

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматики
Кафедра автоматизації та інтелектуальних інформаційних технологій

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія оптимізації планування траєкторії групи
безпілотних літальних апаратів для оцінки забруднення повітря»

Виконала: студентка 2-го курсу, групи
ІСТ-23м, спеціальності 126 –

Інформаційні системи та технології

(шифр та назва спеціальності)

Анастасія БАРАНОВСЬКА

(ім'я ПРІЗВИЩЕ студента)

Керівник: к.т.н., доц. каф. АІТ

Ярослав КУЛИК

(науковий ступінь, вчене звання / посада, ім'я ПРІЗВИЩЕ керівника)

« 13 » грудня 2024 р.

Опонент: Володимир ОВЕРАНСЬКИЙ

(науковий ступінь, вчене звання / посада, ім'я ПРІЗВИЩЕ опонента)

« 16 » грудня 2024 р.

Допущено до захисту
Завідувач кафедри АІТ

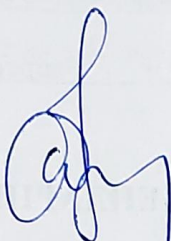
д.т.н., проф. Олег БІСІКАЛО

(науковий ступінь, вчене звання)

« 16 » 12 2024 р.

Вінниця ВНТУ – 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра автоматизації та інтелектуальних інформаційних технологій
Освітньо-кваліфікаційний рівень - магістр
спеціальність 126 Інформаційні системи та технології



ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ
д.т.н., проф. Олег БІСІКАЛО
«19» вересня 2024 р.

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТЦІ
Барановській Анастасії Юріївні

1. Тема роботи: Інформаційна технологія оптимізації планування траєкторії групи безпілотних літальних апаратів для оцінки забруднення повітря
Керівник роботи: к.т.н., доцент каф. АІТ Кулик Я.А.
Затвердженні наказом ВНТУ від «17» вересня 2024 року № 310
2. Термін подання роботи студентом: до «16» грудня 2024 року.
Вихідні дані до роботи: Безпілотний літальний апарат, оперативна пам'ять не менше 4 ГБ, ємність акумулятора не менше 5000 мАг, сенсори MQ-2, DHT11, TMP36, KY-026, Мова програмування C
3. Зміст текстової частини: Вступ; Аналіз існуючих систем та підходів; Порівняльне дослідження структури генетичних алгоритмів; Розробка програмного забезпечення та проектування; Тестування роботи системи; Економічна частина; Висновки; Список використаних джерел.
4. Перелік ілюстративного (або графічного) матеріалу: UML діаграма класів програми, UML діаграма активностей системи, UML діаграма послідовності, UML діаграма розгортання системи, UML діаграма станів, UML діаграма об'єктів, UML діаграма колаборації.

6. Консультанти розділів проекту (роботи)			
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1-4	Ярослав КУЛИК к.т.н., доцент кафедри АІТ	19.10.24	13.10.24
5	Олександр ЛЕСЬКО, к.е.н., проф. каф. ЕПтаВМ	05.10.24	07.11.24

7. Дата видачі завдання: « 18 » вересня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Приміт
1	Аналіз предметної області	20.09.24 - 30.09.24	вск.
2	Вибір оптимальних інформаційних технологій	30.10.24 - 05.10.24	вск.
3	Вибір мови програмування та середовища розробки	05.10.24 - 08.10.24	вск.
4	Програмна реалізація	15.10.24 - 01.11.24	вск.
5	Експериментальна перевірка роботи програми	02.10.24 - 04.11.24	вск.
6	Оформлення економічного розділу	05.10.24 - 07.11.24	вск.
7	Оформлення пояснювальної записки, графічного матеріалу і презентації	08.10.24 - 11.11.24	вск.
8	Попередній захист роботи	До 05.12.24	вск.
9	Захист роботи	До 20.12.24	вск.

Студентка

(підпис)

Анастасія БАРАНОВСЬКА

(прізвище та ініціали)

Керівник роботи

(підпис)

Ярослав КУЛИК

(прізвище та ініціали)

АНОТАЦІЯ

УДК 004.855.5

Барановська А.Ю. Інформаційна технологія оптимізації планування траєкторії групи безпілотних літальних апаратів для оцінки забруднення повітря. Магістерська кваліфікаційна робота зі спеціальності 126 – Інформаційні системи та технології, освітньо-професійна програма – Інформаційні технології аналізу даних та зображень. Вінниця: ВНТУ, 2024. 127 с.

Українською мовою. Бібліогр.: 30 назв; рис.: 46; табл.: 3.

У магістерській кваліфікаційній роботі було досліджено підходи до розробки інтелектуальних систем збору та обробки даних про стан довкілля, зокрема за допомогою сенсорів і генетичних алгоритмів. Для реалізації проєкту було використано мікроконтролери Arduino, сенсори MQ-2, TMP36, KY-026, DHT11, а також мову програмування C. Інтелектуальність системи забезпечує автоматичну генерацію оптимального маршруту дослідження обраної території, скорочуючи час його виконання. Розроблена система є модульною, гнучкою в налаштуванні та відкритою для подальших розширень.

Ключові слова: генетичні алгоритми, безпілотні літальні апарати, алгоритми оптимізації, оптимізація руху, машинне навчання, екологічні проблеми.

ANNOTATION

Baranovska A.Y. Information Technology for Optimizing the Trajectory Planning of Unmanned Aerial Vehicle Groups for Air Pollution Assessment. Master's Thesis in specialty 126 – Information Systems and Technologies, educational-professional program – Information Technologies for Data and Image Analysis. Vinnytsia: VNTU, 2024. 127 pages.

In Ukrainian language. References: 30 titles; Figures: 46; Tables: 3.

The master's thesis investigates approaches to developing intelligent systems for collecting and processing environmental data, particularly using sensors and genetic algorithms. The project implementation utilized Arduino microcontrollers, MQ-2, TMP36, KY-026, and DHT11 sensors, as well as the C programming language. The system's intelligence enables the automatic generation of an optimal research route for a selected area, reducing execution time. The developed system is modular, customizable, and open to further expansions.

Keywords: genetic algorithms, unmanned aerial vehicles, optimization algorithms, motion optimization, machine learning, environmental issues.

ЗМІСТ

ВСТУП	9
1 АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ТА ПІДХОДІВ	12
1.1 Огляд існуючих систем оцінки якості повітря.....	12
1.1.1 Наземні системи оцінки якості повітря	14
1.1.2 Безпілотні системи оцінки якості повітря	18
1.2 Аналіз функціонування алгоритмів сучасних систем	22
1.3 Оцінка наявних аналогів системи.....	28
1.4 Висновки до першого розділу.....	29
2 ПОРІВНЯЛЬНЕ ДОСЛІДЖЕННЯ СТРУКТУРИ ГЕНЕТИЧНИХ АЛГОРИТМІВ	31
2.1 Принцип роботи генетичного алгоритму	31
2.2 Огляд підвидів генетичних алгоритмів.....	33
2.2.1 Генетичні алгоритми з множинним критерієм	33
2.2.2 Паралельні генетичні алгоритми	35
2.2.3 Адаптивні генетичні алгоритми	37
2.2.4 Генетичні алгоритми з використанням додаткових евристик.....	39
2.2.5 Меметичні алгоритми	40
2.3 Аргументація вибору визначеного виду генетичного алгоритму	43
2.4 Висновки до другого розділу	46
3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПРОЄКТУВАННЯ...	47
3.1 Вибір комплектуючих для отримання та обробки даних	47
3.1.1 Плата мікроконтролерів	47
3.1.2 Датчик газу.....	50
3.1.3 Датчик температури.....	52
3.1.4 Датчик вогню	54
3.1.5 Датчик температури і вологості	56
3.2 З'єднання з сенсорами та передача даних	58
3.2.1 MQ-2	58
3.2.2 TMP36.....	59
3.2.3 KY-026.....	59
3.2.4 DHT11.....	60
3.3 Вибір засобів програмування.....	61

3.4 Реалізація алгоритму.....	63
3.4.1 Адаптація параметрів для групи БПЛА.....	63
3.4.2 Реалізація програмного коду.....	66
3.5 Висновки до третього розділу.....	69
4 ТЕСТУВАННЯ РОБОТИ СИСТЕМИ.....	71
4.1 Тестування роботи сенсорів.....	71
4.1.1 Тестування DHT11	71
4.1.2 Тестування KY-026	72
4.1.3 Тестування TMP36	73
4.1.4 Тестування MQ2.....	76
4.2 Тестування роботи алгоритму	77
4.3 Висновки до четвертого розділу.....	81
5 ЕКОНОМІЧНА ЧАСТИНА	83
5.1 Оцінювання наукового ефекту	83
5.2 Розрахунок витрат на здійснення науково-дослідної роботи.....	86
5.2.1 Витрати на оплату праці.....	87
5.2.2 Основна заробітна плата робітників	88
5.2.3 Відрахування на соціальні заходи	90
5.2.4 Сировина та матеріали.....	90
5.2.5 Розрахунок витрат на комплектуючі.....	91
5.2.6 Спецустаткування для наукових (експериментальних) робіт	92
5.2.7 Програмне забезпечення для наукових (експериментальних) робіт	93
5.2.8 Амортизація обладнання, програмних засобів та приміщень	94
5.2.9 Паливо та енергія для науково-виробничих цілей	95
5.2.10 Службові відрядження.....	96
5.2.11 Витрати на роботи, які виконують сторонні підприємства, установи і організації.....	97
5.2.12 Інші витрати.....	97
5.2.13 Накладні (загальновиробничі) витрати.....	98
5.3 Оцінювання важливості та наукової значимості науково-дослідної роботи	99
5.4 Висновок до п'ятого розділу.....	100
ВИСНОВКИ.....	102
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	104

Додатки.....	108
Додаток А (Обов’язковий) Технічне завдання.....	109
Додаток Б (Обов’язковий) Ілюстративна частина	113
Додаток В (Обов’язковий) Лістинг коду	121
Додаток Г (Обов’язковий) Акт впровадження системи.....	126
Додаток Д (Обов’язковий) Протокол перевірки магістерської кваліфікаційної роботи на наявність текстових запозичень.....	127

ВСТУП

Актуальність. Забруднення повітря є однією з основних екологічних проблем, що призводять до передчасної смертності в Європі. З кожним роком ця проблема стає все більш гострою через зростаючий вплив людської діяльності на довкілля. Численні дослідження вказують на зростання антропогенного впливу на якість повітря, наприклад, через збільшення викидів парникових газів, таких як вуглекислий газ, метан і діоксид азоту, внаслідок спалювання викопних палив та інших промислових процесів. Це призводить до кліматичних змін та значного погіршення стану повітря, що, у свою чергу, сприяє розвитку ряду захворювань як у людей, так і у тварин.

Постійний моніторинг якості повітря та екологічна оцінка середовища є надзвичайно важливими, і для цього активно застосовуються технологічні рішення, які забезпечують збір і аналіз даних про рівень забруднення. Однак точне визначення внеску окремих факторів у розвиток захворювань залишається складним завданням через комплексність впливу різних чинників на здоров'я людини. Для ефективної боротьби із забрудненням повітря необхідно використовувати інформаційні технології, які дозволяють ефективно збирати, зберігати та аналізувати великі обсяги даних, що забезпечує швидкий та точний аналіз екологічного стану .

Інформаційні технології набули широкого поширення у сфері моніторингу довкілля. Зокрема, безпілотні літальні апарати (БПЛА) дозволяють автоматизовано збирати дані для оперативної оцінки якості повітря. Застосування цих технологій дозволяє уникати ризиків для здоров'я людей, адже БПЛА можуть виконувати завдання у небезпечних зонах, мінімізуючи потребу у фізичній присутності людини. Це також сприяє економії часу і підвищенню точності зібраних даних.

Метою дослідження є розробка інформаційної технології для оптимізації планування траєкторій групи безпілотних літальних апаратів (БПЛА) для оцінки рівня забруднення повітря.

Для досягнення мети необхідно розв'язати наступні задачі:

1. Провести огляд існуючих інформаційних технологій для оцінки якості повітря, їх комплектуючих компонентів, дослідити аналоги, визначити їх переваги та недоліки.
2. Виконати аналіз існуючих алгоритмів, які можуть бути використані для розв'язання наведеної задачі. Обрати алгоритм для реалізації та обґрунтувати вибір.
3. Провести детальне дослідження підтипів обраного алгоритму та обрати найоптимальніший.
4. Розробити програмне забезпечення.

Об'єктом дослідження є процес розробки технології для оптимального управління групою БПЛА, що виконують моніторинг забруднення повітря.

Предмет дослідження — алгоритми і методи планування траєкторій для ефективного пересування БПЛА в зоні моніторингу.

Джерелами дослідження слугували підручники про роботу апаратного забезпечення безпілотних літальних апаратів. Також активно використовувались електронні ресурси з прикладами роботи програм генетичних алгоритмів та їх результатами.

Методами дослідження є аналіз, класифікація, узагальнення, спостереження та методи представлення результату.

Науково-технічний результат полягає у створенні технології, яка забезпечує ефективне керування польотом групи БПЛА для проведення точного та швидкого моніторингу забруднення повітря. Застосування генетичних алгоритмів для планування траєкторії групи БПЛА дозволяє зменшити час, необхідний для збору екологічних даних, шляхом оптимізації розташування дронів у просторі та мінімізації дублювання маршрутів. Це забезпечує більш ефективний моніторинг та скорочує витрати на виконання

операцій, підвищуючи якість даних та зменшуючи ризики для здоров'я людей, оскільки БПЛА можуть виконувати моніторинг у небезпечних зонах без прямої участі людини.

Практична цінність полягає у створенні комплексної системи для планування траєкторій БПЛА, яка дозволить значно прискорити процес збору та аналізу даних про забруднення повітря, забезпечуючи швидку реакцію на зміни екологічного стану.

Апробація результатів та публікації. За результатами даної роботи опубліковано доповіді на Міжнародній науково-практичній інтернет-конференції «Контроль і управління в складних системах» [1], Міжнародній науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи» [2] та прийнята до друку наукова робота у науковому журналі «Вісник Вінницького політехнічного інституту».

1 АНАЛІЗ ІСНУЮЧИХ СИСТЕМ ТА ПІДХОДІВ

У сучасному середовищі є наявна велика кількість різних систем для моніторингу забруднення повітря, які варіюються за функціональними можливостями та вартістю. Проте, їх застосування часто супроводжується низькою швидкістю роботи та недостатньою точністю отриманих даних, що значно знижує ефективність їх використання. Крім того, окремі завдання вимагають інтеграції кількох систем для досягнення точних результатів, що призводить до підвищення складності та тривалості процесу оцінки якості повітря.

1.1 Огляд існуючих систем оцінки якості повітря

Одним із найбільш поширених та точних методів оцінки якості повітря є вимірювання концентрації таких забруднювальних речовин, як оксиди вуглецю (CO), діоксид сірки (SO₂), оксиди азоту (NO_x), тверді частинки (PM₁₀, PM_{2.5}) та інших шкідливих домішок. Для виконання цих вимірювань застосовуються різні технології, серед яких:

Станції моніторингу якості повітря: це наземні пристрої, що зазвичай встановлюються на будівлях або спеціальних майданчиках. Вони здатні безперервно збирати дані про рівень концентрації забруднювальних речовин у конкретному регіоні та забезпечувати моніторинг якості повітря в режимі реального часу завдяки підключенню до Інтернету. Однак такі станції мають обмежену мобільність і можуть не охоплювати всі ділянки території через свою стаціонарність.

Аерозонди: це невеликі пристрої, які здатні підніматися на різні висоти для збору зразків повітря, які потім аналізуються на наявність забруднювальних речовин у лабораторних умовах. Завдяки своїй здатності

рухатися у вертикальному напрямку, аерозонди можуть охоплювати ширший простір, що дозволяє отримати різноманітні дані про якість повітря на різних висотах. Однак вони є найменш ефективними серед інших методів моніторингу через обмежену тривалість польоту, невелику відстань покриття, а також необхідність лабораторного аналізу зразків, що значно уповільнює процес отримання даних.

Наземні дрони та безпілотні літальні апарати (БПЛА): з впровадженням технологій машинного навчання та штучного інтелекту, дрони стали одними з найсучасніших інструментів для оцінки якості повітря. Вони оснащуються спеціалізованими сенсорами, що дозволяють в реальному часі збирати дані про хімічний склад атмосфери, рівень концентрації шкідливих речовин та інші параметри.

Переваги технологій машинного навчання порівняно з іншими технологіями моніторингу полягають у їхній здатності до оперативного збору даних з великої площі та висоти. Завдяки своїй мобільності, дрони можуть швидко змінювати траєкторію відповідно до задачі, що робить їх ідеальними для моніторингу важкодоступних місць, де стаціонарні пристрої не можуть виконувати вимірювання. Їхні сенсори збирають дані з високою точністю, забезпечуючи вірогідність вимірювань навіть у разі швидких змін в якості повітря [3].

Крім того, дрони можна програмувати для автономного виконання завдань, що дозволяє мінімізувати людський фактор і підвищує точність та відтворюваність результатів. Це робить їх ефективним інструментом для екологічного моніторингу, оскільки дрони можуть збирати дані в безперервному режимі, незалежно від людської присутності, що сприяє отриманню більш повної картини стану повітря у заданій області.

Таким чином, впровадження технологій машинного навчання для оцінки якості повітря демонструє високу ефективність та перспективність у порівнянні з традиційними методами. Їх використання поступово

розширюється завдяки зручності та точності даних, які вони можуть надавати у процесі моніторингу.

1.1.1 Наземні системи оцінки якості повітря

Мобільні наземні станції та наземні роботи є перспективними технологіями для моніторингу якості повітря, однак вони мають обмежені можливості в порівнянні з деякими новітніми технологіями. Ці системи забезпечують збір даних на різних рівнях ефективності та в специфічних умовах, де стаціонарні або важкодоступні ділянки вимагають оперативного чи автономного моніторингу.

Мобільні станції широко використовуються у випадках, коли необхідний короткостроковий моніторинг у міських або промислових зонах. Наприклад, на будівельних майданчиках, поблизу транспортних артерій або у зонах із підвищеними промисловими викидами. Мобільні системи можуть бути розгорнуті в межах певної території для виявлення аномалій у складі повітря або аналізу його якості у разі зміни викидів чи умов середовища. Вони часто використовуються для збору даних під час аварійних ситуацій, таких як викиди небезпечних речовин, для оперативної оцінки ступеня забруднення.

Наземні роботи часто використовуються у зонах, де умови середовища роблять неможливим розміщення стаціонарних або мобільних станцій, наприклад, на індустріальних підприємствах, у кар'єрах або біля об'єктів підвищеного ризику. Завдяки своїй мобільності, роботи можуть збирати більш точні дані про забруднення повітря на різних висотах і ділянках, що є важливим для детального аналізу якості повітря на локальних територіях [4].

Попри свою автономність, наземні роботи та мобільні станції також мають обмежений радіус дії, залежний від часу роботи акумуляторів та умов місцевості. Їх застосування зазвичай локалізується у чітко визначених

промислових або міських зонах, де доступність території є обмеженою, що не дозволяє покривати великі території.



Рисунок 1.1 – Будова наземного робота для оцінки забруднення повітря

На рисунку 1.1 зображено апаратну будову типового робота для оцінки якості повітря.

- Ультразвуковий анемометр — використовується для вимірювання швидкості та напрямку вітру. Завдяки цьому компоненту можна враховувати вплив метеорологічних факторів на поширення забруднень і їх концентрацію в різних точках вимірювання.
- Детектор забруднення повітря — це прилад для виявлення різних видів забруднень у повітрі, таких як оксиди вуглецю, азоту, сірки, тверді частинки (PM10, PM2.5), леткі органічні сполуки (VOC) та інші небезпечні речовини. Детектор забезпечує точні дані про концентрацію забруднювачів в атмосфері, що є основою для оцінки якості повітря.
- Автономна система керування транспортним засобом — забезпечує автономний рух робота без участі оператора. Вона включає GPS-навігацію для точного маршрутування, лідар або ультразвукові сенсори

для уникнення перешкод, а також блок обробки даних для прийняття рішень під час руху.

- Вбудований високопотужний всенаправлений Wi-Fi модуль — забезпечує бездротовий зв'язок робота з іншими пристроями або центром управління для передачі даних у реальному часі. Це дозволяє оперативно отримувати результати вимірювань та віддалено контролювати роботу системи.

Така конфігурація є базовою для типового робота, що використовується для моніторингу якості повітря. Однак існує ряд додаткових компонентів, які можуть бути інтегровані для розширення можливостей системи та підвищення точності збору даних [5]:

- Сенсори температури та вологості — дозволяють фіксувати поточні кліматичні умови, що можуть впливати на точність результатів вимірювань, зокрема на процес розсіювання забруднень у повітрі. Ці параметри є важливими для більш точного аналізу забруднювальних речовин у різних умовах.
- Акумулятор високої ємності — забезпечує тривале автономне функціонування робота. Це особливо корисно для операцій на великих територіях або в умовах, коли доступ до робота для підзарядки є обмеженим. Більша ємність акумулятора дозволяє продовжити час безперервної роботи системи.
- Камера з високою роздільною здатністю — використовується для візуальної фіксації навколишнього середовища та моніторингу можливих джерел забруднення. Відео та фотофіксація можуть допомогти в ідентифікації джерел забруднення або забезпечити додаткові дані для аналізу екологічної ситуації.
- Мікроконтролер для обробки даних — є центральним процесором системи, який відповідає за інтеграцію та обробку даних від різних сенсорів. Він дозволяє роботизованій системі швидко реагувати на зміни в середовищі, обробляти дані в реальному часі та передавати їх до

центрального сервера або оператора для подальшого аналізу. Мікроконтролери, що частіше використовуються в наземних роботах: ESP32 — завдяки вбудованим Wi-Fi і Bluetooth модулям, цей мікроконтролер є популярним у наземних системах. Він дозволяє передавати дані з сенсорів у реальному часі без потреби в зовнішніх модулях зв'язку; Raspberry Pi — його здатність обробляти великі обсяги даних робить його ідеальним для наземних роботів, які можуть бути підключені до складних сенсорних систем. Raspberry Pi часто використовується для наземних роботів, які працюють на великих територіях і збирають багато різних типів даних; STM32 — відомі своєю високою продуктивністю та енергоефективністю, ці мікроконтролери добре підходять для наземних роботів, які повинні працювати протягом тривалого часу без потреби частих перезарядок.

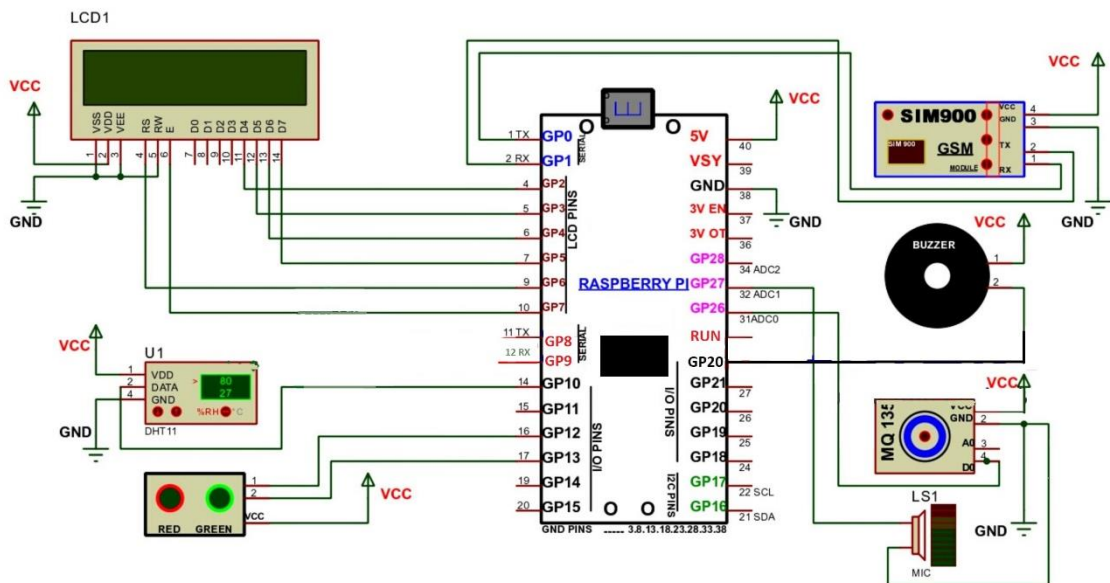


Рисунок 1.2 – Приклад мікроконтролера RPi у наземному роботі з використанням мобільного зв'язку (GSM)

Ці додаткові компоненти дозволяють розширити функціональність роботизованої системи, підвищуючи точність вимірювань, автономність і можливість інтеграції в різні екологічні проекти.

1.1.2 Безпілотні системи оцінки якості повітря

БПЛА (безпілотні літальні апарати) для оцінки якості повітря є однією з найсучасніших технологій для моніторингу атмосфери. Вони оснащуються різноманітними сенсорами, які дозволяють в режимі реального часу збирати точні дані про рівень забруднення, аналізуючи концентрації шкідливих речовин. Основні переваги БПЛА:

- Висока точність вимірювань. Завдяки можливості польоту на різних висотах, БПЛА можуть отримувати дані з більшою точністю, охоплюючи різні шари атмосфери. Це дозволяє проводити детальний аналіз забруднення на різних висотах, чого важко досягти з наземними системами.
- Широкий радіус дії. БПЛА можуть охоплювати великі території за короткий проміжок часу, завдяки чому вони ідеально підходять для моніторингу регіонів з великим географічним розмахом, таких як національні парки або промислові зони, де забруднення може мати широку площу розподілу.
- Швидка реакція на зміни. БПЛА можуть бути швидко розгорнуті у разі виявлення екологічних загроз, що забезпечує своєчасний збір даних та оперативну реакцію на критичні ситуації, такі як викиди небезпечних газів.
- Надійність у важких умовах. БПЛА можуть працювати в умовах, де наземні станції не можуть бути розміщені, наприклад, у віддалених або небезпечних районах, де фізична присутність людей обмежена.
- Гнучкість в налаштуваннях сенсорів. БПЛА можуть бути оснащені різними типами сенсорів в залежності від конкретних завдань моніторингу, такими як газові детектори, метеорологічні сенсори та

камери високої роздільної здатності. Це дозволяє їм адаптуватися до різних сценаріїв оцінки якості повітря.

Область застосування: БПЛА широко використовуються для моніторингу повітря як в урбанізованих зонах, так і в природних середовищах, зокрема у важкодоступних місцях, де звичайні наземні системи мають обмеження. Вони застосовуються для екологічного моніторингу, під час аварійних ситуацій (наприклад, витоку газів), для оцінки впливу промислових об'єктів на довкілля та при дослідженні кліматичних змін.

Основні функції компонентів дронів для моніторингу якості повітря полягають у забезпеченні ефективної обробки, збору та передачі даних, а також у надійній управлінні польотом і сенсорами, які є необхідними для аналізу повітря [6].

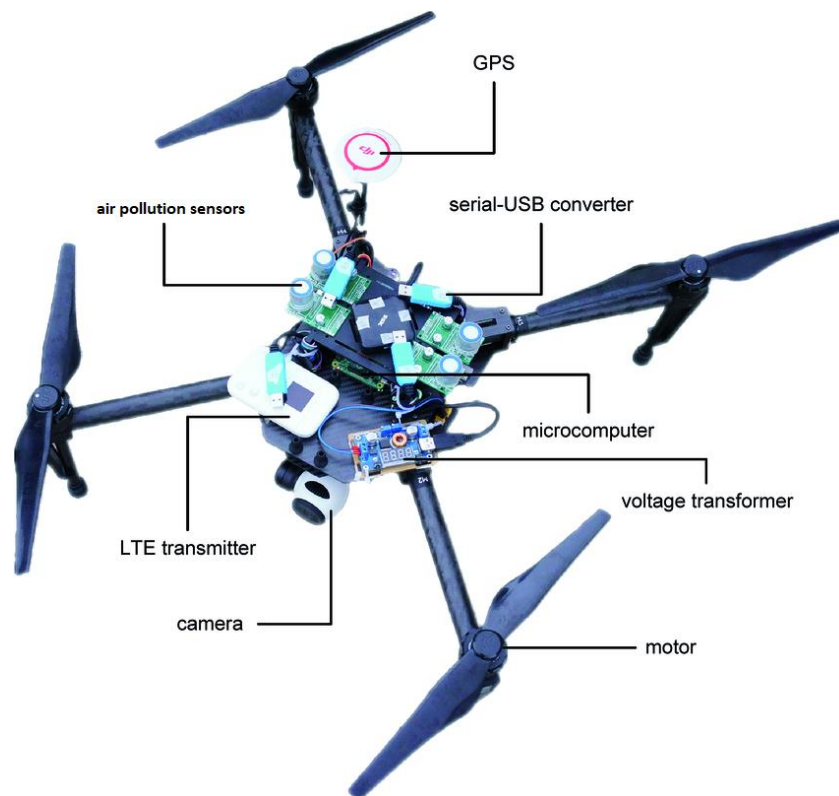


Рисунок 1.3 – Приклад типової конструкції БПЛА для оцінки якості повітря

На рисунку 1.3 описано конфігурацію апаратного забезпечення типового дрона для дослідження забруднення повітря. Основні елементи конструкції включають [7]:

- GPS – цей пристрій забезпечує навігацію, дозволяючи визначати координати дрона в просторі та контролювати його маршрут.
- Повітряні датчики забруднення (Air Pollution Sensors) – ці сенсори вимірюють рівень забруднення повітря, збираючи дані про різні шкідливі частинки та гази, що дозволяє отримувати актуальну інформацію про екологічний стан.
- Конвертер Serial-USB – використовується для передачі даних між компонентами дрона через послідовний порт та USB, що забезпечує зручний обмін інформацією.
- Мікрокомп'ютер – відповідає за управління всіма основними функціями БПЛА, обробку даних, отриманих від сенсорів, та комунікацію з іншими пристроями. Мікроконтролери, що часто використовуються в БПЛА:

Arduino — цей мікроконтролер популярний завдяки своїй простоті у використанні та широкому спектру доступних бібліотек, ідеально підходить для управління різними сенсорами та компонентами дрона. Він дозволяє легко інтегрувати різні датчики забруднення повітря та забезпечує базові функції керування польотом; Jetson Nano — потужний мікрокомп'ютер, розроблений для виконання завдань штучного інтелекту, особливо корисний для обробки великих обсягів даних у реальному часі, зокрема для комп'ютерного зору, що дозволяє БПЛА ідентифікувати джерела забруднення або аналізувати дані з камер; Pixhawk — один з найпопулярніших мікроконтролерів для безпілотних літальних апаратів, що забезпечує високу стабільність польоту і можливості для налаштування, підтримує широкий спектр сенсорів та модулів, що робить його універсальним рішенням для різних видів дронів. Він також дозволяє реалізувати автономні місії та моніторинг без втручання оператора [8].

- Передавач LTE – забезпечує передачу даних на великі відстані через мобільну мережу LTE, що дозволяє отримувати інформацію в режимі реального часу.
- Камера – використовується для відео- або фотофіксації місцевості, що може бути корисно для візуального моніторингу джерел забруднення.
- Перетворювач напруги (Voltage Transformer) – регулює подачу електроенергії до різних компонентів дрона, що забезпечує стабільну роботу всіх систем.
- Мотор – відповідає за рух та стабільність дрона під час польоту, забезпечуючи необхідну маневреність для збору даних з різних точок.

БПЛА передають зібрані дані через різні мережі, такі як Wi-Fi, 3G/4G та Bluetooth. Це забезпечує можливість передачі інформації на великі відстані, що дозволяє оперативно отримувати дані для подальшого аналізу на сервері. Завдяки такій інтеграції, дані про якість повітря можуть бути оброблені в реальному часі, що підвищує ефективність моніторингу та дозволяє швидше реагувати на зміни екологічної ситуації.

Messages:

(a) - Transmission data

(b) - Analyze/Act

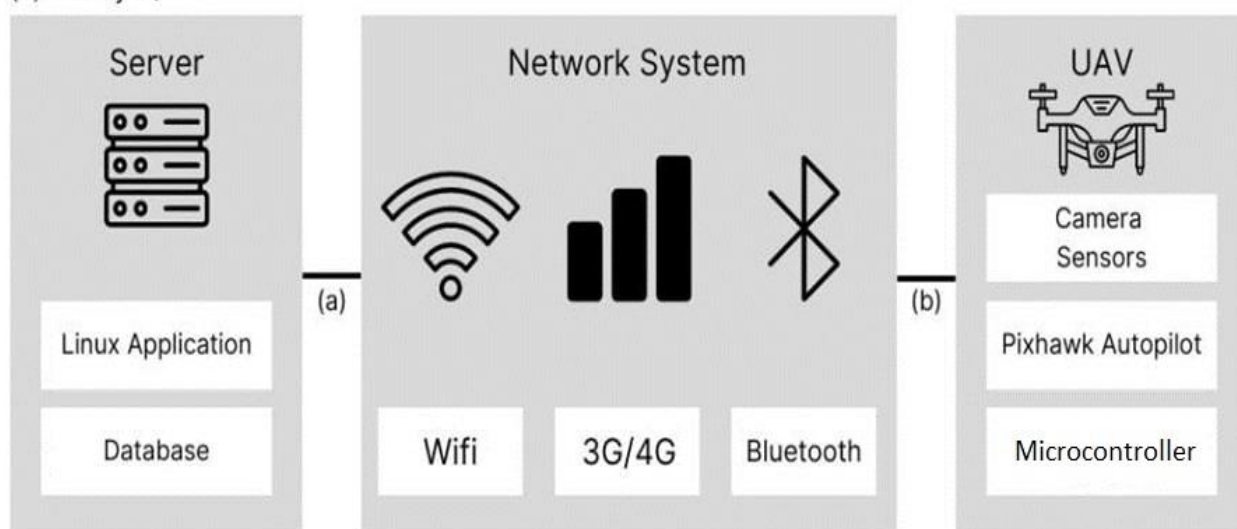


Рисунок 1.4 – Взаємодія БПЛА з сервером через мережі WiFi, 3G/4G, Bluetooth

1.2 Аналіз функціонування алгоритмів сучасних систем

У сучасних умовах динамічної зміни рівня забруднення повітря важливо забезпечити можливість швидкого та точного розрахунку оптимальних маршрутів в режимі реального часу. Для цього використовуються різні алгоритми оптимізації, які враховують такі фактори, як час, відстань, енергоспоживання, а також обмеження, пов'язані з безпекою та регуляторними вимогами.

Мурашині алгоритми є прикладом сучасного підходу до вирішення складних оптимізаційних задач. Вони ґрунтуються на моделюванні поведінки мурах, що прокладають шляхи до джерел їжі, використовуючи феромони для позначення пройдених маршрутів. Кожен агент (мураха) рухається в просторі шляхів, залишаючи феромони, а ймовірність вибору напрямку залежить від їх концентрації. Однак ці алгоритми мають недоліки, зокрема складність налаштувань, обчислювальну витратність та схильність до застрягання в локальних максимумах [9]. Це обмежує їх ефективність при вирішенні глобальних задач оптимізації. Блок-схема мурашиного алгоритму зображена на рисунку 1.5.

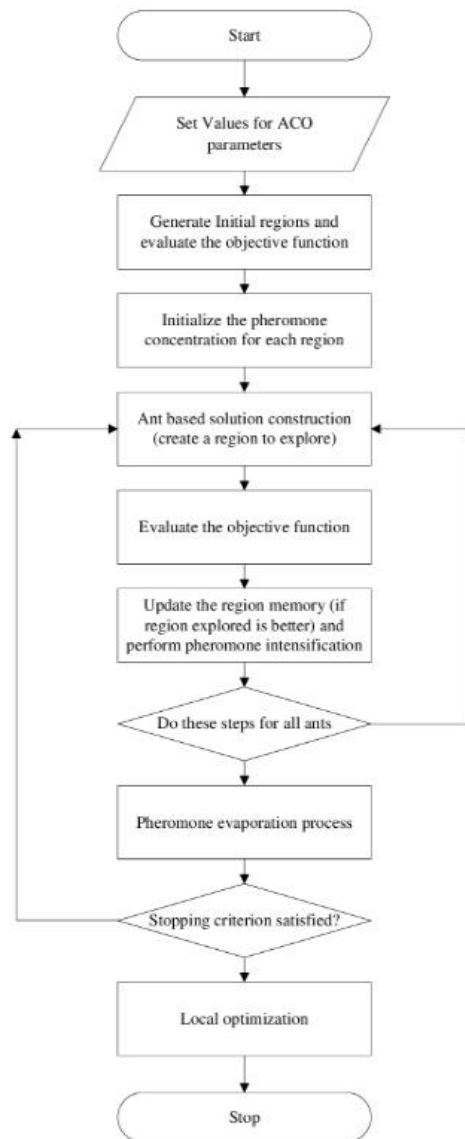


Рисунок 1.5 – Блок-схема алгоритму мурашиного алгоритму

Градiєнтний спуск – один з найпоширенiших методiв для пошуку мiнiмуму чи максимуму функцiй. Алгоритм iтеративно рухається у напрямку, протилежному до градiєнту функцiї, поступово наближаючись до оптимального рiшення. Його перевагами є простота реалiзацiї та можливiсть застосування до широкого кола задач [10]. Водночас, цей метод має серйознi недолiки: чутливiсть до розмiру кроку, ризик застрягання в локальних мiнiмумах та нездатнiсть ефективно працювати з нерiвними або "шумними" функцiями. Блок-схема градiєнтного спуску зображена на рисунку 1.6.

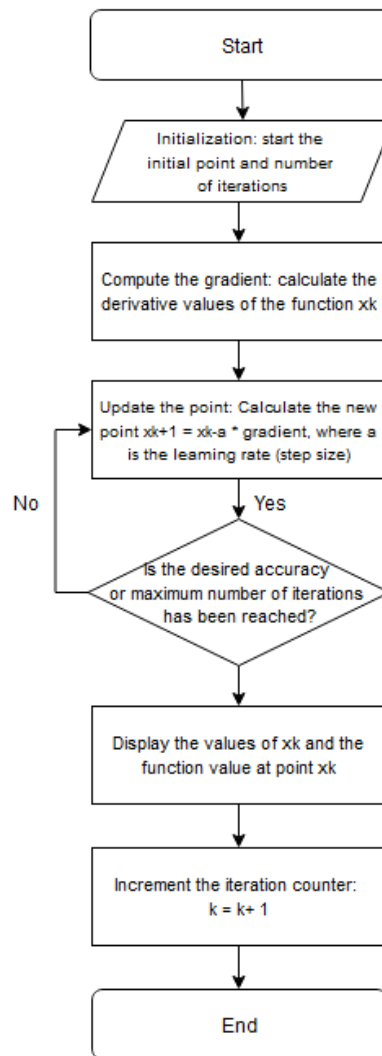


Рисунок 1.6 – Блок-схема алгоритму градієнтного спуску

Алгоритм A^* застосовується для пошуку найкоротших шляхів у графах і використовує евристичну оцінку для прискорення пошуку [11]. Це робить його корисним для завдань маршрутизації з відомими обмеженнями та вагами. Однак, обчислювальна складність та залежність від точності евристичної оцінки можуть обмежувати його ефективність при роботі з великими графами або складними умовами. Блок-схема алгоритму A^* зображена на рисунку 1.7

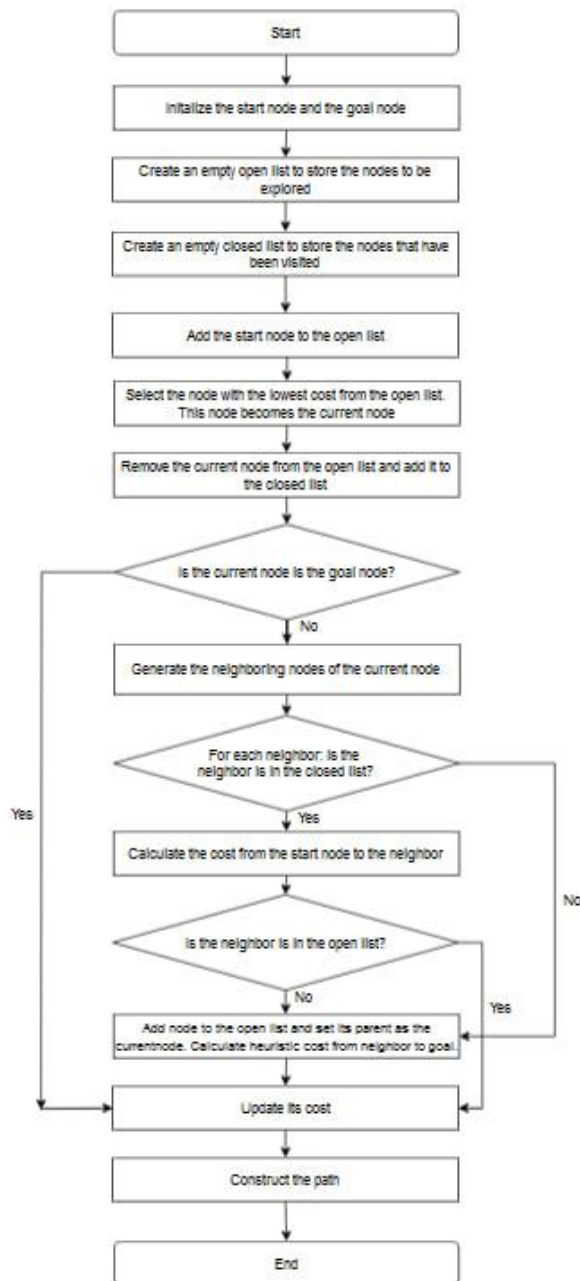


Рисунок 1.7 – Блок-схема роботи алгоритму A*

Алгоритм оптимізації рою частинок (Particle Swarm Optimization, PSO) є одним із сучасних підходів до вирішення задач оптимізації. Він імітує поведінку рою частинок у просторі пошуку, де кожна частинка представляє потенційний розв'язок задачі. Алгоритм ґрунтується на взаємодії між частинками та спільному обміні інформацією про найкращі знайдені розв'язки. Кожна частинка рухається у пошуковому просторі, оновлюючи свою швидкість та позицію на основі індивідуального досвіду (найкраща позиція частинки) та глобального досвіду (найкраща позиція всього рою) [12].

Основною перевагою PSO є його здатність знаходити глобальні оптимуми в складних, нелінійних просторах. Алгоритм використовує когнітивні та соціальні компоненти для регулювання руху частинок. Однак, PSO має й недоліки, такі як схильність до застрягання в локальних оптимумах і залежність результатів від правильного налаштування параметрів. Блок-схема алгоритму PSO зображена на рисунку 1.8

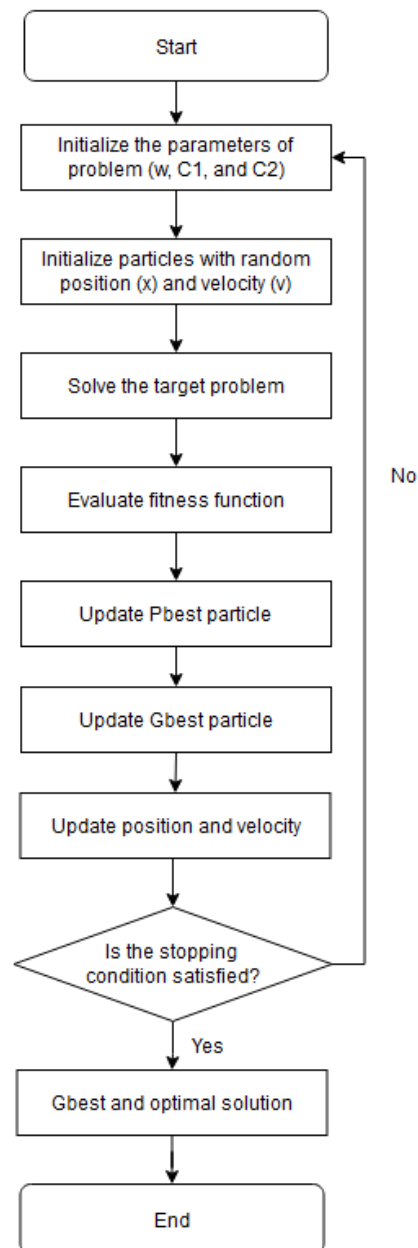


Рисунок 1.8 – Блок-схема роботи алгоритму PSO

Генетичні алгоритми є іншим сучасним методом оптимізації, що імітує еволюційні процеси. Вони використовують операції селекції, кросинговеру та мутації для пошуку оптимальних рішень. Кожен розв'язок представляється як індивід у популяції, і його якість визначається за допомогою функції фітнесу. Алгоритм працює ітераційно, де кожне нове покоління базується на найкращих рішеннях попереднього [13].

Головною перевагою генетичних алгоритмів є їх здатність уникати локальних мінімумів та ефективно досліджувати простір пошуку. Водночас, вони мають високу обчислювальну складність та залежність від параметрів налаштування, що може впливати на якість кінцевих результатів. Блок-схема генетичного алгоритму зображена на рисунку 1.9

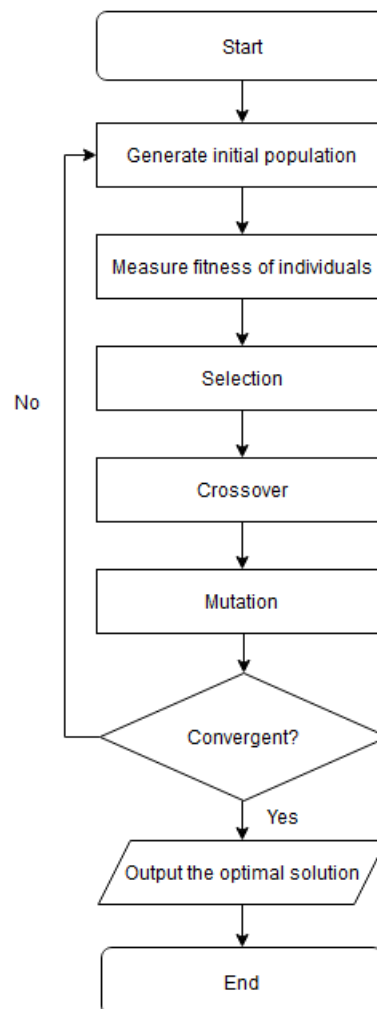


Рисунок 1.9 – Блок-схема роботи генетичного алгоритму

1.3 Оцінка наявних аналогів системи

Наземні роботи та мобільні станції мають обмежене застосування в моніторингу якості повітря через їхні конструктивні недоліки. Вони не здатні забезпечувати тривалий моніторинг на великих площах або в складних умовах. Основні труднощі полягають у відсутності гнучкості, низькій швидкості реагування та обмеженій здатності охоплювати значні території з достатньою точністю. Це обмежує їх ефективність, особливо в важкодоступних або віддалених місцях, де гнучкість і швидкість збору даних є критичними.

Безпілотні літальні апарати (БПЛА), на відміну від наземних систем, мають високу маневровість і здатність здійснювати моніторинг у режимі реального часу. Вони можуть охоплювати великі території за короткий час, забезпечуючи точність збору даних навіть у складних умовах. Завдяки цим перевагам БПЛА є більш ефективним рішенням для моніторингу якості повітря на великих площах та у важкодоступних місцевостях.

Алгоритми, такі як A^* і градієнтний спуск, також мають свої обмеження для вирішення задачі планування траєкторій БПЛА для моніторингу забруднення повітря. A^* є детермінованим методом пошуку, що гарантує знаходження оптимального шляху, однак він сильно залежить від точності моделі середовища та обчислювальних ресурсів. У випадку великих або динамічних середовищ, цей алгоритм може бути неефективним через потребу в значних обчислювальних ресурсах і часу. Градієнтний спуск, хоча і ефективний для мінімізації функцій у просторах, часто застрягає в локальних мінімумах та потребує точного визначення похідних, що ускладнює його застосування в складних і невизначених середовищах, які характерні для задач моніторингу якості повітря за допомогою БПЛА.

Ройові та мурашині алгоритми, хоч і мають певні переваги у вирішенні задач оптимізації, мають низку обмежень для планування траєкторій БПЛА. Наприклад, алгоритм оптимізації рою частинок може застрягати в локальних

оптимумах і сильно залежить від налаштування параметрів, що може впливати на точність та ефективність. Мурашині алгоритми, в свою чергу, також схильні до локальних оптимумів і мають складну конфігурацію, що робить їх менш придатними для задач глобального планування в багатовимірних просторах.

Генетичні алгоритми виділяються серед інших завдяки своїй здатності проводити глобальну оптимізацію навіть у складних багатовимірних пошукових просторах. Їх еволюційний підхід забезпечує ефективне дослідження простору рішень та здатність уникати локальних мінімумів, що є критичним при плануванні траєкторій БПЛА для моніторингу забруднення повітря. Хоча генетичні алгоритми також мають певні обчислювальні вимоги та потребують належної конфігурації параметрів (розмір популяції, ймовірність мутацій), їх гнучкість та ефективність у вирішенні таких задач роблять їх найбільш оптимальним для цієї задачі.

1.4 Висновки до першого розділу

У цьому розділі було проведено аналіз існуючих систем та підходів для оцінки якості повітря. Було досліджено сучасні системи оцінки якості повітря та алгоритми, які вони використовують. В ході аналізу було виявлено, що безпілотні літальні апарати (БПЛА) є більш ефективним рішенням завдяки їх високій маневровості, можливості проводити моніторинг у режимі реального часу та збору даних на великих територіях, у порівнянні з наземними аналогами.

Алгоритми A^* і градієнтний спуск виявляються менш ефективними для планування траєкторій БПЛА через обмеження в масштабованості та здатності працювати в складних багатовимірних просторах, що робить їх менш придатними для оптимізації задач моніторингу якості повітря. Ройові та мурашині алгоритми, попри деякі переваги в оптимізації, мають суттєві

обмеження для планування траєкторій БПЛА. Ройові алгоритми можуть застрягати в локальних оптимумах і сильно залежать від налаштування параметрів, а мурашині алгоритми мають складну конфігурацію та теж схильні до локальних оптимумів, що обмежує їх ефективність у багатовимірних просторах. Генетичні алгоритми вирізняються своєю здатністю до глобальної оптимізації в складних багатовимірних просторах. Їх еволюційний підхід дозволяє уникати локальних мінімумів, що важливо для планування траєкторій при моніторингу забруднення повітря. Попри потребу в налаштуванні параметрів і обчислювальні вимоги, їх гнучкість та ефективність роблять їх найкращим вибором для цієї задачі.

Таким чином, застосування БПЛА у поєднанні з генетичними алгоритмами надає оптимальний баланс між точністю, швидкістю та гнучкістю, що робить цю систему найкращим варіантом для вирішення завдання оцінки якості повітря.

2 ПОРІВНЯЛЬНЕ ДОСЛІДЖЕННЯ СТРУКТУРИ ГЕНЕТИЧНИХ АЛГОРИТМІВ

2.1 Принцип роботи генетичного алгоритму

Генетичні алгоритми належать до класу еволюційних методів, що базуються на ідеях природного відбору, кросинговеру та мутації для пошуку оптимальних рішень у складних просторах пошуку. Вони відтворюють принципи біологічної еволюції, де найбільш ефективні рішення мають вищі шанси на виживання та передають свої властивості наступним поколінням. Цей процес багаторазово повторюється, що призводить до поступового покращення результатів та наближення до оптимуму [14].

Основними етапами генетичних алгоритмів є:

- Ініціалізація: Генерація початкової популяції можливих рішень випадковим чином у просторі пошуку. Кожне рішення кодується як геном, який відображає параметри, наприклад, траєкторії БПЛА.
- Оцінка придатності: Кожен генотип оцінюється через фітнес-функцію, яка визначає його ефективність. Для траєкторій БПЛА фітнес-функція може враховувати точність навігації, швидкість, висоту та інші характеристики польоту.
- Відбір: Найкращі індивіди (розв'язки) обираються для репродукції на основі їхньої фітнес-оцінки, що підвищує ймовірність того, що найефективніші рішення будуть передані в наступне покоління.
- Кросинговер: Обрані індивіди обмінюються частинами свого генетичного матеріалу, що призводить до створення нових потомків зі змішаними характеристиками.
- Мутація: Випадкові зміни в генетичному матеріалі нащадків дозволяють зберігати різноманітність популяції та уникати застрягання в локальних мінімумах.

- **Заміна:** Нові індивіди замінюють частину попередньої популяції, що сприяє прогресивному поліпшенню розв'язків з кожним поколінням.
- **Зупинка:** Алгоритм виконується доти, доки не досягнуто певної умови, наприклад, за кількістю поколінь або збіжності рішень.

Фітнес-функція є ключовою складовою алгоритму, яка визначає, наскільки добре кожен індивід виконує поставлене завдання. Її форма залежить від конкретної задачі: у задачах максимізації шукається збільшення значення фітнесу, а у задачах мінімізації — його зменшення.

Основні переваги генетичних алгоритмів у контексті планування траєкторій БПЛА включають:

- **Глобальна оптимізація:** Вони можуть ефективно досліджувати простір рішень, уникаючи локальних мінімумів.
- **Адаптивність:** Алгоритм легко підлаштовується під змінні умови, що особливо корисно для моніторингу забруднення повітря, де рівень забруднення може змінюватися.
- **Можливість паралелізації:** Генетичні алгоритми легко масштабуються на багатоядерних або розподілених обчислювальних системах, що дозволяє значно прискорити процес обчислень.

Серед недоліків можна виділити:

- **Високу обчислювальну складність:** Велика кількість ітерацій може вимагати значних ресурсів.
- **Залежність від початкових умов:** Результати алгоритму можуть суттєво змінюватися в залежності від початкової популяції та налаштувань.
- **Відсутність гарантій оптимального рішення:** Генетичні алгоритми не завжди гарантують знаходження глобально оптимального розв'язку, особливо для складних задач з великою кількістю обмежень.

Алгоритм можна узагальнити наступним чином:

- **Ініціалізація параметрів:** задаються розмір популяції, ймовірність кросинговеру і мутації.
- **Генерація початкової популяції.**

- Оцінка «fitness» кожного індивіда.
- Повторення наступних кроків до досягнення критерію зупинки:
 - Відбір батьків для кросинговеру.
 - Виконання кросинговеру для отримання потомства.
 - Застосування мутації до потомків.
 - Оцінка «fitness» потомків.
 - Формування нового покоління.
- Виведення найкращого рішення.

Такий підхід забезпечує еволюційне покращення рішень з кожною ітерацією, поступово наближаючись до оптимуму.

2.2 Огляд підвидів генетичних алгоритмів

Існує багато варіацій генетичних алгоритмів, які відрізняються за способами реалізації основних етапів алгоритму. Вони можуть включати модифікації методів відбору, кросинговеру чи мутації, а також додаткові механізми для покращення ефективності пошуку. Такі варіації спрямовані на адаптацію алгоритму до специфічних задач, забезпечуючи краще дослідження простору пошуку або швидшу збіжність до оптимального рішення.

2.2.1 Генетичні алгоритми з множинним критерієм

Генетичні алгоритми з множинним критерієм використовуються для вирішення складних задач, де необхідно одночасно оптимізувати кілька показників, що мають рівнозначне значення для остаточного результату. На відміну від традиційних генетичних алгоритмів, які націлені на пошук єдиного оптимального рішення за одним критерієм, багатокритеріальні алгоритми зосереджені на пошуку компромісних рішень, де враховується декілька

незалежних критеріїв. Основна концепція полягає у формуванні так званої Парето-оптимальної множини, де кожне рішення є таким, що не може бути покращене за одним із критеріїв без одночасного погіршення за іншим. Це дозволяє користувачам обирати найбільш збалансовані рішення відповідно до їхніх пріоритетів [15].

Однією з головних переваг багатокритеріальних генетичних алгоритмів є їх здатність обробляти задачі з конфліктуючими цілями, пропонуючи широкий спектр варіантів для аналізу. Крім того, такі алгоритми забезпечують можливість одночасного дослідження великого простору рішень, що підвищує ймовірність знаходження глобальних оптимумів. Завдяки цьому вони є незамінними в задачах, де важливі кілька параметрів, таких як планування траєкторії БПЛА з урахуванням точності навігації та економії палива або часу польоту.

Проте, такі алгоритми мають і певні недоліки. Одним із них є підвищена обчислювальна складність, оскільки оптимізація за кількома критеріями вимагає значно більшого обсягу обчислень порівняно з одноцільовими алгоритмами. Крім того, багатокритеріальні підходи можуть генерувати надто велику кількість компромісних рішень, що ускладнює вибір оптимального з них без додаткових критеріїв або експертної оцінки.

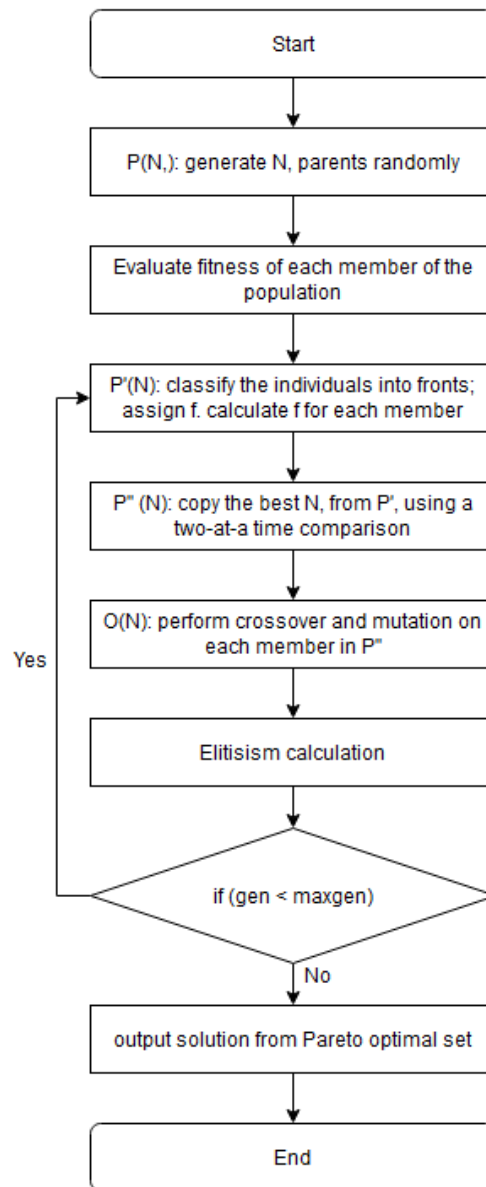


Рисунок 2.1 – Алгоритм генетичного алгоритму з множинними критеріями, що працює з вихідними даними у формі Парето-оптимальної множини

2.2.2 Паралельні генетичні алгоритми

Паралельні генетичні алгоритми значно розширюють можливості стандартних генетичних алгоритмів завдяки розподілу обчислювальних завдань між кількома процесорами або обчислювальними вузлами. Такий підхід дозволяє одночасно виконувати кілька операцій, зокрема обчислення

фітнес-функцій, відбір, кросинговер та мутації в різних частинах популяції, що значно прискорює загальний процес оптимізації. Особливо це важливо для задач, що потребують великих обчислювальних ресурсів або оперують із великими популяціями рішень [16].

Однією з основних переваг паралельних генетичних алгоритмів є їх здатність суттєво знижувати час обчислень за рахунок одночасного виконання операцій. Вони також можуть більш ефективно досліджувати простір пошуку, розподіляючи популяцію на кілька підпопуляцій, що працюють незалежно, з періодичним обміном інформацією між ними. Це сприяє кращій диверсифікації пошуку та знижує ризик застрягання в локальних мінімумах. Додатково, такі алгоритми дозволяють масштабуватися відповідно до доступних обчислювальних ресурсів, використовуючи багатоядерні процесори або розподілені обчислювальні системи.

Однак, паралельні генетичні алгоритми мають і свої недоліки. По-перше, збільшення кількості паралельних процесів може призводити до складностей у синхронізації та обміні даними між вузлами, що може негативно впливати на ефективність роботи алгоритму. По-друге, значний розподіл обчислень може вимагати спеціалізованого програмного забезпечення та інфраструктури, що підвищує складність реалізації таких алгоритмів. Нарешті, хоча паралельні алгоритми можуть швидше знаходити рішення, вони також потребують збільшених обчислювальних ресурсів, що не завжди є доступним.

Таким чином, паралельні генетичні алгоритми забезпечують значні переваги в швидкодії та ефективності, особливо для складних задач, але вимагають додаткових ресурсів і складніших механізмів керування процесами.

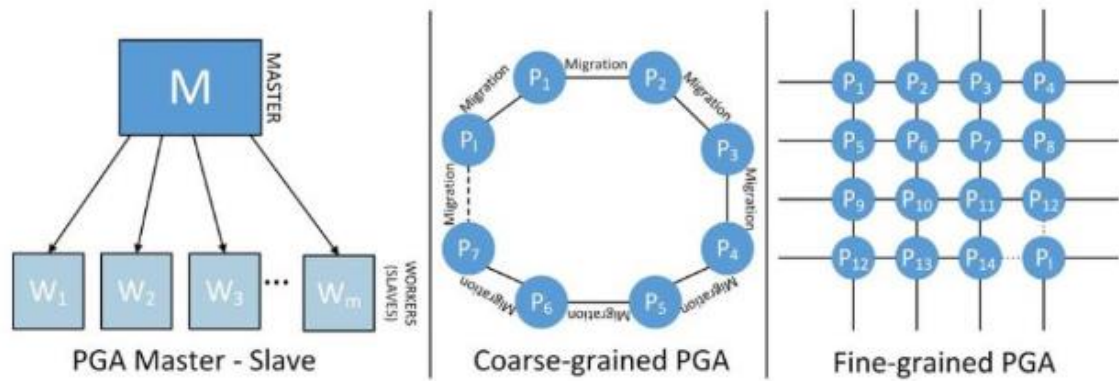


Рисунок 2.2 – Приклад різних типів залежностей у процесі паралелізації генетичних алгоритмів

2.2.3 Адаптивні генетичні алгоритми

Адаптивні генетичні алгоритми впроваджують механізми, що динамічно змінюють параметри алгоритму в процесі його виконання, реагуючи на специфіку задачі та умови середовища. У звичайних генетичних алгоритмах такі параметри, як розмір популяції, ймовірність кросинговеру або мутації, зазвичай залишаються незмінними протягом всього процесу оптимізації. В адаптивних же версіях ці параметри можуть автоматично коригуватися, щоб краще відповідати змінним умовам або етапам пошуку. Наприклад, якщо на початкових етапах пошуку необхідно більше варіацій для дослідження простору рішень, ймовірність мутації може бути вищою, а пізніше, коли наближається збіжність, її можна зменшити для стабілізації результатів [17].

Основною перевагою адаптивних генетичних алгоритмів є їх здатність самостійно підлаштовуватися до задач з мінливими характеристиками, такими як складні обмеження або непередбачувані зміни в середовищі. Це робить алгоритм більш гнучким і здатним швидше пристосовуватися до нових умов, що суттєво підвищує його ефективність і ймовірність знаходження оптимальних рішень. Адаптивні алгоритми також можуть зменшувати ризик застрягання у локальних оптимумах, оскільки вони здатні налаштовувати

параметри для більш глибокого дослідження простору рішень у складних випадках.

Недоліки таких алгоритмів: додаткові механізми адаптації можуть ускладнювати реалізацію алгоритму, оскільки потрібно ретельно розробити стратегію зміни параметрів, щоб вона була ефективною і не погіршувала роботу. Крім того, підвищена складність алгоритму може призвести до збільшення обчислювальних витрат і часу роботи. Також є ризик, що неправильне налаштування механізмів адаптації може негативно вплинути на ефективність алгоритму або навіть зробити його нестабільним.

Таким чином, адаптивні генетичні алгоритми надають значні переваги у гнучкості та ефективності пошуку оптимальних рішень в умовах змінних характеристик задач, проте потребують обережного налаштування і вимагають більших обчислювальних ресурсів для реалізації.

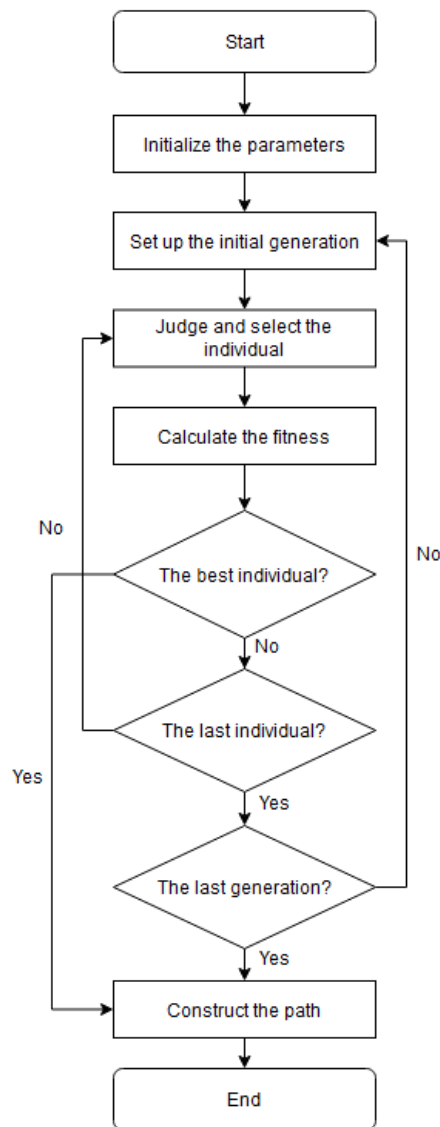


Рисунок 2.3 – Опис алгоритму адаптивного генетичного алгоритму з удосконаленими механізмами мутації та кросинговеру

2.2.4 Генетичні алгоритми з використанням додаткових евристик

Генетичні алгоритми з використанням додаткових евристик поєднують класичні принципи генетичних алгоритмів із додатковими правилами або методами, які допомагають покращити процес пошуку оптимальних рішень. Евристики, у даному контексті, виступають як спеціалізовані підходи, що можуть спрямовувати або прискорювати пошук, ґрунтуючись на певних знаннях або припущеннях про структуру проблеми. Наприклад, можна

використовувати локальні пошукові методи разом із генетичним алгоритмом для покращення якості розв'язків на певних етапах еволюції, що дозволяє швидше досягати оптимуму [18].

Використання додаткових евристик дозволяє значно прискорити процес оптимізації, оскільки такі методи можуть ефективно скоротити простір пошуку, зосереджуючи увагу на більш перспективних ділянках. Це особливо корисно у випадках, коли простір рішень є великим і складним для аналізу звичайними методами. Завдяки евристичним правилам можна також підвищити якість кінцевих рішень, оскільки алгоритм отримує додаткову інформацію для ефективного знаходження глобальних оптимумів.

Проте є і певні недоліки, пов'язані з використанням додаткових евристик у генетичних алгоритмах. По-перше, застосування цих методів може суттєво ускладнити алгоритм, що вимагає додаткових обчислювальних ресурсів та часу на реалізацію. Крім того, не завжди легко підібрати відповідні евристичні правила, оскільки їх ефективність залежить від конкретної задачі. Неправильно обрані евристики можуть навіть погіршити результати, приводячи до передчасної збіжності або обмеження пошуку лише на певних ділянках простору рішень. Також існує ризик застрягання у локальних оптимумах, якщо евристика занадто звужує простір пошуку.

Таким чином, генетичні алгоритми з використанням додаткових евристик забезпечують значні переваги в швидкості та якості пошуку, особливо для складних задач, проте їх реалізація вимагає ретельного вибору евристичних правил і може збільшити складність та вимоги до ресурсів алгоритму.

2.2.5 Меметичні алгоритми

Меметичні алгоритми являють собою вдосконалений підхід, який об'єднує принципи генетичних алгоритмів із локальними пошуковими

методами, забезпечуючи оптимальний баланс між глобальним і локальним пошуком рішень. На відміну від класичних генетичних алгоритмів, що переважно зосереджуються на глобальній оптимізації через еволюційні механізми, меметичні алгоритми додають локальні пошукові стратегії, що дає можливість більш ретельно досліджувати окремі рішення для їх подальшого поліпшення. Це дозволяє їм ефективніше адаптуватися до складних задач, де необхідно не лише дослідити широкий простір рішень, але й тонко налаштовувати знайдені рішення на локальному рівні.

Основна перевага меметичних алгоритмів полягає в тому, що вони значно скорочують час на пошук оптимальних рішень у складних завданнях, оскільки глобальний пошук генетичного алгоритму доповнюється локальними вдосконаленнями, що забезпечує більш точну оптимізацію. Це особливо корисно в задачах з великим простором рішень або такими, що мають багато локальних оптимумів, які генетичні алгоритми можуть оминати. Меметичні алгоритми не тільки виявляють глобальні оптимуми, але й підвищують точність знайдених рішень, що робить їх потужним інструментом для вирішення складних оптимізаційних завдань [19].

Крім того, завдяки використанню додаткових евристик у локальному пошуку, меметичні алгоритми забезпечують гнучкість і ефективність при пошуку оптимальних рішень. Евристики дозволяють тонко налаштовувати процес пошуку, спрямовуючи його в найбільш перспективні області простору рішень. Це підвищує ефективність алгоритму та зменшує обчислювальні витрати на знаходження оптимального розв'язку.

Проте меметичні алгоритми також мають певні недоліки. Вони є більш складними у реалізації, оскільки потребують ретельного налаштування як генетичних операцій, так і локальних пошукових механізмів. Крім того, поєднання глобального та локального пошуку може збільшувати обчислювальну вартість алгоритму, особливо у випадках великих даних або ресурсномістких завдань. Також існує ризик надмірної уваги до локальних

рішень, що може призвести до недостатнього дослідження глобального простору і, як наслідок, до втрати деяких важливих варіантів.

Таким чином, меметичні алгоритми з використанням додаткових евристик є надзвичайно ефективним інструментом для вирішення складних оптимізаційних задач, поєднуючи переваги глобального пошуку з точними локальними вдосконаленнями. Вони забезпечують високу якість рішень і прискорюють процес оптимізації, проте потребують обережного налаштування і можуть бути обчислювально інтенсивними.

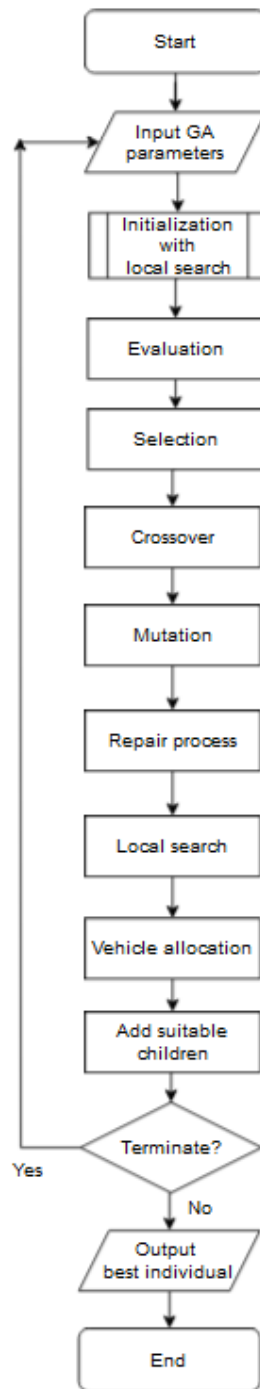


Рисунок 2.4 – Блок-схема функціонування меметичного алгоритму

2.3 Аргументація вибору визначеного виду генетичного алгоритму

Для аналізу різних типів генетичних алгоритмів були використані наступні критерії: складність впровадження, збіжність, масштабованість, можливість паралельної обробки, точність результатів та адаптивність.

- Складність впровадження: цей параметр показує, наскільки складно реалізувати алгоритм на практиці. Високий рівень складності вказує на

те, що для створення програмного коду необхідні значні зусилля та ресурси, тоді як низький рівень означає, що алгоритм можна реалізувати швидко і з мінімальними витратами.

- **Збіжність:** характеризує ефективність алгоритму у досягненні оптимального або прийнятного результату. Висока збіжність свідчить про здатність алгоритму швидко знаходити розв'язок, тоді як низька — про те, що алгоритму потрібно більше часу та ресурсів для отримання результату.
- **Масштабованість:** відображає здатність алгоритму працювати з великими обсягами даних або додатковими обчислювальними ресурсами. Висока масштабованість вказує на те, що алгоритм зберігає ефективність при збільшенні обсягу вхідних даних, тоді як низька масштабованість означає наявність обмежень щодо обробки великих масивів інформації.
- **Паралельна обробка:** показник, що визначає, наскільки добре алгоритм може розподіляти обчислювальні завдання між різними процесорами або обчислювальними вузлами. Висока здатність до паралельного обчислення дозволяє алгоритму швидко виконувати завдання на багатоядерних системах без значних витрат на комунікацію.
- **Точність результатів:** визначає, наскільки точними та коректними є результати, що отримуються від алгоритму. Висока точність свідчить про здатність алгоритму забезпечувати надійні результати, тоді як низька точність може вказувати на ризик отримання похибок чи неточностей.
- **Адаптивність:** цей критерій описує здатність алгоритму ефективно підлаштовуватися до змін параметрів чи умов завдання. Високий рівень адаптивності означає, що алгоритм легко адаптується до нових вимог, тоді як низька адаптивність свідчить про певні труднощі в пристосуванні до зміни умов.

За цими критеріями можна провести порівняння різних підвидів генетичних алгоритмів:

Таблиця 2.1 – Порівняння різних підвидів генетичних алгоритмів

Критерії/ Алгоритми	Складність реалізації	Збіжність	Масштабованість	Підтримка паралельного обчислення	Точність результатів	Адаптив ність
Генетичні алгоритми з множинним критерієм	Середня	Висока	Висока	Середня	Середня	Висока
Паралельні генетичні алгоритми	Висока	Висока	Висока	Висока	Середня	Середня
Адаптивні генетичні алгоритми	Середня	Середня	Висока	Низька	Середня	Висока
Генетичні алгоритми з використан ням додаткових евристик	Середня	Висока	Висока	Середня	Середня	Висока
Меметичні алгоритми	Середня	Висока	Висока	Висока	Висока	Висока

Алгоритми оцінювалися за шкалою з трьома рівнями: "Високий", "Середній" та "Низький" відповідно до кожного з визначених параметрів. Ця шкала використовується для визначення ступеня відповідності алгоритму кожному критерію. "Високий" рівень свідчить про те, що алгоритм має значні переваги у відповідному аспекті та працює ефективно. "Середній" вказує на задовільну відповідність критерію, тоді як "Низький" рівень демонструє обмежені можливості або низьку ефективність у даному аспекті, що може свідчити про труднощі у його застосуванні. Ця шкала допомагає краще оцінювати сильні та слабкі сторони кожного алгоритму та приймати рішення щодо їхньої придатності для конкретних завдань.

На основі таких оцінок, меметичний алгоритм вважається найефективнішим методом для моніторингу екологічних показників. Завдяки поєднанню глобального і локального пошуку, він здатний оперативно адаптуватися до змін навколишнього середовища, що є важливою властивістю

для оцінки якості повітря, де умови можуть швидко змінюватися. Також цей алгоритм дозволяє досягти високої точності прогнозування завдяки використанню методів локальної оптимізації, які підвищують збіжність та зменшують похибки оцінки.

2.4 Висновки до другого розділу

У цьому розділі було проведено детальний аналіз різних типів генетичних алгоритмів. На основі порівняльної оцінки за критеріями складності реалізації, збіжності, масштабованості, можливості паралельної обробки, точності результатів та адаптивності було визначено, що меметичний алгоритм є найбільш оптимальним для розв'язання завдань, де потрібна швидка адаптація до змінних умов і високий рівень точності розрахунків. Його ключова перевага полягає у поєднанні глобального та локального пошуку, що забезпечує не тільки ефективне знаходження оптимального рішення, але й його подальше уточнення.

Меметичний алгоритм було вибрано завдяки його здатності швидко адаптуватися до змін середовища, зокрема у випадках, коли стан параметрів задачі змінюється в реальному часі. Це робить його особливо ефективним для моніторингу динамічних систем, де необхідна швидка реакція на зміни та точне прогнозування. Крім того, цей метод демонструє високу збіжність та мінімізацію похибок завдяки використанню локальних методів оптимізації, що дозволяє йому перевершувати інші види генетичних алгоритмів у складних завданнях з великою кількістю параметрів.

3 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ПРОЄКТУВАННЯ

3.1 Вибір комплектуючих для отримання та обробки даних

У процесі розробки системи моніторингу якості повітря важливим етапом є вибір комплектуючих, які забезпечують точне отримання, обробку та передачу даних. До таких комплектуючих належать датчики для вимірювання параметрів навколишнього середовища, мікроконтролери для збору й аналізу даних. Обґрунтований вибір цих компонентів визначає ефективність, точність і надійність роботи системи, що особливо важливо для застосувань із використанням безпілотних літальних апаратів (БПЛА), де обмежені енергетичні та вагові ресурси.

3.1.1 Плата мікроконтролерів

Мікроконтролери є основними елементами багатьох сучасних електронних пристроїв, включаючи безпілотні літальні апарати (БПЛА). Вони представляють собою компактні обчислювальні системи, здатні виконувати різноманітні завдання, пов'язані з обробкою сигналів, управлінням сенсорами та іншими компонентами. У контексті БПЛА для оцінки якості повітря плати мікроконтролерів використовуються як основа для інтеграції сенсорних модулів, збору та обробки даних, а також забезпечення передачі результатів на наземні станції для подальшого аналізу [20].

Основні функції мікроконтролерів у БПЛА:

- Збір даних із сенсорів: Мікроконтролери забезпечують підключення сенсорів, таких як датчики вуглекислого газу, летких органічних сполук (VOC) та температури, та обробку їх сигналів у режимі реального часу.
- Обчислення та управління: Вони виконують основні розрахунки, необхідні для аналізу якості повітря, та управляють іншими

компонентами БПЛА, такими як мотори чи системи позиціонування [21].

- Передача даних: Через інтерфейси зв'язку (Wi-Fi, Bluetooth) мікроконтролери передають зібрані дані на віддалені сервери для більш глибокого аналізу або візуалізації.

Для реалізації подібних завдань найбільш поширеними є наступні плати: Arduino Uno, Raspberry Pi Pico, та ESP32 [22]. Нижче наведено їх порівняльний аналіз:

Таблиця 3.1 – Порівняння популярних плат мікроконтролерів у контексті наданої задачі.

Характеристика	Arduino Uno	RPi Pico	ESP32
Процесор	ATmega328P (8 бітний)	RP2040 (2 ядра, 32-бітний)	Tensilica Xtensa (32-бітний)
Тактова частота	16 МГц	133 МГц	До 240 МГц
ОЗП	2 КБ SRAM	264 КБ SRAM	520 КБ SRAM
Інтерфейси підключення	UART, SPI, I2C, Wi-Fi, Bluetooth	UART, SPI, I2C	UART, SPI, I2C, Wi-Fi, Bluetooth
Енергоспоживання	Низьке	Середнє	Високе
Ціна	Низька	Середня	Висока
Простота у використанні	Дуже висока	Висока	Середня

Критерії оцінювання визначаються за шкалою:

- Дуже високий: Значення характеристики значно перевищує базовий рівень, забезпечуючи високі показники продуктивності або ефективності.
- Високий: Значення характеристики перевищує базовий рівень, забезпечуючи високі показники продуктивності або ефективності.
- Середній: Характеристика знаходиться на прийнятному рівні, достатньому для виконання завдань, але не є оптимальною.
- Низький: Характеристика обмежена або недостатня для повноцінного

виконання завдань, що може спричинити труднощі у використанні.

Серед трьох розглянутих плат Arduino Uno є оптимальним вибором для реалізації задачі оцінки якості повітря БПЛА, що зумовлено наступними факторами [23]:

- Простота використання: Arduino Uno має інтуїтивно зрозумілий інтерфейс програмування та доступну документацію, що спрощує інтеграцію сенсорів і налаштування системи навіть для користувачів з базовими знаннями в електроніці.
- Енергоефективність: Низьке енергоспоживання робить плату ідеальною для використання в умовах обмеженого ресурсу батареї на БПЛА.
- Достатня кількість ресурсів: Хоча Raspberry Pi Pico та ESP32 мають більш потужні процесори, 8-бітного мікроконтролера ATmega328P Arduino Uno достатньо для обробки сигналів сенсорів у реальному часі та виконання базових обчислень.
- Ціна та доступність: Arduino Uno є економічно вигідним варіантом із широкою доступністю на ринку, що важливо для проєктів із обмеженим бюджетом.

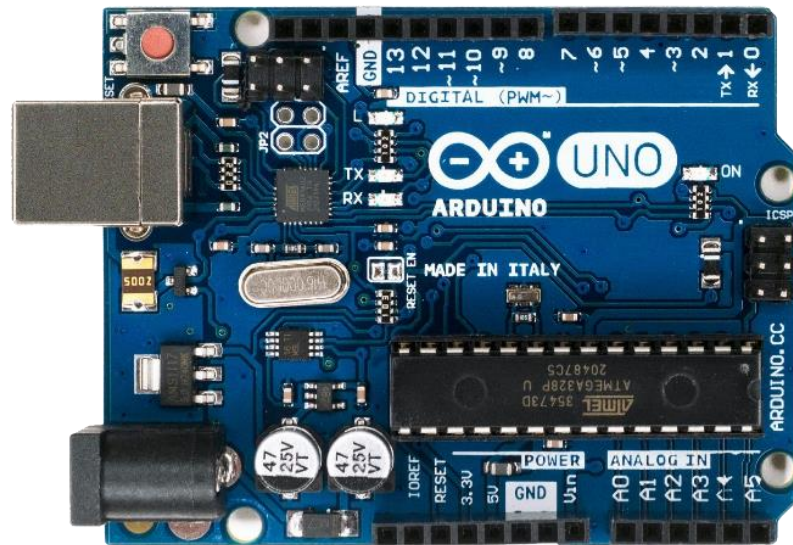


Рисунок 3.1 – Плата мікроконтролерів Arduino Uno

3.1.2 Датчик газу

Датчик газу є електронним пристроєм, призначеним для виявлення концентрації певних газів у повітрі. Основна функція таких датчиків полягає у перетворенні фізико-хімічних властивостей газу в електричний сигнал, що може бути проаналізований та використаний для визначення якості повітря.

MQ-2 є одним із найпоширеніших газових датчиків, який використовується для виявлення горючих газів та диму, таких як водень, пропан, метан, бутан, оксид вуглецю (CO) та інші. Основу його роботи складає напівпровідниковий чутливий елемент, який змінює свій опір залежно від концентрації газу в повітрі [24].



Рисунок 3.2 – Датчик газу MQ-2

У конструкції MQ-2 використовується матеріал на основі діоксиду олова (SnO_2). У чистому повітрі поверхня SnO_2 має високу опірність. При наявності горючих газів молекули цих газів вступають у хімічну реакцію з адсорбованим киснем на поверхні чутливого шару. Ця реакція знижує кількість адсорбованого кисню, що призводить до зменшення опору матеріалу. Зміна опору фіксується електричною схемою, яка перетворює цей параметр у напругу, пропорційну концентрації газу.

MQ-2 було обрано для використання в даному проєкті з таких причин:

- Широкий спектр виявлення газів. MQ-2 здатен виявляти різні горючі гази, що робить його універсальним рішенням для моніторингу якості повітря.
- Чутливість. Датчик має високу чутливість до низьких концентрацій газів, що дозволяє виявляти забруднення навіть у невеликих кількостях.
- Низька вартість. У порівнянні з іншими типами газових датчиків, MQ-2 є економічно вигідним варіантом, що є важливим для масштабованих рішень, таких як БПЛА для моніторингу повітря.
- Простота інтеграції. MQ-2 легко інтегрується у мікроконтролерні

системи, такі як Arduino або RPi, завдяки простій схемі підключення та можливості працювати з аналоговими і цифровими сигналами.

- Низьке енергоспоживання. Це критично важливо для мобільних платформ, таких як безпілотні літальні апарати (БПЛА), де економія енергії має ключове значення.

3.1.3 Датчик температури

Датчик температури – це пристрій, який перетворює температуру навколишнього середовища в електричний сигнал. Ці сигнали можуть бути використані для вимірювання, моніторингу та контролю температури в різних умовах. TMP36 є одним із таких датчиків, який забезпечує точне і просте вимірювання температури [25].



Рисунок 3.3 – Датчик температури TMP36

TMP36 – це аналоговий датчик температури з низьким

енергоспоживанням, який генерує напругу, пропорційну температурі в градусах Цельсія. TMP36 базується на напівпровідниковій технології і є повністю каліброваним для прямого використання.

Особливості роботи TMP36:

- Пропорційний сигнал. Датчик забезпечує вихідну напругу, яка лінійно залежить від температури.
- Температурний діапазон. TMP36 здатний вимірювати температури від $-40\text{ }^{\circ}\text{C}$ до $+125\text{ }^{\circ}\text{C}$, що робить його придатним для різних середовищ.
- Калібрування. Датчик попередньо калібрований, що виключає необхідність додаткових налаштувань для більшості застосувань.

TMP36 був обраний для використання у проєкті моніторингу якості повітря з таких причин:

- Висока точність. TMP36 забезпечує високу точність вимірювань.
- Простота інтеграції. Датчик працює без необхідності додаткових зовнішніх компонентів і може бути легко підключений до мікроконтролерів, таких як Arduino.
- Низьке енергоспоживання. TMP36 споживає менше 50 мкА струму, що дозволяє використовувати його на пристроях з обмеженими енергоресурсами, таких як безпілотні літальні апарати (БПЛА).
- Широкий діапазон напруги живлення. TMP36 працює при напрузі від 2.7 В до 5.5 В , що дає змогу застосовувати його в різних електронних системах.
- Стабільність роботи. TMP36 характеризується низькою чутливістю до шуму та стабільною роботою навіть у несприятливих умовах.

Таким чином, TMP36 забезпечує надійне та енергоефективне вимірювання температури, що є важливим компонентом для екосистем моніторингу якості повітря за допомогою БПЛА.

3.1.4 Датчик вогню

KY-026 – це електронний модуль, призначений для виявлення полум'я або джерел інфрачервоного випромінювання, що виникає під час горіння. Цей датчик використовується для моніторингу навколишнього середовища, зокрема в системах раннього виявлення пожеж або аналізу якості повітря, пов'язаної із забрудненням продуктами горіння. KY-026 складається з інфрачервоного фотодіода та підсилювача сигналу, який перетворює зміну інфрачервоного випромінювання в електричний сигнал [26].

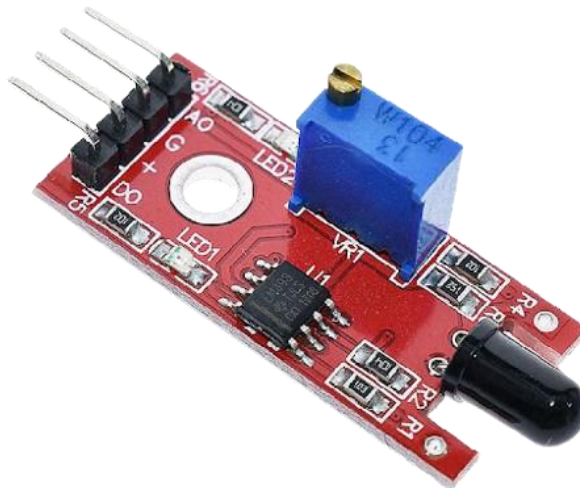


Рисунок 3.4 – Датчик вогню та інфрачервоного випромінювання KY-026

Основні елементи роботи:

- Виявлення інфрачервоного випромінювання. Полум'я випромінює світло в інфрачервоному діапазоні (~760–1100 нм), яке фіксується фотодіодом.
- Підсилення сигналу. Вловлене випромінювання обробляється внутрішньою схемою, яка посилює сигнал для подальшого використання в системі.
- Аналоговий і цифровий вихід. KY-026 підтримує два режими роботи:

- Аналоговий вихід (A0): Відображає рівень інфрачервоного випромінювання.
- Цифровий вихід (D0): Подає сигнал "виявлено" при перевищенні заданого порогу.

КУ-026 був обраний для використання в проєкті аналізу якості повітря через наступні переваги:

- Сенситивність до полум'я. Датчик ефективно виявляє навіть слабкі джерела полум'я, що дозволяє оцінювати наявність продуктів горіння в реальному часі.
- Дворежимний вихід. Наявність аналогового і цифрового виходу спрощує інтеграцію з різними мікроконтролерами та дозволяє більш гнучко налаштовувати систему.
- Компактність. КУ-026 має невеликий розмір і вагу, що робить його придатним для встановлення на безпілотних літальних апаратах (БПЛА).
- Низька вартість. У порівнянні з іншими пристроями виявлення полум'я, КУ-026 є економічно вигідним варіантом.
- Енергоефективність. Завдяки низькому енергоспоживанню, датчик може працювати тривалий час навіть у системах з обмеженим живленням.

У контексті аналізу якості повітря, КУ-026 використовується для виявлення джерел горіння. Завдяки його здатності працювати на мобільних платформах, таких як БПЛА, датчик може бути застосований для моніторингу пожежонебезпечних зон або промислових районів [27].

Таким чином, КУ-026 забезпечує ефективне, компактне та доступне рішення для аналізу продуктів горіння у рамках завдань моніторингу якості повітря.

3.1.5 Датчик температури і вологості

DHT11 – це цифровий датчик, який дозволяє одночасно вимірювати температуру і відносну вологість повітря. Завдяки поєднанню двох функцій у одному пристрої, він широко використовується для моніторингу кліматичних умов у різних сферах, зокрема у проектах, пов'язаних із контролем якості повітря. DHT11 складається з двох основних компонентів: термістора для вимірювання температури та ємнісного сенсора для визначення вологості. Дані вимірювань передаються через цифровий інтерфейс, що значно спрощує інтеграцію датчика у системи моніторингу [28].

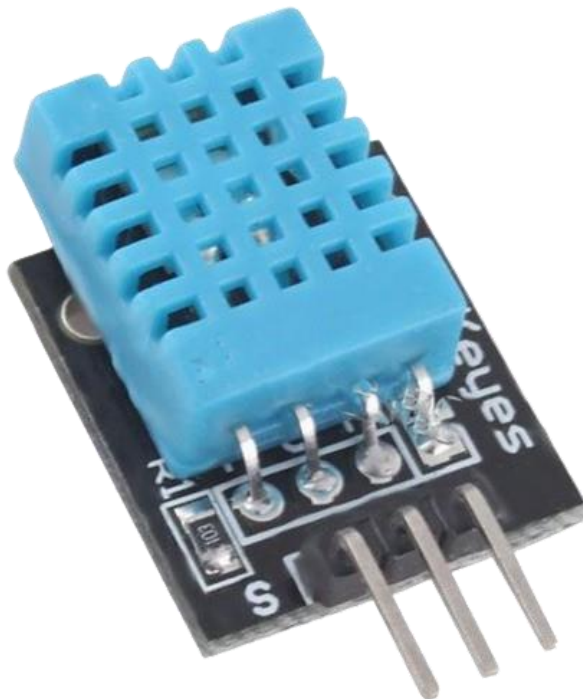


Рисунок 3.5 – Датчик температури і вологості DHT11

Особливості роботи:

- Температура. DHT11 вимірює температуру в діапазоні від 0 °C до +50 °C із точністю ± 2 °C. Термістор змінює свій опір залежно від температури, а вбудований мікроконтролер перетворює ці зміни на цифровий сигнал.

- Відносна вологість. Ємнісний сенсор вимірює вологість у діапазоні 20–90% із точністю $\pm 5\%$. Сенсор визначає зміни ємності в залежності від кількості водяної пари у повітрі.
- Цифровий вихід. DHT11 передає дані у вигляді цифрового сигналу через однопровідний інтерфейс, що зменшує кількість необхідних підключень [29].

DHT11 був обраний для використання у проєкті з моніторингу якості повітря з таких причин:

- Поєднання двох функцій. Вимірювання температури і вологості за допомогою одного датчика дозволяє оптимізувати апаратну частину системи.
- Цифровий інтерфейс. Простота підключення до мікроконтролерів, таких як Arduino або ESP, завдяки цифровому виходу.
- Доступність. DHT11 має низьку вартість у порівнянні з іншими подібними датчиками, що важливо для масштабованих проєктів.
- Низьке енергоспоживання. Датчик споживає менше 2 мА під час вимірювання, що дозволяє використовувати його на пристроях із обмеженим енергоресурсом, таких як безпілотні літальні апарати (БПЛА).
- Стабільність у стандартних умовах. DHT11 демонструє надійність у діапазоні температур і вологості, що характерні для моніторингу кліматичних умов у міських або природних зонах.

Хоча DHT11 є простим і зручним у використанні, варто врахувати його обмеження: вузький діапазон вимірювань і відносно низька точність у порівнянні з більш сучасними аналогами. Однак у комбінації з датчиком TMP36, для екологічних досліджень ці характеристики є цілком прийнятними.

Таким чином, DHT11 є доступним, компактним і енергоефективним рішенням для моніторингу кліматичних параметрів, що є важливим компонентом у системах аналізу якості повітря.

3.2 З'єднання з сенсорами та передача даних

Одним із ключових етапів створення системи моніторингу є організація з'єднання датчиків із мікроконтролером та налаштування передачі даних для їх подальшого аналізу. Для цього необхідно забезпечити коректну інтеграцію всіх компонентів, використовуючи відповідні інтерфейси, такі як аналогові, цифрові чи послідовні порти, залежно від типу сенсора. Ефективна схема підключення дозволяє зчитувати дані з датчиків у реальному часі та передавати їх для обробки і збереження, що є основою для точного і надійного функціонування системи.

У даному проєкті всі сенсори підключатимуться до мікроконтролера Arduino Uno, який слугуватиме центральним вузлом для збору даних. Для перевірки коректної роботи датчиків і налагодження системи буде використовуватися середовище розробки Arduino IDE, що дозволяє здійснювати тестування, зчитування та візуалізацію даних у зручному форматі. Такий підхід забезпечує простоту інтеграції комплектуючих і надійність їх функціонування у складі системи.

3.2.1 MQ-2

Датчик MQ-2 є аналоговим сенсором, що дозволяє виявляти гази, такі як пропан, метан, бутан та дим. Для його інтеграції з мікроконтролером Arduino Uno необхідно виконати підключення через аналоговий вихід.

Етапи підключення MQ-2 до Arduino Uno:

- Електроживлення датчика: підключено контакт VCC датчика до 5V на платі Arduino Uno; з'єднано контакт GND датчика з GND на Arduino.
- Зчитування сигналу: Для використання аналогового виходу MQ-2 підключено контакт AOUT до одного з аналогових входів Arduino – A0. Це дозволяє отримати сигнал у вигляді напруги, пропорційної концентрації газу.

- Підключення піна для візуалізації сигналу (додатково): у цифровий сокет під'єднано лампочку червоного кольору, для сигналізування інтенсивності газу.

3.2.2 TMP36

Датчик TMP36 є цифровим температурним сенсором, що забезпечує точні вимірювання температури. Він має вбудований аналого-цифровий перетворювач (ADC) і працює через інтерфейс I2C. Для інтеграції TMP36 з мікроконтролером Arduino Uno потрібно правильно підключити його до відповідних пінів і налаштувати зчитування температури через Arduino IDE.

Етапи підключення TMP36 до Arduino Uno:

- Електроживлення датчика: Підключено контакт VCC TMP36 до 5V на платі Arduino Uno; з'єднано контакт GND датчика з GND на Arduino.
- Підключення до I2C шини: Підключено контакт SDA TMP36 до SDA на Arduino Uno пін A4; підключено контакт SCL TMP36 до SCL на Arduino Uno пін A5.
- Підключення до мікроконтролера: Для TMP36 потрібно використовувати бібліотеку для I2C зв'язку.
- Додатково: LED пін, який міняє свій колір залежно від температури – синій – нижче 20 градусів за Цельсієм, зелений – між 20 та 25 градусів за Цельсієм, жовтий – між 25 та 30 градусів за Цельсієм, червоний – вище 30 градусів за Цельсієм.

3.2.3 KY-026

Датчик KY-026 є сенсором для вимірювання відкритого полум'я та інфрачервоного випромінювання. Він має аналоговий вихід для точного

вимірювання інфрачервоного випромінювання в навколишньому середовищі та цифровий вихід для виявлення перевищення заданого порогу концентрації. Для інтеграції KY-026 з мікроконтролером Arduino Uno необхідно правильно підключити його до пінів і налаштувати зчитування даних через Arduino IDE.

Етапи підключення KY-026 до Arduino Uno:

- Електроживлення датчика: Підключено контакт VCC KY-026 до 5V на платі Arduino Uno; з'єднано контакт GND датчика з GND на Arduino.
- Підключення виходів датчика: Для використання цифрового виходу підключено контакт DOUT до цифрового піну Arduino D2. Цифровий вихід активується, коли концентрація перевищує заданий поріг, який можна налаштувати за допомогою потенціометра на самій платі датчика.
- Додатково: пін червоного кольору, який сигналізуватиме про появу вогню.

3.2.4 DHT11

Датчик DHT11 використовується для вимірювання температури та вологості навколишнього середовища. Це цифровий сенсор, який передає дані через один пін, використовуючи спеціальний протокол. Для підключення датчика DHT11 до мікроконтролера Arduino Uno необхідно з'єднати його з відповідними пінами на платі та налаштувати зчитування даних через Arduino IDE.

Етапи підключення DHT11 до Arduino Uno:

- Електроживлення датчика: контакт VCC датчика підключено до 5V на платі Arduino Uno; контакт GND датчика з'єднано з GND на Arduino.
- Підключення виходу датчика: контакт DATA датчика підключено до цифрового піну на Arduino D2. Це дозволяє передавати цифрові сигнали від датчика до мікроконтролера.
- Підключення резистора: для коректної роботи датчика необхідно

підключити резистор на 10 кОм між контактами VCC і DATA, що забезпечить стабільність роботи датчика.

- Додатково: 2 піни червоного (для значення температури) та синього (для значення вологості) кольору, які міняють інтенсивність кольору залежно від зміни величини параметрів.

3.3 Вибір засобів програмування

Для реалізації системи збору, обробки та передачі даних із сенсорів у цьому проєкті необхідно вибрати оптимальні засоби програмування, які забезпечать ефективність, простоту реалізації та високу продуктивність. Основними критеріями вибору є сумісність із мікроконтролером Arduino Uno, можливість роботи з обмеженими ресурсами, підтримка сенсорів і бібліотек, а також зручність розробки.

У даному контексті для розгляду були обрані мови C, Python, C++ та JavaScript, оскільки вони покривають деякі з ключових аспекти розробки та роботи з системами збору й обробки даних. Мова C є основним вибором для програмування мікроконтролерів завдяки її високій ефективності роботи з апаратними ресурсами та сумісності з платформою Arduino Uno. Python, хоч і не підходить для прямого використання на мікроконтролері, є зручним інструментом для попереднього тестування алгоритмів та аналізу даних.

Мова C++ дозволяє поєднати ефективність C із зручностями об'єктно-орієнтованого програмування, що може бути корисним у великих або масштабованих системах. JavaScript із Node.js підходить для створення серверної частини та обробки даних у хмарі чи на стороні клієнта, що робить його корисним для інтеграції з веб-сервісами. Таким чином, кожна з цих мов має свої переваги та можливості для вирішення різних завдань у межах цього проєкту.

Детальний порівняльний аналіз засобів програмування зображено на таблиці 3.2.

Таблиця 3.2 – Порівняльний аналіз мов програмування у контексті розробки інформаційної технології оптимізації планування траєкторії групи БПЛА для оцінки забруднення повітря

Мова	Ефективність роботи з ресурсами	Простота використання	Сумісність з Arduino	Підтримка сенсорів	Основна сфера застосування	Недоліки
C	Висока	Середня	Повна	Широка	Програмування мікроконтролерів	Відсутність об'єктно-орієнтованого підходу, що може ускладнити масштабування
Python	Низька	Висока	Непряма	Широка	Тестування алгоритмів, серверна розробка	Великий обсяг ресурсів для виконання, непридатність для мікроконтролерів
C++	Висока	Низька	Повна	Широка	Об'єктно-орієнтовані системи	Збільшений розмір коду, складність і затримки через об'єктно-орієнтованість
JavaScript	Низька	Висока	Непряма	Обмежена	Веб-розробка, IoT	Низька продуктивність, непридатність для програмування низькорівневих пристроїв

Критерії оцінювання визначаються за шкалою:

- Високий: Значення характеристики перевищує базовий рівень, забезпечуючи високі показники продуктивності або ефективності.
- Середній: Характеристика знаходиться на прийнятному рівні, достатньому для виконання завдань, але не є оптимальною.
- Низький: Характеристика обмежена або недостатня для повноцінного виконання завдань, що може спричинити труднощі у використанні.

3.4 Реалізація алгоритму

У даному розділі розглядається практична реалізація алгоритму оптимізації планування траєкторії для групи безпілотних літальних апаратів, спрямованого на ефективну оцінку забруднення повітря. Основна мета цього етапу — інтеграція теоретично розроблених методів у програмне середовище з урахуванням специфічних особливостей експлуатації БПЛА. Процес реалізації включає два основні аспекти: адаптацію параметрів алгоритму до конкретних умов роботи групи БПЛА та написання програмного коду, який забезпечує виконання задач оптимізації в режимі реального часу.

3.4.1 Адаптація параметрів для групи БПЛА

Перенесення алгоритмів із наземних роботів на безпілотні літальні апарати (БПЛА) вимагає адаптації до тривимірного простору і специфічних особливостей польоту. У той час як наземні роботи функціонують у двовимірному середовищі з фіксованим контактом із поверхнею, БПЛА працюють у складнішій тривимірній системі з динамічними умовами. Основні етапи такого перенесення включають:

1. Адаптація координатної системи:

- Включення Z-координати розширює маневреність, дозволяючи апарату підніматися, опускатися, уникати повітряних перешкод і адаптуватися до різних умов.
- Планування маршруту в тривимірному просторі передбачає врахування об'єктів на висоті, таких як будівлі, дерева або інші апарати.
- Навігація у тривимірному просторі ускладнює обчислення, оскільки алгоритм аналізує не лише відстані до перешкод, але й їхню тривимірну форму.

2. Розширення цільової функції:

- Оптимізація швидкості та висоти з урахуванням енергоефективності та мінімізації витрат енергії на висотні зміни.
- Забезпечення тривимірної безпеки через врахування допустимої відстані до інших об'єктів у повітрі та уникнення колізій.

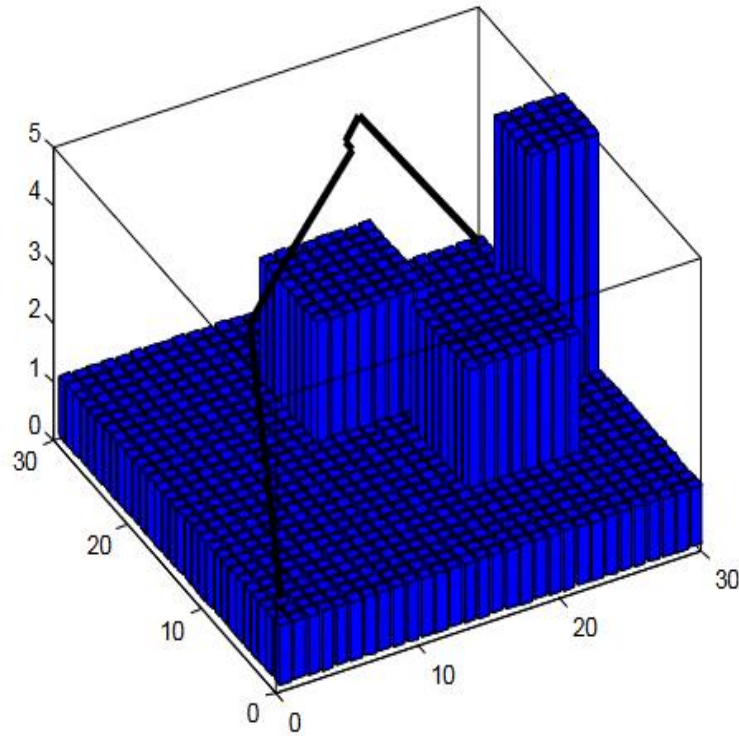


Рисунок 3.6 – Візуалізація пошуку найкращого шляху у тривимірному просторі

- Врахування щільності об'єктів у повітрі, особливо у міських районах або при роботі в складних середовищах із високою кількістю інших БПЛА.
3. Оптимізація маршрутів у груповій роботі БПЛА:
- Матричний підрахунок колізій (dc): У груповому сценарії необхідно враховувати ймовірність колізій між БПЛА. На першому етапі для кожного БПЛА встановлюються початкові параметри його позиції p_i , швидкості v_i , та траєкторії t_i . Для всіх БПЛА ці параметри формують множину вхідних даних, яка визначає їх просторове положення та маршрут. Далі визначаються зони потенційного зіткнення, що оточують кожен

БПЛА, які представляються у вигляді сферичної області радіуса r_{safe} . Радіус безпеки r_{safe} визначається на основі динамічних характеристик БПЛА, таких як швидкість, можливість маневрування, та мінімальна допустима відстань між апаратами. Для кожної пари БПЛА формується матриця колізій, де кожен елемент матриці представляє оцінку ймовірності зіткнення між i -м та j -м БПЛА. Визначення цих значень ґрунтується на декількох параметрах: відстань між поточними позиціями БПЛА, зміна відстані між траєкторіями. Якщо відстань між позиціями є меншою ніж мінімальна допустима, то значення матриці колізій збільшується. Матричний підрахунок колізій інтегрується в генетичний алгоритм як елемент цільової функції, що відповідає за безпеку польотів. Зокрема, значення колізій впливають на адаптацію ваги безпеки у цільовій функції генетичного алгоритму, що забезпечує мінімізацію ймовірності колізій у кожній ітерації оптимізації.

- Комбінація індивідуальних та колективних витрат: загальна цільова функція повинна включати індивідуальні витрати, такі як довжина шляху та безпека, разом з витратами на уникнення колізій між БПЛА. Формалізація такої функції може бути виражена як:

$$F_{total} = \alpha * F_{path} + \beta * F_{safety} + \gamma * F_{collision} \quad (3.1)$$

де $F_{collision}$ містить компонент на основі матриці колізії, що враховує всі можливі колізії між БПЛА. Якщо для певних пар БПЛА отримані високі значення ймовірності зіткнення, то генетичний алгоритм перезапускає еволюційний процес з оновленими параметрами траєкторій та зон небезпеки, що забезпечує уникнення зіткнень на наступних етапах оптимізації.

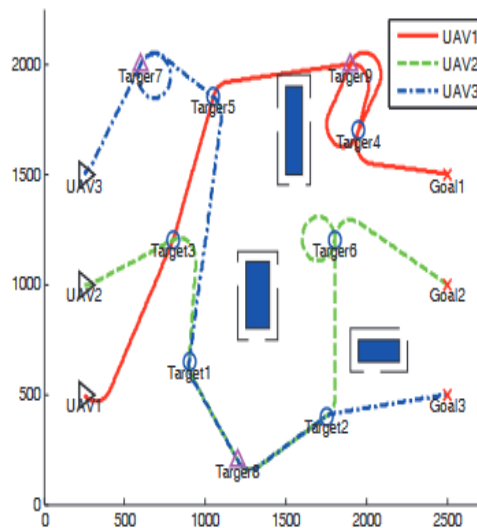


Рисунок 3.7 – Візуалізація роботи групи БПЛА у середовищі з перешкодами

3.4.2 Реалізація програмного коду

Для симуляції польоту БПЛА було використано середовище Webots, яке є професійною платформою для моделювання робототехнічних систем. Webots надає інструменти для створення віртуальних моделей роботів, симуляції їхньої поведінки та взаємодії з навколишнім середовищем. У цьому коді використовується API Webots, яке дозволяє програмувати сценарії поведінки на мові C. Основна ідея реалізації — адаптація генетичного алгоритму для оптимізації траєкторії БПЛА, враховуючи динамічне середовище.

Для забезпечення структурованості, спрощення розуміння, тестування та масштабування алгоритму, код розбито на наступні модулі:

Модуль 1: Ініціалізація середовища (цей модуль відповідає за підготовку середовища симуляції):

- Функція `wb_robot_init()`: Ініціалізує Webots та забезпечує доступ до всіх пристроїв моделі робота.
- Отримання пристроїв: `wb_robot_get_device("emitter")` — пристрій передачі даних для взаємодії з іншими вузлами;

`wb_robot_get_device("display")` — дисплей для візуалізації результатів.

- Налаштування базових параметрів: `time_step` — тривалість одного кроку симуляції; `display_width`, `display_height` — параметри дисплея для графіків.

Модуль 2: Реалізація генетичного алгоритму (генетичний алгоритм, що використовується для оптимізації траєкторій):

- Створення популяції: `population_create()` генерує початкову популяцію розміром `POPULATION_SIZE`.
- Еволюція: для кожного покоління оцінюються всі генотипи (`evaluate_genotype()`).
- Найкращий генотип визначається за функцією пристосованості (`measure_fitness()`).
- Розмноження відбувається через `population_reproduce()`.
- Оцінка пристосованості: використовується функція `measure_fitness()`, яка обчислює відстань, на яку був переміщений об'єкт у просторі, що імітує ефективність виконання завдання.

Модуль 3: Оцінка та візуалізація результатів

- Результати генетичного алгоритму виводяться у вигляді графіків: функція `plot_fitness()` — малює графіки найкращих (`best_fitness`) та середніх (`average_fitness`) значень пристосованості для кожного покоління.
- Графіки будуються на дисплеї, доступному через Webots API.

Модуль 4: Реалізація симуляції — функції симуляції забезпечують безперервне виконання алгоритму:

- Функція `run_seconds()` виконує симуляцію протягом заданого часу.
- Функція `run_optimization()`: організовує повний процес еволюції популяції; зберігає найкращий генотип до файлу `fittest.txt`.
- Функція `run_demo()`: демонструє роботу найкращого генотипу після завершення еволюції.

Модуль 5: Взаємодія із середовищем. Код використовує функції

супервізора для роботи з об'єктами в середовищі:

- `wb_supervisor_field_get_*` — доступ до полів позиції та орієнтації робота і об'єктів.
- `wb_supervisor_field_set_*` — оновлення стану об'єктів у симуляції.
- `wb_emitter_send()` — передача параметрів генотипу на пристрій БПЛА для тестування.

Для структурованого розуміння роботи системи було створено UML діаграми:

1. UML діаграма класів програми: діаграма класів демонструє структуру програми, включаючи класи, їх атрибути, методи та зв'язки між ними. Вона використовується для моделювання об'єктно-орієнтованої архітектури програми, забезпечуючи зрозуміле представлення логічної структури системи. Діаграму зображено у додатку Б (рис. Б.1).
2. UML діаграма активностей системи: діаграма активностей ілюструє потік виконання процесів у системі, зокрема їхню послідовність, розгалуження та синхронізацію. Вона дозволяє моделювати алгоритми, бізнес-процеси або поведінку компонентів системи. Діаграму зображено у додатку Б (рис. Б.2)
3. UML діаграма послідовності: ця діаграма описує порядок обміну повідомленнями між об'єктами системи в часі. Вона використовується для деталізації сценаріїв взаємодії користувача або підсистем, забезпечуючи візуалізацію їхньої взаємодії. Діаграму зображено у додатку Б (рис. Б.3)
4. UML діаграма розгортання системи: діаграма розгортання показує фізичну архітектуру системи, включаючи сервери, мережеві компоненти та їх взаємозв'язки. Вона використовується для проектування інфраструктури, на якій буде працювати система, та оцінки її ефективності. Діаграму зображено у додатку Б (рис. Б.4)
5. UML діаграма станів: діаграма станів моделює різні стани об'єкта та переходи між ними залежно від подій або умов. Вона дозволяє

проаналізувати логіку змін стану системи або її окремих компонентів.

Діаграму зображено у додатку Б (рис. Б.5)

6. UML діаграма об'єктів: діаграма об'єктів є конкретною реалізацією діаграми класів, що демонструє об'єкти системи та їхні взаємозв'язки на момент часу. Вона корисна для тестування або ілюстрації конкретних сценаріїв роботи системи. Діаграму зображено у додатку Б (рис. Б.6)
7. UML діаграма колаборації: ця діаграма відображає взаємодію між об'єктами для досягнення спільної мети, фокусуючись на структурних аспектах комунікації. Вона використовується для виявлення зв'язків між об'єктами та оцінки ефективності їхньої співпраці. Діаграму зображено у додатку Б (рис. Б.7)

Повний код програми наведений у додатку В.

3.5 Висновки до третього розділу

У даному розділі було проведено огляд сенсорів і вибрано оптимальні комплектуючі для задачі моніторингу навколишнього середовища. Зокрема:

- Плата мікроконтролерів: з урахуванням вимог до системи збору та аналізу даних про якість повітря на БПЛА, Arduino Uno є найбільш доцільним вибором серед представлених плат мікроконтролерів. Її характеристики забезпечують надійність, економічність та ефективність реалізації систем.
- Датчик газу MQ-2: вибрано для вимірювання концентрації газів у повітрі завдяки його чутливості до ряду шкідливих речовин.
- Датчик температури TMP36: обрано через простоту використання, високу точність і сумісність з Arduino.
- Датчик вогню KY-026: обрано для виявлення джерел вогню та забезпечення безпеки під час роботи системи.
- Датчик температури і вологості DHT11: обрано як недорогий і

ефективний засіб для одночасного вимірювання температури та вологості.

Було детально описано з'єднання обраних сенсорів із мікроконтролером Arduino. Враховано специфіку кожного датчика та взаємодію із мікроконтролером. Обґрунтовано вибір мови програмування C для реалізації програмного забезпечення. Мова C забезпечує ефективну роботу з апаратною частиною, низький рівень споживання ресурсів і зручність роботи з платформою Arduino. Виконано адаптацію алгоритмів під задачі збору та обробки даних із сенсорів. створено програмне забезпечення для збору, обробки та передачі даних із сенсорів на основі мови C. Таким чином, у цьому розділі було розроблено та обґрунтовано як апаратну, так і програмну частину системи моніторингу, що забезпечує ефективний збір, обробку та передачу даних для оцінки стану навколишнього середовища.

4 ТЕСТУВАННЯ РОБОТИ СИСТЕМИ

4.1 Тестування роботи сенсорів

Для забезпечення коректної роботи системи, що використовує сенсори для отримання даних про навколишнє середовище, необхідно провести тестування кожного компоненту. Це дозволяє виявити потенційні несправності, перевірити точність зчитуваних значень та підтвердити правильність підключення сенсорів до мікроконтролера. У цьому розділі наведено процес тестування сенсорів, що використовуються в проєкті, зокрема перевірка їх функціональності за допомогою Arduino Uno і Arduino IDE. Отримані результати тестів слугуватимуть основою для подальшої інтеграції сенсорів у систему.

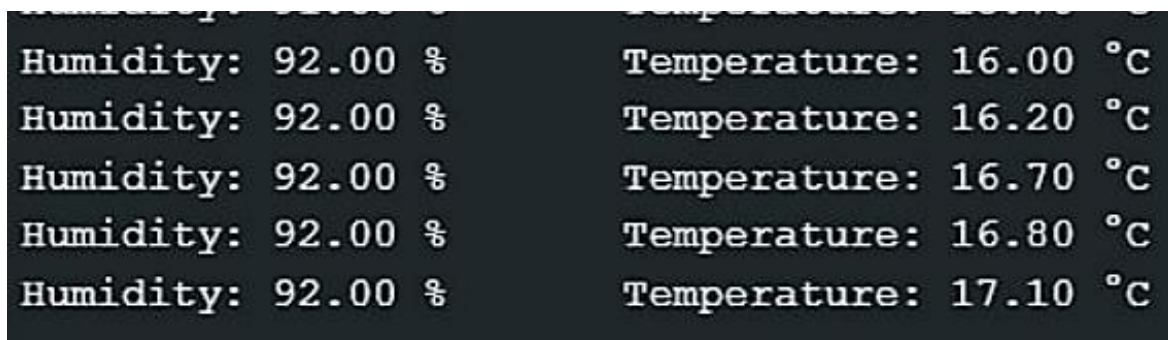
4.1.1 Тестування DHT11

Для перевірки роботи використовувалось переміщення датчика у різні температурні умови та умови, що відрізняються за ступенем вологості.



Humidity: 49.00 %	Temperature: 26.90 °C
-------------------	-----------------------

Рисунок 4.1 – Приклад значень датчика DHT11 у теплих сухих умовах



Humidity: 92.00 %	Temperature: 16.00 °C
Humidity: 92.00 %	Temperature: 16.20 °C
Humidity: 92.00 %	Temperature: 16.70 °C
Humidity: 92.00 %	Temperature: 16.80 °C
Humidity: 92.00 %	Temperature: 17.10 °C

Рисунок 4.2 – Приклад значень датчика DHT11 у прохолодних вологих

умовах



Рисунок 4.3 – Вигляд датчика DHT11 у теплих вологих умовах

4.1.2 Тестування KY-026

Для перевірки роботи було використано сірники. Датчик виводить значення, інтерпретацію яких можна спостерігати у середовищі Arduino. Коли значення перетинають зазначений поріг, виводиться фраза “Flame detected!”, у звичайному ж стані – “No flame detected”.

```
Flame detected!  
Flame detected!  
Flame detected!  
No flame detected  
No flame detected
```

Рисунок 4.4 – Вивід інтерпретованих даних у середовищі Arduino IDE

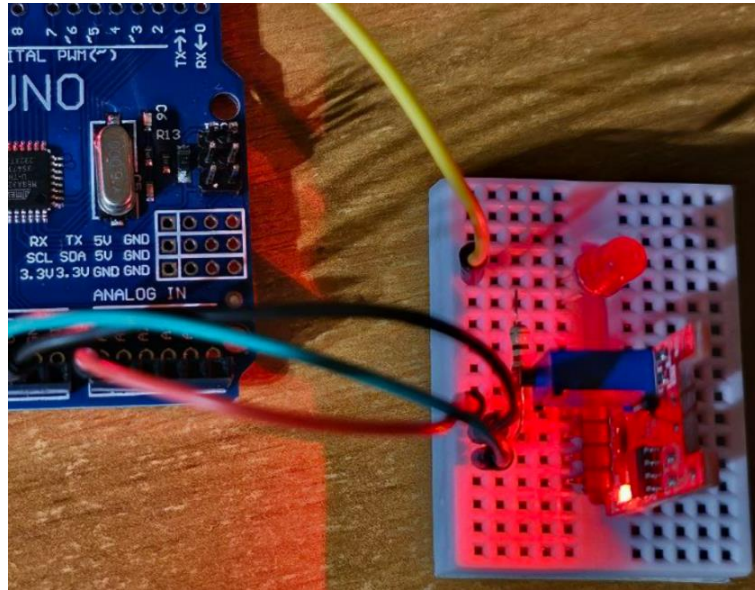


Рисунок 4.5 – Підключений датчик KY-026 без взаємодії з вогнем

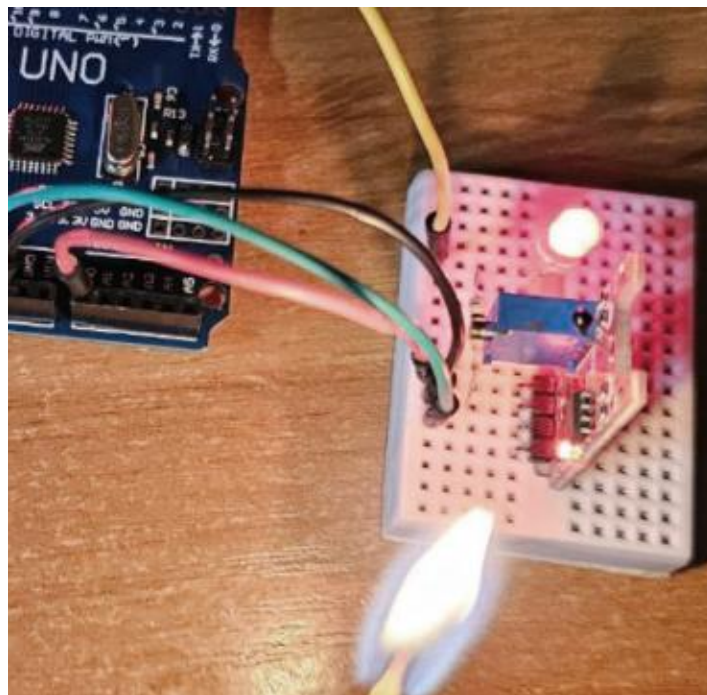


Рисунок 4.6 – Реакція датчика KY-026 на вогонь

4.1.3 Тестування TMP36

Для тестування роботи датчику було використано сірник, та переміщення у різні температурні середовища. Датчик виводить значення, що

можна спостерігати у середовищі Arduino.

```
Temperature: 26.90 °C  
Temperature: 27.20 °C  
Temperature: 27.30 °C  
Temperature: 27.90 °C  
Temperature: 27.80 °C
```

Рисунок 4.7 – Приклад значень датчика TMP36 у Arduino IDE

Принцип роботи датчика візуалізовано на наступних рисунках:

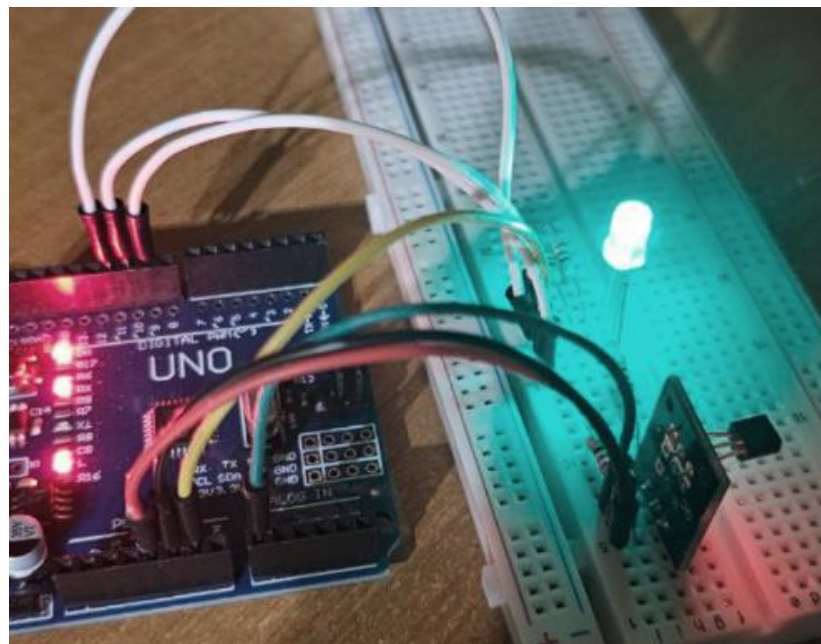


Рисунок 4.8 – Реакція датчика TMP36 на кімнатну температуру

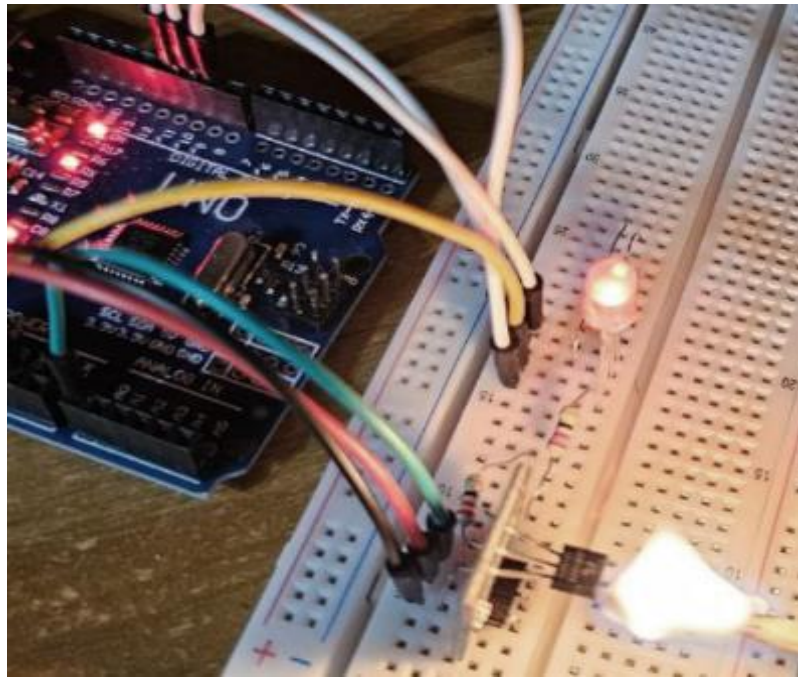


Рисунок 4.9 – Реакція датчика TMR36 на нагрівання

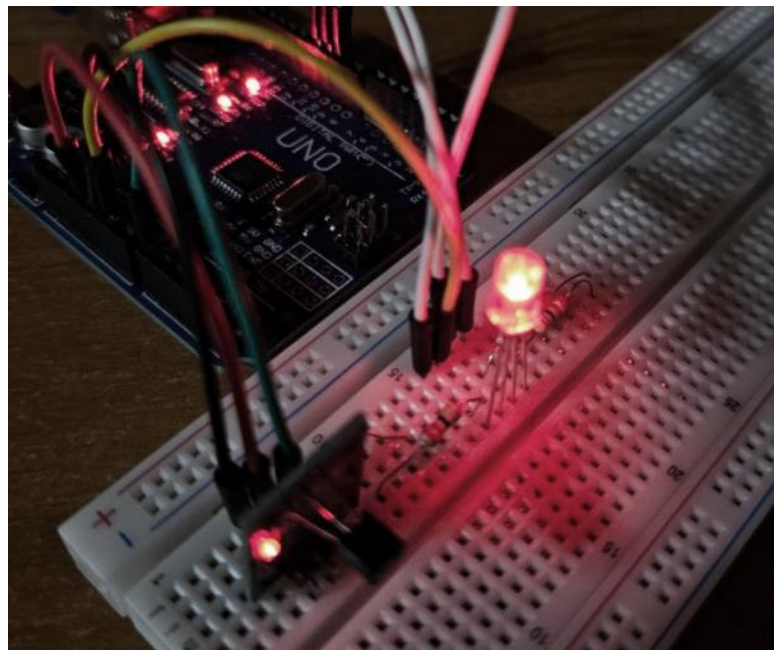


Рисунок 4.10 – Реакція датчика TMR36 на подальше нагрівання



Рисунок 4.11 – Реакція датчика TMR36 на температуру нижче 20 градусів за Цельсієм

4.1.4 Тестування MQ2

Для тестування роботи датчику було використано газову запальничку. Датчик виводить значення, що можна спостерігати у середовищі Arduino.

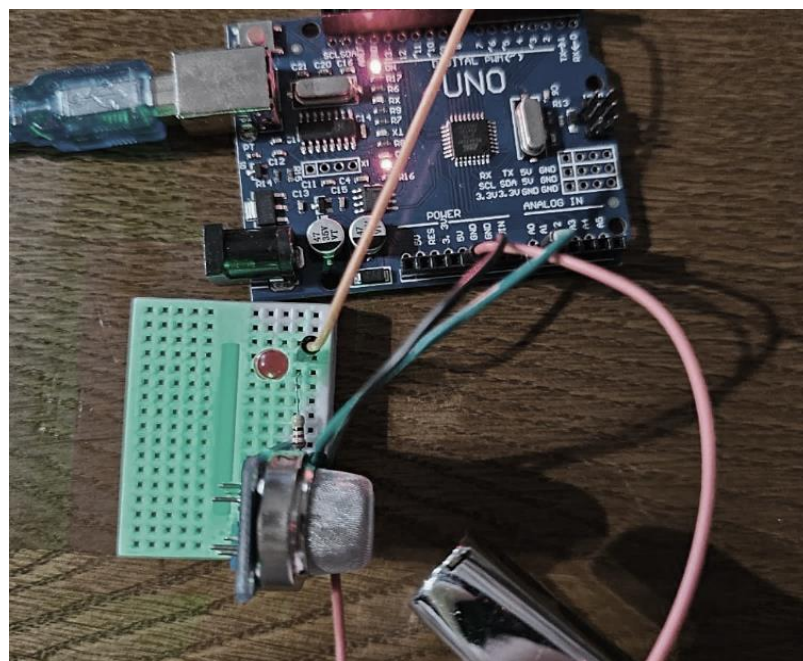


Рисунок 4.12 – Запуск газу на датчик за допомогою газової запальнички

```

Analog output: 47
Analog output: 48
Analog output: 48
Analog output: 50
Analog output: 47

```

Рисунок 4.13 – Значення датчика MQ-2 у Arduino IDE

Коли значення стає вищим за 48, датчик сигналізує про появу газу за допомогою інтенсивності світла вбудованої лампочки та під'єданого піна.

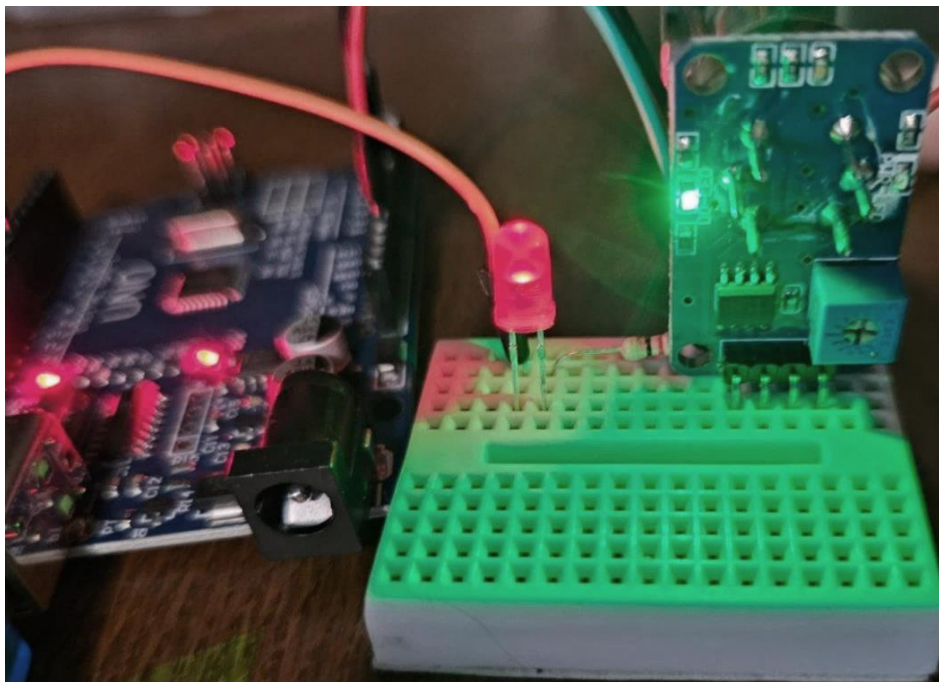


Рисунок 4.14 – Сигнал датчика про появу газу

4.2 Тестування роботи алгоритму

Тестування роботи алгоритму є ключовим етапом перевірки функціональності системи збору та обробки даних. Воно дозволяє оцінити

ефективність розроблених методів обробки сигналів від сенсорів, виявлення аномалій, а також передачі та збереження отриманих результатів. Результати тестування допоможуть визначити ступінь відповідності алгоритму поставленим вимогам і готовність системи до практичного застосування.

Для тестування алгоритму необхідно запустити симуляцію програмного коду у середовищі Webots. Візуалізація такого алгоритму у середовищі Webots виглядатиме наступним чином:

Крок 1 – Ініціалізація симуляції, робот штовхатиме фіолетовий елемент по траєкторії прямо перед ним.

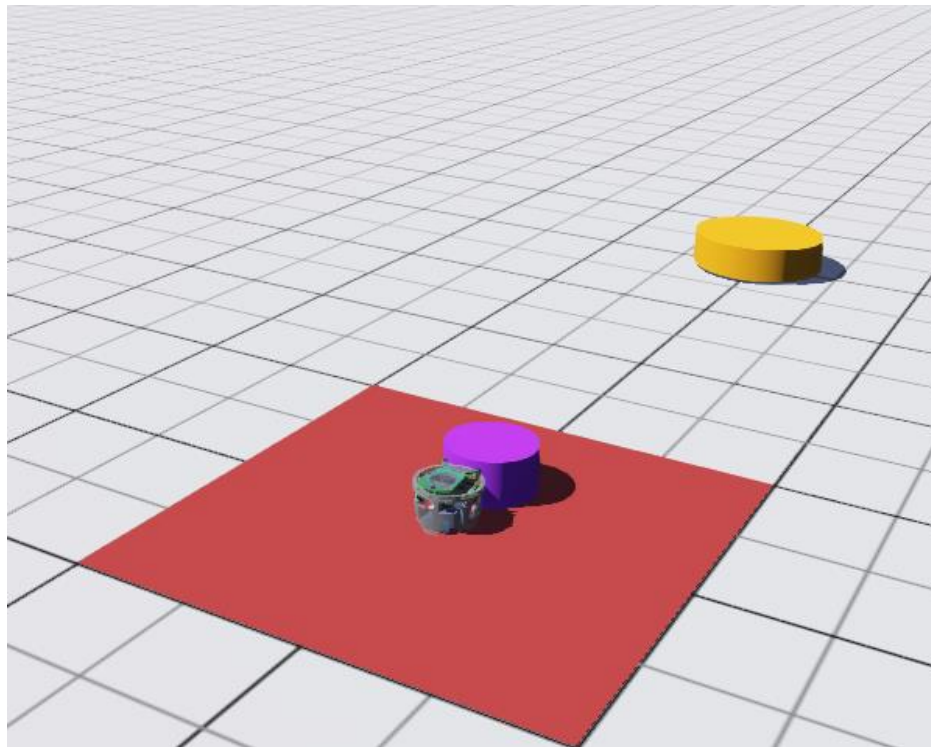


Рисунок 4.15 – Ініціалізація одного робота

Крок 2 – Переносячи «вантаж», робот стикається з перешкодою у вигляді жовтого елемента.

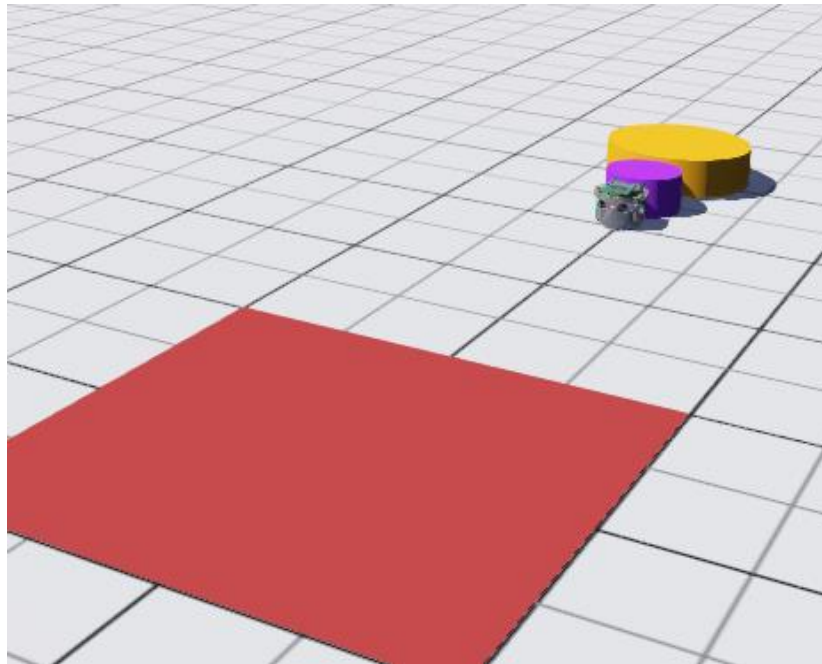


Рисунок 4.16 – Зіткнення з перешкодою

Крок 3 – Користувач середовища ініціює оптимізацію.

```

Console - All
---
running demo of best individual ...
select the 3D window and push the '0' key
to start genetic algorithm optimization
fitness: 0.500754
---
starting GA optimization ...
population size is 50, genome size is 16
generation: 0, genotype: 0

```

Рисунок 4.17 – Ініціалізація оптимізації

Крок 4 – Генерується до 50 поколінь, обирається найкраще покоління для досягнення головної мети – обхід перешкоди та досягнення точки «мети». Найкраще покоління зберігається у файлі, симуляція закінчується. При повторному відкритті програми відтворюється найкращий результат.

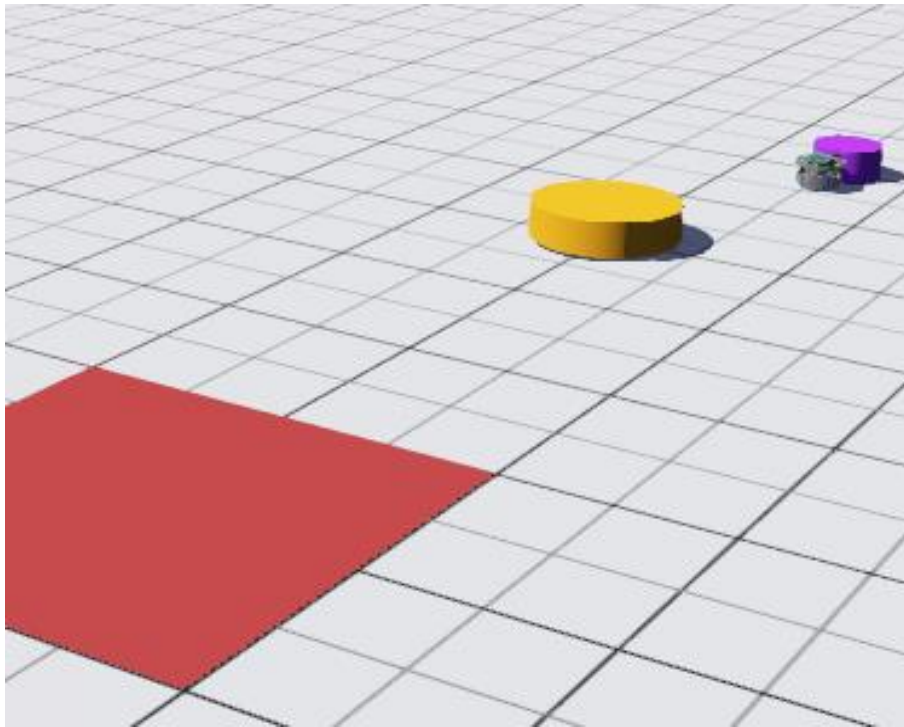


Рисунок 4.18 – Обхід перешкоди

Крок 5 – Додавання інших роботів, повтор алгоритму для перевірки функціонування в групі.

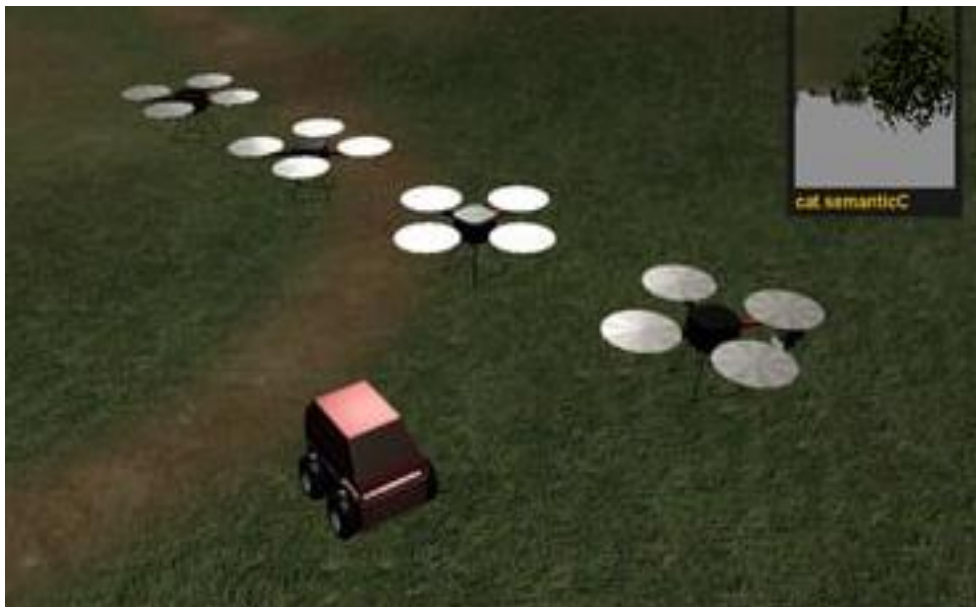


Рисунок 4.19 – Запуск групи БПЛА (роботів) у середовищі Webots

4.3 Висновки до четвертого розділу

У даному розділі було проведено тестування кожного сенсора з метою перевірки їх функціональності, точності отриманих даних і правильності підключення до плати мікроконтролера Arduino. Результати тестування підтвердили можливість використання сенсорів для моніторингу навколишнього середовища. Зокрема:

- Тестування DHT11: підтверджено, що датчик коректно вимірює температуру та вологість у різних умовах, демонструючи адекватну реакцію на зміни навколишнього середовища.
- Тестування KY-026: перевірено чутливість датчика до вогню; реалізовано вивід відповідних повідомлень у середовищі Arduino IDE.
- Тестування TMP36: встановлено, що датчик точно реагує на зміни температури та виводить коректні значення у середовищі Arduino.
- Тестування MQ2: датчик продемонстрував високу чутливість до наявності газів у повітрі, зокрема до пропану, що було перевірено за допомогою запальнички.

Також було проведено тестування алгоритму оптимізації роботи системи у середовищі Webots. Результати підтвердили ефективність алгоритму у виконанні наступних завдань:

- Ініціалізація симуляції та базовий рух БПЛА та групи БПЛА (внаслідок адаптації параметрів наземних роботів).
- Реакція на перешкоди під час пересування.
- Використання генетичного алгоритму для знаходження найбільш оптимального рішення.
- Генерація поколінь і вибір найкращого результату для досягнення мети.

У симуляції згенеровано 50 поколінь, що дозволило поступово покращити траєкторію руху групи роботів для уникнення перешкод та досягнення кінцевої мети. Оптимізація тривала приблизно 30 хвилин, протягом яких алгоритм досяг значного зменшення часу виконання завдання

– на 40% порівняно з початковим результатом.

Ці вдосконалення стали можливими завдяки адаптації генетичного алгоритму до умов тестування та правильному налаштуванню параметрів симуляції, таких як розмір популяції, кількість поколінь і точність оцінки функції пристосованості. Результати підтвердили здатність алгоритму забезпечувати ефективну адаптацію руху роботів у складному середовищі.

Як наслідок, алгоритм коректно обробляє сигнали від сенсорів, адаптує траєкторію та ефективно реалізує оптимізацію руху. Результати підтвердили його готовність до подальшої інтеграції у систему моніторингу повітря.

Даний розділ надає підтвердження працездатності як апаратної, так і програмної частини розробленої системи. Проведене тестування забезпечило впевненість у точності роботи сенсорів та ефективності реалізованого алгоритму для задач моніторингу й оптимізації.

5 ЕКОНОМІЧНА ЧАСТИНА

Виконання науково-дослідної роботи завжди передбачає отримання певних результатів і вимагає відповідних витрат. Результати виконаної роботи завжди дають нам нові знання, які в подальшому можуть бути використані для удосконалення та/або розробки (побудови) нових, більш продуктивних зразків техніки, процесів та програмного забезпечення.

Дослідження на тему «Інформаційна технологія оптимізації планування траєкторії групи БПЛА для оцінки забруднення повітря» може бути віднесено до фундаментальних і пошукових наукових досліджень і спрямоване на вирішення наукових проблем, пов'язаних з практичним застосуванням. Основою таких досліджень є науковий ефект, який виражається в отриманні наукових результатів, які збільшують обсяг знань про природу, техніку та суспільство, які розвивають теоретичну базу в тому чи іншому науковому напрямку, що дозволяє виявити нові закономірності, які можуть використовуватися на практиці.

Для цього випадку було виконано такі етапи робіт:

- 1) здійснено проведення наукового аудиту досліджень, тобто встановлення їх наукового рівня та значимості;
- 2) проведено планування витрат на проведення наукових досліджень;
- 3) здійснено розрахунок рівня важливості наукового дослідження та перспективності, визначено ефективність наукових досліджень.

5.1 Оцінювання наукового ефекту

Основними ознаками наукового ефекту науково-дослідної роботи є новизна роботи, рівень її теоретичного опрацювання, перспективність, рівень розповсюдження результатів, можливість реалізації. Науковий ефект НДР на

тему «Інформаційна технологія оптимізації планування траєкторії групи БПЛА для оцінки забруднення повітря» можна охарактеризувати двома показниками: ступенем наукової новизни та рівнем теоретичного опрацювання.

Значення показників ступеня новизни і рівня теоретичного опрацювання науково-дослідної роботи в балах наведені в таблицях 5.1 та 5.2.

Таблиця 5.1 – Показники ступеня новизни науково-дослідної роботи ви-
ставлені експертами

Ступінь новизни	Характеристика ступеня новизни	Значення ступеня новизни, бали		
		Експерти (ПІБ, посада)		
		1	2	3
Принципово нова	Робота якісно нова за постановкою задачі і ґрунтується на застосуванні оригінальних методів дослідження. Результати дослідження відкривають новий напрям в даній галузі науки і техніки. Отримані принципово нові факти, закономірності; розроблена нова теорія. Створено принципово новий пристрій, спосіб, метод	0	0	0
Нова	Отримана нова інформація, яка суттєво зменшує невизначеність наявних значень (по-новому або вперше пояснені відомі факти, закономірності, впроваджені нові поняття, розкрита структура змісту). Проведено суттєве вдосконалення, доповнення і уточнення раніше досягнутих результатів	55	59	56
Відносно нова	Робота має елементи новизни в постановці задачі і методах дослідження. Результати дослідження систематизують і узагальнюють наявну інформацію, визначають шляхи подальших досліджень; вперше знайдено зв'язок (або знайдено новий зв'язок) між явищами. В принципі відомі положення розповсюджені на велику кількість об'єктів, в результаті чого знайдено ефективне рішення. Розроблені більш прості способи для досягнення відомих результатів. Проведена часткова раціональна модифікація (з ознаками новизни)	0	0	0
Традиційна	Робота виконана за традиційною методикою. Результати дослідження мають інформаційний характер. Підтверджені або поставлені під сумнів відомі факти та твердження, які потребують перевірки. Знайдено новий варіант рішення, який не дає суттєвих переваг в порівнянні з існуючим	0	0	0
Не нова	Отримано результат, який раніше зафіксований в інформаційному полі, та не був відомий авторам	0	0	0

Середнє значення балів експертів	56,7
---	-------------

Згідно отриманого середнього значення балів експертів ступінь новизни характеризується як нова, тобто отримана нова інформація, яка суттєво зменшує невизначеність наявних знань (по-новому або вперше пояснені відомі факти, закономірності, впроваджені нові поняття, розкрита структура змісту) та проведено суттєве вдосконалення, доповнення і уточнення раніше досягнутих результатів.

Таблиця 5.2 – Показники рівня теоретичного опрацювання науково-дослідної роботи виставлені експертами

Характеристика рівня теоретичного опрацювання	Значення показника рівня теоретичного опрацювання, бали		
	Експерт (ІІБ, посада)		
	1	2	3
Відкриття закону, розробка теорії	0	0	0
Глибоке опрацювання проблеми: багатоаспектний аналіз зв'язків, взаємозалежності між фактами з наявністю пояснень, наукової систематизації з побудовою евристичної моделі або комплексного прогнозу	70	75	65
Розробка способу (алгоритму, програми), пристрою, отримання нової речовини	0	0	0
Елементарний аналіз зв'язків між фактами та наявною гіпотезою, класифікація, практичні рекомендації для окремого випадку тощо	0	0	0
Опис окремих елементарних фактів, викладення досвіду, результатів спостережень, вимірювань тощо	0	0	0
Середнє значення балів експертів	70,0		

Згідно отриманого середнього значення балів експертів рівень теоретичного опрацювання науково-дослідної роботи характеризується як глибоке опрацювання проблеми: багатоаспектний аналіз зв'язків, взаємозалежності між фактами з наявністю пояснень, наукової систематизації з побудовою евристичної моделі або комплексного прогнозу.

Показник, який характеризує рівень наукового ефекту, визначаємо за формулою [30]:

$$E_{\text{нау}} = 0,6 \cdot k_{\text{нов}} + 0,4 \cdot k_{\text{теор}}, \quad (5.1)$$

де $k_{\text{нов}}, k_{\text{теор}}$ - показники ступеня новизни та рівня теоретичного опрацювання науково-дослідної роботи, $k_{\text{нов}} = 56,7, k_{\text{теор}} = 70,0$ балів;

$0,6$ та $0,4$ – питома вага (значимість) показників ступеня новизни та рівня теоретичного опрацювання науково-дослідної роботи.

$$E_{\text{нау}} = 0,6 \cdot k_{\text{нов}} + 0,4 \cdot k_{\text{теор}} = 0,6 \cdot 56,7 + 0,4 \cdot 70,00 = 62,00 \text{ балів}. \quad (5.2)$$

Визначення характеристики показника $E_{\text{нау}}$ проводиться на основі висновків експертів виходячи з граничних значень, які наведені в табл. 5.3.

Таблиця 5.3 – Граничні значення показника наукового ефекту

Досягнутий рівень показника	Кількість балів
Високий	70...100
Середній	50...69
Достатній	15...49
Низький (помилкові дослідження)	1...14

Відповідно до визначеного рівня наукового ефекту проведеної науково-дослідної роботи на тему «Інформаційна технологія оптимізації планування траєкторії групи БПЛА для оцінки забруднення повітря», даний рівень становить 62,00 балів і відповідає статусу - середній рівень. Тобто у даному випадку можна вести мову про потенційну фактичну ефективність науково-дослідної роботи.

5.2 Розрахунок витрат на здійснення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Інформаційна технологія оптимізації планування траєкторії групи БПЛА для

оцінки забруднення повітря», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

5.2.1 Витрати на оплату праці

До статті «Витрати на оплату праці» належать витрати на виплату основної та додаткової заробітної плати керівникам відділів, лабораторій, секторів і груп, науковим, інженерно-технічним працівникам, конструкторам, технологам, креслярам, копіювальникам, лаборантам, робітникам, студентам, аспірантам та іншим працівникам, безпосередньо зайнятим виконанням конкретної теми, обчисленої за посадовими окладами, відрядними розцінками, тарифними ставками згідно з чинними в організаціях системами оплати праці.

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою [30]:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (5.3)$$

де k – кількість посад дослідників залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – число днів роботи конкретного дослідника, дн.;

T_p – середнє число робочих днів в місяці, $T_p = 22$ дні.

$$Z_o = 18600,00 \cdot 32 / 22 = 27054,55 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.4 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Керівник науково-дослідної роботи	18600,00	845,45	32	27054,55
Старший науковий співробітник	17500,00	795,45	12	9545,45

Консультант (оператор БПЛА)	18000,00	818,18	5	4090,91
Інженер-розробник ПЗ	18000,00	818,18	22	18000,00
Технік	8110,00	368,64	28	10321,82
Всього				69012,73

5.2.2 Основна заробітна плата робітників

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт НДР на тему «Інформаційна технологія оптимізації планування траєкторії групи БПЛА для оцінки забруднення повітря» розраховуємо за формулою:

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (5.4)$$

де C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинну тарифну ставку робітника відповідного розряду C_i можна визначити за формулою:

$$C_i = \frac{M_M \cdot K_i \cdot K_c}{T_p \cdot t_{зм}}, \quad (5.5)$$

де M_M – розмір прожиткового мінімуму працездатної особи, або мінімальної місячної заробітної плати (в залежності від діючого законодавства), приймемо $M_M=8000,00$ грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду (табл. Б.2, додаток Б) [30];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середнє число робочих днів в місяці, приблизно $T_p = 22$ дн;

$t_{зм}$ – тривалість зміни, год.

$$C_1 = 8000,00 \cdot 1,10 \cdot 1,15 / (22 \cdot 8) = 57,50 \text{ грн.}$$

$$З_{р1} = 57,50 \cdot 8,00 = 460,00 \text{ грн.}$$

Таблиця 5.5 – Величина витрат на основну заробітну плату робітників

Найменування робіт	Тривалість роботи, год	Розряд роботи	Тарифний коефіцієнт	Погодинна тарифна ставка, грн	Величина оплати на робітника грн
Установка обладнання для проведення пошукових робіт	8,00	2	1,10	57,50	460,00
Підготовка робочого місця наукового співробітника	5,00	3	1,35	70,57	352,84
Підготовка робочого місця інженера-розробника програмних засобів	4,50	4	1,50	78,41	352,84
Інсталяція програмного забезпечення для моделювання та розробки	6,00	4	1,50	78,41	470,45
Тренування системи взаємодії програмних функціональних блоків управління	8,00	4	1,50	78,41	627,27
Тестування підсистем	12,00	3	1,35	70,57	846,82
Монтаж інтерфейсних блоків функціонального управління	6,00	4	1,50	78,41	470,45
Всього					3580,68

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод} = (З_o + З_p) \cdot \frac{H_{дод}}{100\%}, \quad (5.6)$$

де $H_{дод}$ – норма нарахування додаткової заробітної плати. Прийmemo 11%.

$$З_{дод} = (69012,73 + 3580,68) \cdot 11 / 100\% = 7985,28 \text{ грн.}$$

5.2.3 Відрахування на соціальні заходи

Нарахування на заробітну плату дослідників та робітників розраховуємо як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$З_n = (З_o + З_p + З_{од}) \cdot \frac{H_{zn}}{100\%} \quad (5.7)$$

де H_{zn} – норма нарахування на заробітну плату. Приймаємо 22%.

$$З_n = (69012,73 + 3580,68 + 7985,28) \cdot 22 / 100\% = 17727,31 \text{ грн.}$$

5.2.4 Сировина та матеріали

До статті «Сировина та матеріали» належать витрати на сировину, основні та допоміжні матеріали, інструменти, пристрої та інші засоби і предмети праці, які придбані у сторонніх підприємств, установ і організацій та витрачені на проведення досліджень за темою «Інформаційна технологія оптимізації планування траєкторії групи БПЛА для оцінки забруднення повітря».

Витрати на матеріали на даному етапі проведення досліджень в основному пов'язані з використанням моделей елементів та моделювання роботи і досліджень за допомогою комп'ютерної техніки та створення експериментальних математичних моделей або програмного забезпечення, тому дані витрати формуються на основі витратних матеріалів характерних для офісних робіт.

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n H_j \cdot Ц_j \cdot K_j - \sum_{j=1}^n B_j \cdot Ц_{ej}, \quad (5.8)$$

де H_j – норма витрат матеріалу j -го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу j -го найменування, грн/кг;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$);

B_j – маса відходів j -го найменування, кг;

C_{vj} – вартість відходів j -го найменування, грн/кг.

$$M_1 = 2,0 \cdot 192,00 \cdot 1,02 - 0 \cdot 0 = 391,68 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.6 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Папір канцелярський офісний АМОСОOL-500 (A4)	192,00	2,0	0	0	391,68
Папір для заміток АМОСОOL-B (A5)/70	86,00	3,0	0	0	263,16
Начиння канцелярське АМОСОOL	148,00	2,0	0	0	301,92
Органайзер офісний АМОСОOL light	125,00	4,0	0	0	510,00
Картридж для принтера HP-7021	1560,00	2,00	0	0	3182,40
Диск оптичний COOL-CD/RW	25,50	4,0	0	0	104,04
FLASH-пам'ять KingsBAFF 32GB	120,00	2,0	0	0	244,80
Тека для паперів	46,00	4,0	0	0	187,68
Всього					5185,68

5.2.5 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_e), які використовують при проведенні НДР на тему «Інформаційна технологія оптимізації планування траєкторії групи

БПЛА для оцінки забруднення повітря», розраховуємо, згідно з їхньою номенклатурою, за формулою:

$$K_{\epsilon} = \sum_{j=1}^n H_j \cdot C_j \cdot K_j \quad (5.9)$$

де H_j – кількість комплектуючих j -го виду, шт.;

C_j – покупна ціна комплектуючих j -го виду, грн;

K_j – коефіцієнт транспортних витрат, ($K_j = 1,1 \dots 1,15$).

$K_{\epsilon} = 1 \cdot 5239,00 \cdot 1,02 = 5343,78$ грн.

Проведені розрахунки зведемо до таблиці.

Таблиця 5.7 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
Акумулятор для дрона RacePow Lipo 6s 12000mAh 60C XT60 (12000mAh-6s-60C)	1	5239,00	5343,78
Комплект кріплень	1	2460,00	2509,20
Пропеллери (комплект)	1	1359,00	1386,18
Мотор	2	3200,00	6528,00
Сенсори для вимірювання погодних умов	2	2799,00	5709,96
GPS-пристрої та системи навігації	2	6899,00	14073,96
Сенсори забруднення повітря	2	3250,00	6630,00
Модуль зв'язку	1	2600,00	2652,00
Всього			44833,08

5.2.6 Спецустаткування для наукових (експериментальних) робіт

До статті «Спецустаткування для наукових (експериментальних) робіт» належать витрати на виготовлення та придбання спецустаткування необхідного для проведення досліджень, також витрати на їх проектування, виготовлення, транспортування, монтаж та встановлення.

Балансову вартість спецустаткування розраховуємо за формулою:

$$B_{спец} = \sum_{i=1}^k C_i \cdot C_{пр.і} \cdot K_i , \quad (5.10)$$

де C_i – ціна придбання одиниці спецустаткування даного виду, марки, грн;

$C_{пр.і}$ – кількість одиниць устаткування відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує доставку, монтаж, налагодження устаткування тощо, ($K_i = 1,10 \dots 1,12$);

k – кількість найменувань устаткування.

$$B_{спец} = 26999,00 \cdot 1 \cdot 1,02 = 27538,98 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 5.8 – Витрати на придбання спецустаткування по кожному виду

Найменування устаткування	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Наземна станція для моніторингу та управління БПЛА (Ноутбук Xiaomi RedmiBook Pro 14 R7/16/512/W11)	1	26999,00	27538,98
Всього			27538,98

5.2.7 Програмне забезпечення для наукових (експериментальних) робіт

До статті «Програмне забезпечення для наукових (експериментальних) робіт» належать витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення, (програм, алгоритмів, баз даних) необхідних для проведення досліджень, також витрати на їх проектування, формування та встановлення.

Балансову вартість програмного забезпечення розраховуємо за формулою:

$$B_{прог} = \sum_{i=1}^k C_{прог} \cdot C_{прог.і} \cdot K_i , \quad (5.11)$$

де $C_{прог}$ – ціна придбання одиниці програмного засобу даного виду, грн;

$C_{\text{прг.і}}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_i – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, ($K_i = 1, 10 \dots 1, 12$);

k – кількість найменувань програмних засобів.

$$B_{\text{прг}} = 12800,00 \cdot 1 \cdot 1,02 = 13056,00 \text{ грн.}$$

Отримані результати зведемо до таблиці:

Таблиця 5.9 – Витрати на придбання програмних засобів по кожному виду

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
Спеціалізоване програмне забезпечення для планування маршрутів та оптимізації	1	12800,00	13056,00
Середовище симуляції Webots	1	6500,00	6630,00
Середовище розробки ArduinoIDE	1	4200,00	4284,00
Всього			23970,00

5.2.8 Амортизація обладнання, програмних засобів та приміщень

В спрощеному вигляді амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню тощо, розраховуємо з використанням прямолінійного методу амортизації за формулою:

$$A_{\text{обл}} = \frac{Ц_{\text{б}}}{T_{\text{е}}} \cdot \frac{t_{\text{вик}}}{12}, \quad (5.12)$$

де $Ц_{\text{б}}$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, грн;

$t_{\text{вик}}$ – термін використання обладнання, програмних засобів, приміщень під час досліджень, місяців;

$T_{\text{е}}$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{\text{обл}} = (29599,00 \cdot 2) / (3 \cdot 12) = 1644,39 \text{ грн.}$$

Проведені розрахунки зведемо до таблиці.

Таблиця 5.10 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Персональний комп'ютер HP ProBook 4540s	29599,00	3	2	1644,39
Обчислювально-графічна система програмної розробки	42888,00	3	2	2382,67
Робоче місце розробника програмного забезпечення	8599,00	5	2	286,63
Пристрій виводу текстової інформації	6500,00	4	2	270,83
Оргтехніка	8925,00	4	2	371,88
Приміщення лабораторії	520000,00	25	2	3466,67
Прикладний Офісний пакет Microsoft Office 2016	6540,00	2	2	545,00
OS Windows 11	6800,00	2	2	566,67
Всього				9534,73

5.2.9 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{gni}}{\eta_i}, \quad (5.13)$$

де W_{yi} – встановлена потужність обладнання на визначеному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), прийmemo $C_e = 10,98$ грн;

K_{eni} – коефіцієнт, що враховує використання потужності, $K_{eni} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$B_e = 0,35 \cdot 240,0 \cdot 10,98 \cdot 0,95 / 0,97 = 922,32$ грн.

Проведені розрахунки зведемо до таблиці.

Таблиця 5.11 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Персональний комп'ютер HP ProBook 4540s	0,35	240,0	922,32
Обчислювально-графічна система програмної розробки	0,25	240,0	658,80
Робоче місце розробника програмного забезпечення	0,12	150,0	197,64
Пристрій виводу текстової інформації	0,10	2,5	2,75
Оргтехніка	0,12	2,0	2,64
Наземна станція для моніторингу та управління БПЛА (Ноутбук Xiaomi RedmiBook Pro 14 R7/16/512/W11)	0,10	150,0	164,70
Система координації та забезпечення зв'язку	0,12	150,0	197,64
Всього			2146,48

5.2.10 Службові відрядження

До статті «Службові відрядження» дослідної роботи на тему «Інформаційна технологія оптимізації планування траєкторії групи БПЛА для оцінки забруднення повітря» належать витрати на відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, аспірантів, зайнятих розробленням досліджень, відрядження,

пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень.

Витрати за статтею «Службові відрядження» розраховуємо як 20...25% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cv} = (Z_o + Z_p) \cdot \frac{H_{cv}}{100\%}, \quad (5.14)$$

де H_{cv} – норма нарахування за статтею «Службові відрядження», прийmemo $H_{cv} = 20\%$.

$$B_{cv} = (69012,73 + 3580,68) \cdot 20 / 100\% = 14518,68 \text{ грн.}$$

5.2.11 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» розраховуємо як 30...45% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{cn} = (Z_o + Z_p) \cdot \frac{H_{cn}}{100\%}, \quad (5.15)$$

де H_{cn} – норма нарахування за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації», прийmemo $H_{cn} = 30\%$.

$$B_{cn} = (69012,73 + 3580,68) \cdot 30 / 100\% = 21778,02 \text{ грн.}$$

5.2.12 Інші витрати

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками.

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\text{е}} = (З_{\text{о}} + З_{\text{р}}) \cdot \frac{H_{\text{ів}}}{100\%}, \quad (5.16)$$

де $H_{\text{ів}}$ – норма нарахування за статтею «Інші витрати», прийmemo $H_{\text{ів}} = 50\%$.

$$I_{\text{е}} = (69012,73 + 3580,68) \cdot 50 / 100\% = 36296,70 \text{ грн.}$$

5.2.13 Накладні (загальновиробничі) витрати

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін.

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$B_{\text{нзв}} = (З_{\text{о}} + З_{\text{р}}) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (5.17)$$

де $H_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{\text{нзв}} = 100\%$.

$$B_{\text{нзв}} = (69012,73 + 3580,68) \cdot 100 / 100\% = 72593,41 \text{ грн.}$$

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія оптимізації планування траєкторії групи БПЛА для оцінки забруднення повітря» розраховуємо як суму всіх попередніх статей витрат за формулою:

$$B_{\text{заг}} = З_{\text{о}} + З_{\text{р}} + З_{\text{одд}} + З_{\text{н}} + M + K_{\text{е}} + B_{\text{спец}} + B_{\text{прз}} + A_{\text{обл}} + B_{\text{е}} + B_{\text{св}} + B_{\text{сп}} + I_{\text{е}} + B_{\text{нзв}}. \quad (5.18)$$

$$B_{\text{заг}} = 69012,73 + 3580,68 + 7985,28 + 17727,31 + 5185,68 + 44833,08 + 27538,98 + 23970,00 + 9534,73 + 2146,48 + 14518,68 + 21778,02 + 36296,70 + 72593,41 = 356701,76 \text{ грн.}$$

Загальні витрати $ЗВ$ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою:

$$ЗВ = \frac{B_{\text{заг}}}{\eta}, \quad (5.19)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta = 0,9$.

$$ЗВ = 356701,76 / 0,9 = 396335,29 \text{ грн.}$$

5.3 Оцінювання важливості та наукової значимості науково-дослідної роботи

Оцінювання та доведення ефективності виконання науково-дослідної роботи фундаментального чи пошукового характеру є достатньо складним процесом і часто базується на експертних оцінках, тому має вірогідний характер.

Для обґрунтування доцільності виконання науково-дослідної роботи на тему «Інформаційна технологія оптимізації планування траєкторії групи БПЛА для оцінки забруднення повітря» використовується спеціальний комплексний показник, що враховує важливість, результативність роботи, можливість впровадження її результатів у виробництво, величину витрат на роботу.

Комплексний показник K_p рівня науково-дослідної роботи може бути розрахований за формулою:

$$K_p = \frac{I^n \cdot T_c \cdot R}{B \cdot t}, \quad (5.20)$$

де I – коефіцієнт важливості роботи. Приймемо $I = 4$;

n – коефіцієнт використання результатів роботи; $n = 0$, коли результати роботи не будуть використовуватись; $n = 1$, коли результати роботи будуть використовуватись частково; $n = 2$, коли результати роботи будуть використовуватись в дослідно-конструкторських розробках; $n = 3$, коли результати можуть використовуватись навіть без проведення дослідно-конструкторських розробок. Прийmemo $n=3$;

T_C – коефіцієнт складності роботи. Прийmemo $T_C = 2$;

R – коефіцієнт результативності роботи; якщо результати роботи плануються вище відомих, то $R = 4$; якщо результати роботи відповідають відомому рівню, то $R = 3$; якщо нижче відомих результатів, то $R = 1$. Прийmemo $R = 4$;

B – вартість науково-дослідної роботи, тис. грн. Прийmemo $B = 396335,29$ грн;

t – час проведення дослідження. Прийmemo $t = 0,17$ років, (2 міс.).

Визначення показників I , n , T_C , R , B , t здійснюється експертним шляхом або на основі нормативів [30].

$$K_p = \frac{I^n \cdot T_C \cdot R}{B \cdot t} = 4^3 \cdot 2 \cdot 4 / 396,3 \cdot 0,17 = 7,75. \quad (5.21)$$

Якщо $K_p > 1$, то науково-дослідну роботу на тему «Інформаційна технологія оптимізації планування траєкторії групи БПЛА для оцінки забруднення повітря» можна вважати ефективною з високим науковим, технічним і економічним рівнем.

5.4 Висновок до п'ятого розділу

Витрати на проведення науково-дослідної роботи на тему «Інформаційна технологія оптимізації планування траєкторії групи БПЛА для оцінки забруднення повітря» складають 396335,29 грн. Відповідно до проведеного аналізу та розрахунків рівень наукового ефекту проведеної науково-дослідної роботи на тему «Інформаційна технологія оптимізації планування траєкторії

групи БПЛА для оцінки забруднення повітря» є середній, а дослідження актуальними, рівень доцільності виконання науково-дослідної роботи $K_p > 1$, що свідчить про потенційну ефективність з високим науковим, технічним і економічним рівнем.

ВИСНОВКИ

У цій роботі було спроектовано інформаційну технологію оптимізації планування траєкторії групи БПЛА для оцінки забруднення повітря.

У першому розділі було проведено аналіз існуючих систем і підходів до моніторингу якості повітря, включаючи наземні та безпілотні системи. Проаналізовано ефективність функціонування таких систем та їх алгоритмів. Було виявлено, що сучасні системи мають обмеження у точності, мобільності та гнучкості, що обґрунтовує необхідність створення адаптивної системи, здатної працювати у різних умовах.

Другий розділ присвячений порівняльному дослідженню генетичних алгоритмів. Розглянуто кілька типів, зокрема паралельні, адаптивні та меметичні. Обрано меметичний генетичний алгоритм, оскільки він найкраще відповідає вимогам задачі, забезпечуючи глобальний пошук і високу точність за обмежений час.

У третьому розділі було спроектовано апаратну частину системи: обрано мікроконтролер Arduino та набір сенсорів (DHT11, MQ-2, TMP36, KY-026). Реалізовано алгоритм взаємодії з сенсорами, адаптовано параметри та розроблено програмний код на мові C. Це дозволило створити стабільну платформу для отримання, обробки та передачі даних.

Четвертий розділ присвячений тестуванню роботи системи. Проведено перевірку кожного сенсора на коректність роботи в різних умовах. Тестування алгоритму оптимізації у середовищі Webots показало суттєве скорочення часу виконання завдання – з 50 до 30 хвилин, що становить зниження на 40%. Алгоритм генерував до 50 поколінь, обираючи найкращі рішення для досягнення мети.

Таким чином, розроблена система ефективно інтегрує апаратне забезпечення та алгоритми для моніторингу параметрів навколишнього середовища. Інтелектуальна система забезпечує можливість визначення лише

цільової території дослідження, автоматично генеруючи оптимальний маршрут для її обстеження. Вона демонструє високу точність, надійність та здатність до адаптації, що підтверджує доцільність її використання в реальних умовах.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кулик Я. А., Барановська А. Ю., Перенесення параметрів генетичних алгоритмів для задачі руху наземних роботів на БПЛА для оптимізації руху. *Контроль і управління в складних системах: тези міжнародної наук.-практ. конф. (м. Вінниця, 16 листопада – 17 листопада 2024 року) м. Вінниця, 2024.* URL: <https://conferences.vntu.edu.ua/index.php/mccs/mccs2024/paper/view/22080> (дата звернення: 09.09.2024)
2. Кулик Я. А., Барановська А. Ю., Оптимізація руху безпілотних літальних апаратів за допомогою генетичних алгоритмів. *«Молодь в науці: дослідження, проблеми, перспективи»: тези міжнародної наук.-практ. конф. (м. Вінниця, 15 жовтня 2024 року) м. Вінниця, 2024.* URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/schedConf/overview> (дата звернення: 09.09.2024)
3. Health impacts. *World Health Organization (WHO).* URL: <https://www.who.int/teams/environment-climate-change-and-health/air-quality-energy-and-health/health-impacts> (дата звернення: 09.09.2024)
4. Air Quality System (AQS) - Healthy People 2030 | odphp.health.gov. *Home of the Office of Disease Prevention and Health Promotion - health.gov.* URL: <https://odphp.health.gov/healthypeople/objectives-and-data/data-sources-and-methods/data-sources/air-quality-system-aqs> (дата звернення: 09.09.2024)
5. Air Quality Monitoring Stations. *Envira.* URL: <https://envira.global/AQMS-ambient-air-quality-monitoring-system/> (дата звернення: 09.09.2024)
6. . Unmanned Aerial Vehicles for Air Pollution Monitoring: A survey / N. H. Motlagh et al. *IEEE Internet of Things Journal.* 2023. P. 1. URL: <https://doi.org/10.1109/jiot.2023.3290508> (дата звернення: 09.09.2024)
7. Air Quality Monitoring and Forecasting using Smart Drones and Recurrent Neural Network for Sustainable Development in Chennai City / R. R. Hemamalini

et al. *Sustainable Cities and Society*. 2022. P. 104077. URL: <https://doi.org/10.1016/j.scs.2022.104077> (дата звернення: 09.09.2024)

8. Review on Drone-Assisted Air-Quality Monitoring Systems / P. Ashok Kokate et al. *Drones and Autonomous Vehicles*. 2023. Vol. 1, no. 1. P. 10005. URL: <https://doi.org/10.35534/dav.2023.10005> (дата звернення: 09.09.2024)

9. Salam Z., Ahmed J., Merugu B. S. The application of soft computing methods for MPPT of PV system: A technological and status review. *Applied Energy*. 2013. Vol. 107. P. 135–148. URL: <https://doi.org/10.1016/j.apenergy.2013.02.008> (дата звернення: 09.09.2024)

10. Swaminathan S. Gradient Descent Algorithms. *Medium*. URL: <https://medium.com/@saishruthi.tn/gradient-descent-algorithms-cefa1945a774> (дата звернення: 09.09.2024)

11. A* Search | Brilliant Math & Science Wiki. *Brilliant* | Learn interactively. URL: <https://brilliant.org/wiki/a-star-search/> (дата звернення: 09.09.2024)

12. GeeksforGeeks. Particle Swarm Optimization (PSO) - An Overview - GeeksforGeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/particle-swarm-optimization-pso-an-overview/> (дата звернення: 09.09.2024)

13. AnalytixLabs. A Complete Guide to Genetic Algorithm—Advantages, Limitations & More. *Medium*. URL: <https://medium.com/@byanalytixlabs/a-complete-guide-to-genetic-algorithm-advantages-limitations-more-738e87427dbb> (дата звернення: 09.09.2024)

14. Kanade V. Genetic Algorithms - Meaning, Working, and Applications - Spiceworks. *Spiceworks Inc.* URL: <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-are-genetic-algorithms/> (дата звернення: 09.09.2024)

15. Islier A. A. A genetic algorithm approach for multiple criteria facility layout design. *International Journal of Production Research*. 1998. Vol. 36, no. 6. P. 1549–1569. URL: <https://doi.org/10.1080/002075498193165> (дата звернення: 09.09.2024)

16. Poddar S. Parallel Genetic Algorithm. *Medium*. URL: <https://medium.com/swlh/parallel-genetic-algorithm-3d3314c8373c> (дата звернення: 09.09.2024)

17. Adaptive Genetic Algorithm Based on Mutation and Crossover and Selection Probabilities. *IEEE Xplore*. URL: <https://ieeexplore.ieee.org/document/9443124/> (дата звернення: 09.09.2024)

18. Deb I., Gupta R. K. A genetic algorithm based heuristic optimization technique for solving balanced allocation problem involving overall shipping cost minimization with restriction to the number of serving units as well as customer hubs. *Results in Control and Optimization*. 2023. P. 100227. URL: <https://doi.org/10.1016/j.rico.2023.100227> (дата звернення: 09.09.2024)

19. Neri F., Cotta C. Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*. 2012. Vol. 2. P. 1–14. URL: <https://doi.org/10.1016/j.swevo.2011.11.003> (дата звернення: 09.09.2024)

20. UNO R3. *Arduino Docs*. URL: <https://docs.arduino.cc/hardware/uno-rev3/> (дата звернення: 09.09.2024)

21. Hanewich T. Taking Flight with the Raspberry Pi Pico & MicroPython: DIY Quadcopter Drone. *Medium*. URL: <https://timhanewich.medium.com/taking-flight-with-the-raspberry-pi-pico-micropython-diy-quadcopter-drone-61ed4f7ee746> (дата звернення: 09.09.2024)

22. Staff L. E. In-Depth: How MQ2 Gas/Smoke Sensor Works? & Interface it with Arduino. *Last Minute Engineers*. URL: <https://lastminuteengineers.com/mq2-gas-senser-arduino-tutorial/> (дата звернення: 09.09.2024)

23. TMP36 Temperature Sensor. *Adafruit Learning System*. URL: <https://learn.adafruit.com/tmp36-temperature-sensor/overview> (дата звернення: 09.09.2024)

24. KY-026 Flame Sensor Module - ArduinoModulesInfo. *ArduinoModulesInfo*. URL: <https://arduinomodules.info/ky-026-flame-sensor-module/> (дата звернення: 09.09.2024)

25. DHT11–Temperature and Humidity Sensor. *Components101*. URL: <https://components101.com/sensors/dht11-temperature-sensor> (дата звернення: 09.09.2024)
26. Rocha G. D. #101 Developing Autonomous Drones Using Python and AI. *Medium*. URL: <https://medium.com/@genedarocha/101-developing-autonomous-drones-using-python-and-ai-3a85bc79a80b> (дата звернення: 09.09.2024)
27. The Role of C++ in Drone Technology. *Cppcat*. URL: <https://cpp-cat.com/c-in-drone-technology/> (дата звернення: 09.09.2024)
28. Complete Guide to Drone Programming. *Turing Journal*. URL: <https://www.turingjournal.com/complete-guide-to-drone-programming-for-beginners> (дата звернення: 09.09.2024)
29. Flying a DJI Tello Drone with React and Node.js. *Wes Bos*. URL: <https://wesbos.com/drone-javascript> (дата звернення: 09.09.2024)
30. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с.

Додатки

Додаток А
(Обов'язковий)

Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації

ПОГОДЖЕНО

Керівник або заступник
ТОВ «Тайгергруп»

(підпис) Гліб МИСИК
(Ім'я, ПРИЗВИЩЕ)
«14» жовтня 2024 р.

ЗАТВЕРДЖУЮ

Завідувач кафедри АІТ

(підпис) д.т.н., проф. Олег БІСІКАЛО
(науковий ступінь, вчене звання)
«14» жовтня 2024 року

ТЕХНІЧНЕ ЗАВДАННЯ

на магістерську кваліфікаційну роботу

Інформаційна технологія оптимізації планування траєкторії групи
безпілотних літальних апаратів для оцінки забруднення повітря

08-31.МКР.001.02.000 ТЗ

Керівник: к.т.н., доц. каф. АІТ

Ярослав КУЛИК

« 11 » жовтня 2024 р.

Розробила студентка гр. ІІСТ-23м

Анастасія БАРАНОВСЬКА

« 11 » жовтня 2024 р.

Вінниця 2024

1. Назва та галузь застосування.

Інформаційна технологія оптимізації планування траєкторії групи безпілотних літальних апаратів для оцінки забруднення повітря. Інформаційні системи та технології.

2. Підстава для проведення робіт.

Підставою для виконання роботи є наказ №__ по ВНТУ від «17» вересня 2024 р., та індивідуальне завдання на МКР, затверджене протоколом №__ засідання кафедри АІТ від «__» _____ 2024 р.

3. Мета та призначення роботи.

Метою дослідження є розробка інформаційної технології для оптимізації планування траєкторій групи безпілотних літальних апаратів (БПЛА) з метою оцінки рівня забруднення повітря.

4. Джерела розробки:

- 1) GeeksforGeeks. C Programming Language Tutorial - GeeksforGeeks. *GeeksforGeeks*. URL: <https://www.geeksforgeeks.org/c-programming-language/> (дата звернення: 02.10.2024).
- 2) UNO R3. Arduino Docs. URL: <https://docs.arduino.cc/hardware/uno-rev3/> (дата звернення: 06.10.2024).
- 3) Cyberbotics: Robotics simulation with Webots. *Cyberbotics: Robotics simulation with Webots*. URL: <https://cyberbotics.com/> (дата звернення 08.10.2024)
- 4) Complete Guide to Drone Programming. Turing Journal. URL: <https://www.turingjournal.com/complete-guide-to-drone-programming-for-beginners> (дата звернення 09.10.2024)
- 5) Staff L. E. In-Depth: How MQ2 Gas/Smoke Sensor Works? & Interface it with Arduino. Last Minute Engineers. URL: <https://lastminuteengineers.com/mq2-gas-sensor-arduino-tutorial/> (дата звернення 09.10.2024)
- 6) TMP36 Temperature Sensor. Adafruit Learning System. URL: <https://learn.adafruit.com/tmp36-temperature-sensor/overview> (дата звернення: 09.10.2024)

- 7) KY-026 Flame Sensor Module - ArduinoModulesInfo. ArduinoModulesInfo. URL: <https://arduinomodules.info/ky-026-flame-sensor-module/> (дата звернення: 10.10.2024)
- 8) DHT11 – Temperature and Humidity Sensor. Components101. URL: <https://components101.com/sensors/dht11-temperature-sensor> (дата звернення: 10.09.2024)
- 9) Getting Started With Drone Programming. Nilebits Digital Solutions. URL: <https://www.nilebits.com/blog/2022/10/getting-started-with-drone-programming/> (дата звернення: 11.09.2024)
- 10) Ballesteros J. R. A Simple Guide to Code using a Drone. Medium. URL: <https://medium.com/@jrballesteros/a-simple-guide-to-code-using-a-drone-8bbf398bc40f> (дата звернення: 12.09.2024)

5. Показники призначення

Основні технічні вимоги та мінімальні системні вимоги до програми: безпілотний літальний апарат, оперативна пам'ять не менше 4 ГБ, ємність акумулятора не менше 5000 мАг, сенсори MQ-2, DHT11, TMP36, KY-026, мова програмування C.

Фактори, які обмежуватимуть розробку: швидкість вітру не повинна перевищувати 10 м/с, оскільки при показниках понад 12 м/с траєкторія стає нестійкою, збільшуючи похибку маршруту до 15%. Оптимальний температурний діапазон для польоту — від -10°C до +35°C. При температурі нижче -10°C ємність акумулятора зменшується на 30%. Акумулятор із ємністю 5000 мА/год забезпечує близько 10 хв польоту на відстань до 5 км в одну сторону. Висота польоту обмежується 150 м, оскільки при підйомі вище цього рівня енергоспоживання зростає на 10% через збільшений опір повітря. Концентрація шкідливих домішок у повітрі має перевищувати 5 мг/м³, оскільки саме ці дані є метою вимірювання. Зміна вітрових умов через 60 хв може підвищити похибку оцінок забруднення на 20%, що робить дотримання тимчасових обмежень критично важливим.

Результати роботи програми: створення технології, яка забезпечує ефективне керування польотом групи БПЛА для проведення точного та швидкого моніторингу забруднення повітря. Застосування генетичних алгоритмів для планування траєкторії групи БПЛА дозволяє зменшити час, необхідний для збору екологічних даних, шляхом оптимізації розташування дронів у просторі та мінімізації дублювання маршрутів. Це забезпечує більш ефективний моніторинг та скорочує витрати на виконання операцій, підвищуючи якість даних та зменшуючи ризики для здоров'я людей, оскільки БПЛА можуть виконувати моніторинг у небезпечних зонах без прямої участі людини.

6. Економічні показники

До економічних показників входять:

- витрати на розробку – до 40 тис. грн.
- мінімальна дохідність – не менше 60 тис. грн.
- термін окупності – не більше 1 року

7. Стадії розробки:

1. Аналіз предметної області
2. Вибір оптимальних інформаційних технологій
3. Вибір мови програмування та середовища розробки
4. Програмна реалізація
5. Експериментальна перевірка роботи програми
6. Оформлення пояснювальної записки, графічного матеріалу і презентації

8. Порядок контролю та приймання

Рубіжний контроль провести до «15» _____ листопада _____ 2024 р.

Попередній захист МКР провести до «05» _____ грудня _____ 2024 р.

Захист МКР провести до «20» _____ грудня _____ 2024 р.

Розробила студентка групи ІСТ-23м _____ Анастасія БАРАНОВСЬКА

Додаток Б
(Обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА
ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ОПТИМІЗАЦІЇ ПЛАНУВАННЯ
ТРАЄКТОРІЇ ГРУПИ БЕЗПЛОТНИХ ЛІТАЛЬНИХ АПАРАТІВ ДЛЯ
ОЦІНКИ ЗАБРУДНЕННЯ ПОВІТРЯ



Рисунок Б.1 – UML діаграма класів програми

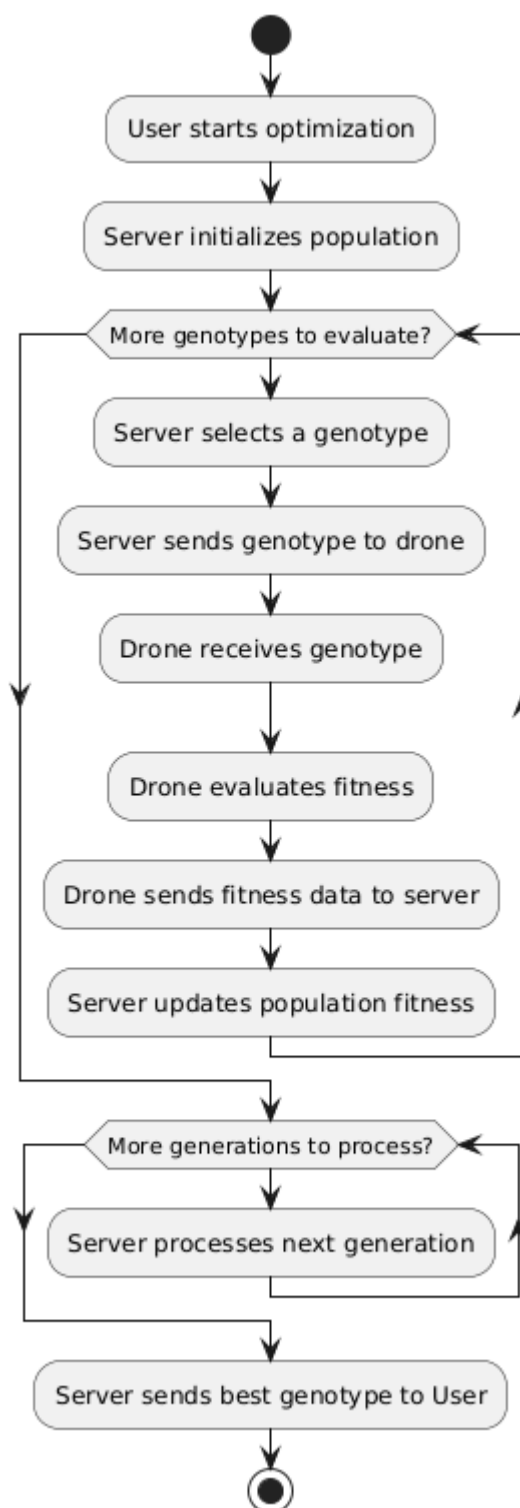


Рисунок Б.2 – UML діаграма активностей системи

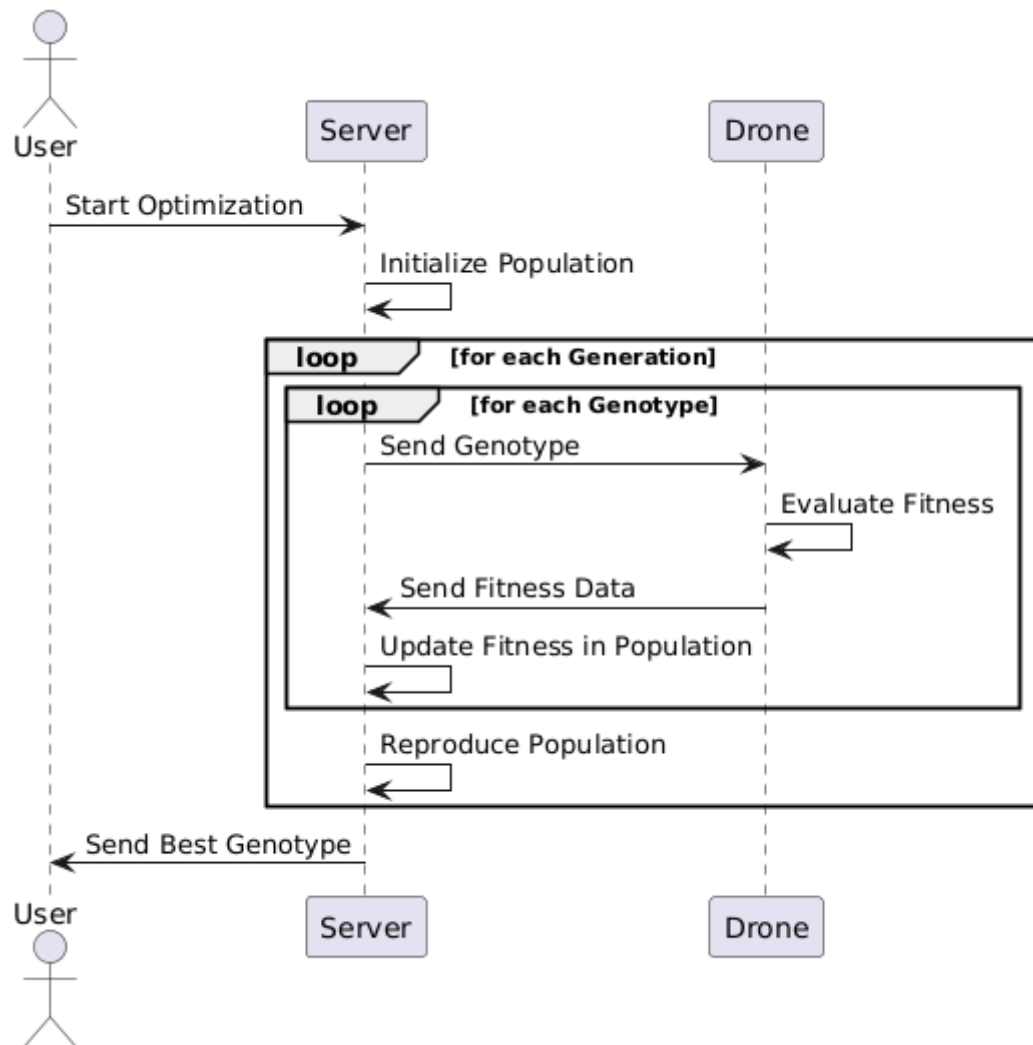


Рисунок Б.3 – UML діаграма послідовності

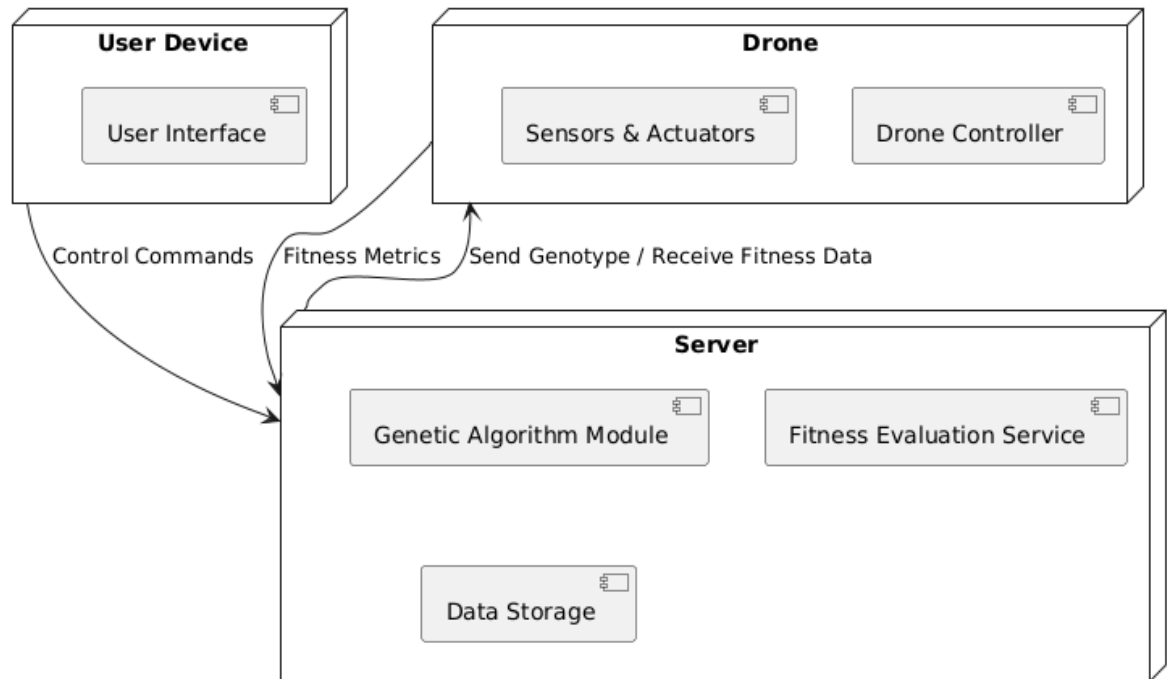


Рисунок Б.4 – UML діаграма розгортання системи

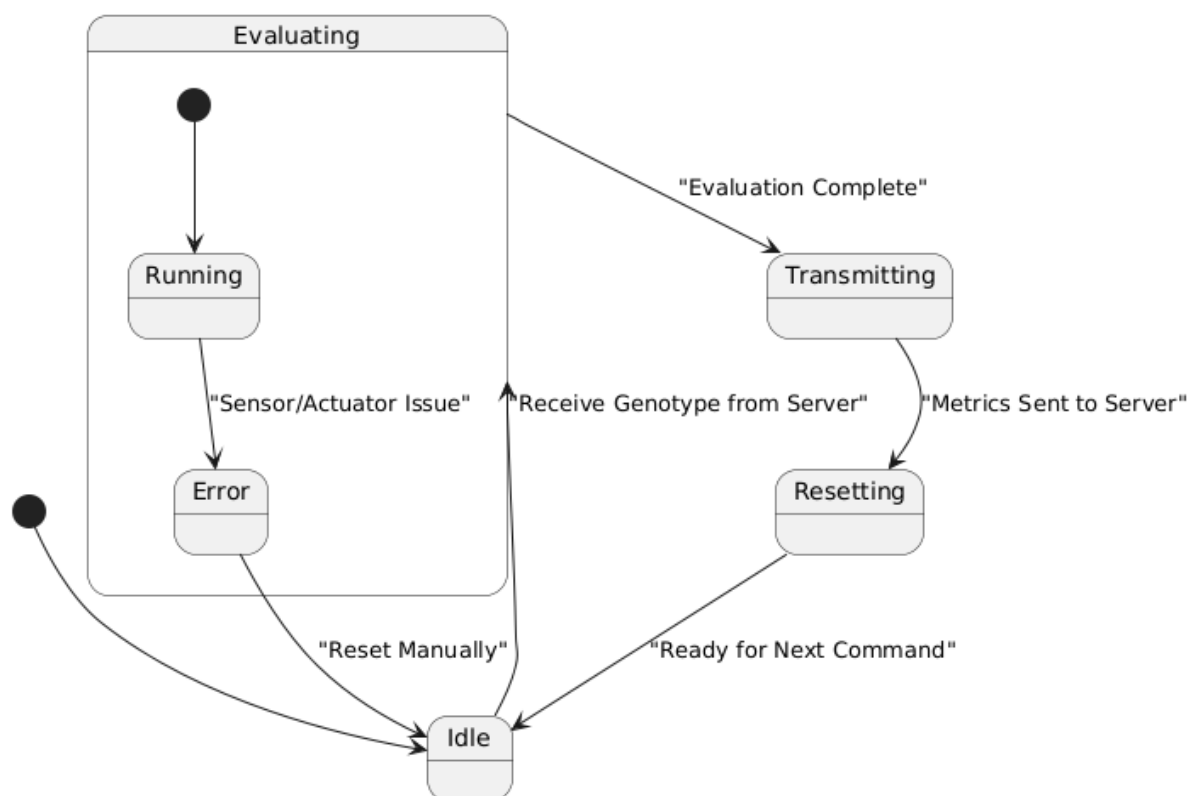


Рисунок Б.5 – UML діаграма станів

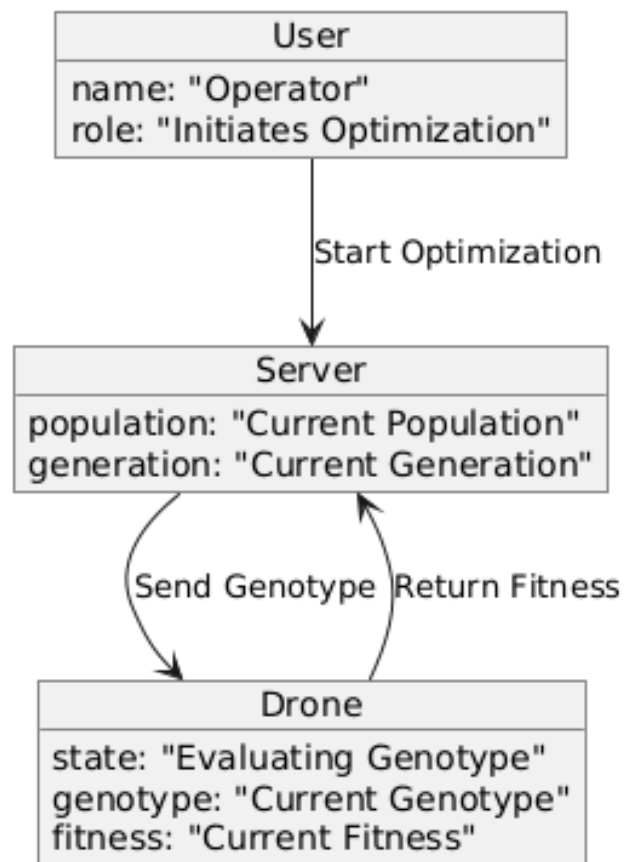


Рисунок Б.6 – UML діаграма об'єктів

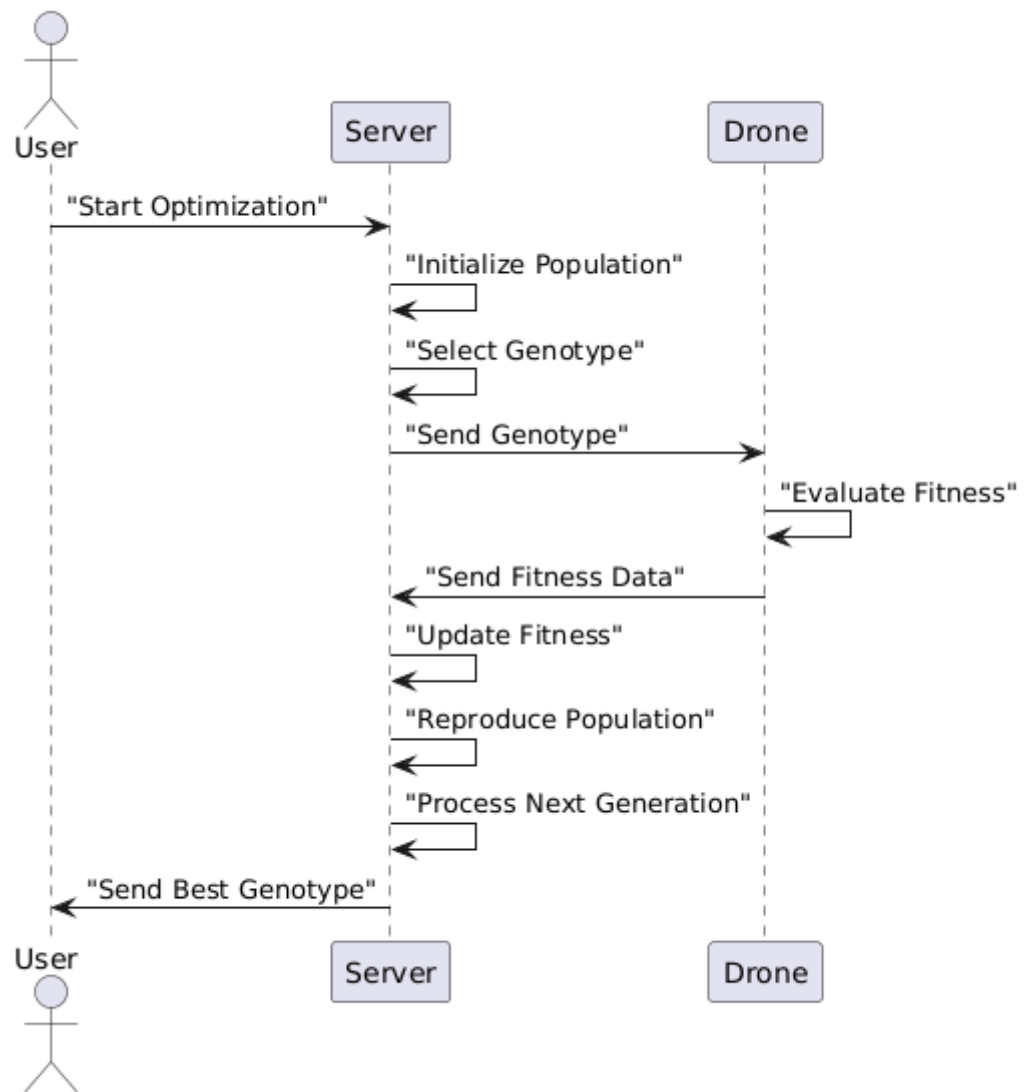


Рисунок Б.7 – UML діаграма колаборації

Додаток В
(Обов'язковий)
Лістинг коду

```
#include <math.h>
#include <stdio.h>
#include <string.h>
#include <webots/display.h>
#include <webots/emitter.h>
#include <webots/keyboard.h>
#include <webots/robot.h>
#include <webots/supervisor.h>
#include "genotype.h"
#include "population.h"

static const int POPULATION_SIZE = 50;
static const int NUM_GENERATIONS = 25;
static const char *FILE_NAME = "fittest.txt";
// must match the values in the advanced_genetic_algorithm.c code
static const int NUM_SENSORS = 8;
static const int NUM_WHEELS = 2;
#define GENOTYPE_SIZE (NUM_SENSORS * NUM_WHEELS)
// index access
enum { X, Y, Z };
static int time_step;
static WbDeviceTag emitter; // to send genes to robot
static WbDeviceTag display; // to display the fitness evolution
static int display_width, display_height;
// the GA population
static Population population;

// for reading or setting the robot's position and orientation
static WbFieldRef robot_translation;
static WbFieldRef robot_rotation;
static double robot_trans0[3]; // a translation needs 3 doubles
static double robot_rot0[4]; // a rotation needs 4 doubles
// for reading or setting the load's position
static WbFieldRef load_translation;
static double load_trans0[3];
// start with a demo until the user presses the 'O' key
// (change this if you want)
static bool demo = true;
```

```

void draw_scaled_line(int generation, double y1, double y2) {
    const double XSCALE = (double)display_width / NUM_GENERATIONS;
    const double YSCALE = 10.0;
    wb_display_draw_line(display, (generation - 0.5) * XSCALE, display_height -
y1 * YSCALE, (generation + 0.5) * XSCALE,
                        display_height - y2 * YSCALE);
}
// plot best and average fitness
void plot_fitness(int generation, double best_fitness, double average_fitness)
{
    static double prev_best_fitness = 0.0;
    static double prev_average_fitness = 0.0;
    if (generation > 0) {
        wb_display_set_color(display, 0xff0000); // red
        draw_scaled_line(generation, prev_best_fitness, best_fitness);
        wb_display_set_color(display, 0x00ff00); // green
        draw_scaled_line(generation, prev_average_fitness, average_fitness);
    }
    prev_best_fitness = best_fitness;
    prev_average_fitness = average_fitness; }
// run the robot simulation for the specified number of seconds
void run_seconds(double seconds) {
    int i, n = 1000.0 * seconds / time_step;
    for (i = 0; i < n; i++) {
        if (demo && wb_keyboard_get_key() == 'O') {
            demo = false;
            return; // interrupt demo and start GA optimization
        }
        wb_robot_step(time_step);
    }
}
// compute fitness as the euclidian distance that the load was pushed
double measure_fitness() {
    const double *load_trans =
wb_supervisor_field_get_sf_vec3f(load_translation);
    double dx = load_trans[X] - load_trans0[X];
    double dz = load_trans[Z] - load_trans0[Z];
    return sqrt(dx * dx + dz * dz);
}
// evaluate one genotype at a time
void evaluate_genotype(Genotype genotype) {
    // send genotype to robot for evaluation
    wb_emitter_send(emitter, genotype_get_genes(genotype), GENOTYPE_SIZE *

```

```

sizeof(double));
    // reset robot and load position
    wb_supervisor_field_set_sf_vec3f(robot_translation, robot_trans0);
    wb_supervisor_field_set_sf_rotation(robot_rotation, robot_rot0);
    wb_supervisor_field_set_sf_vec3f(load_translation, load_trans0);
    // evaluation genotype during one minute
    run_seconds(60.0);

    // measure fitness
    double fitness = measure_fitness();
    genotype_set_fitness(genotype, fitness);
    printf("fitness: %g\n", fitness);
    WbNodeRef robot = wb_supervisor_node_get_from_def("ROBOT");
    wb_supervisor_node_reset_physics(robot); }
void run_optimization() {
    wb_keyboard_disable();
    printf("---\n");
    printf("starting GA optimization ...\n");
    printf("population size is %d, genome size is %d\n", POPULATION_SIZE,
GENOTYPE_SIZE);
    int i, j;
    for (i = 0; i < NUM_GENERATIONS; i++) {
        for (j = 0; j < POPULATION_SIZE; j++) {
            printf("generation: %d, genotype: %d\n", i, j);
// evaluate genotype
            Genotype genotype = population_get_genotype(population, j);
            evaluate_genotype(genotype); }
        double best_fitness =
genotype_get_fitness(population_get_fittest(population));
        double average_fitness = population_compute_average_fitness(population);
        // display results
        plot_fitness(i, best_fitness, average_fitness);
        printf("best fitness: %g\n", best_fitness);
        printf("average fitness: %g\n", average_fitness);
        // reproduce (but not after the last generation)
        if (i < NUM_GENERATIONS - 1)
            population_reproduce(population); }
    printf("GA optimization terminated.\n");
    // save fittest individual
    Genotype fittest = population_get_fittest(population);
    FILE *outfile = fopen(FILE_NAME, "w");
    if (outfile) {
        genotype_fwrite(fittest, outfile);

```

```

        fclose(outfile);
        printf("wrote best genotype into %s\n", FILE_NAME);
    } else
        printf("unable to write %s\n", FILE_NAME);
    population_destroy(population); }
// show demo of the fittest individual
void run_demo() {
    wb_keyboard_enable(time_step);
    printf("---\n");
    printf("running demo of best individual ...\n");
    printf("select the 3D window and push the 'O' key\n");
    printf("to start genetic algorithm optimization\n");
    FILE *infile = fopen(FILE_NAME, "r");
    if (!infile) {
        printf("unable to read %s\n", FILE_NAME);
        return; }
    Genotype genotype = genotype_create();
    genotype_fread(genotype, infile);
    fclose(infile);
    while (demo)
        evaluate_genotype(genotype); }
int main(int argc, const char *argv[]) {
    // initialize Webots
    wb_robot_init();
    // get simulation step in milliseconds
    time_step = wb_robot_get_basic_time_step();
    // the emitter to send genotype to robot
    emitter = wb_robot_get_device("emitter");
    // to display the fitness evolution
    display = wb_robot_get_device("display");
    display_width = wb_display_get_width(display);
    display_height = wb_display_get_height(display);
    wb_display_draw_text(display, "fitness", 2, 2);
    // initial population
    population = population_create(POPULATION_SIZE, GENOTYPE_SIZE);
    // find robot node and store initial position and orientation
    WbNodeRef robot = wb_supervisor_node_get_from_def("ROBOT");
    robot_translation = wb_supervisor_node_get_field(robot, "translation");
    robot_rotation = wb_supervisor_node_get_field(robot, "rotation");
    memcpy(robot_trans0, wb_supervisor_field_get_sf_vec3f(robot_translation),
sizeof(robot_trans0));
    memcpy(robot_rot0, wb_supervisor_field_get_sf_rotation(robot_rotation),
sizeof(robot_rot0));

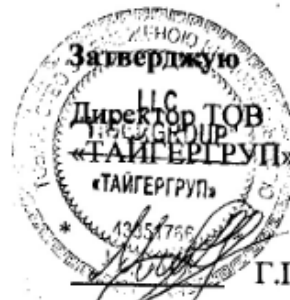
```

```
// find load node and store initial position
WbNodeRef load = wb_supervisor_node_get_from_def("LOAD");
load_translation = wb_supervisor_node_get_field(load, "translation");
memcpy(load_trans0,      wb_supervisor_field_get_sf_vec3f(load_translation),
sizeof(load_trans0));
if (demo)
    run_demo();
// run GA optimization
run_optimization();
// cleanup Webots
wb_robot_cleanup();
return 0; // ignored
}
```

Додаток Г
(Обов'язковий)

Акт впровадження системи

ТОВ «ТАЙГЕРГРУП»/LLC TIGERGROUP
21000, Україна, Вінницький р-н, Вінницька обл.
місто Вінниця, вулиця Бандери Степана, будинок, 9.
ЄДРПОУ 43351766



Г.І. Мисик

«23» листопада 2024 року

АКТ

Впровадження результатів магістерської дипломної роботи Барановської Анастасії Юріївни на тему: «Інформаційна технологія оптимізації планування траєкторії групи БПЛА для оцінки забруднення повітря»

Залучений фахівець ТОВ «ТАЙГЕРГРУП» з розробки програмного забезпечення Мисика Гліба Ігоровича розглянув магістерську дипломну роботу Барановської Анастасії Юріївни та встановив, що використання запропонованого програмного забезпечення, дозволяє забезпечити функціонал, збору та обробки даних про стан довкілля, задовольняє вимогам по ресурсоемкості, швидкодії та стійкості до навантаження.

Актуальність та практична цінність роботи Барановської А.Ю. не викликає сумнівів. Програмне забезпечення за рахунок гнучкого налаштування, дозволяє забезпечити збір та обробку даних про стан довкілля для клієнтів на різних платформах.

Беручи до уваги вищевказані переваги, результати роботи було впроваджено до системи обслуговування бізнесів, що отримують від компанії послуги з аналітики якості повітря.

Представник установи:

Г.І. Мисик

Додаток Д
(Обов'язковий)

Протокол перевірки магістерської кваліфікаційної роботи на наявність
текстових запозичень

Назва роботи: «Інформаційна технологія оптимізації планування траєкторії групи БПЛА для оцінки забруднення повітря»

Тип роботи: магістерська кваліфікаційна робота

Підрозділ: кафедра автоматизації та інтелектуальних інформаційних технологій

Показники звіту подібності Turnitin

Оригінальність 98,0 %

Схожість 2,0 %

Аналіз звіту подібності (відмітити потрібне):

- Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.
- Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____
(підпис)

Роман МАСЛІЙ

Ознайомлені з повним звітом подібності, який був згенерований системою Turnitin щодо роботи.

Автор роботи _____
(підпис)

Анастасія БАРАНОВСЬКА

Керівник роботи _____
(підпис)

Ярослав КУЛИК