь-якої автономної системи є забезпечення безпеки руху, що передбачає своєчасне виявлення та обхід перешкод. Цей процес вимагає від бортового комп'ютера обробки масивів даних від камер, лідарів та інерціальних датчиків у реальному часі. Штучний інтелект виступає тут у ролі "мозку", що інтерпретує "сирі" дані сенсорів та формує керуючі команди [45].

Функціонал системи уникнення перешкод реалізується через кілька етапів. Першочергово відбувається семантичний аналіз сцени: алгоритми комп'ютерного зору класифікують об'єкти на шляху дрона, розрізняючи статичні структури (будівлі, дерева) та інші літальні апарати, формуючи динамічну карту загроз. Наступним кроком є прогнозування поведінки динамічних об'єктів. ШІ екстраполює траєкторії руху людей або транспортних засобів, що дозволяє превентивно розрахувати точку можливого зіткнення та уникнути її. Фінальним етапом є генерація альтернативного маршруту. При виявленні перешкоди система автоматично перебудовує політне завдання, оптимізуючи нову траєкторію за критеріями енергоефективності та мінімізації часу затримки [46].

Для підвищення надійності навігації використовується згадана вище комбінація мультисенсорного SLAM та нейронних мереж. Хоча SLAM дозволяє будувати карту та визначати положення відносно неї, він схильний до накопичення дрейфу. GPS надає абсолютні координати, але має низьку частоту оновлення та вразливість до перешкод. Нейронні мережі вирішують цю проблему, навчаючись розпізнавати візуальні орієнтири, що дозволяє дрону "прив'язуватися" до місцевості подібно до того, як це робить людина, навіть за повної відсутності зовнішнього зв'язку [46].

Нейромережевий компонент виконує три критичні функції. По-перше, він прогнозує координати на основі кінематичної моделі руху, згладжуючи дані IMU. По-друге, використання згорткових нейронних мереж (CNN) дозволяє детектувати характерні ознаки ландшафту (наприклад, специфічні будівлі), що слугують опорними точками для корекції позиції. По-третє, мережа може навчатися на історії помилок SLAM-алгоритму, виявляючи систематичні відхилення та програмно компенсуючи їх [47, 48].

Математичне об'єднання цих даних здійснюється через фільтр Калмана за наступним алгоритмом: Спочатку формується прогноз стану на основі інерціальних даних та передбачень нейромережі, що дозволяє миттєво реагувати на маневри. Далі відбувається етап оновлення (Update step), де

використовуються відносні координати від SLAM та візуальні орієнтири, розпізнані нейромережею. Якщо доступний сигнал GPS, він використовується як третє джерело для глобальної корекції та обнулення накопиченої помилки. Цікавою особливістю такої архітектури є можливість динамічного керування ресурсами: при стабільному сигналі GPS енергоємні модулі ШІ можуть бути тимчасово відключені для заощадження заряду батареї [49].

Окремим перспективним напрямком є використання методології навчання з підкріпленням (Reinforcement Learning - RL). Цей підхід імітує процес природного навчання, де агент (дрон) взаємодіє із середовищем і отримує зворотний зв'язок у вигляді "нагород" за правильні дії або "штрафів" за помилки.

Це дозволяє реалізувати справжню адаптивність: дрон не просто виконує жорстко прописаний код, а оптимізує свою стратегію поведінки на основі накопиченого досвіду. Завдяки RL, апарат вчиться краще маневрувати у складних умовах, мінімізувати витрати енергії при вітровому навантаженні та знаходити неочевидні шляхи обходу перешкод. Головна перевага методу — здатність системи зберігати стабільність та ефективність навіть у абсолютно нових умовах, з якими вона не стикалася на етапі розробки [50, 51]. Схематично цей ітеративний процес взаємодії агента з середовищем зображено на рисунку, рис. 1.9.

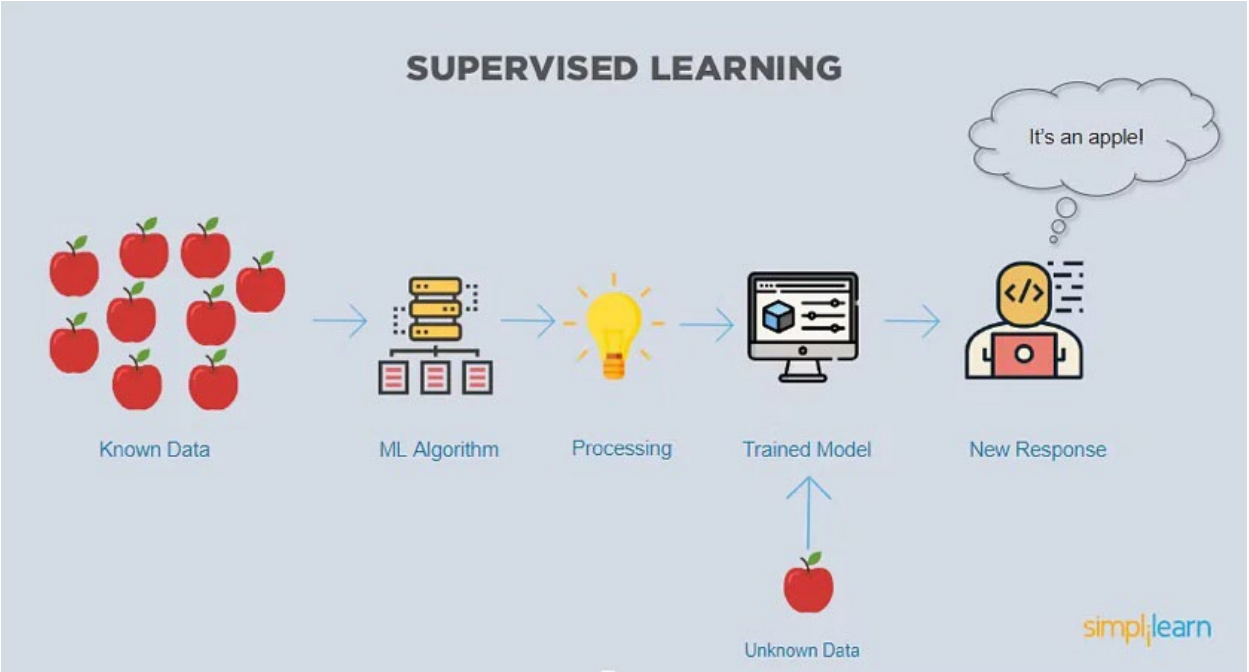


Рисунок 1.9 – Концептуальна схема циклу навчання з підкріпленням

Здатність розпізнавати об'єкти є фундаментом інтелектуальної навігації. Використання згорткових нейронних мереж (CNN) дозволяє дрону не просто бачити перешкоди як набір точок, а розуміти їх природу. Система класифікує об'єкти на різні категорії (транспорт, рослинність, інфраструктура), що дозволяє пріоритизувати загрози. Наприклад, дрон може ігнорувати траву, але уникати гілок дерев. Крім того, ШІ здатен відрізняти статичні об'єкти від динамічних, прогнозуючи вектор руху останніх, що критично важливо для уникнення зіткнень у спільному повітряному просторі [53, 54].

Для практичної реалізації використовуються різні архітектури CNN, вибір яких залежить від балансу між швидкістю роботи та точністю детекції [55]. Порівняльний аналіз найбільш поширених архітектур наведено у таблиці, табл. 1.4.

Таблиця 1.4 – Порівняльна характеристика нейромережевих архітектур для детекції об'єктів

Архітектура мережі	Ключові переваги	Технічні обмеження
YOLO (You Only Look Once)	Забезпечує надвисоку швидкість обробки, ідеально підходить для роботи в режимі реального часу (Real-time).	Може демонструвати знижену точність при детекції дрібних об'єктів на великій відстані.
SSD (Single Shot Detector)	Пропонує оптимальний компроміс між швидкістю інференсу та точністю розпізнавання.	Реалізація та налаштування можуть бути складнішими порівняно з простішими моделями.
Faster R-CNN	Характеризується найвищою точністю детекції та якістю локалізації об'єктів.	Вимагає значних обчислювальних ресурсів, має меншу швидкість роботи (FPS).

Реалізація алгоритмів штучного інтелекту на борту легких безпілотників стикається з жорсткими обмеженнями по вазі та енергоспоживанню. Тому замість потужних графічних станцій використовуються спеціалізовані мікроконтролери з підтримкою периферійних обчислень (Edge AI).

Одним із лідерів у цьому сегменті є мікроконтролери серії STM32. Компанія STMicroelectronics активно розвиває екосистему для запуску нейромереж безпосередньо на контролері. Головною перевагою є екстремально низьке енергоспоживання та малі габарити, що критично для мікро-дронів. Однак, обчислювальна потужність обмежує використання лише найпростішими моделями класифікації, які потребують значної оптимізації [56].

Альтернативою є платформа Arduino Nano 33 BLE Sense. Це рішення "все-в-одному", що вже містить на борту набір сенсорів (гіроскоп, акселерометр, мікрофон) та підтримує фреймворк TensorFlow Lite for Microcontrollers. Це дозволяє швидко розгортати моделі машинного навчання, хоча обмеження по швидкодії аналогічні іншим мікроконтролерам [57].

Також популярністю користується чіп Espressif ESP32. Його головна фішка — вбудовані модулі Wi-Fi та Bluetooth при низькій вартості. Це дозволяє створювати розподілені системи, де частина обчислень виконується на борту, а частина — на сервері. ESP32 також підтримує бібліотеки для машинного навчання, такі як MicroML, але вимагає ретельної оптимізації коду для роботи в реальному часі [58].

Зведені характеристики розглянутих платформ представлені в таблиці (табл. 1.5).

Таблиця 1.5 – Характеристика апаратних платформ для вбудованого ШІ

Апаратна платформа	Функціональні особливості	Рівень енергоефективності
STM32 (Edge AI)	Нативна підтримка ML-інструментів, оптимізація під вбудовані системи.	Високий (наднизьке споживання).
Arduino Nano 33 BLE Sense	Інтегрований набір сенсорів, сумісність з TensorFlow Lite.	Високий.
Espressif ESP32	Вбудовані бездротові інтерфейси, низька собівартість.	Високий (залежить від режиму Wi-Fi).

Використання таких платформ, попри обмеження у складності моделей (наприклад, неможливість запуску повноцінного візуального SLAM), дозволяє

реалізувати базові інтелектуальні функції: стабілізацію польоту на основі нейромереж, розпізнавання простих візуальних маркерів та реактивне уникнення перешкод, зберігаючи при цьому легкість та автономність дрона.

2 ТЕОРЕТИКО-МЕТОДОЛОГІЧНІ АСПЕКТИ ПОБУДОВИ СИСТЕМ АВТОНОМНОГО КЕРУВАННЯ БЕЗПІЛОТНИМИ ЛІТАЛЬНИМИ АПАРАТАМИ

2.1 Формалізація та визначення ключових завдань навігації в контексті автономних безпілотних систем

Стрімка еволюція безпілотних технологій та розширення спектру їх застосування — від цивільної логістики до виконання спеціальних місій — висувають нові, більш жорсткі вимоги до навігаційних комплексів. На сучасному етапі розвитку техніки автономна навігація розглядається не просто як допоміжна функція, а як критичний компонент, що визначає саму можливість виконання польотного завдання. Генеральною метою проектування таких систем є забезпечення здатності літального апарата самостійно, без участі оператора, переміщуватися від точки старту до пункту призначення з високою надійністю. Ключовим аспектом при цьому стає мінімізація впливу людського фактору, який історично є причиною значної частки аварій, а також забезпечення стійкості до втрати сигналів глобальних навігаційних супутникових систем (GNSS), що є поширеною проблемою в умовах урбаністичної забудови або дії засобів радіоелектронної боротьби.

Для реалізації повноцінної автономності навігаційний комплекс повинен задовольняти низку суворих технічних та експлуатаційних вимог. Першочерговою вимогою є висока точність позиціонування. Це критично важливо для таких сценаріїв використання, як доставка вантажів у конкретну точку ("остання миля"), геодезична зйомка, моніторинг інфраструктурних об'єктів або інспекція ліній електропередач, де відхилення навіть на кілька десятків сантиметрів може призвести до невиконання місії або аварії.

Не менш важливим аспектом є гарантування безпеки польоту. Система повинна бути оснащена алгоритмами активного сприйняття середовища, здатними в реальному часі детектувати статичні та динамічні перешкоди та автоматично коригувати траєкторію для уникнення зіткнень. Це вимагає інтеграції складних сенсорних систем та швидкодіючих обчислювальних модулів.

Ще однією характеристикою якісної системи є її адаптивність. Дрон повинен коректно реагувати на непередбачувані зміни зовнішніх умов, такі як раптові пориви вітру, зміна освітлення, поява нових об'єктів у зоні польоту або зміна пріоритетів місії "на льоту" [59].

Надійність системи визначається її здатністю продовжувати виконання завдання при деградації вхідних даних, зокрема при повному зникненні сигналу GPS/GNSS. У таких випадках система повинна автоматично перемикатися на альтернативні методи навігації (інерціальні, візуальні тощо), забезпечуючи безперервність керування. Окрім того, враховуючи обмеженість ресурсу бортових акумуляторів, важливою вимогою є енергоефективність, яка досягається шляхом математичної оптимізації маршруту, вибору економних режимів польоту та мінімізації зайвих маневрів.

Систематизований перелік ключових вимог до навігаційного забезпечення автономних БПЛА представлено у таблиці, табл. 2.1.

Таблиця 2.1 – Функціональні вимоги до підсистем автономної навігації

Параметр системи	Характеристика та функціональне призначення
Точність геопросторової прив'язки	Забезпечення мінімального відхилення від заданих координат, що є критичним для точного виконання цільових завдань (посадка, скидання вантажу, зйомка).
Експлуатаційна безпека	Здатність системи автономно ідентифікувати загрози зіткнення та реалізовувати маневри

	ухилення для збереження цілісності апарата та оточення.
Ситуативна адаптивність	Гнучке реагування алгоритмів керування на динамічні зміни метеоумов, освітлення або появу незапланованих перешкод.
Відмовостійкість та надійність	Гарантування стабільного функціонування та утримання курсу в умовах радіоперешкод, відсутності супутникового сигналу або часткової відмови сенсорів.
Енергетична ефективність	Алгоритмічна оптимізація траєкторії руху для мінімізації витрат енергії, що дозволяє збільшити радіус дії та тривалість польоту.

2.2 Аналіз апаратно-програмних засобів та алгоритмічних стратегій реалізації автономного польоту

Створення ефективної системи автоматичної навігації для безпілотних літальних апаратів (БПЛА) вимагає комплексного підходу, що базується на синергії апаратних сенсорів та складних програмних алгоритмів. Головною метою такої інтеграції є досягнення повної автономності, яка гарантує високоточне позиціонування у просторі та безпеку виконання місій без участі оператора. Для реалізації цих завдань необхідно об'єднати методи збору телеметрії, інтелектуальної обробки даних та адаптивного керування виконавчими механізмами, що дозволить дрону зберігати стабільність навіть у складних та непередбачуваних умовах експлуатації.

Архітектура навігаційного комплексу сучасного дрона базується на кількох фундаментальних підсистемах: глобальних супутникових системах навігації (GNSS/GPS), інерціальних вимірювальних блоках (IMU),

магнітометричних датчиках (компасах) та системах технічного зору. GPS виступає основним джерелом абсолютних координат, забезпечуючи прив'язку апарата до глобальної карти. Однак, надійність супутникового сигналу може суттєво знижуватися в умовах щільної міської забудови, де виникають ефекти перевідбиття сигналу, або в закритих приміщеннях, де він повністю відсутній. У таких випадках критичну роль відіграє IMU, яка є комбінацією акселерометрів та гіроскопів. Цей модуль вимірює лінійні прискорення та кутові швидкості, дозволяючи обчислювати орієнтацію дрона та його відносне переміщення. Для компенсації дрейфу інерціальних датчиків та роботи в зонах "GPS-denied" застосовується комп'ютерний зір. Аналізуючи відеопотік з бортових камер, система виділяє ключові точки на об'єктах та орієнтирах, що дозволяє будувати карту глибини та визначати положення дрона візуальним методом. Порівняльний аналіз основних технологічних компонентів наведено у таблиці, табл. 2.2.

Таблиця 2.2 – Порівняльна характеристика сенсорних систем навігації

Тип технології	Функціональні особливості та сфера застосування
GPS/GNSS	Забезпечує високу точність визначення глобальних координат на відкритій місцевості, проте має критичні обмеження стабільності в приміщеннях та зонах радіоперешкод.
IMU (Інерціальний модуль)	Здійснює високочастотний моніторинг динаміки руху (прискорення, нахили), що є базою для стабілізації польоту, але схильний до накопичення похибок.
Технічний зір	Дозволяє виконувати локалізацію та орієнтування у просторі за відсутності супутникового зв'язку шляхом аналізу візуальних сцен.

Програмне ядро	Відповідає за комплексування (фузію) розрізнених даних сенсорів та реалізацію логіки навігаційних алгоритмів.

Перехід до повністю автоматичної навігації передбачає мінімізацію або повне виключення людського фактору з контуру управління. Це досягається шляхом передачі повноважень прийняття рішень бортовому комп'ютеру, який здатен реагувати на зміни ситуації в режимі реального часу. Функціональність такої системи базується на трьох ключових стовпах: автоматичному плануванні маршруту, системі активного уникнення перешкод та адаптивному управлінні.

Процес автоматичного планування маршруту є фундаментальним етапом автономної місії. Він передбачає використання математичних алгоритмів для генерації оптимальної траєкторії руху між стартовою та фінішною точками. При цьому критерії оптимізації можуть варіюватися: від мінімізації пройденої відстані до економії заряду батареї або уникнення зон з несприятливими метеоумовами.

Не менш важливою є підсистема уникнення перешкод, яка перетворює дрон з "сліпого" виконавця команд на "зрячого" робота. Використовуючи дані з лідарів, ультразвукових сонарів або стереокамер, бортовий комп'ютер будує динамічну карту навколишнього середовища. Застосування алгоритмів глибокого машинного навчання, зокрема згорткових нейронних мереж, дозволяє не просто виявляти перешкоди, а й класифікувати їх (наприклад, відрізняти дерево від іншого дрона), що дозволяє приймати більш зважені рішення щодо маневру ухилення.

Третім компонентом є адаптивне управління, яке забезпечує стійкість системи до зовнішніх збурень. У реальних умовах дрон постійно піддається впливу вітру, турбулентності або змін маси вантажу. Алгоритми адаптивного контролю аналізують відхилення від заданої траєкторії та автоматично коригують параметри роботи двигунів для компенсації цих впливів. Це

дозволяє виконувати складні маневри та утримувати стабільну позицію без втручання зовнішнього пілота [60, 61].

Зв'язуючою ланкою всіх апаратних компонентів виступає спеціалізоване програмне забезпечення. Саме на програмному рівні реалізуються графи пошуку шляху, такі як алгоритми Дейкстри або A* (A-star), які інтегрують глобальні карти з локальними даними сенсорів. Критично важливою частиною ПЗ є модуль фільтрації даних, найчастіше реалізований на базі розширеного фільтра Калмана, який відсіює шуми сенсорів та формує єдину узгоджену оцінку стану системи. Детальний опис функціональних модулів програмного комплексу наведено нижче у таблиці, табл. 2.3.

Таблиця 2.3 – Функціональна декомпозиція програмного комплексу керування

Програмний модуль	Основне завдання	Технічна реалізація та деталізація
Обробка сенсорних даних	Первинна фільтрація та згладжування сигналів.	Імплементація алгоритмів сенсорної фузії (зокрема фільтра Калмана) для мінімізації вимірювальних шумів та дрейфу.
Навігаційне ядро	Побудова маршруту та динамічна корекція курсу.	Розрахунок оптимальної траєкторії з урахуванням виявлених перешкод та поточних обмежень простору.
Система інтеграції	Забезпечення системної цілісності та узгодженості.	Синхронізація потоків даних між модулями сприйняття, планування та виконання команд для забезпечення безперебійної роботи.
Модуль безпеки	Активне виявлення та обхід загроз.	Застосування методів комп'ютерного зору для детекції

		об'єктів та генерації команд для екстреного маневрування.
Контур адаптації	Динамічне налаштування параметрів контролера.	Автоматична корекція коефіцієнтів стабілізації польоту у відповідь на зміну зовнішніх метеорологічних факторів.

2.3 Обґрунтування вибору інструментальних засобів та програмного стека реалізації системи

Процес проектування автономних навігаційних комплексів вимагає ретельного підходу до вибору базової мови програмування, оскільки саме вона визначає швидкість розробки, гнучкість архітектури та можливості інтеграції різнорідних апаратних компонентів. У сучаній інженерній практиці для створення програмного забезпечення БПЛА найчастіше розглядають тріаду мов: C++, Java та Python. Кожен із цих інструментів має свою специфіку, що визначає сферу його доцільного використання.

Мова C++ традиційно вважається стандартом для розробки вбудованих систем реального часу. Її головними перевагами є надзвичайно висока продуктивність виконання коду та можливість прямого керування оперативною пам'яттю, що є критичним для мікроконтролерів з обмеженими ресурсами. Проте, ці переваги нівелюються високим порогом входження, складним синтаксисом та значними часовими витратами на налагодження програм, що може сповільнити етап наукових досліджень та прототипування. Java, у свою чергу, пропонує високу стабільність та кросплатформеність завдяки використанню віртуальної машини (JVM). Однак, накладні витрати

ресурсів на роботу віртуальної машини часто роблять її менш привабливою для задач, де критичною є кожна мілісекунда затримки.

На противагу їм, Python демонструє оптимальний баланс для розробки інтелектуальних систем. Його простий синтаксис дозволяє розробникам фокусуватися на логіці алгоритмів, а не на особливостях компіляції. Крім того, наявність розвиненої екосистеми бібліотек для машинного навчання та комп'ютерного зору робить його безальтернативним вибором для задач штучного інтелекту [62, 63, 64]. Порівняльний аналіз зазначених мов програмування наведено у таблиці, табл. 2.4.

Таблиця 2.4 – Порівняльна характеристика мов програмування для задач робототехніки

Мова програмування	Ключові технічні переваги	Обмеження та недоліки
C++	Забезпечує максимальну швидкодію та повний контроль над апаратними ресурсами та пам'яттю.	Характеризується крутою кривою навчання, складністю синтаксису та тривалим циклом розробки.
Java	Гарантує переносимість коду між платформами та високу надійність виконання.	Має знижену продуктивність у системах реального часу через механізми віртуалізації.
Python	Дозволяє швидке прототипування, має лаконічний код та найширшу підтримку AI/ML бібліотек.	Поступається у швидкості виконання обчислювально важких операцій (компенсується C-розширеннями).

При розробці навігаційного забезпечення важливо також визначитися з фреймворками. Фундаментом для сучасної робототехніки є ROS (Robot Operating System) — мета-операційна система, що забезпечує апаратну абстракцію та комунікацію між процесами. Вона підтримує як C++, так і Python, дозволяючи інтегрувати сенсори та симулятори в єдину мережу [65]. Для задач візуального сприйняття стандартом є бібліотека OpenCV, яка надає інструментарій для обробки відеопотоку, детекції маркерів та аналізу сцени [66]. У сфері штучного інтелекту лідерами є платформи TensorFlow та PyTorch, що дозволяють створювати, тренувати та розгортати нейронні мережі для класифікації об'єктів та прийняття рішень [67, 68].

У рамках даної роботи як основний інструмент обрано мову Python. Це рішення обумовлене необхідністю швидкої ітеративної розробки та тестування складних алгоритмів обробки даних, що надходять від сенсорів дрона. Python виступає як "клей", що поєднує високорівневу логіку поведінки дрона з низькорівневими драйверами пристроїв. Завдяки наявності таких модулів, як PySerial та PyUSB, реалізується безшовна інтеграція з апаратними інтерфейсами камер, інерціальних блоків (IMU), GPS-приймачів та лідарів. Схематично процес обробки даних у цьому середовищі зображено на рисунку, рис. 2.3.



Рисунок 2.3 – Схема інформаційних потоків у середовищі Python при обробці сенсорних даних

Функціональні можливості системи значно розширюються завдяки використанню спеціалізованих бібліотек. Математичне ядро забезпечується бібліотеками NumPy та SciPy, які реалізують високоефективні операції з багатовимірними масивами та матрицями, що є необхідним для роботи фільтрів Калмана та обробки сигналів. Задачі технічного зору покладено на OpenCV, що дозволяє в реальному часі аналізувати відеопотік для візуальної одометрії. Для реалізації інтелектуальних функцій використовується TensorFlow або Keras, які надають доступ до попередньо навчених нейромережевих моделей [66, 67]. Крім того, важливим етапом є аналіз результатів, для чого застосовується бібліотека Matplotlib, що дозволяє візуалізувати траєкторії руху та графіки роботи сенсорів [69]. Узагальнений перелік використаних бібліотек подано в таблиці, табл. 2.5.

Таблиця 2.5 – Функціональне призначення бібліотек програмного комплексу

Назва бібліотеки	Сфера застосування в проекті
NumPy, SciPy	Виконання матричних обчислень, наукових розрахунків та реалізація математичних моделей руху.
OpenCV	Алгоритми комп'ютерного зору, попередня обробка зображень, детекція об'єктів.
TensorFlow, Keras	Побудова архітектури нейронних мереж, навчання моделей та інференс у реальному часі.
Matplotlib	Графічна візуалізація телеметрії, побудова графіків похибок та аналіз ефективності алгоритмів.

Підсумовуючи, вибір Python створює гнучке та потужне середовище, яке дозволяє ефективно поєднувати класичні алгоритми навігації з новітніми підходами штучного інтелекту, забезпечуючи при цьому високу швидкість адаптації програмного забезпечення до нових вимог.

2.4 Методологія віртуального прототипування та верифікації алгоритмів керування

Етап віртуального моделювання та лабораторного тестування є невід'ємною складовою життєвого циклу розробки будь-якої автономної системи. Перш ніж переходити до натурних випробувань, які завжди пов'язані з ризиком пошкодження дороговартісного обладнання та небезпекою для оточення, необхідно провести серію експериментів у контрольованому цифровому середовищі. Використання сучасних симуляторів дозволяє реалізувати концепцію Software-in-the-Loop (SITL), де реальний польотний код виконується на комп'ютері, взаємодіючи з моделлю фізичного світу. Це дає

змогу верифікувати коректність роботи навігаційних алгоритмів, оцінити їхню стійкість до збурень та оптимізувати параметри регуляторів без фізичних витрат [70].

На сьогоднішній день індустрія пропонує широкий спектр програмних продуктів для симуляції, кожен з яких має свою спеціалізацію. Одним із найбільш універсальних рішень є середовище Gazebo. Цей симулятор здобув популярність завдяки глибокій інтеграції з робототехнічною екосистемою, підтримці складних фізичних взаємодій та можливості конструювання деталізованих 3D-сцен. Альтернативою є платформа AirSim від корпорації Microsoft, яка базується на рушії Unreal Engine. Її сильною стороною є фотореалістична графіка, що робить її ідеальною для навчання нейромереж комп'ютерного зору. Також варто відзначити пакет RotorS, який спеціалізується на високоточному моделюванні аеродинаміки мультикоптерів, та FlightGear — авіаційний симулятор з відкритим кодом, що часто використовується для відпрацювання загальних польотних сценаріїв [70, 71]. Порівняльний вигляд інтерфейсу симулятора наведено на рисунку, рис. 2.4.

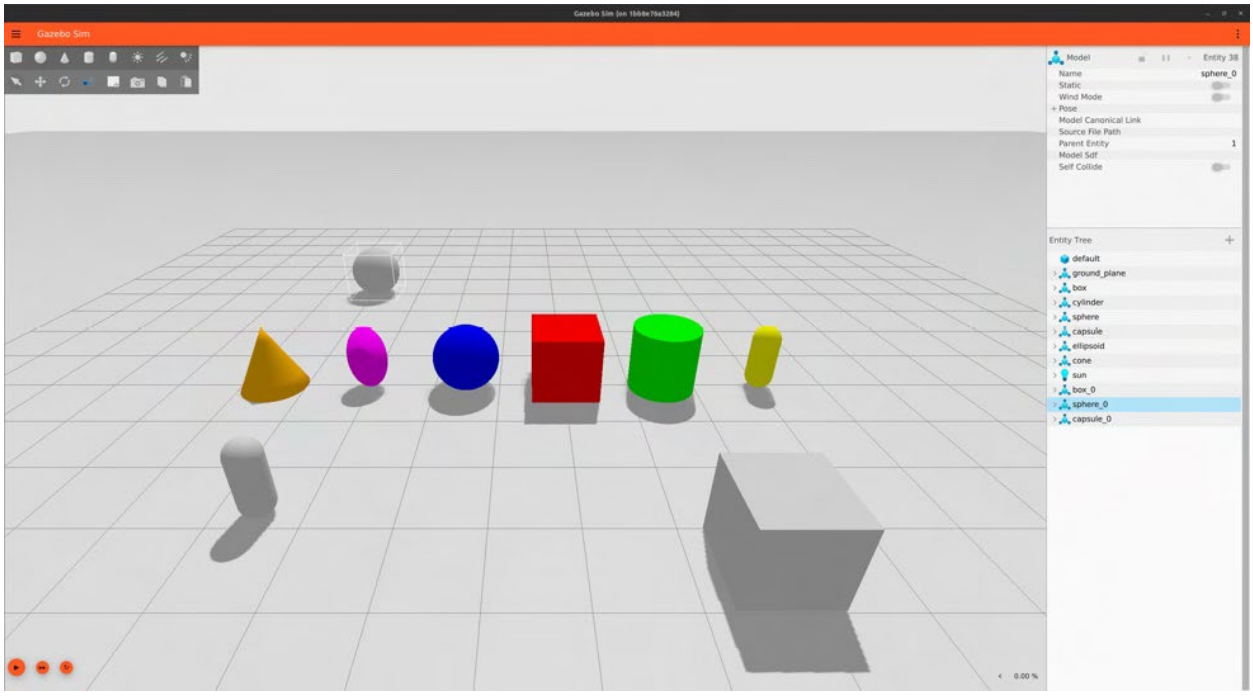


Рисунок 2.4 – Візуалізація процесу симуляції в середовищі Gazebo

У рамках даної роботи як базовий інструмент обрано симулятор Gazebo. Вибір обумовлений його архітектурною сумісністю з ROS та наявністю розробленого Python API. Це дозволяє емулювати роботу всього сенсорного стека дрона (камер, лазерних далекомірів, гіроскопів) та отримувати телеметрію в тому ж форматі, що й на реальному апараті. Такий підхід забезпечує "безшовний" перенос коду з симулятора на бортовий комп'ютер дрона [73].

Фундаментом програмної архітектури проекту виступає ROS (Robot Operating System). Попри назву, це не операційна система в класичному розумінні, а потужний набір бібліотек та інструментів (middleware), що стандартизує розробку робототехніки. ROS вирішує проблему апаратної абстракції, дозволяючи уніфікувати роботу з різними типами камер, модулів GPS та IMU. Головною особливістю ROS є модульна, граф-орієнтована архітектура. Система складається з окремих вузлів (nodes) — незалежних програм, які виконують вузькоспеціалізовані задачі (наприклад, зчитування даних з камери або розрахунок керуючого сигналу) і обмінюються даними через іменовані канали, так звані "топіки" (topics) [74]. Структуру такої взаємодії зображено на схемі.

Завдяки широкому набору вбудованих пакетів, ROS надає готові реалізації алгоритмів SLAM (Simultaneous Localization and Mapping), навігаційних стеків та бібліотек комп'ютерного зору, що значно прискорює розробку. Крім того, пряма сумісність із симулятором Gazebo дозволяє реалізувати замкнений цикл тестування.

Методологія верифікації системи у віртуальному середовищі ROS/Gazebo передбачає послідовне виконання кількох етапів. На першому етапі відбувається формалізація критеріїв успішності місії, до яких відносяться середньоквадратичне відхилення від траєкторії, стабільність утримання висоти та швидкість реакції на перешкоди. Другий етап включає генерацію тестових сценаріїв, що імітують складні умови експлуатації: польоти в "міських колодязях" з обмеженим оглядом супутників,

маневрування в закритих приміщеннях або роботу в умовах вітрового навантаження. Завершальним етапом є аналіз накопичених даних (логів), що дозволяє виявити слабкі місця в логіці уникнення перешкод та провести тонке налаштування алгоритмів перед фінальним завантаженням на польотний контролер.

2.5 Стратегії виключення оператора з контуру керування та алгоритмічне забезпечення автономності

Фундаментальною метою еволюції навігаційних систем безпілотних літальних апаратів є поступова мінімізація або повна елімінація людського фактору з процесу безпосереднього пілотування. Це зумовлено тим, що саме помилки оператора, викликані втомою, затримкою реакції або втратою концентрації, є найчастішою причиною аварійних ситуацій. Перехід до автоматизованого управління дозволяє забезпечити детерміновану, надійну та безпечну поведінку дрона навіть у критично складних умовах навколишнього середовища. Досягнення такої автономності базується на синергії високоефективних алгоритмів регулювання, інтелектуальної обробки сенсорної телеметрії та механізмів адаптації до динамічних збурень.

Технічною основою автоматичного пілотування є побудова замкнених контурів автоматичного керування (Closed-loop control systems). Принцип їх роботи полягає у безперервному порівнянні бажаного стану системи (цільової висоти, швидкості, курсу) з її реальним станом, отриманим на основі моніторингу датчиків. На базі цієї різниці (похибки) система генерує керуючі впливи на двигуни для мінімізації відхилень. Саме наявність негативного зворотного зв'язку дозволяє дрону самостійно компенсувати зовнішні впливи, такі як пориви вітру, зміни атмосферного тиску або зміщення центру мас, забезпечуючи стабілізацію без втручання іззовні.

Стандартом де-факто для низькорівневої стабілізації польоту є використання пропорційно-інтегрально-диференціальних регуляторів (PID-контролерів). Цей математичний алгоритм дозволяє плавно та точно керувати обертами двигунів для утримання заданої орієнтації у просторі. Його ефективність базується на обчисленні керуючого сигналу як суми трьох складових, кожна з яких відповідає за свій аспект реакції на помилку керування.

Перша складова, пропорційна, формує сигнал, що прямо залежить від величини миттєвої помилки: чим більше дрон відхилився від цілі, тим сильнішою буде реакція системи для повернення назад. Друга складова, інтегральна, відповідає за накопичення інформації про помилки у минулому. Вона дозволяє компенсувати систематичні зовнішні впливи, наприклад, постійний боковий вітер, який пропорційна складова не може усунути повністю. Третя складова, диференціальна, працює на випередження, реагуючи на швидкість зміни помилки. Вона діє як демпфер, запобігаючи різким коливанням та "перельотам" цільового значення, забезпечуючи плавність маневрів [75, 76]. Структурну схему такої взаємодії наведено на рисунку, рис. 2.5.

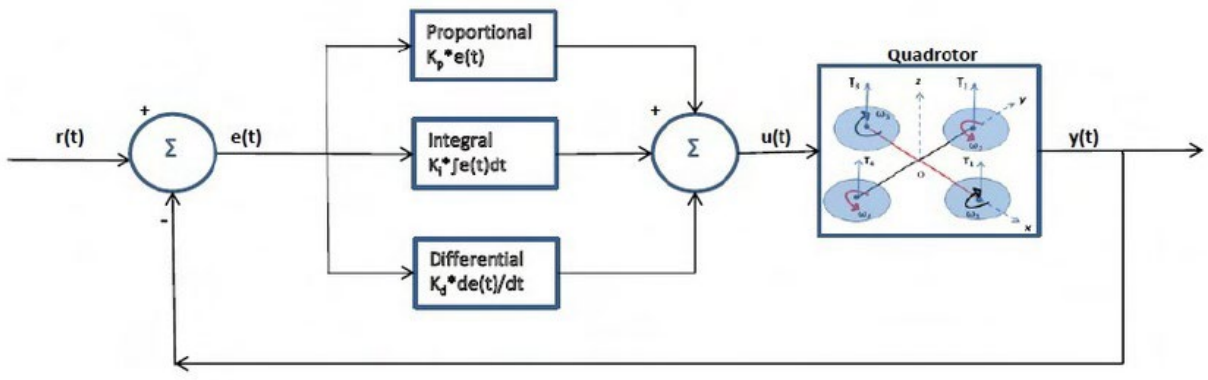


Рисунок 2.5 – Функціональна схема контуру керування зі зворотним зв'язком

Вищий рівень автономності реалізується через підсистеми автоматичного планування місій та ситуаційної обізнаності. Процес прокладання маршруту не обмежується з'єднанням двох точок прямою лінією;

це складна оптимізаційна задача. Бортовий комп'ютер використовує алгоритми пошуку на графах, такі як метод Дейкстри або евристичний алгоритм A* (A-star), для знаходження траєкторії, яка є оптимальною за сукупністю критеріїв: мінімальна дистанція, найменші витрати енергії та безпечна відстань від заборонених зон.

Критично важливою функцією є система активного уникнення перешкод (Sense-and-Avoid). Сучасні дрони використовують дані з лідарів, ультразвукових сонарів та камер глибини для побудови локальної карти зайнятості простору. Це дозволяє виявляти загрози заздалегідь і автоматично перебудовувати маршрут у реальному часі. Окрім реактивних дій, система повинна вміти обробляти нештатні ситуації, такі як раптова відмова одного з сенсорів або різке погіршення метеоумов, переходячи в безпечні режими польоту (Fail-safe).

Вершиною розвитку систем керування є впровадження методів штучного інтелекту, зокрема навчання з підкріпленням (Reinforcement Learning). На відміну від жорстко запрограмованих PID-регуляторів, нейромережеві агенти здатні навчатися на власному досвіді, оптимізуючи стратегію польоту методом проб і помилок у симуляційному середовищі. Це дозволяє дрону адаптуватися до нелінійної аеродинаміки або пошкоджень пропелерів.

Паралельно з цим, згорткові нейронні мережі (CNN) вирішують задачу семантичного сприйняття сцени. Вони дозволяють системі не просто бачити перешкоду як набір точок, а класифікувати її (наприклад, розрізняти гілки дерев, дроти або будівлі). Це забезпечує контекстно-залежну навігацію, де дрон може використовувати візуальні орієнтири для уточнення свого положення, коли супутниковий сигнал недоступний, та приймати більш зважені рішення щодо маневрування [53, 54].

3 ПРОЄКТУВАННЯ ІНТЕГРОВАНОЇ СИСТЕМИ АВТОНОМНОГО НАВЕДЕННЯ БПЛА

3.1 Апаратна архітектура та сенсорне забезпечення навігації

Архітектура системи автономної навігації безпілотного літального апарата являє собою складний симбіоз програмних та апаратних засобів, метою яких є безперервний аналіз навколишнього простору, планування траєкторії та безпосереднє керування рухом. Процес функціонування системи можна розділити на послідовні етапи: сканування оточення, інтерпретація отриманих даних, розрахунок маршруту та його виконання. Взаємодія компонентів починається зі збору інформації про фізичне середовище через набір датчиків, які фіксують параметри статичних і динамічних перешкод. Отриманий масив даних надходить до обчислювального модуля, де відбувається локалізація дрона та формування цифрової моделі простору. На базі цієї моделі будується оптимальний шлях, який коригується в реальному часі під час виконання польотного завдання, рис.3.1.



Рисунок 3.1 – Етапи реалізації автономного польоту дрона

Для забезпечення стабільності та точності навігації використовується комплекс різнорідних сенсорів. Інерційна вимірювальна одиниця (IMU) відповідає за визначення прискорення та кутової швидкості, що є критичним для орієнтації, хоча цей метод схильний до накопичення похибок. Для вимірювання дистанцій та побудови детальних карт застосовуються лідари або стереокамери (RGBD), які забезпечують високу точність, проте залежать від погодних умов та освітлення. Барометричні датчики дозволяють контролювати висоту польоту, а сенсори оптичного потоку допомагають оцінювати зміщення відносно поверхні, що є незамінним за відсутності супутникового сигналу. Камери з підтримкою алгоритмів комп'ютерного зору (наприклад, Yolov5) виконують функцію класифікації об'єктів. Вітрові датчики можуть додатково використовуватися для корекції курсу за складних погодних умов.

Вибір конкретної конфігурації сенсорів базується на пошуку балансу між енергоефективністю, необхідною точністю та вартістю апарата, рис.3.2.



Рисунок 3.2 – Складові елементи навігаційного комплексу

Енергоємні прилади, такі як лідари, значно скорочують час польоту, тоді як IMU та оптичні датчики є більш ощадливими. Для високоточної картографії необхідні дорогі сенсори з високою роздільною здатністю, тоді як для базової стабілізації достатньо простіших рішень.

Загальна схема функціонування передбачає циклічну обробку даних: сенсори збирають інформацію, алгоритми SLAM інтегрують її для побудови карти, планувальник (наприклад, алгоритм A*) прокладає шлях, а контролери забезпечують рух. При виявленні нових перешкод маршрут динамічно перебудовується.



Рисунок 3.3 – Алгоритм функціонування системи

Окремої уваги заслуговують методи підтримки орієнтації. IMU, рис.3.4, що складається з акселерометрів та гіроскопів, обчислює траєкторію шляхом інтеграції даних про лінійне прискорення та кутові швидкості.

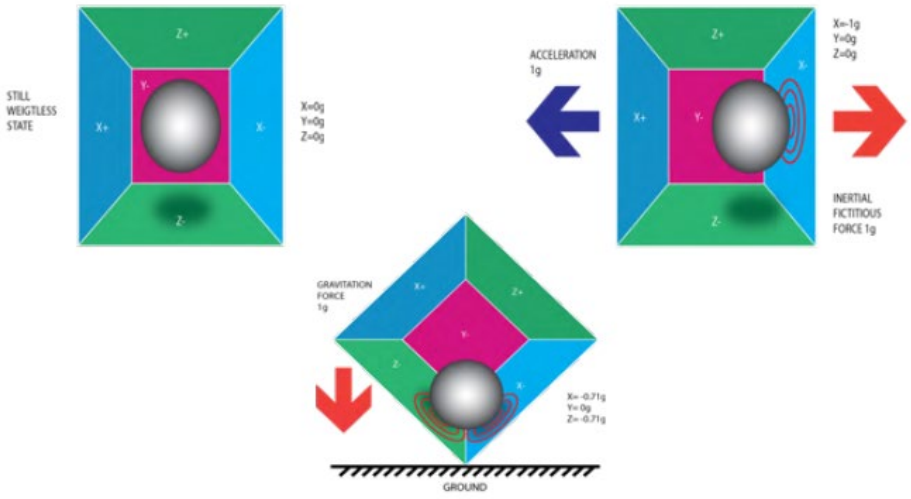


Рисунок 3.4 – Схема роботи інерційного модуля та розрахунку орієнтації

Однак через дрейф показників виникає необхідність у корекції, для чого часто залучають магнітні компаси, що орієнтують дрон відносно магнітного поля Землі, рис.3.5, та сенсори оптичного потоку, які аналізують зміщення пікселів між кадрами для визначення швидкості руху, рис3.6.

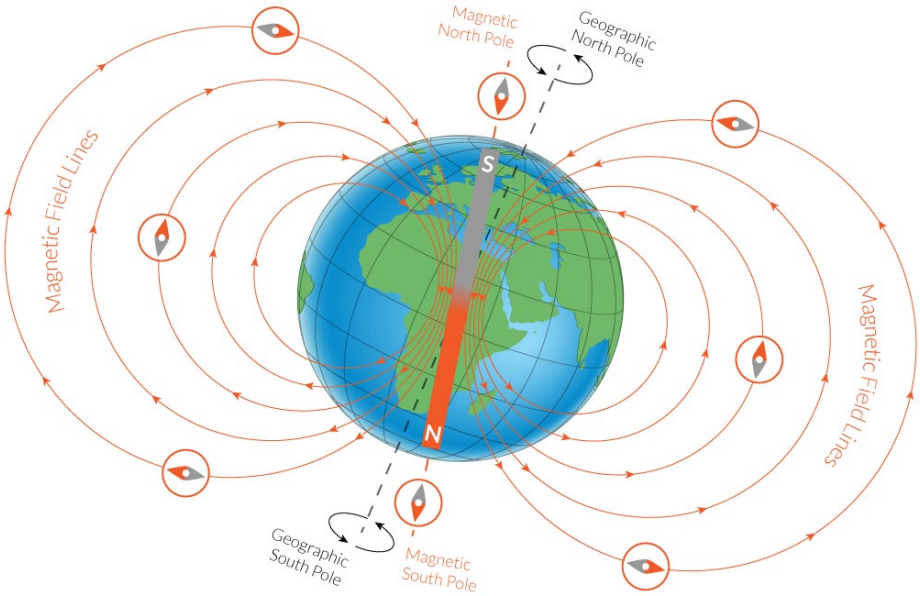


Рисунок 3.5 – Принцип орієнтування за допомогою магнітного компаса



Рисунок 3.6 – Функціональна схема датчика оптичного потоку

Інтеграція даних з усіх джерел здійснюється за допомогою математичних фільтрів, що дозволяє нівелювати слабкі сторони окремих сенсорів та підвищити загальну надійність системи.

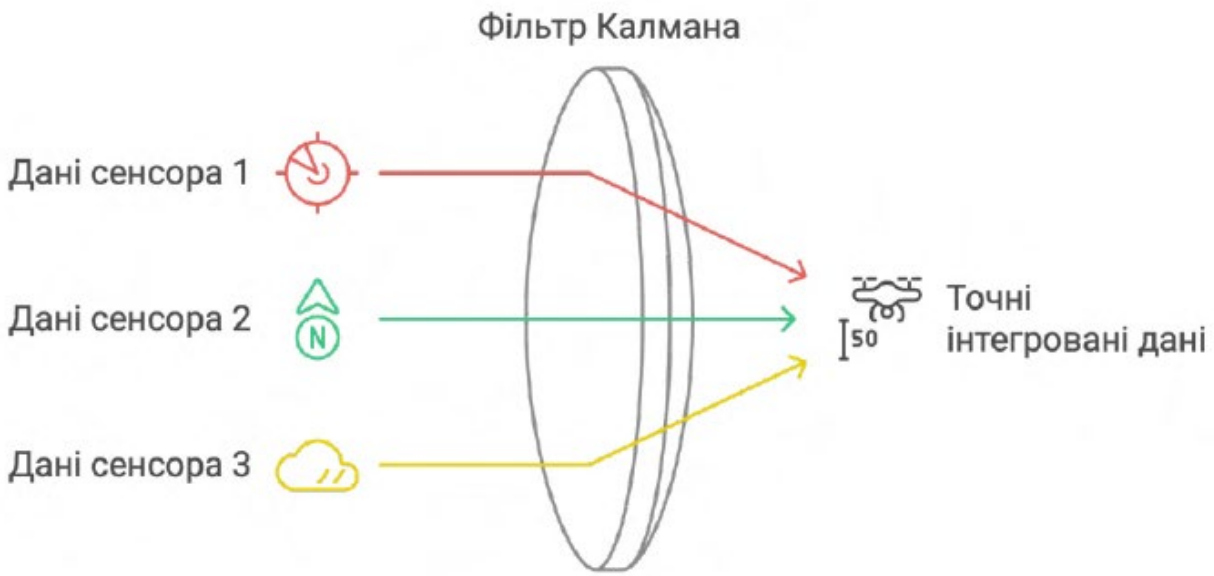


Рисунок 3.7 – Структура об’єднання сенсорних даних

3.2 Математичні методи оцінки стану та картографування

Для обробки зашумлених даних від сенсорів у реальному часі широко застосовується фільтр Калмана. У контексті навігації дронів, яка є нелінійною системою, використовується його розширена версія (EKF). Алгоритм працює у два етапи: прогнозування та оновлення. На етапі прогнозування система оцінює свій майбутній стан на основі попередніх даних та моделі руху. Оскільки реальні процеси нелінійні, для їх спрощення застосовується процес лінеаризації за допомогою матриці Якобіана. На етапі оновлення прогноз коригується з урахуванням реальних вимірювань від датчиків (IMU, компаса, барометра тощо).

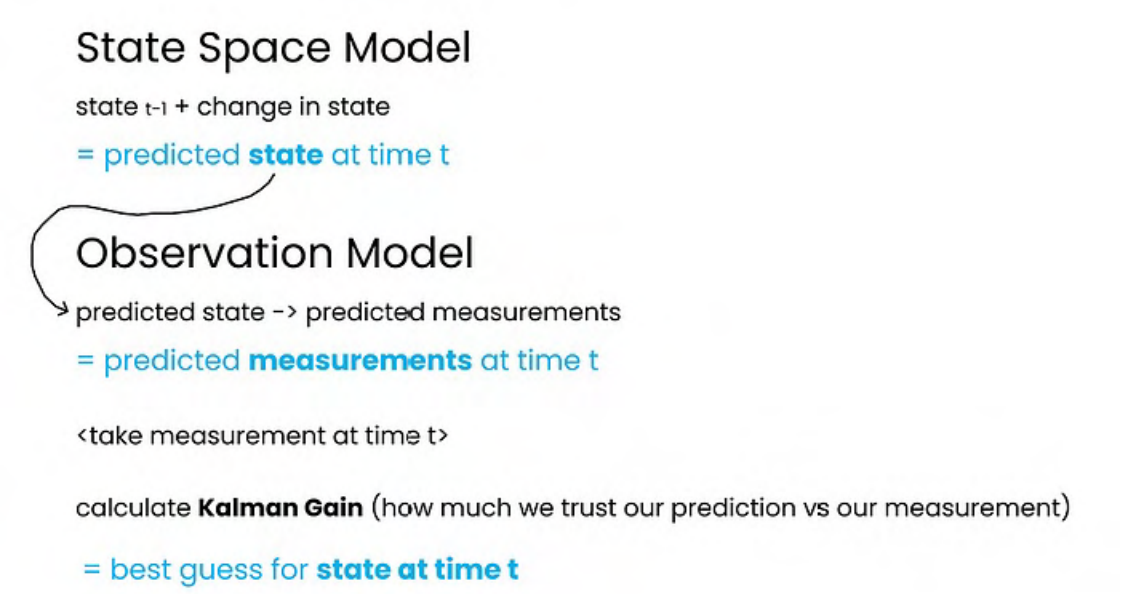


Рисунок 3.8 – Схема алгоритму фільтрації Калмана

Математична сутність методу полягає у роботі з матрицями. У класичному фільтрі Калмана стан системи на поточному кроці визначається як добуток матриці переходу станів на попередній стан плюс добуток матриці керування на вектор керування, з додаванням процесу шуму. Вимірювання описуються як лінійна комбінація станів. У розширеному фільтрі (EKF) замість лінійних матриць використовуються нелінійні функції динаміки та спостереження. Оновлення стану базується на різниці між реальним вимірюванням та прогнозованим значенням, помноженим на коефіцієнт підсилення Калмана. Для лінеаризації нелінійних функцій обчислюється матриця Якобіана, яка є матрицею частинних похідних функції по всіх змінних вектора стану.

Для одночасної навігації та побудови карти використовується метод EKF SLAM. Він вирішує дві задачі: визначає координати дрона та координати орієнтирів (маяків) у просторі. Вектор стану такої системи включає положення робота та координати всіх відомих орієнтирів. Важливим елементом є матриця коваріації, яка описує невизначеність положення робота, невизначеність положення орієнтирів, а також їхню взаємну кореляцію. Саме наявність

кореляції дозволяє уточнювати положення дрона при повторному спостереженні вже відомих орієнтирів.

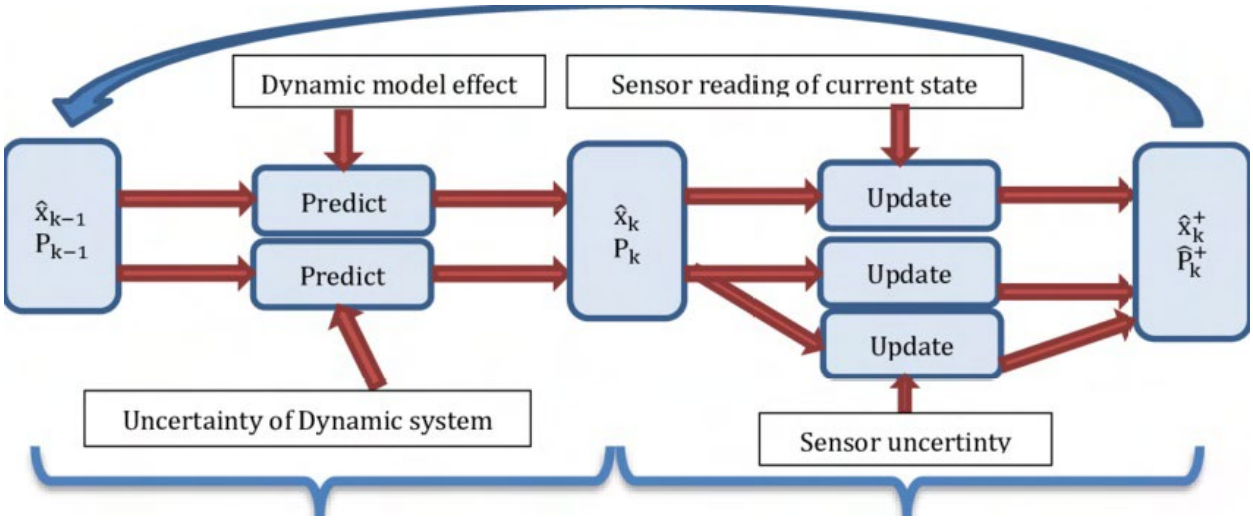


Рисунок 3.9 – Послідовність дій у методі EKF SLAM

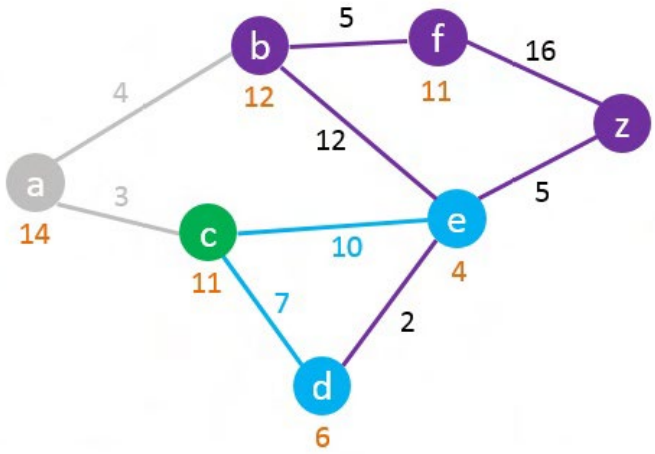
Процес SLAM включає передбачення переміщення дрона, оновлення даних при виявленні орієнтирів та ініціалізацію нових об'єктів на карті. Нові орієнтири додаються до вектора стану на основі виміряної відстані та кута до них. Метод Multi-sensor EKF SLAM дозволяє об’єднувати дані з лідарів, камер та GPS, що підвищує точність. Однак цей підхід має обмеження: обчислювальна складність зростає квадратично зі збільшенням кількості орієнтирів, а лінеаризація може вносити похибки. Для оптимізації роботи з великими картами та складними об'єктами часто застосовують спрощення геометрії перешкод або використовують допоміжні алгоритми комп'ютерного зору для виділення ключових ознак об'єктів.

3.3 Алгоритми планування шляху та розпізнавання об’єктів

Для побудови оптимального маршруту в середовищі з перешкодами використовується алгоритм A* (A-star). Цей метод поєднує стратегію пошуку

в ширину з евристичною оцінкою відстані до мети. Принцип роботи полягає у мінімізації сумарної вартості шляху, яка складається з вартості переміщення від стартової точки до поточної та евристичної оцінки відстані від поточної точки до фінішу. Алгоритм перебирає можливі варіанти руху, зберігаючи їх у списках відкритих (потенційних) та закритих (вже перевірених) вершин, доки не буде знайдено шлях до цілі або не вичерпаються варіанти.

Для дронів A* спочатку генерує глобальний маршрут на основі відомої карти. У процесі польоту, при виявленні нових перешкод сенсорами, траєкторія може динамічно перераховуватися. Перевагами методу є гарантоване знаходження найкоротшого шляху та гнучкість, проте він вимогливий до ресурсів пам'яті у великих просторах.



Node	Status	Shortest Distance From A	Heuristic Distance to Z	Total Distance*	Previous Node
A	Visited	0	14	14	
B		4	12	16	A
C	Current	3	11	14	A
D		∞ 3+7=10	6	16	C
E		∞ 3+10=13	4	17	C
F		∞	11		
Z		∞	0		

* Total Distance = Shortest Distance from A + Heuristic Distance to Z

Рисунок 3.10 – Візуалізація пошуку шляху алгоритмом A*

Важливим компонентом сучасної навігації є інтеграція штучного інтелекту, зокрема нейромережі YOLO (You Only Look Once), для розпізнавання об'єктів. На відміну від класичних методів, YOLO аналізує зображення за один прохід, розбиваючи його на сітку та прогнозуючи наявність об'єктів і їхні класи в кожній клітинці. Це забезпечує високу швидкість обробки, що дозволяє використовувати модель у реальному часі на борту дрона.

Інтеграція YOLO в систему SLAM дозволяє вирішувати кілька завдань: ідентифікувати та фільтрувати динамічні об'єкти (люди, машини), які можуть завадити побудові карти; використовувати розпізнані стабільні об'єкти (знаки, стовпи) як надійні орієнтири; а також створювати семантичні карти, де об'єкти мають не лише геометричні, а й змістові характеристики. Реалізація такої системи передбачає захоплення відеопотоку з камери, його обробку нейромережею для отримання обмежувальних рамок та класів об'єктів, і подальше використання цієї інформації навігаційним модулем для уникнення перешкод та уточнення локалізації.

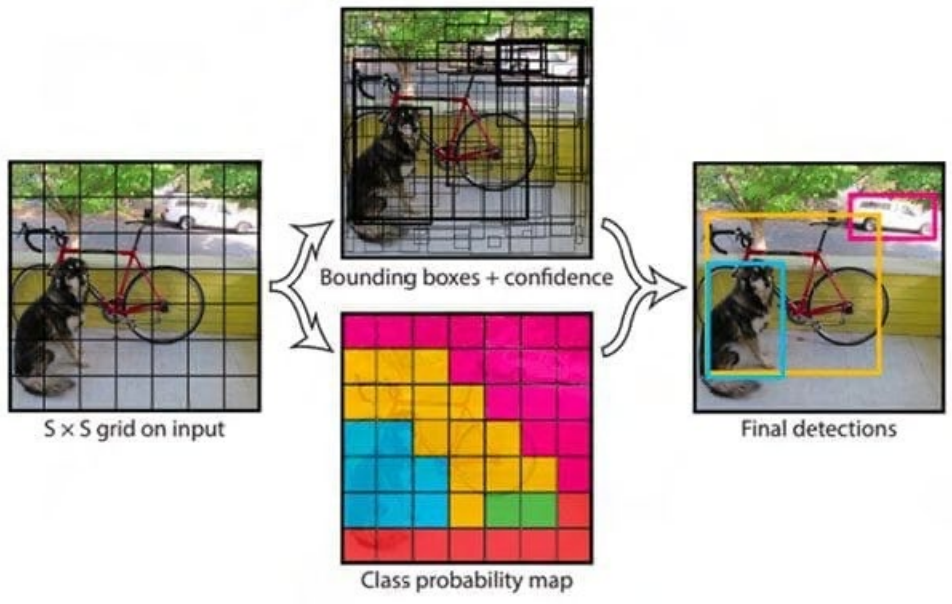


Рисунок 3.11 – Схема функціонування алгоритму детекції об’єктів YOLO

ВИСНОВКИ

У магістерській кваліфікаційній роботі вирішено науково-прикладне завдання щодо створення системи керування для автономного перехоплення повітряних цілей, яка базується на комплексуванні даних від різнорідних сенсорів та використанні штучного інтелекту.

Основні результати роботи полягають у наступному:

1. Проаналізовано проблематику навігації в умовах протидії РЕБ. Встановлено, що покладання виключно на GPS є критичною вразливістю. Доведено необхідність використання інерціальних вимірювальних блоків у поєднанні з візуальними методами навігації.

2. Обґрунтовано використання розширеного фільтра Калмана (EKF) як математичного ядра системи орієнтації. Цей алгоритм дозволяє об'єднувати дані акселерометрів, гіроскопів та комп'ютерного зору, що забезпечує точну оцінку координат та кутів орієнтації дрона-перехоплювача навіть при високій динаміці польоту.

3. Розроблено та імплементовано підсистему комп'ютерного зору на базі архітектури нейронної мережі YOLO. Система забезпечує високу швидкість детекції повітряних об'єктів, що є критичним для задач перехоплення. Продемонстровано, що використання нейромереж дозволяє класифікувати цілі та відрізняти їх від перешкод, як це показано на схемі роботи алгоритму детекції.

4. Реалізовано алгоритми одночасної локалізації та побудови карти для орієнтації у просторі. Це дозволяє перехоплювачу не лише бачити ціль, але й розуміти своє положення відносно навколишніх перешкод, уникаючи зіткнень під час маневрування.

5. Створено програмний комплекс мовою Python з використанням бібліотек ROS. Це дозволило забезпечити модульність системи, де блоки

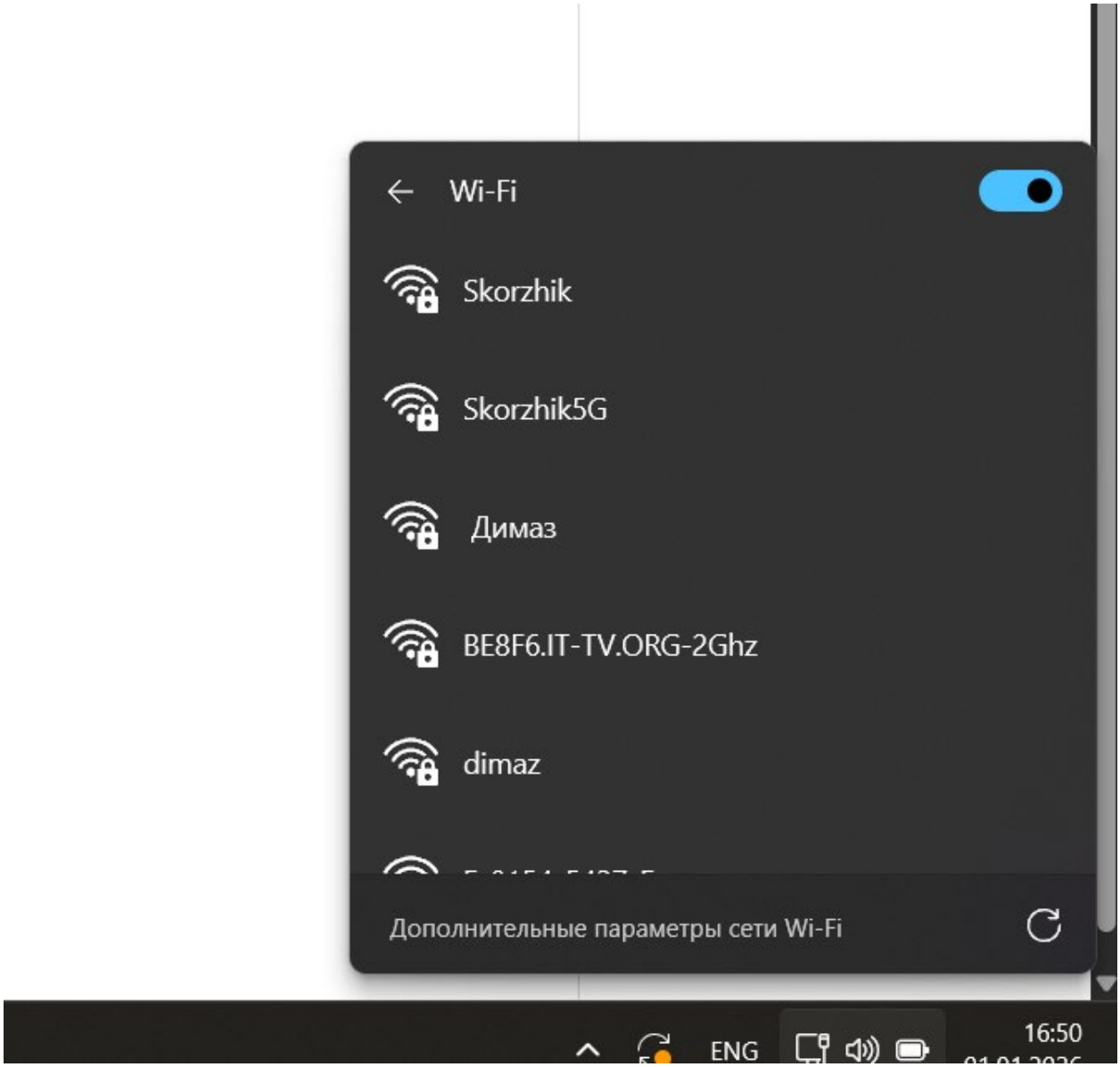
розпізнавання образів, планування траєкторії та керування двигунами працюють як незалежні вузли, обмінюючись даними в реальному часі.

6. Здійснено віртуальне прототипування системи у середовищі Gazebo. Проведені експерименти підтвердили працездатність алгоритмів наведення: дрон успішно розпізнає ціль, будує оптимальний маршрут перехоплення та виконує зближення.

7. Впроваджено елементи адаптивного керування, що дозволяють компенсувати зовнішні збурення, такі як вітер, та дрейф інерціальних датчиків. Використання зворотного зв'язку в контурі керування, гарантує стабільність польоту перехоплювача на всіх етапах місії.

Розроблена система створює фундамент для побудови повністю автономних засобів протиповітряної оборони ближнього радіусу дії, здатних

ефективно функціонувати в умовах сучасної високотехнологічної війни.



СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. J. Zhu et al. "Camera, LiDAR, and IMU Based Multi-Sensor Fusion SLAM: A Survey." Tsinghua Science and Technology, 29 (2024): 415–429. https://doi.org/10.26599/tst.2023.9010010.

2. C. Urrea et al. "Kalman Filter: Historical Overview and Review of Its Use in Robotics 60 Years after Its Creation." Journal of Sensors (2021): 1–21. https://doi.org/10.1155/2021/9674015.

3. K. Ito. "An Optimal Optical Flow." SIAM Journal on Control and Optimization, 44 (2005): 728–742. https://doi.org/10.1137/s0363012904433444.

4. T. Bailey et al. "Consistency of the EKF-SLAM Algorithm." 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (2006). https://doi.org/10.1109/iros.2006.281644.

5. Bo Dai et al. "SIMSF: A Scale Insensitive Multi-Sensor Fusion Framework for Unmanned Aerial Vehicles Based on Graph Optimization." IEEE Access (2020): 1-1. 10.1109/ACCESS.2020.3000555.

A. Alfarano et al. "Estimating optical flow: A comprehensive review of the state of the art." Computer Vision and Image Understanding (2024): 104160. https://doi.org/10.1016/j.cviu.2024.104160.

6. S. Popowski et al. "Individual Autonomous Navigation System." Transactions on Aerospace Research (2017): 84–106. https://doi.org/10.2478/tar-2017-0023.

7. V. Tchernykh et al. "Optical Flow Navigation For An Outdoor UAV Using A Wide Angle Mono Camera And DEM Matching." IFAC Proceedings Volumes, 39 (2006): 590–595. https://doi.org/10.3182/20060912-3-de-2911.00103.

8. Ullah et al. "Simultaneous Localization and Mapping Based on Kalman Filter and Extended Kalman Filter." Wireless Communications and Mobile Computing (2020): 1–12. https://doi.org/10.1155/2020/2138643

9. E. Sandström et al. "UncLe-SLAM: Uncertainty Learning for Dense Neural SLAM." 2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW) (2023). https://doi.org/10.1109/iccvw60793.2023.00488.

10. Zulu et al. "A Review of Control Algorithms for Autonomous Quadrotors." Open Journal of Applied Sciences, 04 (2014): 547–556. https://doi.org/10.4236/ojapps.2014.414053.

11. F. Saeed Faizi et al. "Visual-Based Simultaneous Localization and Mapping (VSLAM) Techniques for Robots: A Scientific Review." Academic Journal of Nawroz University, 12 (2023): 213–229. https://doi.org/10.25007/ajnu.v12n3a1500.

12. M. S. A. Khan et al. "Multi-Sensor SLAM for efficient Navigation of a Mobile Robot." 2021 4th International Conference on Computing & Information Sciences (ICCIS) (2021). https://doi.org/10.1109/iccis54243.2021.9676374.

13. D. C. Li et al. "Integrated Extended Kalman Filter and Deep Learning Platform for Electric Vehicle Battery Health Prediction." Applied Sciences, 14 (2024): 4354. https://doi.org/10.3390/app14114354.

14. T. Takleh Omar Takleh et al. "A Brief Survey on SLAM Methods in Autonomous Vehicle." International Journal of Engineering & Technology, 7 (2018): 38. https://doi.org/10.14419/ijet.v7i4.27.22477.

15. Armah. "Implementation of Autonomous Navigation Algorithms on Two-Wheeled Ground Mobile Robot." American Journal of Engineering and Applied Sciences, 7 (2014): 149–164. https://doi.org/10.3844/ajeassp.2014.149.164.

16. M. Holland. "Introduction to the Extended Kalman Filter." Nova Science Publishers (2020).

17. L. A. McGee et al. "Discovery of the Kalman Filter as a Practical Tool for Aerospace and Industry." NASA Technical Memorandum (1985). https://ntrs.nasa.gov/api/citations/19860003843/downloads/19860003843.pdf.

18. GeeksforGeeks. "A* Search Algorithm." GeeksforGeeks. https://www.geeksforgeeks.org/a-search-algorithm/.

19. Advanced Navigation. "Inertial Measurement Unit (IMU) – An Introduction." Advanced Navigation. https://www.advancednavigation.com/tech-articles/inertialmeasurement-unit-imu-an-introduction/.

20. IBM. "What is Computer Vision?" IBM. https://www.ibm.com/topics/computer-vision

21. Amazon Web Services. "What is Computer Vision? - Image recognition AI/ML Explained." AWS. https://aws.amazon.com/what-is/computer-vision/.

22. G. M. Smith. "What is Sensor Fusion?." Data Acquisition | Test and Measurement Solutions. https://dewesoft.com/blog/what-is-sensor-fusion.

23. NVIDIA. "Recent posts for: Optical Flow SDK." NVIDIA Technical Blog. https://developer.nvidia.com/blog/recentposts/?products=Optical+Flow+SDK.

24. All About Circuits. "Rudolf Kalman: The Creator of Kalman Filters and Modern Control Theory." All About Circuits. https://www.allaboutcircuits.com/news/rudolf-kalman-the-creator-of-kalmanfilters-modern-control-theory/.

25. NASA. "What Was the Apollo Program? (Grades 5-8)." NASA. https://www.nasa.gov/learning-resources/for-kids-and-students/what-was-the-apolloprogram-grades-5-8/.

26. Cornell University. "Kalman Filter Applications." Department of Computer Science. https://www.cs.cornell.edu/courses/cs4758/2012sp/materials/MI63slides.pdf.

27. MathWorks. "Understanding Kalman Filters, Part 3: Optimal State Estimator." MATLAB & Simulink. https://www.mathworks.com/videos/understanding-kalman-filters-part-3-optimal-stateestimator--1490710645421.html.

28. Zucconi. "The Mathematics of the Kalman Filter." Alan Zucconi. https://www.alanzucconi.com/2022/07/24/kalman-gain/.

29. Kalman Filter Tutorial. "Online Kalman Filter Tutorial." Kalman Filter Tutorial. https://www.kalmanfilter.net/default.aspx.

30. W. Franklin. "Kalman Filter Explained Simply." The Kalman Filter. https://thekalmanfilter.com/kalman-filter-explained-simply/.

29. Zucconi. "The Extended Kalman Filter." Alan Zucconi. https://www.alanzucconi.com/2022/07/24/extended-kalman-filter/.

30. Satya. "Sensor Fusion With Kalman Filter." Medium. https://medium.com/@satya15july_11937/sensor-fusion-with-kalman-filterc648d6ec2ec2.

31. MathWorks. "Example: Estimate 2-D Target States with Angle and Range Measurements Using trackingEKF." MATLAB & Simulink. https://www.mathworks.com/help/fusion/ug/extendedkalman-filters.html.

32. S. Przybylski. "The math behind Extended Kalman Filtering." Medium. https://medium.com/@sasha_przybylski/the-math-behind-extended-kalmanfiltering-0df981a87453.

33. Advanced Navigation. "Dead-Reckoning." Advanced Navigation. https://www.advancednavigation.com/glossary/dead-reckoning/.

34. RF Wireless World. "What is Dead reckoning navigation method?." RF Wireless World. https://www.rfwireless-world.com/Articles/Advantages-and-disadvantages-of-Dead-Reckoning-navigation-method.html.

35. MathWorks. "What Is SLAM (Simultaneous Localization and Mapping)." MATLAB & Simulink. https://www.mathworks.com/discovery/slam.html.

36. E. McMinn. "Understanding SLAM in Robotics and Autonomous Vehicles." Flyability. https://www.flyability.com/blog/simultaneous-localization-and-mapping.

37. L. Bermudez. "Overview of SLAM." Medium. https://medium.com/machinevision/overview-of-slam-50b7f49903b7.

38. A3. "What is Visual SLAM Technology and What is it Used For?." A3. https://www.automate.org/vision/blogs/what-is-visual-slam-technology-and-what-is-itused-for.

39. S. Malik. "Lidar SLAM: The Ultimate Guide to Simultaneous Localization and Mapping." Wevolver. https://www.wevolver.com/article/lidarslam.

40. MathWorks. "Monocular Visual-Inertial SLAM." MATLAB & Simulink. https://www.mathworks.com/help/vision/ug/monocular-visual-inertial-slam.html.

41. Sally Robotics. "A review on ORB-SLAM-2 paper." Medium. https://medium.com/@sallyrobotics.blog/a-review-on-orb-slam-2-paper-3554d4fcaa7c.

42. D. Kohler et al. "Introducing Cartographer." Google Open Source Blog. https://opensource.googleblog.com/2016/10/introducingcartographer.html.

43. GitHub. "rovio." GitHub. https://github.com/ethz-asl/rovio.

44. Google Cloud. "What Is Artificial Intelligence (AI)?." Google Cloud. https://cloud.google.com/learn/what-is-artificial-intelligence.

45. Visionplatform. "AI and Drones: Advancements and Applications in Unmanned Aerial Vehicles." visionplatform. https://visionplatform.ai/artificialintelligence-drones.

46. D. Bacon. "AI and the Future of Drone Technology." Toll Uncrewed Systems. https://tolluncrewedsystems.com/blog/ai-andthe-future-of-drone-technology/.

47. Robotnik. "How do AI robots work? Artificial intelligence and mobile robotics." Robotnik. https://robotnik.eu/how-do-ai-robots-work-artificialintelligence-and-mobile-robotics/.

48. Evergreen. "Що таке навчання з підкріпленням? Розбираємо теорію і реальні кейси." Evergreen. https://evergreens.com.ua/ua/articles/reinforcement-learning.html.

49. IT-Enterprise. "Machine Learning, ML." IT-Enterprise. https://www.it.ua/knowledge-base/technologyinnovation/machine-learning.

50. C. C. O. Bootcamps. "What Is Machine Learning and How Does It Work?." Medium. https://medium.com/@caltechbootcamps/https-www-simplilearn-com-tutorials-machine-learning-tutorial-what-is-machine-learning-df7b91f10f6.

51. IBM. "What are Convolutional Neural Networks?." IBM. https://www.ibm.com/topics/convolutional-neural-networks.

52. GeeksforGeeks. "Introduction to Convolution Neural Network." GeeksforGeeks. https://www.geeksforgeeks.org/introductionconvolution-neural-network/.

53. EasyODM. "How To Perform Object Detection With CNNS." EasyODM.tech. https://easyodm.tech/object-detection-with-cnns/.

54. STMicroelectronics. "Give your product an Edge." STMicroelectronics - STM32 AI. https://stm32ai.st.com/.

55. Arduino. "Nano 33 BLE Sense Rev2." Arduino Docs. https://docs.arduino.cc/hardware/nano-33-ble-sense-rev2/.

56. Espressif Systems. "ESP32-S3 Wi-Fi & BLE 5 SoC." Espressif Systems. https://www.espressif.com/en/products/socs/esp32-s3.

57. TransiTiva. "How Does a UAV Navigation System Work?." TransiTiva. https://transitiva.com/knowledgebase/how-does-a-uav-navigation-system-work/.

58. MathWorks. "Autonomous Navigation." MATLAB & Simulink. https://www.mathworks.com/videos/series/autonomousnavigation.html.

59. Sriram. "Exploring Software Solutions for Training and Deploying Machine Learning Models on Drones." Medium. https://medium.com/@cloudgeek/exploring-software-solutions-for-training-anddeploying-machine-learning-models-on-drones-a0b5d95d840d.

60. G. D. Rocha. "#101 Developing Autonomous Drones Using Python and AI." Medium. https://medium.com/@genedarocha/101-developing-autonomousdrones-using-python-and-ai-3a85bc79a80b.

61. ROS. "ROS 2 Documentation: Foxy documentation." ROS Documentation. https://docs.ros.org/en/foxy/index.html.

62. OpenCV. "Home." OpenCV. https://opencv.org/.TensorFlow. "Introduction to TensorFlow." TensorFlow. https://www.tensorflow.org/learn.

63. PyTorch. "PyTorch." PyTorch. https://pytorch.org/.

64. Matplotlib. "Matplotlib documentation." Matplotlib. https://matplotlib.org/stable/.

65. Industry Wired. "Top 10 Tools for Developing Autonomous Drones." Industry Wired. https://industrywired.com/top-10-tools-for-developing-autonomous-drones/.

66. Upcoming Engineer. "Best Robot Simulation Software." Upcoming Engineer. https://upcomingengineer.com/best-robot-simulation-software/.

67. Gazebo. "Understanding the GUI." Gazebo ionic documentation. https://gazebosim.org/docs/latest/gui/.

68. Gazebo. "Gazebo Transport: Python Support." Gazebo. https://gazebosim.org/api/transport/14/python.html.

69. Automatic Addison. "ROS 2 Architecture Overview." Automatic Addison. https://automaticaddison.com/ros-2-architecture-overview/.

70. NI. "The PID Controller & Theory Explained." Mess- und Prüfsysteme, bei Emerson. https://www.ni.com/en/shop/labview/pid-theory-explained.html.

71. RealPars. "PID Controller Explained." YouTube. https://www.youtube.com/watch?v=fv6dLTEvl74.

72. UAV Navigation. "GNSS-Denied Navigation Kit." UAV Navigation. https://www.uavnavigation.com/products/navigation-systems/gnss-denied-navigationkit.

73. M. Hansen. "Solutions for GNSS-Denied Simulation and Training." Kongsberg Digital. https://www.kongsbergdigital.com/resources/solutions-for-gnss-denied-simulation-andtraining.

74. Washington University in St. Louis. "Inertial Measurement Unit (IMU) Use." ESE205 Wiki. https://classes.engineering.wustl.edu/ese205/core/index.php?title=Inertial_Measurement_Unit_(IMU)_Use.

75. Dalvey. "How does a compass work." Dalvey Blog. https://www.dalvey.com/in/blog/how-does-a-compass-work.

76. Robotics and Perception Group. "Event Cameras, Event camera SLAM, Event-based Vision." UZH. https://rpg.ifi.uzh.ch/research_dvs.html.

77. Dioram. "What is an event-based camera?." dioram. https://dioramslam.com/2021/10/08/event-cameras/.

Додаток А

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)