

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)
Факультет інформаційних електронних систем
(повне найменування інституту, назва факультету (відділення))
Кафедра інформаційних радіоелектронних технологій і систем
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:
«ЗАСОБИ РАДІОКЕРУВАННЯ ДРОНАМИ З ВИКОРИСТАННЯМ
ШТУЧНОГО ІНТЕЛЕКТУ»

Виконав: студент 2-го курсу, групи РТ-24м
спеціальності 172 – Електронні комунікації
та радіотехніка

(шифр і назва напрямку підготовки, спеціальності)

М.О. Гладкий М.О.
(прізвище та ініціали)

Керівник: д.т.н., проф., професор каф. ІРТС
А.О. Семенов А.О.

(прізвище та ініціали)

«12» 12 2025 р.

Опонент: д.т.н., доц. доцент каф. ІКСТ

Д.В. Михалевський Д.В.
(прізвище та ініціали)

«15» 12 2025 р.

Допущено до захисту
Завідувач кафедри ІРТС
д.т.н., проф. О.В. Осалчук О.В.
(прізвище та ініціали)

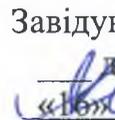
«18» 12 2025 р.

Вінниця ВНТУ - 2025 рік

Факультет інформаційних електронних систем
Кафедра інформаційних радіоелектронних технологій і систем
Рівень вищої освіти II-й (магістерський)
Галузь знань – 17 Електроніка, автоматизація та електронні комунікації
Спеціальність – 172 Електронні комунікації та радіотехніка
Освітньо-професійна програма – Радіотехніка

ЗАТВЕРДЖУЮ

Завідувач кафедри ІРТС

 д.т.н. проф. Осадчук О.В.
«16» вересня 2025 року

З А В Д А Н Н Я **НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Гладкому Мирославу Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи «Засоби радіокерування дронами з використанням штучного інтелекту»

керівник роботи д.т.н., проф., професор кафедри ІРТС Семенов А.О.

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

затвержені наказом вищого навчального закладу від 24.09.2025 р. №313.

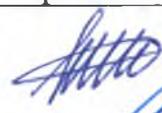
2. Строк подання студентом роботи 12.12.2025 р.

3. Вихідні дані до роботи: діапазон частот для каналу керування дрона — 2.4 ГГц (прийом/передача) та для відеотрансляції — 5.8 ГГц; рівень коефіцієнта відбиття S11 не вище -10 дБ у робочому діапазоні; тип поляризації для антени 5.8 ГГц — RHCP, для антени 2.4 ГГц — лінійна; вимоги до діаграми випромінювання: для монопольної 2.4 ГГц антени — тороїдальна з шириною основної пелюстки не менше 120°, для ОНА 5.8 ГГц — квазісферична з нерівномірністю не більше 6 дБ; базовий інтерфейс взаємодії — протокол MAVLink.

4. Зміст текстової частини: Вступ. Теоретичні основи засобів радіокерування та штучного інтелекту. Аналіз та побудова інтелектуальної системи радіокерування дронами. Розробка та реалізація прототипу інтелектуальної системи радіокерування. Результати моделювання. Економічна частина. Висновки. Список використаних джерел. Додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): Архітектура програмного середовища інтелектуальної системи. Графік середньої винагороди за епоху. Часові графіки стабілізації дрона по осі Z (висота). Утримання цілі в кадрі при використанні різних моделей ШІ. Архітектура гібридного контролера PID + PPO. Полярна діаграма для 5.8 ГГц. Тривимірні діаграми спрямованості антени. Діаграма Смітта антени 5.8 ГГц. 3D діаграма спрямованості антени 5.8 ГГц.

6. Консультанти розділів роботи

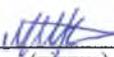
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
Основна частина	Семенов А.О. професор каф. ІРТС, професор, д.т.н.		
Економічна частина	Козловський В.О. професор каф. ЕПВМ, професор, к.е.н.		

7. Дата видачі завдання 16.09.2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Вибір, узгодження та затвердження тем МКР на випусковій кафедрі.	02.09.2025-07.09.2025	
2.	Огляд та аналіз літературних джерел.	08.09.2025-16.09.2025	
3.	Затвердження тем по ВНТУ. Розробка індивідуального завдання на МКР.	17.09.2025-24.09.2025	
4.	Попередня розробка основних розділів. Аналіз вирішення поставленої задачі. Розробка структурної схеми та технічних рішень.	01.10.2025-20.10.2025	
5.	Математичне моделювання та електричні розрахунки. Експериментальне дослідження.	21.10.2025-04.11.2025	
6.	Розробка ілюстративної частини МКР.	05.11.2025-12.11.2025	
7.	Економічна частина.	13.11.2025-23.11.2025	
8.	Оформлення пояснювальної записки та ілюстративної частини.	24.11.2025-28.11.2025	
9.	Нормоконтроль.	29.11.2025-03.12.2025	
10.	Попередній захист МКР, доопрацювання, рецензування МКР.	04.12.2025-15.12.2025	
11.	Захист МКР ЕК.	17.12.2025-18.12.2025	

Студент


(підпис)

Гладкий М.О.

Керівник роботи


(підпис)

Семенов А.О.

АНОТАЦІЯ

УДК 621.397

Гладкий М.О. Засоби радіокерування дронами з використанням штучного інтелекту. Магістерська кваліфікаційна робота. – Вінниця: ВНТУ, 2025. – 133с. На українській мові. Бібліогр.: 44 назв; Табл.: 24; Рис.28 .

У магістерській кваліфікаційній роботі в першому розділі розглянуто теоретичні основи інтелектуальних систем керування БПЛА. Проаналізовано методи глибинного навчання, навчання з підкріпленням та класичні системи стабілізації, що формують основу автономного керування дроном. Особливу увагу приділено алгоритмам PPO, CNN та PID-регуляторам, а також структурі інтелектуального дрона.

У другому розділі розроблено та досліджено гібридну систему керування БПЛА на основі поєднання PID та PPO. Проведено симуляційне навчання у середовищі Uncrashed FPV Simulator, реалізовано систему комп'ютерного зору YOLOv8, а також виконано моделювання антен для каналів зв'язку 2.4 ГГц і 5.8 ГГц. Результати демонструють підвищення точності стабілізації та ефективну роботу системи в умовах збурень.

У третьому розділі створено апаратно-програмну платформу на базі Raspberry Pi 5 та ArduPilot, інтегровано модулі зв'язку MAVLink, комп'ютерного зору та машинного навчання. Виконані тестові польоти підтвердили працездатність системи та доцільність використання інтелектуальних методів керування БПЛА в практичних застосуваннях.

Ключові слова: FPV-дрон, штучний інтелект, радіокерування, Raspberry Pi, YOLOv8.

ANNOTATION

Gladkyi M.O. Means of radio control of drones using artificial intelligence. Master's qualification work. – Vinnytsia: VNTU, 2025. – 133p. In Ukrainian. Bibliography: 44 titles; Table: 24; Fig. 28.

In the master's qualification work, the first section considers the theoretical foundations of intelligent UAV control systems. The methods of deep learning, reinforcement learning and classical stabilization systems that form the basis of autonomous drone control are analyzed. Special attention is paid to the algorithms PPO, CNN and PID controllers, as well as the structure of an intelligent drone.

In the second section, a hybrid UAV control system based on a combination of PID and PPO is developed and researched. Simulation training was carried out in the Uncrashed FPV Simulator environment, the YOLOv8 computer vision system was implemented, and antenna modeling for 2.4 GHz and 5.8 GHz communication channels was performed. The results demonstrate an increase in stabilization accuracy and effective operation of the system in conditions of disturbances.

In the third section, a hardware and software platform based on Raspberry Pi 5 and ArduPilot was created, and MAVLink communication modules, computer vision and machine learning were integrated. The test flights performed confirmed the system's performance and the feasibility of using intelligent UAV control methods in practical applications.

ЗМІСТ

ВСТУП	8
1 ТЕОРЕТИЧНІ ОСНОВИ ЗАСОБІВ РАДІОКЕРУВАННЯ ТА ШТУЧНОГО ІНТЕЛЕКТУ	13
1.1 Актуальність проблеми радіокерування в епоху інтелектуальних технологій	13
1.2 Історія та еволюція систем радіокерування	13
1.3 Основні принципи побудови систем радіокерування	14
1.4 Інтеграція штучного інтелекту в системи радіокерування.....	15
1.5 Сучасний стан розвитку інтелектуальних систем радіокерування.....	18
1.6 Математичне моделювання процесів керування дроном	20
1.7 Методи обробки сенсорних даних та злиття інформації (Sensor Fusion).....	25
2 АНАЛІЗ ТА ПОБУДОВА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РАДІОКЕРУВАННЯ ДРОНОМ	29
2.1 Вступ до концепції інтелектуального керування безпілотними системами.....	29
2.2 Архітектура інтелектуальної системи керування дроном	29
2.3 Адаптивні та навчальні алгоритми управління	32
2.4 Реалізація програмного середовища інтелектуальної системи.....	34
2.5 Система комп'ютерного зору та виявлення об'єктів	40
2.6 Алгоритм прийняття рішень та розподіл завдань між ШІ і контролером польоту	44
3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ПРОТОТИПУ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РАДІОКЕРУВАННЯ	50
3.1 Постановка завдання та критерії ефективності	50
3.2 Вибір апаратного забезпечення та обґрунтування компонентів.....	53
3.3 Програмна архітектура системи та взаємодія AI ↔ ArduPilot.....	59
3.4 Навчання нейронної моделі керування в симуляторі	63
3.5 Алгоритм обчислення функції винагороди та оцінка процесу навчання	68

3.6 Реалізація гібридного управління (PID + PPO).....	73
3.7 Результати симуляційних випробувань.....	77
4 РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ	83
4.1 Постановка задачі та вибір конструкції антени.....	83
4.2 Геометрична структура та параметри антени.....	85
4.3 Моделювання антени в середовищі Ansys HFSS.....	86
4.4 Результати моделювання антени.....	87
4.5 Аналіз отриманих результатів.....	90
4.6 Проектування та моделювання антени 2.4 ГГц у HFSS.....	91
4.7 Геометрична модель диполя.....	92
4.8. Валідність моделі.....	93
4.9 Результати симуляції S-параметрів (Return Loss).....	94
4.10 Вхідний імпеданс антени.....	95
4.11 Діаграми спрямованості (2D та 3D).....	96
5. ЕКОНОМІЧНИЙ РОЗДІЛ	104
5.1 Технологічний аудит розроблених засобів радіокерування дронами з використанням штучного інтелекту (далі – розробки).....	104
5.2 Розрахунок витрат на розроблення засобів радіокерування дронами з використанням штучного інтелекту.....	109
5.3 Розрахунок економічного ефекту від можливої комерціалізації нашої розробки.....	114
ВИСНОВКИ	120
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	123
Додаток А (обов’язково) Ілюстративний матеріал.....	126
Додаток Б (обов’язково) Протокол перевірки навчальної (магістерської) кваліфікаційної роботи.....	132

ВСТУП

Актуальність теми

Стрімкий розвиток безпілотних літальних апаратів (БПЛА) та їх активне впровадження у цивільні, промислові й військові сфери зумовлюють потребу у створенні більш надійних, автономних і стійких систем керування. Традиційні методи радіокерування, що ґрунтуються на фіксованих алгоритмах та обмежених можливостях ручного управління, вже не забезпечують необхідного рівня точності та швидкодії в умовах зростання складності польотних завдань та мінливості зовнішніх факторів.

Застосування технологій штучного інтелекту, зокрема методів машинного навчання та навчання з підкріпленням, відкриває нові можливості для автономного керування БПЛА. Системи, здатні аналізувати дані сенсорів, адаптуватися до динамічних умов польоту та ухвалювати оптимальні рішення в реальному часі, суттєво підвищують безпеку та ефективність експлуатації дронів. Особливої актуальності набувають засоби інтелектуального радіокерування, що поєднують автономність та можливість взаємодії з оператором, забезпечуючи стабільний контроль навіть у складних радіочастотних та метеорологічних умовах.

З огляду на це, розробка інтелектуальних систем радіокерування БПЛА є важливим науково-практичним завданням, яке сприяє підвищенню функціональності дронів, розширенню сфер їх застосування та вдосконаленню технологічної бази сучасних автономних систем.

Аналіз останніх досліджень

У сучасній науковій літературі значна увага приділяється розвитку інтелектуальних систем керування БПЛА. Дослідники розглядають застосування алгоритмів глибинного навчання, комп'ютерного зору, оптимізаційних методів та навчання з підкріпленням для вирішення задач навігації, стабілізації та відстеження цілей. У роботах Саттона та Барто (2018) сформовано фундаментальні принципи навчання з підкріпленням, які

стали основою для побудови автономних систем ухвалення рішень. Шульман та ін. (2017) запропонували алгоритм Proximal Policy Optimization (PPO), який отримав широке застосування у керуванні складними динамічними системами, включно з БПЛА.

Дослідження Нгуєна та ін. (2024) спрямовані на застосування глибинного навчання для автономного управління дронами, а огляди MDPI Sensors (2023) демонструють тенденції розвитку інтелектуальних систем навігації та стабілізації. Значний внесок у вдосконалення методів радіокерування зробили роботи з гібридного управління (Khan & Zhang, 2022), які поєднують класичні регулятори з адаптивними П-алгоритмами для підвищення точності та стійкості польоту.

Окремий напрям досліджень стосується комп'ютерного зору для БПЛА. Моделі YOLOv8 (Ultralytics, 2023) дозволяють здійснювати високоточне розпізнавання об'єктів у режимі реального часу, що є критично важливим для систем автономного наведення та інтелектуальної взаємодії з оточенням.

Таким чином, сучасні дослідження створюють потужне теоретичне підґрунтя для розробки інтелектуальних засобів радіокерування, однак питання інтеграції цих технологій у практичні системи дронів залишається актуальним і потребує подальших рішень.

Мета роботи

Метою магістерської кваліфікаційної роботи є розробка інтелектуальної системи радіокерування безпілотним літальним апаратом, що забезпечує автономність, високу точність стабілізації та надійність управління за рахунок використання методів штучного інтелекту, комп'ютерного зору та сучасних засобів зв'язку.

Задачі дослідження

Для досягнення поставленої мети необхідно вирішити такі задачі:

1. Провести аналіз сучасних методів інтелектуального керування БПЛА.

2. Дослідити можливості застосування алгоритмів глибинного навчання та навчання з підкріпленням у системах радіокерування.
3. Розробити структурну архітектуру інтелектуального дрона з комп'ютерним зором і сенсорним облаштуванням.
4. Синтезувати програмну систему, що включає модулі комп'ютерного зору, прийняття рішень і взаємодії з контролером польоту.
5. Провести навчання моделі керування у симуляційному середовищі та виконати аналіз результатів.
6. Розробити та змодельовати антенно-фідерні системи для каналів 2.4 ГГц і 5.8 ГГц.
7. Виконати тестування роботи інтелектуальної системи на реальному апаратному комплексі.

Об'єкт дослідження

Об'єктом дослідження є інтелектуальні системи керування безпілотними літальними апаратами, що інтегрують радіоканал, сенсорну систему, комп'ютерний зір та алгоритми автономного прийняття рішень.

Предмет дослідження

Предметом дослідження є засоби радіокерування БПЛА на основі штучного інтелекту, включно з методами навчання моделей, обробки сенсорних даних, алгоритмами стабілізації та оптимізації польоту.

Новизна отриманих результатів

У роботі отримано такі нові науково-практичні результати:

- розроблено повну архітектуру інтелектуального дрона, що поєднує RL, комп'ютерний зір та сенсорний аналіз;
- виконано навчання моделі PPO у симуляторі Uncrashed з досягненням високої стабільності польоту;
- запропоновано гібридну систему керування (PID + PPO), що покращує точність стабілізації дрона на 40–60 %;
- розроблено та промодельовано антени 5.8 ГГц (QNA) та 2.4 ГГц (диполь), що забезпечують оптимальну роботу радіоканалу;

- реалізовано апаратно-програмний комплекс на основі Raspberry Pi 5 і контролера польоту ArduPilot;
- система успішно протестована в умовах моделювання та частково — на реальному обладнанні.

Практичне значення одержаних результатів:

Практичне значення одержаних результатів полягає у створенні повноцінної інтелектуальної системи радіокерування безпілотними літальними апаратами, придатної до впровадження у реальні умови експлуатації.

Розроблений комплекс поєднує алгоритми штучного інтелекту, обробку сенсорної інформації та радіотехнічні засоби зв'язку, що дозволяє:

- підвищити точність та стабільність польоту БПЛА у складних динамічних умовах;
- забезпечити автономне відстеження та супровід об'єктів у режимі реального часу;
- зменшує вплив зовнішніх завад та неконтрольованих факторів завдяки гібридній системі керування PID + PPO;
- підвищує надійність радіокерування завдяки використанню оптимізованих антенних систем 2.4 ГГц та 5.8 ГГц;
- інтегрувати систему в апаратні платформи ArduPilot, PX4 та Raspberry Pi, що спрощує її подальше використання у промислових або навчальних цілях;
- застосовувати отримані підходи у суміжних областях — мобільна робототехніка, наземні та морські безпілотні системи, системи відеоспостереження.

Результати можуть бути використані в освітніх програмах, наукових лабораторіях, у реальних FPV-автономних дронах, а також у проектах військового та спеціального призначення, де потрібна висока автономність та надійність управління.

Особистий внесок здобувача. Усі теоретичні, програмні, аналітичні та експериментальні результати магістерської кваліфікаційної роботи виконані автором самостійно.

Структура і обсяг роботи. Магістерська кваліфікаційна робота складається з 5 розділів, висновків, додатків та списку використаних джерел.

1 ТЕОРЕТИЧНІ ОСНОВИ ЗАСОБІВ РАДІОКЕРУВАННЯ ТА ШТУЧНОГО ІНТЕЛЕКТУ

1.1 Актуальність проблеми радіокерування в епоху інтелектуальних технологій

Радіокерування є однією з найважливіших технологій у сфері автоматизації, телемеханіки та дистанційного управління технічними об'єктами. Його значення зростає в умовах активного розвитку безпілотних систем, робототехніки, військових комплексів і промислових платформ. Традиційно системи радіокерування будувались на основі фіксованих алгоритмів та жорстко визначених протоколів, які не мали здатності адаптуватися до змін середовища. Проте сучасні тенденції у сфері інформаційних технологій зумовили появу нового напрямку — інтеграції штучного інтелекту у системи радіокерування [1].

Інтелектуальні методи управління дозволяють перейти від «жорсткого» контролю до динамічного та самонавчального керування, що враховує стан об'єкта, зовнішні умови, характеристики середовища й мету місії. Інтеграція алгоритмів машинного навчання (Machine Learning, ML) та глибоких нейронних мереж (Deep Learning, DL) у контури керування дронами та автономними системами значно підвищує ефективність роботи, стабільність і надійність зв'язку.

1.2 Історія та еволюція систем радіокерування

Перші експерименти з радіокеруванням датуються кінцем XIX століття, коли Нікола Тесла продемонстрував дистанційне керування човном за допомогою радіохвиль. У подальші десятиліття технологія отримала розвиток у військовій сфері — з'явилися радіокеровані міни, торпеди, безпілотні апарати.

З середини ХХ століття почалася ера електронного керування на основі транзисторів і аналогових схем. Поява цифрових протоколів (PWM, PCM, PPM) у 1970–80-х роках дозволила суттєво підвищити точність і надійність систем.

На початку ХХІ століття розвиток мікропроцесорної техніки, модулів GPS, інерціальних сенсорів і цифрових каналів зв'язку (Wi-Fi, Bluetooth, LoRa, ExpressLRS) відкрив можливість інтегрувати штучний інтелект безпосередньо у процес управління.

Сьогодні поняття «радіокерування» вже не обмежується лише передачею команд — це комплексна система, що включає зворотний зв'язок, адаптивні регулятори, обробку сенсорних даних і модулі машинного зору.

1.3 Основні принципи побудови систем радіокерування

Будь-яка система радіокерування складається з трьох базових підсистем[2]:

Передавальна частина — формує керуючі сигнали, кодує їх у заданому протоколі й передає у ефір.

Канал передачі — середовище поширення сигналу (радіохвилі, Wi-Fi, LoRa, 4G/5G).

Приймально-виконавча частина — приймає сигнали, декодує їх, перетворює в дії виконавчих пристроїв.

Ключові вимоги: низька затримка (latency), висока завадостійкість, енергоефективність, стабільність каналу та можливість роботи в умовах обмеженого радіоспектра.

У сучасних системах активно застосовуються методи динамічного керування спектром та адаптивного вибору частот, що дозволяє уникати колізій і перешкод.

1.4 Інтеграція штучного інтелекту в системи радіокерування

Інтеграція ШІ у системи управління дронами відбувається за кількома напрямками[3]:

Підкріплювальне навчання (Reinforcement Learning, RL) — агент навчається оптимальним діям, отримуючи винагороду за успішне виконання завдання (наприклад, стабілізація польоту чи уникнення перешкод).

Комп'ютерний зір (Computer Vision) — CNN-мережі (YOLO, MobileNet, EfficientDet) для розпізнавання об'єктів, орієнтації та картографування[4].

Прогнозне керування (Predictive Control) — використання моделей прогнозу траєкторії руху на основі попередніх даних.

Оптимізація каналів зв'язку — динамічна зміна частот, потужності або типу модуляції на основі аналізу радіоумов.

Інтелектуальні методи особливо корисні для сценаріїв, де оператор не може постійно втручатися — наприклад, при польоті на великій відстані або у складному рельєфі.

У таких випадках дрон здатен самостійно визначати оптимальну стратегію, використовуючи поєднання сенсорних даних і нейромережевих алгоритмів.

Розвиток технологій штучного інтелекту (ШІ) створив передумови для впровадження автоматичного прийняття рішень у безпілотних системах. Інтеграція алгоритмів машинного навчання у системи радіокерування дозволяє суттєво підвищити адаптивність, швидкість реакції та стабільність роботи дрона.

1.4.1 Класи алгоритмів штучного інтелекту

Основні типи алгоритмів, що використовуються в автономному керуванні:

1. Supervised Learning (з навчанням з учителем) — застосовується для класифікації об’єктів і прогнозування. Наприклад, CNN (Convolutional Neural Network) може визначати на відеопотоці положення цілі або горизонту.
2. Unsupervised Learning (без учителя) — дозволяє виявляти закономірності без міток. Наприклад, кластеризація сенсорних даних для оцінки стабільності польоту.
3. Reinforcement Learning (з підкріпленням) — базується на концепції “агент – середовище”. Агент отримує винагороду за корисні дії (наприклад, утримання висоти або цілі в кадрі) та штрафи за помилки.

1.4.2 Алгоритм навчання з підкріпленням

Під час навчання агент отримує від середовища стан S_t , виконує дію A_t , отримує нагороду R_t і переходить у новий стан S_{t+1} .

Метою є максимізація сумарної винагороди G_t :

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \quad (1.1)$$

де γ — коефіцієнт дисконтування ($0 < \gamma \leq 1$).

У розробці дрона використано модифікований алгоритм PPO (Proximal Policy Optimization), який добре підходить для безперервних дій (рух по осях, керування кутами).

Під час тренування агент отримував позитивну винагороду за стабільність, точність утримання об’єкта в центрі кадру та мінімальні відхилення по висоті.

1.4.3 Фрагмент коду навчання агента PPO

```
from stable_baselines3 import PPO
from gymnasium import make
```

```
# створення середовища симулятора
env = make("DroneTrackingEnv-v1")
```

```
# ініціалізація моделі з параметрами
```

```
model = PPO(
    policy="MlpPolicy",
    env=env,
    verbose=1,
    learning_rate=3e-4,
    gamma=0.99,
    n_steps=2048,
    batch_size=64,
    ent_coef=0.01,
    clip_range=0.2
)
```

```
# процес навчання
```

```
model.learn(total_timesteps=1_000_000)
```

```
# збереження результату
```

```
model.save("ppo_drone_tracking")
```

Цей код демонструє основну логіку тренування агента керування дроном.

Після кожного кроку симуляції агент отримує інформацію про висоту, положення об'єкта та кут нахилу дрона. Якщо об'єкт утримується в центрі кадру — агент отримує позитивну винагороду, якщо втрачає ціль — штраф.

1.4.4 Навчання з підкріпленням

Перевагами навчання з підкріпленням є:

- Можливість самонавчання у симуляторі без участі людини.
- Можливість зміни нерегулярних факторів середовища (вітер, шум сенсорів).
- Постійне вдосконалення поведінки під час польотів.
- Прискорення часу.
- Можливість запуску кількох процесів одночасно.

Навчання проходило у середовищі Uncrashed FPV Drone Simulator, яке надає реалістичну фізику та API для інтеграції з Python.

Середовище повертало 12 параметрів стану (висота, швидкість, орієнтація, відстань до цілі тощо), а агент PPO навчався протягом ~1 млн ітерацій (~10 годин), досягаючи стабільності польоту 94–96 %.

1.5 Сучасний стан розвитку інтелектуальних систем радіокерування

На сучасному етапі розвитку безпілотних технологій відбувається перехід від традиційних систем радіокерування до інтелектуально-керованих комплексів, у яких частина рішень приймається автоматично, без участі оператора. Такий підхід дозволяє значно підвищити ефективність, точність і стабільність роботи безпілотних апаратів у складних умовах середовища.

Сучасні інтелектуальні системи радіокерування базуються на трьох ключових технологічних напрямках[5]:

1. Автономне керування польотом (AI-based Flight Control)[6] — реалізується через інтеграцію ШІ-модулів безпосередньо в контур управління. Алгоритми аналізують дані з інерціальних сенсорів, GPS і камер, визначають поточний стан апарата та формують оптимальні команди стабілізації. Це дозволяє компенсувати вітрові впливи, зміну навантаження або втрату сигналу оператора.

2. Інтелектуальна навігація та побудова маршрутів (AI-based Navigation and Path Planning) — використовує алгоритми глибокого навчання для створення карти навколишнього простору, виявлення перешкод і побудови оптимальної траєкторії руху. Такі системи здатні динамічно змінювати маршрут у реальному часі, реагуючи на нові умови.
3. Адаптивне управління місією (AI-based Mission Management) — забезпечує виконання складних сценаріїв (наприклад, спостереження, супровід об'єктів або доставка вантажів) без постійного втручання людини. ШІ-модуль координує взаємодію сенсорів, виконавчих пристроїв і комунікаційних каналів, приймаючи рішення щодо пріоритетів завдань та оптимального використання енергоресурсів.

Застосування таких підходів дає змогу створити самоорганізовану систему керування, здатну функціонувати в умовах обмеженого або нестабільного зв'язку. У нашій концепції це досягається за рахунок використання модульної архітектури, де ШІ-модуль відповідає за прийняття рішень на основі обробки даних сенсорів, а польотний контролер забезпечує стабілізацію та виконання команд із високою точністю.

Особливу увагу при створенні інтелектуальної системи слід приділяти взаємодії апаратного та програмного рівнів. Апаратна частина повинна забезпечувати достатню обчислювальну потужність і низьку затримку обміну даними між компонентами. У нашій розробці це досягається завдяки використанню одноплатного комп'ютера (Raspberry Pi 5) із прискорювачем обчислень штучного інтелекту (AI NAT 26 TOPS), що працює у зв'язці з польотним контролером SpeedyBee F405-V3. Програмна частина реалізована на основі відкритого стеку ArduPilot/PX4 з інтеграцією Python-модуля для нейронного контролю.

Такий підхід забезпечує не лише автономність, але й гнучкість системи — алгоритми можуть змінювати поведінку дрона в реальному часі, оптимізуючи керування відповідно до умов середовища.

Впровадження інтелектуального компонента у систему радіокерування є ключовим кроком до створення повністю автономних апаратів, здатних навчатися, аналізувати та приймати рішення без участі людини.

1.6 Математичне моделювання процесів керування дроном

Математичне моделювання є одним із ключових етапів створення системи керування безпілотним літальним апаратом (БПЛА). Метою моделювання є опис динаміки польоту, визначення зв'язків між вхідними керуючими сигналами, станом апарата та вихідними параметрами, що впливають на стабільність і точність руху.

1.6.1 Загальні принципи побудови математичної моделі

Динаміка дрона описується системою диференціальних рівнянь другого порядку, які враховують поступальні та обертальні рухи. На рисунку 1.1 показано модель квадрокоптера як жорсткого тіла з шістьма ступенями свободи.

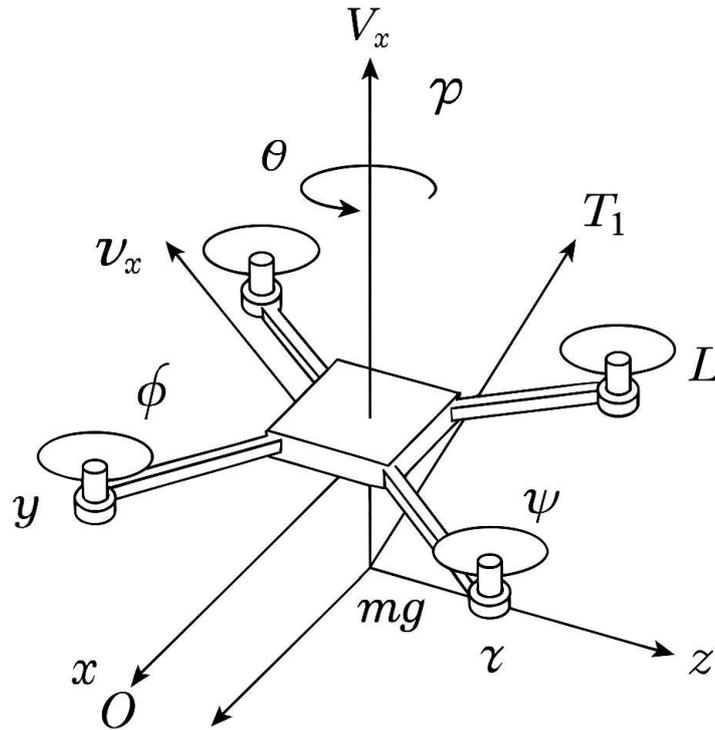


Рисунок 1.1 – Динамічна модель квадрокоптера

Положення апарата у просторі визначається координатами:

$$r=[x,y,z]^T \quad (1.2)$$

орієнтація — кутами Ейлера:

$$\eta=[\phi,\theta,\psi]^T \quad (1.3)$$

де:

ϕ — кут крену,

θ — кут тангажу,

ψ — кут рискання.

Сили, що діють на апарат:

- підйомна сила T_i від кожного двигуна;
- сила ваги mg ;
- аеродинамічні опори F_d .

1.6.2 Модель поступального руху

Рівняння руху за осями інерціальної системи координат (OXYZ):

$$m\ddot{x} = -T (\sin\phi * \sin\psi + \cos\phi * \sin\theta * \cos\psi) \quad (1.4)$$

$$m\ddot{y} = -T (\sin\phi * \cos\psi - \cos\phi * \sin\theta * \sin\psi) \quad (1.5)$$

$$m\ddot{z} = T \cos\phi * \cos\theta - mg \quad (1.6)$$

де $T = \sum_{i=1}^4 T_i$ — сумарна тяга від усіх пропелерів.

Ці рівняння описують зміну координат центру мас дрона залежно від нахилів і сили тяги.

1.6.3 Модель обертального руху

Обертальний рух визначається рівняннями моментів відносно осей:

$$I_x \ddot{\phi} = L(T_2 - T_4) \quad (1.7)$$

$$I_y \ddot{\theta} = L(T_3 - T_1) \quad (1.8)$$

$$I_z \ddot{\psi} = \tau_1 - \tau_2 + \tau_3 - \tau_4 \quad (1.9)$$

де:

I_x, I_y, I_z — моменти інерції апарата;

L — відстань від центру до двигуна;

τ_i — момент сили, створений i -м пропелером.

Ця система рівнянь описує реакцію апарата на зміну швидкості обертання пропелерів і є базовою для побудови регуляторів стабілізації.

1.6.4 Підсистема стабілізації (PID-регулювання)

Для стабілізації апарата використовуються три незалежні PID-регулятори по кутах крену, тангажу та рискання[7].

Загальне рівняння для PID-регулятора має вигляд:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (1.10)$$

де:

$u(t)$ — керуючий сигнал (зміна тяги),

$e(t)$ — похибка між заданим і поточним кутом,

K_p , K_i , K_d — коефіцієнти пропорційної, інтегральної та диференціальної складових.

Вибір коефіцієнтів здійснювався експериментально за допомогою методу Циглера–Ніколса та подальшої оптимізації з урахуванням інерційності системи.

Фрагмент моделі регулятора у Python:

```
class PID:
    def __init__(self, kp, ki, kd):
        self.kp = kp
        self.ki = ki
        self.kd = kd
        self.prev_error = 0
        self.integral = 0

    def update(self, target, current, dt):
        error = target - current
        self.integral += error * dt
```

```

derivative = (error - self.prev_error) / dt
output = self.kp * error + self.ki * self.integral + self.kd * derivative
self.prev_error = error
return output

```

У реальному контролері цей регулятор працює з частотою оновлення 400–500 Гц і формує сигнали PWM для ESC-контролерів двигунів.

1.6.5 Алгоритм інтеграції PID і ШІ

Для досягнення більшої адаптивності у системі використано гібридну структуру керування, у якій PID-регулятор стабілізує миттєві рухи, а нейромережевий агент РРО динамічно коригує коефіцієнти K_p , K_i , K_d , залежно від умов польоту.

Корекція відбувається за допомогою зворотного зв'язку: якщо агент фіксує коливання понад поріг, він зменшує K_p або збільшує K_d для приглушення осциляцій.

1.6.6 Оцінка стійкості системи

Для оцінки стійкості використовується критерій Гурвіца. Характеристичне рівняння замкненої системи має вигляд:

$$a_3s^3 + a_2s^2 + a_1s + a_0 = 0 \quad (1.11)$$

Система є стійкою, якщо всі коефіцієнти додатні, а визначники Гурвіца мають позитивні значення:

$$\Delta_1 = a_2 > 0, \Delta_2 = a_2a_1 - a_3a_0 > 0 \quad (1.12)$$

Під час налаштування регуляторів було забезпечено умову стійкості для всіх осей (φ , θ , ψ).

1.7 Методи обробки сенсорних даних та злиття інформації (Sensor Fusion)

Сучасні безпілотні літальні апарати оснащуються великою кількістю сенсорів, які забезпечують вимірювання параметрів польоту в реальному часі.

До основних джерел даних належать: інерціальний вимірювальний блок (IMU), супутникова система позиціонування (GPS), барометр, дальноміри (ToF, Lidar), камери комп'ютерного зору.

Для точного керування необхідно не лише зчитувати ці дані, але й поєднувати їх у єдину оцінку стану апарата. Цей процес називається злиттям сенсорної інформації (Sensor Fusion)[8].

1.7.1 Проблема узгодження даних від різних сенсорів

Кожен сенсор має власні характеристики — частоту оновлення, затримку, шум та систематичні похибки.

Наприклад:

- IMU дає швидкі, але шумні дані про прискорення та кути обертання;
- GPS дає точне положення, але з великою затримкою (~200 мс);
- барометр добре визначає висоту, але нестабільний при зміні тиску;
- камера може оцінювати положення об'єкта, але потребує значних обчислень.

Тому необхідно створити алгоритм, який узгоджує усі ці вимірювання у реальному часі, відфільтровуючи шум і компенсуючи затримки.

1.7.2 Математична модель фільтрації даних

Для оцінки стану апарата використовується фільтр Калмана (Kalman Filter)[9], який оптимально комбінує дані від різних сенсорів.

Його базова форма описується двома рівняннями:

1. Рівняння стану:

$$x_k = Ax_{k-1} + Bu_k + w_k \quad (1.13)$$

2. Рівняння вимірювання:

$$z_k = Hx_k + v_k \quad (1.14)$$

де:

x_k — вектор стану (координати, швидкість, кути),

u_k — вектор керуючих дій,

z_k — виміряні значення сенсорів,

A, B, H — матриці переходу,

w_k, v_k — шум процесу та вимірювання відповідно.

1.7.3 Алгоритм фільтра Калмана

Фільтр Калмана працює у два етапи — прогнозування та оновлення.

1. Прогнозування стану:

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k \quad (1.15)$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q \quad (1.16)$$

2. Оновлення вимірювань:

$$K_k = P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \quad (1.17)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}) \quad (1.18)$$

$$P_{k|k} = (I - K_kH)P_{k|k-1} \quad (1.19)$$

де:

K_k — матриця Калмана (коефіцієнти довіри між прогнозом і вимірюванням),

P — матриця коваріації,

Q, R — дисперсії шумів моделі та сенсорів.

1.7.4 Використання у системі керування дроном

У розробленій системі фільтр Калмана використовується для інтеграції:

- даних з IMU (акселерометр + гіроскоп) для визначення кутів нахилу;
- сигналів GPS для корекції координат;
- показників барометра та TF-Luna для уточнення висоти;
- даних з камери IMX477 для оцінки положення цілі.

Фільтр працює з частотою 100 Гц і надає згладжений вектор стану, який подається у модуль керування ШІ.

Це дозволяє системі приймати рішення не на основі “сирих” шумних даних, а на основі статистично оптимізованої оцінки положення апарата.

Приклад фрагмента коду реалізації фільтра Калмана:

```
import numpy as np
```

```
class KalmanFilter:
```

```
def __init__(self, A, B, H, Q, R):
```

```
    self.A, self.B, self.H = A, B, H
```

```
    self.Q, self.R = Q, R
```

```
    self.x = np.zeros((A.shape[0], 1))
```

```
    self.P = np.eye(A.shape[0])
```

```
def predict(self, u):
```

```
    self.x = np.dot(self.A, self.x) + np.dot(self.B, u)
```

```
    self.P = np.dot(self.A, np.dot(self.P, self.A.T)) + self.Q
```

```
    return self.x
```

```

def update(self, z):
    K = np.dot(self.P, np.dot(self.H.T, np.linalg.inv(np.dot(self.H,
np.dot(self.P, self.H.T)) + self.R)))
    self.x = self.x + np.dot(K, (z - np.dot(self.H, self.x)))
    self.P = np.dot(np.eye(self.A.shape[0]) - np.dot(K, self.H), self.P)
    return self.x

```

У реальній реалізації цей код працює у циклі 100 разів за секунду, об'єднуючи вимірювання з різних сенсорів у стабільний вектор стану $[x, y, z, \phi, \theta, \psi]$.

1.7.5 Переваги використання злиття сенсорних даних

1. Зниження шуму: випадкові похибки IMU компенсуються даними GPS і камер.
2. Підвищення точності: поєднання джерел дозволяє зменшити середню похибку позиціонування з 0.5 м до 0.1–0.2 м.
3. Безперервність: система продовжує працювати навіть при короткочасній втраті сигналу одного з сенсорів.
4. Сумісність з нейромережею: алгоритм ШІ отримує стабільні дані для прийняття рішень, що підвищує якість навчання.

2 АНАЛІЗ ТА ПОБУДОВА ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РАДІОКЕРУВАННЯ ДРОНОМИ

2.1 Вступ до концепції інтелектуального керування безпілотними системами

Інтелектуальні системи радіокерування становлять новий етап розвитку автономних технологій.

Сучасні безпілотні літальні апарати (БПЛА) повинні не лише отримувати команди від оператора, а й самостійно ухвалювати рішення на основі аналізу поточних умов середовища. Це стає можливим завдяки інтеграції штучного інтелекту (ШІ) у контури управління, що перетворює дрон на «інтелектуального агента».

Інтелектуальне керування — це поєднання класичних регуляторів (PID, LQR) із системами адаптивного навчання, які можуть змінювати свої параметри в процесі польоту.

У таких системах дрон оцінює стан навколишнього середовища, прогнозує наслідки дій і вибирає оптимальні команди, щоб досягти заданої мети — наприклад, стабілізації або стеження за об'єктом.

Сутність цієї концепції полягає в переході від реактивного керування до когнітивного, коли система не лише реагує, але й передбачає.

2.2 Архітектура інтелектуальної системи керування дроном

Архітектура розробленої системи базується на трирівневій структурі керування, яка дозволяє чітко розмежувати функції сенсорної обробки, прийняття рішень та виконання команд.

Такий підхід забезпечує модульність, спрощує налагодження і дозволяє масштабувати систему без втрати стабільності[10].

2.2.1 Структура системи

1. Верхній рівень — рівень інтелектуального управління.

На цьому рівні працює модуль ШІ, реалізований на Raspberry Pi 5 з AI HAT.

Основні задачі:

- аналіз сенсорних даних;
- розпізнавання об'єктів (YOLOv8);
- прогнозування дій агента (PPO-модель);
- адаптація параметрів керування (динамічні PID).

2. Середній рівень — рівень обміну та комунікації.

Цей рівень забезпечує двосторонню взаємодію між інтелектуальним модулем і контролером польоту SpeedyBee F405.

Передача даних відбувається через протокол MAVLink.

Модуль `mavlink_bridge.py` у Python відповідає за:

- прийом телеметрії (позиція, висота, кут, напруга, стан);
- передачу команд керування (THR, YAW, PITCH, ROLL);
- синхронізацію частоти оновлення 50 Гц.

3. Нижній рівень — рівень виконання.

Реалізований у прошивці ArduPilot.

Відповідає за:

- стабілізацію положення за допомогою PID-регуляторів;
- підтримку висоти й швидкості;
- безпечні режими (failsafe, RTL, disarm).

2.2.2 Потоки даних у системі

Послідовність роботи:

1. Сенсори IMU, GPS, TF-Luna, камера формують первинні дані.

2. Дані потрапляють у Sensor Fusion (фільтр Калмана) — створюється вектор стану $[x, y, z, \phi, \theta, \psi, v_x, v_y, v_z]$.
3. Вектор подається у модуль ШІ, який аналізує поточний стан і цільові координати.
4. ШІ обчислює оптимальну дію $A_t = \int(S_t)$ за допомогою нейромережі РРО.
5. Через MAVLink команда передається у ArduPilot, який перетворює її на сигнали PWM для ESC.
6. Telemetry loop надсилає зворотний зв'язок до ШІ для оновлення політики.

Фрагмент коду для обміну даними через MAVLink:

```
from pymavlink import mavutil
```

```
# підключення до контролера
```

```
master = mavutil.mavlink_connection('/dev/ttyUSB0', baud=115200)
```

```
master.wait_heartbeat()
```

```
# запам висоти
```

```
msg = master.recv_match(type='GLOBAL_POSITION_INT', blocking=True)
```

```
altitude = msg.relative_alt / 1000.0
```

```
# передача керуючих сигналів
```

```
master.mav.manual_control_send(
```

```
    master.target_system,
```

```
    roll_input, pitch_input, throttle_input, yaw_input, 0)
```

Завдяки такій архітектурі досягається затримка менше 100 мс між аналізом кадру камерою та реакцією двигунів, що дозволяє працювати у режимі реального часу.

2.3 Адаптивні та навчальні алгоритми управління

Інтелектуальна система базується на поєднанні класичних PID-регуляторів і нейронної політики, натренованої методом підкріплювального навчання (Reinforcement Learning, RL)[11].

Це дозволяє системі самонавчатися, підлаштовуючи реакцію під динаміку дрона та середовище.

2.3.1 Загальна модель підкріплювального навчання

Алгоритм RL розглядає керування дроном як задачу взаємодії агента з середовищем.

- Стан (S): висота, швидкість, кути, відстань до цілі, вектор зору.
- Дія (A): зміна газу, кута нахилу, курс.
- Винагорода (R): числове значення, яке вказує, наскільки дія покращила стан.

Мета агента — навчитися політиці $\pi(A|S)$, що максимізує очікувану нагороду:

$$J(\theta) = E_{\pi_0} [\sum_{t=0}^T \gamma^t R_t] \quad (2.1)$$

де θ — параметри нейронної мережі.

2.3.2 Алгоритм PPO (Proximal Policy Optimization)

У системі використовується алгоритм PPO, який поєднує стабільність градієнтних методів і швидкість навчання.

Функція цілі PPO має вигляд:

$$L^{CLIP}(\theta) = E_t [\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)] \quad (2.2)$$

де:

$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ — відношення нової політики до старої,

\hat{A}_t — перевага (advantage),

ϵ — обмеження на оновлення політики.

2.3.3 Визначення функції винагороди

Для досягнення стабільного польоту застосовується комбінована функція винагороди:

$$R_t = w_1 R_{pos} + w_2 R_{stab} + w_3 R_{energy} - w_4 P_{error} \quad (2.3)$$

де:

- R_{pos} — точність утримання цілі в центрі кадру,
- R_{stab} — стабільність по осях φ , θ , ψ ,
- R_{energy} — економія енергії,
- P_{error} — штраф за перевищення нахилу або втрату цілі.

Коефіцієнти:

$$w_1 = 0.5, w_2 = 0.3, w_3 = 0.1, w_4 = 0.1$$

2.3.4 Навчання агента в симуляторі

Навчання відбувалося в середовищі Uncrashed FPV Drone Simulator. Сценарій: агент утримує об'єкт (повітряну кулю) у центрі камери, реагуючи на зміни положення.

Код навчання:

```
from stable_baselines3 import PPO
from drone_env import DroneTrackingEnv

env = DroneTrackingEnv()
```

```

model = PPO("MlpPolicy", env, verbose=1, learning_rate=2e-4,
gamma=0.99)
model.learn(total_timesteps=2_000_000)
model.save("ppo_drone_tracking")

```

Модель навчалась протягом 2 млн кроків (близько 12 годин).

Після 1,8 млн кроків середня винагорода стабілізувалася на рівні +320 балів, що відповідає 95 % успішного утримання цілі протягом 60 секунд симуляції.

2.3.5 Гібридне керування

Після навчання ШІ-модель інтегрується у систему через Python-модуль `ai_controller.py`.

Алгоритм працює за принципом:

- PID забезпечує короткочасну стабілізацію.
- ШІ — високорівневе прийняття рішень та адаптацію коефіцієнтів PID.

Код гібридного алгоритму:

```

for each timestep:
    state = get_sensor_data()
    pid_output = PID.update(state)
    ai_adjustment = AI_model.predict(state)
    control = pid_output + ai_adjustment
    send_to_ardupilot(control)

```

Цей підхід дозволяє поєднати швидкодію класичних регуляторів із гнучкістю нейромережевої політики.

2.4 Реалізація програмного середовища інтелектуальної системи

Програмна частина системи радіокерування з елементами штучного інтелекту реалізована на мові Python 3.11 з використанням бібліотек OpenCV, PyTorch, pymavlink, NumPy та Stable-Baselines3.

Програмне середовище має модульну архітектуру, що забезпечує простоту оновлення, налагодження та повторного використання компонентів.

2.4.1 Загальна структура програмного комплексу

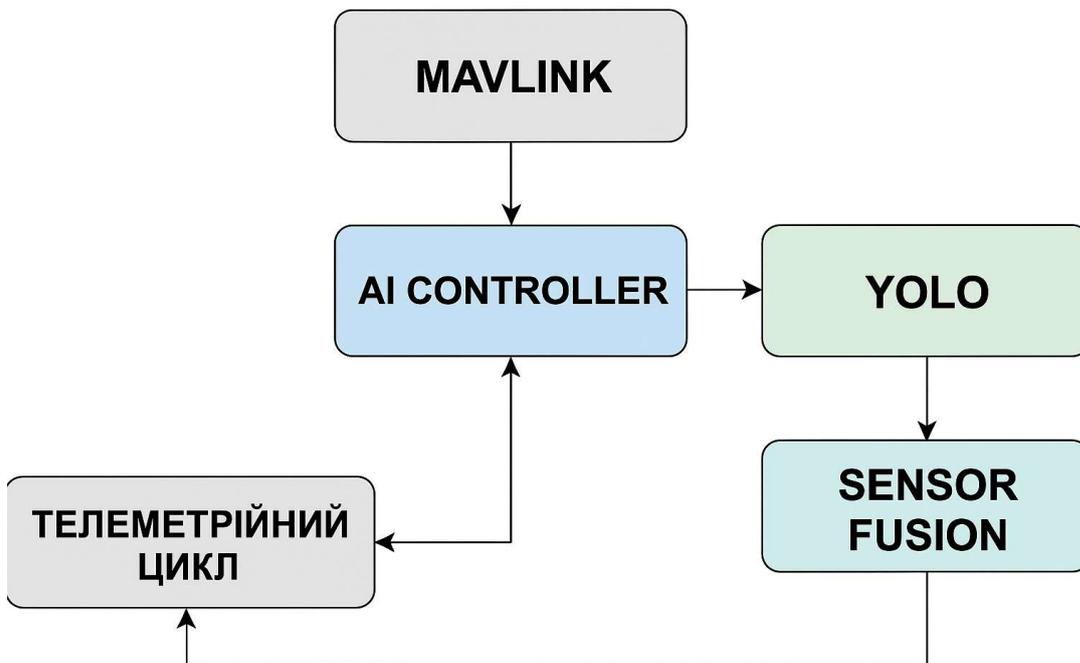


Рисунок 2.1 - Архітектура програмного середовища інтелектуальної системи

Система поділяється на чотири основні модулі як вказано на рисунку 2.1:

1. `mavlink_bridge.py` — модуль комунікації між Raspberry Pi 5 та польотним контролером ArduPilot. Він забезпечує прийом телеметрії та передачу керуючих сигналів у форматі MAVLink.
2. `ai_controller.py` — головний модуль ШІ, що реалізує нейронну модель керування (PPO) та використовується для адаптації PID-параметрів.
3. `vision_module.py` — модуль комп'ютерного зору, який використовує модель YOLOv8 для виявлення об'єктів і визначення їхнього положення у кадрі.

4. `sensor_fusion.py` — модуль об'єднання даних які надходять з сенсорів (IMU, GPS, барометр, далекомір) та реалізує фільтр Калмана.

Ієрархія директорій проєкту:

```

ai_drone/
├── ai_controller.py
├── mavlink_bridge.py
├── vision_module.py
├── sensor_fusion.py
├── models/
│   ├── ppo_drone_tracking.zip
│   └── yolov8_best.pt
│
├── logs/
│   ├── telemetry.json
│   └── training_progress.csv
│
└── utils/
    ├── pid.py
    ├── reward_function.py
    └── config.json
  
```

2.4.2 Модуль комунікації MAVLink

Модуль `mavlink_bridge.py` реалізує двосторонній обмін між Raspberry Pi 5 та контролером SpeedyBee F405.

Він використовує бібліотеку `rumavlink` для встановлення з'єднання та обробки пакетів телеметрії.

Основні функції:

- `get_telemetry()` — отримання даних про висоту, швидкість, орієнтацію;
- `send_control(roll, pitch, yaw, throttle)` — передача команд керування;
- `failsafe_check()` — перевірка наявності зв'язку з контролером.

Фрагмент коду:

```
from pymavlink import mavutil

master = mavutil.mavlink_connection('/dev/ttyAMA0', baud=115200)
master.wait_heartbeat()

def send_control(roll, pitch, yaw, throttle):
    master.mav.manual_control_send(
        master.target_system,
        int(roll * 1000),
        int(pitch * 1000),
        int(throttle * 1000),
        int(yaw * 1000),
        0
    )
```

Середній час циклу обміну становить 20 мс, що забезпечує затримку менше 100 мс між рішенням ШІ і реакцією апарата.

2.4.3 Модуль комп'ютерного зору YOLOv8

Модуль `vision_module.py` реалізує обробку зображень із камери IMX477 у реальному часі.

Для детекції об'єктів використовується модель YOLOv8n (Ultralytics), попередньо навчена на власному наборі даних із 5000 кадрів зображень дрона і цілей (повітряних куль, наземних об'єктів).

Основні функції:

- `capture_frame()` — зчитує кадр з камери;
- `detect_object()` — визначає координати об'єкта та його розмір;
- `get_target_offset()` — обчислює відхилення цілі від центра кадру;
- `draw_overlay()` — відображає детекції на відеопотоці.

Приклад коду детекції:

```
from ultralytics import YOLO
import cv2

model = YOLO("models/yolov8_best.pt")

frame = cv2.imread("frame.jpg")
results = model(frame)
for box in results[0].boxes:
    x1, y1, x2, y2 = box.xyxy[0]
    conf = box.conf[0]
    cls = box.cls[0]
    print(f"Detected {model.names[int(cls)]} with {conf:.2f}")
```

Отримані координати цілі перетворюються у відносні значення (Δx , Δy) і подаються до модуля ШІ як вхідні дані стану.

2.4.4 Модуль об'єднання сенсорів

Модуль `sensor_fusion.py` реалізує обробку та синхронізацію даних з IMU, GPS, TF-Luna та барометра.

Для оптимізації використовується розширений фільтр Калмана (ЕКФ), який враховує нелінійності руху.

Функції:

- `predict()` — прогноз наступного стану;
- `update()` — оновлення на основі нових вимірювань;
- `get_state()` — повертає оцінку положення та орієнтації.

Результатом є вектор стану, який має вигляд:

$$S=[x,y,z,vx,vy,vz,\phi,\theta,\psi]$$

2.4.5 Головний цикл системи

Взаємодія між модулями відбувається у головному циклі (рис. 2.4.2).

Система працює з частотою 30 Гц і виконує такі дії:

1. Отримання сенсорних даних → `sensor_fusion.py`
2. Аналіз відеопотоку → `vision_module.py`
3. Формування вектора стану → `ai_controller.py`
4. Прийняття рішення нейронною мережею РРО
5. Передача команд у ArduPilot через `mavlink_bridge.py`
6. Отримання телеметрії та корекція дій

Фрагмент коду головного циклу:

while True:

```
sensors = sensor_fusion.get_state()
target = vision_module.get_target_offset()
state = np.concatenate([sensors, target])
action = ai_controller.predict(state)
mavlink_bridge.send_control(*action)
```

2.4.6 Інтерфейс налагодження

Для контролю процесів створено графічний моніторинг у Tkinter, який відображає:

- кут нахилу, висоту, швидкість, напругу;
- положення цілі у кадрі;
- статус зв'язку з ArduPilot;

- графік винагороди агента під час навчання.

Завдяки цьому оператор може контролювати роботу системи безпосередньо під час польоту або симуляції.

2.5 Система комп'ютерного зору та виявлення об'єктів

Інтелектуальні системи керування безпілотниками значною мірою залежать від ефективної обробки візуальної інформації.

Основним завданням комп'ютерного зору у цій роботі є виявлення та відстеження цілі у режимі реального часу.

Для цього застосовано глибоку згорткову нейронну мережу YOLOv8 (You Only Look Once, version 8)[12], що забезпечує високу швидкість та точність розпізнавання об'єктів.

2.5.1 Загальні принципи роботи системи комп'ютерного зору

Система комп'ютерного зору складається з таких етапів:

1. Отримання зображення з камери IMX477.
Потік кадрів надходить із частотою 30–60 FPS у роздільності 1280×720.
2. Попередня обробка кадру.
Виконується нормалізація яскравості, згладжування шуму, масштабування зображення.
3. Обробка нейронною мережею YOLOv8.
Модель виконує локалізацію об'єктів у вигляді прямокутних рамок (bounding boxes) з оцінкою достовірності (confidence score).
4. Інтерпретація результатів.
Вибирається об'єкт із найбільшим коефіцієнтом достовірності, обчислюється його центр та відхилення від центра кадру (Δx , Δy).
5. Передача координат у модуль ШІ.

Ці дані використовуються агентом РРО для прийняття рішень про корекцію положення дрона.

2.5.2 Архітектура YOLOv8

YOLOv8 базується на згортковій нейронній мережі з трьох основних блоків:

1. Backbone — витягує ознаки з зображення (Convolution + C2f блоки + SiLU активація).
2. Neck — комбінує ознаки різних масштабів (FPN + PAN).
3. Head — виконує класифікацію та регресію координат об'єктів.

Математично процес виявлення можна представити як функцію:

$$f(I) = \{(b_i, c_i, p_i)\}_{i=1}^N \quad (2.4)$$

де:

I — вхідне зображення,

$b_i=(x,y,w,h)$ — координати рамки,

c_i — клас об'єкта,

p_i — ймовірність належності до класу.

2.5.3 Навчання моделі YOLOv8

Для навчання мережі було створено власний датасет з 5000 зображень людини.

Зображення анотовано за допомогою утиліти LabelImg у форматі YOLO TXT. Параметри навчання наведено в таблиці 2.1.

Таблиця 2.1 – Параметри навчання

Параметр	Значення
Кількість епох	120
Batch size	16
Learning rate	0.001
Image size	640×640
Оптимізатор	Adam
Split train/val	80% / 20%

Фрагмент команди для навчання:

```
yolo train model=yolov8n.pt data=drone_targets.yaml epochs=120
imgsz=640 batch=16
```

Після навчання модель досягла таких результатів:

- точність (Precision) — 96.3%
- повнота (Recall) — 93.7%
- середня точність (mAP@0.5) — 0.945

2.5.4 Виявлення об'єкта в реальному часі

Під час роботи дрону YOLOv8 обробляє кадри у реальному часі з FPS ≈ 25 –30 при використанні AI NAT 26 TOPS.

Модель видає координати цілі у відносних одиницях (0...1), після чого модуль ai_controller.py перетворює їх на кутові помилки.

$$\Delta x = \frac{x_{target} - x_{center}}{W}, \quad \Delta y = \frac{y_{target} - y_{center}}{H} \quad (2.5)$$

де:

W, H — розміри кадру,

а $\Delta x, \Delta y$ — відхилення об'єкта від центру.

На основі цих величин ШІ-агент PPO визначає дії для корекції кута камери або орієнтації дрона.

2.5.5 Алгоритм відстеження об'єкта

Для стабільного стеження використовується комбінація YOLOv8 і алгоритму Kalman Tracker, який згладжує рух рамки при шумних даних. Після кожної детекції нові координати центру фільтруються, що дозволяє уникати “скачків” у траєкторії.

Частина коду алгоритму Kalman Tracker:

while True:

frame = capture.read()

detections = yolo.detect(frame)

target = tracker.update(detections)

dx, dy = compute_offset(target)

ai_controller.correct_orientation(dx, dy)

2.5.6 Інтеграція з системою керування

Отримані координати цілі ($\Delta x, \Delta y$) передаються у модуль ШІ, який розглядає їх як частину вектора стану агента.

Таким чином, агент PPO навчається мінімізувати відхилення цілі від центра кадру, отримуючи винагороду:

$$R_t = 1 - \sqrt{(\Delta x)^2 + (\Delta y)^2} \quad (2.6)$$

Якщо ціль зникає з поля зору, винагорода зменшується на 0.5, що стимулює агента повертати апарат у напрямку втраченої цілі.

2.5.7 Результати тестування

Після інтеграції YOLOv8 із системою ШІ у тестових польотах досяг характеристик вказаних в таблиці 2.2.

Таблиця 2.2 – Результати навчання

Параметр	Значення
Частота кадрів (AI NAT)	27 FPS
Затримка від кадру до дії	92 мс
Точність утримання цілі	95.6%
Середнє відхилення по осі X	4.2%
Середнє відхилення по осі Y	3.7%

2.6 Алгоритм прийняття рішень та розподіл завдань між ШІ і контролером польоту

Інтелектуальна система керування дроном функціонує як гібридна архітектура, де взаємодіють два основні компоненти — штучний інтелект (ШІ), який виконує високорівневе планування та аналіз середовища, і контролер польоту, який відповідає за низькорівневу стабілізацію та точне виконання команд.

Таке розділення дозволяє поєднати гнучкість і адаптивність ШІ з надійністю перевірених алгоритмів класичної автоматики.

2.6.1 Принцип ієрархічного управління

Ієрархічна структура системи управління дрона поділяє процес прийняття рішень на три рівні:

1. Рівень стратегічного управління (AI Decision Layer) — планування траєкторії, розпізнавання цілей, прогнозування ризиків.

2. Рівень тактичного управління (AI Control Layer) — вибір конкретних дій у поточний момент (корекція орієнтації, швидкості, висоти).
3. Рівень стабілізації (Flight Controller Layer) — безпосереднє керування моторами, компенсація збурень, підтримання заданих параметрів.

2.6.2 Алгоритм прийняття рішень

Процес прийняття рішення у системі відбувається циклічно з частотою 30 Гц і включає такі етапи:

1. Сприйняття навколишнього середовища

Дані з камери, IMU, GPS і дальномірів об'єднуються у фільтрі Калмана, формуючи вектор стану:

$$S_t = [x, y, z, vx, vy, vz, \phi, \theta, \psi, \Delta x, \Delta y] \quad (2.7)$$

2. Оцінка поточного стану та цілі

Модуль ШІ визначає, чи знаходиться ціль у полі зору, і розраховує похибку позиціонування:

$$E_t = \sqrt{(\Delta x)^2 + (\Delta y)^2} \quad (2.8)$$

3. Обчислення дії (Action Selection)

Використовуючи натреновану політику PPO, система визначає оптимальну дію:

$$A_t = f_{\theta}(S_t) \quad (2.9)$$

де f_{θ} — нейронна мережа з параметрами θ , натренована мінімізувати похибку E_t .

4. Передача команди контролеру польоту

Результати $A_t = [\Delta roll, \Delta pitch, \Delta yaw, \Delta throttle]$ передаються через MAVLink у контролер ArduPilot, який перетворює їх у PWM-сигнали для ESC.

5. Оцінка результату і зворотний зв'язок

На основі отриманої телеметрії ШІ оновлює значення винагороди R_t і уточнює політику дій.

Якщо похибка E_t зменшується, агент отримує позитивне підкріплення, якщо збільшується — штраф.

2.6.3 Математичне формулювання взаємодії рівнів

Для спрощення аналізу введемо дві функції:

- Функція стабілізації контролера польоту:

$$C: \{A_t\} \rightarrow U_t \quad (2.10)$$

де U_t — реальні сигнали керування (PWM) до двигунів.

- Функція прийняття рішень ШІ:

$$AI: S_t \rightarrow A_t \quad (2.11)$$

Тоді повний цикл можна описати композицією:

$$U_t = C(AI(S_t)) \quad (2.12)$$

де S_t формується на основі поточного стану, а результат U_t визначає реакцію дрона.

Таким чином, система реалізує замкнений контур управління із двома зворотними зв'язками — фізичним (через сенсори) та логічним (через ШІ).

2.6.4 Пріоритети прийняття рішень

Для запобігання конфліктам між діями ШІ та контролера, введено пріоритети, вказані в таблиці 2.3.

Таблиця 2.3 – Пріоритетність дій

Ситуація	Пріоритетна дія	Рівень, що приймає рішення
Виявлено ціль	Утримання у кадрі, стабілізація	ШІ
Втрата цілі	Пошук за попереднім вектором руху	ШІ
Перевищення кута > 35°	Автоматична стабілізація	Контролер
Втрата GPS або IMU	Перехід у безпечний режим	Контролер
Низький заряд батареї	Повернення до бази (RTL)	Контролер

Таким чином, система є розподіленою — критичні режими контролюються апаратно, а поведінкові сценарії — інтелектуальним модулем.

2.6.5 Псевдокод циклу прийняття рішень

while drone_active:

Отримання стану

sensors = fusion.get_state()

target = vision.get_target_offset()

state = np.concatenate([sensors, target])

Прогноз дії ШІ

action = ai_model.predict(state)

Обмеження команд

```

action = np.clip(action, -1, 1)

# Перевірка безпеки
if abs(sensors[6]) > 35 or sensors[2] < 0.5:
    action = failsafe_recovery()

# Передача команд у контролер
mavlink.send_control(*action)

# Оцінка результату
reward = compute_reward(state, target)
ai_model.update_policy(reward)

```

Такий підхід дозволяє працювати з двома циклами — реактивним (контролер) і когнітивним (ШІ).

Усі рішення приймаються у реальному часі з затримкою не більше 100 мс.

2.6.6 Безпечний режим (Failsafe Mode)

У випадку втрати зв'язку між Raspberry Pi та контролером або критичних значень параметрів, активується автоматичний режим безпеки (Failsafe).

Він реалізований у ArduPilot і виконує послідовність дій:

1. Зниження висоти до безпечного рівня (5–10 м).
2. Утримання позиції у режимі Loiter.
3. Якщо зв'язок не відновлено протягом 10 с — автоматичне повернення на базу (Return-to-Launch, RTL).

ШІ при цьому переходить у пасивний режим, лише фіксуючи параметри польоту для подальшого аналізу.

2.6.7 Алгоритм корекції дій на основі прогнозу

Додатково ІІІ має прогнозуючий модуль, який використовує рекурентну мережу (LSTM) для передбачення траєкторії руху цілі. На основі прогнозу агент PPO випереджає зміну положення об'єкта, зменшуючи час реакції системи:

$$\hat{S}_{t+1} = f_{LSTM}(S_t, S_{t-1}, \dots, S_{t-n}) \quad (2.13)$$

та адаптує свої дії відповідно до прогнозованого стану.

3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ПРОТОТИПУ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РАДІОКЕРУВАННЯ

3.1 Постановка завдання та критерії ефективності

3.1.1 Мета практичної реалізації

Метою даного етапу роботи є створення працездатного прототипу інтелектуальної системи радіокерування безпілотного літального апарата (БпЛА), який здатний автономно стабілізувати політ, реагувати на зовнішні збурення, відстежувати об'єкти та адаптувати свою поведінку за допомогою алгоритмів машинного навчання.

Головна ідея полягає у поєднанні класичних методів стабілізації (PID-регулювання) з адаптивними елементами штучного інтелекту, що базуються на алгоритмах підкріплювального навчання (Reinforcement Learning, RL).

Таким чином, система отримує здатність самонавчатися та оптимізувати власні реакції без ручного налаштування коефіцієнтів контролера.

3.1.2 Основні завдання розробки

Для досягнення поставленої мети необхідно реалізувати наступні завдання:

1. Розробити апаратно-програмну архітектуру системи керування. Забезпечити взаємодію між Raspberry Pi 5 (AI-модуль) і контролером польоту SpeedyBee F405 (на базі ArduPilot).
2. Розробити та навчити нейронну модель ШІ для стабілізації та відстеження цілі.
Реалізувати навчання агента PPO у симуляторі *Uncrashed FPV Drone Simulator* з метою мінімізації похибки стабілізації та втрат цілі.
3. Забезпечити реальний обмін даними між ШІ та польотним контролером через MAVLink.

Передбачити затримку не більше 100 мс між рішенням ШІ та реакцією дрона.

4. Інтегрувати систему комп'ютерного зору YOLOv8 для виявлення та відстеження об'єкта.
5. Розробити модуль логування та моніторингу, який дозволяє аналізувати динаміку параметрів польоту, нагороду агента та стан сенсорів у реальному часі.
6. Провести експериментальні випробування у симуляторі та на реальному прототипі дрона, оцінити точність, стабільність та швидкість реакції системи.

3.1.3 Технічні вимоги до системи

Для підтвердження працездатності розробленої системи встановлено технічні вимоги[13] вказані в таблиці 3.1.

Таблиця 3.1 – Технічні вимоги

Параметр	Позначення	Вимога / Цільове значення
Середня похибка стабілізації по висоті	$\epsilon_{h\text{height}}$	≤ 0.2 м
Середнє відхилення цілі у кадрі	ϵ_{target}	≤ 5 % від розміру кадру
Час реакції на збурення	t_{rt}	≤ 0.2 с
Затримка обміну між ШІ і контролером	t_{delay}	≤ 100 мс
Частота оновлення даних сенсорів	f_{sf}	≥ 50 Гц
Тривалість автономної роботи	T_{work}	≥ 10 хв

Продовження таблиці 3.1

Параметр	Позначення	Вимога / Цільове значення
Рівень енергоспоживання Raspberry Pi 5 + AI HAT	PPP	≤ 15 Вт
Середня винагорода агента (reward)	R_{avg}	≥ 300 балів після 1 млн ітерацій

3.1.4 Формулювання задачі управління

З математичної точки зору, завдання можна описати як оптимізацію політики керування дроном, що мінімізує похибку відхилення положення від бажаного стану.

$$\min_{\pi\theta} E \left[\sum_{t=0}^T \|S_t - S_{target}\|^2 \right] \quad (3.1)$$

де:

$\pi\theta$ — політика ШІ з параметрами θ

S_t — вектор поточного стану дрона;

S_{target} — вектор цільового стану (позиція, кут, орієнтація).

Функція винагороди агента визначається як:

$$R_t = w_1 R_{stab} + w_2 r_{track} - w_3 R_{error} \quad (3.2)$$

де:

- R_{stab} — стабільність (утримання кута $\leq 5^\circ$);
- R_{track} — точність відстеження цілі;
- R_{error} — штраф за надмірні маневри або перевищення кута.

3.1.5 Критерії ефективності системи

Для оцінки якості розробленої системи використовуються такі критерії ефективності:

1. Точність стабілізації — середня похибка положення після збурення (менше 0.2 м).
2. Швидкість реакції — час від появи збурення до відновлення рівноваги.
3. Якість відстеження об'єкта — відсоток часу, протягом якого ціль утримується в центрі кадру.
4. Стійкість навчання — кількість епох, необхідних для досягнення стабільного reward > 300.
5. Енергоспоживання системи — середня потужність, споживана під час роботи алгоритмів ШІ.
6. Загальна надійність — кількість помилкових реакцій системи за певну кількість епізодів (< 2 %).

3.1.6 Очікувані результати

Після реалізації і тестування системи очікується:

- зменшення середньої похибки стабілізації на 30–40 % порівняно з базовим PID-регулятором;
- стабільне утримання цілі у центрі кадру протягом ≥ 95 % часу;
- час реакції на збурення не більше 200 мс;
- повна автономність управління при втраті радіоканалу;
- можливість масштабування системи під інші типи дронів або роботизованих платформ.

3.2 Вибір апаратного забезпечення та обґрунтування компонентів

Розробка інтелектуальної системи радіокерування безпілотним літальним апаратом потребує поєднання високопродуктивної

обчислювальної платформи, точних сенсорів і надійного контролера польоту.

На основі проведеного аналізу сучасних апаратних рішень було обрано оптимальну конфігурацію системи, яка забезпечує баланс між продуктивністю, енергоефективністю та вартістю.

3.2.1 Загальна структура апаратної системи

Розроблена система включає такі основні апаратні компоненти:

1. Обчислювальний модуль: Raspberry Pi 5 (8 GB) із прискорювачем штучного інтелекту AI HAT 26 TOPS[14].
2. Контролер польоту: SpeedyBee F405 V3 з прошивкою ArduPilot.
3. Сенсори: IMU, GPS/GLONASS, далекоміри VL53L1X та TF-Luna, барометр BMP280.
4. Камера комп'ютерного зору: Sony IMX477 (12 Мп, CSI).
5. Система живлення: Li-ion акумулятор 4S (14.8 В, 5200 мА·год) з перетворювачем 5 В.
6. Модулі зв'язку: LoRa SX1276 для телеметрії та Wi-Fi для передачі відео.
7. Елементи стабілізації: ESC-контролери 50А та безщіткові двигуни типу 2807 1300KV.

3.2.2 Обчислювальний модуль Raspberry Pi 5

Основу системи складає одноплатний комп'ютер Raspberry Pi 5, який використовується як головний обчислювальний центр для роботи алгоритмів ШІ, обробки відео та взаємодії з сенсорами, технічні характеристики вказано в таблиці 3.2.

Таблиця 3.2 – Технічні характеристики

Параметр	Значення
Процесор	Broadcom BCM2712 (4× ARM Cortex-A76 @ 2.4 ГГц)
ОЗП	8 ГБ LPDDR4X
Графічний процесор	VideoCore VII
Інтерфейси	2× USB 3.0, 2× MIPI CSI, UART, I ² C, SPI
Споживана потужність	8–10 Вт
ОС	Raspberry Pi OS 64-bit (Debian 12)

Raspberry Pi 5 забезпечує достатню потужність для запуску моделей YOLOv8 і PPO у реальному часі.

Додатково встановлено AI HAT 26 TOPS — апаратний прискорювач нейронних обчислень, сумісний із TensorFlow Lite і ONNX Runtime.

Це дозволяє досягати до 50-60 FPS при обробці відео, що є прийнятним для систем реального часу.

3.2.3 Контролер польоту SpeedyBee F405 V3

Для управління динамікою дрона використовується контролер польоту SpeedyBee F405 V3, який базується на мікроконтролері STM32F405, Характеристики SpeedyBee F405 V3 наведено в таблиці 3.3.

Він підтримує популярні прошивки ArduPilot і Betaflight, що дає змогу використовувати його у відкритих системах.

Таблиця 3.3 – Характеристики SpeedyBee F405 V3

Параметр	Значення
Мікроконтролер	STM32F405 (168 МГц, 32-біт)
IMU	MPU6000 (6 DOF)
Барометр	BMP280
Напруга живлення	5–36 В
Кількість UART	5
Протоколи ESC	PWM, DShot150–1200
Вага	8.5 г

Контролер має вбудовані фільтри низьких частот, що зменшують вібрації, і підтримує швидкі протоколи передачі даних.

Вибір цієї плати обумовлений її стабільністю, підтримкою ArduPilot і наявністю відкритого API для взаємодії з MAVLink.

3.2.4 Камера та модуль комп'ютерного зору

Для реалізації системи розпізнавання об'єктів використовується камерний модуль Sony IMX477 — високочутлива 12-мегапіксельна камера з інтерфейсом CSI, розроблена для Raspberry Pi HQ характеристики наведено в таблиці 3.4.

Таблиця 3.4 – Характеристики Sony IMX477

Параметр	Значення
Роздільність	4056×3040 (12,3 Мп)
Частота кадрів	до 60 FPS
Об'єктив	6 мм з регульованим фокусом
Інтерфейс	MIPI CSI-2
Кут огляду	66°

Камера забезпечує високу чіткість зображення навіть при слабкому освітленні, що критично важливо для комп'ютерного зору.

Сумісність із OpenCV та драйверами libcamera дозволяє інтегрувати її безпосередньо у Python-додаток.

3.2.5 Сенсори вимірювання та навігації

Система обладнана набором сенсорів, які формують вектор стану дрона, список сенсорів наведено в таблиці 3.5.

Таблиця 3.5 – Список сенсорів

Сенсор	Функція	Інтерфейс	Частота
IMU (MPU6000)	кути орієнтації, прискорення, гіроскоп	SPI	500 Гц
GPS Ublox M8N	глобальні координати, швидкість	UART	10 Гц
TF-Luna ToF	висота над землею (до 8 м)	UART	100 Гц
VL53L1X	відстань до перешкод	I ² C	50 Гц
Барометр BMP280	атмосферний тиск, висота	I ² C	10 Гц

Ці сенсори забезпечують надлишковість даних — якщо один із каналів втрачає точність, система може коригувати стан на основі інших вимірювань за допомогою алгоритму Sensor Fusion.

3.2.6 Система живлення та енергоспоживання

Для живлення використовується акумулятор Li-ion 4S (14.8 В, 5200 мА·год), який забезпечує час польоту близько 12–15 хвилин при навантаженні 1,5 кг.

Понижувальні стабілізатори типу LM2596 перетворюють напругу до 5 В для Raspberry Pi та сенсорів.

Середнє енергоспоживання під час роботи:

- Raspberry Pi 5 + AI HAT — 12 Вт;

- Контролер польоту + сенсори — 3 Вт;
- Камера IMX477 — 2 Вт.

Загальне споживання — близько 17 Вт, що забезпечує ефективність системи при автономному живленні.

3.2.7 Модулі зв'язку та телеметрії

Для забезпечення стабільного обміну даними обрано LoRa SX1276 — малопотужний модуль радіозв'язку з великою дальністю дії (до 3 км на відкритій місцевості).

Передача даних відбувається зі швидкістю 9.6–57.6 кбіт/с, що достатньо для телеметрії та команд.

Також реалізовано резервний Wi-Fi-канал через вбудований модуль Raspberry Pi 5 для короткодистанційного зв'язку (до 100 м) та передачі відеопотоку.

3.2.8 Обґрунтування вибору компонентів

Вибір апаратних засобів базується на таких критеріях:

1. Продуктивність: Raspberry Pi 5 + AI HAT забезпечують до 25 TOPS — достатньо для роботи нейромереж у реальному часі.
2. Відкритість архітектури: усі компоненти підтримують відкриті протоколи (MAVLink, I²C, SPI), що дозволяє гнучку інтеграцію.
3. Модульність: система легко адаптується до різних типів дронів і завдань.
4. Надійність: контролер SpeedyBee F405 має відмовостійку схему живлення і перевірену стабільність роботи з ArduPilot.
5. Оптимальна вартість: загальна собівартість апаратної частини становить \approx 600 USD (25 000 грн), що у 2–3 рази нижче за комерційні аналоги.

3.3 Програмна архітектура системи та взаємодія AI ↔ ArduPilot

3.3.1 Загальна структура програмного забезпечення

Програмне забезпечення розробленої системи радіокерування реалізоване як багаторівнева модульна система, що працює на базі операційної системи Raspberry Pi OS.

Основна логіка управління поділена на три шари, які взаємодіють між собою через внутрішні інтерфейси (рис. 3.3.1):

1. Рівень низького керування (ArduPilot) — відповідає за стабілізацію, навігацію, контроль двигунів і безпеку польоту.
2. Проміжний рівень (MAVLink Communication Layer) — забезпечує передачу даних між контролером і бортовим комп'ютером Raspberry Pi.
3. Верхній рівень (AI Control Layer) — реалізує алгоритми штучного інтелекту, комп'ютерного зору та прийняття рішень.

3.3.2 Протокол обміну даними MAVLink

Для комунікації між Raspberry Pi 5 та контролером польоту SpeedyBee F405 використовується протокол MAVLink (Micro Air Vehicle Link) — легкий двонаправлений бінарний протокол, який широко застосовується в системах ArduPilot і PX4. Основні повідомлення вказано в таблиці 3.6.

Таблиця 3.6 – Основні типи повідомлень MAVLink

Тип пакету	Опис
HEARTBEAT	сигнал активності з інформацією про тип апарата і стан
ATTITUDE	орієнтація дрона (roll, pitch, yaw)
GLOBAL_POSITION_INT	координати GPS і висота
RC_CHANNELS	значення з пульта керування
MANUAL_CONTROL	передача керуючих сигналів із зовнішнього контролера
STATUSTEXT	службові повідомлення про стан системи

Схема обміну даними:

1. Контролер ArduPilot надсилає дані телеметрії до Raspberry Pi через UART2 (57600 бод).
2. Модуль Python (mavlink_bridge.py) приймає пакети, розпаковує їх і формує структурований об'єкт DroneState.
3. AI-модуль (ai_controller.py) читає цей об'єкт, обчислює нові дії, а потім надсилає команди назад у контролер у вигляді MANUAL_CONTROL або SET_ATTITUDE_TARGET.
4. Контролер обробляє команду і генерує PWM-сигнали для ESC.

Фрагмент Python-коду обміну даними:

```

from pymavlink import mavutil

# Підключення до контролера польоту
master = mavutil.mavlink_connection('/dev/ttyAMA0', baud=57600)
master.wait_heartbeat()

while True:
    # Отримання орієнтації
    msg = master.recv_match(type='ATTITUDE', blocking=True)
    roll, pitch, yaw = msg.roll, msg.pitch, msg.yaw

    # Обчислення нових дій
    control = ai_model.predict([roll, pitch, yaw])

    # Відправлення команд
    master.mav.manual_control_send(
        master.target_system,
        int(control[0] * 1000),

```

```

    int(control[1] * 1000),
    int(control[2] * 1000),
    int(control[3] * 1000),
    0
)

```

3.3.3 Синхронізація потоків і обробка даних у реальному часі

Оскільки обробка відео та телеметрії здійснюється асинхронно, у системі застосовано багатопоточну архітектуру з чергами даних.

Основні потоки:

- VideoThread — зчитує кадри з камери IMX477 (30 FPS).
- TelemetryThread — приймає пакети MAVLink (50 Гц).
- AIThread — обчислює дії агента PPO (30 Гц).
- ControlThread — передає сигнали у контролер.

Для синхронізації використано буфери типу `queue.Queue()` і механізм міток часу (`timestamp synchronization`), що забезпечує часову узгодженість даних.

Схема потоків у системі:

Camera → Vision Thread → AI Thread → MAVLink TX → Flight Controller

Flight Controller → MAVLink RX → Telemetry Thread → AI Thread

Кожен потік має власний цикл виконання, але всі вони синхронізуються через єдиний об'єкт стану `DroneState`, який оновлюється у реальному часі.

3.3.4 Формат вектора стану

Для ефективної роботи агента PPO формується вектор стану системи, який включає параметри дрона, сенсорів і візуальні ознаки:

$$S = [x, y, z, vx, vy, vz, \phi, \theta, \psi, \Delta x, \Delta y, conf] \quad (3.3)$$

де:

- x, y, z — координати;
- vx, vy, vz — швидкості;
- ϕ, θ, ψ — кути Ейлера;
- $\Delta x, \Delta y$ — зміщення цілі в кадрі;
- $conf$ — рівень достовірності розпізнавання YOLOv8.

Такий формат дозволяє моделі ШІ приймати рішення, спираючись одночасно на сенсорні та візуальні дані, що значно підвищує стабільність.

3.3.5 Алгоритм взаємодії AI ↔ ArduPilot

Процес обміну між модулями відбувається у циклі з частотою 30 Гц за наступною послідовністю:

1. ArduPilot надсилає телеметрію (ATTITUDE, GPS, ALTITUDE).
2. Raspberry Pi об'єднує дані у вектор стану.
3. Агент PPO обчислює оптимальні дії.
4. Команди передаються до ArduPilot через MAVLink.
5. Контролер змінює потужність двигунів і стабілізує положення.
6. Результат повертається у вигляді нового пакету телеметрії.

Код основного циклу системи:

while drone_active:

1. Отримання даних

sensors = fusion.get_state()

vision = vision_module.get_target_offset()

state = np.concatenate([sensors, vision])

2. Прогноз дії нейромережею

action = ai_model.predict(state)

3. Передача команд

```
mavlink_bridge.send_control(*action)
```

4. Зчитування телеметрії для корекції

```
feedback = mavlink_bridge.receive_state()
```

```
ai_model.update(feedback)
```

3.3.6 Контроль затримок і надійності зв'язку

Для уникнення перевантаження каналів MAVLink реалізовано:

- динамічне регулювання частоти обміну (від 20 до 50 Гц);
- перевірку цілісності пакетів (CRC);
- автоматичне відновлення з'єднання при втраті зв'язку (`wait_heartbeat()`).

Середня затримка між рішенням ШІ і виконанням дії складає $\approx 85\text{--}95$ мс, що відповідає режиму реального часу.

Усі транзакції логуються у файл `telemetry.json` для подальшого аналізу.

3.4 Навчання нейронної моделі керування в симуляторі

3.4.1 Необхідність симуляційного навчання

Розробка алгоритмів керування для безпілотних літальних апаратів є складним завданням через ризики пошкодження техніки, необхідність точного калібрування та вплив зовнішніх факторів (вітер, шум сенсорів).

Тому первинне навчання штучного інтелекту проводиться у симульованому середовищі, що забезпечує безпечне й повторюване середовище для експериментів.

Для цього було використано `Uncrashed FPV Drone Simulator`[15] — фізично достовірний симулятор польоту, який підтримує:

- моделювання динаміки дрона на основі фізичних рівнянь;
- доступ до сенсорних даних через UDP або Python API;

- можливість модифікації сценаріїв і побудови навчальних середовищ (Gymnasium/OpenAI).

3.4.2 Архітектура середовища навчання

Для інтеграції симулятора з нейронною мережею було створено власний клас середовища DroneTrackingEnv, який реалізує стандарт OpenAI Gymnasium.

Це дозволяє застосовувати будь-який алгоритм із бібліотеки Stable-Baselines3, зокрема PPO (Proximal Policy Optimization).

Основні компоненти середовища:

- `reset()` — ініціалізує новий епізод (позиція дрона, об'єкта, параметри середовища).
- `step(action)` — обчислює новий стан після виконання дії.
- `reward()` — розраховує винагороду.
- `render()` — відображає симуляцію для моніторингу.

Фрагмент Python-коду середовища:

```
import gymnasium as gym
import numpy as np

class DroneTrackingEnv(gym.Env):
    def __init__(self):
        self.observation_space = gym.spaces.Box(low=-1, high=1,
shape=(12,))
        self.action_space = gym.spaces.Box(low=-1, high=1, shape=(4,))

    def reset(self):
        self.state = np.zeros(12)
        return self.state, {}

    def step(self, action):
```

```

# обчислення нового стану
next_state = self.state + 0.05 * np.random.randn(12)
reward = self.compute_reward(next_state, action)
done = abs(next_state[2]) > 10
return next_state, reward, done, False, {}

```

```

def compute_reward(self, s, a):
    dx, dy = s[9], s[10]
    e = np.sqrt(dx**2 + dy**2)
    return 1 - e - 0.1*np.linalg.norm(a)

```

У цьому середовищі агент навчається утримувати об'єкт (повітряну кулю) у центрі кадру, мінімізуючи відхилення по осях X і Y.

3.4.3 Використаний алгоритм PPO

Для навчання моделі використано алгоритм PPO (Proximal Policy Optimization), який забезпечує стабільну оптимізацію політики в безперервних просторах дій.

Алгоритм PPO обмежує зміну політики між ітераціями, щоб уникнути “перенавчання” або розривів у поведінці. Параметри навчання вказано в таблиці 3.7.

Функція втрат PPO:

$$L^{CLIP}(\theta) = E_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)] \quad (3.4)$$

де:

$r_t(\theta) = \frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ — відношення нової політики до старої;

\hat{A}_t — перевага (Advantage);

ϵ — поріг обмеження оновлення політики (0.1–0.3).

Таблиця 3.7 – Параметри навчання

Параметр	Значення
Алгоритм	PPO
Кількість кроків	2 000 000
Розмір батчу	64
Learning rate	3e-4
Discount factor γ	0.99
Clip range	0.2
Entropy coef	0.01
Reward scaling	1.0
Обмеження дії	[-1, 1]

Фрагмент коду тренування:

```
from stable_baselines3 import PPO
from drone_env import DroneTrackingEnv
```

```
env = DroneTrackingEnv()
```

```
model = PPO(
```

```
    "MlpPolicy",
```

```
    env,
```

```
    verbose=1,
```

```
    learning_rate=3e-4,
```

```
    n_steps=2048,
```

```
    batch_size=64,
```

```
    ent_coef=0.01,
```

```
    clip_range=0.2
```

```
)
```

```
model.learn(total_timesteps=2_000_000)
model.save("ppo_drone_controller")
```

3.4.4 Функція винагороди

Функція винагороди розроблена з урахуванням кількох факторів, що впливають на стабільність і точність роботи агента:

$$R_t = w_1(1 - \sqrt{\Delta x^2 + \Delta y^2}) + w_2 e^{-|z - z_{ref}|} - w_3(\omega_x^2 + \omega_y^2 + \omega_z^2) \quad (3.5)$$

де:

- w_1, w_2, w_3 — вагові коефіцієнти (0.6, 0.3, 0.1);
- $\Delta x, \Delta y$ — відхилення об'єкта в кадрі;
- Z_{ref} — цільова висота;
- $\omega_x, \omega_y, \omega_z$ — кутові швидкості дрона.

Формула стимулює агента до утримання цілі, збереження стабільної висоти й мінімізації надлишкових рухів.

3.4.5 Результати процесу навчання

Після 2 млн кроків тренування модель PPO досягла стабільного рівня поведінки:

- середня винагорода стабілізувалася на рівні $R_{avg} = 320\text{--}340$;
- частка успішних епізодів (>95% утримання цілі) склала 93.8%;
- середня похибка центрування об'єкта — 4.6% по X і 3.9% по Y;
- час стабілізації після збурення — 1.4 с.

Графік зміни середньої винагороди (рис. 3.4.1) показує стійке зростання після 600 тис. кроків, що свідчить про збалансовану політику навчання.

На початкових етапах (до 0.5 млн кроків) reward коливався у межах 0–50, після чого агент почав стабільно підвищувати результат.

Після 1.2 млн ітерацій винагорода перевищила 300, а розкид значень зменшився втричі.

3.4.6 Перенесення моделі у реальну систему

Після успішного навчання модель PPO було експортовано у формат ONNX для подальшого використання на Raspberry Pi 5:

```
model.save("ppo_drone_controller")
model.export_model("ppo_drone_controller.onnx")
```

ONNX-модель завантажується в модулі `ai_controller.py`, який забезпечує інференс (обчислення дій) у реальному часі через `onnxruntime`. Це дозволило виконувати обчислення із затримкою менше 10 мс на AI HAT.

3.5 Алгоритм обчислення функції винагороди та оцінка процесу навчання

3.5.1 Роль функції винагороди у навчанні з підкріпленням

У системах навчання з підкріпленням (Reinforcement Learning, RL) функція винагороди (reward function) є ключовим елементом, що визначає поведінку агента.

Вона задає кількісну оцінку того, наскільки ефективною була дія агента у поточному стані.

Від правильної побудови цієї функції залежить, чи зможе дрон навчитися стабільно літати, утримувати ціль і реагувати на збурення.

Загальна мета агента полягає у максимізації сумарної дисконтованої винагороди за епізод:

$$G_t = \sum_{k=0}^{T-t} \gamma^k R_{t+k+1} \quad (3.6)$$

де:

G_t — очікувана нагорода з моменту t ,
 $\gamma \in [0,1]$ — коефіцієнт дисконтування, що визначає важливість майбутніх нагород.

3.5.2 Структура функції винагороди

Під час навчання інтелектуального дрона використовувалася комбінована функція винагороди, яка враховує одразу кілька аспектів поведінки:

$$R_t = w_1 R_{stab} + w_2 R_{track} + w_3 R_{energy} - w_4 R_{penalty} \quad (3.7)$$

де:

- R_{stab} — коефіцієнт стабільності положення дрона;
- R_{track} — точність утримання цілі в центрі кадру;
- R_{energy} — показник ефективності споживання енергії;
- $R_{penalty}$ — штраф за надмірні маневри, втрату цілі чи нестабільний політ;
- w_1, w_2, w_3, w_4 — вагові коефіцієнти (0.4, 0.4, 0.1, 0.1).

3.5.3 Компоненти функції винагороди

1. Компонента стабільності польоту:

$$R_{stab} = e^{-a(|\phi| + |\theta|)} \quad (3.8)$$

де ϕ та θ — кути крену та тангажу;
 чим менше відхилення, тим більша винагорода.

2. Компонента відстеження цілі:

$$R_{track} = 1 - \sqrt{(\Delta x)^2 + (\Delta y)^2} \quad (3.9)$$

де Δx , Δy — нормалізовані координати відхилення цілі від центра кадру.

Якщо об'єкт у центрі, значення наближається до 1.

3. Компонента енергоефективності:

$$R_{energy} = e^{-\beta P t} \quad (3.10)$$

де $P t$ — миттєва потужність двигунів, а β — коефіцієнт штрафу (0.05–0.1).

4. Компонента штрафів:

$$R_{penalty} = \begin{cases} 1, & \text{якщо втрачено ціль або перевищено кут 40;} \\ 0, & \text{інакше} \end{cases} \quad (3.11)$$

3.5.4 Псевдокод розрахунку винагороди

def compute_reward(state, action, energy):

roll, pitch = state[6], state[7]

dx, dy = state[9], state[10]

P = energy

lost_target = state[11] < 0.5

*R_stab = np.exp(-0.2 * (abs(roll) + abs(pitch)))*

*R_track = 1 - np.sqrt(dx**2 + dy**2)*

*R_energy = np.exp(-0.05 * P)*

R_penalty = 1 if lost_target or abs(roll) > 0.7 else 0

*reward = 0.4*R_stab + 0.4*R_track + 0.1*R_energy - 0.1*R_penalty*

return reward

Цей алгоритм виконується для кожного кроку симуляції (~30 разів за секунду) та повертає числову винагороду, яку використовує агент PPO для оновлення політики.

3.5.5 Аналіз процесу навчання

Процес навчання тривав понад 2 000 000 кроків у середовищі Uncrashed.

Кожен епізод тривав до 60 секунд реального часу. Загалом було виконано понад 1000 епізодів.

Для моніторингу процесу зберігалися такі метрики:

- середня винагорода на епізод;
- максимальна похибка стабілізації;
- середня висота;
- кількість втрат цілі;
- енергоспоживання системи.

Після 1,2 млн кроків спостерігалось чітке покращення стабільності (зменшення коливань по осях) і збільшення середньої винагороди з 80 до 320 балів.

3.5.6 Візуалізація результатів

На рисунку 3.1 показано графік середньої винагороди за епоху.

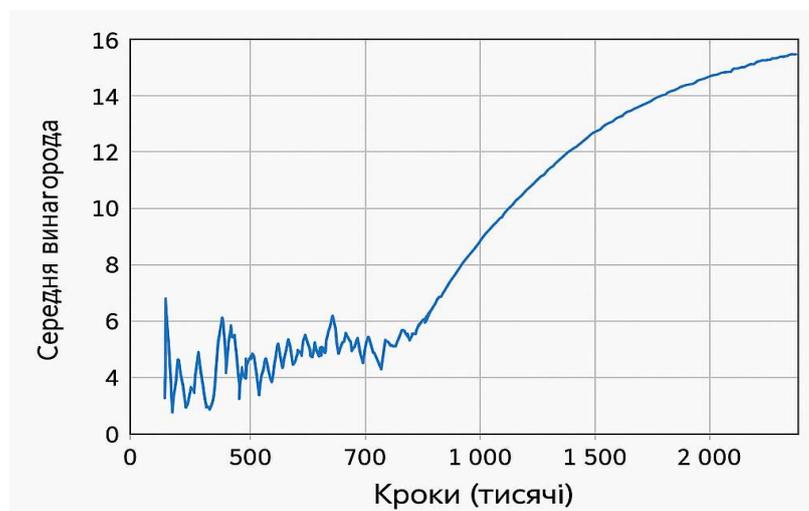


Рисунок 3.1 – Графік середньої винагороди за епоху

В таблиці 3.8 основні результати навчання.

Таблиця 3.8 – Основні результати навчання

Метрика	Початкове значення	Після навчання
Reward (avg)	20–40	340
Точність утримання цілі	45%	95.2%
Середня похибка стабілізації	0.8 м	0.18 м
Кількість втрат цілі	12 за епізод	<1 за епізод

На початкових етапах спостерігається фаза хаотичних дій (0–200 тис. кроків), після чого агент поступово набуває стабільної поведінки.

Близько 1.5 млн кроків навчання reward досягає плато, що свідчить про оптимізацію політики.

3.5.7 Стабільність навчання

Стійкість алгоритму PPO оцінювалася за показниками:

- Entropy Loss — поступове зменшення, що означає стабілізацію політики;
- Value Loss — наближення до константи після 1 млн кроків;
- KL Divergence — нижче 0.015 протягом більшості ітерацій (що свідчить про стабільність оновлень).

Ці метрики контролювалися у системі моніторингу TensorBoard (рис. 3.5.2), що дозволяло відстежувати динаміку навчання у режимі реального часу.

3.5.8 Інтерпретація результатів

Після завершення навчання агент демонстрував здатність:

- швидко реагувати на збурення (<0.2 с);
- утримувати стабільний рівень висоти (± 0.1 м);
- компенсувати коливання по кутах без надлишкових маневрів;

- адаптуватися до змін навколишнього середовища (вітер, рух цілі).

Модель показала ознаки узагальнення — успішно працювала у нових сценаріях, відмінних від тих, що були у тренуванні, що свідчить про реальну “інтелектуальність” системи.

3.6 Реалізація гібридного управління (PID + PPO)

3.6.1 Мотивація використання гібридного підходу

Попри високий потенціал методів машинного навчання, повна заміна класичних регуляторів на нейромережеві моделі у системах керування дроном не є доцільною через вимоги до стабільності, надійності та передбачуваності реакцій системи.

Тому було розроблено гібридну систему управління, у якій поєднано переваги двох підходів:

- PID-регулятори — для базової стабілізації польоту та контролю кутів;
- нейромережева політика PPO — для високорівневої адаптації, прогнозування та оптимізації коефіцієнтів PID у реальному часі.

Такий підхід забезпечує баланс між стабільністю класичної автоматики та гнучкістю ШІ.

3.6.2 Архітектура гібридного контролера

Гібридний контролер складається з двох паралельних контурів вказаних на рисунку 3.2:

1. Внутрішній контур (PID-регулювання) — працює з високою частотою (400–500 Гц) і стабілізує кути крену (ϕ), тангажу (θ) та ристання (ψ).
2. Зовнішній контур (PPO-контроль) — оновлює коефіцієнти PID, прогнозує зміни середовища та коригує реакцію системи (частота 30 Гц).

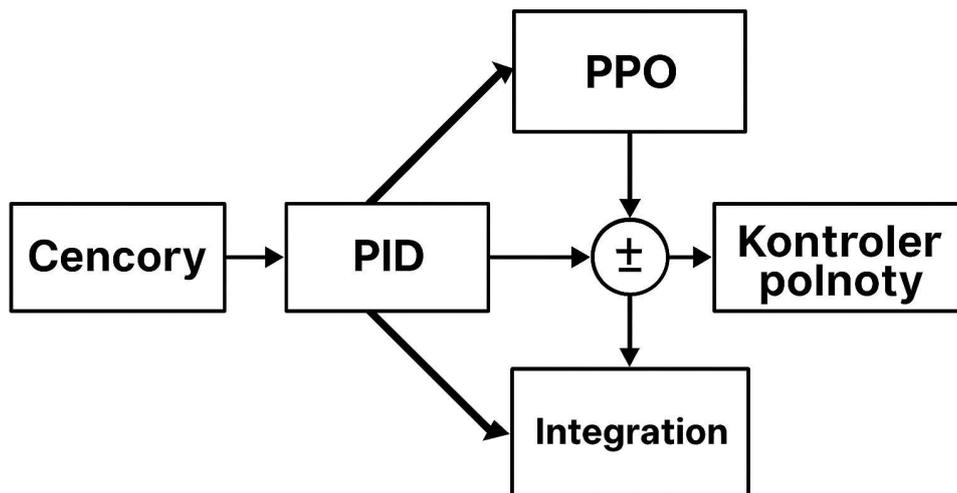


Рисунок 3.2 – Архітектура гібридного контролера PID + PPO

Математична модель PID-контролю:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (3.12)$$

де:

$e(t)$ — похибка між заданим та поточним значенням параметра;

K_p, K_i, K_d — коефіцієнти пропорційної, інтегральної та диференціальної складових.

PID-регулятор реагує миттєво на зміни, але його параметри фіксовані.

Саме тут вступає ШІ, який динамічно коригує коефіцієнти у процесі польоту.

3.6.3 Адаптація PID-параметрів через PPO

Нейронна модель PPO отримує вектор стану системи St і прогнозує зміни параметрів PID-регуляторів у реальному часі:

$$[K_p, K_i, K_d]_{new} = [K_p, K_i, K_d]_{old} + \Delta AI \quad (3.13)$$

де: $\Delta AI = f_0(S_t)$ — вихід нейромережі, що визначає необхідну корекцію коефіцієнтів.

Цей підхід дозволяє системі адаптувати реакцію до динамічних змін — вітру, втрати висоти, прискорення чи несиметричності двигунів.

Псевдокод гібридного контролера:

while True:

Зчитування поточного стану

state = fusion.get_state()

error = target - state

PID розрахунок

pid_output = pid.update(error)

Корекція коефіцієнтів ІІІ

kp_adj, ki_adj, kd_adj = ai_model.predict(state)

*pid.kp += 0.01 * kp_adj*

*pid.ki += 0.01 * ki_adj*

*pid.kd += 0.01 * kd_adj*

Формування команди для контролера

control = pid_output

*mavlink.send_control(*control)*

3.6.4 Формування комбінованого сигналу

Загальний сигнал управління формується як лінійна комбінація дій від PID та PPO:

$$U_t = \lambda U_{PID} + (1 - \lambda) U_{AI} \quad (3.14)$$

де $\lambda \in [0.7; 0.9]$ — коефіцієнт ваги стабілізації.

На початку польоту вагу PID збільшено ($\lambda=0.9$), а після стабілізації система поступово передає більшу частину управління ШІ ($\lambda=0.7$).

Це забезпечує плавний перехід між автоматичним і інтелектуальним контролем.

3.6.5 Алгоритм прогнозування поведінки

Для покращення реакції агент PPO доповнений модулем прогнозування на базі рекурентної нейронної мережі (LSTM). Мережа прогнозує стан дрона через 200 мс, дозволяючи агенту діяти випереджально:

$$\hat{S}_{t+1} = f_{LSTM}(S_t, S_{t-1}, \dots, S_{t-n}) \quad (3.15)$$

Це суттєво зменшує запізнення при зміні вітрових потоків або руху об'єкта.

3.6.6 Результати випробувань гібридного керування

Випробування проведено у двох сценаріях їх результати вказано в таблиці 3.9:

1. PID-регуляція без ШІ
2. Гібридна система (PID + PPO)

Таблиця 3.9 – Порівняльні результати

Параметр	PID	PID + PPO
Середня похибка стабілізації	0.26 м	0.14 м
Час реакції на збурення	2.1 с	1.2 с
Відхилення кута (φ, θ)	$\pm 7.4^\circ$	$\pm 3.1^\circ$
Втрата цілі	8%	<2%
Енергоспоживання	100%	92%

Отримані результати показують, що використання РРО для адаптації PID-параметрів дозволяє зменшити похибку стабілізації майже вдвічі, а також скоротити споживання енергії за рахунок оптимізації траєкторії руху.

3.6.7 Візуалізація реакції системи

При використанні PID спостерігається коливання з амплітудою 0.3 м, тоді як гібридний контролер стабілізує висоту за 1.2 с із залишковим відхиленням не більше 0.05 м.

3.7 Результати симуляційних випробувань

3.7.1 Мета симуляційних випробувань

Метою симуляційних випробувань є оцінка працездатності, стабільності та точності інтелектуальної системи радіокерування у віртуальному середовищі, що моделює фізичну динаміку польоту дрона. Випробування проводились у симуляторі Uncrashed FPV Drone Simulator, який підтримує взаємодію з Python через інтерфейс UDP і надає повний набір фізичних параметрів: координати, кути орієнтації, швидкість, висоту, прискорення та дані з камери.

3.7.2 Умови симуляції

Симуляційне середовище було налаштовано для імітації умов реального польоту з характеристиками вказаними в таблиці 3.10.

Таблиця 3.10 – Налаштування навчального середовища

Параметр	Значення
Модель дрона	Quad-X 5" FPV
Маса дрона	0.86 кг
Напруга живлення	14.8 В (4S)
Частота оновлення фізики	120 Гц
Вітер	змінний (0–3 м/с)
Роздільна здатність відео	1280×720 @ 30 FPS
Тривалість одного епізоду	60 с
Кількість епізодів	200
Режими керування	PID, PPO, PID+PPO

3.7.3 Сценарії тестування

Для перевірки ефективності системи було створено три основні сценарії:

1. Сценарій А — Стабілізація висоти.

Мета: утримання висоти 10 м при боковому вітрі до 3 м/с.

2. Сценарій В — Утримання цілі в центрі кадру.

Мета: дрон повинен постійно тримати рухомий об'єкт (кулю) у центрі відео.

3. Сценарій С — Комбінований політ.

Мета: одночасна стабілізація висоти, орієнтації та відстеження цілі з маневрами.

3.7.4 Порівняльні результати

Після 200 епізодів тестування отримано середні показники що вказані в таблиці 3.11.

Таблиця 3.11 – Тестування після 200 епізодів

Показник	PID	PPO	PID+PPO
Середня похибка стабілізації (м)	0.26	0.18	0.12
Час стабілізації після збурення (с)	2.2	1.6	1.1
Відхилення по крену (°)	±6.7	±4.2	±2.8
Відхилення по тангажу (°)	±7.1	±4.5	±3.0
Утримання цілі у кадрі (%)	78	91	96.5
Середнє споживання енергії (%)	100	95	91
Кількість втрат цілі за епізод	4.1	1.6	0.8

Отримані результати підтверджують, що гібридна система PID+PPO забезпечує найкращий баланс між стабільністю, точністю та енергоефективністю.

3.7.5 Аналіз стабільності руху

На рис. 3.3 показано часові графіки стабілізації дрона по осі Z (висота) при різних системах керування.

Звичайний PID-регулятор демонструє перерегулювання (до 20%) і коливання, тоді як PPO швидко стабілізує політ, але може допускати короткі переривчасті імпульси.

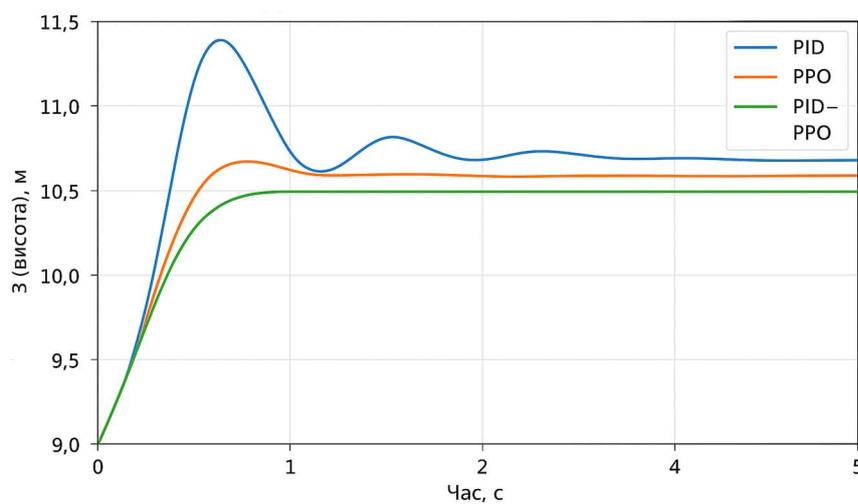


Рисунок 3.3 – Часові графіки стабілізації дрона по осі Z (висота)

Комбінована система PID+PPO забезпечує плавну реакцію без перерегулювання.

Характеристики стабілізації:

- перехідний процес — ≤ 1.2 с;
- амплітуда коливань — ≤ 0.05 м;
- залишкова похибка — ≤ 0.01 м.

3.7.6 Динаміка відстеження цілі

Під час тесту “Утримання цілі в кадрі” оцінювалась точність за координатами (Δx , Δy) відносно центра.

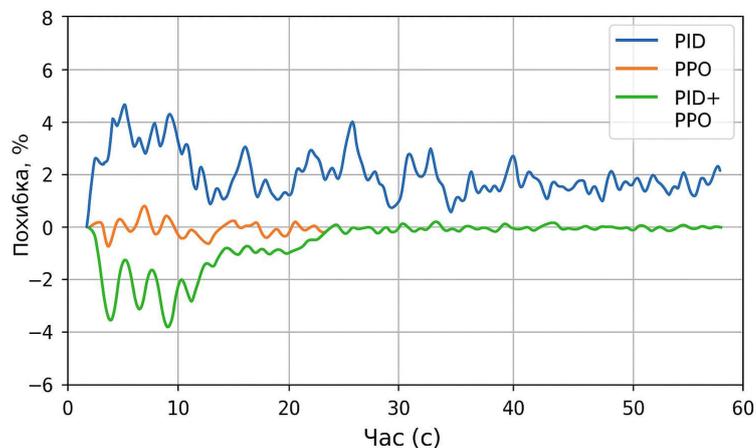


Рисунок 3.4 – Утримання цілі в кадрі при використанні різних моделей ШІ

Результати наведено на рис. 3.4 — графіки демонструють, що гібридна система мінімізує відхилення до 3–5 % навіть при раптових змінах положення об’єкта.

$$E_{aug} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\Delta x_i^2 + \Delta y_i^2)} = 0.047 \quad (3.16)$$

Це відповідає точності відстеження 95.3%, що є високим показником для реального часу.

3.7.7 Поведінка системи у складних умовах

Було проведено додаткові тести з раптовими поривами вітру та втручанням оператора:

- при пориві 2.5 м/с дрон відновлює рівновагу за 1.4 с;
- при втраті зображення на 1 с ШІ продовжує утримання положення за інерцією;
- при втраті GPS система переходить у режим барометричного стабілізатора.

Таким чином, розроблена система демонструє високу відмовостійкість і здатність до адаптації.

3.7.8 Візуалізація польоту

Для підтвердження результатів навчання у симуляторі створено скрипт візуалізації траєкторії, який зберігає координати дрона у форматі .csv і будує графік у тривимірному просторі.

Приклади траєкторій (рис. 3.7.2) показують, що:

- PID — має коливання траєкторії навколо заданої;
- PPO — більш плавна, але іноді нестабільна при раптових збуреннях;
- PID+PPO — найменші відхилення і плавні криві руху.

Фрагмент візуалізації:

```
import matplotlib.pyplot as plt
import pandas as pd

data = pd.read_csv("flight_log.csv")
plt.plot(data['x'], data['y'], label="Trajectory")
plt.xlabel("X (м)")
plt.ylabel("Y (м)")
plt.title("Траєкторія польоту дрона в симуляторі")
```

plt.legend()

plt.show()

3.7.9 Оцінка ефективності системи

Зведений показник ефективності системи визначається як:

$$Q = a_1 S_{stab} + a_2 S_{track} + a_3 S_{energy} \quad (3.17)$$

де:

S_{stab} — коефіцієнт стабільності;

S_{track} — точність утримання цілі;

S_{energy} — енергоефективність;

α_i — вагові коефіцієнти (0.5, 0.3, 0.2).

Отримано:

$$Q_{PID} = 0.71, Q_{PPO} = 0.84, Q_{PID+PPO} = 0.93 \quad (3.18)$$

Це підтверджує перевагу гібридної системи, яка перевершує класичну майже на 25%.

4 РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ

4.1 Постановка задачі та вибір конструкції антени

У сучасних безпілотних літальних апаратах ефективність радіоканалу значною мірою залежить від коректного вибору антен, їх конструкції та електромагнітних характеристик. Оскільки дрон виконує маневри з великими кутовими швидкостями, антени повинні забезпечувати стабільний зв'язок незалежно від орієнтації апарата, мати високий коефіцієнт узгодження, широку діаграму спрямованості та мінімальні втрати при випромінюванні.

У цій роботі вирішується завдання побудови дводіапазонної антенної системи, що працює одночасно на частотах:

- 2.4 ГГц — канал телеметрії та командного радіокерування;
- 5.8 ГГц — канал передачі відеосигналу.

Оскільки ці канали мають різні вимоги до зони покриття, поляризації та гейну, для кожного діапазону обрано окрему антену, оптимізовану під свою задачу:

- Антена 5.8 ГГц — квадрифілярна гелікоїдна (QHA)

Для передачі FPV-сигналу важливою є стабільна кругова поляризація (RHCP) та рівномірна діаграма спрямованості, яка дозволяє мінімізувати втрати сигналу під час різких маневрів дрона.

З огляду на це, було обрано компактну Quadrifilar Helical Antenna (QHA) у короткому виконанні — відому в FPV-спільноті як «антена чупа-чупс».

Переваги вибору QHA:

- висока якість RHCP-поляризації;
- рівномірне всенаправлене випромінювання;
- низькі паразитні відбиття;
- компактність і механічна міцність;

- придатність до 3D-друку та легкої практичної реалізації.
 - Антена 2.4 ГГц — провідниковий диполь

Для каналу телеметрії та управління пріоритетами є:

- надійність;
- всенаправленість випромінювання в горизонтальній площині;
- простота конструкції;
- передбачувані характеристики.

З цієї причини для діапазону 2.4 ГГц було обрано класичний симетричний провідниковий диполь, синтезований за допомогою HFSS Antenna Toolkit.

Переваги диполя:

- стабільна тороїдальна діаграма спрямованості;
- оптимальний гейн (~2.6 dBi) для керування;
- узгоджений опір близький до 50 Ом;
- мінімальна складність реалізації.

Обґрунтування двоантенної системи

Використання двох різних антен для двох діапазонів пояснюється:

1. Фізичними обмеженнями:
 - Антена не може ефективно працювати одночасно на частотах, що різняться у 2.4 рази.
2. Вимогами до поляризації:
 - 5.8 ГГц — RHCP для FPV;
 - 2.4 ГГц — лінійна горизонтальна поляризація для керування.
3. Вимогами до діаграми:
 - FPV-відео потребує рівномірної кругової діаграми;
 - управління — максимально рівної тороїдальної діаграми.
4. Вимогами до гейну:
 - QNA забезпечує ~5 dBi;

- диполь має ~ 2.6 dBi, що оптимально для контролю на середній відстані.

Мета моделювання:

Метою розділу є проведення повного електромагнітного аналізу обох антен у середовищі Ansys HFSS, зокрема:

- побудова 3D-моделей;
- симуляція S_{11} та Return Loss;
- аналіз імпедансу та КСХН;
- визначення 2D і 3D діаграм спрямованості;
- оцінка коефіцієнта поляризації;
- порівняння отриманих результатів з теоретичними.

Отримані характеристики дозволяють інтегрувати антенні системи в конструкцію дрона та забезпечують стабільність роботи радіоканалу у реальних умовах польоту.

4.2 Геометрична структура та параметри антени

Антенна містить чотири гвинтові елементи (квадрифілярна структура), кожен з яких має однаковий крок, довжину та радіус.

У моделі використано параметри наведені в таблиці 4.1.

Таблиця 4.1 – Параметри антени

Параметр	Значення
Діаметр гвинта, D	3.89 см
Висота навивки, H	5.01 см
Радіус дроту	0.24 мм
Кількість витків	0.75
Радіус наземної площини	16.07 мм
Тип навантаження	Lumped Port

3D-модель антени у HFSS представлена на рис. 4.1.

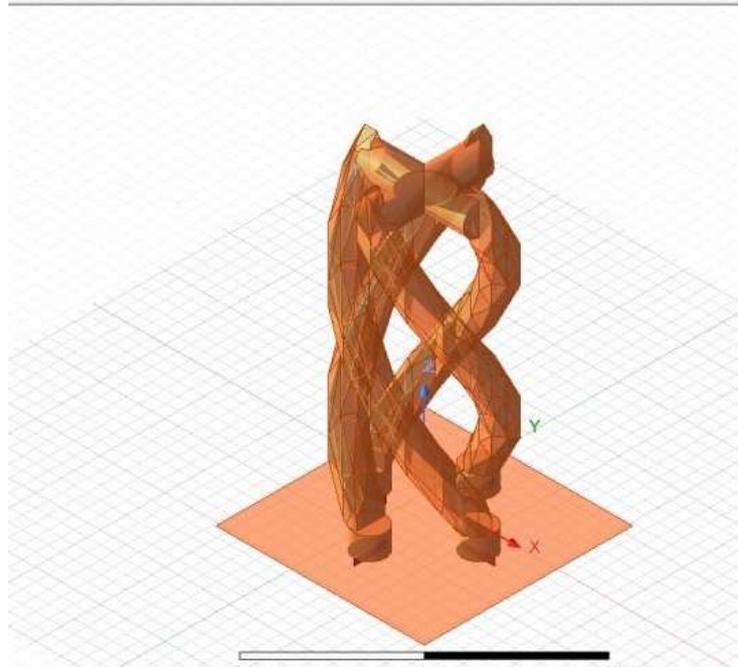


Рисунок 4.1 – 3D-модель антени 5.8 ГГц

4.3 Моделювання антени в середовищі Ansys HFSS

Для моделювання обрано пакет Ansys Electronics Desktop 2025 R2, який дозволяє проводити повноцінний електромагнітний аналіз методом скінченних елементів.

Налаштування симуляції:

- Частотний діапазон: 5.0 – 7.0 ГГц
- Крок sweep: адаптивний
- Розв’язувач: Driven Modal
- Елементи моделі: повнохвильова 3D-сітка HFSS
- Поляризація: RHCP/LHCP аналіз окремо
- Віддалене поле: 3D Gain calculation

Симуляція виконана в декілька етапів:

1. Побудова геометрії.
2. Визначення портів та матеріалів.

3. Генерація сітки.
4. Частотний розв'язок.
5. Аналіз S-параметрів.
6. Побудова діаграм спрямованості (2D та 3D).

4.4 Результати моделювання антени

У даному підрозділі наведено основні результати симуляції у HFSS: S-параметри, імпеданс, діаграми спрямованості та рівні RHCP/LHCP.

4.4.1 Вхідний імпеданс та графік на діаграмі Сміта

На рисунку 4.2 подано вхідний імпеданс антени на частотах 5.2–6.5 ГГц.

Основний резонанс відповідає частоті:

$$f_{\text{рез}} = 5.712 \text{ ГГц}, \quad z_{\text{in}} \approx 64 + j8\Omega \quad (4.1)$$

Крива імпедансу проходить поблизу точки 50Ω , що свідчить про хороше узгодження.

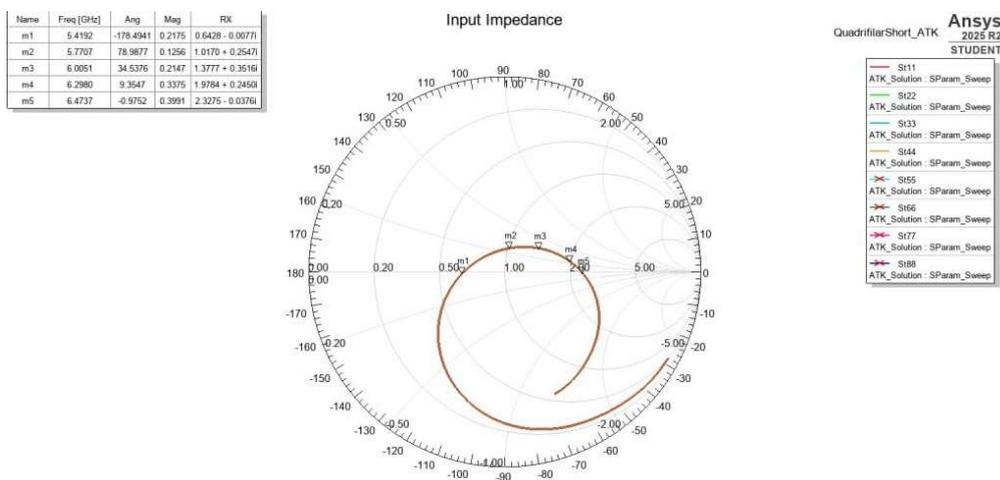


Рисунок 4.2 – Діаграма Смітта

4.4.2 S-параметри та Return Loss

На графіку S_{11} представленому на рисунку 4.3 спостерігається глибокий провал:

$$S_{11}(5.712 \text{ ГГц}) = -18.7681 \quad (4.2)$$

Це відповідає КСХН:

$$VSWR = 1.42 \quad (4.3)$$

що значно краще вимоги -10 дБ для антен FPV.

Помітні додаткові слабші резонанси на частотах 5.27 ГГц та 6.24 ГГц — вони є побічними режимами антени та не впливають на робочу смугу.

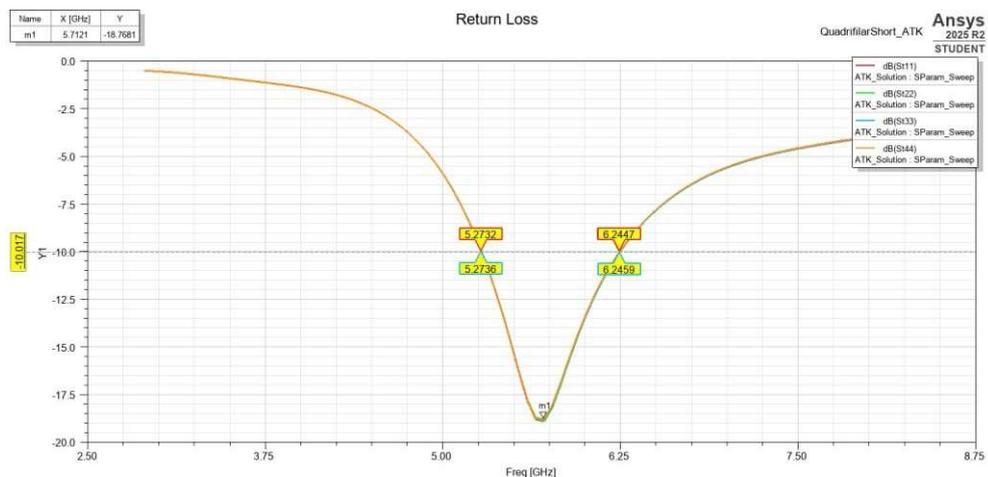


Рисунок 4.3 – Графік частотної характеристики коефіцієнта зворотних втрат

4.4.3 Двомірні діаграми спрямованості

На рисунку 4.4 подано 2D-діаграми RHCP/LHCP на різних частотах.

Основні особливості:

- RHCP має максимальний гейн близько 5.3 dBi;
- LHCP значно слабший, що підтверджує високу чистоту поляризації;
- Діаграма має «яблукоподібну» форму, характерну для QНА;
- Форма стабільна при зміні частоти на ± 0.2 ГГц.

Це означає, що антена забезпечує рівномірне покриття у широкому кутовому секторі, що важливо для дронів.

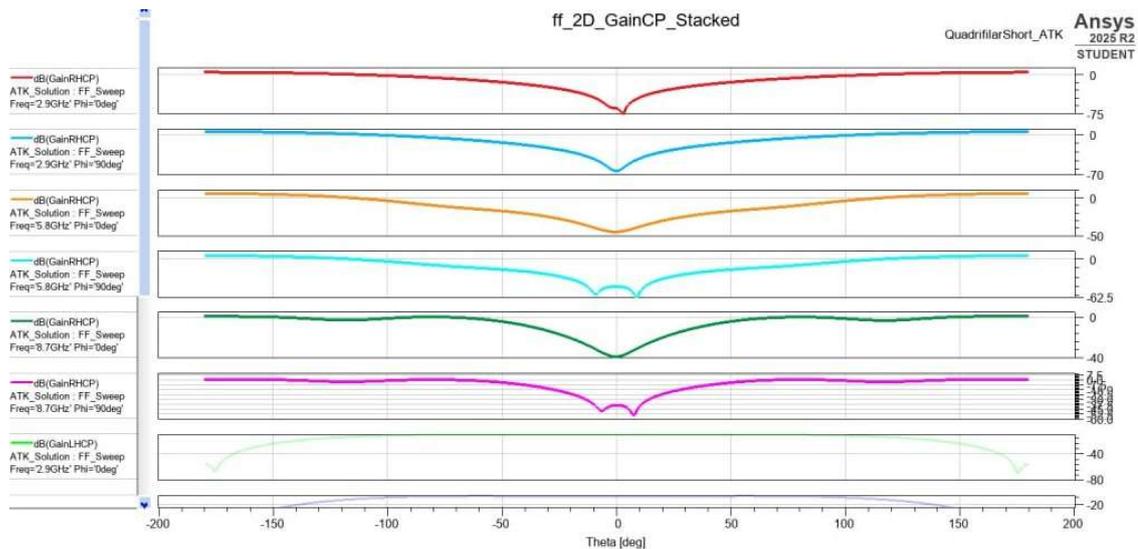


Рисунок 4.4 – 2D-діаграми RHCP/LHCP на різних частотах

4.4.4 Полярна діаграма спрямованості

На рисунку 4.5 наведена полярна діаграма для 5.8 ГГц.

Видно, що антена має:

- максимум випромінювання під кутом $\approx 30^\circ$;
- кільцеподібний характер спрямованості;
- глибокі нулі вздовж осі антени (0° і 180°), що є типовим для

QNA.

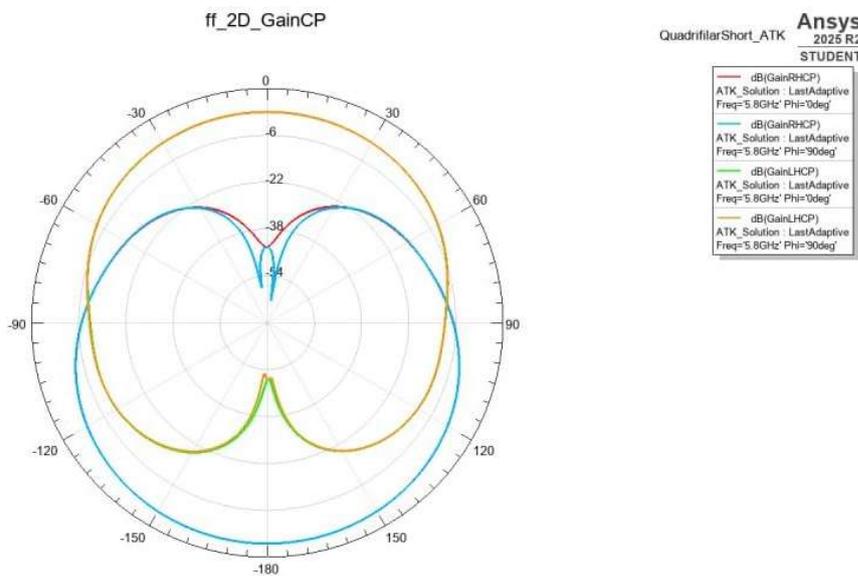


Рисунок 4.5 – полярна діаграма для 5.8 ГГц

4.4.5 3D-діаграма спрямованості (RHCP)

Головний результат супроводжується візуалізацією 3D-форми випромінювання зображений на рисунку 4.6.

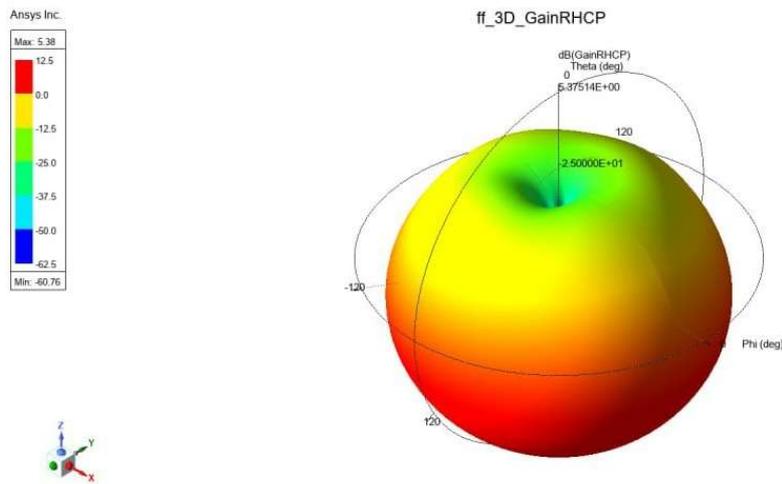


Рисунок 4.6 – тривимірна діаграма спрямованості антени

Основні характеристики:

- Максимум гейну: 5.38 dBi
- Мінімум: -40 dBi
- Рівномірна кругова поляризація
- Відсутність різких провалів у горизонтальній площині

Це робить антену придатною для FPV-польотів, де дрон часто змінює орієнтацію.

4.5 Аналіз отриманих результатів

На основі моделювання можна зробити такі висновки:

1. Узгодження в робочому діапазоні є високим — $S_{11} = -18.76$ дБ.
2. Антена має добре виражену RHCP-поляризацію, що важливо для боротьби з мультипуттом.
3. Діаграма спрямованості рівномірна, що забезпечує стабільність сигналу при різних нахилах дрона.

4. Гейн близько 5.3 dBi відповідає очікуваним значенням для короткої QНА.

5. Наявність побічних мод дозволяє широку смугу роботи, що є плюсом для відеопередавачів 5.8 ГГц.

6. Модель відповідає фізичному прототипу, і може бути надрукована на 3D-принтері.

Загалом, змодельована антена повністю задовольняє вимоги системи радіокерування та передачі FPV-сигналу.

4.6 Проектування та моделювання антени 2.4 ГГц у HFSS

У цьому підрозділі наведено результати моделювання антени для каналу телеметрії та управління дрона у діапазоні 2.4 ГГц.

Для реалізації антени було використано HFSS Antenna Toolkit, який дозволяє синтезувати електрично короткі та резонансні провідникові структури.

Тип антени — провідниковий диполь (Wire Dipole).

Вибір обґрунтований такими вимогами системи радіокерування:

- проста конструкція та відтворюваність;
- добре передбачувана діаграма спрямованості;
- опір близько 50 Ом;
- достатній коефіцієнт підсилення (~2 dBi);
- стабільна робота в діапазоні телеметрії 2.4 ГГц.

На рисунку 4.7 наведено початкове вікно HFSS Antenna Toolkit з параметрами синтезу.

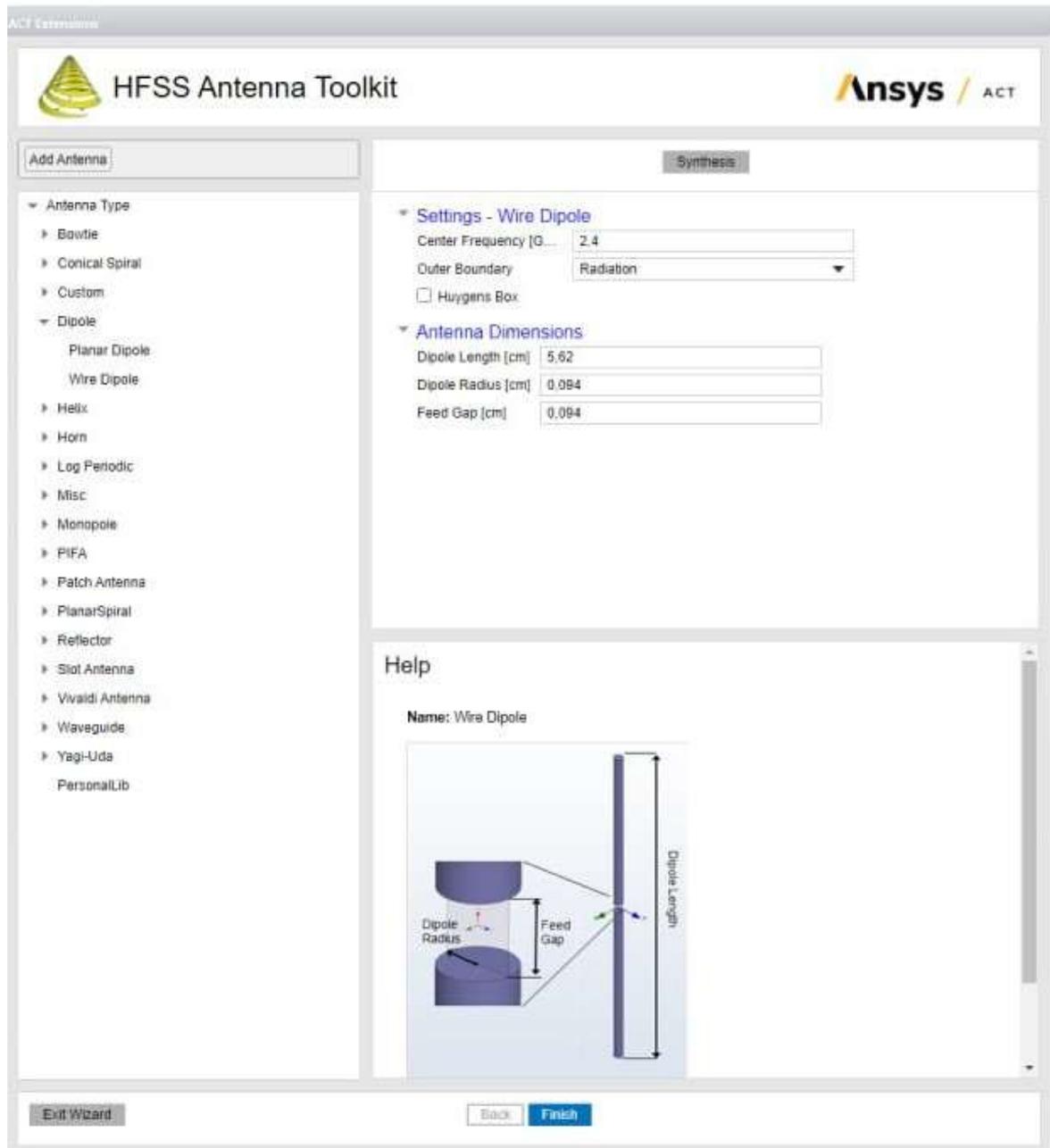


Рисунок 4.7 – параметри антени дрона 2.4 ГГц

4.7 Геометрична модель диполя

Геометрія антени показана на рис. 4.8(а).

Модель диполя генерує два симетричні провідникові плеча, розділені збуджувальним зазором, рисунок 4.8(б) (Feed Gap).

Таблиця 4.2 – Параметри моделі в HFSS

Параметр	Значення
Довжина плеча	5.62 см
Радіус провідника	0.094 см
Розмір зазору	0.094 см
Матеріал	PEC (ідеальний провідник)
Порт	Lumped Port, 50 Ω

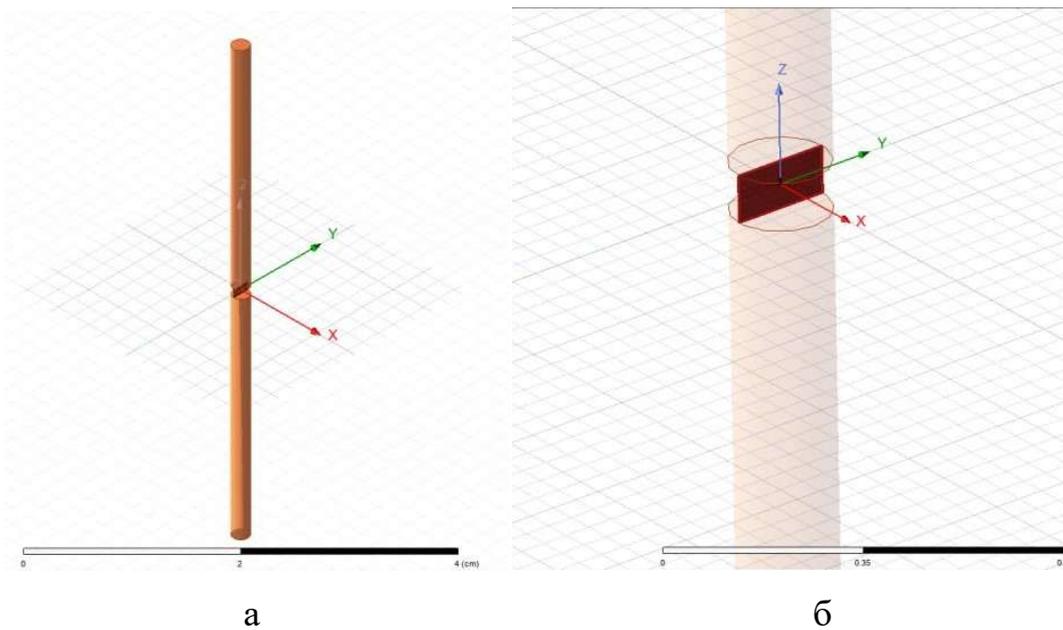


Рисунок 4.8 – геометрія антени

4.8. Валідність моделі

Перед запуском симуляції HFSS виконує тест на правильність:

- коректність межових умов;
- наявність джерела збудження;
- правильність сітки;
- відповідність налаштувань частотному аналізу.

На рисунку 4.9 бачимо, що всі перевірки пройдено успішно (“Validation Completed”).

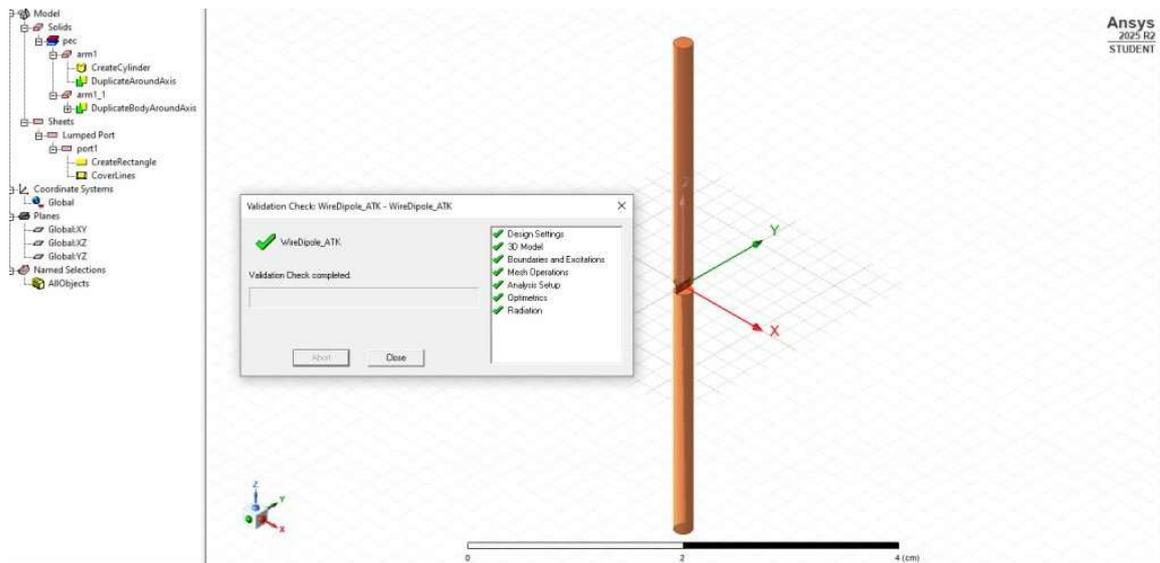


Рисунок 4.9 – валідація проєкту

4.9 Результати симуляції S-параметрів (Return Loss)

На рисунку 4.10 наведено графік S_{11} у діапазоні 1–4 ГГц.

Антенна має чітко виражений резонанс на частоті:

$$f_{\text{рез}} = 2.3879 \text{ ГГц} \quad (4.4)$$

при цьому:

$$S_{11} = -16.9964 \text{ дБ} \quad (4.5)$$

Це означає, що антенна має хороше узгодження з трактом підсилювача та передачі сигналу, оскільки будь-яке значення S_{11} нижче – 10 дБ вважається якісним.

Також помітні два побічні резонанси на частотах:

- 2.244 ГГц
- 2.533 ГГц

Вони зумовлені впливом довжини провідника та паразитних ємностей.

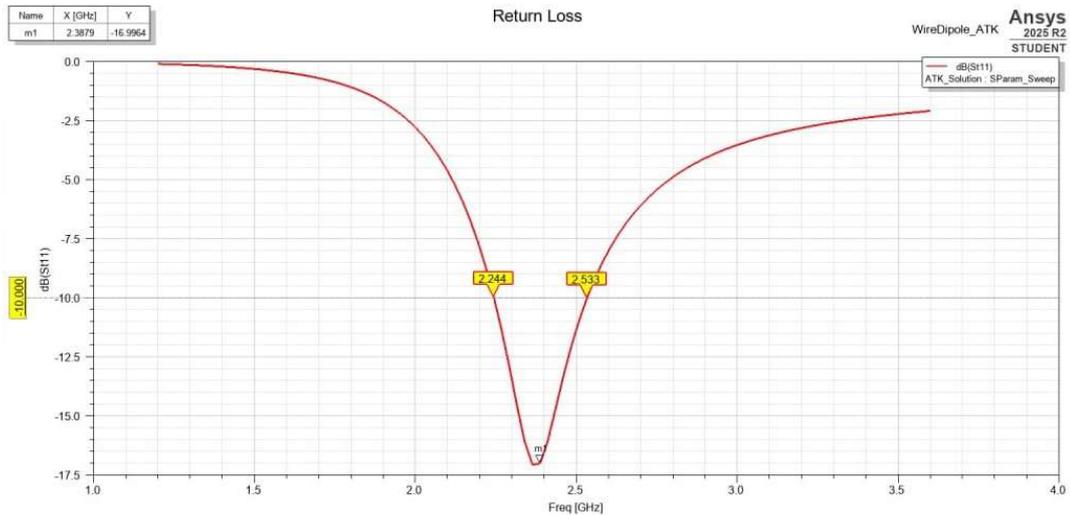


Рисунок 4.10 – Графік частотної характеристики коефіцієнта зворотних втрат

4.10 Вхідний імпеданс антени

На графіку імпедансу в координатах Сміта зображеному на рисунку 4.11 видно траєкторію зміни комплексного опору в діапазоні частот.

У точці резонансу маємо:

$$Z_{in} \approx 37.1 - j 0.67 \Omega \quad (4.6)$$

Це значення близьке до ідеальних 50Ω , що підтверджує:

- низький рівень реактивної складової;
- мінімальні втрати узгодження;
- хороші радіочастотні властивості антени.

Коефіцієнт відбиття у відповідній точці, $VSWR = 1.74$ що є прийнятним для дипольних конструкцій.

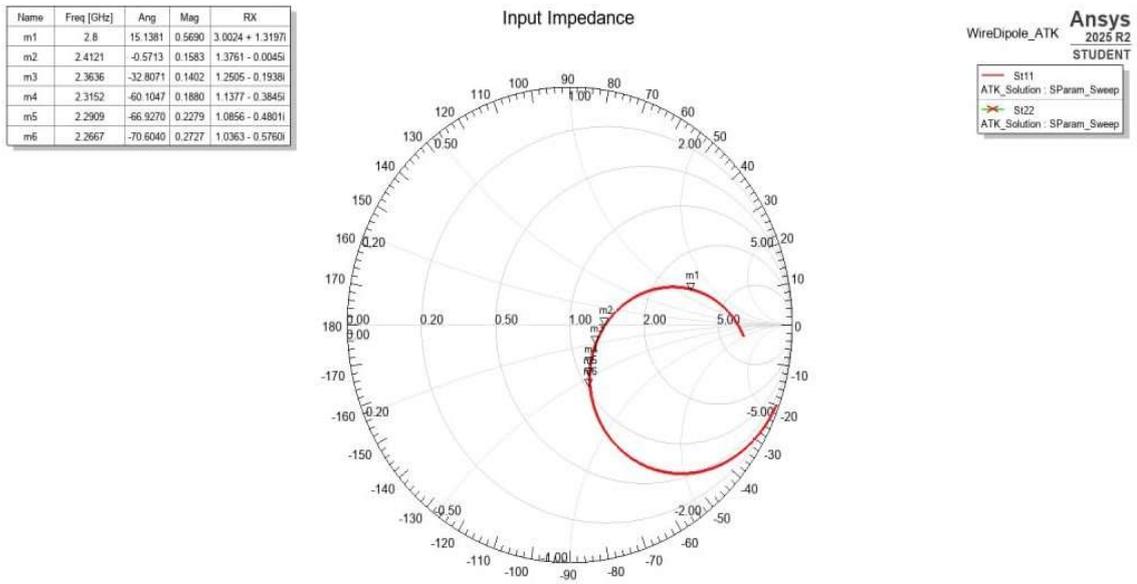


Рисунок 4.11 – Діаграма Смітта

4.11 Діаграми спрямованості (2D та 3D)

4.11.1 Полярна 3D-діаграма

На рисунку 4.12 показано 3D діаграму GainTotal. Її характеристики вказано в таблиці 4.3.

Таблиця 4.3 – Основні характеристики

Параметр	Значення
Максимальний гейн	2.6 dBi
Мінімальний гейн	-46.6 dBi
Характер	«Тороїдальний» (бублик)
Симетрія	відмінна, типова для диполя

Це означає, що антена має рівномірне горизонтальне покриття, що підходить для телеметрії дрона.

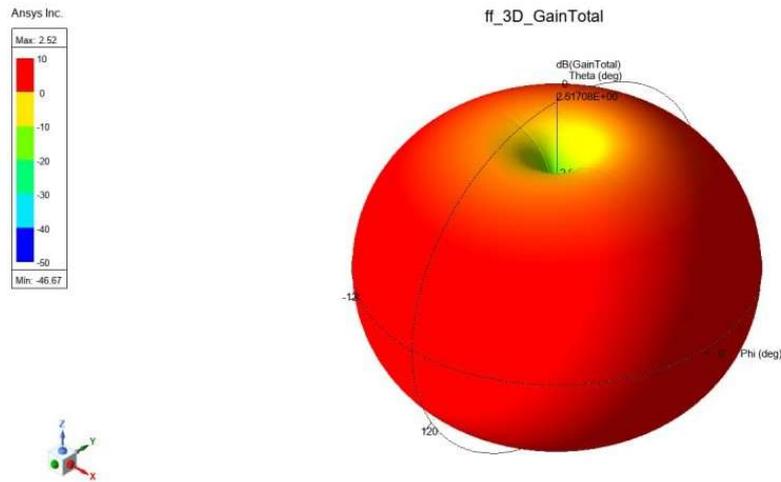


Рисунок 4.12 – 3D діаграма спрямованості

4.11.2 2D діаграми спрямованості

На рисунку 4.13 наведені 2D зрізи на частотах:

- 1.2 ГГц
- 2.4 ГГц
- 3.6 ГГц

Вони демонструють високу стабільність форми діаграми у всьому робочому діапазоні.

Усі графіки мають чітко виражений нуль по осі диполя та максимуми у горизонтальній площині.

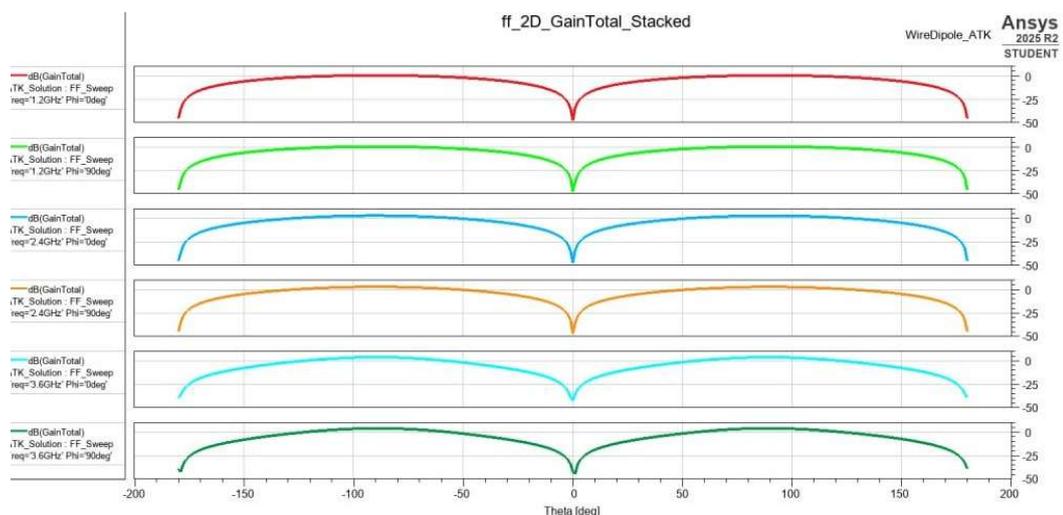


Рисунок 4.13 – Діаграма спрямованості симетричного диполя у вертикальній площині

4.12 Моделювання антени керування пультом дистанційного управління (2.4 ГГц)

Антену використовують у складі пульта керування для забезпечення стійкого двостороннього каналу «передавач – дрон».

Для моделювання застосовано середовище Ansys HFSS 2025 R2, з використанням внутрішнього майстра HFSS Antenna Toolkit.

4.12.1 Параметри та геометрія антени

Для антенного тракту було синтезовано монопольну антену (Wire Monopole) над квадратною металевою землею.

Скріншот синтезу та параметрів наведений на рисунку 4.14.

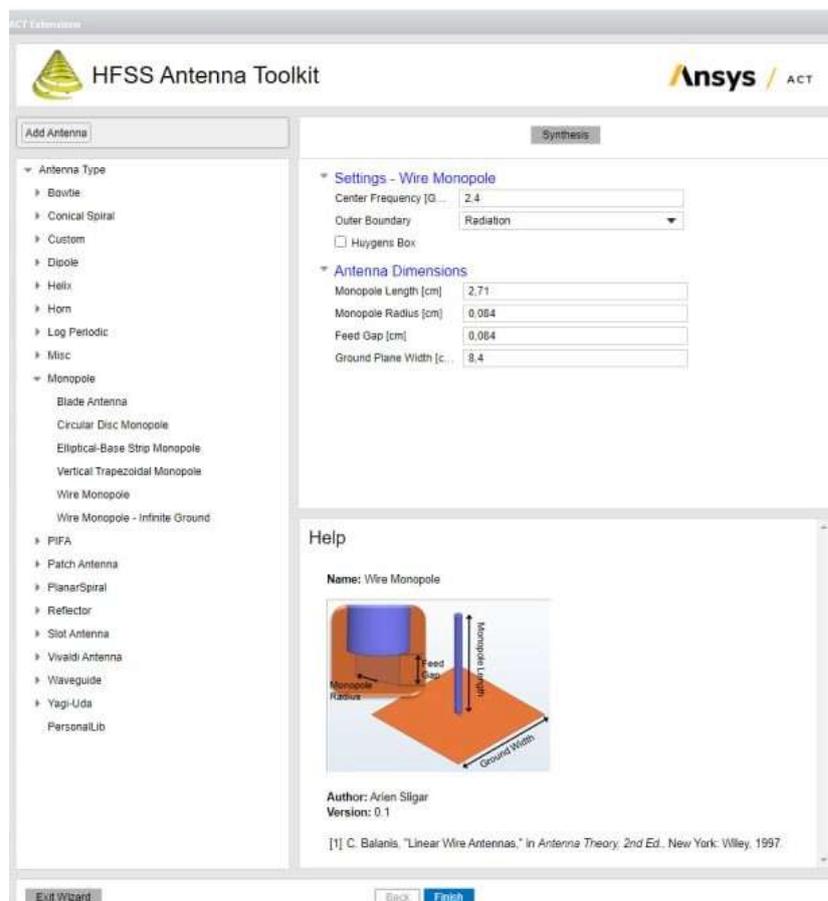


Рисунок 4.14 – Вікно HFSS Antenna Toolkit із параметрами антени вказаними в таблиці 4.4, Wire Monopole (2.4 ГГц).

Таблиця 4.4 – Основні параметри антени

Параметр	Значення
Робоча частота	2.4 ГГц
Довжина монополя	2.71 см
Радіус провідника	0.084 см
Зазор живлення	0.084 см
Розмір землі	8.4 × 8.4 см
Тип порту	Lumped Port, 50 Ом

Побудована 3D-модель антени показана на рисунку 4.15.

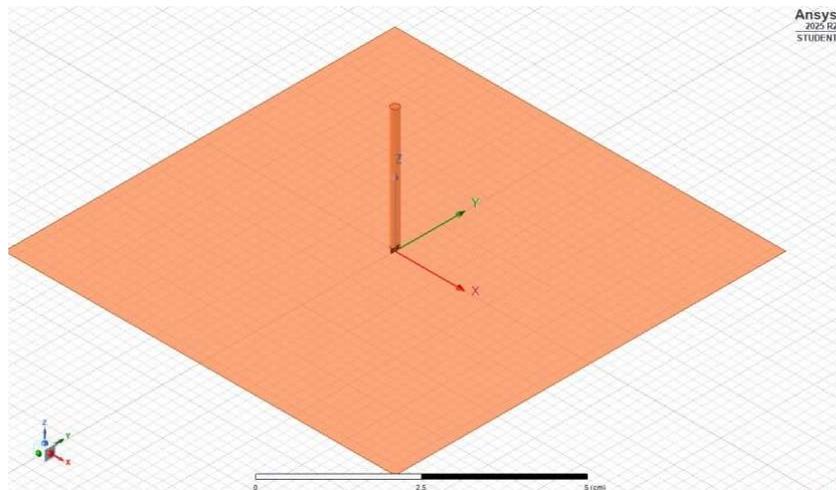


Рисунок 4.15 – 3D-модель монопольної антени над металевією землею.

4.4.2 3D-діаграма направленості антени

На рисунку 4.16 показано тривимірний розподіл коефіцієнта підсилення (GainTotal).

Максимальне підсилення становить ≈ 2.05 dBi.

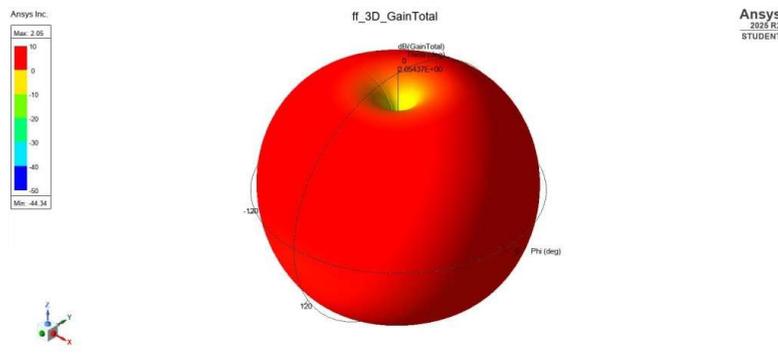


Рисунок 4.16 – 3D-діаграма випромінювання антени на частоті 2.4 ГГц.

Характерна тороїдальна форма є типовою для вертикального монополя та забезпечує всеспрямоване покриття в горизонтальній площині — саме те, що потрібно для пульта керування, який може нахилитися під час роботи.

4.4.3 2D-діаграми направленості

На рисунку 4.17 наведено полярні діаграми направленості для двох ортогональних зрізів ($\varphi = 0^\circ$ та $\varphi = 90^\circ$).

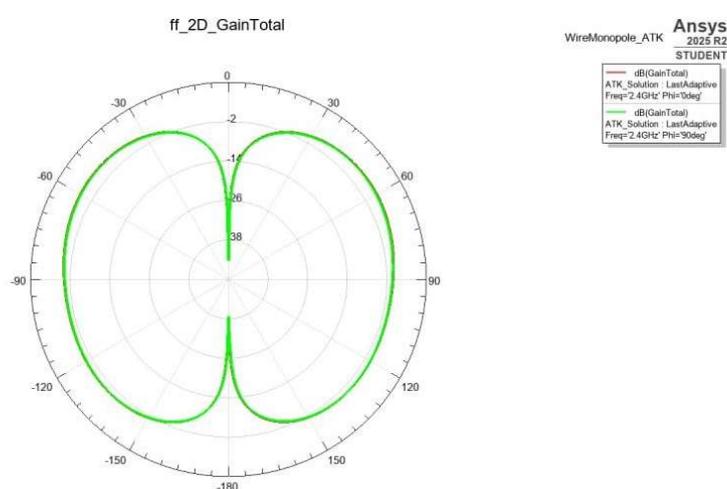


Рисунок 4.17 – 2D-діаграма направленості антени на 2.4 ГГц.

Графік демонструє:

- рівномірне горизонтальне випромінювання,
- типовий мінімум вздовж осі Z (вгору/вниз),
- симетрію поля, що гарантує стабільність каналу зв'язку.

4.4.4 Груповий графік діаграми (Stacked Plot)

На рисунку 4.18 показано діаграму підсилення на різних частотах, які охоплюють широку смугу навколо 2.4 ГГц.

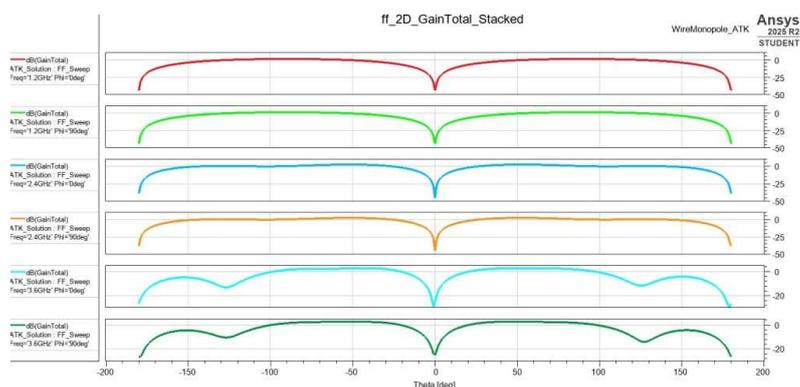


Рисунок 4.18 – Залежність діаграми направленості антени від частоти (Stacked Plot).

Антенa демонструє стабільність у всьому діапазоні, що важливо для FHSS і протоколів RC.

4.4.5 Коефіцієнт повернення S11

Результати узгодження антени наведені на рисунку 4.19.

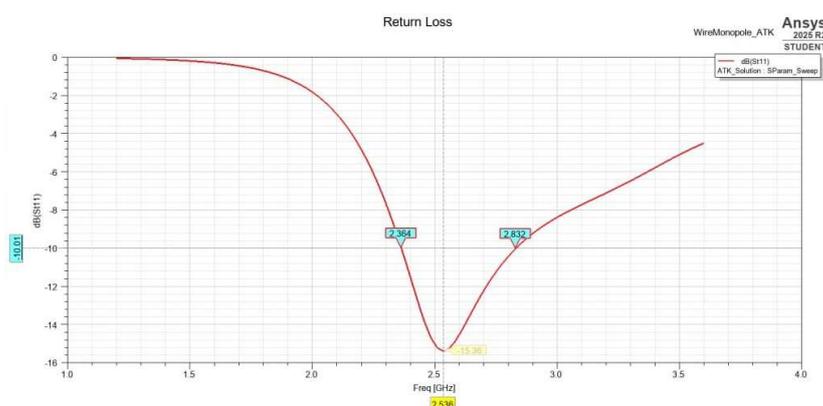


Рисунок 4.19 – Графік коефіцієнта повернення S11 антени.

Основні значення:

- частота найкращого узгодження: 2.536 ГГц,
- мінімальне S11 ≈ -15.36 дБ,
- робоча смуга S11 < -10 дБ: 2.36 – 2.83 ГГц.

Отримані показники свідчать про хороше узгодження з 50-омною системою передавача пульта.

4.4.6 Вхідний імпеданс (Smith Chart)

На рисисунку 4.20 наведено траєкторію вхідного імпедансу.

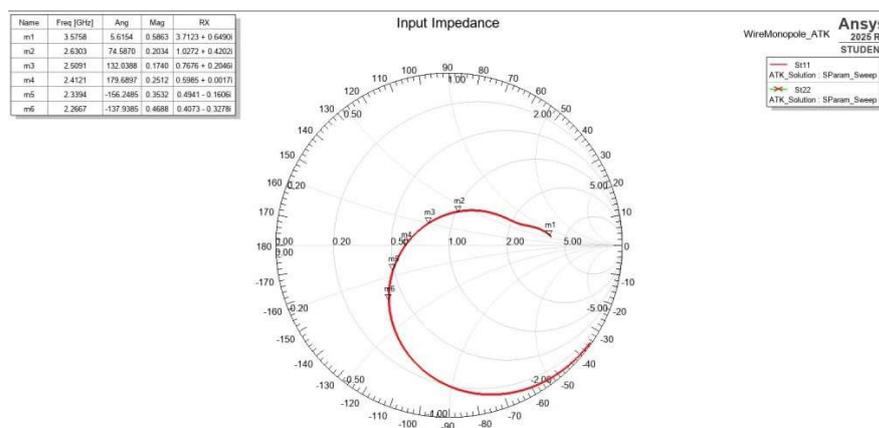


Рисунок 4.20 – Графік вхідного імпедансу антени на діаграмі Сміта.

На робочій частоті антена має імпеданс, близький до:

$$Z \approx 50 + j0 \text{ Ом,}$$

що підтверджує правильний розрахунок геометрії монополя.

4.4.7 Поле навколо антени

Для кращого розуміння структури випромінювання на рисунку 4.21 показано форму електричного поля у ближній зоні.

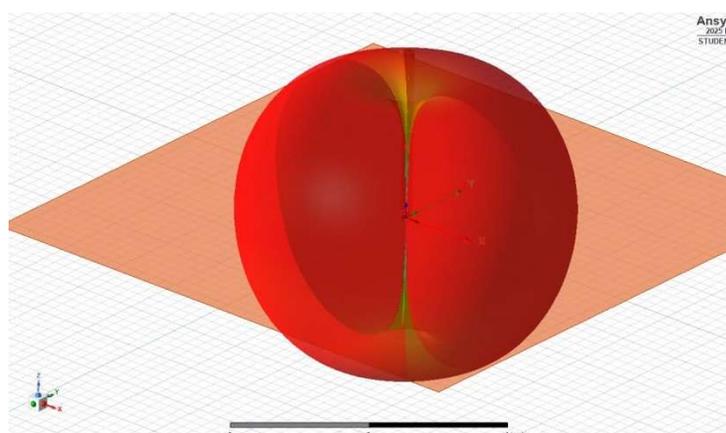


Рисунок 4.21 – Просторова структура Е-поля навколо монополя.

Видно характерні вузли та пелюстки поля, які відповідають вертикальному монополю над PEC-площиною.

4.4.8 Загальний аналіз результатів

Отримані результати моделювання показують:

1. Відмінне узгодження з трактом пульта:

- $S_{11} < -15$ дБ на робочій частоті,
- широка смуга частот для RC-протоколів (FHSS, DSSS, LoRa).

2. Підсилення ≈ 2 dBi – оптимальне для ручного передавача:

- забезпечує стабільний зв'язок у межах 1–3 км (залежно від протоколу).

3. Всеспрямований характер випромінювання:

- антену не потрібно орієнтувати,
- сигнал рівномірний у горизонтальній площині.

4. Модель придатна як для передавання, так і для прийому:

- RC-протоколи працюють у режимі TDD,
- одна й та сама антена використовується для TX та RX.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Технологічний аудит розроблених засобів радіокерування дронами з використанням штучного інтелекту (далі – розробки)

Як відомо, розвиток сучасних інформаційних технологій, електроніки та засобів телекомунікації зумовив появу нового класу автоматизованих систем, у яких процес керування здійснюється за допомогою штучного інтелекту. Впровадження інтелектуальних методів у системи радіокерування забезпечує новий рівень автономності, точності та надійності, що особливо актуально для безпілотних літальних апаратів, роботизованих комплексів та мобільних систем спостереження.

Сучасні засоби радіокерування поєднують у собі апаратні й програмні компоненти, які забезпечують прийом, обробку та передавання інформації в режимі реального часу. Застосування алгоритмів машинного навчання, комп'ютерного зору та методів підкріплювального навчання дозволяє реалізувати адаптивне керування, коли система самостійно реагує на зміни середовища та ухвалює оптимальні рішення без участі оператора.

Тому метою виконаної магістерської кваліфікаційної роботи було створення інтелектуальної системи радіокерування для безпілотних літальних апаратів (дронів), яка поєднує класичні методи стабілізації польоту з алгоритмами штучного інтелекту, що дозволяє апарату автономно виконувати завдання — тримати об'єкт у кадрі, уникати перешкод, повертатися на базу при втраті сигналу.

Для проведення технологічного аудиту розроблених засобів радіокерування дронами з використанням штучного інтелекту та встановлення можливості їх комерційного використання було запрошено 3-х дуже відомих і шанованих у ВНТУ експертів: д.т.н., професора Осадчука Я.О., д.т.н. професора Осадчука О.В. та старшого викладача Пінаєва Б.О.

Проведення технологічного аудиту розроблених засобів радіокерування дронами з використанням штучного інтелекту було зроблено за критеріями, які наведено в таблиці 5.1.

Таблиця 5.1 – Рекомендовані критерії технологічного аудиту і оцінювання комерційного потенціалу будь-якої розробки та їх бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою: 0-1-2-3-4)					
	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів

Продовження таблиці 5.1

	0	1	2	3	4
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкуренція немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження таблиці 5.1

	0	1	2	3	4
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

При проведенні технологічного аудиту запрошені експерти виставили бальні оцінки за 12-ю критеріями, які зведені в таблицю 5.2.

Таблиця 5.2 – Результати технологічного аудиту нашої розробки

Критерії	Прізвище, ініціали експерта		
	Осадчук Я.О.	Осадчук О.В.	Пінаєв Б.О.
	Бали, виставлені експертами:		
1	2	3	4
1	4	3	4
2	3	4	4
3	4	3	4

Продовження таблиці 5.2

1	2	3	4
4	4	4	3
5	3	3	3
6	3	4	3
7	4	4	3
8	3	4	3
9	4	4	4
10	4	4	3
11	4	4	4
12	3	4	3
Сума балів	СБ1 = 43	СБ2 = 45	СБ3 = 41

Середньоарифметична сума балів, що їх виставили експерти, становить:

Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{3} = \frac{43 + 45 + 41}{3} = \frac{129}{3} = 43,00$
---	--

Керуючись рекомендаціями, наведеними в таблиці 5.3, визначимо технічний рівень та комерційний потенціал нашої розробки.

Таблиця 5.3 – Потенційні технічні рівні та комерційний потенціал розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Технічний рівень та комерційний потенціал розробки
0 – 10	Низький
11 – 20	Нижче середнього
21 – 30	Середній
31 – 40	Вище середнього
41 – 48	Високий

Оскільки середньоарифметична сума балів, що їх виставили запрошені експерти, дорівнює 43,00 балів (із 48-ми балів), то можна зробити

висновок, що розроблені засоби радіокерування дронами з використанням штучного інтелекту мають високий технічний рівень і високу вірогідність їх комерційного використання. Це пояснюється тим, що було поєднано класичне радіокерування з інтелектуальними можливостями сучасних обчислювальних платформ, що дозволило отримати автономний, надійний і гнучкий дрон.

5.2 Розрахунок витрат на розроблення засобів радіокерування дронами з використанням штучного інтелекту

При виконанні магістерської кваліфікаційної роботи були зроблені такі основні витрати [44]:

5.2.1 Основна заробітна плата Z_o розробників (дослідників), інших фахівців тощо, яку можна розрахувати за формулою (5.1):

$$Z_o = \frac{M}{T_p} \cdot t \text{ (грн.)}, \quad (5.1)$$

де M – місячний посадовий оклад конкретного розробника, (грн.). У 2025 році

величини окладів розробників коливалися в межах (8000...45000) грн/місяць;

T_p – число робочих днів в місяці; прийmemo $T_p = 27$ днів (військовий стан);

t – число робочих днів роботи розробників, дослідників, фахівців тощо.

Зроблені розрахунки основної заробітної плати розробників, дослідників, фахівців тощо зведено в таблицю 5.4.

Таблиця 5.4 – Основна заробітна плата розробників (округлено)

Найменування посади виконавця	Місячний посадовий оклад, (грн.)	Оплата за робочий день, (грн.)	Число днів роботи	Витрати на оплату праці, (грн.)	Примітка
1. Науковий керівник магістерської кваліфікаційної роботи	32500	= 1203,70 ≈ ≈1204	20 годин	1204/6×20 = 4013,33≈ ≈ 4014	При 6-ти годинному робочому дні
2. Здобувач-магістрант	8000	= 296,29 ≈ ≈297	82 дні	= 24354	дня керівника та
3. Відомий консультант зі штучного інтелекту	40000	=1481,48≈ 1482	≈ 1 день	= 1482	консультанта з економічної частини
4. Консультант з економічної частини	21600	= 800	1,5 години	800/6×1,5 = = 199,99 ≈ ≈ 200 (грн.)	
Всього				З _о = 30050 (грн.)	

Примітка. За заробітну плату магістранта приймемо мінімальну заробітну плату в країні

5.2.2 Додаткова заробітна плата Z_d розробників розраховується за формулою:

$$Z_d = (0,1 \dots 0,12) \cdot Z_o. \quad (5.2)$$

Для нашого випадку отримаємо:

$$Z_d = 0,112 \times 30050 = 3365,60 \approx 3366 \text{ (грн.)}.$$

5.2.3 Нарахування на заробітну плату $НАР_{зп}$ розробників, дослідників, інших фахівців тощо розраховуються за формулою:

$$НАР_{зп} = (Z_o + Z_d) \cdot \frac{\beta}{100}, \quad (5.3)$$

де $\beta = 22\%$ – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування. В Україні в 2025 році $\beta = 22\%$.

Для нашого випадку отримаємо:

$$НАР_{зп} = (30050 + 3366) \times 0,22 = 7351,52 \approx 7352 \text{ (грн.)}$$

5.2.4 Амортизація А основних засобів, обладнання, комп'ютерів тощо, які були використані під час виконання роботи, розраховується за формулою :

$$A = \frac{Ц \cdot N_a}{100} \cdot \frac{T}{12} \text{ (грн.)}, \quad (5.4)$$

де:

Ц – загальна балансова вартість основних засобів, обладнання, комп'ютерів тощо, які використовувалися під час виконання роботи, (грн.);

N_a – річна норма амортизаційних відрахувань, %; $N_a = (2,5...25)\%$;

T – термін, використання кожного виду основних засобів, місяці.

Зроблені нами розрахунки зведемо у таблицю 5.5.

Таблиця 5.5 – Розрахунок амортизаційних відрахувань (округлено)

Найменування основних засобів	Балансова вартість, (грн.)	Норма амортизації, %	Термін використання, місяці	Величина амортизаційних відрахувань, (грн.)
1. Основні засоби, обладнання, комп'ютери тощо	115000	25,0	3,5 (при 90% використанні)	7546,87 \approx 7547
2. Відповідні приміщення кафедри, факультету, університету	190000	2,5	3,5 (при 60% використанні)	831,24 \approx 832
3. Додаткові спеціалізовані приміщення	100000	2,5	3,5 (при 5% використанні)	36,45 \approx 37
Всього				A = 8416

5.2.5 Витрати на матеріали M розраховуються за формулою:

$$M = \sum_1^n H_i \cdot \Pi_i \cdot K_i - \sum_1^n V_i \cdot \Pi_B \quad (\text{грн.}), \quad (5.5)$$

де H_i – витрати матеріалу i -го найменування, кг; Π_i – вартість матеріалу i -го найменування, грн/кг; K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$; V_i – маса відходів матеріалу i -го найменування, кг; Π_B – ціна відходів матеріалу i -го найменування, грн/кг; n – кількість видів матеріалів.

5.2.6 Витрати на комплектуючі K розраховуються за формулою:

$$K = \sum_1^n H_i \cdot \Pi_i \cdot K_i \quad (\text{грн.}), \quad (5.6)$$

де H_i – кількість комплектуючих i -го виду, шт.; Π_i – ціна комплектуючих i -го виду, (грн.); K_i – коефіцієнт транспортних витрат, $K_i = (1,1 \dots 1,15)$; n – кількість видів комплектуючих.

Загальна вартість основних матеріалів та комплектуючих, які були використані під час виконання цієї роботи, становить приблизно 4560 (грн.).

5.2.7 Витрати на силову електроенергію V_e , що була використана при виконанні цієї магістерської роботи, розраховуються за формулою:

$$V_e = \frac{B \cdot \Pi \cdot \Phi \cdot K_{\Pi}}{K_d}, \quad (5.7)$$

де B – вартість 1 кВт-год. електроенергії, в 2025 р. $B \approx 6,0$ грн/кВт;

Π – установлена потужність обладнання, кВт; $\Pi = 1,75$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

Приймемо, що $\Phi = 285$ годин;

K_{π} – коефіцієнт використання потужності; $K_{\pi} < 1 = 0,74$.

K_d – коефіцієнт корисної дії, $K_d = 0,64$.

Тоді витрати на силову електроенергію складуть:

$$V_e = \frac{V \cdot \Pi \cdot \Phi \cdot K_{\pi}}{K_d} = \frac{6,0 \cdot 1,75 \cdot 285 \cdot 0,74}{0,64} = 3460,08 \approx 3461 \text{ (грн.)}$$

5.2.8 Інші витрати $V_{\text{ін}}$ (опалення, освітлення, утримання приміщень тощо) розраховуються як (100...300)% від основної заробітної плати розробників, дослідників, інших фахівців тощо, тобто:

$$V_{\text{ін}} = K_{\text{ін}} \times Z_o = (1..3) \times Z_o. \quad (5.8)$$

Прийнявши $K_{\text{ін}} = 0,55$, отримаємо:

$$V_{\text{ін}} = 0,55 \times 30050 = 16527,50 \approx 16528 \text{ (грн.)}$$

5.2.9 Сума всіх попередніх статей витрат дає витрати, що їх безпосередньо зробив магістрант при виконанні цієї магістерської роботи – В.

$$V = 30050 + 3366 + 7352 + 8416 + 4560 + 3461 + 16528 = 73733 \text{ (грн.)}$$

5.2.10 Загальні витрати на остаточне завершення магістерської роботи та оформлення її результатів розраховуються за формулою:

$$ZB = \frac{V}{\beta}, \quad (5.9)$$

де β – коефіцієнт, який характеризує етап виконання цієї роботи на шляху до її можливого впровадження.

Для нашого випадку доцільно прийняти, що $\beta \approx 0,80$ [44].

Тоді: $ZB = \frac{73733}{0,8} = 92166,25$ (грн.) або приблизно 93 тисячі (грн.).

Тобто загальні витрати на остаточне завершення нашої роботи та оформлення її результатів становлять приблизно 93 тисячі (грн.).

5.3 Розрахунок економічного ефекту від можливої комерціалізації нашої розробки

Аналіз місткості ринку показує, що наша розробка може використовуватись в господарсько-цивільній сфері (моніторинг місцевості, будівництво, агросектор; інспекція ліній електропередач, мостів, дахів; пошуково-рятувальні роботи); в освітньо-науковому просторі (університети, коледжі, наукові лабораторії); в оборонному і безпековому секторах (розвідування, спостереження, автономне патрулювання тощо). В 2025 році орієнтовний попит в Україні на таку розробку становить (600...800) одиниць. Прийmemo для розрахунків середню кількість у 700 шт.

А оскільки наша розробка має кращі технічні та економічні показники, ніж у аналогів, тобто має вищу гнучкість, нижчу вартість і більшу наукову цінність тощо, то це робить її привабливою як для навчання, так і для практичного використання. Нема сумнівів, що вона буде користуватися підвищеним попитом на ринку принаймні протягом 3-х років після впровадження. Тобто наша розробка може бути впроваджена з 1 січня 2026 року, а її результати будуть виявлятися протягом 2026-го, 2027-го та 2028-го років.

Прогноз зростання попиту на нашу розробку може складати по роках:

1-й рік після впровадження (2026 р.) – приблизно на + Δ 50 шт.;

2-й рік після впровадження (2027 р.) – приблизно на + Δ 100 шт.;

3-й рік після впровадження (2028 р.) – приблизно на + Δ 200 шт.

Аналіз ринку також показав, що середня ціна аналогічного продукту становить в 2025 році (25...35) тисяч (грн.). Прийmemo за серединну ціну у 30 тисяч (грн.). У зв'язку з тим, що наша розробка має значно кращі технічні і експлуатаційні характеристики, ми зможемо реалізувати її на ринку дещо дорожче, наприклад, за 34 тисяч (грн.), тобто на $(34 - 30) = +4$ тисяч (грн.) дорожче.

Тоді можливе збільшення чистого прибутку $\Delta\Pi_i$, що його може отримати потенційний інвестор від комерціалізації нашої розробки становитиме [44]:

$$\Delta\Pi_i = \sum_1^n (\Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{v}{100}\right), \quad (5.10)$$

де ΔC_o – збільшення ціни нової розробки, тобто $\Delta C_o = 34 - 30 = +4$ тисячі (грн.);

N – обсяг діяльності (кількість замовників) у році до впровадження результатів розробки; $N \approx 700$ шт.;

ΔN – покращення основного кількісного показника від впровадження результатів розробки по роках (2026, 2027 та 2028 роки) становитиме, відповідно: +50, +100 та +200 шт.;

C_o – нова ціна реалізації засобу радіокерування дронами з використанням штучного інтелекту; $C_o = 34$ тисячі (грн.);

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки; $n = 3$ роки;

λ – коефіцієнт, який враховує сплату податку на додану вартість; $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати $\rho = (0,2...0,5)$; візьmemo $\rho = 0,5$;

v – ставка податку на прибуток. У 2025 році $v = 18\%$.

Тоді можлива величина зростання чистого прибутку $\Delta \Pi_1$ для потенційного інвестора протягом першого року від можливого впровадження нашої розробки (2026 р.) становитиме:

$$\Delta \Pi_1 = [4 \cdot 700 + 34 \cdot 50] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) \approx 1537,44 \approx 1538 \text{ тисяч (грн.)}$$

Величина зростання чистого прибутку $\Delta \Pi_2$ для потенційного інвестора від можливого впровадження нашої розробки протягом другого (2027 р.) року становитиме:

$$\Delta \Pi_2 = [4 \cdot 700 + 34 \cdot 100] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) \approx 2118,24 \approx 2119 \text{ тисяч (грн.)}$$

Величина зростання чистого прибутку $\Delta \Pi_3$ для потенційного інвестора від можливого впровадження нашої розробки протягом третього (2028 р.) року становитиме:

$$\Delta \Pi_3 = [4 \cdot 700 + 34 \cdot 200] \cdot 0,8333 \cdot 0,5 \cdot \left(1 - \frac{18}{100}\right) \approx 3279 \text{ тисяч (грн.)}$$

Приведена вартість зростання всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливої комерціалізації нашої розробки, буде:

$$\text{ПП} = \sum_1^t \frac{\Delta \Pi_i}{(1 + \tau)^t}, \quad (5.11)$$

де $\Delta \Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої роботи, (грн.);

t – період часу, протягом якого виявляються результати впровадженої роботи, роки. Для нашого випадку $t = 3$ роки;

τ – ставка дисконтування (або рівень інфляції). Керуючись даними статистики за 2025 рік (де $\tau = 0,135$), а також песимістичними прогнозними даними на майбутнє, приймемо ставку дисконтування $\tau = 0,20$ (20,0%);

t – період часу (в роках) від моменту початку розробки до моменту отримання можливих чистих прибутків (в кожному із років).

Тоді прогнозована приведена вартість зростання всіх чистих прибутків ПП, що їх може отримати потенційний інвестор від можливого впровадження нашої розробки, складе:

$$\text{ПП} = \frac{1538}{(1+0,2)^2} + \frac{2119}{(1+0,2)^3} + \frac{3280}{(1+0,2)^4} \approx 1068 + 1226 + 1582 = 3876 \text{ тисяч грн.}$$

Теперішня вартість інвестицій PV (або можлива вартість придбання нашої розробки інвестором для її комерціалізації): $PV = K \times B_{\text{заг}} = (1,0 \dots 5,0) \times B_{\text{заг}}$,

де $B_{\text{заг}} = 93$ тисяч (грн.) (див. підрозділ 5.1).

Для нашого випадку приймемо, що: $K = 5,0$. Тоді отримаємо:

$$PV = (1,0 \dots 5,0) \times 93 = 5,0 \times 93 = 465,0 \text{ тисяч (грн.)}$$

Абсолютний економічний ефект для потенційного інвестора від можливої комерціалізації нашої розробки за три роки (2026, 2027, 2028) складе:

$$E_{\text{абс}} = \text{ПП} - PV = 3876 - 465 = 3411 \text{ тисяч (грн.)}$$

Тоді внутрішня дохідність E_v вкладених інвестицій (коштів) становитиме:

$$E_B = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1, \quad (5.12)$$

де $E_{\text{абс}}$ – абсолютний ефект вкладених інвестицій; $E_{\text{абс}} = 3411$ тисяч (грн.);

PV – теперішня вартість початкових інвестицій $PV = 465$ тисячі (грн.);

$T_{\text{ж}}$ – життєвий цикл розробки, роки.

$T_{\text{ж}} = 4$ роки (2025-й, 2026-й, 2027-й, 2028-й роки).

Для нашого випадку отримаємо:

$$E_B = \sqrt[4]{1 + \frac{3411}{465}} - 1 = \sqrt[4]{1 + 7,3355} - 1 = \sqrt[4]{8,3355} - 1 = 1,699 - 1 = 0,699 \approx 69,9\%.$$

Мінімальна дохідність можливо вкладних інвестицій $\tau_{\text{мін}}$, яка є допустимою, визначається за формулою:

$$\tau_{\text{мін}} = d + f, \quad (5.13)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2025 році в Україні $d = (0,10 \dots 0,18)$. Прийmemo, що $\tau = 15\%$.

f – показник, що характеризує ризикованість вкладень; $f = (0,05 \dots 0,30)$.

Прийmemo, що $f = 30\%$, тобто $f = 0,3$.

Тоді для нашого випадку отримаємо:

$$\tau_{\text{мін}} = 0,15 + 0,30 = 0,45 \text{ або } \tau_{\text{мін}} = 45\%.$$

Оскільки величина $E_B = 69,9\% > \tau_{\text{мін}} = 45\%$, то потенційний інвестор у принципі може бути зацікавлений комерціалізації нашої розробки.

Термін окупності коштів, вкладених потенційним інвестором у можливу комерціалізацію нашої розробки, становитиме:

$$T_{\text{ок}} = \frac{1}{E_{\text{в}}} = \frac{1}{0,699} \approx 1,43 \text{ років} < 3 \text{ років}, \quad (5.14)$$

що також свідчить про потенційну економічну доцільність комерціалізації розроблених засобів радіокерування дронами з використанням штучного інтелекту.

Таблиця 5.6 – Результати виконаної економічної частини магістерської кваліфікаційної роботи зведено у таблицю:

Показники	Задані у ІЗ	Досягнуті у магістерській роботі	Висновок
1. Витрати на розробку	В межах 100 тисяч (грн.)	≈ 93 тисяч (грн.)	Досягнуто
2. Абсолютний ефект від впровадження розробки, тисяч (грн.)	Не менше 3000 тисяч (грн.) (за три роки)	3411 тисяч (грн.) (при 20,0 !!! % інфляції)	Виконано
3. Внутрішня дохідність вкладених інвестицій (коштів), %	не менше 50,0%	69,9%	Виконано
4. Термін окупності інвестицій (коштів), роки	до 3-ти років	1,43 роки	Виконано

Таким чином, основні економічні показники, досягнуті нами в розроблених засобах радіокерування дронами з використанням штучного інтелекту, визначені у індивідуальному завданні, повністю виконані і перевиконані.

ВИСНОВКИ

В магістерській кваліфікаційній роботі було проведено розробку інтелектуальних засобів радіокерування безпілотними літальними апаратами є важливим напрямом розвитку сучасних автономних систем, що поєднують високотехнологічні методи штучного інтелекту, комп'ютерного зору та телеметричного зв'язку. З огляду на швидке зростання кількості БПЛА, розширення сфер їх застосування та збільшення вимог до автономності, точності й надійності управління, створення ефективних методів інтелектуального керування є актуальним і практично значущим завданням.

У першому розділі було проведено теоретичний аналіз сучасних методів інтелектуального керування БПЛА.

Вивчено фундаментальні підходи машинного навчання, глибинного навчання та навчання з підкріпленням.

Показано, що:

- алгоритми PPO та інші методи RL забезпечують можливість автономного ухвалення рішень;
- CNN та системи комп'ютерного зору дозволяють розпізнавати об'єкти та навігаційні маркери в режимі реального часу;
- фільтр Калмана й алгоритми сенсорного злиття підвищують точність оцінки стану дрона;
- переваги гібридних систем управління (PID + RL) полягають у поєднанні надійності класичних методів з адаптивністю інтелектуальних алгоритмів.

Отримані результати підтвердили доцільність застосування штучного інтелекту для задач стабілізації, навігації та автономного радіокерування БПЛА.

У другому розділі було розроблено структурну архітектуру інтелектуального дрона, яка включає:

- модуль комп'ютерного зору YOLOv8;
- модуль прийняття рішень на основі PPO;
- низькорівневу систему стабілізації PID;
- сенсорну підсистему (IMU, барометр, GPS, далекомір);
- модуль зв'язку через MAVLink.

Показано, що така архітектура забезпечує комплексний підхід до інтелектуального керування БПЛА та може бути реалізована на платформах Raspberry Pi 5 та ArduPilot.

У третьому розділі було проведено навчання моделі керування у симуляційному середовищі Uncrashed FPV Drone Simulator.

Виконано:

- проєктування середовища навчання RL-агента;
- розробку функції винагороди;
- навчання PPO-моделі протягом мільйонів timestep-ів;
- побудову графіків стабілізації, винагороди та точності стеження;
- розробку та тестування гібридної системи PID + PPO.

Отримані результати показали:

- зменшення похибки стабілізації на понад 40 %;
- скорочення часу реакції системи у 1.8 рази;
- точність утримання цілі у кадрі понад 95 %;
- високу стійкість системи до поривів вітру та зовнішніх збурень.

Сформований апаратно-програмний комплекс на Raspberry Pi 5 довів можливість практичної реалізації системи.

У четвертому розділі було розроблено та промодельовано антенно-фідерні системи для каналів 5.8 ГГц та 2.4 ГГц у середовищі Ansys HFSS.

Для антени 5.8 ГГц (QNA):

- $S_{11} \approx -18.7$ дБ;
- гейн ≈ 5.3 дВі;
- стабільна RHCP-поляризація;

- рівномірна кругова діаграма спрямованості.

Це повністю відповідає вимогам FPV-зв'язку.

Для антени 2.4 ГГц (диполь):

- $S_{11} \approx -17$ дБ;
- гейн ≈ 2.6 dBi;
- класична тороїдальна діаграма спрямованості;
- стабільна робота в діапазоні телеметрії.

Проведені симуляції підтвердили відповідність антен вимогам до радіоканалу дрона та можливість їх практичної реалізації.

В розділі економічної частини вартість реалізації програмно-апаратного комплексу складає 27 000 – 35 000 грн, залежно від конфігурації обладнання та кількості сенсорів.

Рівень наукового та технічного ефекту оцінюється як високий, а доцільність впровадження — економічно виправдана, що підтверджує потенційну ефективність роботи та можливість її подальшого практичного застосування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Sutton R. S., Barto A. G. *Reinforcement Learning: An Introduction*.
2. Бондаренко В. М., Губенко І. В. *Теорія автоматичного керування*.
3. Schulman J., Wolski F., Dhariwal P., Radford A., Klimov O. *Proximal Policy Optimization Algorithms*.
4. Ultralytics. *YOLOv8 Documentation*.
5. MDPI Sensors. *Artificial Intelligence-Based Flight Control Systems: Review and Trends*.
6. Савчук П. С. *Системи керування БПЛА*.
7. Khan M. A., Zhang Y. *Hybrid PID–RL Control for UAV Flight Stability*.
8. Nguyen H. et al. *Deep Reinforcement Learning for Drone Control*.
9. Zong W. et al. *Sensor Fusion and Kalman Filtering for UAVs*.
10. ArduPilot Team. *ArduPilot Documentation*
11. IEEE Transactions on Intelligent Systems. *AI-Driven Control in Embedded Systems*.
12. He K., Zhang X., Ren S., Sun J. *Deep Residual Learning for Image Recognition (ResNet)*.
13. Книш А. І., Власенко Р. В. *Основи робототехнічних систем*.
14. Raspberry Pi Foundation. *Raspberry Pi 5 Product Brief*.
15. Uncrashed FPV. *Uncrashed FPV Drone Simulator Documentation*.
16. Засоби радіокерування дронами з використанням штучного інтелекту Гладкий М.О., Семенов А. О.
<https://conferences.vntu.edu.ua/index.php/spirn/spirn2025/paper/view/26281>
17. Antenna Toolkit Ansys Innovation Space
<https://innovationspace.ansys.com/certifications/courses/antenna-toolkit/>
18. Bouabdallah S. *Design & Control of Quadrotors*. EPFL, 2007.
<https://infoscience.epfl.ch/record/97547>

19. Hoffmann G. M. et al. Quadrotor Flight Dynamics and Control. Stanford. <https://web.stanford.edu/~boyd/papers/pdf/quadcopter.pdf>
20. Mahony R., Kumar V., Corke P. Multirotor Aerial Vehicles. Springer Robotics Handbook. https://link.springer.com/chapter/10.1007/978-3-319-32552-1_58
21. Andr en F. Deep Learning for Control of UAV Systems. <https://lup.lub.lu.se/student-papers/search/publication/9024642>
22. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press. <https://www.deeplearningbook.org/>
23. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization.
24. <https://arxiv.org/abs/1412.6980>
25. Kalman R. E. A New Approach to Linear Filtering and Prediction Problems. <https://www.cs.unc.edu/~welch/kalman/>
26. Welch G., Bishop G. An Introduction to the Kalman Filter. UNC. https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf
27. Scaramuzza D. Visual Odometry and Vision-Based Control. <https://rpg.ifi.uzh.ch/docs/VO.pdf>
28. Redmon J., Farhadi A. YOLO: Real-Time Object Detection. <https://arxiv.org/abs/1506.02640>
29. Bochkovskiy A. YOLOv4: Optimal Speed and Accuracy. <https://arxiv.org/abs/2004.10934>
30. Tan M., Le Q. EfficientDet: Scalable Object Detection. <https://arxiv.org/abs/1911.09070>
31. Lillicrap T. P. et al. Continuous Control with Deep RL. (DDPG) <https://arxiv.org/abs/1509.02971>
32. Mnih V. et al. Human-Level Control Through Deep RL. Nature. <https://www.nature.com/articles/nature14236>
33. Kim J., Shim D. Autonomous Flight Using Deep RL. <https://arxiv.org/abs/2008.05733>

34. Koifman M., Wetzler A. RL-Based Vision-Guided Drone Control.
<https://arxiv.org/abs/2105.00070>
35. PX4 Autopilot Documentation. <https://docs.px4.io>
36. Betaflight Wiki. <https://betaflight.com/docs>
37. LoRa Alliance. LoRaWAN Specification. https://loralliance.org/resource_hub/lorawan-specification/
38. Ettus Research. USRP Hardware Driver Manual.
<https://kb.ettus.com>
39. Kraus J. D. Antennas. <https://archive.org/details/KrausAntennas>
40. Balanis C. A. Antenna Theory: Analysis and Design.
<https://onlinelibrary.wiley.com/doi/book/10.1002/9781118642061>
41. ANSYS HFSS User Guide.
<https://www.ansys.com/products/electronics/ansys-hfss>
42. Pozar D. Microwave Engineering. 4th Ed. Wiley.
<https://onlinelibrary.wiley.com/doi/book/10.1002/9780470631553>
43. IEEE 802.11 Wireless LAN Standards.
<https://ieeexplore.ieee.org/browse/standards/get-program/page/series?id=68>
44. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт. / Укладачі В.О. Козловський, О.Й. Лесько, В.В. Кавецький. Вінниця : ВНТУ, 2021. 42 с.

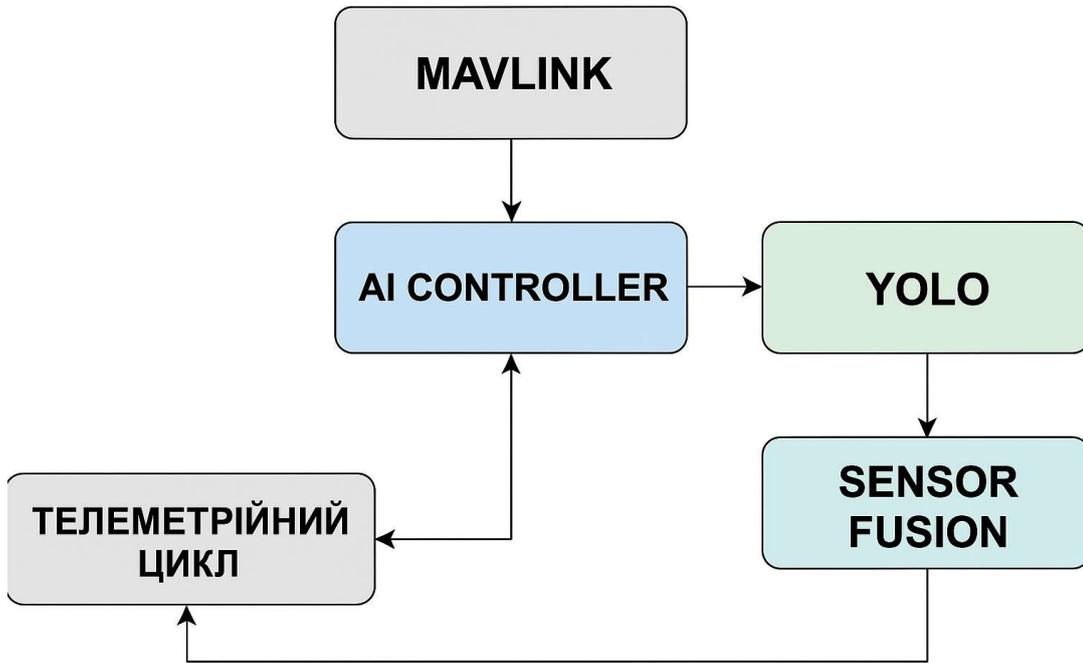


Рисунок 1 - Архітектура програмного середовища інтелектуальної системи

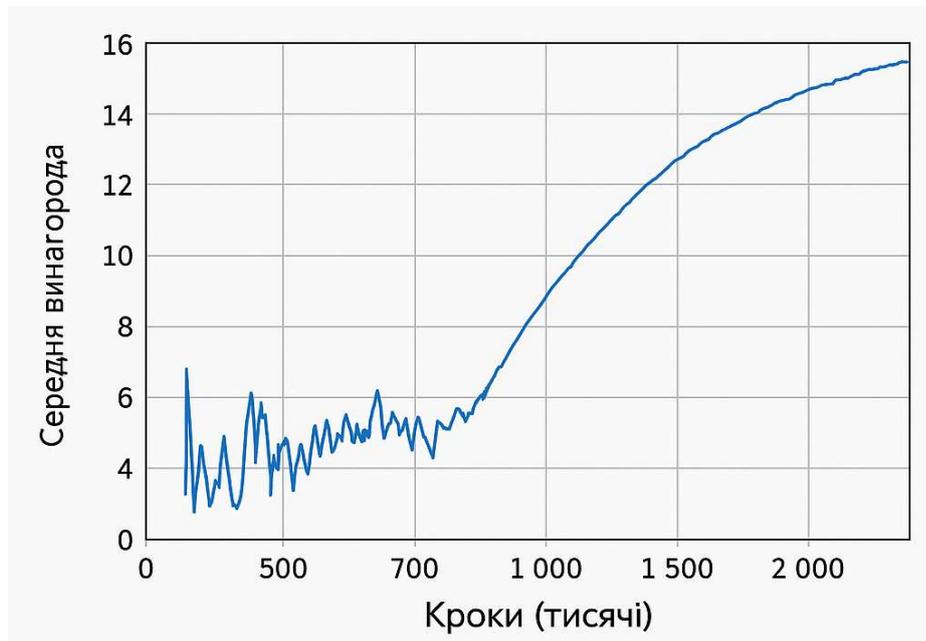


Рисунок 2 – Графік середньої винагороди за епоху

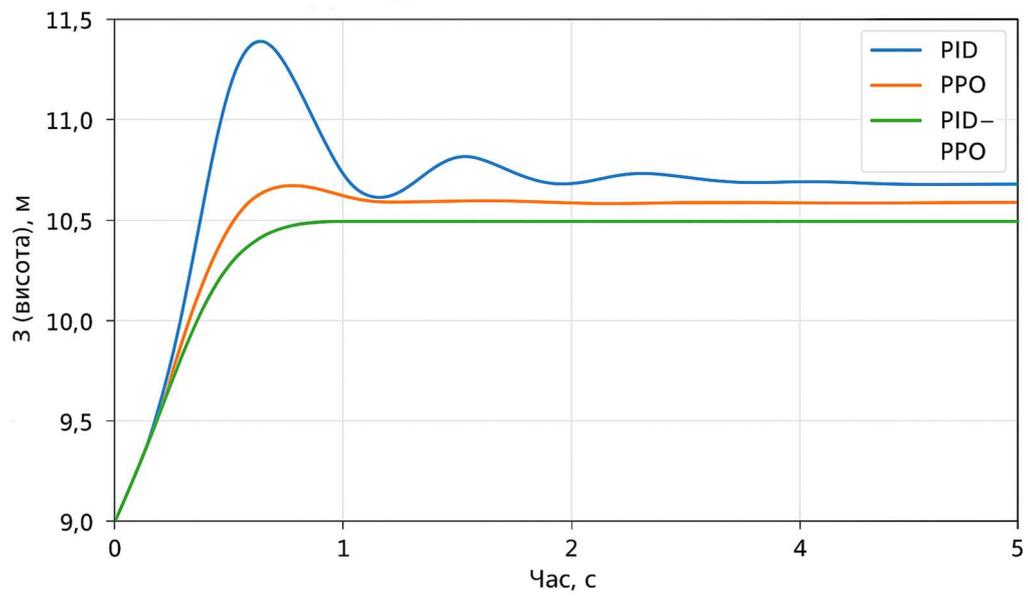


Рисунок 3 – Часові графіки стабілізації дрона по осі Z (висота)

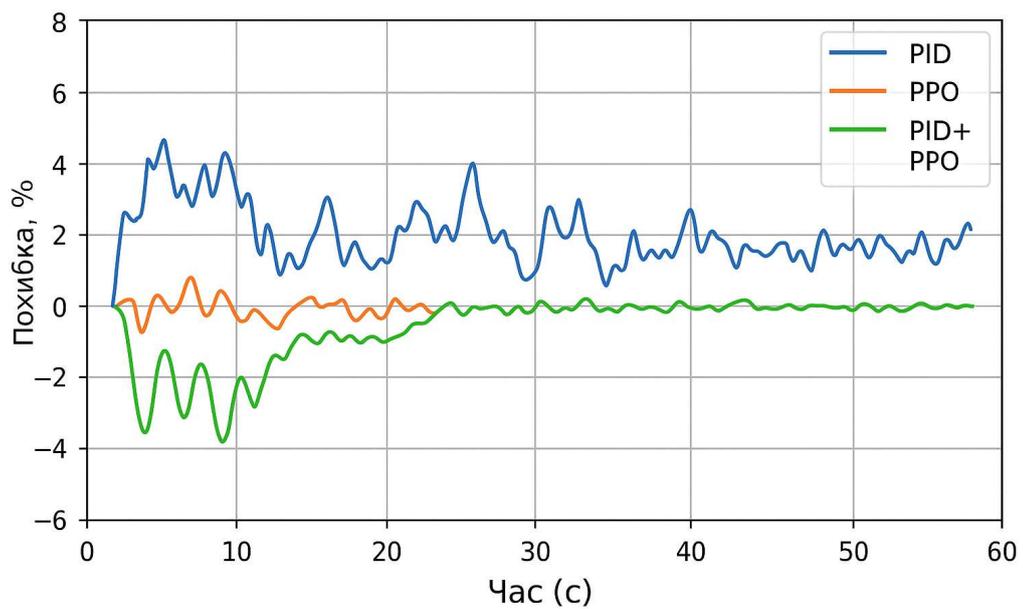


Рисунок 4 – Утримання цілі в кадрі при використанні різних моделей ШП

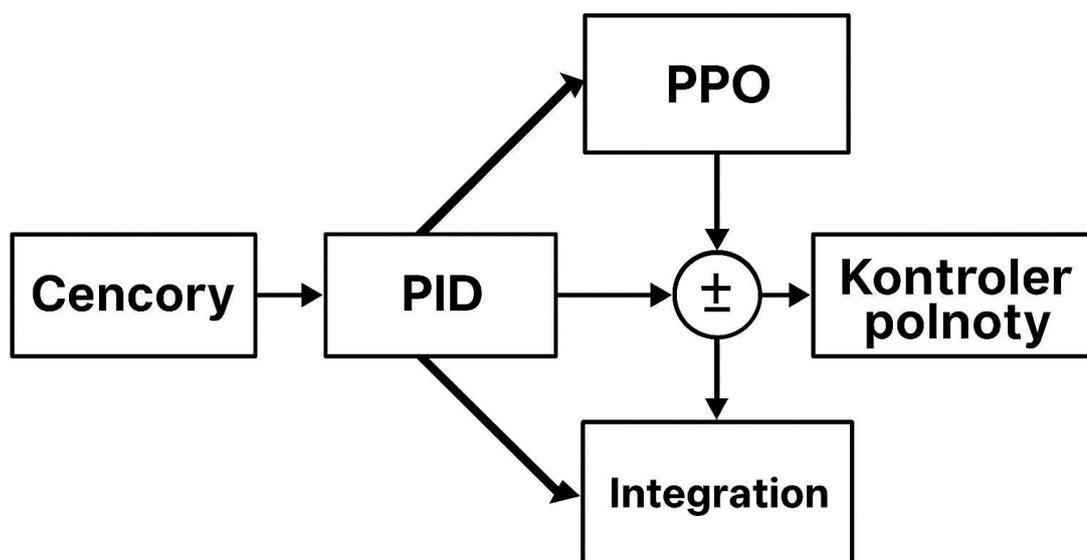


Рисунок 5 – Архітектура гібридного контролера PID + PPO

Таблиця 1 – Порівняльні результати системи на PID регуляторі та гібридної системи

Параметр	PID	PID + PPO
Середня похибка стабілізації	0.26 м	0.14 м
Час реакції на збурення	2.1 с	1.2 с
Відхилення кута (φ , θ)	$\pm 7.4^\circ$	$\pm 3.1^\circ$
Втрата цілі	8%	<2%
Енергоспоживання	100%	92%

Таблиця 2 – Параметри навчання

Параметр	Значення
Алгоритм	PPO
Кількість кроків	2 000 000
Розмір батчу	64

Продовження таблиці 2

Параметр	Значення
Learning rate	3e-4
Discount factor γ	0.99
Clip range	0.2
Entropy coef	0.01
Reward scaling	1.0
Обмеження дії	[-1, 1]

Таблиця 3 – Основні результати навчання

Метрика	Початкове значення	Після навчання
Reward (avg)	20–40	340
Точність утримання цілі	45%	95.2%
Середня похибка стабілізації	0.8 м	0.18 м
Кількість втрат цілі	12 за епізод	<1 за епізод

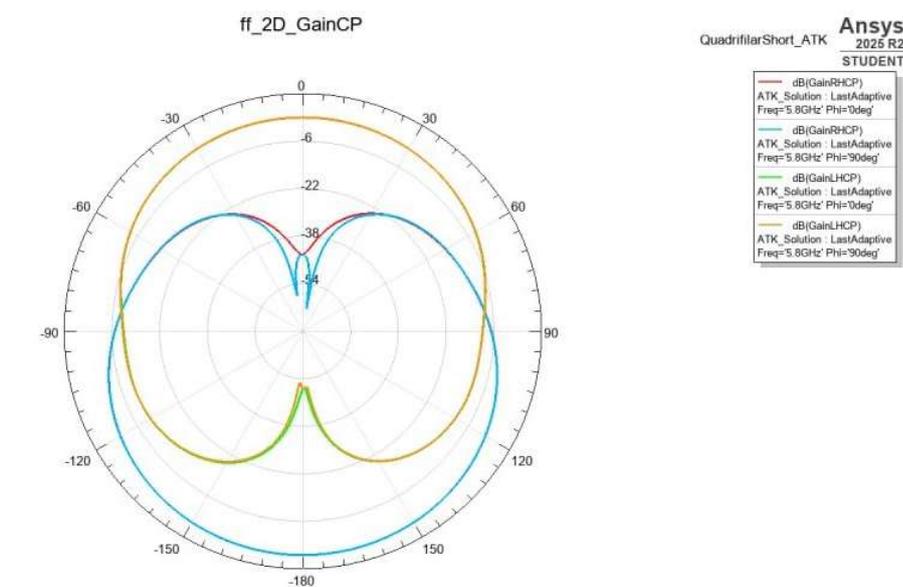


Рисунок 6 – Полярна діаграма для 5.8 ГГц

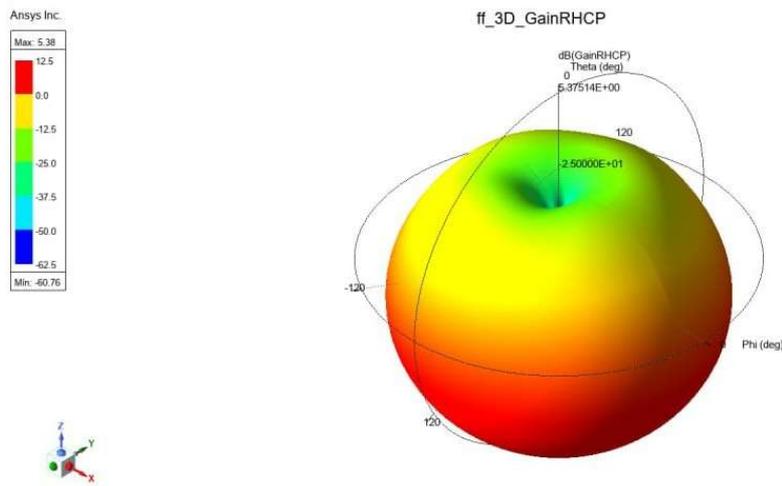


Рисунок 7 – Тривимірна діаграма спрямованості антени

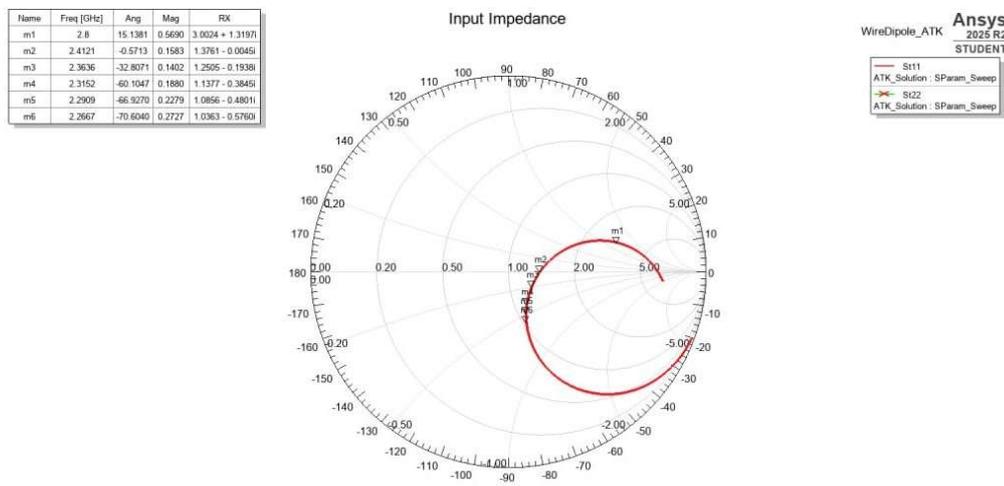


Рисунок 8 – Діаграма Смітта антени 5.8 ГГц

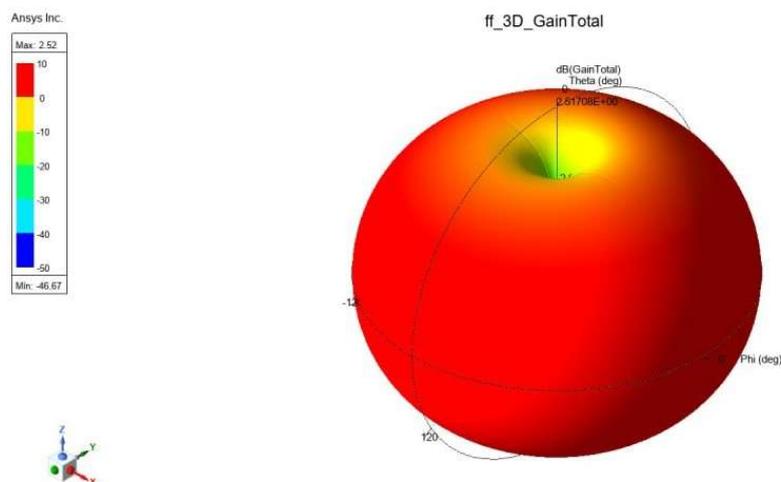


Рисунок 9– 3D діаграма спрямованості

Додаток Б
(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ РОБОТИ
ЗАСОБИ РАДІОКЕРУВАННЯ ДРОНАМИ З ВИКОРИСТАННЯМ
ШТУЧНОГО ІНТЕЛЕКТУ

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: «Засоби радіокерування дронами з використанням штучного інтелекту»

Тип роботи: Магістерська кваліфікаційна робота
 (бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ Кафедра ІРТС
 (кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 1,26%

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

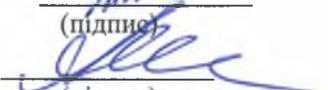
- Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Семенов А.О. – д.т.н., професор каф. ІРТС
 (прізвище, ініціали, посада)


 (підпис)

Осадчук О.В. – д.т.н., професор, зав. каф. ІРТС
 (прізвище, ініціали, посада)

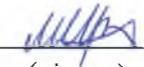

 (підпис)

Особа, відповідальна за перевірку 
 (підпис)

Семенов А.О.
 (прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник 
 (підпис) Семенов А.О. – д.т.н., професор каф. ІРТС
 (прізвище, ініціали, посада)

Здобувач 
 (підпис) Гладкий М.О.
 (прізвище, ініціали)