МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютеризованих систем управління

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри

_____
"_____"_____202_ р.

КВАЛІФІКАЦІЙНА РОБОТА

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: " Програмний модуль системи стабілізації польоту квадрокоптера з використанням регуляторів на базі нейронних мереж"

Виконавець: _____КЛОПЕНКО Роман Юрійович

Керівник: _____СУПРУН Ольга Миколаївна

Нормоконтролер: _____ТУПОТА Євгеній Вікторович

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук та технологій

Кафедра комп'ютеризованих систем управління

Спеціальність 123 «Комп'ютерна інженерія»

(шифр, найменування)

Освітньо-професійна програма «Системне програмування»

Форма навчання денна

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____

«_____»_____202 __ р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

_____Клопенка Романа Юрійовича_____

(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема дипломної роботи (проєкту):« Програмний модуль системи стабілізації польоту квадрокоптера з використанням регуляторів на базі нейронних мереж»

затверджена наказом ректора від «28» серпня 2023 р. № 1494/ст

2. Термін виконання роботи (проєкту): з 02.10.2023 р. по 31.12.2023 р.

3. Вихідні дані до роботи (проєкту): Програмний модуль системи стабілізації польоту квадрокоптера з використанням регуляторів на базі нейронних мереж

4. Перелік обов'язкового графічного (ілюстративного) матеріалу:

1. Основні маневри (зліва направо): рух по прямій, крен/тангаж та рискання.
2. Креслення з’єднання гіроскопа і акселерометра GY-521 *MPU-6050* до Arduino UNO.
3. Блок-схема рішення проблеми стабілізації БПЛА.
4. Порівняння методу навчання Deep-Q-Network та стадартного DQN.

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1.	Створення та узгодження графіку роботи дипломного проектування Написання 1 розділу, обговорення з керівником	02.10.23 – 04.10.23	
2.	Завчасний друк 1 розділу та додаткових сторінок (чернетки) - титульної, завдання, графік, реферат, зміст, вступ, 1-й нормо-контроль.	05.10.23 – 07.10.23	
3.	Написання 2 розділу, обговорення з керівником	08.10.23 – 15.10.23	
4.	Написання 3 розділу, обговорення з керівником	16.10.23 – 24.10.23	
5.	Написання 4 розділу, обговорення з керівником	25.10.23 – 05.11.23	
6.	Закінчення редагування та друк пояснювальної записки	06.11.23 – 15.11.23	
7.	Проходження нормо-контролю, перевірка на антиплагіат, перепліт пояснювальної записки.	16.11.23 – 20.11.23	

8.	Створення тексту доповіді. Представлення графічного матеріалу для презентації	21.11.23 – 25.11.23	
9.	Отримання відгуку керівника, рецензії.	26.11.23 – 30.11.23	
10.	Підготовка матеріалів для передачі секретарю ДЕК в папці	01.12.23	

7. Дата видачі завдання: “02" жовтня 2023 р.

Керівник дипломної роботи (проєкту) _____ СУПРУН Ольга Миколаївна

(підпис керівника) (П.І.Б.)

Завдання прийняв до виконання _____ КЛОПЕНКО Роман Юрійович

(підпис здобувача вищої освіти) (П.І.Б.)

РЕФЕРАТ

Дипломний проект: 66 с., 26 рис., 21 джерело.

Мета дипломної роботи: розробка системи стабілізації квадрокоптера, використовуючи датчики положення БЛА (акселерометр, гіроскоп).

Створена варіація квадрокоптера має в основі таку конструкцію: рами перехрестного типу (X), мотори A2212/13T на 1000 kV, двухлопастні різноспрямовані пропелери, ESC регулятори 30A DC:5.5V-12.6VBEC 5/2A, літій – полімерної батереї (LiPo).

Проект був створений на апаратній платформі Arduino Uno.

В основі квадрокоптеру міститься мікросхема *MPU-6050*, в основі якої 3-х осьовий модуль гіроскопа та акселерометра *GY-521*.

Реалізація коду стабілізації була створена у середовищі програмування *Arduino Software (IDE)*.

Створена та втілена ефективна система стабілізації польоту квадрокоптера.

КВАДРОКОПТЕР, СИСТЕМА СТАБІЛІЗАЦІЇ ПОЛЬОТУ, *ARDUINO UNO*, ФІЗИЧНА МОДЕЛЬ ПРОГРАМУВАННЯ

ЗМІСТ

ВСТУП

Квадрокоптером називають безпілотні літальні апарати (БПЛА), що оснащені чотирма гвинтами, а також це один із різновидів мультикоптера. Основою БПЛА є платформа з чотирма роторами, перша пара яких обертається за годинниковою стрілко, а інша проти. Якщо порівнювати квадрокоптер та вертольотний тип літадьних апаратів з кріпильними гвинтами, то можна побачити, що такі БПЛА мають ряд переваг – простоту конструкції, більшу стабільність та більший кут польоту, відносно невелику масу при підйомі, а також вищу швидкість польоту.

В наш час всі прекрасно розуміють середовище використання БПЛА. Якщо раніше квадрокоптери в більшості своїй слугували як ефективний спосіб отримання відео та фото з повітря, то зараз вони використовуються як бойові літальні машини у війні проти так званої «росії». Десятки підрозділів використовують квадрокоптери для знищення піхоти, техніки та ін.

Квадрокоптер, оснащений камерою з тепловізором може побачити ворога на дуже великій відстані, доставляти медикаменти, припаси, а іноді навіть і зброю, тому в цій війні такі БПЛА стали дуже ефективними та корисними для наших захисників.

Проте, завжди основною проблемою коптерів була їх нестабільність від зовнішніх факторів, з якими стикається БПЛА під час виконання своєї роботи.

Метою дипломної роботи єрозробка системи стабілізації польоту квалрокоптера, використовуючи при цьому нейронні мережі та датчики положення БПЛА акселерометр та гіроскоп.

Для того, щоб виконати поставлену задачу, необхідно виконати наступні пункти :

- Аналіз конструкцій квадрокоптера;
- Вибір апаратної платформи;
- Вибірсередовища для програмування;
- Розробка алгоритму програми стабілізації квадрокоптера;
- Реалізація програмного забезпечення;
- Створення та тестування фізичного макету квадрокоптера.

Об'єкт дослідження: безпілотний літальний апарат – квадрокоптер.

АНАЛІЗ КОНСТРУКЦІЙ КВАДРОКОПТЕРІВ

Квадрокоптери, коптери, мультикоптери – це все безпілотні літальні апарати, які можна використовувати для розвах, зйомки фото та відео, або ж визначення позицій ворога.

Зазвичай БПЛА класифікують між собою за кількістю двигунів – на самому початку розташований бікоптер – він має два двигуни, в фінальній ланці стоїть октакоптер – вісім двигунів. Класикою ж вважається квадрокоптер з 4 двигунами, які розміщені на перехрестних променях. Таке розміщення в 1920 спробував втілити в реальність француз Етьєн Омишен (Étienne Oehmichen), і в 1922 році у нього це навіть вдалося. Такий варіант літального апарату є найпростішим та дешевшим, а також функціональнішим – такий апарат без проблем може підняти в повітря маленьку камеру, таку як GoPro. Якщо ж необхідно злетіти з більш серйозною відеотехнікою, варто обрати коптер з білшим числом двигунів – це не лише вплине на вантажопідйомність, але і додасть декілька балів до надійності, якщо під час польоту якийсь із двигунів вийде з ладу.

1.1 Теорія польоту

В аеродинаміці (вона ж теорія польоту) за основу виділяють 3 кути (вісь) обертання, які керують орієнтацією та напрямом вектора руху квадрокоптера. Їх в народі називають «крен», «тангаж» та «рискання». Крен (Roll) – це поворот БПЛА навколо його так званого Y (вісь, яка проходить від носа до хвоста). Тангаж (Pitch) – поворот навколо «талії» (кут носу, задирання хвоста). Рискання (Yaw) — поворот навколо вертикальної осі (рис.1.1), найбільше схожий на поворот в «наземному» розумінні. Та загальний газ (Throttle).

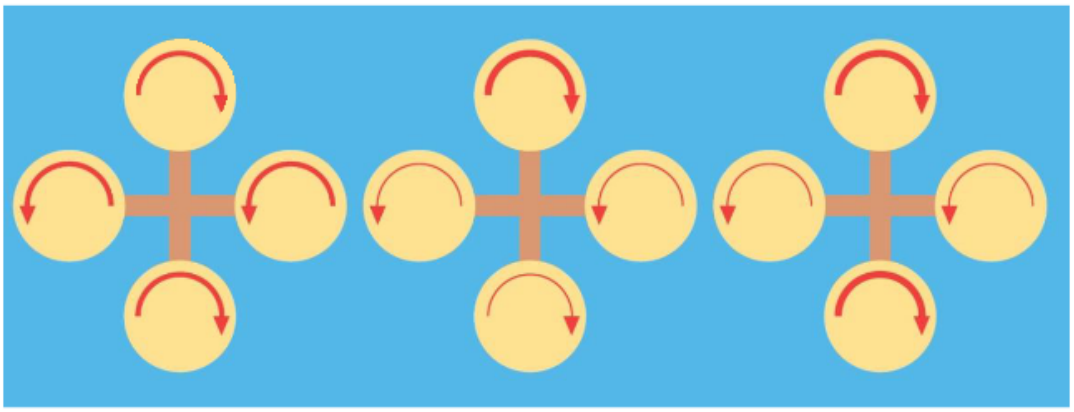


Рисунок 1.1 Основні маневри (зліва направо): рух по прямій, крен/тангаж та рискання

В стандартній схемі квадрокоптера головний гвинт під дією автомата перекосу двигунів керує креном та тангажем. Виходячи з того що гвинт має ненульовий опір наколишнього середовища, у гвинтокрила виникає момент оберту, який спрямовано в сторону, протилежну обертанню гвинта і для його компенсації існує хвостовий гвинт. Регулючи продуктивність цього гвинта, стандартний вертоліт керує своїм поворотом навколо вертикальної осі. З квадрокоптером ситуація інакша. В нього присутні 4 гвинти, 2 з яких обертаються за часовою стрілкою, 2 інші – проти неї. В переважній кількості конфігурацій присутні гвинти з незмінним креном і керуються такі лише їх обертами. В тому випадку, коли вони всі працюватимуть із сталою для всіх чотирьох швидкістю, то

вони будуть компенсувати один одного та всі три основні маневри будуть також однаковими.

Під час підвищення обертів одного гвинта, який обертається за годинниковою стрілкою та спаданні обертів іншого гвинта в якого напрям оберту такий же, існуватиме зберігання спільного моменту: обертання і рискання, як і раніше, буде нульовим, але крен або тангаж (в залежності від того, де був розміщений його «ніс») будуть іншими. Але при підвищенні обертів на обох гвинтах, які обертаються за однією стрілкою, а в той час на гвинтах, які обертаються в іншому напрямку, зменшити момент оберту (щоб зберегти загальну підйомну силу), то виникне обертаючий момент, який змінить кут рискання. Цю всю інформацію передаватиме та оброблятиме бортовий комп'ютер, що буде отримувати сигнали з пульта керування, регулювати дані з акселерометра та гіроскопа і обертати гвинти в необхідному напрямку. Для проектування коптеру потрібно знайти баланс між вагою, годинами польоту, силою двигунів та іншими характеристиками. Все залежить від необхідного завдання. Основна мета полягає в тому, щоб БПЛА літав вище, швидше та довше, проте середній час літання матиме від 10 до 20 хвилин в залежності від ємності акумулятора і фактичної ваги польоту. Все це пов'язано між собою, для прикладу, підвищення ємності акумулятора вплине на збільшення ваги і, як результат, до меншого часу польоту.

1.2 Рама

Головну проблему, яку необхідно вирішити під час обирання рами – це використовувати готову чи створити самодільну. Якщо вибір пав на готову раму, все стає на порядок легше. Але, свою раму, у разі поломки,

можна полагодити набагато легше. Своїми руками можна створити будь-яку конструкцію.

Самий легкий варіант – це скористатися квадратною алюмінієвою трубою 12 х 12 та алюмінієвим листом в 1,5мм. Із цього всього можливо зробити раму потрібного нам типу за допомогою ножівки або дрелі. Проблема в тому, що така конструкція, в більшості своїй, живе недовго. Причиною тому є м'який матеріал (АД31-33), при польоті його дуже легко зігнути

Для зразку рами можна взяти спрощену раму з заводу або переглянути в інтернеті готові креслення. Якщо ж наявні складні матеріали (вуглепластик), то такі можна замінити на той же алюміній – навіть якщо буде важче, то не настільки критично. Головне звернути увагу на довжину та симетричність променів. Їх довжина обирається в залежності від діаметра пропелерів таким чином, щоб після їх монтажу відстань між колом обертових гвинтів була не менша 1-2 см, і вони не мають перетинатись. Двигуни, які монтовано на променях, необхідно рівновіддалети від центру рами, де буде розміщений «центр керування» та, в більшості своїй, бути на однаковій відстані один від одного, створюючи при цьому рівносторонній багатокутник. При плануванні варто урахувати, що центр рами повинен співпасти з центром ваги.

За розміщенням двигунів, відносно напрямку польоту, виділяють два основних типи рам: «+» та «Х». Однією з категорій рами типу «Х» виділяють раму типу «Н» (рис.1.2). Найпопулярніші рами типу «Х». Основною перевагою такої рами є комфортне розміщення камери, коли промені коптера не з’являються в кадрі. У рами класу «Х» найвища стійкість до незначних аварій.

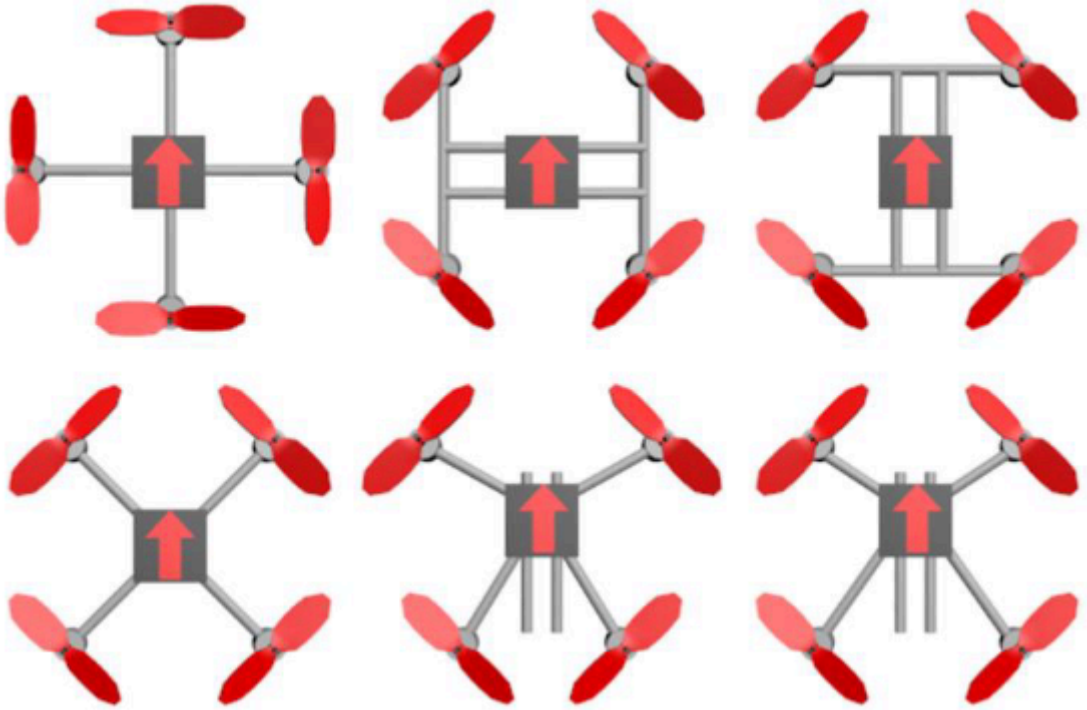


Рисунок 1.2 – Класифікація рам квадрокоптерів ("+", X та H

типу)

У випадку «+» класу рами частіше за все удар приймає один промінь, який дуже страждає, в той час як при розподілу удару на два промені, пошкодження зазвичай обмежуються зламаними пропелерами. Окрім того, великій кількості «літунів» на психологічному рівні легше керувати типом «X» (рис. 1.3). Проте, у квадрокоптера з рамою типу «+» швидша та «гостріша» реакція на команди "вправо-вліво" і "вперед-назад", тому такий тип більш коректно використовувати для «швидкого» пілотування.

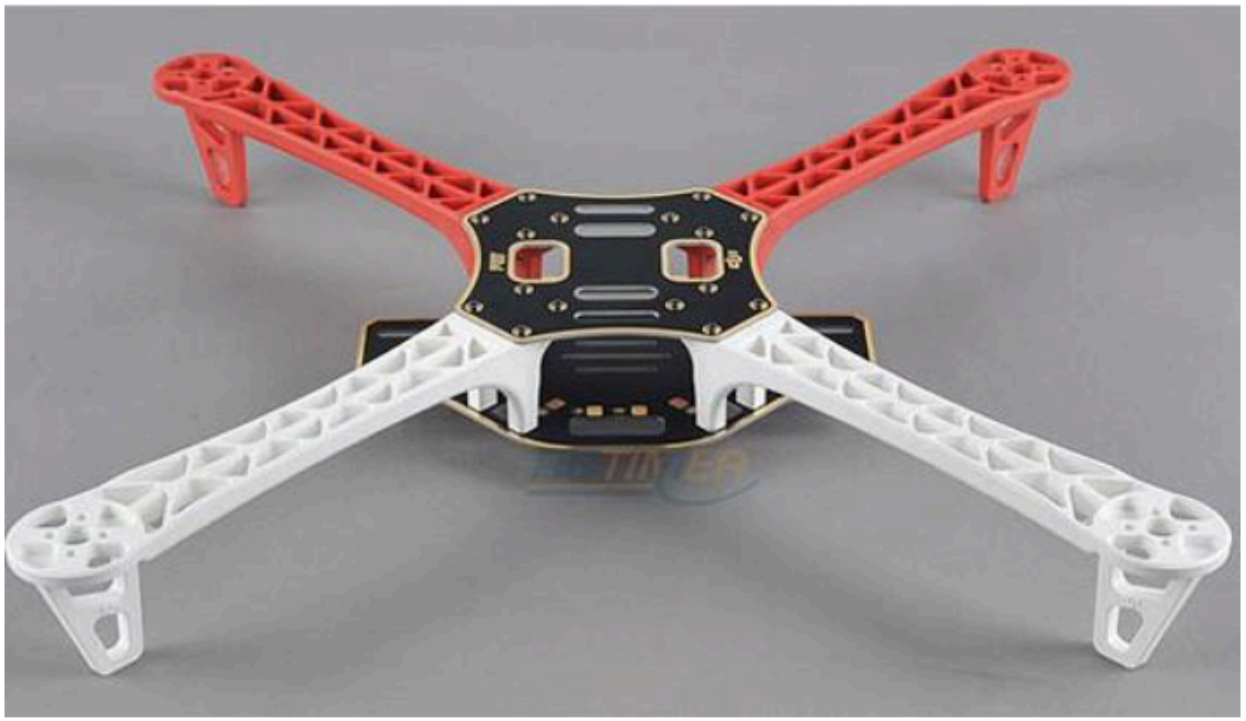


Рисунок 1.3 – Рама квадрокоптера X-типу

Раму типу «H» комфортно використовувати під час побудови дуже маленьких квадрокоптерів, а також для повнорозмірних конструкцій з карбонових трубок та зазвичай використовується під час конструювання спеціальних коптерів для зйомки та польотів по GPS, вся справа в кількості місця для обладнання.

Проаналізувавши всі переваги та недоліки, для квадрокоптера було обрано раму типу «X», зважаючи на її стабільність та міцність.

1.3 Двигуни

У кожного з пропелерів є 2 основні параметри: діаметр і крок. Вони можуть позначатись абсолютно по різному: 12 × 6.5, 12 × 65 або лише 1265. Це можна розшифрувати як діаметр пропелера 12 дюймів, а його крок 6,5 дюймів. Чим довший пропелер і більший крок, тим сильнішу тягу можна створювати, проте і навантаження на двигуни збільшиться,

як і підвищиться споживання енергії, а як наслідок він може сильно перегрітись та електрика «перегорить». Тому гвинти варто підібрати саме під двигун. Є, звісно ж, і пропелери із змінним кроком, що, в теорії, мають вищу маневреність, але по факту це лише додача складних механік, які матимуть властивості зношуватись та ламатись, а ремонт таких деталей коштує дорожче.

Безколекторні двигуни (рис.1.4). У будь-якого двигуна інсує головний параметр – kV. Простими словами це кількість обертів на хвилину,які виконає двигун, на поданий вольт напруги. Це не потужність самого двигуна, а його передатне число. Чим менше kV, тим менше обертів, але вищий крутний момент. Рівнопропорційно, якщо більше kV під час такої ж потужності, тим вища кількість обертів і менший момент. Під час вибору двигуна звертаються увагу на те, що в стандартному режимі він буде працювати в потужність 50% від максимально допустимої. Помилково вважати, що чим kV вище – тим краще, для квадрокоптеру з стандартною 3S- батареєю прийнятне число міститься в границях між 700 до 1000 kV.

Рисунок 1.4 – Двигун A2212 /13T на 1000 kV

Для квадрокоптеру взято двигуни на 1000 kV. Проаналізувавши потужність, вони є найбільш сприятливі для поставленої цілі.

1.4 Пропелери

Через крутний момент двигуна в різні боки постає необхідність використання різноспрямованих пропелерів: зворотнього оберту (за годинниковою стрілкою) та прямого (проти неї). В більшості випадків, обираються двухлопастні пропелери (рис. 1.5), вони простіше балансуються та їх простіше дістати, в той час як трьохлопатеві дають більшу тягу при менших діаметрах гвинта, але з ними дуже багато

мороки під час балансування. Поганий (без балансу, відносно недорогий) пропелер зазвичай розвалюється в польоті або створює надто сильні вібрації, що в совю чергу отримує датчик польотного контролера. Зазвичай такі пропелери призводять до тяжких проблем із стабілізацією та призведе до сильного змазування картинки під час польоту.

Виходячи з цього всього було обрано різноспрямовані двухлопастні пропелери, через їх легше балансування та стабілізацію.

Рисунок 1.5 – Двухлопастні пропелери

1.5 ESC регулятори

Проте, двигуни підключені до акумулятору не на «пряму», а через так звані регулятори швидкості (рис. 1.6). Регулятори швидкості (вони ж ESC) регулюють швидкість обертання двигунів, примушуючи коптер балансувати в одній позиції або направляти політ в потрібному

напрямку. Велика кількість регуляторів оснащені вбудованими регуляторами струму на 5 В, від яких можна підживлювати електрику (наприклад, «мозок»). Обираються контролери швидкості, зважаючи на використання двигуном струму, а також можливість перепрошивки. Стандартні регулятори зазвичай повільні в своїх відгуках, на які отримується сигнал та містять велику кількість непотрібних налаштувать для квадрокоптеру.

Двигуни під’єднано до регулятора швидкості трьома проподами, їх черга не має значення, проте при заміні місцями будь-які два з трьох проводів місцями, то двигун буде обертатись в іншому напрямку, а для квадрокоптеру це дуже важливо.

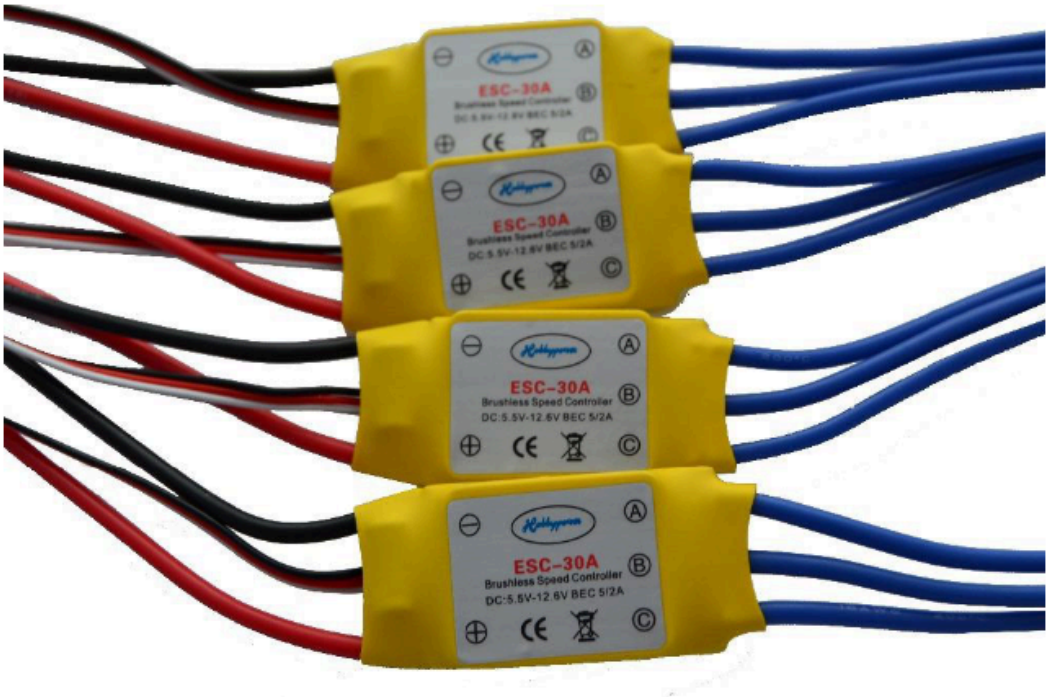


Рисунок 1.6 – ESC регулятори 30 А

1.6 Живлення та його контролери

Під час вибору акумулятору необхідно приділити увагу на декілька параметрів: ємність, що вимірюється в міліампер-годинах, максимальний струм розряду в ємностях акумулятора (C) і число клітинок (S). Якщо двигун має високу потужність, відповідно більший акумулятор йому необхідний. Проте, якщо акумулятор буде більший, то відповідно зросте і його вага, а це в свою чергу змусить коригувати вибір гвинтів та двигунів. В наш час високу популярність отримали літій-полімерні батареї (LiPo). Такі батереї легкі, мають велику місткість та високий струм розряду. Проте, не все так гладко, вони мають і мінуси – при понижених температурах вони працюють погано, тому їх потрібно тримати в теплі (кишеня наприклад) і з'єднувати прямо перед польотом, то під час розряджання вони самі будуть нагріватись і не будуть мерзнути.

Виходячи з всього сказаного, в магістерський роботі вибрано конструкція, яка складається з:

- Рами Х-типу;
- Моторів A2212 /13T на 1000 kV;
- Різноспрямованних двухлопастнних пропелерів;
- ESC регуляторів 30 A DC :5.5V-12.6VBEC 5/2A;
- Літій-полімерної батареї (LiPo).

Основними перевагами такої конструкції є простота управління та універсальність деталей. Такий вибір складових коптера враховує поставлену задачу роботи. Після чого можна концентруватись на задачах диплому.

УПРАВЛІННЯ КВАДРОКОПТЕРОМ

2.1 Апаратне забезпечення

2.1.1 Версії платформи Arduino

Arduino Uno R3

У контролер перед використанням прошивають завантажувач Boot-Loader, виходячи з цього, зовнішнє програмне забезпечення непотрібне. Прилад програмується за допомогою USB без використання сторонніх утиліт. Є декілька версій платформи Arduino. Версія Leonardo побудована на мікроконтролері ATmega32u4. Uno, Nano, Duemilanove базується на мікроконтролері Atmel ATmega328. Більш ранні варіанти платформи Diecimila і перша діюча Duemilanoves були створені на основі Atmel ATmega168. Arduino Mega2560, в свою чергу, побудовано на мікроконтролері ATmega2560. А найпізніша Arduino Due – на базі мікропроцесора Cortex.

Версія Arduino Uno – одна з найпопулярніших та має великий попит для невеликих проектів. Така версія універсальна, у зв’язку з тим, що входи розпаяно однорядними конекторами типу «мама». Зазвичай, таку плату обирають для прототипу проекту, а створення дійсного пристрою відбувається на менших платах Arduino, наприклад Arduino Nano і Arduino Pro. Це не становить великих труднощів хоча б тому, що прошивки сумісні, і в переважній більшості випадків, номери контактів ідентичні. Для Arduino Uno створено велика кількість плат розширення, наприклад Ethernet shield, motor shield, servo shield та інші.

Arduino Uno створена в багатьох версіях, таких як R1, R2, R3. Всі версії ілентичні одна одній по своєму функціоналу, тому для моєї роботи було обрано версію Aruino Uno R3 (рис.2.1). В ній міститься два методи рішення: DIP I SMD. Їх різниця в тому, що мікроконтролер може виконувати DIP виконання і вставлений в колодку, або лише розпаяний на плті, якщо це версія CMS.

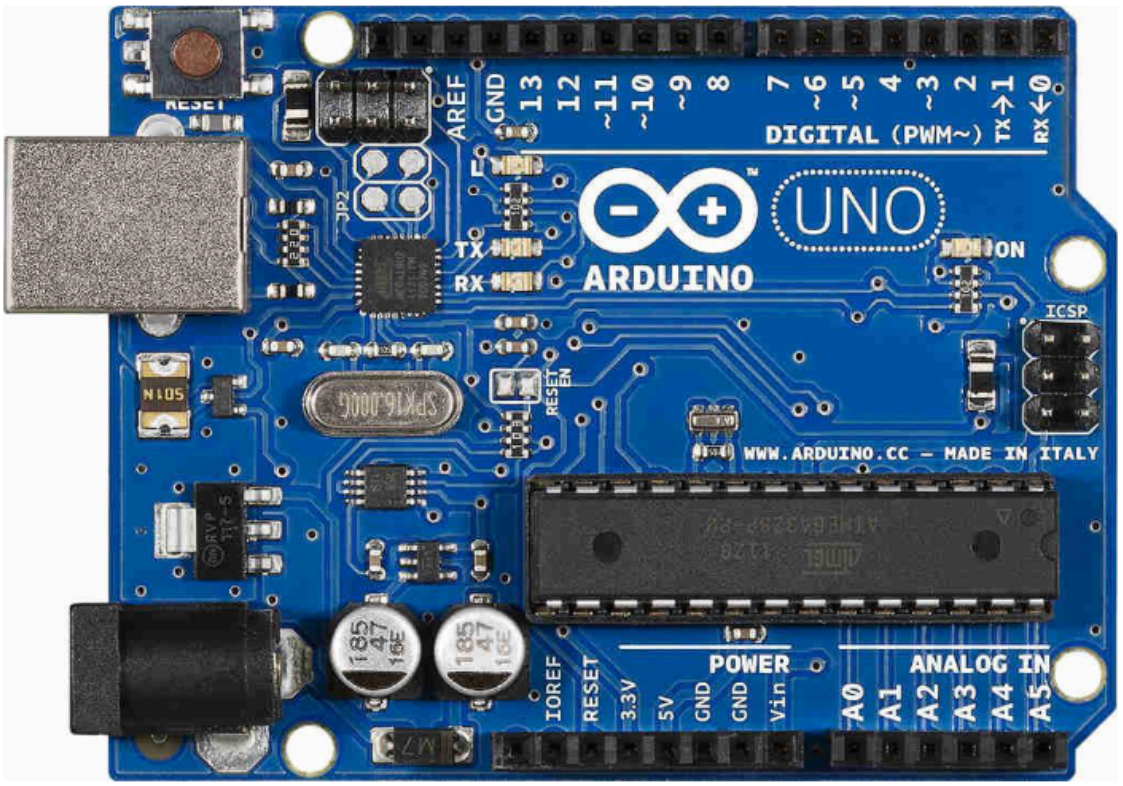


Рис. 2.1 – Arduino Uno R3

Такий мікроконтролер містить чіп ATmega328, в якого тактова частота 16 Мгц. В самій платі міститься: порт USB, роз'єм живлення, кварцовий резонатор, стабілізатори напруги 5 В і 3.3 В, світлодіоди та кнопку "перезавантаження".

Характеристики Arduino UNO:

1) Мікроконтролер: ATmega328.

2) Діапазон допустимої напруги: 5-20 В.

3) Рекомендована напруга живлення: від 7 до 12 В.

4) Кількість цифрових виводів: 14.

5) ШІМ: 6 цифрових пінів можуть бути використані в якості виводів ШІМ.

6) Кількість аналогових виходів: 8.

7) Максимальний струм: 40 мА/г з одного 500 мА/г з усіх виводів.

8) Флеш-пам'ять: 32 Кб.

9) SRAM: 2 Кб.

10) EEPROM: 1 Кб.

11) Тактова частота: 16 МГ

Таку плату можна заживити в чотири способи:

1) USB – порт. В такий спосіб плата живиться з ПК, PowerBank, смартфон (за умови, що такий підтримує режим OTG), а також через адаптер змінного струму.
2) Пін +5В. Такий пін одночасно є піном вводу та виводу. Проте, є й обмеження – на такий пін необхідно подавати саме 5 вольт, в іншому випадку він згорить
3) Через живлення на платі. Можна користуватись батарейками, акумулятори та блоки живлення. Такий роз’єм підключається до піну VIN.
4) Через пін VIN. Струм через такий пін протікає через вбудований стабілізатор напруги. Залежно від виробника інсує можливість дати від 5 до 20 В. Стабілізатор не має ККД 100% при використанні 5 В для піна VIN, напруги живлення може бути не

достатньо для живлення контролеру, а на цифрових виходах буде менше 5 В. Задля безпеки не треба працювати в максимальну напругу. При таків напрузі стабілізатор буде нагріватись, поки не вийде із ладу. Тому, рекомендується використовувати напругу в межах 7 – 12 В.

На платі присутні 14 цифрових пінів. На друкованій платі їх позначено буквою “D” (Digital). Вони двухсторонні – вхідні та вихідні. Робоча напруга таких контактів 5 В. Будь-який з них має підтягуючий резистор і напруга,яку подають на ці піни нижча 5 В, проте вона все одно буде вважатися як логічна одиниця (5 В).

Аналогові піни позначаються ведучою «А». Такі піни зазвичай є входами і не містять в собі підтягуючі резистори. Вони слугують для вимірювання напруги, яку отримують і повертають значення в діапазоні від 0 до 1024. Такі піни вимірюють напругу з точністю до 0,005 В.

ШІМ Arduino Uno

Якщо детально подивитись на плату, то при перегляді можна помітити значок тільди (~) поряд з деякими цифровими пінами. Такий значок показує, що пін може використовуватись як вихід ШІМ. На певних платах Arduino даний значок відсутній, через те, що виробники можуть не знайти місце для символу на платі. На платі Arduino Uno є 6 виводів ШІМ, це піни D3, D5, D6, D9, D10 і D11. Для використання ШІМ в Arduino є спеціальна функція analogWrite().

Інші піни:

- rx0 і tx1 створені для передачі даних по

послідовному інтерфейсу;

- вивіди D10 (SS), D11 (MOSI), D12 (MISO), D13 (SCK)

призначені для зв'язку по інтерфейсу SPI;

- на виводі D13 є вбудований в плату світлодіод;

- A4 (SDA) та A5 (SCL) існують для зв'язку

з іншими пристроями по шині I2C.

В Arduino Uno містяться запобіжники, які відновлюються та захищають USB-порт комп'ютера від перевантажень та коротких замикань.

Не двилячись на те, що в великої кількості комп’ютерів присутній захист, такі запобіжники це дуже великий плюс. Коли трапляється так, що від USB-порту використовується струб більший за 500мА, запобіжник розриває з’єднання до усунення причини замикання.

Arduino Uno за розмірами наступний – 69мм довжини, 53мм ширини, проте живлення та USB випирають за межі друкованої плати. Arduino Uno за вагою приблизно 25 грам. Вона має 4 отвори для можливості її кріплення. Відстань між виводами – 2,5 мм, окрім сьомого та восьмого виводів – між ними чотири міліметри.

У висновку, найкращим рішенням функціонування-габарити буде Arduino Uno R3. Її характеристики гарно підходять для даного завдання.

2.1.2 Мікросхема MPU-6050

Мікросхема MPU-6050 містить патентований компанією InvenSense процесор оброблення сигналу, які викликаються рухом Digital Motion Processor (DMP), який зданий взаємодіяти з алгоритмами MotionFusion. DMP використовується також для важких розрахувань. Особистий процесор може робити розрахунки, не потревоживши контролер та

навіть здатний обробити інформацію іншого датчика, який підключений до іншої шини I2C. Особлива програма мовою команд DMP пише в пам’ять щоразу піся подання струму. По часу це займає трішки менше секунди. ПЗ фільтрує значення акселерометра та гіроскопа. Далі інформація передається в буфер FIFO. Напруга живлення 2,3 – 3,6 В. Номінальна 3,3 В.

Для більш точного слідкування за рухами існує рішення запису в пам'ять MPU-6050 поточних даних вимірювання. Їх можна зчитувати з реєстрів збереження або буферів FIFO в розмірі 1024 байти. Мікросхема MPU-6050 також функціонує в режимі майстер на шині I2C для контактів XDA і XCL. Всередині неї розміщено АЦП в 16 біт. Присутній регістр з ім’ям Who am I, він містить в собі адресу модуля GY-521 на шині I2C. Значення в регістрі 104 десяткове або 68 шістнадцяткове. Мікросхема MPU-6050 має в собі більше 100 регістрів.

Посилання мікросхеми буває двох значень в залежності від стану виводу AD0 модуля трьохосьового гіроскопа і акселерометра GY-521 MPU-6050: 68 (16), коли AD0 з'єднано із спільним шнуром, 69 (16), якщо AD0 з'єднано з потенціалом лог. 1.

Акселерометр мікросхеми MPU-6050 взаємодіє за допомогою п'єзоелектричного ефекту.

Гіроскоп представляє собою рухому коливанням п'єзоелектричну пластину. Під час повороту пластина піддається викривленню та її параметри електрики змінюються. Всю цю інформацію фіксує мікросхема.

Живлення модуля GY-521 приймається на вході стабілізатора напруги Q2 мікросхеми MIC5205-3.3BM5 з напругою виходу в 3,3 В. На стабілізаторі в цей час фіксується спад напруги 0,3 – 0,4 В, тому через це

напруга живлення модуля повинна бути вищою ніж 3,3 В. Показники живлення модуля 3-х осьового гіроскопа і акселерометра GY-521 MPU-6050 – світлодіод D1. Розміри резисторів R4 і R5 можуть бути інакшими від розмірів, які описані на кресленні.

Призначення контактів:

- VCC – напруга живлення;

- GND – загальний провід;

- SCL – сигнал такту I2C;

- SDA – інформація I2C;

- XDA – інформація шини I2C під час роботи в режимі майстра;

- XCL – тактовий сигнал шини I2C під час роботи в режимі майстра;

- AD0 – біт 0 адреси I2C;

- INT – вихід сигналу про готовність даних для використання як зовнішнього переривання МК;

Підключення модуля 3-х осьового гіроскопа і акселерометра GY-521 MPU-6050 до Arduino UNO (рис.2.2):

- 5 V Arduino – VCC;

- GND Arduino – GND;

24

- A4 Arduino – SDA;

- A5 Arduino – SCL;

- GND Arduino – AD0.

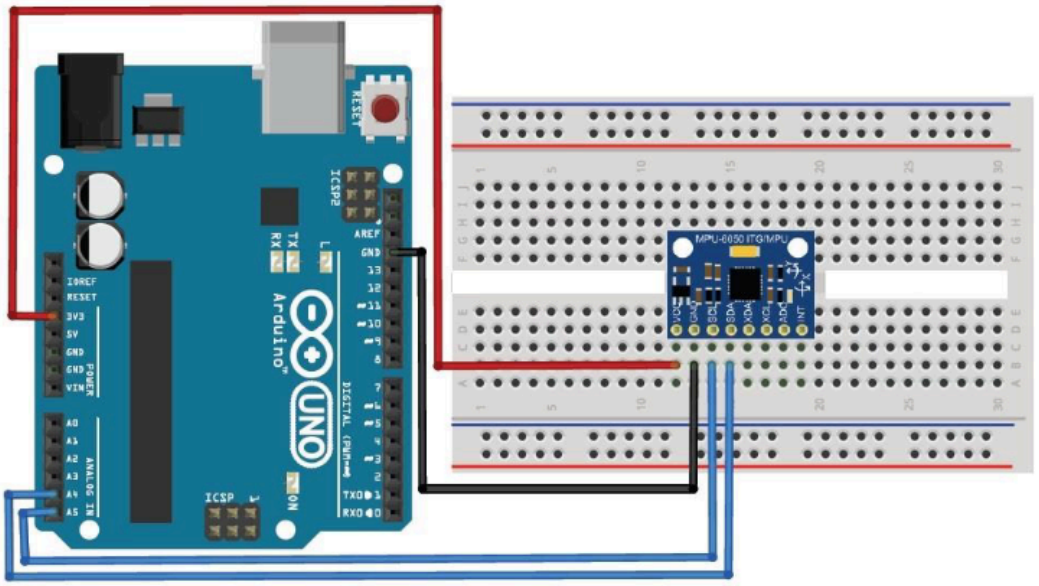


Рисунок 2.2 – Креслення з’єднання гіроскопа і акселерометра GY-521

MPU-6050 до Arduino UNO

Всі модулі GY-521 вимагають калібровки. Показники датчиків зазвичай дещо зміщені відносно нуля, що в свою чергу створює похибку визначення даних. Під час створення мікросхем не можна одержати ідеальну точність кута між осями датчиків, виходячи з цього в кут між ними отримаємо похибку. Спостереження між трьома осями вірогідно матимуть різну пропорційність.

Щоб отримати найбільш точні результати математичних перетворень необхідно, щоб отримані параметри руху було одержано при одночасному зчитуванні даних акселерометра та гіроскопа.

Під час створення машини, якою ми будемо керувати варто уникнути в механізмі коливання та їх гармонік з частотою резонансу, яку зафіксовано в характеристиках.

Під час першого огляду модулю аналізують його роботу на шині I2C. Щоб виконати все правильно, необхідно використати I2C сканер.

За допомогою такого модулю можна отриматі дані про положення та зміщення пристрою в просторі: кут крену, тангаж, використовуючи вектор сили тяжіння та швидкості обертання. Під час переміщення визначати лінійне прискорення та кутову швидкість по 3 осях, в результаті це демонструє повну інформацію про стан.

Фактично, такий модуль є електронним аналогом вестибюлярного апарату людини. Дякуючи йому або такому ж схожому органу живі створіння відчувають напрям тяжіння. Підсвідомо вестибюлярний апарат допомагає нам втримуватись в рівновазі.

Відстеження вектору сили тяжіння допомагає акселерометру використовуватись як електронна заміна будівельним рівням. На його основі створюються кутоміри, про тяжіння землі – інклінометри. Їх використовують для архітектурних споруд, при бурінні копалин.

Стак із гіроскопа і акселерометра використовують для регулювання об’єкта в потрібній позиції при зовнішніх факторах. Для прикладу – зйомка під час їзди на авто, катери чи коптері.

При використанні гіроскопа-акселерометра GY-521 можна жабути про джойстик та будь які механічні прилади в штатних пультах керування.

Основне використання гіроскопа та акселерометра – розміщення його в рухомих напівавтоматичних та автоматичних системах. Пристрій визначає та інформує мікроконтролер про пришвидшення та орієнтацію. Змінні параметри рухомого об’єкту, що визначає модуль 3-х осьового гіроскопа і акселерометра GY-521 MPU-6050: тангаж (ніс вгору і вниз), рискання (ніс вліво і вправо) і крен (за годинниковою стрілкою або проти годинникової стрілки дивлячись з кабіни об'єкта). В системі навігації вони подаються в МК, який визначає поточне положення. Політ коптера без цього модулю не є можливим.

Характеристики:

- максимальна частота інтерфейсу I2C становить 400 кГц;

- формат даних: кути Ейлера, кватерніони, матриця повороту або необроблені дані;

- напруга живлення 3,7 – 5,5 В;

- струм до 10 мА;

- гіроскоп споживає 3,6 мА у режимі очікування 5 мкА;

- акселерометр споживає 350 мкА;

В режимі очікування споживання дорівнює:

- 10 мкА для 1,25 Гц;

- 20 мкА для 5 Гц;

- 60 мкА для 20 Гц;

- 110 мкА для 40 Гц.

В тому випадку, коли пристрій розміщено строго горизонтально і не рухається по проекції пришвидшення сили тяжіння на осі X та Y дорівнює 0. Силу тяжіння вловлюються тільки чутливими механізмами вертикальної осі Z. Через певний проміжок часу у стані спокою відбувається перевірка та калібрування акселерометру. Під час руху пристрій постійно то прискорюється, то сповільнюється. Ідеального рівномірного переміщення не існує. Це і дає можливість використання акселерометру не лише для визначення місцезнаходження об’єкта, але і для визначення змінних параметрів під час руху. Акселерометр фіксує суму пришвидшення під час переміщення та гравітації. В тому випадку, коли всі елементи акселерометра по осях X, Y, Z надають значення

близькі до нуля, то це означає, що двигун вимкнено та об’єкт знаходиться у вільному падінні.

Коли коптер при горизонтальному положенні нахилився, акселерометр фіксує зміну пришвидшення по осях. Під час пришвидшення руху додається пришвидшення тяжіння в другому порядку, не в такому ж, як під час польоту без крену. В кінці система керування прийде до результату про рух в сторону крену, проте насправді відбудеться горизонтальний рух. Для вірної обробки даних руху і правильного розпізнавання динамічних параметрів використовують разом акселерометр та гіроскоп.

Під час використання гіроскопа, той автоматично визначить кут повороту та дасть можливість правильно перетворити дані для акселерометра. Користуватись гіроскопом без акселерометра неможливо через певні нюанси гіроскопа, що накопичують похибку. Спеціальна математика дає можливість об’єднувати роботу з даними від двох датчиків.

Гіроскопи і акселерометри активно застосовують в авіації, ракетній, космічній та військовій техніці. Тому, для магістерської роботи було обрано саме мікросхему MPU-6050 датчиками 3-х осьового гіроскопа і акселерометра GY-521. Для поставленої мети роботи та для вирішення поставлених задач, набір таких модулів вважається достатнім. Але необхідно розширити функціонал квадрокоптера, згодом можна додати до цього датчик барометра (для вимірювання висоти та утримання квадрокоптера на ній) та магнітометра (для утримання напряму руху).Для автономного польоту, можна додати ще GPS приймач.

2.2 Вибір середовища програмування

2.2.1 Arduino Software (IDE)

Програмне забезпечення Arduino (IDE) – має в собі редактор тексту написання коду, перелік повідомлень, консоль, меню та панель інструментів. До апаратних засобів Arduino та Genuino він приєднується для завантаження програм та обміну інформацією з ними.

Програмне забезпечення, яке написано з використанням Arduino (IDE) в народі називають ескізами. Їх пишуть в редакторі тексту та зберігають з розширенням .ino. Редактор обладнаний функціями вирізання/вставлення та пошуку і редагування тексту. Область Повідомлень надає зворотний зв’язок під час збереження та експорту і ще демонструє користувачу помилки. Консоль показує текстову інформацію програми, в тому ж числі і основні помилки. У нжиньому правому куті показана плату,яку ми зараз налаштовуєо та наступний порт. Панель інструментів дає можливість контролювати та інсталювати програми, редагувати, відкривати ескізи та переглядати послідовний монітор. Програма Arduino (IDE) використовує скетчбук – основне місце для збереження програм та ескізів. Ескізи відкриваються за наступним шляхом: Меню - Файл - блокнот або через кнопку "відкрити" на панелі. Під час першого запуску програмного забезпечення Arduino, він буде сам створювати новий скетч за стандартним шляхом розміщення скетчбука.

Також існує можливість керувати не лише одним ескізом (будь-який з них має власну вкладку). Зазвичай це стандартні файли коду Arduino (без видимого розширення), з файлами (.розширення C), C++ файлів (.CPP), або файли бібліотек (.h).

Завантаження ескізу відбувається за допомогою завантажувача Arduino – маленьке програмне забезпечення, що інстальовано на мікроконтролер на цій платі. В свою чергу це дає змогу запускати код без допомоги додаткових апаратних програм. Він активується за декілька секунд після того, як плата перезавантажується та/або скидається. Після чого починається завантаження саме для ескіза, якого завантажили в мікроконтролер. Він мигає стандартним світлодіодом під час запуску.

Програмне забезпечення Arduino (для IDE) має стандартне забезпечення різних плат, які засновані на AVR ядрі. Менеджер, який міститься в базовій установці має змогу підтримувати наростаюче число нових плат, які створено на базі неідентичних одна одній ядер, наприклад Arduino due, Arduino zero, Edison, Galileo і так далі.

Перш ніж починати працювати з програмою варто обрати необхідну нам платформу Arduino (рис. 2.3) та порт.

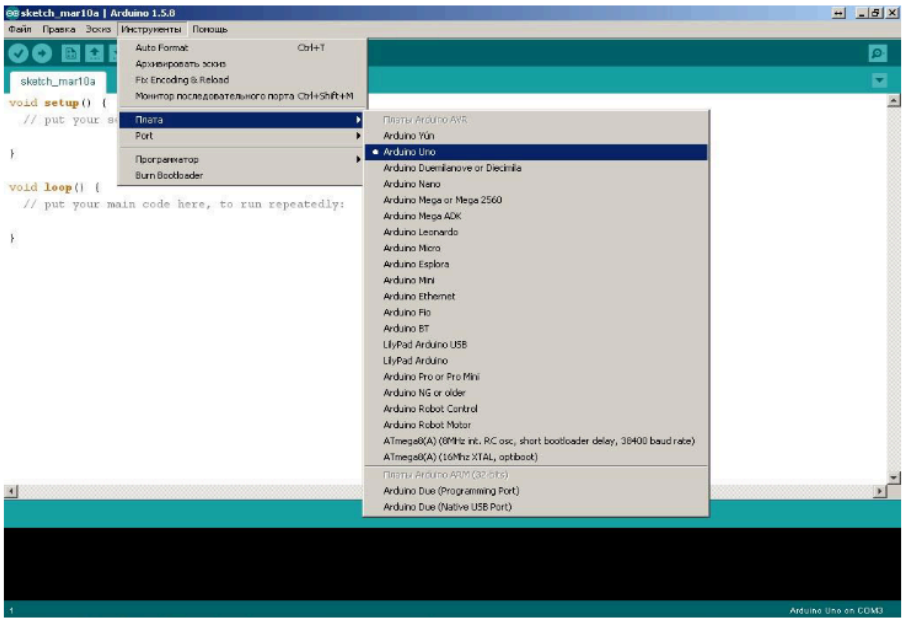


Рисунок 2.3 – Встановлення необхідної плати в налаштуваннях програми Arduino (IDE)

Така версія програми є найбільш підходящою за функціями та універсальністю для поставлених цілей магістерської роботи.

РОЗРОБКА АЛГОРИТМУ ПРОГРАМИ СТАБІЛІЗАЦІЇ КВАДРОКОПТЕРА

3.1 Розробка математичної складової алгоритму

Комп’ютерна програма Arduino дає змогу інсталяції бібліотеки для GY-521. Існує складна база даних I2Cdevlib, що надає доступ до MPU-6050 і інших пристроїв на шині I2C. Її інсталюють в папку MPU6050. За основу вона використовує буфер пристрою мікросхеми та цифрову обробку параметрів руху MPU-6050 задля зміни даних між різнимим системами координат та об’єднує інформацію із декількох датчиків.

Також варто згадати біліотеку FreeIMU, яка створена для інерціально вимірювального блоку , який містить в собі декілька датчиків та може оброблювати велику кількість складних даних. Через універсальність, у роботі обрано бібліотеку I2Cdevlib.

Для прикладу, згенерованою бібліотекою інформацію продемонстровано далі:

MPU-6050

Read accel, temp and gyro, error = 0

accel x,y,z: -123, -180, 14547

gyro x,y,z : -6, -20, 52,

MPU-6050

Read accel, temp and gyro, error = 0

accel x,y,z: -195, -203, 14510

gyro x,y,z : -15, 14, 72,

MPU-6050

Read accel, temp and gyro, error = 0

accel x,y,z: -232, -268, 14490

gyro x,y,z : -4, -7, 45,

MPU-6050

Read accel, temp and gyro, error = 0

accel x,y,z: -189, -170, 14632

gyro x,y,z : -4, -7, 50,

Спочатку програми необхідно відкалібрувати перші 10 даних для того, щоб одержати постійну похибку (зміщення) від датчика. Зміщення змінюється через необроблені значення датчика до зміни даних на кути.

Щоб вирахувати орієнтацію акселерометром, необхідно врахувати , що на коптер існує постійне тяжіння масою в 1 g. В той час, коли нмає супротиву з інших значень, що взаєиодіють на акселерометр, то стала величина пришвидшення матиме значення 1 g.

Звірившись із описом *MPU-6050*, стандартна інформація про акселерометр змінюється програмним забезпеченням на прискорення вільного падіння g=9,8 м/с^2 за допомогою поділу на коефіцієнт 16384. Під час зміщення та повороту датчика, інформація з акселерометра сильно змінюється. Коли її можна почистити від шумів, тоді акселерометр демонструватиме більш точні результати.

Щоб вирахувати орієнтацію, потрібно для початку ініціалізувати гіроскоп з знаним нам значенням, після чого визначити кутови швидкість ω за допомогою інтервалу часу Δt. Виходячи з цього - ω × T = зміна кута. Але тут постає нова проблема інтегрування. Якщо щоразу буде відбуватися додавання ω × T з часом в результаті буде отримана помилка, що буде зростати. Це являється основною проблемою гідроскопічного дрейфу.

Ділення свіжої інформації з гіроскопа на 131 в результаті дасть швидкість кута в градусах на секунду. 131 – це коефіцієнт чутливості гіроскопу в режимі 250 град/с. Виходячи з того, що він має АЦП 16 біт, маємо що модуль найбільшого значеня, яке необробили буде дорівнювати 32767. Після чого 32767 / 250 = 131 фактичних одиниць на градус на секунду. В результаті, коли неопрацьоване значення становить 131, то швидкість куту матиме значення 1 градус в секунду.

Із використанням цих даних можна одержати місцеположення об’єкта. Щоб виконати це рішення необхідно кутову швидкість помножити на певну кількість часу між запитами датчика гіроскопа. Для прикладу, дозволяємо 2000 градусів на секунду, час між запитами 0,1, значення фактичної швидкості 210, виходячи з цього - 210*0,1=21 – це умовний час, за скільки відбувся поворот на 21 градус. Після чого наступне одержане значення варто додати до попереднього.

Інформація з акселерометру та гіроскопу часто видає постійні помилки. Акселерометр відповідає за одержання більш коректної інформації протягом довгого часу, проте створює сильні перешкоди коли використовується в короткостроковій перспективі. Гіроскоп же пропорційно навпаки добре себе показує на коротких дистанціях, але при зчитуванні інформації отримуємо дрейф.

Щоб вирішити таку проблему, необхідно використовувати інформацію з цих пристроїв так, щоб проблеми, які постають були взаємнознищені. Прийнятим способом для вирішення такого питання вважають фільтр Калмана, проте це доволі складна методика. Для більш простого вирішення такої проблеми можна використати комплементарний фільтр. Комбінація відбувається за допомогою наступної формули:

Кут фільтра = α × (Кут від гіроскопа) + (1 − α) × (Кут від акселерометра)

$$\alpha=\tau/(\tau + \Delta t) \qquad (1.1)$$

Кут від гіроскопа = (Останній виміряний кут фільтром) + ω×T

T = час вибірки

τ = постійна часу перевищує інтервали між шумами

Кут фільтра – відфільтрований, кінцевий кут нахилу.

Бажаний час вибірки приблизно 0,04 с та стала часу приблизно 1 с, виходячи з цього отримуємо, що α≈0,96.

Кут нахилу в своїй основі це додавання інтегрованої інформації із гіроскопа та фактичного значення акселерометра. Основною ціллю такого фільтру є ліквідація дрейфу нуля гіроскопа та помилок інтегрування. На кожній ділянці інтегрування (циклічний крок керування) регулюється інтеграл кута похилу, використовуючи показники акселерометру. Сила такої корекції отримується від коефіцієнту комплементарного фільтру α. Для того, щоб обрати α, необхідно врахувати розмір дрейфу гіроскопа, швидкість отримання та фіксування помилок та від зовнішніх факторів використання. Дуже мале

значення α в результаті призведе до вібрацій корпусу. Коефіцієнт комплементарного фільтра регулюється під кожну ситуацію самостійно.

3.2 PID-регулятори

Регулятор, який виконує свої функції по цьому принципу дуже точний. Іншими властивостями якісного регулювання, наприклад стійкістю або швидкодією цей регулятор похвалитись не може.

Щоб отримати високі значення для всіх критеріїв варто користуватись регулятором, що містить в собі інші закони регулювання. Такий пристрій існує і це пропорційно-інтегрально-диференціюючий (ПІД) регулятор.

Основна його властивість – це обчислення трьох критеріїв , що мають різні передавальні характеристики в один сигнал. Завдяки цьому він має потужну властивість регулювання та може покращити керування з певними характеристиками.

У створення сигналу такого регулятору користуються наступними рішеннями:

- Пропорційна складова – розмір пропорційної помилки нузгодженості (віднімання від заданого та реального значення змінного зпараметру)
- Інтегруюча складова – інтеграл помилки неузгодженості
- Дифференцююча складова – похідна від помилки.

Математично, формулу ПІД можна записати наступним чином:

$$o(t) = P + I + D = Kp\ e(t) + Ki \int e(t)dt + Kd\ de(t)/dt \qquad (1.2)$$

в якій –

- o(t) – основний сигнал;
- P – складова пропорції;
- I – складова інтеграції;
- D – диференціююча складова;
- Кр, Кі, Kd - коефіцієнти пропорційної, інтегруючої, дифференцюючої ланки;
- • e (t) - помилка неузгодженості.

Якщо забразити це структурно та схематично ПІД можна показати наступним чином:

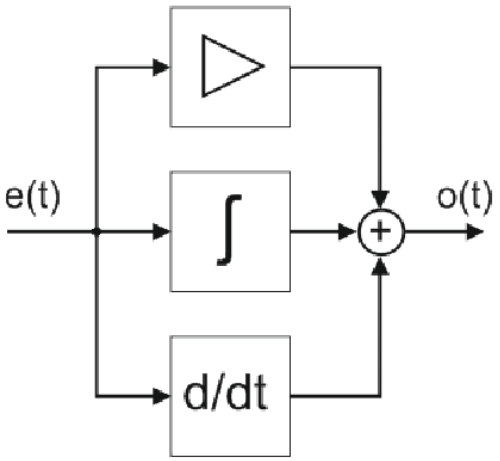
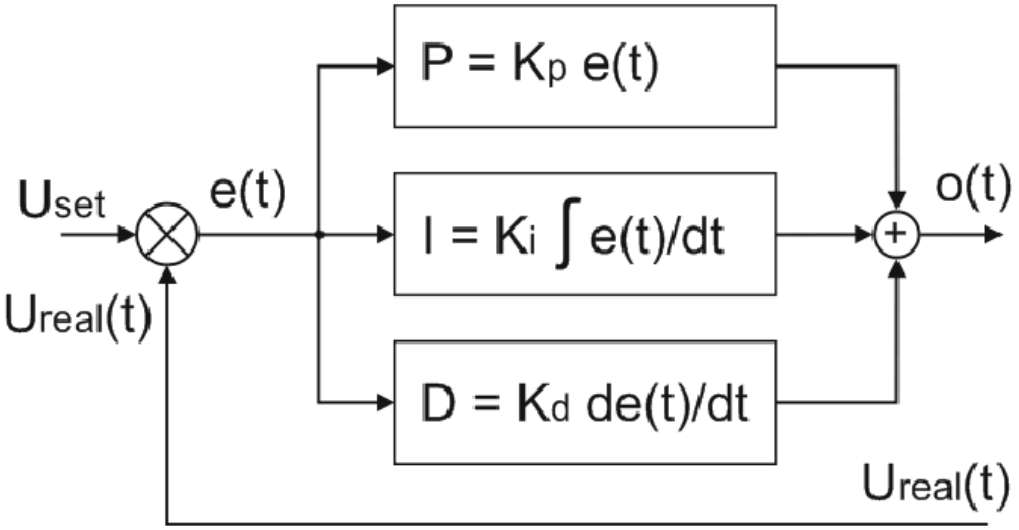


Рисунок 2.2 – Схематичне зображення ПІД – регулятора



Рискунок 2.3 – Структура ПІД регулятора напругою U

Ureal (t) – напруга, яку вимірюють, віднімають із початкового *Uset*. Вада неузгодженості e (t), яку одержали, відправляється на пропорцію, яка змінює ланку дифференції. Як висновок, одержуємо вплив керування o (t), що потім отримує регулюючий елемент, за допомогою суми частин. Під час дії ПІД в програмі зміна значень сигналу, який виходить здійснюється однаково в часових ділянках. Це означає, що ПІД за часом дискретний.

Сигнал, що одержується з ПІД регулятору це додавання трьох складових – диференціюючої, пропорційної та інтегруючої.

Пропорційна складова:

$$P(t) = Kp * e(t) \quad (1.3)$$

Відсутня пам’ять - це означає, що вихідний сигнал незалежний від минулого стану системи. Вада неузгодженості, яку примножили з коефіцієнтом, направляють на вихід. Сигнал виходу стабілізує похибку параметру регулювання. Якщо буде вищий сигнал, то і пропорційн овища вада неузгодженості. Коли вада має значення 0, значення на виході також матиме 0.

Складова пропорції не має можливості анулювати ваду повністю. Це демонструє формула. Сигнал на виході в Кр кількість вищу ніж помилка. Коли вада неузгодженості матиме значення 0, тоді і сигнал регулятора на виході теж буде 0. І, виходячи з цього, анулювати ваду немає чим.

Через це в регуляторах пропорції в будь якому випадку існує, як її називають, статична помилка. Зробити її меншою можливо завдяки збільшенню Кр, проте це може викликати падіння стійкості системи та, в деяких випадках, до автоколихань.

Основними недоліками пропорційних регуляторів вважають:

- присутність фактичної вади регулювання;
- надто невелика стійкість під час збільшення коефіцієнта;

Проте, існує і велика перевага – це швидкодія регулювання. Єдине обмеження реакції ПІД на ваду неузгодженості це час , яка який система дискретизується.

ПІД, які працюють лище за законом пропорції, використовують рідко.

Основною ідеєю в регуляторі виступає підвищення швидкодії.

Інтегруюча Складова

$$I(t) = Ki \int e(t)dt \qquad (1.4)$$

Ваду неузгодженості, якщо пам’ятати про дискретність ПІД варто записувати наступним чином:

$$I(t) = I(t\text{-}1) + Ki * e(t) \qquad (1.5)$$

I(t-1) - фактичне І в минулому відрізку не постійної дискретизації.

Вада неузгодженості перемножується із кофіцієнтом та сумується з минулим значенням інтегралу. Це означає, що сигнал, який подано на вихід постійно додається між собою та протягом часу все сильніше

впливає на об’єкт. Виходячи з цього, вада неузгодженості у всій мірі компенсована, незважаючи на невеликі позначки ПІД та його Кі. У стані спокою сигнал виходу ПІД абсолютно утримується частиною інтегралу.

Основними мінусами даного регулятору є:

- Відносно невелика швидкодія;
- Середня стійкість.

Основними плюсами варто зазначити можливість компенсації всієї вади неузгодженості , незважаючи на коефіцієнт підсилення.

Практично інтегруючими та пропроційно - інтегруючими ПІД користуються частіше. Основним рішенням ланки інтеграції в регуляторах ПІД – компенсувати фактичну помилку та одержання найвищої точності регуляції.

Диференціююча складова.

$$D(t) = Kd\ de(t)/dt \qquad (1.6)$$

Дана складова прогнозує похибку значення, яке регулюється надалі та опір такій похибці. Фактично, відбувається компенсування запізнення сприяння регулятору на предмет та робить вищою систему стійкості.

Враховуючи непостійність дискретності ПІД данну складову можна обчислювати так:

$$D(t) = Kd * (e(t) - e(t-1)) \qquad (1.7)$$

Це демонструє, як сильно редагується рішення неузгодженої вади на одну фактичну одиницю регулятора дискретності.

ПІД, які містять в собі одну ланку диференціювання не існує.

Основною проблемою ланки диференціювання ланки в пристрої – фактично вища стійкість.

Високий клас регуляції ПІД великою мірою в більшості вартує від того, як гарно підібрано коефіцієнти. Вони регулюються практично під час коригування справжнього рішення методом підбирання.

За практичну цінність аналізують виходячи із змінних значень регулятору. Фактично це означає графічне зображення регуляції за певний проміжок часу.

Стандартними рішеннями послідовного налаштування ПІД вважається виявити найкращі можливості регуляції.

Окремі ПІД зазвичай виконують свої функції в неідеальних умовах та вони зазвичай прагнуть до стабільності, інші ж випадки за мету ставлять швидкість своєї дії, проте не точність. Оптимізаційні фактори зазвичай відрізняються. У стандартному рішенні налаштування відбувається задля одержання найбільшого фактору стабільності регулювання всіх рішень.

Основні фактори регулюються незалежно один від одного.

- Вимикаються диференційовані та інтегровані рішення та одержується коефіцієнт пропорції значень. Коли ПІД не має значення диференціювання варто спробувати повну відсутність похибки на значеннях переходу. Коли відбувається настройка ПІД на найвищу швидкість виконання, похибки мають місце бути. Але це не суттєво, через їх анулювання ланкою диференціювання.
- Фактично відбувається з’єднання ланки, через її середнє значення варто видалити похибки фактичної регуляції. В ситуації, коли це неможливо, необхідно відняти частку пропорції.
- Завдяки ланці інтеграції існує можливість прибрати залишкову ваду неузгодженості.

Під характеристику ітерації також підпадає і регулювання ПІД. Це означає, що режими підбирання рішень робитимуться знову і знову до того часу, коли відбудеться необхідне рішення.

Враховуючи помітно кращі властивості та стандартизаціїї використання обрані регулятори часто використовують під час автоматичних рішень на підприємствах.

3.3 Блок-схема стабільної роботи програмного забезпечення

Одержане функціональне математичне середовище можна обрати для модулю стабілізації польоту БПЛА. Рисунок 3.1 демонструє її принцип дії. Важливими рішеннями варто виділити наступні: коригування інформації, яку було одержано із нашої мікросхеми MPU, інформація про поточне розміщення кута крену та змінна інформація стосовно швидкості кута та регуляції її на будь-якому з чотирьох двигунів.

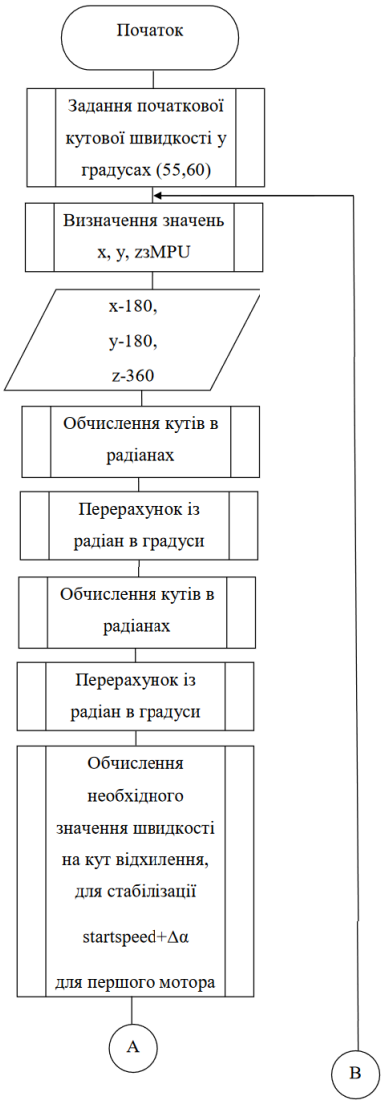
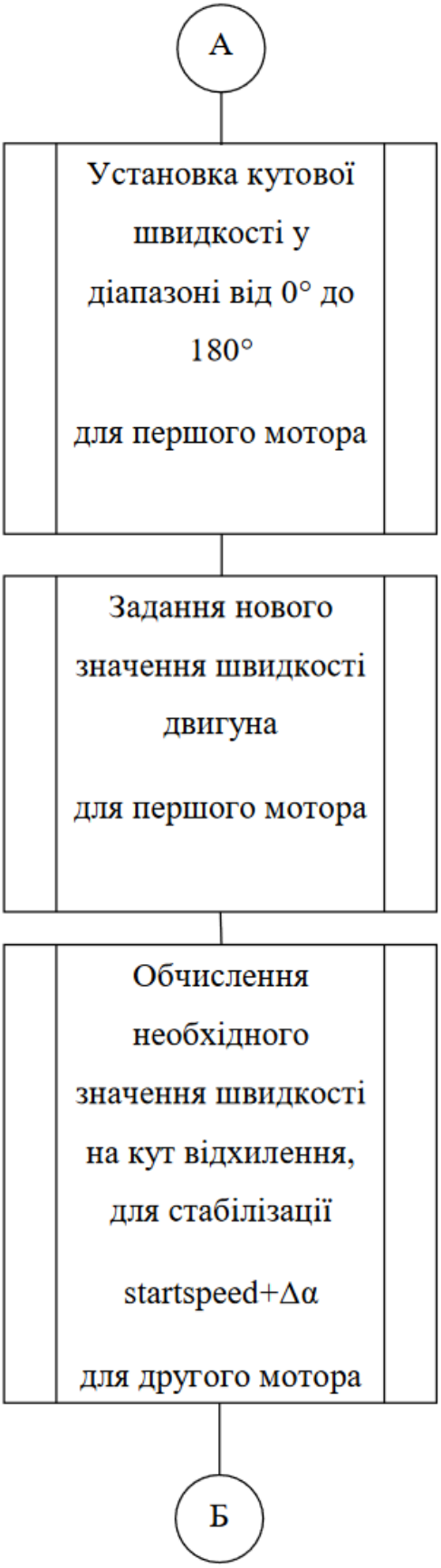
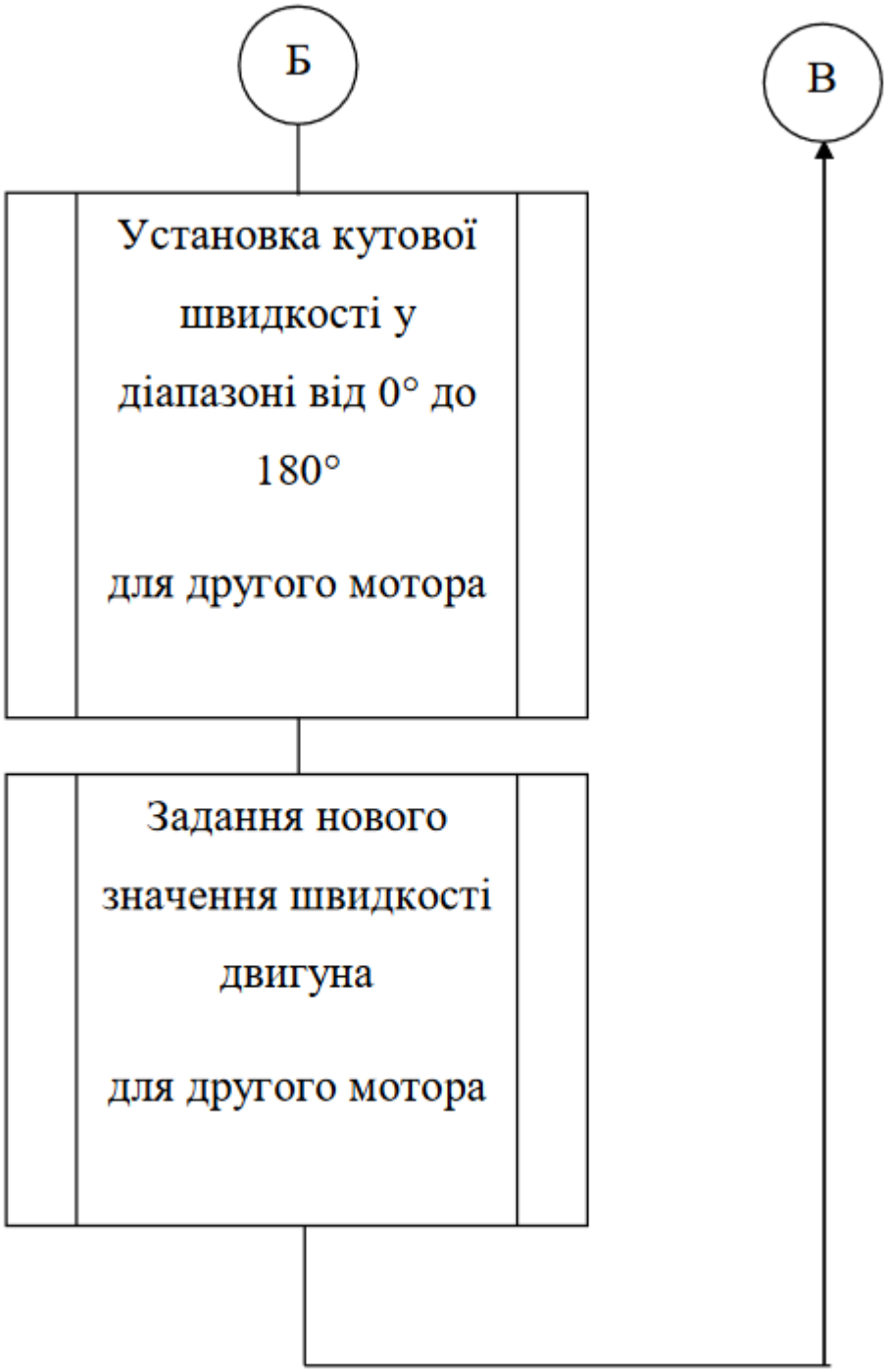


Рис 3.1 – Блок-схема рішення проблеми стабілізації БПЛА



Продовження рис.3.1 – Блок-схема рішення проблеми стабілізації БПЛА



Продовження рис.3.1 – Блок-схема рішення проблеми стабілізації БПЛА

Завдяки програмній бібліотеці I2C, а також програмному забезпеченню Arduino було програмно створено рішення проблеми стабілізації безпілотного літального апарату.

НЕЙРОННІ МЕРЕЖІ В ПРОБЛЕМАХ СТАБІЛІЗАЦІЇ КВАДРОКОПТЕРА

4.1 Призначення та середовища використання автоматичних дронів

БПЛА, в яких міститься ШІ все частіше використовують як пристрої для великого ряду задач. Основним призначенням дронів в наш час є воєнна справа, такі дрони дуже допомагають у ході ведення бойових, розвідувальних та інших операціях. Збільшення ефективності розвідки, наведення високоточних ударів по цілях та інших бойових задач.

Технології включення штучного інтелекту в бойові дрони допомагають стабілізувати дрон, коли по ньому «б'ють глушилками» або збивають кулями. Завдяки штучному інтелекту дрон прямує незмінно тримає курс на ціль. Завдяки цьому такі БПЛА стають більш боєздатнішими, ніж ті, які наведено тільки по координатам. Вони допомагають нашим бійцям бити по найбільш чутливим об'єктам у тилу противника та нарощувати точність ударів артилерії.

Також вже є випадки використання дронів із ШІ для доставки товарів, особливо у місця, в які є проблеми з досткавкою. Дрони можуть привезти продукти, воду та інші предмети для цивільних людей та людей, до яких немає фактичного доступу.

Зараз дронів також використовують для патрулювання територій, безпеки територій та аналіз на потенційні загрози, а також БПЛА сприяють науковим дослідженням, для цього їх відправляють в місця, до яких важко дістатись або небезпечно для людини, такі як вулкани, радіоактивні зони і т д.

4.2 Способи навчання та їх види

Головною перевагою нейронних мереж є те, що вони знають, що вони вивчають та в якій області вони працюють завдяки інформації, що вони отримують, що в свою чергу крок за кроком підвищують їх ефективність. Така подія підвищення змінюється пропорційно до бажаних значень. За весь час навчання можна видозмінити як побудову архітектури мережі або ітерованої зміни значень ваги задля поліпшення рішення поставленого завдання.

У ході вивчення нейронною мережею одержується деяка ітерація рішень: отримані значення прямують до мережі, після чого певні значення мережі підлягають зміні, які в подальшому змінюють структуру всередині та , як висновок, мережа відгукується на одержані значення іншим способом. Основний закон навчання регулюється за допомогою стабілізації синапсів нейронів. Ще не було винайдено ідеальний алгоритм вивчення, що стабільно і правильно навчав би всі ланки рішень нейронних мереж. Така мережа має можливість редагувати змінні завдяки головній навчальній машині. Таке рішення впливає на мережу краще, аніж мережі, які діють за встановленими рішеннями.

Щоб розпочати процес навчання варто отримати модель зовнішнього середовища, всередині якого діють нейрони, це означає одержання інформації, що потрібна для рішення необхідної задачі. Не дивлячись на це, також необхідно дізнатись, як саме змінювати параметри нейронів. Алгоритм вивчення виділяє методи, із рішеннями яких рішення вивчення використовують задля зміни та стабілізації значень. 3 основних методики навчання , що використовуються це:

1. Контрольоване навчання із вчителем
2. Самонавчання (неконтрольоване)

3. Змішане

Контрольоване навчання із вчителем має на меті користування вірними рішеннями для всіх прикладів зміни мережевих рішень. Вони редагуються так, щоб нейрони створювали рішення, що найбільш схожі із вже знаними вірними рішеннями.

Самонавчання не вимагає якихось знань вірних рішень на всі приклади змінних в переліку. В цьому рішенні задіяна інформаційна структура, що міститься всередині та кореляція прикладів в переліку навчання, а це в свою чергу дає змогу поділу прикладів за певними даними.

Під час змішаного навчання певна частина рішень відбувається із вчителем, в той час як інша частина регулюється самонавчанням.

4.3 Дрони та їх симуляція

Для створення поставленої симуляції ми будемо користуватись Unity 2021 LTS за допомогою UnityHUB (рис. 3.1). Починаємо наше рішення та інсталюємо всі необхідні додаткові завдання.

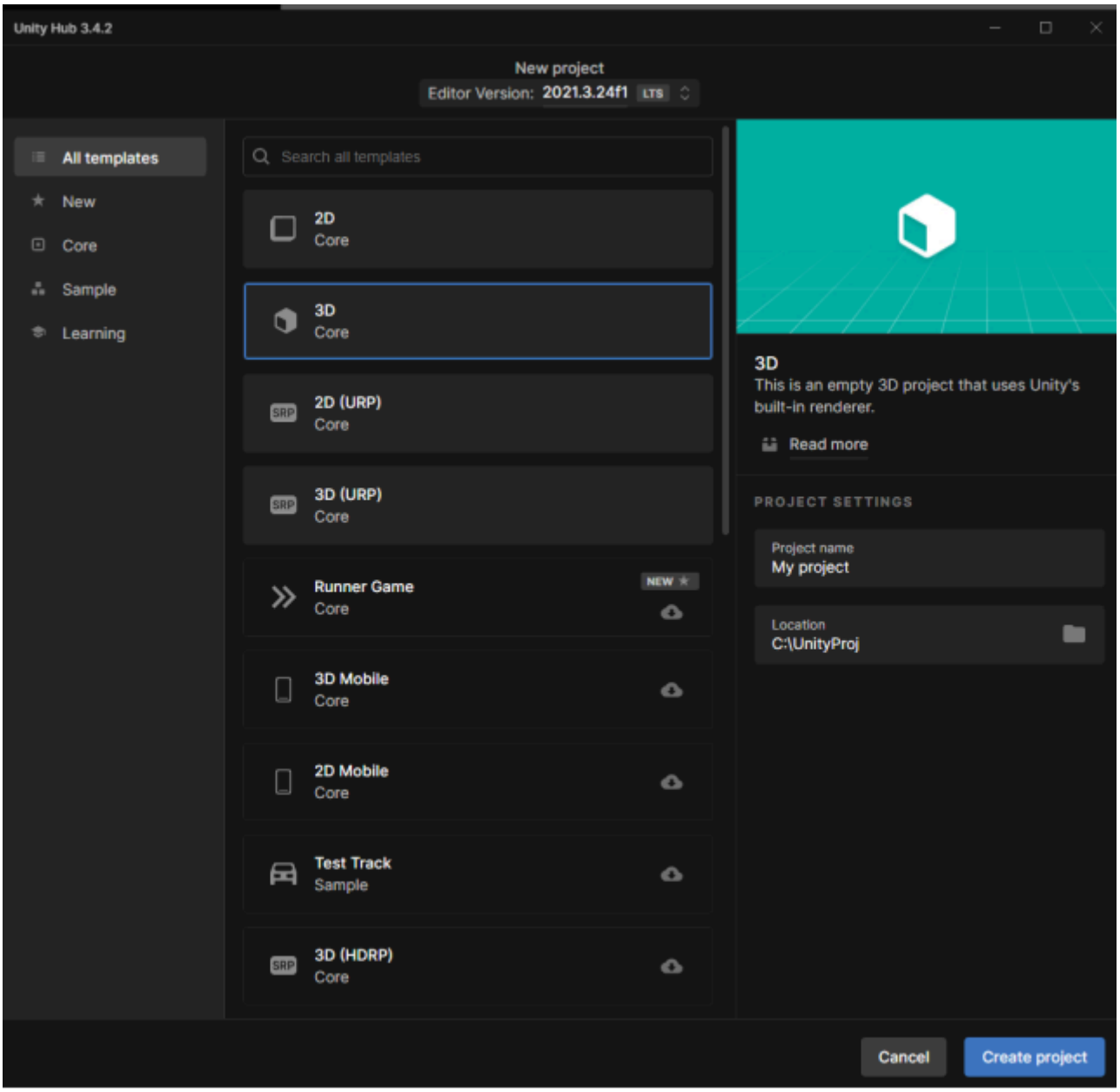


Рис. 4.1. Unity HUB

Наступним кроком буде інсталяція віртуального середовища та його запуску, використовуючи команду «activate». Після того, як середовище було інстальовано, варто додати інформацію про зовнішні об’єкти (рис. 4.2). Наприклад, це різні споруди, сам квадрокоптер, його візуальне зображення та ін. Насправді, це більше естетичний елемент на мапі.

4.3.1 Фізика дронів

Імітація фізичних властивостей дронів в забезпеченні Unity заснована на загальнодоступному движку PhysX, який створила компанія NVIDIA. Він має достатньо ресурсів для того, щоб демонструвати найближчі до реальності дії об’єктів у віртуальній реальності із врахуванням фізичних значень, наприклад, колізії, тяги, крену, тангажу опору повітря на ін.

Головними рішеннями Unity задля імітації фізичних властивостей є:

RigidBody. Це властивість, що сумується з об’єктом задля додавання наступного в фізичне рішення. Властивості підвласні змінні значення дослідного рішення, наприклад вага, інерція, пришвидшення.

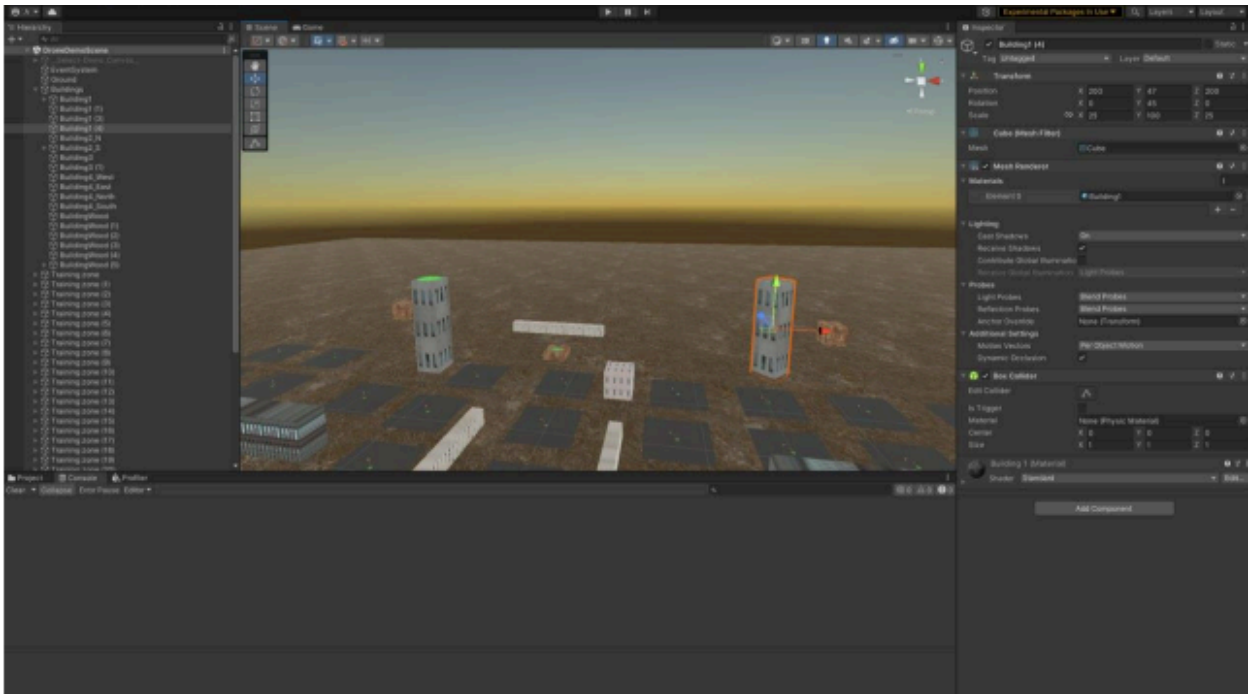


Рис. 4.2. Один із варіантів «сцени» створеної для імітації

Collider. Самі по собі колайдер це геометрія об’єктів в фізичній імітації, а також можна позначити як взаємодію об’єктів один з одним при їх ударі. Unity може користуватись певним переліком різних колайдерів, наприклад Capsule, Mesh, Sphere, Box.

Physics Material. Природні об’єкти отримують поверхневе значення об’єкту в відношенні один одному, наприклад рикошет або зіткнення. Це дає можливість контролювати об’єкти під час аварій.

Joints. Елементи з’єднання мають право відділяти рух об’єктів в відношенні один одному, завдяки обмежувачам руху, пружинам або шарнірам.

Physics Manager. Це частина програмного забезпечення Unity, що створена для налаштування глобальних значень фізики та їх зміни, наприклад гравітація. Це такі параметри, які безпосередньо мають значення в симуляції.

Імітація в Unity відбувається з використанням ітеративних ситуацій, що змінюються в кожному кадрі. Движок дії фізики в цей час обчислює фізичні сили, які тиснуть на об’єкти, їх аварії, а також зміна інформації про їх переміщення в цей момент. Висновки імітації фізичних явищ що показані на схемах, дають нам дійсне уявлення про взаємодії об’єктів між собою у віртуальній машині.

4.3.2 Інсталяція середовища навчання

Створюємо так звану «арену» для всіх піддослідних та площу тренувань для окремих піддослідних, яку повторюємо велику кількість разів, для того, щоб паралельно могли навчатись велика кількість піддослідних.

Зона тренування містить в собі початкову та бажану точки, візуальні обмеження у вигляді стін, стелі і так далі, та зону порушень. Бажана точка кожного разу змінює позицію на ділянку розміром 5 метрів в сторону та висоту 3-6 метрів.

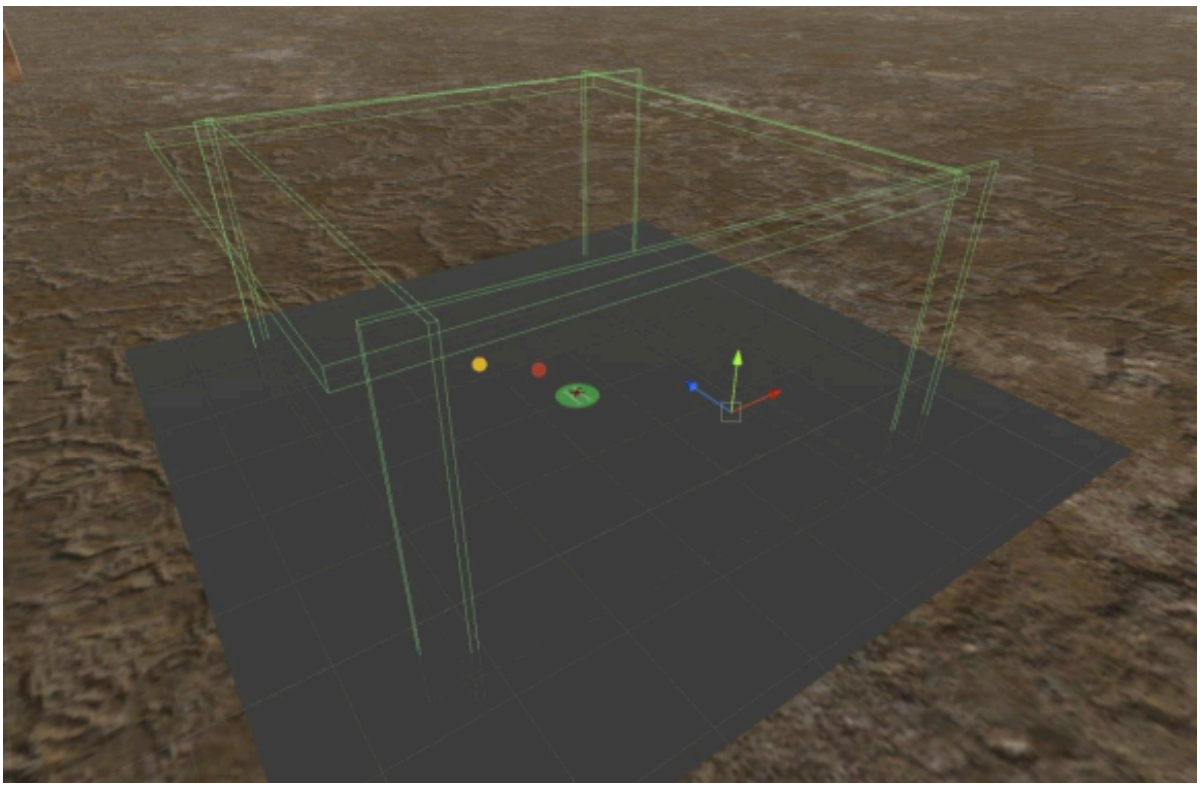


Рис. 4.3 Зона тренувань

Зона порушень міститься на довільній точці з максимальною відстанню від бажаної точки в 3 метри.

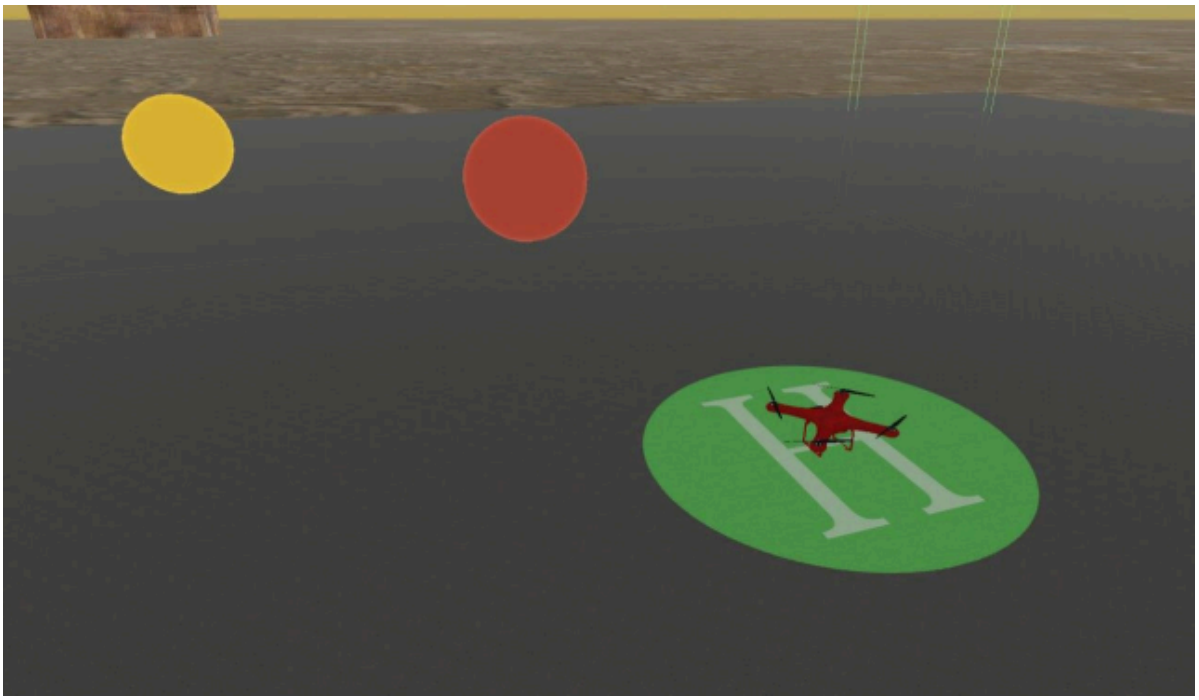


Рис. 4.4 Наближений вид на зону тренувань

Випадковим чином ми обрали навчання, яке буде проходити на 37 піддослідних водночас. Загальна кількість, яка буде проходити разом не є такою важливою. Все впирається в можливості та швидкість отримання бажаного результату. Проте, не варто забувати про спільне «поле» для кожного з рішень.

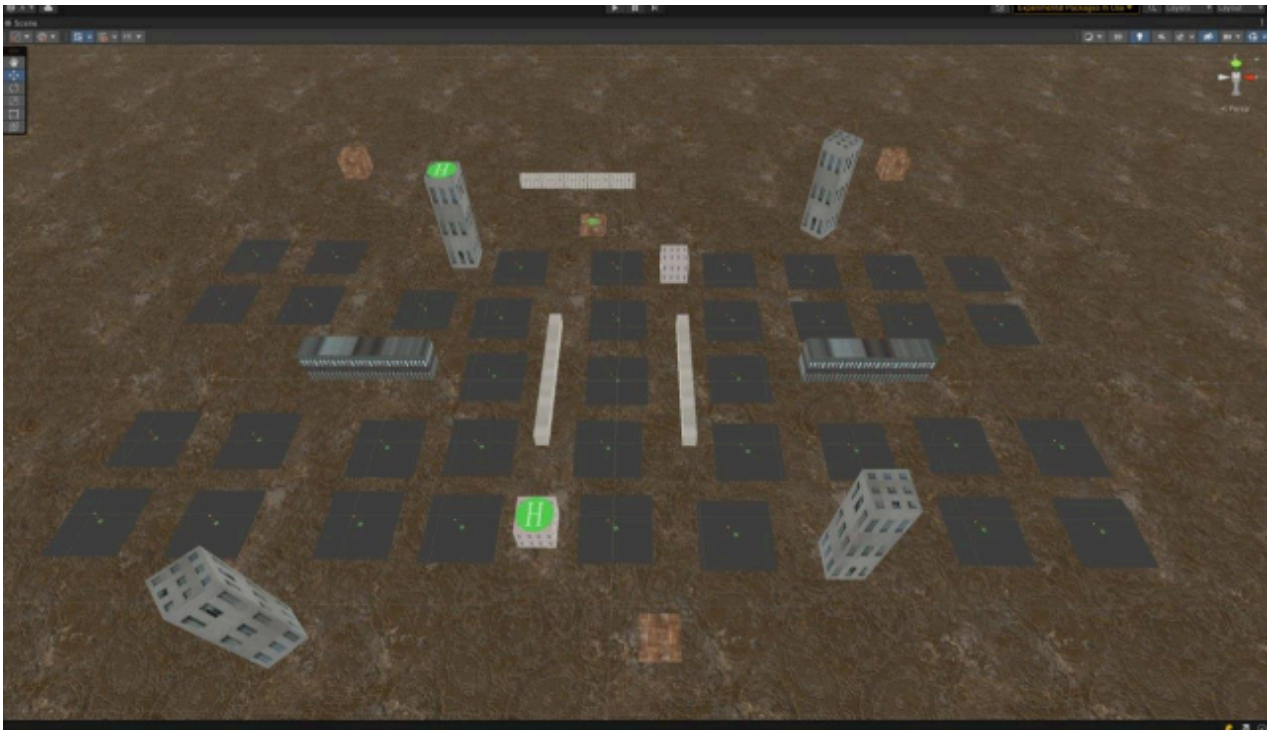


Рис. 4.5 Спільне поле рішень

Щоб всі тести проходили в відповідності один одному ми керуємось спільними суперпараметрами системи. Спочатку спроби були не дуже вдалі через те, що не було стимулу рухатись до бажаної зони. Завдяки формулі 1.1 ми додаємо таке рішення:

$$R=(L-M)*3/L, \qquad (2.1)$$

де R – нагорода, L – відстань від місця появи до бажаної зони та M – це відстань від піддослідного до бажаної зони.

Порівняння виконаних рішень до та після того, як ввели формулу 2.1 показані на рисунку 4.6.

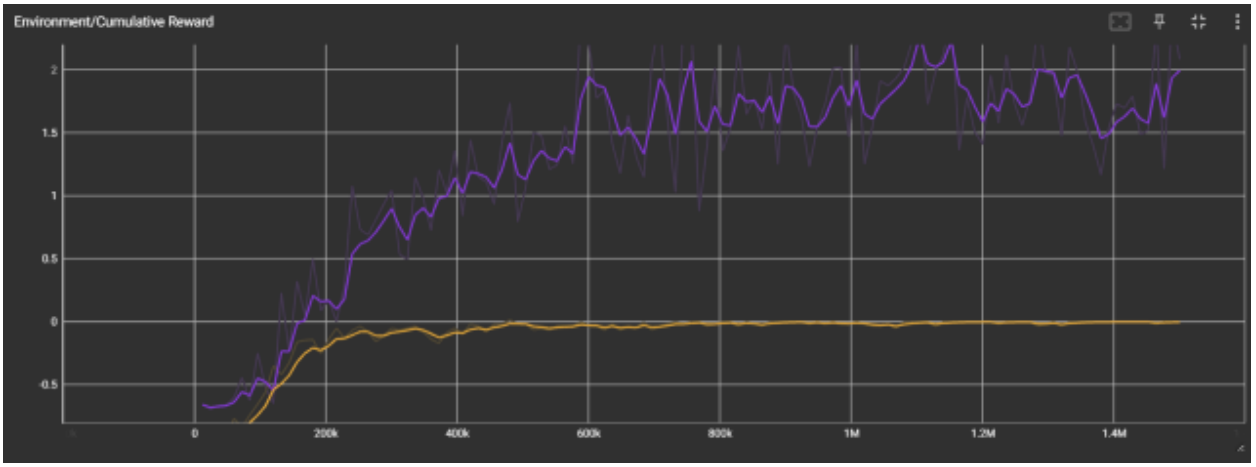


Рисунок 4.6 Порівняння виконаних рішень до та після створення формули

На рисунку можна побачити, що квадрокоптер до введення нагороди (жовтий – помаранчевий) перед бажаною зоною навчився висіти над ціллю і на цьому все (рис. 4.7). Після введення формули піддослідний зрозумів важливість бажаної зони.

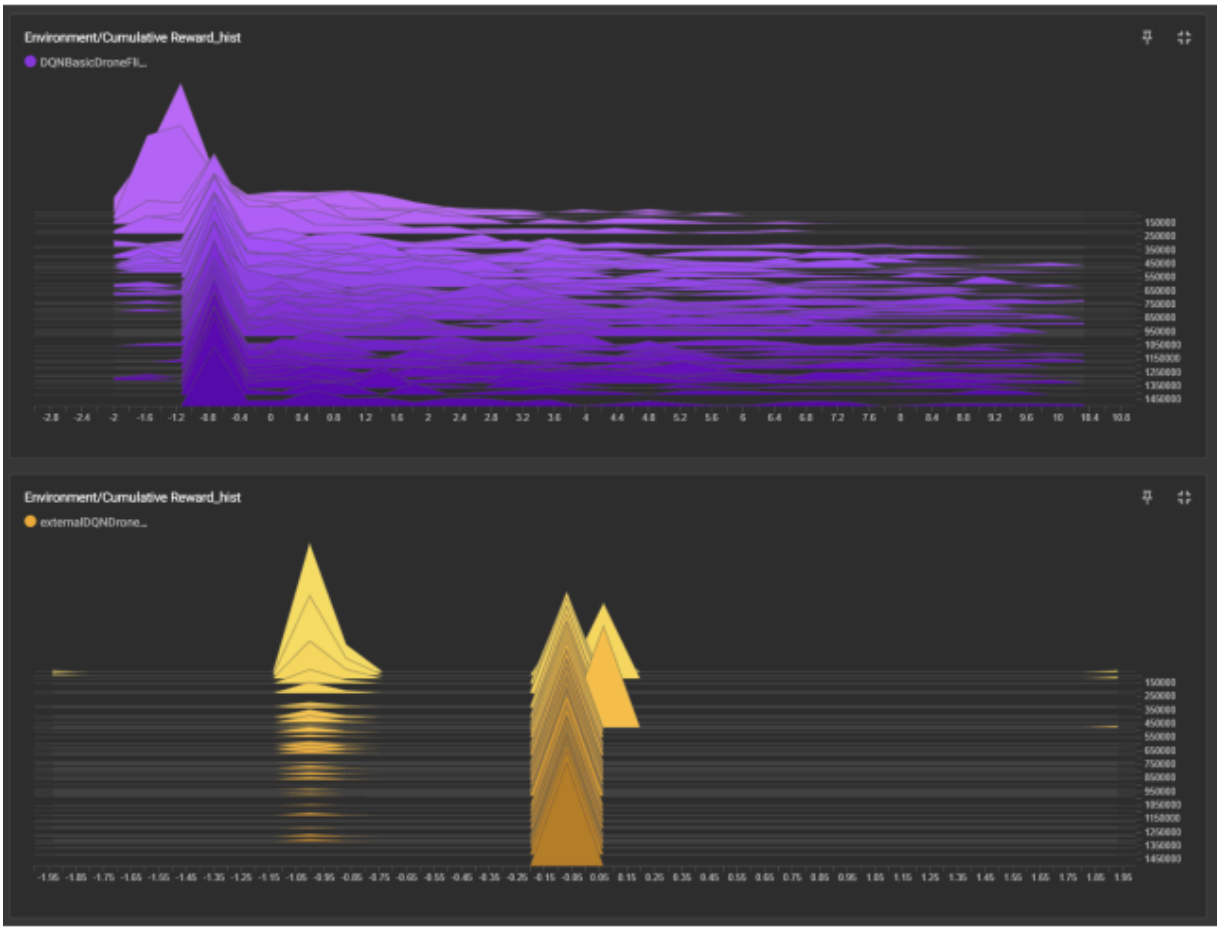


Рис. 4.7 Порівняння виконаних рішень до та після створення формули у вигляді гістограми

4.3.3 Навчання за алгоритмом Deep Q-Network

Далі необхідно зробити власну задачу тренування для піддослідних. Для цього створюється клас тренування Deep Q-Learn. Після цього варто додати зберігання доступу та одержання властивостей досвіду з Зберігача Досвіду. В основному ці методи будуть модифіковано найближчим часом. Наступним кроком будуємо середовище в мережі DQN.

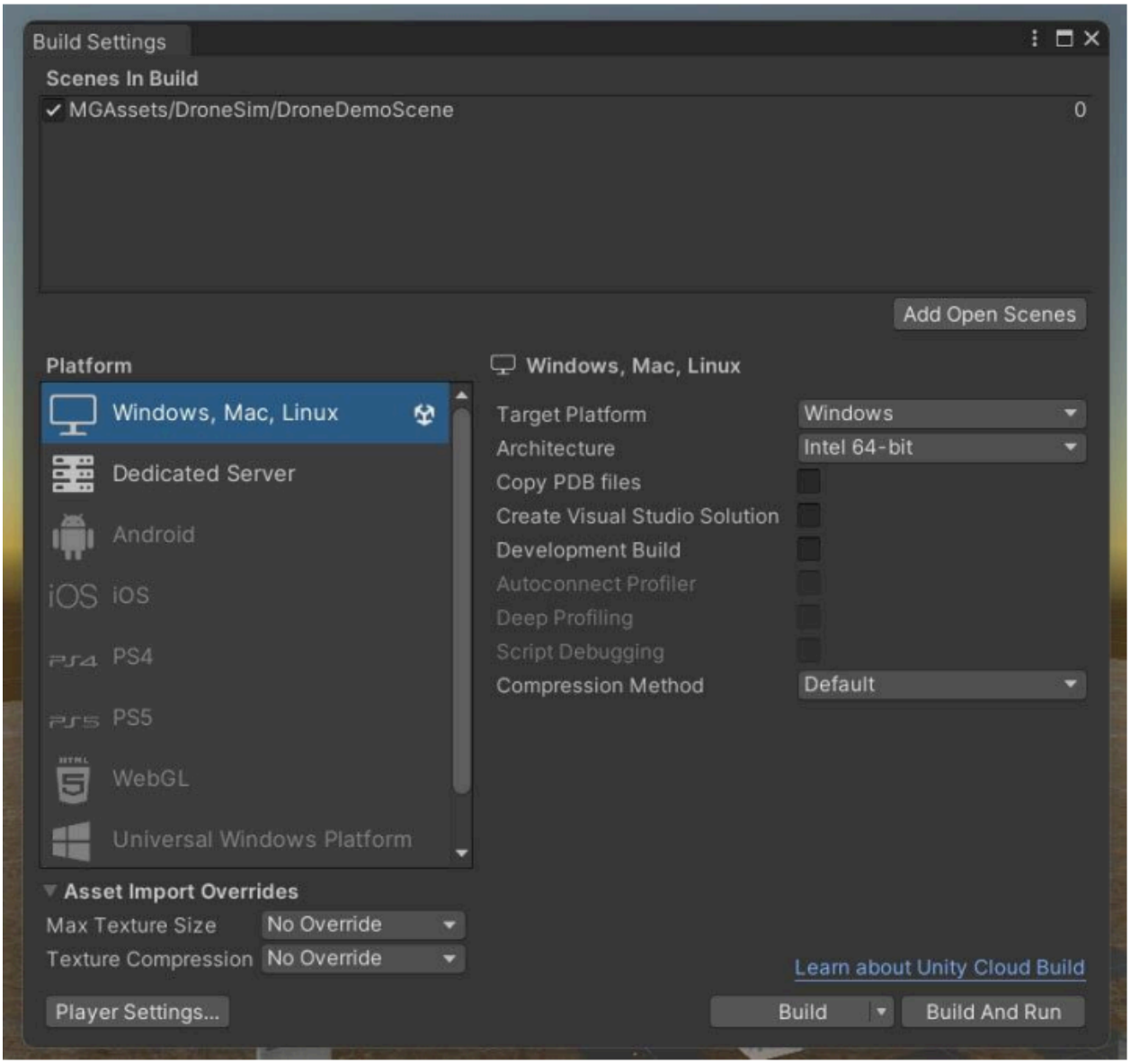


Рис. 4.8 Настройки

Після налаштування створеного віртуального середовища запускаємо тести.

Пересічну нагороду за цикл показано на наступному рисунку:

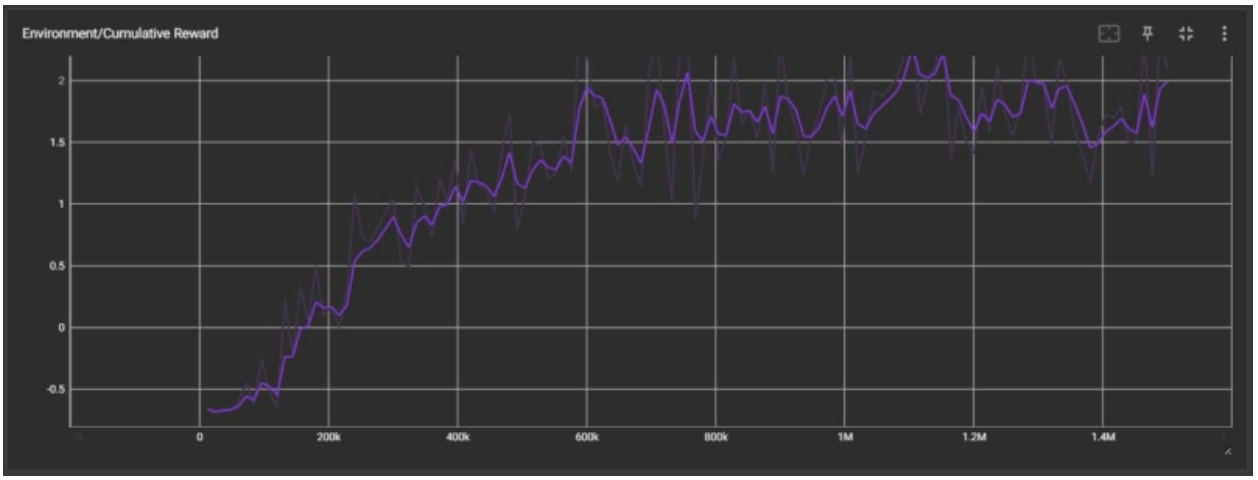


Рис. 4.9 Пересічна нагорода

4.3.4 Навчання за покращеним алгоритмом

Покращення такого алгоритму складається із двох рішень. Для покращеного алгоритму необхідно реалізувати найефективніші знання. В класі DQN додається, окрім таблиці з досвідом, таблиця із посиланням на нього та числом його важливості. Додаємо алгоритм обчислення важливості в наш код та завдяки класу Random створюємо відкидання частки та на основі цього ж класу створюємо прибирання однієї частини досвіду через кожні 10 відкидань. Також необхідно прибрати всі посилання на помилки, щоб не було подальших проблем.

Пересічну нагороду за цикл показано на наступному рисунку:

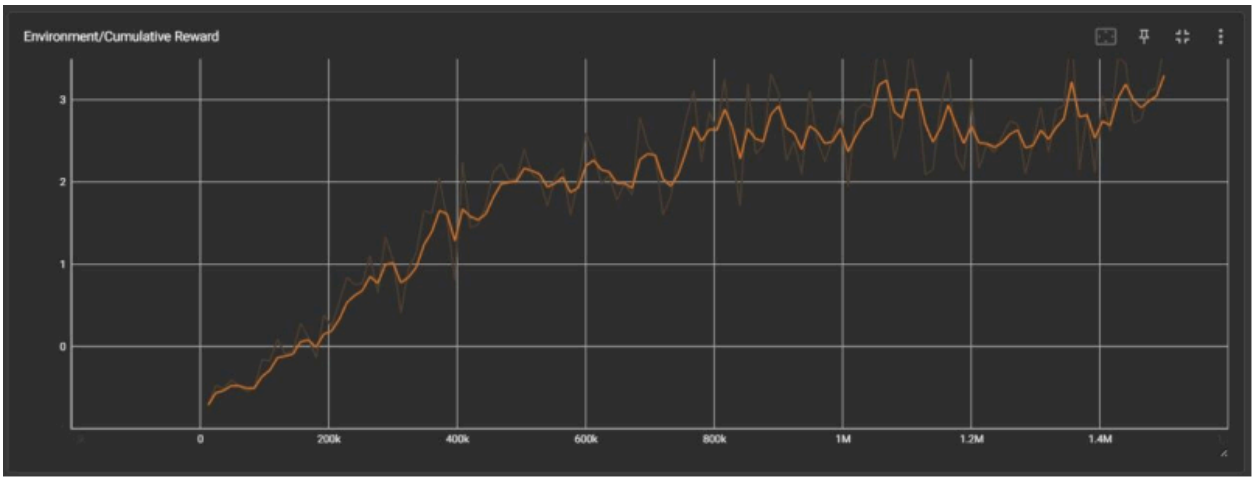


Рис. 4.10 Пересічна нагорода за покращеним алгоритмом

4.3.5 Огляд результатів

Аналізуючи рисунок порівняння (рис. 4.11) ми одержали фактичні результати. Оранжевий колір – це покращений алгоритм, пурпуровий – стандартний DQN. Квадрокоптер, що керувався покращеним алгоритмом дав вдвічі кращі результати задля одержання бонусу.

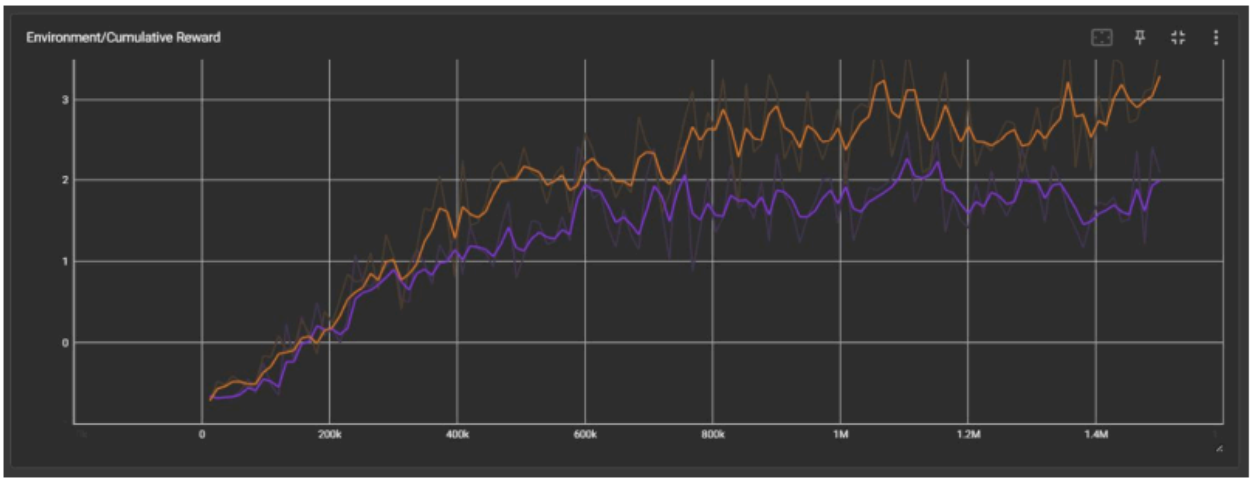


Рис. 4.11 Порівняння методів

Проте, використання такого алгоритму має і свої мінуси. Надійність передбачення винагороди, через те, що ми видаляємо наші попередні

результати, не дає можливості отримати такі ж хороші результати, як в алгоритму DQN, проте залишається в межах п’ятнадцяти відсотів допустимої похибки.

Незважаючи на дозволене відхилення в 15 відсотків для бажаної нагороди, покращений метод показав нейиовірні результати в умовах експерементального рішення. Це дає нам зрозуміти, що такий метод має дуже хороший потенціал для ефективного навчання квадрокоптера, незалежно від різниці фактичної нагороди.

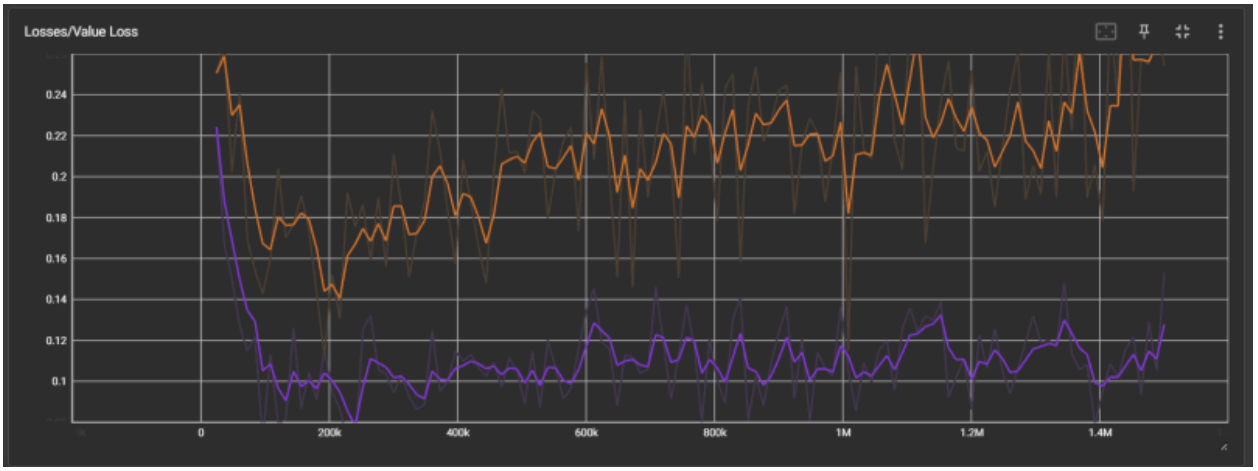


Рис. 4.12 Похибка прогнозування нагороди

ВИСНОВКИ

Даний магістерський проект зосереджений на створенні та порівнянні методології стабілізації польоту квадрокоптера на базі нейронних мереж. Під час проведення дослідження було проаналізовано найкращу модель для створення квадрокоптеру, визначення апаратної частини керування, розроблено алгоритм програми стабілізації та, використовуючи нейронні мережі, проведено синтез методів управління дронами, з урахуванням таких ключових параметрів польоту, як точність позиціонування, стабільність польоту та оптимізація вибору маршруту з огляду на мінімізацію часу. Основне завдання було проаналізовона задля найбільш правильного розуміння задач та рішень такого дослідження. Структурний аналіз та проектування архітектури фізичної моделі дрона дозволили розробити адекватну робочу фізичну модель.

Створення квадрокоптера за обраною моделлю навчання, разом із аналізом результатів, які вийшли, створили високу ефективність такого підходу та фактичної симуляції. Під час дослідження одержано наступні наукові та практичні результати:

- Отримано високі показники за критерієм оптимізації вибору маршруту квадрокоптеру, виходячи з найменшою кількістю

витраченого часу, стабільності польоту та найкращого позиціонування;

- Запропонована методологія за однакових умов навколишнього середовища та схожих цілях призводить до підвищення бажаних характеристик;
- Підвищення помилки передбачення нагороди, що піднімає проблему перенавчання.

Наукові та практичні результати, отримані в ході цієї роботи, можуть бути використані в різних областях. Логістика, нейронні мережі, Інтернет Речей та інші більш прикладні сфери, такі як пошта чи екологія.

Список використаних джерел

1. «EASA. Concept of Operations for Drones—A Risk Based Approach to Regulation of Unmanned Aircraft». [Електронний ресурс]. https://www.easa.europa.eu/sites/default/files/dfu/204696_EASA_concept_drone_brochure_web.pdf
2. «SEAR. European ATM Master Plan, Executive Summary». [Електронний ресурс]. https://www.sesarju.eu/masterplan
3. Хебб Д. О. Організація поведінки [Текст]/ Ann Amer Acad Pol Soc Sci. – 1950 – 7с.
4. Nguyen, A.T., Xuan-Mung, N.: «Quadcopter Adaptive Trajectory Tracking Control: A New Approach via Backstepping Technique» [Текст]/ Nguyen, A.T., Xuan-Mung, N. -- Appl. Sci. 2019, 3873 – 9 с.
5. Kendoul, F. «Nonlinear hierarchical flight controller for unmanned rotorcraft: Design, stability, and experiments». [Текст]/ J. Guid. Control Dyn. 2009, 1954–1958 – 32 с.
6. «Using Deep Q-Networks to Train an Agent to Navigate the Unity ML-Agents Banana Environment». https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3881878

7. «Autonomous Flight Trajectory Control System for Drones in Smart City Traffic Management». https://www.mdpi.com/2220-9964/10/5/338
8. «A Deep-learning-aided Automatic Vision-based Control Approach for Autonomous Drone Racing in Game of Drones Competition». http://proceedings.mlr.press/v123/kim20b/kim20b.pdf
9. «Coordinated Path-Following Control of Fixed-Wing Unmanned Aerial Vehicles». [Електронний ресурс]. https://ieeexplore.ieee.org/document/9335475
10. Dung, N.D.: «Developing models for managing drones in the transportation system in smart cities» [Текст]/ Sci. J. -- Riga Tech. Univ. Electr. Control Commun. Eng. 2019, -- 15, 71–78 с.
11. «The drone-following models in smart cities». https://ieeexplore.ieee.org/document/8659813
12. « Robust planning the landing process of unmanned aerial vehicles». https://www.researchgate.net/publication/333257564_Robust_planning_the_landing_process_of_unmanned_aerial_vehicles.
13. «Low-altitude unmanned aerial vehicles-based internet of things services: Comprehensive survey and future perspectives». https://ieeexplore.ieee.org/document/7572034
14. «Unmanned Aircraft Systems Traffic Management (UTM)—A Common Framework with Core Principles for Global Harmonization». https://www.icao.int/safety/UA/Documents/UTMFramework%20Edition%202.pdf
15. «Recent research progress of unmanned aerial vehicle regulation policies and technologies in urban low altitude». https://www.icao.int/safety/UA/Documents/UTMFramework%20Edition%202.pdf

16. SESAR. «Automated Support for Dynamic Sectorisation». https://www.sesarju.eu/sesar-solutions/automatedsupport-dynamicsectorisation
17. Alessandro Palmas, Emanuele Ghelfi et al. «The Reinforcement Learning Workshop». [Текст]/Packt Publication, United Kingdom, 2020 -- 17-42.
18. Miguel Morales. «Grokking Deep Reinforcement Learning». [Текст]/ Manning Publications, Shelter Island, New York, United States, 2020.
19. «Human-level control through deep reinforcement learning». [https://www.deepmind.com/publications/human-level-controlthrough-deep-reinforcement-learning
20. «Deep Learning with Javascript». https://www.manning.com/books/deep-learning-with-javascript.
21. «Deep Reinforcement Learning in Action». https://www.manning.com/books/deepreinforcement-learning-in-action.