

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

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«ІНФОРМАЦІЙНО - ВИМІРЮВАЛЬНА СИСТЕМА ВИЗНАЧЕННЯ
ПЕРЕШКОД ТА ЇХ ОМИНАННЯ»**

Виконав: студент 2-го курсу, групи КІВТ-24м
спеціальності 175 – Інформаційно -
вимірювальні технології

(шифр і назва напрямку підготовки, спеціальності)

Гайдамашко А.О.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. ІРТС

Притула М.О.
(прізвище та ініціали)

«15» 2025 р.

Опонент: к.т.н., доцент каф. ІКСТ

Стальченко О.В.
(прізвище та ініціали)

«16» 2025 р.

Допущено до захисту
Завідувач кафедри ІРТС
д.т.н., проф. Осадчук О.В.

(прізвище та ініціали)
«17» 2025 р.

Факультет інформаційних електронних систем
Кафедра інформаційних радіоелектронних технологій і систем
Рівень вищої освіти II-й (магістерський)
Галузь знань – 17 Електроніка, автоматизація та електронні комунікації
Спеціальність – 175 Інформаційно-вимірювальні технології
Освітньо-професійна програма – Комп'ютеризовані інформаційно-вимірювальні технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІРТС

д.т.н., проф. Осадчук О.В.

«16» вересня 2025 року

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

Гайдамашку Артему Олексійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи «Інформаційно - вимірювальна система визначення перешкод та їх оминання»

керівник роботи к.т.н., доцент кафедри ІРТС Притула М.О.

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

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

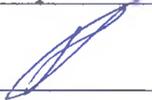
2. Строк подання студентом роботи 15.12.2025р.

3. Вихідні дані до роботи: Напруга живлення – 5 В. Струм споживання – 350...800 мА. Діапазон визначення відстані – 2...400 см. Похибка вимірювання – до ± 1.7 см. Порогове значення спрацьовування – 20...30 см. Час обробки інформації – 0,01...0,1 с. Кут повороту сервоприводу – 0...180°.

4. Зміст текстової частини: Вступ. Аналіз сучасних систем визначення перешкод. Розробка інформаційно-вимірювальної системи. Моделювання роботи дослідного зразку. Економічна частина. Висновки. Список використання джерел. Додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень): Схематична діаграма методу на основі активного маяка. Методи вимірювання відстані сенсором: ехолокаційний метод та односпрямований метод. Схематичне зображення шарів базової архітектури. Процедура оминання перешкод. Спрощена структурна схема логічних взаємозв'язків рухомої платформи. Дослідний зразок готовий до тестування. Перевірка точності сенсору за допомогою лінійки та перешкоди. Результати сенсору при порівнянні. Рух сервомотору на 45°.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
Основна частина	Притула М.О. к.т.н., доцент каф. ІРТС		
Економічна частина	Кавецький В.В. доцент каф. ЕПВМ, доцент, к.е.н.		

7. Дата видачі завдання 16.09.2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Вибір, узгодження та затвердження тем МКР на випусковій кафедрі.	02.09.2025-05.09.2025	
2.	Огляд та аналіз літературних джерел.	07.09.2025-14.09.2025	
3.	Затвердження тем по ВНТУ. Розробка завдання на МКР.	15.09.2025-28.09.2025	
4.	Попередня розробка основних розділів. Аналіз вирішення поставленої задачі. Розробка структурної схеми та технічних рішень.	30.09.2025-20.10.2025	
5.	Математичне моделювання та електричні розрахунки. Експериментальне дослідження.	21.10.2025-03.11.2025	
6.	Розробка ілюстративної частини МКР.	28.10.2025-01.11.2025	
7.	Економічна частина.	02.11.2025-12.11.2025	
8.	Оформлення пояснювальної записки та ілюстративної частини.	13.11.2025-17.11.2025	
9.	Нормоконтроль.	29.11.2025-03.12.2025	
10.	Попередній захист МКР, доопрацювання, рецензування МКР.	04.12.2025-17.12.2025	
11.	Захист МКР ЕК.	18.12.2025-19.12.2025	

Студент


(підпис)

Гайдамашко А.О.

Керівник роботи


(підпис)

Притула М.О.

АНОТАЦІЯ

УДК 621.391

Гайдамашко А.О. Інформаційно – вимірювальна система визначення перешкод та їх оминання. Магістерська кваліфікаційна робота зі спеціальності 175 – інформаційно-вимірювальні технології, освітня програма – комп’ютеризовані інформаційно-вимірювальні технології. – Вінниця: ВНТУ, 2025. – 118 с. На українській мові. Бібліогр.: 22 назв; Табл. 9; Рис. 33.

У магістерській кваліфікаційній роботі досліджено та розроблено інформаційно-вимірювальну систему для визначення перешкод на шляху рухомої платформи та їх оминання. У першому розділі проведено аналіз сучасних систем виявлення перешкод, розглянуто класифікацію сенсорів, технології сенсорної обробки даних та архітектури систем уникнення перешкод. У другому розділі представлено розробку системи: обґрунтовано вибір ультразвукового сенсора, механізмів руху та повороту, створено загальну і детальну структуру системи, а також розроблено програмне забезпечення для її керування. У третьому розділі наведено практичну реалізацію розробленої системи, описано процес складання, підключення компонентів, а також тестування алгоритмів визначення перешкод і їх оминання в реальних умовах роботи. Результати дослідження можуть бути використані при створенні автономних мобільних роботів, систем технічного зору та інших робототехнічних комплексів, що потребують навігації в умовах наявності перешкод.

Ключові слова: ультразвуковий сенсор, рухома платформа, перешкоди, вимірювальна система, мікроконтролер, робототехніка.

ABSTRACT

Haidamashko A.O. Information and Measurement System for Obstacle Detection and Avoidance. Master's qualification thesis in specialty 175 – Information and Measurement Technologies, educational program – Computerized Information and Measurement Technologies. – Vinnytsia: VNTU, 2025. – 118 p. In Ukrainian. Bibliography: 22 titles; Tables: 9; Figures: 33.

The master's qualification thesis investigates and develops an information and measurement system designed to detect obstacles in the path of a mobile platform and to perform their avoidance. The first chapter provides an analysis of modern obstacle detection systems, including sensor classification, data processing technologies, and architectures of obstacle avoidance systems. The second chapter presents the system design: the selection of the ultrasonic sensor, motion and steering mechanisms, the development of the general and detailed structure, and the creation of control software. The third chapter describes the practical implementation of the developed system, including the assembly process, component connections, and testing of obstacle detection and avoidance algorithms under real operating conditions. The results of the research can be applied in the development of autonomous mobile robots, machine vision systems, and other robotic complexes requiring navigation in environments with obstacles.

Keywords: ultrasonic sensor, mobile platform, obstacles, measurement system, microcontroller, robotics.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ СУЧАСНИХ СИСТЕМ ВИЗНАЧЕННЯ ПЕРЕШКОД	11
1.1 Класифікація сенсорів, що застосовуються для виявлення перешкод.....	11
1.2 Технології та методи сенсорної обробки даних.....	23
1.3 Сучасні архітектури та підходи до побудови систем уникнення перешкод.....	32
1.4 Висновки до розділу	41
2 РОЗРОБКА ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНОЇ СИСТЕМИ	43
2.1 Розробка загальної структури	43
2.2 Вибір ультразвукового сенсору для системи	47
2.3 Вибір механізмів для забезпечення руху.....	52
2.4 Вибір механізмів для забезпечення повороту	57
2.5 Розробка детальної структури системи.....	62
2.6 Розробка програмного забезпечення.....	65
3 МОДЕЛЮВАННЯ РОБОТИ ДОСЛІДНОГО ЗРАЗКУ	72
3.1 Конструктивно-технічна частина дослідного зразку	72
3.2 Розроблення програмного забезпечення та налагодження	74
3.3 Тестування, калібрування та оцінювання роботи системи.....	78
3.4 Висновки до розділу	81
4 ЕКОНОМІЧНА ЧАСТИНА	82
4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки	83
4.2 Розрахунок витрат на проведення науково-дослідної роботи.....	84
4.3 Оцінювання важливості та наукової значимості науково-дослідної роботи фундаментального чи пошукового характеру.....	95
4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором.....	96
4.5 Висновки до розділу	101

ВИСНОВКИ	103
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	105
Додаток А (обов'язковий) Ілюстративний матеріал.....	108
Додаток Б (обов'язковий) Протокол перевірки навчальної (магістерської) кваліфікаційної роботи.....	113
Додаток В (довідниковий). Лістинг програми	115

ВСТУП

Актуальність

Актуальність теми «Інформаційно – вимірювальна система визначення перешкод та їх оминання» зумовлена кількома ключовими аспектами.

1. Розвиток автономних робототехнічних систем.

Сучасні тенденції у галузі робототехніки спрямовані на створення автономних мобільних платформ, здатних самостійно орієнтуватися у просторі та взаємодіяти з навколишнім середовищем. Для цього необхідні високоточні вимірювальні системи, що забезпечують визначення відстаней до об'єктів і виявлення перешкод у реальному часі. Розробка таких систем має важливе значення для підвищення безпеки, надійності та ефективності роботизованих пристроїв.

2. Зростання ролі сенсорних технологій.

Вимірювальні сенсорні системи відіграють ключову роль у побудові інтелектуальних пристроїв. Зокрема, ультразвукові сенсори є одним із найбільш поширених засобів для виявлення перешкод завдяки своїй простоті, надійності, енергоефективності та відносно низькій вартості. Використання таких сенсорів дозволяє реалізувати точне вимірювання відстаней навіть у складних умовах освітлення чи запиленості.

3. Технології навігації та безпеки.

Розвиток систем навігації, автоматичного керування транспортними засобами, дронів та роботів потребує ефективних алгоритмів уникнення перешкод. Створення інформаційно-вимірювальних систем цього типу є важливим кроком у напрямку розвитку автономних систем безпеки, здатних приймати рішення без втручання людини.

4. Застосування у промисловості та побуті.

Системи виявлення перешкод широко використовуються не лише в робототехніці, а й у промисловій автоматизації, транспортних засобах, розумних будинках, системах моніторингу та охорони. Від точності вимірювань та

швидкості реакції залежить ефективність таких систем, тому актуальним є створення оптимізованих рішень з високою стабільністю роботи.

5. Інновації у сфері мікроконтролерних технологій.

Сучасні мікроконтролери забезпечують можливість створення компактних, енергоефективних і програмно гнучких систем, що дозволяє інтегрувати вимірювальні модулі, сенсори, виконавчі механізми та алгоритми обробки сигналів в єдину систему. Це відкриває перспективи для реалізації інтелектуальних функцій оминання перешкод та автономного прийняття рішень.

Аналіз останніх досліджень.

1. Сенсорні технології та методи вимірювання відстані.

У сучасних дослідженнях активно розвиваються технології визначення перешкод із використанням ультразвукових, інфрачервоних, лазерних (LiDAR) та стереокамерних систем. Особливу увагу приділено алгоритмам фільтрації шумів і підвищення точності визначення відстані в динамічних умовах.

2. Обробка сенсорних даних та алгоритми уникнення перешкод.

Наукові праці спрямовані на вдосконалення методів інтеграції даних з кількох сенсорів (sensor fusion) та застосування алгоритмів машинного навчання для прогнозування траєкторії руху об'єктів і вибору оптимального шляху.

3. Використання мікроконтролерів і вбудованих систем.

Мікроконтролери (наприклад, серії Arduino, STM32, Raspberry Pi) широко застосовуються для керування рухомими платформами. Вони забезпечують гнучке налаштування системи, низьке енергоспоживання та можливість обробки даних у реальному часі.

4. Системи штучного зору.

Паралельно розвиваються системи машинного зору, що дозволяють визначати перешкоди за допомогою аналізу відеозображень. Проте, такі системи є складнішими та потребують значно більших обчислювальних ресурсів, ніж ультразвукові вимірювальні комплекси.

5. Інтеграція в робототехнічні платформи.

Сучасні дослідження демонструють тенденцію до поєднання вимірювальних систем із автономними роботами, безпілотними транспортними засобами, а також пристроями Інтернету речей (IoT), що дозволяє створювати мережеві інтелектуальні комплекси навігації.

Мета роботи

Метою роботи є розробка інформаційно-вимірювальної системи для визначення перешкод та реалізація алгоритмів їх оминання на основі мікроконтролерних технологій.

Об'єктом дослідження є процеси виявлення та вимірювання відстані до перешкод за допомогою ультразвукових сенсорів у складі рухомих платформ.

Предметом дослідження є методи побудови та програмна реалізація інформаційно-вимірювальної системи, яка забезпечує виявлення перешкод, обробку сенсорних даних і формування керуючих сигналів для оминання об'єктів.

Наукова новизна одержаних результатів – розроблено структурно-програмний підхід до побудови системи визначення перешкод та їх оминання, що поєднує вимірювальні й керуючі функції на основі ультразвукового сенсора та мікроконтролера. Запропоновано алгоритм обробки даних у реальному часі, який дозволяє підвищити точність визначення відстаней і швидкість реакції системи.

1 АНАЛІЗ СУЧАСНИХ СИСТЕМ ВИЗНАЧЕННЯ ПЕРЕШКОД

1.1 Класифікація сенсорів, що застосовуються для виявлення перешкод

У наш час ми спостерігаємо стрімке зростання у сфері автономних мобільних роботів (АМР). Завдяки їхній високій працездатності, АМР поступово витісняють людей у низці внутрішніх середовищ. Мобільні є дуже важливими у зниженні робочого навантаження на людей та підвищенні загальної ефективності виробництва. Вони знайшли широке застосування в багатьох галузях, включаючи складське господарство, логістику, охорону здоров'я, ресторанний сервіс, а також послуги для приватних осіб [1].

Базові принципи функціонування мобільних роботів, рухомих платформ, охоплюють такі напрямки, як рух, сприйняття оточення, аналіз даних та орієнтування у просторі.

Питання руху зосереджується переважно на системі переміщення робота. Дизайн цієї системи залежить від вимог до майбутньої роботи, таких як характер робочого простору, легкість управління, енергоефективність, стійкість та інші важливі характеристики. Сприйняття полягає в отриманні інформації як про зовнішнє робоче середовище, так і про власний стан мобільного робота. Аналіз даних включає обробку та оцінку інформації, зібраної системою сприйняття, після чого надаються команди для рухової системи, що забезпечує виконання завдань. Навігація мобільного робота вимагає координації сприйняття, аналізу даних та контролю руху для успішного переміщення з початкової позиції до кінцевої мети у робочому просторі, незалежно від його знання. Сприйняття є центральним елементом досліджень мобільної робототехніки. Щоб АМР могли безпечно та продуктивно виконувати свої функції, вони повинні адекватно відчувати своє оточення і на основі цього формувати свої рішення. Якщо мобільний робот не здатний точно й швидко сприймати навколишній світ, він не зможе впоратися навіть із простими операціями.

Система сприйняття мобільного робота використовує різноманітні сенсорні технології для збору даних про навколишнє середовище, сам пристрій та їхній взаємозв'язок. Ця інформація згодом використовується для розробки маршруту, керування рухом і, зрештою, для успішного завершення навігаційного процесу. Основними завданнями системи сприйняття є визначення положення, створення мапи та ідентифікація об'єктів [2].

Визначення положення — це процес оцінки поточної позиції мобільного робота відносно його робочої зони під час руху. Ця інформація є фундаментальною основою для всієї навігації робота. Методи локалізації мобільних роботів включають інерціальну навігацію, глобальні системи позиціонування (GPS), орієнтування за активними маяками (рис. 1.1), навігацію за виділеними об'єктами (віхами) та зіставлення з моделями. Інерціальна навігація базується на використанні даних від сенсорів, що реєструють інерційні сили. Глобальна система позиціонування (GPS) є супутниковою технологією, яка надає точні координати (довготу, широту, висоту) для будь-якої точки на Землі. Це високоефективний спосіб визначення місця розташування роботів на відкритому повітрі.

Однак, використання GPS суттєво обмежене для локалізації в приміщеннях через проблеми з відсутністю прямої видимості та блокуванням сигналу у складному середовищі. Навігація за активними маяками та об'єктами використовує відносні координати маяків або відомих орієнтирів у середовищі для визначення позиції.

Зіставлення з моделями — метод позиціонування, при якому дані від сенсорів порівнюються з конкретною картою-моделлю. Наприклад, у 2D LiDAR локалізації, вимірювання LiDAR надають двовимірну хмару лазерних точок, а карта-модель представлена сіткою зайнятих/вільних просторів. Позиціонування робота виконується шляхом узгодження хмари точок із цією сітковою картою.

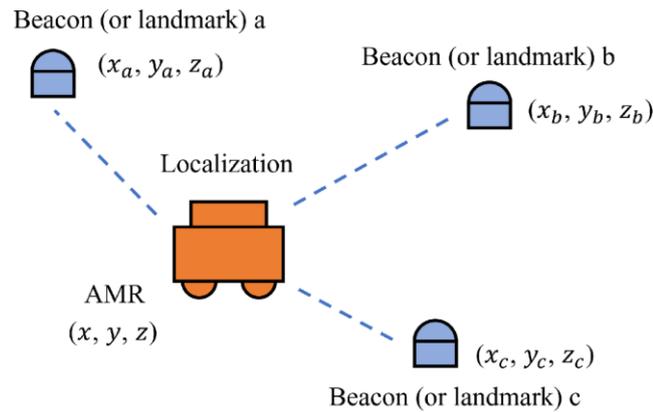


Рисунок 1.1 - Схематична діаграма методу на основі активного маяка

Створення карт найчастіше реалізується за допомогою методології Simultaneous localization and mapping (SLAM), що є наріжним каменем у дослідженнях мобільних роботів. SLAM — це технологічний підхід, який дозволяє автономному пристрою визначати своє місцезнаходження у невідомій зоні і паралельно з цим будувати точне графічне представлення цього середовища (рис. 1.2). Компонент локалізації відповідає за оцінку позиції робота на вже наявній карті, тоді як компонент картографування займається формуванням самої карти оточення. Ці дві складові нерозривно пов'язані і працюють разом для завершення процесу створення карти, постійно підвищуючи точність з кожною ітерацією. Найпоширенішими сенсорами, які використовуються в SLAM, є LiDAR (лазерний далекомір) та відеокамери [2].

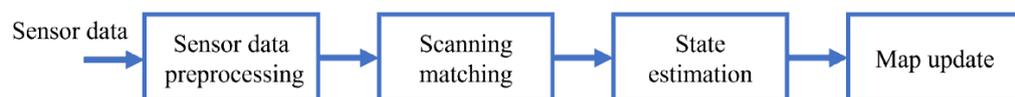


Рисунок 1.2 - Схематична діаграма методу SLAM на основі фільтра

Методики SLAM можуть бути поділені на дві основні групи, виходячи з принципів оптимізації протягом короткого або тривалого часу: фільтраційні методи та методи на основі графової оптимізації (рис. 1.3).

Фільтраційні алгоритми SLAM зазвичай використовують різні типи математичних фільтрів для оцінки та корегування траєкторії руху робота і створеної ним карти. Двома найбільш вживаними фільтраційними алгоритмами є SLAM на основі фільтра Калмана та SLAM на основі частинкового фільтра.

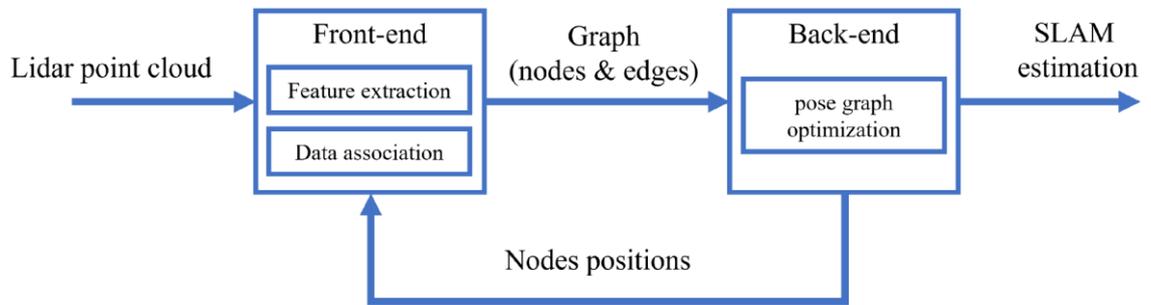


Рисунок 1.3 - Схематична діаграма процесу оптимізації графа SLAM

Алгоритм графової оптимізації переважно включає дві функціональні частини: інтерфейс та обробник. Цей метод моделює шлях переміщення робота та просторову структуру його оточення у вигляді графа. Для визначення позиції робота та побудови карти середовища, алгоритм мінімізує похибку, шляхом оптимізації вузлів і ребер графа [5].

Зазвичай вузли графа представляють змінні, які потребують оптимізації у процесі SLAM, такі як місцезнаходження сенсора, позиції виявлених орієнтирів або оцінена траєкторія руху. Ребра графа позначають обмеження або різницю між цими змінними. Процес оптимізації виконується ітераційно для зменшення похибки ребер, що зрештою призводить до знаходження найбільш точного рішення.

Завдяки прогресу у сфері штучного інтелекту, мобільні роботи поступово звільняються від необхідності мати попередньо побудовані карти. Це дозволяє їм приймати рішення щодо пересування, базуючись лише на візуальній інформації. Основний механізм полягає у встановленні системи заохочень або-ж винагород, для різних етапів дослідження. Наприклад, винагорода за досягнення цілі, штраф за зіткнення, або винагорода за прояв «цікавості». Далі, використовуючи різні методи навчання, створюється відповідна нейронна

мережа, яка тренується використовуючи набори даних, які додає користувач. Результатом є здатність автономного прийняття навігаційних рішень мобільним роботом [5].

Для того, щоб ефективно і точно виконати навігаційне завдання, робот зобов'язаний уникати зіткнень із перешкодами, що є життєво важливою функцією для автономних мобільних систем, що рухаються в непередбачуваних умовах та на різних локаціях. На відміну від завдань, що вимагають знання глобальної позиції робота, для уникнення перешкод достатньо отримувати локалізовану інформацію про об'єкти, розташовані безпосередньо поблизу самого робота. Виявлення перешкод також є критично важливим для системи сприйняття, то когнітивний блок мобільного робота, спираючись на дані про відстань і напрямок перешкод, отримані від системи сприйняття, планує оптимальний шлях для їх обходу. В основному, дальномірні сенсори (такі як ультразвукові сенсори та LiDAR) ідеально підходять для завдань ідентифікації перешкод, а також, деякі відеокамери можуть не тільки розпізнавати статичні перешкоди, але й потенційно прогнозувати траєкторію руху динамічних об'єктів.

Система сприйняття виконує ці функції за допомогою різноманітних сенсорів та вилучення необхідної інформації з їхніх вимірювань. Сенсори, що використовуються для збору даних, поділяються на дві основні категорії: пропріоцептивні/екстероцептивні та активні/пасивні. Пропріоцептивні сенсори збирають внутрішні дані про стан робота, такі як швидкість двигунів або кут повороту коліс. До них зараз найчастіше відносять енкодери, акселерометри та гіроскопи. Екстероцептивні сенсори збирають зовнішню інформацію про оточення мобільного робота, наприклад, зображення, звукові хвилі чи радіосигнали. Прикладами екстероцептивних сенсорів є відеосенсори, ультразвукові сенсори та LiDAR. Активні сенсори випромінюють енергію у зовнішнє середовище, а потім вимірюють відбиту відповідь (наприклад, сонар або LiDAR). Пасивні сенсори реєструють енергію, яка надходить до них із навколишнього середовища, як-от відеосенсори (на основі технологій CCD або CMOS).

Інерційний вимірювальний блок (IMU) є ключовим електронним компонентом, який об'єднує декілька сенсорів в одному корпусі: акселерометр для фіксації лінійного прискорення, гіроскоп для вимірювання кутової швидкості та магнітометр для визначення орієнтації відносно магнітного поля Землі. За своєю суттю, IMU класифікується як пропріоцептивний сенсор, оскільки його функція полягає у зборі внутрішніх даних про фізичний стан та рух самого об'єкта, до якого він прикріплений [5].

На базі IMU функціонує Інерційна навігаційна система (INS). Це система автономного обчислення координат (рис. 1.4), яка широко використовується для орієнтування автономних мобільних роботів. Хоча магнітометр теоретично надає дані про орієнтацію, його практичне використання для високоточних навігаційних завдань часто ускладнене: параметри, виміряні магнітометром, характеризуються низькою точністю, оскільки легко спотворюються через наявність металевих конструкцій, електронного обладнання та інших джерел електромагнітних перешкод у приміщеннях. З цієї причини, найбільш надійними вихідними параметрами, та тими що найчастіше використовують, від IMU, є прискорення та кутова швидкість [6].

Метод автономного обчислення координат є основоположним для INS. Використовуючи отримані від IMU дані, траєкторія руху внутрішнього АМР розраховується шляхом послідовного інтегрування (дворазового для прискорення та одноразового для кутової швидкості) цих вимірянних значень. Для успішного виконання цих обчислень абсолютно необхідно мати точні початкові умови: відому початкову позицію, початковий напрямок та початкову швидкість робота.

Таким чином, INS дозволяє роботу самостійно відстежувати своє переміщення, накопичуючи дані про зміну швидкості та орієнтації з плином часу.

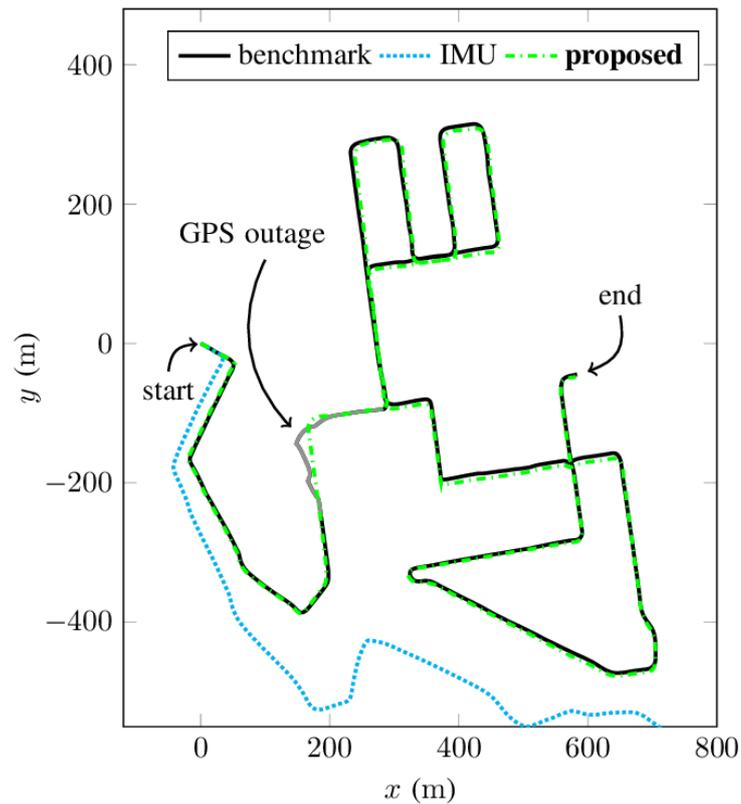


Рисунок 1.4 - Навігація за допомогою інерційного вимірювального блоку (IMU)

Незважаючи на свої переваги, IMU має певні суттєві обмеження при застосуванні на мобільних роботах:

1. Залежність від початкових умов, через яку, процес розрахунку навігаційних параметрів завжди вимагає точних початкових умов (координат, орієнтації та швидкості). Будь-яка неточність у цих початкових значеннях безпосередньо впливає на точність усієї подальшої навігації.
2. Через накопичення помилок з часом, проблема навігації зростає і накопичується з плином часу. Це обумовлено тим, що для визначення положення потрібно двічі інтегрувати вимірювання прискорення, що призводить до швидкого накопичення навіть невеликих помилок вимірювання.

Варто відзначити, що тривалість роботи та точність сенсорів ІМУ мають безпосередній і значний вплив на загальну точність навігаційної системи, заснованої на ІМУ [7].

Для ефективного подолання цих недоліків ІМУ дуже часто використовується не самостійно, а в комбінації з іншими сенсорами. Також він використовується як допоміжний вимірювальний пристрій в інших навігаційних методах [7].

Наступним варто розглянути ультразвуковий сенсор, або ж сонар. Ультразвук являє собою звукові хвилі, частота коливань яких перевищує 20 кГц, що знаходиться за межами верхньої межі сприйняття людського слуху. Технологія сонара також використовує цей діапазон частот. У мобільній робототехніці ультразвукові сенсори зазвичай застосовуються як сенсори для вимірювання відстані.

Сенсори, призначені для вимірювання дистанції, можна класифікувати за двома основними підходами до вимірювання: метод вимірювання за типом відбиття (ехолокаційний) та метод односпрямованого вимірювання (рис. 1.5).

1. Ехолокаційний метод заключає у собі єдиний пристрій (або інтегрований модуль) що випромінює акустичний сигнал у простір. Коли ця хвиля стикається з фізичною перешкодою, вона відбивається і повертається назад до сенсора, який тепер виконує функцію приймача. Шляхом вимірювання часу між випромінюванням та прийомом відбитого сигналу, сенсор може точно обчислити дистанцію до перешкоди.
2. Односпрямований метод, вимагає просторового розділення компонентів: передавач і приймач розміщуються у різних точках. Приймач реєструє сигнал, надісланий передавачем, і, ґрунтуючись на цьому прийомі, розраховується відстань між самими передавачем і приймачем.

Ультразвукове визначення відстані базується на трьох головних критеріях, які використовуються для розрахунку дистанції: час поширення сигналу, зсув фази та аналіз акустичної вібрації. Кожен із цих критеріїв використовується для перетворення фізичного вимірювання на точну відстань до об'єкта.

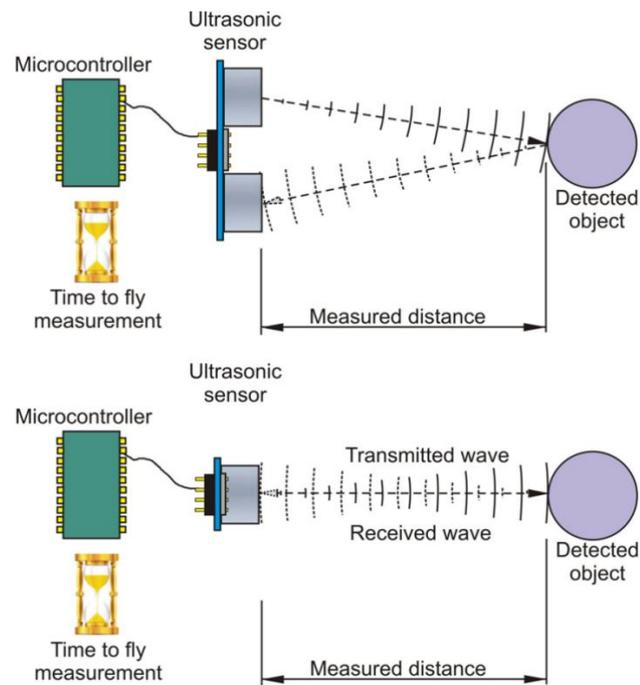


Рисунок 1.5 - Методи вимірювання відстані сенсором: ехолокаційний метод та односпрямований метод

Ультразвукові сенсори здебільшого використовують ехолокаційний метод для розрахунку дистанції до об'єкта. Це робить їх корисними у мобільній робототехніці для початкового визначення положення та елементарної навігації. Однак, сонари мають низку суттєвих недоліків, які значно обмежують їхню ефективність у складних навігаційних завданнях. Однією з головних проблем є надмірна ширина ультразвукового променя. Через це промінь є занадто розфокусованим, що значно ускладнює точне визначення напрямку розташування об'єкта, призводячи до нечіткості при ідентифікації кутових координат перешкод [7]. Крім того відбиваючись від гладких або похилих поверхонь об'єктів, часто виникає явище дзеркального відбиття. Таке відбиття може спрямувати акустичну хвилю убік від приймача, що часто призводить до некоректних розрахунків відстані або навіть повної втрати сигналу, що знижує загальну надійність вимірювань дистанції від сонарів. Як наслідок, коли частота помилкових або неточних вимірювань відстані стає надто високою, стає неможливим побудувати цілісну та достовірну сіткову карту навколишнього середовища, використовуючи лише дані, отримані від ультразвукових сенсорів.

Це робить ультразвукові сенсори непридатними як єдине джерело інформації для надто складного картографування [7].

Сенсори, що працюють на інфрачервоному випромінюванні, використовують випромінювач для генерації інфрачервоного променя та приймач для фіксації та вимірювання інтенсивності цього променя (рис. 1.6). Принципово важливим є те, що інтенсивність інфрачервоного випромінювання поступово зменшується пропорційно до збільшення пройденої відстані. Завдяки цій властивості, інфрачервоні сенсори можуть бути використані для вимірювання дистанції шляхом оцінки залишкової інтенсивності отриманого променя. Крім того, вони можуть застосовувати метод часу польоту, який ґрунтується на вимірюванні часу, необхідного для проходження світлом певної відстані. Подібно до ультразвукових сенсорів, інфрачервоні сенсори також використовують як ехолокаційний метод, так і метод односпрямованого вимірювання відстані.

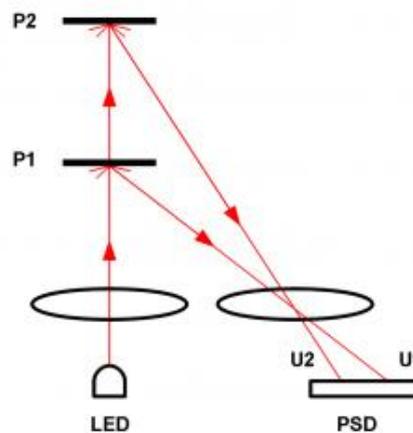


Рисунок 1.6 - Маршрут світлового променя від інфрачервоного сенсора

LiDAR (Light Detection and Ranging) — вищезгаданий сенсор дистанційного зондування, який використовує лазерні промені для визначення точного розташування та відстані до об'єктів-мішеней. Принцип його дії ґрунтується на випромінюванні множинних лазерних імпульсів із передавачів, коли ці імпульси досягають об'єктів у навколишньому середовищі, вони відбиваються і повертаються назад, де їх фіксують спеціалізовані приймачі.

Дистанція, пройдена кожним окремим променем, вимірюється за допомогою методу часу польоту, після чого, шляхом комбінування та обробки цієї інформації про відстань, отриманої від усіх лазерних імпульсів, система створює детальне тривимірне або двовимірне зображення навколишнього простору. За своєю функціональністю LiDAR має схожість із радаром, але вирізняється значно вищою точністю, більшою швидкістю вимірювання та розширеним діапазоном дистанцій порівняно з традиційними радіолокаційними системами [7].

У сфері навігації мобільних роботів усередині приміщень переважно використовується 2D-сканування LiDAR. Такий тип сенсорів є достатнім для збору необхідної інформації про площинне середовище та побудови двовимірних карт. На противагу цьому, 3D-LiDAR значно частіше застосовується в галузі інтелектуального водіння та автономних транспортних засобів. Це пояснюється тим, що системи автономного керування потребують набагато більшого обсягу просторової інформації для аналізу та безпечної обробки складних дорожніх сценаріїв та ситуацій.

Візуальні сенсори, основними різновидами яких є монокулярні камери, бінокулярні (стерео) камери, RGBD-камери та камери подій, це пристрої, розроблені для імітації ключових функцій зорової системи людини здатні захоплювати, обробляти та інтерпретувати інформацію видимого світла з навколишнього простору, перетворюючи її на дискретні цифрові сигнали або інші форми виведення. Ці дані потім використовуються для глибокого аналізу, подальшої обробки та прийняття рішень роботами чи іншими інтелектуальними системами. На відміну від більшості інших сенсорів, візуальні системи мають унікальну перевагу у можливості представляти зовнішнього середовища у форматі повноцінного зображення. Ця особливість дозволяє їм детально фіксувати та описувати складні текстурні та візуальні характеристики оточення. До того ж, відносно низька собівартість виробництва та інтеграції робить сенсори зору поширеними у сфері мобільної робототехніки [18].

Монокулярна камера функціонує на принципах, які є спільними для звичайних фотоапаратів. Її процес формування зображення зазвичай описується через спрощену модель обскури. У цій моделі, світло проходить крізь невеликий отвір і проєктується на світлочутливу площину, яка найчастіше представлена CMOS або CCD-сенсором. Отримане проєкційне зображення потім проходить етапи корекції геометричних спотворень та піксельного вирівнювання, щоб сформувати двовимірну картинку, візуально подібну до тієї, що сприймається людським оком. Попри економічність, монокулярні системи стикаються із значними дослідницькими труднощами, оскільки безпосереднє обчислення інформації про глибину є надзвичайно складним завданням. Стереокамера являє собою сенсорний комплекс, що складається з двох монокулярних камер, конфігурація яких повторює структуру зорової системи людини. Глибина до об'єкта може бути визначена шляхом аналізу базової відстані між оптичними центрами цих двох камер. Оцінка фактичної глибини z для будь-якої точки в кадрі досягається за допомогою фундаментального геометричного співвідношення, що описується формулою:

$$z = \frac{fb}{d}, \quad (1.1)$$

де z є оцінюваною глибиною, f — фокусна відстань, b — фактична базова відстань між камерами, а d — це паралакс (зсув пікселів) між проєкціями однієї і тієї ж точки на лівому та правому зображеннях. З цієї формули можна зробити важливий висновок: через пряму залежність від базової відстані b , якщо вимірювана глибина z стає надто великою, виникає значна помилка усікання, що призводить до недостатньої точності оцінки глибини на великих дистанціях [7].

RGBD-камери є результатом технологічної інтеграції, вони поєднують сенсор глибини з традиційним монокулярним сенсором. Сенсор глибини активно виконує вимірювання відстані для кожного окремого пікселя зображення. В наш час основними технологіями, які використовуються для реалізації цих сенсорів в

RGBD-камерах, є інфрачервоні сенсори структурованого світла та сенсори часу польоту (ToF). Ці дві технології принципово відрізняються своїм підходом до оцінки глибини тим, що інфрачервоний сенсор структурованого світла розраховує глибинну інформацію, аналізуючи спотворення попередньо спроектованого, відомого зображення на поверхнях об'єктів. З іншого боку, сенсор часу польоту (ToF) обчислює глибину кожного пікселя шляхом випромінювання імпульсів світла і точного вимірювання часу, необхідного для того, щоб це світло повернулося до приймача. Отримана карта глибини потім точно накладається на RGB-зображення шляхом використання відомих параметрів відносного просторового розташування між обома сенсорами, що дозволяє отримати композитне зображення, яке одночасно містить дані про колір і відстань. Однак, через неминучі фізичні обмеження, пов'язані з технологією сенсорів глибини, достовірний пошук інформації про глибину є неможливим у великомасштабних та просторових середовищах із великою дальністю вимірювання [7].

1.2 Технології та методи сенсорної обробки даних

Основні завдання цифрових сенсорних мереж є триєдиними. По-перше, вони здійснюють вимірювання ключової інформації із зовнішнього фізичного середовища, по-друге, вони проводять вибірку сигналів із внутрішніх компонентів системи; і, по-третє, вони формують значущі дані із сирих вимірювань для підтримки процесів прийняття рішень. Ці мережі є фундаментальною частиною єдиної архітектури, яка охоплює як базові сенсорні системи, так і інтелектуальні прикладні сервіси.

Прогрес у цій сфері відкриває шлях до створення передових, взаємопов'язаних додатків, таких як "розумні" будівлі, інтелектуальні системи охорони здоров'я, розумні транспортні мережі та системи ефективного споживання енергії. Очікується, що світовий ринок таких технологій зростатиме завдяки розвитку бездротових комунікаційних технологій та впровадженню

нових обчислювальних парадигм, зокрема хмарних платформ. За прогнозами Gartner, до 2020 року цей глобальний ринок мав охопити 5,8 мільярда застосувань, що підкреслює стрімке зростання попиту на підключені пристрої та відповідні сервіси [17].

Ці додатки, що використовують можливості глобальної мережі пристроїв, значною мірою залежать від бездротової сенсорної мережі. Ці бездротові сенсори, як правило, розташовані довільно і здатні створювати динамічні ad hoc мережі без необхідності попереднього розгортання сталої інфраструктури. Бездротової сенсорної мережі підтримується за допомогою низьковартісних та енергоефективних технологій, таких як Wi-Fi, Bluetooth, Zigbee та Near Frequency Communication.

У наступні роки в епоху Глобальної мережі пристроїв, здатність системи розуміти контекст виступає як критично важливий зв'язок між фізичним світом та віртуальними обчислювальними сутностями.

Це розуміння вимагає аналізу даних, мережевої комунікації та зондування середовища. Однак, бездротові сенсорні мережі стикаються зі значними операційними труднощами, зокрема інтерференцією, втратою пакетів, надмірністю та різномірністю генерування даних. З огляду на це, сирі дані від цифрових сенсорів зазвичай містять велику кількість неочищеної, некорисної та забрудненої інформації

Такий необроблений сигнал, що демонструє небажані спотворення та зміни, якщо його залишити без корекції, призводить до надмірного споживання обчислювальних ресурсів та високих вимог до потужності. Отже, обробка вихідного сенсорного сигналу є абсолютно необхідною. Наявність значної кількості зайвих даних призводить до високих обчислювальних витрат та перевантаження ресурсів у мережах з обмеженими можливостями [17].

Існує широкий спектр існуючих рішень, спрямованих на обробку сирих даних від сенсорів. Найпоширеніші техніки обробки включають виявлення та фільтрацію шумів, заповнення пропущених значень, ідентифікацію аномальних

викидів та агрегування інформації. Лише після того, як сирі дані пройдуть процес очищення, можна виконати аналіз для отримання релевантної інформації.

Машинне та глибоке навчання пропонують надзвичайно перспективні підходи для ефективного аналізу даних, зібраних сенсорами. Застосування цих аналітичних технік дозволяє отримати глибоке розуміння суті даних, виявити приховані закономірності та підтримати подальше прийняття рішень. У цьому контексті, дана робота детально описує різні існуючі підходи до аналізу даних.

Дані, що надходять від сенсорів, мають складні властивості, відомі як "3V" - величезний обсяг (voluminousness), достовірність (veracity) та швидкість (velocity). Це вимагає ефективного зберігання для подальшого аналізу та досягнення цілей застосувань, що базуються на сенсорних мережах [17].

Інтеграція сенсорних систем із новими обчислювальними парадигмами забезпечує ефективні методи для керування динамічним та комплексним характером сенсорної інформації. У сенсорній мережі вузли часто є розподіленими, і багато з них можуть виконувати однакові операції. З цієї причини інтеграція та злиття даних із множини сенсорів є обов'язковим для підвищення загальної точності в різноманітних прикладних сервісах. Як приклад можна навести систему моніторингу трафіку в реальному часі, де для злиття даних використовується заповнення пропущених даних. Наприклад, дані, зібрані минулого тижня, можуть бути інтегровані з інформацією, отриманою після моменту втрати даних. Цей процес може охоплювати вимірювання швидкості транспортних засобів, автоматичне розпізнавання номерних знаків та інші аспекти.

У сучасну епоху технологічного розвитку, головною метою Глобальної мережі пристроїв є підвищення якості людського життя через встановлення зв'язку між фізичними об'єктами та віртуальними обчисленнями. Це досягається завдяки створенню передових прикладних програм, постійному вдосконаленню сенсорних технологій, комунікаційних мереж та методів обробки інформації. Серед прикладів таких високотехнологічних застосувань можна виділити інтелектуальні системи "розумного міста", просунуті транспортні мережі,

електронну охорону здоров'я, автоматизовані системи керування будівлями та розумні енергетичні мережі. Для забезпечення стрімкого прогресу в цій сфері, сенсорні системи відіграють критично важливу роль, збираючи ключові дані як із внутрішніх функціональних механізмів пристроїв, так і з зовнішнього середовища [17].

З цієї причини бездротові сенсорні мережі набули значної популярності, оскільки їх можна розгорнути самостійно без попередньої вимоги до встановлення стаціонарної інфраструктури. Ці бездротові сенсорні системи мають здатність до самоорганізації та можуть бути розміщені у просторі з високою довільністю.

Базова архітектура складається з шарів обробки, злиття та аналізу даних від сенсорів взаємопов'язаних пристроїв, де шар сенсорних даних переважно формується різноманітними цифровими сенсорами, здатними вимірювати фізичні параметри оточення та фіксувати зміни середовища у реальному часі (рис. 1.7). До числа часто використовуваних сенсорів належать сенсори температури, тиску, вологості, рівня, акселерометри, газові сенсори, гіроскопи, сенсори руху, оптичні та візуальні сенсори, сенсори радіочастотної ідентифікації (RFID) та інфрачервоні (ІЧ) сенсори. Кожен сенсорний пристрій зазвичай комплектується мікропроцесорним блоком, модулем зберігання даних, керуючим блоком, системою живлення та інтерфейсами бездротового зв'язку. Важливо, що ці сенсорні пристрої мають обмеження щодо фізичних розмірів, обчислювальної потужності, об'єму пам'яті, мережевих можливостей та простору для зберігання даних. Для забезпечення комунікації між цими сенсорними пристроями найчастіше застосовуються такі протоколи бездротового зв'язку, як Wi-Fi, ZigBee, Bluetooth, ближній безконтактний зв'язок (NFC) та мобільні технології стандарту LTE/4G [18].

Більшість інформації, отриманої від цифрових сенсорів, вимагає обробки в режимі реального часу для підтримки промислових операцій, медичних застосувань та наукових досліджень.

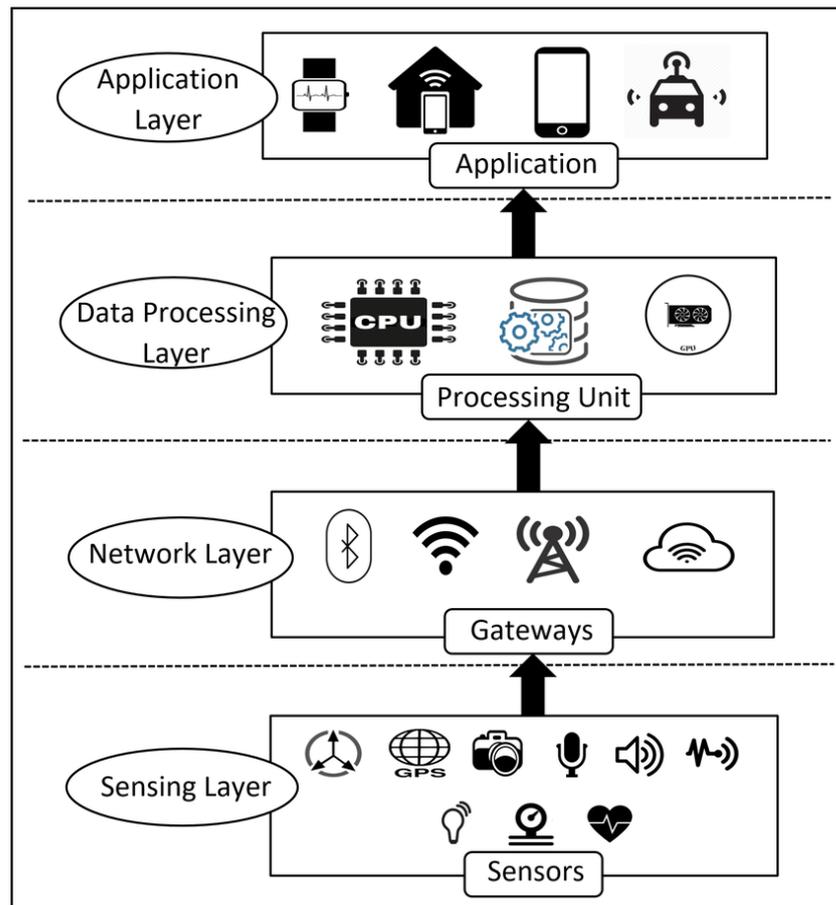


Рисунок 1.7 – Схематичне зображення шарів базової архітектури

Наприклад, біометричні сенсори, що використовуються для моніторингу життєво важливих показників пацієнтів, можуть генерувати надзвичайно великі обсяги даних. Ці виміряні дані є критично важливими і повинні бути ретельно очищені для усунення будь-якої невизначеності перед подальшим аналізом. Це дозволяє формувати знання та забезпечувати прийняття обґрунтованих рішень. Отже, рівень обробки даних націлений на виконання різних функцій, включаючи фільтрацію шуму, виявлення аномальних значень, заповнення пропущених даних та агрегування інформації [18].

Рівень злиття даних є необхідним для ефективної роботи зі складними викликами, пов'язаними з інформацією, що генерується множиною різноманітних сенсорних пристроїв. Основна мета цього рівня полягає в інтеграції достовірних вимірювань, отриманих від різних джерел. Комбіновані дані з цих різних пристроїв потім передаються на рівень аналізу даних для ефективного формування знань та підтримки процесів прийняття рішень.

Основний підхід до злиття передбачає пряму інтеграцію сирих вимірювань із різних сенсорних пристроїв. Більш вдосконалений метод включає початкове вилучення ключових ознак, після чого відбувається сам процес злиття. Найбільш розширений метод включає вилучення ознак, за яким слідує декларація ідентичності (визначення джерела), і лише потім виконується злиття даних. Цей багатоетапний підхід дозволяє досягти вищого рівня висновків знань та значно точнішого прийняття рішень.

Протягом останніх років впровадження сучасних обчислювальних парадигм — таких як хмарні обчислення, та периферійні обчислення, здійснило революцію у сфері аналізу сенсорних даних. Ці технології забезпечують надійну, загальнопоширену та зручну платформу для ефективної роботи з динамічними та різномірними варіантами сенсорної інформації.

Таким чином, рівень аналітики даних спрямований на розробку інтелектуальних функціональних можливостей для широкого спектра прикладних систем, де ключові цілі впровадження цих розподілених платформ включають зниження витрат на обчислення та зберігання, підвищення надійності передачі даних у мережі, скорочення затримок у мережі, посилення безпеки та конфіденційності, забезпечення масштабованості, а також гарантування безвідмовних та безпечних рішень [19].

Вимірювальні пристрої генерують інформацію або послідовно, або у відповідь на зовнішній тригер, але також існує інший важливий аспект процесу, який полягає в тому, що дані, отримані від сенсорних вузлів, повинні бути зібрані, агреговані, проаналізовані та візуалізовані для вилучення корисної інформації. Ця кінцева інформація потім інтерпретується для створення репрезентативної та зрозумілої форми, яка може бути доставлена кінцевому користувачеві, а також для формування реакції на зовнішній подразник або подію. Окрім інформації, створеної бездротовими сенсорними мережами, потоки даних надходять і з інших незалежних джерел. Відтак, виникає потреба в безпрецедентному обсязі агрегації та збереження цих даних, а також у їхній постійній потоковій передачі із заданою мережевою швидкістю до віддалених

сховищ для виконання історичного аналізу. Однак, цей процес супроводжується низкою характеристик та проблем, притаманних сенсорним даним.

Сенсорна інформація демонструє високу складність через такі чинники, як надзвичайний обсяг, динамічність зміни даних, необхідність оновлення в реальному часі, швидке старіння критичних даних та взаємозалежність між даними, що надходять із різних джерел. Зазвичай, сенсорні елементи інтегруються безпосередньо у людське тіло, фізичні об'єкти чи певні локації. Таким чином, найбільш значущі характеристики даних, отриманих від сенсорів взаємопов'язаних систем, є наступними:

1. Технічні обмеження - компактний розмір самого сенсорного пристрою призводить до суттєвих технічних обмежень у плані обчислювальної потужності, ємності батареї, мережевих можливостей, обсягу пам'яті та простору для зберігання даних. Внаслідок цих обмежень, такі сенсори стають вразливими до відмов, атак та легкого виходу з ладу, що, у свою чергу, спричиняє втрату сенсорних даних та отримання неточних висновків.

2. Обробка в режимі реального часу - сучасна сенсорна мережа повинна бути здатна виконувати складні мережеві завдання та забезпечувати перетворення вихідних вимірювань на більш цінну та аналітичну інформацію саме в режимі реального часу.

3. Масштабованість - у фізичному світі сенсорна мережа може включати джерела даних від численної кількості сенсорів та виконавчих механізмів. Сенсорні системи повинні мати високу масштабованість, щоб ефективно адаптуватися до експоненційного зростання кількості підключених пристроїв, належним чином обробляти збільшення обсягів даних та відповідати різноманітним цілям прикладних рішень, що базуються на взаємопов'язаних системах.

4. Формат представлення даних - типовий формат сенсорних даних є невеликим за розміром кортежем із чіткою структурованою інформацією. Різні представлення сенсорної інформації включають булеві значення, бінарні коди, фіксовані ознаки, безперервні дані та числові величини.

5. Гетерогенність, або-ж різноманітність - дані, отримані від сенсорів взаємопов'язаних систем, за своєю природою є гетерогенними. Вони надходять із багатьох різноманітних джерел, серед яких: жорстко структуровані масиви даних, мережі, що генерують інформацію в реальному часі, вбудовані системи з сенсорами, потоки даних із соціальних мереж, а також інші учасницькі сенсорні мережі [19].

У сучасних цифрових сенсорних мережах для обміну інформацією широко використовуються бездротові комунікаційні протоколи, які часто функціонують у неліцензованих частотних діапазонах, що забезпечує значну гнучкість і масштабованість при розгортанні сенсорних систем. Однак, експлуатація цих протоколів у неліцензованих смугах частотних діапазонів призводить до виникнення неконтрольованої інтерференції, або іншими словами - взаємних перешкод. Сигнали інтерференції можуть спричинити неналежну передачу даних, результатом чого стають дані з шумом, пропущеними значеннями, аномальними викидами та надмірністю. У цьому розділі детально розглядаються різноманітні методи аналізу даних, які застосовуються для вирішення цих проблем, притаманних сенсорній інформації, зокрема: фільтрація шуму, заповнення пропущених значень, виявлення аномальних викидів та агрегування даних [19].

Масивні обсяги сенсорних даних, що безперервно генеруються в глобальних мережах пристроїв, потребують комплексного аналізу, який переважно орієнтований на прийняття рішень у реальному часі. Характеристики цих вимірювань є надзвичайно складними, включаючи високу швидкість надходження, величезний обсяг та динамічну зміну значень і типів. Крім того, ці дані забруднюються в процесі передачі, створюючи численні перешкоди на шляху до отримання необхідного аналітичного результату та прийняття своєчасних рішень.

Шум визначається як некорельований компонент сигналу, який спричиняє небажані зміни та модифікації в оригінальних векторних представленнях вимірюваного сигналу. Наявність шумової складової призводить до непотрібної

обробки та марного використання обчислювальних ресурсів для роботи з непридатними даними. Для ефективного вирішення проблеми оцінювання сигналу існують методи вейвлет-перетворення, які мають високу ефективність у представленні сигналу. Зокрема, вейвлет-трансформація здатна зберігати коефіцієнти оригінального сигналу, одночасно усуваючи шум з вимірювання. Це досягається шляхом порогового відсікання коефіцієнтів, що відповідають шумовим сигналам, отже, ідеальна схема визначення порогу є критично важливою.

Імпутація (заповнення пропущених значень) є критично важливим етапом попередньої обробки даних, необхідним для роботи з неповними масивами інформації (рис. 1.8). Багато сучасних галузей та сфер, як-от інтелектуальні міські системи, електронна охорона здоров'я, глобальні системи позиціонування (GPS) та розумні транспортні мережі, використовують технології взаємопов'язаних цифрових систем як ключовий елемент, що генерує величезні обсяги даних [25]. Навчальні алгоритми, які застосовуються для аналізу цих масивів інформації, як правило, за замовчуванням припускають, що дані є повними та цілісними. Однак, оскільки пропущені дані є надзвичайно поширеним явищем у сенсорних мережах, проведення аналітики на неповних масивах інформації може призвести до недостовірних, неточних або нерелевантних результатів. З огляду на це, точна оцінка відсутніх значень є життєво необхідною для цілісності цих цифрових систем. Для ефективного вирішення проблеми пропущених даних необхідно виконати три основні послідовні кроки. Першим етапом є встановлення причини відсутності даних, де типовими причинами неповних результатів є низька якість мережевого з'єднання, несправність сенсорних елементів, вплив зовнішніх факторів середовища та проблеми синхронізації між пристроями.

Пропущені дані поділяються на три основні типи - повністю випадково пропущені (MCAR), пропущені випадково (MAR) та пропущені не випадково (NMAR).

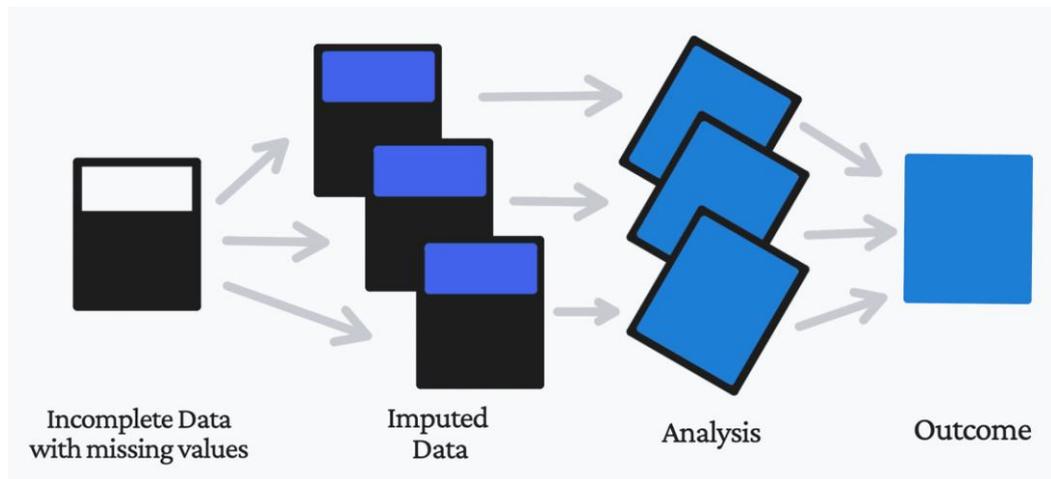


Рисунок 1.8 – Схема роботи методу введення відсутніх даних

Наступний етап передбачає аналіз структури пропусків, на якому існує два основні підходи до класифікації цих закономірностей: монотонні схеми пропусків (MMP) та випадкові схеми пропусків (AMP). Нарешті, останній крок, полягає у створенні моделі заповнення пропущених значень для використання в сенсорних мережах.

Ця модель застосовується для наближеної оцінки відсутніх значень. У науковій літературі представлено низку алгоритмів імпутації, які можна розділити на алгоритми одиначної та багатовимірної імпутації. Важливо зазначити, що традиційні алгоритми імпутації часто не є придатними для роботи зі складними та динамічними даними, які генеруються цифровими сенсорними мережами.

1.3 Сучасні архітектури та підходи до побудови систем уникнення перешкод

Автономні роботи — це машини або пристрої, які володіють здатністю функціонувати незалежно та приймати рішення в межах свого операційного середовища. Для збору інформації про оточення, такої як картографування, навігація та виявлення перешкод, ці роботи оснащені сенсорами та вбудованими обчислювальними системами. У цьому контексті, уникнення перешкод відіграє

критично важливу роль у функціонуванні автономних роботів, забезпечуючи їм можливість ефективної та безпечної навігації. Алгоритми уникнення перешкод допомагають роботам мінімізувати зіткнення, дозволяючи безпечно дістатися до місця призначення та успішно виконати поставлені завдання. Таким чином, здатність уникнення перешкод є незамінним елементом для забезпечення надійної та ефективної роботи будь-якого автономного робота. Основні алгоритми, які розглядаються, детально обговорюються в наступних розділах [3].

Глобальне планування шляху передбачає навігацію робота на основі попередньо існуючих даних про середовище. Ця інформація завантажується в систему планування робота для обчислення повної траєкторії від початкової точки до кінцевого пункту призначення.

Суть цього методу полягає у генерації завершеного маршруту ще до того, як робот розпочне рух, що по суті забезпечує поступову оптимізацію всієї подорожі. Глобальне планування свідомо визначає найкращий можливий шлях для переміщення робота із заданого старту до фінальної мети. У випадку глобального планування маршруту, траєкторія руху робота від початкової локації до цілі вже є відомою, і тільки після цього робот запускається у визначене середовище [3].

На противагу цьому, локальне планування шляху охоплює навігацію робота в динамічних або невідомих умовах, де алгоритм повинен адаптуватися в режимі реального часу до виникаючих перешкод та змін у середовищі (рис. 1.9).

Цей метод переважно фокусується на уникненні перешкод у реальному часі, використовуючи дані, отримані безпосередньо від сенсорів, для забезпечення безпечної навігації. Як правило, робот слідує найкоротшим, прямолінійним шляхом від старту до мети, доки не виявляє перешкоду. Після виявлення він відхиляється від початкового шляху, одночасно оновлюючи критично важливі деталі, такі як нова відстань до цілі та точка обходу перешкоди.

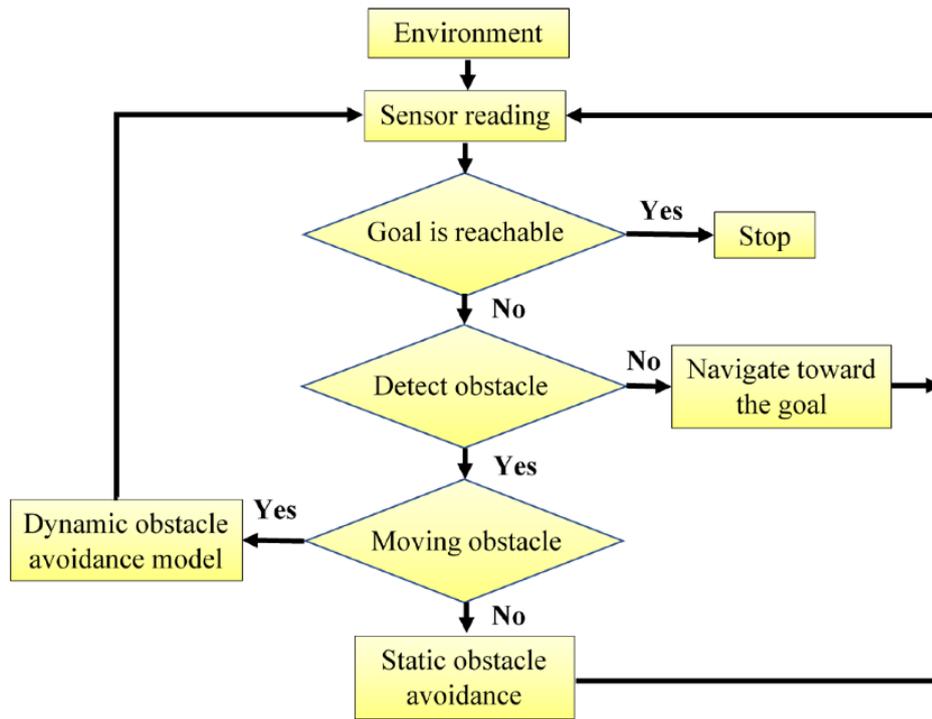


Рисунок 1.9 – Процедура оминання перешкод

Наступним, розглянемо перелік класичних методологій, які використовуються в алгоритміці планування маршрутів:

1. Алгоритм Дейкстри (АД) був вперше представлений нідерландським вченим Едсгером Вібе Дейкстрою у 1956 році, та офіційно опублікований у 1959 році. Цей метод розв'язує задачу знаходження найкоротшого шляху між двома вузлами у зваженому графі з додатними вагами ребер. Він є однією з найбільш поширених технік для побудови маршрутів у ізольованих або статичних робочих просторах. Незважаючи на те, що АД є загальновизнаною та добре вивченою стратегією, його ефективність знижується, коли початкова точка та кінцевий пункт призначення розташовані на великій відстані один від одного. У таких випадках алгоритм виконує обчислення найкоротших шляхів для усіх вузлів графа, навіть для тих, які є абсолютно неважливими для формування оптимального маршруту до цілі. Це призводить до того, що значна частина обчислень стає надлишковою, що робить увесь процес затратним за часом виконання. Ще одним чинником, який може сприяти збільшенню часу обробки, є наявність у графі ребер значної довжини. В умовах графа з такими

характеристиками АД змушений витратити непропорційно великий час на обробку цих довгих сполучень [4].

Для успішного планування найкоротшого маршруту за допомогою АД, необхідно чітко визначити початкову позицію. Крім того, вводиться використання двох ключових структур даних у вигляді двох множин S і U . Множина S слугує для запису вершин, для яких вже знайдено найкоротший шлях від стартової точки, тоді як множина U містить вершини, для яких найкоротший шлях ще не визначено, разом із поточною оцінкою відстані між кожною вершиною та початковою точкою.

2. Алгоритм Флойда-Воршелла (Floyd-Warshall algorithm), опублікований Робертом Флойдом у 1962 році, є яскравим прикладом динамічного програмування. Його основна функція полягає в ефективному та одночасному пошуку найкоротших шляхів між усіма можливими парами вершин у зваженому графі, який може бути як неорієнтованим, так і орієнтованим. Це робить його інструментом для глобального аналізу зв'язності графа.

Алгоритм працює, порівнюючи всі можливі проміжні маршрути для кожної пари точок на графі. Припустимо, вершини графа пронумеровані від 1 до n , де n — загальна кількість вершин. Вводиться рекурентна функція $f(i,j,k)$, яка визначає найкоротший шлях від вершини i до вершини j , використовуючи як проміжні точки лише вершини з множини $\{1,2,\dots,k\}$. Кінцевою метою використання цієї функції є знаходження найкоротшого шляху між кожною парою вершин (i,j) , але тепер із можливістю інтерполяції вершин із множини до $k+1$. Процес обчислення починається з визначення функції $f(i,j,1)$ для кожної пари (i,j) , тобто з використанням лише вершини 1 як потенційного посередника. Отримані результати потім використовуються для обчислення $f(i,j,2)$, і так далі. Цей ітеративний процес триває, доки проміжний індекс k не досягне значення n . На цьому етапі найкоротший шлях буде знайдено для всіх пар (i,j) з урахуванням усіх можливих проміжних вершин [4].

Алгоритм Флойда-Воршелла складається з двох основних компонентів: побудови матриці шляхів та рівняння переходу станів.

1. Побудова матриці шляхів, яка на основі початкової матриці ваг графа послідовно конструює фінальну матрицю D_n , яка містить довжини найкоротших шляхів між кожною парою точок. Елемент на перетині рядка i та стовпця j матриці D_n представляє довжину найкоротшого шляху від вершини i до вершини j .
2. Рівняння переходу станів, представляє собою математичне рівняння, яке забезпечує ітеративне обчислення найкоротшої відстані між кожною точкою. Воно виглядає наступним чином:

$$d_{ij}^k = \min\{d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1}\}, \quad (1.2)$$

d_{ij}^k репрезентує найкоротший шлях від i до j , який може проходити через вершину k (i , можливо, через вершини з індексами меншими за k).

Наприклад, d_{ij}^0 є початковою довжиною ребра між вершинами i та j (або нескінченністю, якщо ребра немає). Завдяки своїй структурі, час обчислення цього алгоритму становить (n^3) , де n — кількість вершин у графі.

3. Алгоритм Белмана-Форда (Bellman-Ford algorithm, BF) є класичною методологією, призначеною для обчислення найкоротших шляхів від єдиного джерела у зваженому графі. Ключова перевага цього алгоритму полягає в його здатності коректно працювати з ребрами, що мають від'ємні ваги. Це дає йому змогу ефективно обробляти графи, які можуть містити цикли з від'ємною сумарною вагою. Такі цикли є проблематичними, оскільки вони дозволяють генерувати нескінченну кількість шляхів від початку до кінця, де кожен додатковий обхід циклу призводить до зменшення загальної довжини шляху. Алгоритм BF розроблений таким чином, щоб виявляти ці цикли та коректно визначати найкоротший шлях, якщо такий існує [4].

4. Незважаючи на існування значно ефективніших сучасних алгоритмів планування шляху, Bug-алгоритми зберігають свою важливість у робототехніці. Вони були одними з найперших методів навігації та уникнення перешкод, які

продемонстрували відносно надійні результати при високій швидкості обчислення. Ці алгоритми розроблені на основі спрощеної моделі, яка припускає, що робот є єдиною точкою в двовимірному просторі, а його рух моделюється як перехід між послідовними точками (рис. 1.10). Bug-алгоритми є популярним класом навігаційних стратегій для роботів, які забезпечують проходження траєкторії вздовж контуру перешкоди в сценаріях з невідомими статичними перешкодами, імітуючи поведінку комахи, яка повзає навколо об'єкта. Алгоритми поділяються на три основні варіанти, що відрізняються своєю логікою обходу перешкод [4].

Алгоритм Bug-1 активується в момент, коли робот виявляє перешкоду на своєму шляху. Робот негайно починає повний обхід навколо перешкоди, доки не повернеться до точки свого первісного зіткнення. Під час цього повного обходу він фіксує та постійно обчислює найкоротшу відстань від поточної позиції до кінцевого пункту призначення. Робот реєструє точку відходу, яка відповідає мінімальній відстані до цілі, зафіксованій під час обходу. Після завершення повного кола, він відновлює рух уздовж контуру перешкоди, рухаючись до зареєстрованої точки відходу. Досягнувши цієї точки, робот відновлює прямолінійний рух до пункту призначення, ігноруючи перешкоду.

Алгоритм Bug-2 спочатку визначає пряму лінію, що сполучає стартову позицію з кінцевою. Робот слідує цій лінії, доки не зіткнеться з перешкодою. У момент переривання прямого шляху, він починає рухатися вздовж краю перешкоди. Під час обходу робот безперервно перевіряє своє положення щодо прямої лінії. Він обчислює нову траєкторію від кожної поточної позиції. Коли нова траєкторія знову збігається з початковою лінією і дозволяє рухатися до мети, ця позиція реєструється. Досягнувши цієї нової "точки відходу" на прямій лінії, робот відновлює рух по цій лінії до кінцевого пункту призначення [8].

На відміну від попередніх, алгоритм Dist-Bug покладається на постійний аналіз відстаней як до цілі, так і до контуру перешкоди. При виявленні перешкоди, робот починає слідувати її краю, одночасно на кожному кроці обчислюючи відстань між своєю поточною позицією та пунктом призначення.

Точка відходу в цьому алгоритмі визначається як позиція з найменшою дистанцією до цілі з усіх, що були зафіксовані під час обходу. Ця позиція називається точкою мінімальної дистанції. Як тільки робот, рухаючись навколо перешкоди, знаходить цю точку мінімальної дистанції, він негайно створює новий шлях і залишає контур, щоб продовжити рух до мети.

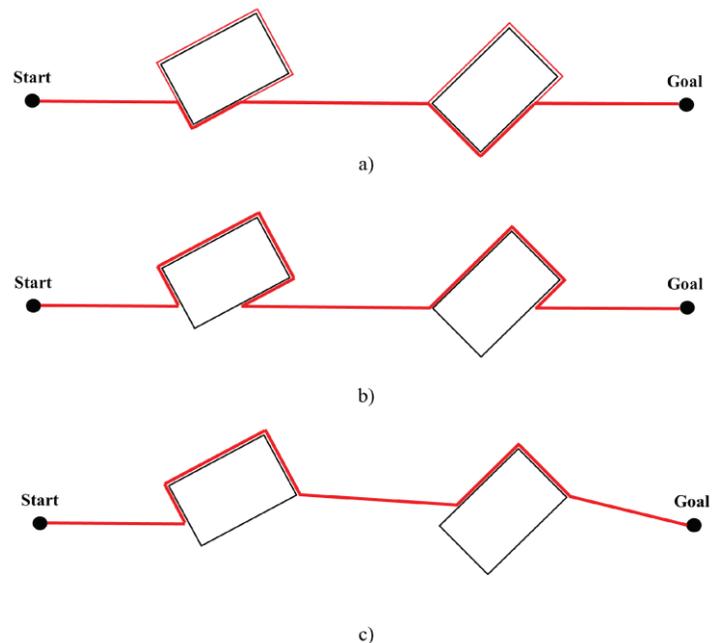


Рисунок 1.10 – Уникнення перешкод за допомогою алгоритмів Bug

5. Метод розкладання на клітини (Cell-by-cell technique, CD) - це стратегія планування простору, при якій вся робоча область поділяється на множину сіток, що не перекриваються, які називаються клітинами. Навігація в цьому середовищі здійснюється шляхом послідовного переходу від однієї клітини до іншої, починаючи від початкової клітини і прокладаючи шлях до цільової (рис. 1.11). Цей метод класифікується на три основні категорії залежно від того, як визначаються межі між клітинами: точне, апроксимативне та імовірнісне розкладання на клітини. Для точного розкладання на клітини характерна відсутність втрати роздільної здатності (тобто розрізнявальна здатність зберігається), а форма та розмір клітин не є фіксованими. Кожен елемент простору, або регіон, отримує унікальний ідентифікатор. Навпаки, при апроксимативному розкладанні на клітини результат декомпозиції лише

наближає реальну карту, а сітка має фіксовану форму та розмір (наприклад, квадратні клітини). Нарешті, імовірнісне розкладання на клітини схоже на апроксимативне, за винятком того, що межі клітин не мають прямого фізичного значення, а представляють собою лише концептуальні роздільники простору [8].

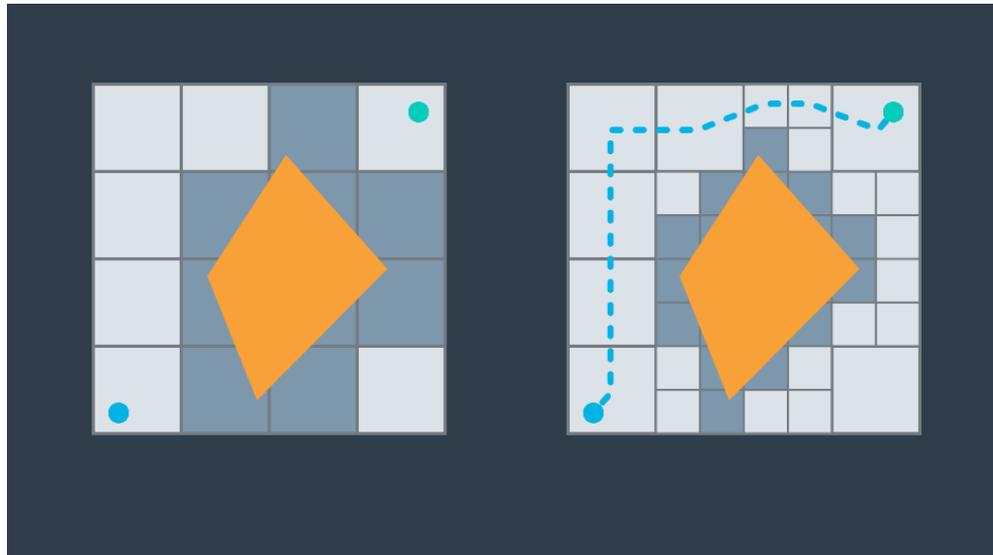


Рисунок 1.11 – Знаходження найоптимальнішого шляху методом розкладання на клітини

6. Класичні підходи до планування шляху стикаються з низкою суттєвих недоліків, що обмежують їхню ефективність у реальних сценаріях. Серед основних проблем — високі вимоги до часу обчислення при роботі з великомасштабними картами, а також схильність до застрягання у локальних мінімумах. Ці обмеження роблять класичні алгоритми малоефективними для практичного застосування. Щоб подолати ці недоліки та суттєво підвищити швидкість та надійність планування, були запропоновані імовірнісні алгоритми. Ці методи націлені на генерацію практичних і функціональних маршрутів для роботів, які переміщуються у статичних робочих просторах [8].

7. Одним із найважливіших прикладів у цій категорії є Метод Імовірнісної Дорожньої Карти (Probabilistic Roadmap Method, PRM). Цей підхід використовує ребра для моделювання зв'язності вільних від перешкод областей робота. PRM фактично будує представлення простору за допомогою графа, вузли якого

розташовані у вільній зоні, а ребра сполучають ці вузли. Серед ключових концепцій, які використовуються для формування цієї "дорожньої карти", виділяють граф видимості та діаграму Вороного. Граф видимості з'єднує дві точки, якщо пряма лінія між ними не перетинає жодної перешкоди. Натомість, діаграма Вороного створює мережу, яка проходить посередині між перешкодами, максимізуючи відстань від робота до будь-якої стіни, чим забезпечує підвищену безпеку маршруту (рис. 1.12).

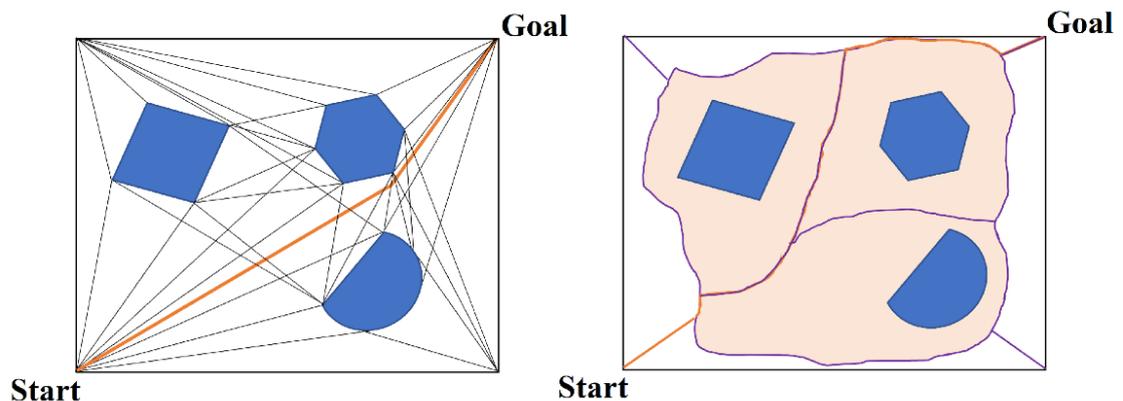


Рисунок 1.12 – Графік видимості (ліворуч); графік Вороного (праворуч)

У графі видимості перешкоди в робочому середовищі моделюються у вигляді багатокутників. Вертикальні вузли цих полігональних перешкод з'єднуються таким чином, щоб мінімізувати загальну довжину шляху, при цьому ребра проходять максимально близько до контурів перешкод. На противагу цьому, діаграма Вороного використовує дві найближчі точки на краях перешкод для планування та розбиває робочу область на піддомени. У випадку діаграми Вороного, робот навмисно рухається далі від перешкод, що, безумовно, підвищує безпеку навігації. Однак, цей підхід, як правило, призводить до значно довших маршрутів порівняно з графом видимості [9].

Для встановлення зв'язку між початковим станом робота та цільовою областю, Метод Імовірнісної Дорожньої Карти (PRM) досліджує збудований граф і точно визначає послідовність станів та локальних з'єднань, якими може пройти робот. Хоча теоретично ці алгоритми можуть створювати доволіно точні

представлення простору (якщо кількість пробних зразків прямує до нескінченності), на практиці для визначення маршрутів рішення потрібна лише невелика кількість критично важливих станів. Ці критичні стани часто мають особливу структуру, наприклад, слугують входами до вузьких проходів або "горловин", але їхнє точне виявлення може вимагати лише ретельного вибіркового зондування. Ці алгоритми пропонують високоточні представлення з теоретично нескінченною кількістю зразків. Однак, на практиці, така висока точність потрібна лише в обмежених випадках, як-от необхідність проходження крізь вузьке місце [10].

1.4 Висновки до розділу

Ефективне планування шляху та уникнення перешкод є критичною основою для надійної роботи автономних роботів. Навігаційні стратегії поділяються на глобальне планування (на основі відомих карт для повної оптимізації маршруту) та локальне планування (адаптація до динамічних чи невідомих середовищ у реальному часі). Класичні методи, які розглядалися, включають алгоритми пошуку найкоротшого шляху на графах — Дейкстри, Белмана-Форда та Флойда-Воршелла, а також такі підходи, як метод розкладання на клітини та Bug-алгоритми, необхідні для обходу невідомих об'єктів. Незважаючи на свою фундаментальність, ці підходи часто стикаються з високими часовими вимогами на великих масштабах та ризиком застрягання у локальних мінімумах.

Для подолання цих недоліків і підвищення ефективності були розроблені імовірнісні алгоритми, які генерують практичні маршрути. Найважливішим прикладом є Метод Імовірнісної Дорожньої Карти (PRM). PRM моделює простір, використовуючи графічні представлення, як-от граф видимості (для мінімізації довжини шляху) та діаграма Вороного (для максимізації безпеки та віддаленості від перешкод). Ці методи дозволяють швидко знаходити функціональні

маршрути, навіть якщо лише невелика кількість критичних станів (наприклад, входи до вузьких проходів) має бути точно ідентифікована.

Таким чином, для досягнення надійної автономії потрібен гібридний підхід. Він має поєднувати глобальну оптимізацію маршруту, забезпечену алгоритмами графів, з локальною адаптацією, що використовує імовірнісні методи та сенсорні дані для безпечного уникнення непередбачених перешкод. Подальший розвиток робототехніки вимагає постійної інтеграції та вдосконалення цих фундаментальних алгоритмічних рішень для забезпечення безпеки та ефективності у складних операційних середовищах.

2 РОЗРОБКА ІНФОРМАЦІЙНО-ВИМІРЮВАЛЬНОЇ СИСТЕМИ

2.1 Розробка загальної структури

Розробка загальної структури рухомої платформи є одним із ключових етапів створення системи автономного керування, оскільки саме на цьому етапі визначається архітектура апаратної частини, взаємозв'язки між основними модулями та логіка їхньої взаємодії. Метою розробки загальної структури є формування цілісної системи, здатної виконувати рух у просторі з одночасним виявленням та оминанням перешкод без втручання оператора [1].

Структура розробленої рухомої платформи ґрунтується на принципі модульності, що забезпечує простоту обслуговування, гнучкість модернізації та можливість подальшого розширення функціоналу. Кожен модуль системи виконує певну функцію - від збору даних навколишнього середовища до реалізації руху та повороту коліс. Основними функціональними блоками системи виступають модулі для реалізації керування, руху, обертання сенсору, та самої роботи сенсору.

Розроблена рухома платформа є прототипом автономного мобільного робота, призначеного для пересування по поверхні з можливістю розпізнавання перешкод та ухилення від них. Робота системи базується на циклічному процесі зчитування даних від ультразвукового сенсора, аналізу отриманої інформації мікроконтролером Raspberry Pi та формуванні відповідних керуючих сигналів до виконавчих механізмів [2].

Принцип функціонування можна описати тим, що сенсор здійснює сканування навколишнього простору, виявляючи наявність перешкод на визначеній відстані, після чого контролер обробляє отриману інформацію та приймає рішення про подальший напрямок руху, передаючи сигнали на крокові двигуни й сервоприводи. У разі виявлення перешкоди на небезпечній відстані, система генерує команду для зміни траєкторії руху, що дозволяє уникнути зіткнення та переналаштувати свій маршрут.

Наступним варто розглянути складові елементи загальної структури:

1. Центральним елементом системи виступає мікрокомп'ютер Raspberry Pi, який виконує функції обробки даних, логічного аналізу та формування керуючих сигналів. Завдяки вбудованому процесору та достатній кількості портів введення-виведення, Raspberry Pi забезпечує можливість одночасного підключення кількох сенсорів і виконавчих пристроїв. На ньому реалізується програмна частина системи, що містить алгоритми обробки даних з сенсору, передачі сигналів на поворотну та систему забезпечення руху [7].

2. До сенсорної частини входить ультразвуковий сенсор, який виконує роль основного пристрою виявлення перешкод. Для збільшення поля зору сенсор встановлений на сервопривід, що забезпечує його обертання направо, та наліво. Завдяки цьому реалізується періодичне сканування простору попереду, що дозволяє визначати наявність різних об'єктів на різних напрямках. Це підвищує точність аналізу навколишнього середовища та ефективність ухилення від перешкод.

3. Рух платформи забезпечується двома кроковими двигунами, які розташовані на задніх колесах які отримують керуючі сигнали від контролера через драйвер керування, що дозволяє незалежно регулювати швидкість їх обертання. Такий підхід забезпечує стабільність руху, високу точність позиціонування та можливість реалізації як прямолінійного, так і диференційного керування швидкістю.

4. Реалізація повороту базується на двох сервоприводах, що закріплені на передніх колесах платформи. Вони отримують сигнали з мікроконтролера та змінюють кут положення коліс відповідно до необхідного напрямку руху. Така конструкція дозволяє забезпечити плавний, та можливий більш різкий, поворот конструкції.

5. Живлення всієї системи здійснюється від акумуляторної батареї, що забезпечує автономність роботи платформи. Використання автономного джерела живлення дозволяє експлуатувати пристрій більшу кількість часу, та у

середовищах, де немає прямого доступу до мережі, а також запобігає виникненню шумів у сигнальних лініях [8].

Графічна архітектура системи побудована за ієрархічним методом. На верхньому рівні цієї структури розташований контролер, який виконує функцію координації роботи всіх інших модулів. На нижньому рівні знаходяться виконавчі пристрої, що безпосередньо здійснюють рух, а також сенсорні елементи, які збирають та надають інформацію про стан навколишнього середовища. Між усіма модулями відбувається двосторонній обмін інформацією, в самій основі, це комунікація між сенсором та мікроконтролером, після чого, команди поступають вже на виконавчі, рухомі модулі.

Алгоритм взаємодії між складовими системи можна описати послідовно:

1. Сенсор здійснює сканування простору перед платформою.
2. Дані про відстань до об'єктів передаються до Raspberry Pi.
3. Мікроконтролер аналізує інформацію та визначає, чи є на шляху перешкоди.
4. Якщо перешкода відсутня, контролер надсилає команди на крокові двигуни для прямолінійного руху.
5. Якщо-ж виявлено перешкоду, контролер надсилає команду на зміну напрямку, та продовження руху.
6. Після зміни напрямку цикл повторюється, забезпечуючи безперервний автономний рух із уникненням зіткнень.

На рис. 2.1 показано спрощену структурну схему, що відображає логічні взаємозв'язки між компонентами рухомої платформи.

Обрана структура системи має низку значних переваг. Завдяки модульності, кожен її компонент може бути легко замінений або вдосконалений без необхідності зміни загальної архітектури. Система забезпечує автономність роботи, оскільки живлення від батареї гарантує її незалежність від зовнішніх джерел енергії. Крім того, архітектура демонструє високу гнучкість, адже існує можливість легко розширити систему новими сенсорами, наприклад, інфрачервоними сенсорами (ІЧ) або камерою [15].

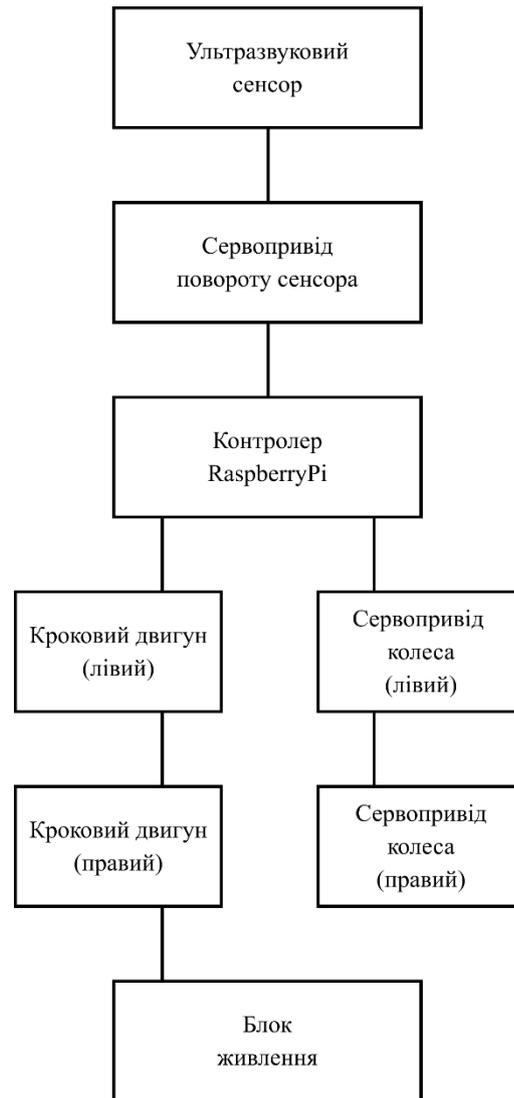


Рисунок 2.1 – Спрощена структурна схема логічних взаємозв'язків рухомої платформи

Що стосується розробки, то простота програмування на базі Raspberry Pi дозволяє реалізувати складні алгоритми аналізу даних в таких середовищах, як Python або C++, ефективно використовуючи спеціалізовані бібліотеки для роботи з сервоприводами та двигунами. Також, програмна реалізація використання крокових двигунів забезпечує можливість налаштування точності керування, гарантуючи контрольоване позиціонування та плавність руху платформи [16].

Також, під час роботи автономної платформи відбувається безперервний процес збору та аналізу даних. Використовуючи зовнішні бібліотеки, у

розробника з'являється задавати власні відстані при яких система зупиняє рух, виконує обертання сенсора для аналізу навколишнього середовища, вибирає оптимальний напрямок та змінює положення передніх коліс. Завдяки цьому реалізується логіка автономного оминання перешкод без участі людини [19].

Такий підхід дозволяє розглядати систему як базову платформу для майбутніх розробок у галузі мобільної робототехніки, включаючи можливість адаптувати програмну частину під більш вузькі потреби, до прикладу різні поверхні, можливі зміни довколишнього світу, або-ж фізичні параметри які можуть створювати перешкоди для роботи сенсору, коліс.

Завершення етапу розробки загальної структури рухомої платформи дозволило чітко визначити основні функціональні модулі системи, а також встановити принципи їхньої взаємодії та розподілу ролей. Запропонована архітектура закладає основу для стабільної роботи, забезпечує гнучкість у подальшій модернізації і демонструє достатню ефективність для вирішення ключових задач: автономного пересування та оминання перешкод. Наступні кроки розробки будуть зосереджені на виборі оптимальних сенсорів, механізмів руху та повороту, а також на детальній реалізації апаратної й програмної частин системи на базі цієї затвердженої структури.

2.2 Вибір ультразвукового сенсору для системи

Ультразвуковий модуль визначення дальності (також відомий як далекомір або сенсор) використовує фундаментальний фізичний принцип відбиття звукових хвиль для точного вимірювання відстані до об'єкта. Його робота ґрунтується на явищі, коли високочастотні звукові коливання, нечутні для людського вуха, відомий як ультразвук, що генеруються та випромінюються модулем, будуть відбиті, стикнувшись із будь-якою перешкодою чи поверхнею на своєму шляху.

Ключовим аспектом функціонування даного пристрою є високоточні вимірювання. Цей процес розпочинається в момент передачі ультразвукового

імпульсу в навколишній простір, після того як випромінене звукове коливання досягає об'єкта, воно відбивається назад до модуля. Сенсор починає відлік часу (зазвичай за допомогою мікроконтролера або інтегрованої схеми) саме з моменту відправлення сигналу [1].

Відлік часового інтервалу завершується лише тоді, коли модуль успішно приймає відлуння - відбиту ультразвукову хвилю. Різниця в часі, зафіксована між моментом випромінювання та прийомом, являє собою загальний час подорожі ультразвукової хвилі.

Для перетворення виміряного часу на фізичну відстань модуль спирається на сталу швидкість поширення звуку в повітрі, яка за стандартних умов є відомою величиною.

Швидкість звуку у сухому повітрі при температурі зазвичай приймається приблизно як 343(м/с). Хоча вона може незначно змінюватися залежно від температури, вологості та тиску, для більшості практичних застосувань ця константа є достатньо точною.

Знаючи швидкість хвилі та загальний час її подорожі, можна обчислити дистанцію від ультразвукового сенсора до перешкоди. Оскільки вимірний час включає шлях туди та назад, загальну відстань потрібно розділити навпіл (рис. 2.2).

Таким чином, формула для розрахунку відстані (s) виглядає так:

$$s = \frac{v \cdot \Delta t}{2}, \quad (2.1)$$

Де:

s - шукана відстань до об'єкта (в метрах);

v - швидкість звуку в повітрі (приблизно 340 м/с);

Δt - вимірний час між випромінюванням та прийомом (в секундах).

Цей механізм дозволяє модулю ефективно та безконтактно визначати розташування об'єктів у просторі.

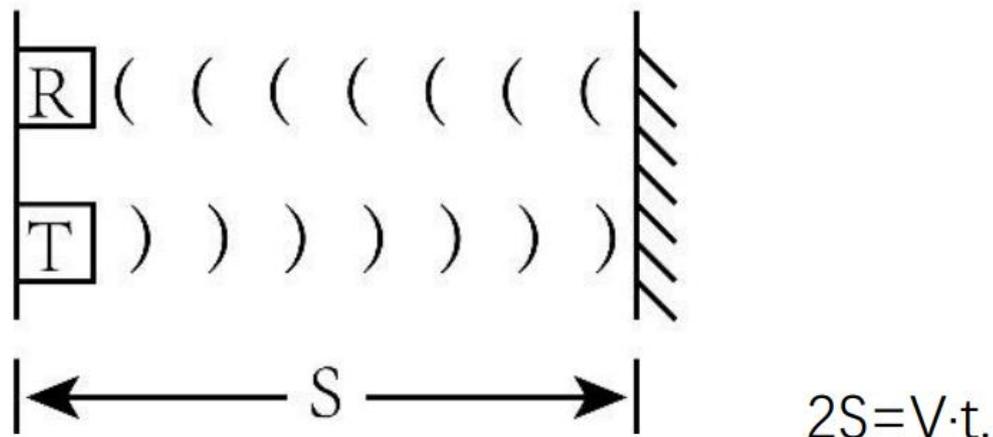


Рисунок 2.2 – Визначення відстані за часом проходження
ультразвукового імпульсу

Ультразвуковий далекомірний модуль HC-SR04 є інтегрованим пристроєм, що ефективно поєднує в собі два ключові компоненти, ультразвуковий випромінювач та ультразвуковий приймач (рис. 2.3). Ця двокомпонентна структура дозволяє модулю виконувати повний цикл вимірювання - від генерації сигналу до реєстрації його відбиття [7].

Роль передавача та приймача описується наступним чином:

1. Ультразвуковий передавач - функція якого полягає в трансформації енергії, яку він перетворює електричний сигнал, що подається з керуючої електроніки (як правило, мікроконтролера), в механічні коливання, а саме - високочастотні звукові хвилі. Ці хвилі відносяться до ультразвукового діапазону, тобто їхня частота перевищує поріг чутності людського вуха. Фактично, передавач здійснює перетворення електричної енергії в механічну, звукову енергію.

2. Ультразвуковий приймач виконує протилежну функцію. Він слугує для уловлювання відбитого від об'єкта ультразвукового імпульсу. Коли механічна енергія відбитої звукової хвилі досягає його поверхні, приймач конвертує ці механічні коливання назад в електричний сигнал. Це завершує цикл

вимірювання, дозволяючи електроніці модуля зареєструвати час прибуття відлуння і, відповідно, обчислити відстань.



Рисунок 2.3 – Вигляд ультразвукового модуля HC-SR04

Таким чином, HC-SR04 використовує цю схему для реалізації принципу ехолокації, він випускає хвилю, фіксує відлуння, і, на основі часу проходження сигналу, визначає дистанцію до найближчої перешкоди [15].

Модуль ультразвукового далекоміра HC-SR04 має чотири основні виводи, кожен з яких виконує критично важливу функцію для живлення та обміну даними з мікроконтролером (рис. 2.4):

1. VCC (Напруга живлення) - основний вхідний контакт, призначений для подачі позитивного полюса живлення. Для коректної роботи модуля на нього зазвичай подається стабілізована напруга 5V.
2. GND (Земля) – є загальною точкою відліку потенціалу для модуля. Його необхідно підключити до загальної шини живлення або мінусового полюса електричної схеми.
3. Trig (Trigger, запуск) - вхідний керуючий контакт, який використовується для початку циклу вимірювання відстані. Щоб

розпочати випромінювання ультразвуку, на цей контакт подається короткий імпульс.

4. Echo (Відлуння) - вихідний контакт даних, який відображає тривалість часу подорожі звукової хвилі. Модуль виводить на цей контакт імпульс високого рівня, тривалість якого прямо пропорційна виміряній відстані до об'єкта.

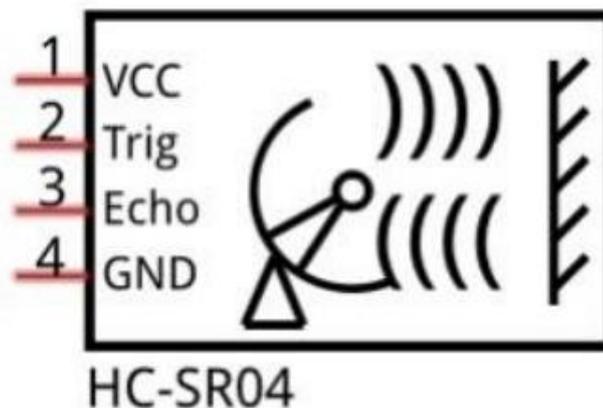


Рисунок 2.4 – Схема підключення ультразвукового далекоміра HC-SR04

Пристрій функціонує від номінальної напруги 5V постійного струму, що є загальноприйнятим стандартом для цифрової електроніки. У режимі активного вимірювання модуль підтримує низьке споживання енергії, вимагаючи близько 12mA робочого струму. Це забезпечує енергоефективність і мінімізує навантаження на джерело живлення схеми [19].

Сенсор забезпечує вимірювання дистанції в чітко визначеному діапазоні. Його здатність до визначення об'єктів починається від мінімальної вимірюваної відстані 2cm, що є найближчою точкою, де можна отримати достовірний сигнал. В той час як максимальна ефективна дальність вимірювання становить 200cm, або ж 2m.

Об'єкти, розташовані далі від цієї межі, в більшості своїй, не реєструються або дають неточні вимірювання через розсіювання або згасання ультразвукової хвилі.

2.3 Вибір механізмів для забезпечення руху

Кроковий двигун є електромеханічним пристроєм, що працює за принципом системи керування з розімкнутим контуром (рис. 2.5). Його головна функція полягає у послідовному перетворенні дискретних електронних імпульсів у точно визначене кутове або лінійне переміщення [3].



Рисунок 2.5 – Вигляд крокового мотору

Ключова перевага крокових моторів полягає в їхній нечутливості до зміни навантаження (за умови, що воно не перевищує максимальне значення). На відміну від двигунів постійного струму, де швидкість і позиція можуть коливатися при зміні зовнішнього опору, швидкість обертання та кінцеве положення крокового двигуна на задніх колесах визначаються виключно частотою та кількістю керуючих імпульсів [4].

Це забезпечує високу точність позиціонування системи у просторі та повторення руху, що критично важливо для задніх коліс рухомих платформ або роботів, які вимагають високо контролю траєкторії, особливо при нестандартних

умовах довколишнього середовища як наприклад пісок, багнюка, каміння, де життєво-необхідно забезпечувати тонке налаштування та можливість забезпечення різної швидкості ходу коліс, для подолання нестабільних місць території.

Для забезпечення коректної роботи крокового двигуна необхідно правильно організувати його електричне підключення до системи керування. Найчастіше електродвигун під'єднують через спеціальний драйвер (наприклад, ULN2003 або A4988), який відповідає за забезпечення живлення між обмотками стартера. Драйвер, у свою чергу, отримує сигнали керування від мікроконтролера, такого як Raspberry Pi або Arduino, через цифрові виходи GPIO. Завдяки цьому мікроконтролер подає послідовність імпульсів, які визначають напрямок, швидкість та режим обертання ротора [8].

Типова схема підключення передбачає подачу живлення до клем VCC і GND, під'єднання чотирьох керуючих ліній (IN1–IN4) між драйвером і обмотками двигуна, а також, за потреби, використання окремого джерела живлення для підвищення крутного моменту (рис. 2.6).

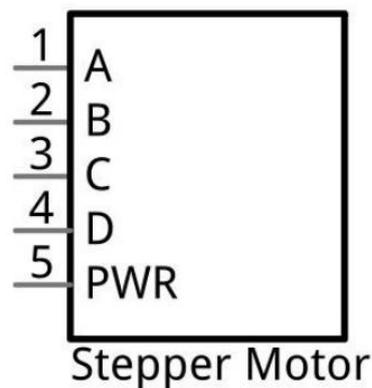


Рисунок 2.6 – Схема підключення крокового мотору

Наведена на рис. 2.7 електрична принципова схема ілюструє типову внутрішню організацію чотирифазного крокового електродвигуна. Розуміння його будови є ключовим для пояснення роботи цього типу приводу:

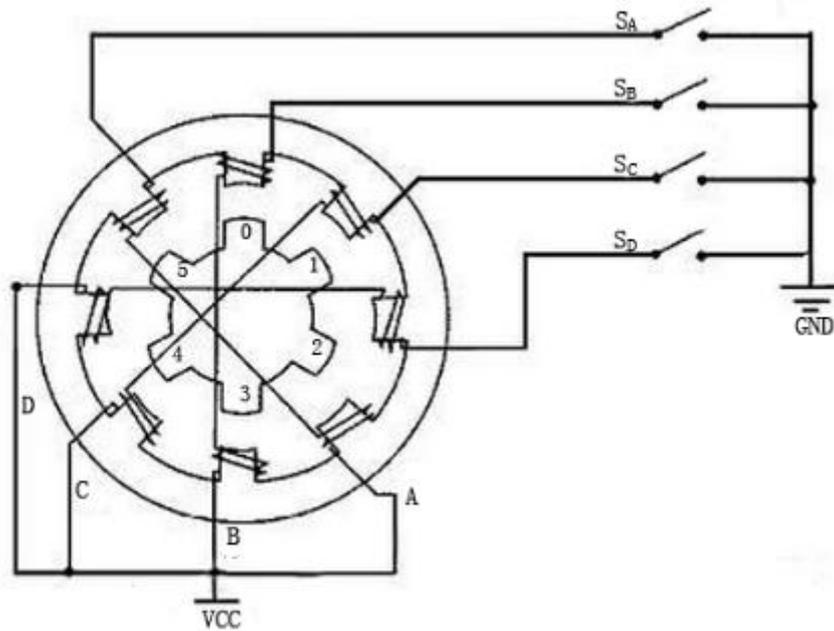


Рисунок 2.7 – Конструктивне зображення крокового двигуна з позначенням фаз

1. Статор - це зовнішній нерухомий корпус або корпус двигуна. На статорі розміщується певна кількість окремих обмоток. Ця кількість зазвичай є цілочисловим кратним кількості фаз двигуна. У чотирифазному двигуні це можуть бути 4, 8, 12 або більше обмоток, згрупованих по чотири.
2. Ротор - це внутрішня рухома частина, яка обертається всередині статора. Сам ротор може бути виготовлений із заліза або, частіше, із постійного магніту (гібридні чи з постійним магнітом). Поверхня ротора часто має спеціальні виступи, зубці, або опуклі діагональні пази, які взаємодіють з магнітним полем статора.

Коли на обмотки статора подається живлення, навколо них формується електромагнітне поле, яке створює силу, яка притягує або відштовхує відповідні магнітні полюси чи феромагнітні виступи на поверхні ротора [9].

Обертання ротора здійснюється через послідовне та впорядковане подання живлення на групи обмоток статора. Цей процес, який називається комутацією фаз, створює магнітне поле, що обертається або "крокує".

Кожен імпульс живлення перемикає магнітне поле на наступну фазу, змушуючи ротор підтягуватися до нового положення. Таким чином, замість

безперервного обертання, кроковий двигун виконує серію роздільних рухів, відомих як "кроки". Ця властивість і забезпечує його високу точність позиціонування, про яку йшлося раніше [10].

Робота крокового двигуна базується на дискретному русі, при якому кожне перемикання живлення обмоток викликає поворот ротора на певний фіксований кут, який називається "кроком" (рис. 2.8). Здатність двигуна до точного позиціонування та контролю швидкості без зворотного зв'язку забезпечується двома основними параметрами керуючого сигналу. Шляхом точного підрахунку кількості комутаційних кроків можна безпосередньо контролювати сумарний кут обертання вала двигуна. Одночасно, визначення часового інтервалу між послідовними кроками дозволяє регулювати швидкість обертання двигуна, адже менший інтервал відповідає вищій частоті імпульсів і, відповідно, більшій швидкості [11].

Найпростішим методом керування є режим повного кроку (Full-Step Mode), коли одночасно живиться лише одна фаза. Для чотирифазного двигуна (A, B, C, D) для обертання за годинниковою стрілкою використовується послідовність подачі живлення: $A > B > C > D > A > \dots$. Ротор рухається відповідно до цієї послідовності, роблячи один повний крок за одне перемикання. Цей метод ще називають чотирикроковим, або-ж чотиричастинним. Для зміни напрямку на проти годинникової стрілки послідовність просто інвертується: $D > C > B > A > D > \dots$.

Існує також більш просунутий метод — режим напівкроку (Half-Step Mode), який дозволяє досягти кращої роздільної здатності руху. У цьому режимі чергується живлення однієї фази та живлення двох сусідніх фаз одночасно. Наприклад, після активації лише фази A, яка орієнтує ротор точно на неї, наступним кроком є одночасна активація фаз A та B. Ротор притягується до центру між фазами A і B, що призводить до повороту лише на половину кута повного кроку.

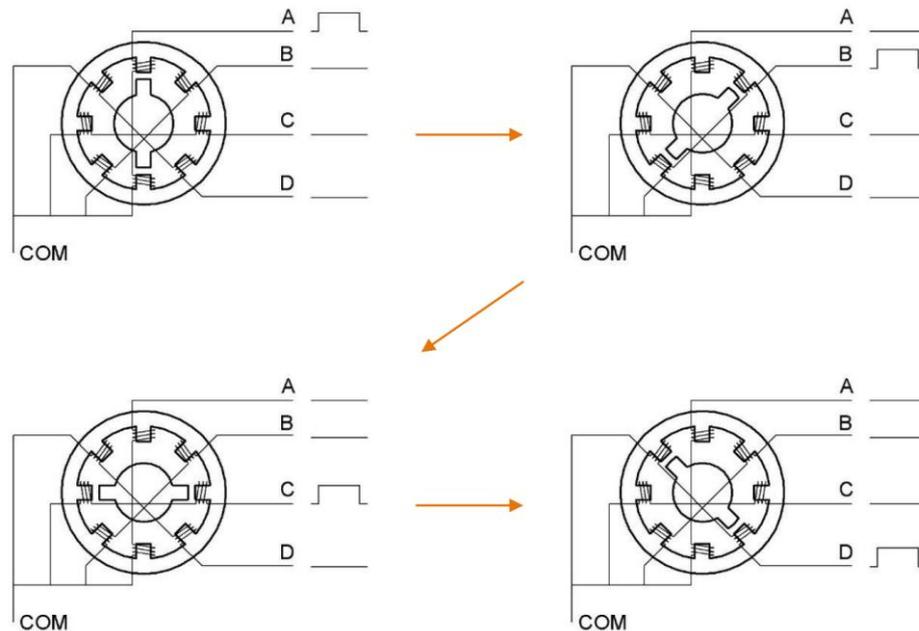


Рисунок 2.8 – Загальна послідовність рухів крокового мотору

Цей підхід підвищує стабільність роботи двигуна та знижує вібрацію і шум. Послідовність живлення при цьому виглядає так: $A > AB > B > BC > C > CD > D > DA > A > \dots$. Рух ротора відповідно до цієї послідовності називають чотирьохкроковим, восьмичастинним, і він також обертається у зворотному напрямку, якщо послідовність інвертується [12].

Для визначення фактичної точності руху кінцевого вала необхідно врахувати вбудований редуктор. У нашому випадку, статор двигуна має 32 магнітні полюси. Це означає, що для здійснення валом двигуна одного повного оберту до редуктора, необхідно подати 64 повних кроки.

Однак вал двигуна з'єднаний із редукційним механізмом, який має передавальне число 1:64. Це означає, що вал двигуна має обернутися 64 рази, щоб вихідний вал (той, що виходить із корпусу двигуна і приводить у рух колеса) обернувся лише один раз [12].

Таким чином, для того, щоб кінцевий вихідний вал зробив один повний оберт, необхідно помножити кількість кроків на оберт двигуна, та на коефіцієнт редукції:

Кроки на оберт вихідного вала = $32 * 64 = 2048$ кроків

Це демонструє, що використання редуктора разом із кроковим двигуном забезпечує високу точність керування у вигляді 2048 кроків на один оберт кінцевого вала.

2.4 Вибір механізмів для забезпечення повороту

Сервопривід (часто згадується як сервомотор) являє собою високоточний, самодостатній механізм, інтегрований у компактний корпус, і спеціально розроблений для забезпечення докладного контролю кутового положення вала та високого обертового моменту (рис. 2.9). Його внутрішня архітектура є складною системою, що працює за принципом замкнутого циклу керування [9].

Така система керування передбачає наявність безперервного циклу обміну даними між виконавчим механізмом і контролером. У процесі роботи контролер постійно отримує інформацію, аналізуючи та порівнюючи з початковим значенням. Якщо виникає похибка, система автоматично генерує коригувальний сигнал, завдяки чому забезпечується висока стабільність роботи навіть при зміні зовнішніх умов або навантаження.



Рисунок 2.9 – Вигляд сервоприводу з кабельним підключенням

Принцип замкненого циклу є фундаментом для більш складних систем автоматичного керування, таких як ПД-регулятори, які дозволяють досягати оптимального поєднання швидкодії, точності та плавності руху.

Для мобільних платформ, особливо тих, що використовують принцип автомобільного керування, сервопривід є ключовим виконавчим механізмом для повороту передніх коліс. Це дозволяє апарату змінювати напрямок руху, точно дотримуючись заданої траєкторії [10].

Сервопривід для керування зазвичай підключається до рульової трапеції або безпосередньо до шарнірів поворотного кулака передніх коліс за допомогою спеціального важеля.

Його робота ґрунтується на позиційному контролі:

1. Мікроконтролер надсилає сервоприводу сигнал ШІМ (широтно-імпульсна модуляція), тривалість якого кодує бажаний кут повороту.

2. Сервопривід інтерпретує тривалість імпульсу як абсолютне положення, і його внутрішня електроніка приводить двигун у рух доти, доки сенсор положення не підтвердить досягнення цього кута.

3. Після досягнення цільового кута, сервопривід активно утримує це положення, чинячи опір зовнішнім силам, наприклад, опору скочування або інерції, що є критично важливим для стабільного керування.

Сервопривід функціонує завдяки синергії чотирьох основних компонентів. Рушійною силою є двигун постійного струму (DC Motor), який генерує необхідну енергію для руху. Ця енергія передається через набір редукційних шестернів. Ця багатоступенева механічна система значно збільшує вихідний крутний момент, дозволяючи сервоприводу долати значні навантаження, при цьому зменшуючи швидкість обертання. Це робить сервомотор набагато ефективнішим для виконання позиційних завдань, ніж простий двигун без редуктора [11].

Залежно від типу конструкції та вимог до точності, у сучасних сервоприводах можуть використовуватися як звичайні колекторні двигуни постійного струму, так і безщіткові двигуни. Останні мають менший рівень

механічного зношування, підвищену ефективність і більшу довговічність, тому часто застосовуються в промислових або високоточних роботизованих системах. У дешевших моделях для спрощення конструкції використовуються щіткові двигуни, які забезпечують достатній рівень точності для навчальних або-ж побутових пристроїв.

Для досягнення високої точності пристрій оснащений сенсором положення, як правило, це прецизійний потенціометр, який постійно фіксує фактичний кут повороту вихідного вала. Ця інформація надходить до електронної плати керування. Після чого, схема порівнює задане зовнішнім сигналом бажане положення з поточним, і, якщо виявляється розбіжність, вона динамічно регулює потужність двигуна, доки вал не досягне цільового кута.

Для більшості стандартних сервоприводів характерний обмежений робочий діапазон руху, який зазвичай дорівнює 180 градусам. Обертання передається через спеціальну насадку, відому як хорн або важіль. Завдяки своїй здатності точно утримувати заданий кут, сервоприводи стали незамінними в багатьох галузях, як до прикладу у виконавчих механізмах в робототехніці (наприклад, для керування суглобами маніпуляторів), а також використовуються в авіа- та авто моделюванні для управління кермовими поверхнями чи рульовим механізмом [12].

Залежно від способу формування керуючого сигналу, сервоприводи поділяються на аналогові та цифрові, перші обробляють стандартний ШІМ-сигнал із частотою близько 50 Гц, реагуючи на зміну тривалості імпульсу, тоді як другі, цифрові, оснащені вбудованим мікроконтролером, який опрацьовує сигнал із частотою до 300–400 Гц. Це забезпечує значно швидшу реакцію, плавніше утримання положення та кращу стабільність при змінному навантаженні. Цифрові сервоприводи часто мають можливість перепрограмування та калібрування параметрів, що робить їх універсальними для різних сфер застосування — від аерокосмічних систем до мобільних роботів.

Підключення сервоприводу оснащена трьома основними провідними лініями, які забезпечують його живлення та можливість приймати команди

керування від зовнішньої електроніки (рис. 2.10). Їхнє функціональне призначення чітко розмежоване:

1. Signal (Сигнальна лінія) - вхідний контакт сервоприводу, який приймає послідовність імпульсів широтно-імпульсної модуляції (ШІМ) від мікроконтролера. Саме тривалість високого рівня у цьому сигналі безпосередньо кодує бажаний кут повороту вихідного вала, дозволяючи системі керування задавати точне позиціонування.
2. VCC (Напруга живлення) - вхідний полюс живлення модуля, через лінію якого подається необхідна робоча напруга, яка забезпечує енергією як внутрішню електронну схему, так і двигун постійного струму, що відповідає за обертання.
3. GND (Земля) - є загальною точкою відліку електричного потенціалу.

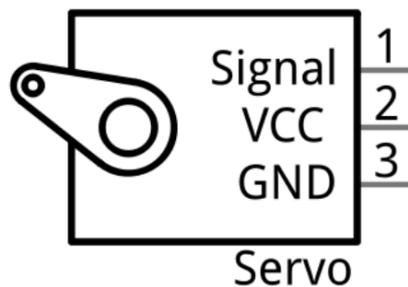


Рисунок 2.10 – Схема підключення сервоприводу

В окремих моделях також може бути передбачена четверта лінія - зворотного зв'язку або телеметрії, яка використовується для зчитування поточного кута повороту, швидкості обертання або температури двигуна в режимі реального часу. Такі рішення особливо корисні у складних багатокоординатних системах, де потрібен постійний моніторинг стану кожного виконавчого елемента. Наявність додаткового каналу даних дозволяє підвищити рівень адаптивності системи та мінімізувати похибку при динамічних змінах навантаження [13].

Для ефективного та точного позиціонування вала сервоприводу використовується метод широтно-імпульсної модуляції (ШІМ), який зображений на рис. 2.11.

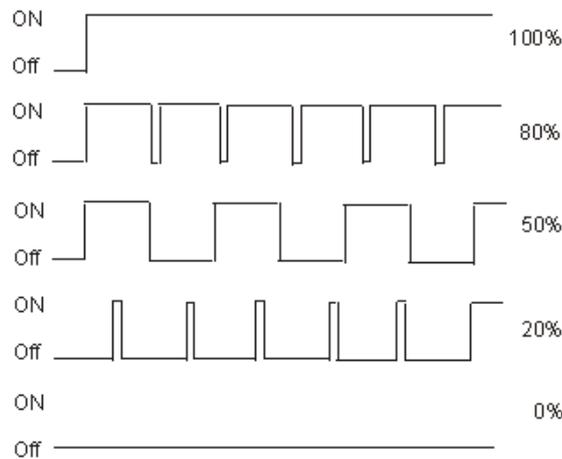


Рисунок 2.11 – Діаграма зміни шпаруватості ШІМ-сигналу

Сервоприводи розраховані на роботу з сигналом, що має фіксовану частоту - в даному випадку, це 50Hz. Період такого сигналу, відповідно, становить 20ms (1/50Hz).

Ключовим параметром, який безпосередньо визначає кутове положення сервоприводу, є тривалість імпульсу високого рівня, тобто коефіцієнт заповнення, у межах циклу, який дорівнює 20ms. Важливою особливістю є те, що кожна конкретна тривалість імпульсу відповідає абсолютному куту повороту вала, а не є накопичувальним приростом до попереднього положення.

Цей керуючий імпульс повинен лежати у строго визначеному діапазоні тривалостей:

- Якщо час високого рівня становить 0.5ms, сервопривід встановлюється у своє початкове положення — 0°.
- Якщо імпульс високого рівня триває 2.5ms, сервопривід досягає свого граничного положення — 180°.

Між цими двома крайніми точками існує лінійна залежність, що забезпечує плавне та передбачуване позиціонування. Наприклад, імпульс

тривалістю 1ms відповідає куту 45° , а імпульс тривалістю 1.5ms встановлює сервопривід точно посередині робочого діапазону на 90° . Продовжуючи цю закономірність, імпульс тривалістю 2ms призведе до повороту на 135° .

Таким чином, для того, щоб перемістити передні колеса на необхідний кут, потрібно просто змінити тривалість високого рівня у сигналі ШІМ на відповідне абсолютне значення. Модуль керування сервоприводу постійно зчитує цю тривалість і внутрішньо регулює живлення двигуна доти, доки сенсор положення не підтвердить досягнення заданого кута, забезпечуючи точне та швидке керування [14].

Важливим аспектом у роботі сервоприводів є коректне калібрування сигналів ШІМ. Невідповідність між заданими межами імпульсів та фактичними механічними обмеженнями може призвести до перевантаження редуктора або пошкодження потенціометра. Тому під час налаштування сервоприводу необхідно враховувати його паспортні характеристики, межі робочого кута та тривалість імпульсів, рекомендовану виробником. У мікроконтролерах калібрування здійснюється програмно шляхом задання мінімального, максимального та нейтрального положення. Такий підхід дозволяє забезпечити максимально плавну та безпечну роботу приводу навіть при тривалому навантаженні [14].

2.5 Розробка детальної структури системи

Етап детального архітектурного проектування структури рухомої платформи є критично важливим для забезпечення її функціональної цілісності, операційної надійності та подальшої масштабованості. Він охоплює систематичне визначення електричних, логічних і програмних інтерфейсів та взаємозв'язків між усіма складовими елементами системи.

На цьому рівні відбувається формалізація не лише призначення та функціоналу кожного модуля (сенсорного, обчислювального, виконавчого), але й методології їх фізичного інтегрування та організації інформаційного обміну.

Це включає специфікацію електричних параметрів (напруги, струми, частоти, схеми живлення) та вибір стандартизованих роз'ємів із забезпеченням цілісності сигналу. Також, відбувається деталізація логічних взаємозв'язків, що передбачає вибір комунікаційних протоколів таких CAN, або Ethernet, та топології шин даних. Особлива увага приділяється формалізації логіки роботи, від визначення операційних режимів (налаштування, автономний рух, діагностика) до деталізації інтерфейсу апаратно-програмної взаємодії (HSI). Такий підхід створює міцну архітектурну основу для подальшої розробки вбудованого програмного забезпечення та гарантує передбачуваність функціонування системи [17].

Системна архітектура рухомої платформи базується на принципі централізованого керування, де мікропроцесорний модуль Raspberry Pi функціонує як основний обчислювальний та керуючий вузол, забезпечуючи виконання алгоритмів керування, обробку даних сенсорів та координацію виконавчих механізмів [17].

Керування рухом реалізується через дві незалежні групи виконавчих пристроїв:

1. Приводи рушійної системи - два крокові двигуни інтегровані через відповідні драйвери (типу A4988 або ULN2003). Використання драйверів є критично важливим для підсилення струму та забезпечення мікрокрокового режиму. Це дозволяє центральному контролеру формувати точні послідовності імпульсів для високоточного позиційного керування та реалізації складних кінематичних траєкторій.

2. Приводи керування напрямком - сервоприводи передніх коліс безпосередньо підключені до GPIO-виходів Raspberry Pi. Керування їхнім кутом повороту здійснюється за допомогою генерації широтно-імпульсної модуляції (ШІМ / PWM). Використання ШІМ-сигналів забезпечує пропорційне керування сервоприводами, необхідне для динамічної зміни напрямку руху платформи.

Система сприйняття оточення побудована навколо ультразвукового далекомірного модуля з функцією активного сканування:

1. Ультразвуковий сенсор (HC-SR04) - модуль працює на принципі вимірювання часу прольоту (Time-of-Flight, ToF). Він підключається за допомогою двох спеціалізованих GPIO-контактів:

1.1. TRIG-контакт використовується для генерації короткого імпульсу тривалістю 10 с, що ініціює передачу ультразвукової хвилі.

1.2. ECHO-контакт приймає відбитий сигнал, передаючи контролеру тривалість імпульсу, на основі якої обчислюється відстань до об'єкта з урахуванням швидкості звуку.

2. Сервопривід сканування - окремий PWM-канал виділено для керування сервоприводом, який забезпечує обертання ультразвукового модуля в горизонтальній площині. Це надає платформі активну можливість сканування та значно розширює її кутове поле зору, що є необхідним для навігації та уникнення перешкод у динамічному середовищі [19].

3. Живлення платформи забезпечується літій-іонною батареєю або спеціалізованим акумуляторним модулем, який є основним мобільним енергетичним резервуаром. Електроживлення подається через систему стабілізації напруги, що необхідна для забезпечення гальванічної розв'язки та відповідності різним енергетичним доменам:

3.1. Напруга 5 В для живлення Raspberry Pi та цифрової логіки.

3.2. Напруга у діапазоні 6–12 В (залежно від специфікації) для забезпечення достатньої потужності двигунів та виконавчих механізмів.

4. Програмна взаємодія між компонентами рухомої платформи реалізується у вигляді багаторівневої архітектури, що забезпечує послідовну обробку, аналіз та передачу даних у режимі реального часу. На нижньому рівні функціонування відбувається безпосереднє керування апаратними засобами через GPIO-інтерфейси Raspberry Pi, з використанням бібліотек, орієнтованих на низькорівневе програмування, таких як RPi.GPIO або pigpio. Цей рівень відповідає за формування керуючих імпульсів, генерацію сигналів PWM та зчитування цифрових даних від сенсорів. Наступний, логічний рівень виконує роль системного посередника, який обробляє інформаційні потоки між

сенсорними та виконавчими модулями, застосовуючи алгоритми попередньої фільтрації, усереднення та виявлення аномалій у сигналах [19].

Ключову роль у забезпеченні когерентності обчислювальних процесів відіграє модуль синхронізації, який координує послідовність виконання окремих програмних потоків. У рамках цього рівня реалізовано структуру взаємодії типу «запит-відповідь» між обчислювальним ядром і периферійними пристроями, що дозволяє досягти детермінованої поведінки системи навіть за наявності змінних зовнішніх умов. Логіка керування рухом базується на безперервному циклі аналізу даних від ультразвукового сенсора, прийнятті рішення щодо корекції траєкторії та генерації відповідних команд для виконавчих механізмів. Алгоритмічна частина передбачає використання пропорційно-інтегрального принципу (PID-регулювання) для плавного коригування швидкості крокових двигунів і кута повороту сервоприводів, що забезпечує стабільність та точність переміщення платформи [21].

Додатковим елементом підвищення надійності є реалізація механізмів обробки помилок на рівні програмних інтерфейсів. Це дозволяє уникати непередбачуваних збоїв під час обміну інформацією між процесами або при тимчасовій втраті зв'язку із сенсорними модулями. Така структурна організація програмної взаємодії забезпечує не лише високу стабільність роботи платформи, але й створює основу для її подальшого інтелектуального розвитку, зокрема впровадження адаптивних алгоритмів навігації та самодіагностики.

2.6 Розробка програмного забезпечення

У цьому підрозділі представлено реалізацію програмного забезпечення для автономної рухомої платформи на базі Raspberry Pi.

Платформа складається з чотирьох коліс, де передні керуються серводвигунами, задні приводяться в рух кроковими моторами, а ультразвуковий сенсор встановлений на додатковому сервомоторі для сканування простору [17].

Реалізована робота системи полягає у безперервному моніторингу відстані до перешкод, автономному руху вперед та виконанні маневрів обходу, коли перешкоди виявляються. Кожен блок коду відповідає за конкретну функціональну складову платформи: керування колесами, рух задніх коліс або визначення перешкод.

Першим кроком підключаються необхідні бібліотеки для роботи платформи:

```
from gpiozero import AngularServo, OutputDevice,
DistanceSensor
from time import sleep
import random
import warnings

warnings.filterwarnings("ignore")
```

Наступним кроком ініціалізуються серводвигуни, які керують передніми колесами платформи. Кожен серводвигун підключається до певного GPIO-піну Raspberry Pi:

```
front_left_servo = AngularServo(17, min_angle=0,
max_angle=180)
front_right_servo = AngularServo(18, min_angle=0,
max_angle=180)
```

Для обертання крокових моторів визначені два послідовні набори сигналів: за годинниковою стрілкою та проти годинникової. Ця схема подачі живлення дозволяє контролювати напрямок обертання мотора:

```
rear_motor_pins = (19, 13, 6, 5) # GPIO для крокових моторів
rear_motors = [OutputDevice(pin) for pin in rear_motor_pins]

CCWStep = (0x01, 0x02, 0x04, 0x08)
```

```

CWStep = (0x08, 0x04, 0x02, 0x01)

def moveOnePeriod(direction, ms):
    if ms < 3:
        ms = 3
    for j in range(4):
        for i in range(4):
            if direction == 1:
                rear_motors[i].on() if (CWStep[j] == 1 << i)
            else rear_motors[i].off()
            else:
                rear_motors[i].on() if (CWStep[j] == 1 << i)
            else rear_motors[i].off()
        sleep(ms * 0.001)

def moveSteps(direction, ms, steps):
    for _ in range(steps):
        moveOnePeriod(direction, ms)

def stopRearMotors():
    for m in rear_motors:
        m.off()

```

Сервомотор, на який встановлено ультразвуковий сенсор, дозволяє змінювати кут його огляду. Це дає змогу сканувати простір перед платформою і визначити наявність перешкод у різних напрямках. Використання сервомотора для сенсора значно підвищує автономність платформи та точність визначення перешкод [19]

Наступний чином виконується його ініціалізація та задання мінімального, та максимального значення обертання:

```

sensor_servo = AngularServo(27, min_angle=0, max_angle=180)

```

Ультразвуковий сенсор підключається до двох GPIO-пінів Raspberry Pi: один для тригера, інший для Echo.

Використовуючи бібліотеку `gpiozero` та клас `DistanceSensor`, можна отримувати відстань до об'єкта перед платформою у метрах. Параметр `max_distance` визначає максимально можливу відстань вимірювання (в даному випадку 3 метри):

```
trigPin = 14
echoPin = 15
sensor = DistanceSensor(echo=echoPin, trigger=trigPin,
max_distance=3)
```

```
# Функції керування рухом
```

```
def moveForward(steps=512):
    front_left_servo.angle = 90    # прямиий рух
    front_right_servo.angle = 90
    moveSteps(1, 3, steps)
    stopRearMotors()
```

```
def turn(direction='left', steps=256):
    if direction == 'left':
        front_left_servo.angle = 45
        front_right_servo.angle = 135
        moveSteps(1, 3, steps)
    else:
        front_left_servo.angle = 135
        front_right_servo.angle = 45
        moveSteps(1, 3, steps)
    stopRearMotors()
    front_left_servo.angle = 90
    front_right_servo.angle = 90
```

```
def scanObstacle():
```

Наступний блок коду відповідає за повертання сенсору вправо, та вліво задля наступного порівняння результатів відстаней, та вирішення, поворот в яку сторону є більш оптимальним:

```

sensor_servo.angle = 90
sleep(0.5)
front_distance = sensor.distance * 100

sensor_servo.angle = 45
sleep(0.5)
left_distance = sensor.distance * 100

sensor_servo.angle = 135
sleep(0.5)
right_distance = sensor.distance * 100

sensor_servo.angle = 90
sleep(0.2)
return front_distance, left_distance, right_distance

```

Основний цикл програми безперервно сканує навколишній простір за допомогою ультразвукового сенсора. Якщо відстань перед платформою перевищує безпечний поріг (20 см), платформа рухається вперед.

У разі виявлення перешкоди платформа обирає бік з більшою відстанню для обходу та виконує поворот і невеликий рух вперед. Такий алгоритм дозволяє платформі автономно пересуватися по площині, уникаючи перешкод:

```

def loop():
    while True:
        front, left, right = scanObstacle()
        print(f"Front: {front:.1f}cm | Left: {left:.1f}cm |
Right: {right:.1f}cm")

```

```

        if front > 20: # якщо попереду більше ніж 20 см,
рухаємося вперед
            moveForward(steps=256)
        else:
            # Перешкода попереду – обираємо сторону для обходу
            if left > right:
                turn('left', steps=128)
            else:
                turn('right', steps=128)
            moveForward(steps=128)

```

Функція `destroy()` закриває всі підключення до GPIO, зупиняє крокові мотори та серводвигуни, а також звільняє ресурси ультразвукового сенсору. Це необхідно для безпечного завершення роботи програми та запобігання пошкодженню апаратної частини:

```

def destroy():
    stopRearMotors()
    front_left_servo.close()
    front_right_servo.close()
    sensor_servo.close()
    sensor.close()

```

І фінальною частиною коду є блок, який стартує та ініціалізує роботу усього файлу, приводячи до роботи циклічний алгоритм, а також задає умову при якій цей цикл буде закінчено:

```

if __name__ == '__main__':
    print("Рухома платформа стартує...")
    try:
        loop()
    except KeyboardInterrupt:
        print("Завершення програми")
    finally:

```

```
destroy()
```

В результаті виконання представленого програмного забезпечення платформа здатна автономно переміщатися, уникати перешкод та коригувати свій рух відповідно до оточення. Завдяки інтеграції серводвигунів, крокових моторів та ультразвукового сенсора, забезпечується скоординована робота всіх компонентів, що дозволяє платформі ефективно виконувати завдання з автономного пересування.

3 МОДЕЛЮВАННЯ РОБОТИ ДОСЛІДНОГО ЗРАЗКУ

3.1 Конструктивно-технічна частина дослідного зразку

Дослідний зразок є автономною системою, серцем якої виступає Raspberry Pi 5, встановлений безпосередньо на плату Freenove Projects Board (рис. 3.1). Ця спеціалізована плата-розширювач відіграє критичну роль інтегрованого комутатора та драйвера, значно спрощуючи апаратну частину. Замість складного монтажу на макетній платі, Freenove Board забезпечує стандартизовані роз'єми та вбудовані схеми захисту [15].



Рисунок 3.1 – Вимкнена макетна плата Freenove Board

Усі компоненти підключаються до відповідних, чітко маркованих роз'ємів на платі (рис. 3.3). Ультразвуковий сенсор підключається до свого 4-пінового роз'єму, використовуючи GPIO 14 (Trigger) та GPIO 15 (Echo), при цьому плата Freenove бере на себе функцію безпечного перетворювача напруги, захищаючи Raspberry Pi від 5V сигналу.

Сервомотор, який використовується для орієнтації, підключається до одного з роз'ємів Servo, і код використовує пін GPIO 17, а плата автоматично подає необхідне 5V живлення для його стабільної роботи. Керування кроковим двигуном здійснюється через вбудований драйвер на платі Freenove, який приймає логічні сигнали з пінів GPIO 19, 13, 6, 5, забезпечуючи необхідний струм для руху. Таким чином, плата Freenove перетворює складну електроніку на надійну, модульну конструкцію [15].



Рисунок 3.2 – Основні компоненти дослідного зразку

Фізична збірка із платою Freenove Projects Board є процесом прямого, модульного підключення. Спочатку Raspberry Pi 5 надійно фіксується у 40-піновому роз'ємі на платі Freenove, встановлюючи електричний та логічний зв'язок.

Сервомотор підключається безпосередньо до роз'єму Servo, що відповідає піну GPIO 17, дотримуючись колірного маркування дротів. Цей роз'єм забезпечує стабільне живлення сервомотора.

Ультразвуковий сенсор підключається до свого 4-пінового роз'єму Ultrasonic, який використовує піни GPIO 14 і GPIO 15. Завдяки інтегрованій

схемі на платі Freenove, де не потрібно турбуватися про використання зовнішнього дільника напруги [15].

Кроковий двигун підключається до роз'єму Stepping Motor, що відповідає вбудованому драйверу U11. Цей драйвер отримує команди від пінів GPIO 19, 13, 6, 5, визначених у нашому коді.

Важливо переконатися, що функціональні перемикачі Function Selection Switch на платі Freenove, які відповідають за мотори та сенсори, встановлені у положення "ON" або в інше необхідне робоче положення, щоб дозволити електричним сигналам досягати вбудованих драйверів.



Рисунок 3.3 – Дослідний зразок готовий до тестування

3.2 Розроблення програмного забезпечення та налагодження

Програмне забезпечення створено мовою Python і працює з використанням стандартної бібліотеки `gpiozero`, яка дозволяє безпосередньо керувати GPIO-пінами Raspberry Pi. Плата Freenove лише комутує ці стандартні виводи, тому жодних додаткових бібліотек чи драйверів встановлювати не потрібно.

Першим кроком є перевірка, чи встановлена `gpiozero` на Raspberry Pi 5. Далі наданий код копіюється у новий файл із розширенням `.py` (рис. 3.6), створений у

зручному середовищі, наприклад, як нашому випадку - у Thonny IDE (рис. 3.4) [17]. Код повністю сумісний із платою, оскільки використовує ті самі GPIO-контакти (17, 14, 15, 19, 13, 6, 5), які фізично доступні на платі.

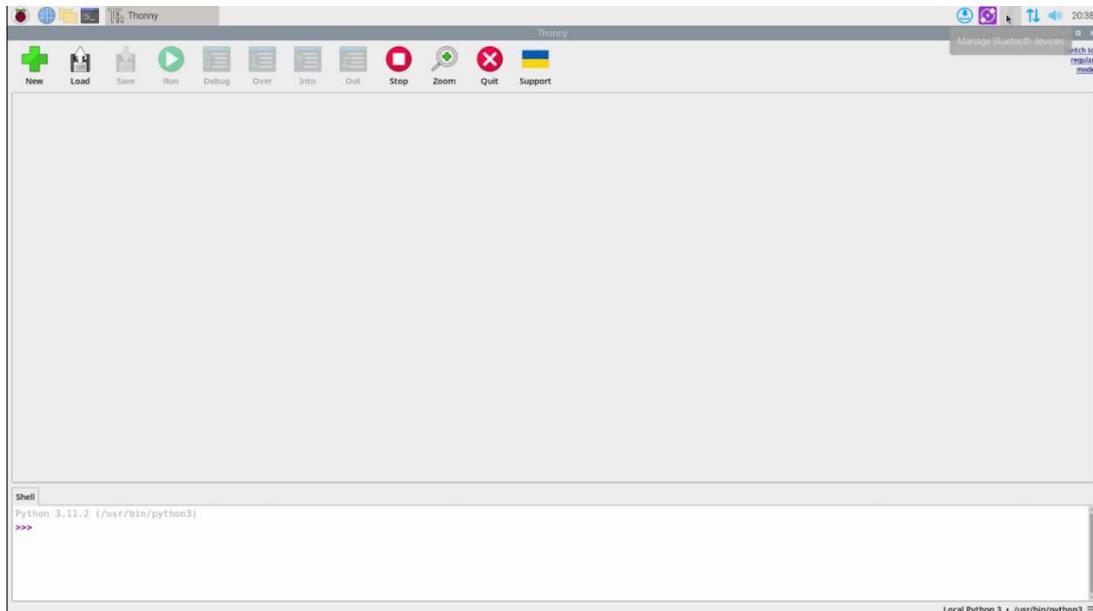


Рисунок 3.4 – Початок роботи у Thonny IDE

Перед запуском важливо правильно налаштувати апаратну частину, тому перемикачі на платі потрібно встановити у робочі положення, щоб Raspberry Pi міг керувати драйверами та отримувати сигнали від сенсорів, після чого вже можна створювати новий файл у форматі .py (рис. 3.5).

Для дослідного зразку використовується спрощений програмний код описаний у попередньому розділі, який слугує мозком системи, реалізуючи цикл реагування на оточення. На початку коду відбувається ініціалізація трьох ключових об'єктів за допомогою gpiozero, а саме AngularServo на GPIO 17 для повороту, DistanceSensor на GPIO 14 і 15 для вимірювання, та чотири OutputDevice на GPIO 19, 13, 6, 5 для керування кроковим двигуном через вбудований драйвер плати Freenove.

Функції `moveOnePeriod` і `moveSteps` створюють послідовність імпульсів, необхідну для обертання крокового двигуна на задану кількість кроків, з контрольованою затримкою, у мілісекундах, для управління швидкістю.

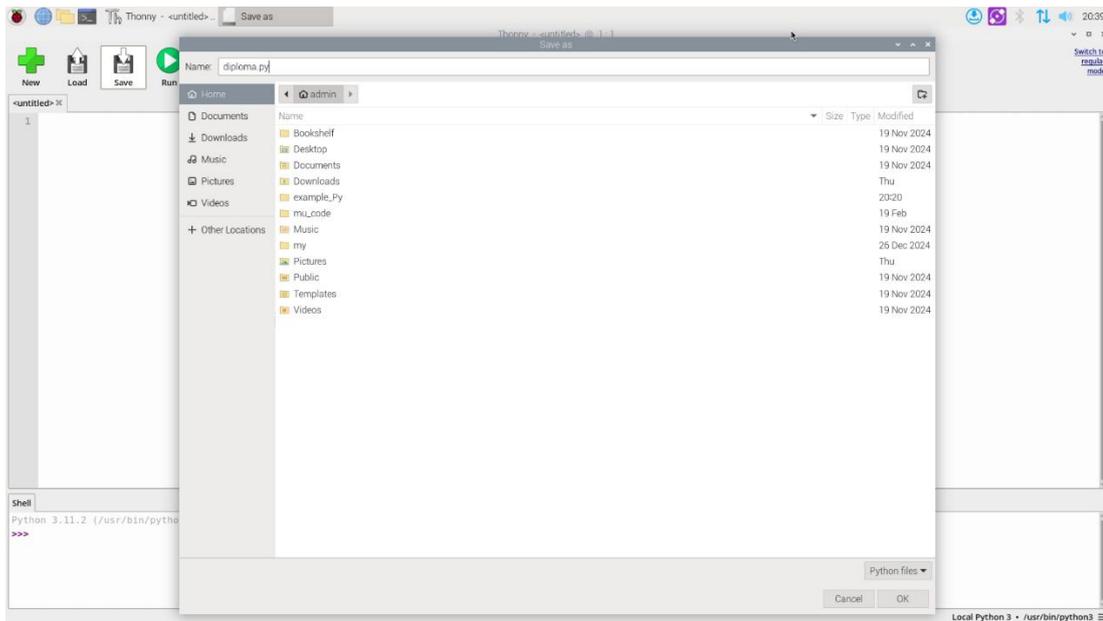


Рисунок 3.5 – Вікно створення нового файлу для роботи з програмним кодом

Вся логіка поведінки зосереджена у нескінченному циклі, під час якого на початку вимірюється відстань до об'єкта, якщо вона менша за 20 см, сервомотор негайно повертається на 45° , в нашому випадку, імітуючи зміну напрямку руху або огляду. Після цього кроковий двигун робить фіксований рух на 128 кроків, та по завершенні руху сервомотор повертається у початкове положення 90° . В кінці-ж коду написана спеціальна функція `destroy()`, яка забезпечує коректне завершення роботи, звільняючи усі GPIO-піни та зупиняючи мотори [18].

Налагодження системи на платі, фокусується на програмних параметрах та апаратних перемикачах, оскільки проблеми з проводкою практично виключені. Але також може бути присутня одна з найпоширеніших причин відсутності реакції - це неправильне положення функціональних перемикачів на платі, які можуть блокувати керуючі сигнали. Якщо, до прикладу, кроковий двигун не рухається, варто перевірити перемикач що активує його на поточній ділянці макету.

Якщо-ж проблема виникає з вібрацією сервомотора або-ж того самого крокового двигуна, де вони не рухаються плавно, то це зазвичай вказує на програмну помилку в таймінгу, при якій потрібно збільшити значення затримки у функції `moveOnePeriod`, щоб знизити швидкість руху.

```

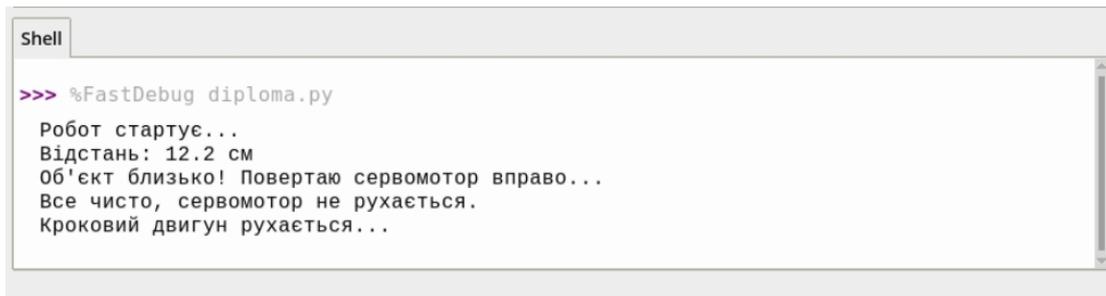
1 from gpiozero import AngularServo, OutputDevice, DistanceSensor
2 from time import sleep
3 import warnings
4 import random
5
6 warnings.filterwarnings("ignore")
7
8 # -----
9 # Налаштування сервомоторів і крокового двигуна
10 # -----
11
12
13 # Серво для повороту
14 servo = AngularServo(17, min_angle=0, max_angle=180, min_pulse_width=0.0005, max_pulse_width=0.0025)
15 initial_angle = 99
16 turn_angle = 45
17
18
19 # Кроковий двигун (4 піни)
20 rear_motor_pins = (19, 13, 6, 5)
21 rear_motors = [OutputDevice(pin) for pin in rear_motor_pins]
22
23 # Послідовності кроків
24 CWStep = (0x01, 0x02, 0x04, 0x08)
25 CCWStep = (0x08, 0x04, 0x02, 0x01)
26
27 # Ультразвуковий сенсор
28 trigPin = 14
29 echoPin = 15
30 sensor = DistanceSensor(echo=echoPin, trigger=trigPin, max_distance=3)
31
32 # -----
33 # Функції для крокового двигуна
34 # -----

```

Рисунок 3.6 – Імплементований програмний код

Також можуть виникати проблеми з ультразвуковим сенсором, що видає некоректні дані. На платі, вони ймовірно усього, пов'язані з його неправильним підключенням до 4-пінового роз'єму (неправильне співвідношення Trigger/Echo) або-ж з програмним шумом, який потрібно усувати шляхом усереднення вимірювань в самому коді [20].

Запуск програми слугує інтегрованим тестом, де перед початком слід переконатися, що функціональні перемикачі на платі Freenove встановлені правильно (рис. 3.7). Одразу-ж, сервомотор повинен одразу зайняти положення 90° , після цього, ультразвуковий сенсор має виводити у консоль значення відстані, підтверджуючі коректну роботу вбудованої схеми [21]. При піднесенні будь-якого об'єкта ближче ані-ж 20 см, сервомотор повинен чітко та негайно повернутися на 45° , підтверджуючи роботу логіки. Слідом за цим, кроковий двигун повинен виконати 128 кроків. Якщо будь-який із цих кроків не виконується, це вказує або на неправильне положення перемикача на платі, або на некоректне підключення самого мотора/сенсора до його роз'єму, як це вже було попередньо описано.



```
Shell
>>> %FastDebug diploma.py
Робот стартує...
Відстань: 12.2 см
Об'єкт близько! Повертаю сервомотор вправо...
Все чисто, сервомотор не рухається.
Кроковий двигун рухається...
```

Рисунок 3.7 – Успішний результат запуску програмного коду

Після запуску програмного забезпечення в консоль виводяться послідовні інформаційні повідомлення, які підтверджують коректну роботу всіх модулів дослідного зразка. Отримані дані свідчать про стабільне функціонування сенсорів, виконавчих механізмів та логічних функцій, що у свою чергу, дозволяє оцінити працездатність системи та забезпечити контроль за поточним станом пристрою під час тестування.

3.3 Тестування, калібрування та оцінювання роботи системи

Перевірка точності ультразвукового сенсора виконується шляхом порівняння його програмних результатів із реальними, перевіреними, вимірюваннями за допомогою лінійки. Завдяки платі Freenove, можна бути впевненим, що сигнал, який надходить до GPIO 15 (Echo), є правильним і захищеним від електричних перешкод, що мінімізує випадкові похибки даних.

Об'єкт розміщується на відомій нам відстані, після чого при запуску сенсора, перевіряється результат, чи він співпадає з відомим значенням відстані (рис. 3.8). Якщо виявлено систематичну похибку, як наприклад, постійне заниження або завищення показань, її слід компенсувати безпосередньо у кодї, додавши або віднявши недостаюче, або надмірне значення. Надійна точність є критично важливою, оскільки саме вона визначає, чи спрацює логіка маневрування на ключовому порозі 20 см [1].



Рисунок 3.8 – Перевірка точності сенсору за допомогою лінійки та перешкоди

```

Shell
>>> %FastDebug diploma.py
Робот стартує...
Відстань: 21.7 см
Все чисто, сервомотор не рухається.
Кроковий двигун рухається...

```

Рисунок 3.9 – Результати сенсору при порівнянні

У ході загального порівняльного аналізу було зафіксовано незначне відхилення вимірювань, яке становить приблизно +1.7 см (рис. 3.9). Дана похибка не впливає на функціональність та достовірність роботи саме нашого дослідницького зразка, а тому може бути визнана некритичною. Враховуючи це, система вважається працездатною, і можна переходити до наступного етапу роботи з макетом.

Етап повного запуску перевіряє всю логіку, від системи вимірювання, і до виконання фізичних функцій. Під час цього виділяється поведінка у двох основних режимах, вільний, та режим перешкоди. У першому режимі кроковий двигун має рухатися рівномірно, а сервомотор залишатися на 90° . У другому – ж режимі, коли об'єкт на відстані менше 20см, сервомотор повинен швидко повернутися на 45° (рис. 3.10), після чого кроковий двигун повинен виконати свій рух [15]. Ключовим моментом є синхронізація оберту сервомотору з

завершенням роботи крокового двигуна, що не має призводити до зіткнення. Успішне проходження цієї перевірки підтверджує, що програмна логіка ефективно керує апаратними ресурсами плати.



Рисунок 3.10 – Рух сервомотору на 45°

Після успішного запуску та підтвердження стабільної роботи апаратної частини на платі, стає можливим перейти до тонкого налаштування програмного забезпечення для підвищення ефективності роботи системи. Можливий аналіз результатів спрямований передусім на оптимізацію логіки руху й підвищення чутливості реагування.

Якщо система демонструє повільну реакцію або наближається до перешкод на надто малій дистанції, можна внести декілька корисних змін у програмний код. По-перше, поріг спрацьовування сенсора може бути недостатнім - його доцільно збільшити до 30 - 40см, щоб забезпечити більше часу на маневр. По-друге, кількість кроків крокового двигуна за цикл, який = 128, варто зменшити, наприклад, до 64, щоб частіше оновлювати дані про довколишнє середовище. По-третє, у разі нестабільного руху необхідно збільшити затримку між кроками двигуна, що забезпечить плавніший рух системи [7].

Такі корегування програмного коду дозволяють адаптувати логіку системи під реальні умови роботи, досягаючи оптимального балансу між швидкістю, точністю та надійністю маневрування [17].

3.4 Висновки до розділу

Цей розділ експериментальної роботи не просто підтвердив працездатність окремих компонентів, але й довів життєздатність і можливість повноцінної реалізації мобільної моделі, описаної в попередніх розділах. Використання плати Freenove Projects Board як інтегрованої апаратної основи значно спростило технічну реалізацію, забезпечивши надійну взаємодію між Raspberry Pi 5, ультразвуковим сенсором, сервомотором та кроковим двигуном.

Програмна логіка, успішно реалізована за допомогою gpiozero, довела, що система може ефективно збирати дані, приймати рішення та виконувати дії у циклі, а також успішне тестування показало, що закладені механізми як поворот сервоприводу, дозволяють реалізувати базові функції уникнення перешкод. Таким чином, цей експериментальний зразок є повністю функціональним прототипом і слугує прямим доказом того, що концептуальна модель, описана в теоретичній частині, може бути перетворена на працюючу фізичну систему.

4 ЕКОНОМІЧНА ЧАСТИНА

Науково-технічні розробки можуть знаходити практичне застосування лише тоді, коли вони повністю відповідають актуальним вимогам сучасного етапу науково-технічного прогресу, а також демонструють реальну економічну доцільність свого впровадження. Іншими словами, будь-яка інновація повинна бути не лише технічно обґрунтованою, а й фінансово вигідною для підприємства чи галузі.

Саме тому оцінювання економічної ефективності результатів науково-дослідної роботи є обов'язковою та невід'ємною складовою процесу її впровадження, адже дозволяє визначити можливі вигоди, окупність та перспективність застосування отриманих результатів.

Нині існує багато методик, за якими можна виконувати оцінювання ефективності науково-дослідницьких робіт (НДР). Під час виконання магістерської кваліфікаційної роботи на тему «Інформаційно – вимірвальна система визначення перешкод та їх оминання», було обрано напрям виконання науково-технічних робіт, які з самого початку призначаються для виведення на ринок (або рішення про виведення науковотехнічної розробки на ринок може бути прийнято у процесі проведення самої роботи), тобто коли відбувається так звана комерціалізація науковотехнічної розробки.

Цей напрям є пріоритетним, оскільки результатами розробки можуть користуватися інші споживачі, отримуючи при цьому певний економічний ефект. Але для цього потрібно знайти потенційного інвестора, який би взявся за реалізацію цього проекту і переконати його в економічній доцільності такого кроку.

Для цього випадку мають бути виконані такі етапи робіт:

- а) проведення комерційного аудиту науково-технічної розробки, тобто встановлення її науково-технічного рівня та комерційного потенціалу;
- б) розрахунок витрат на здійснення науково-технічної розробки;

в) розрахунок економічної ефективності науково-технічної розробки у випадку її впровадження та комерціалізації потенційним інвестором і обґрунтування економічної доцільності комерціалізації потенційним інвестором розробленої у магістерській кваліфікаційній роботі науково-технічної розробки.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Метою проведення комерційного і технологічного аудиту дослідження за темою «Інформаційно - вимірвальна система визначення перешкод та їх оминання» є оцінювання науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності, тобто під час виконання магістерської кваліфікаційної роботи.

Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням 5-ти бальної системи оцінювання за 12-ма критеріями, що наведено в таблиці 4.1 [22].

Таблиця 4.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

Критерії	Експерт (ПІБ, посада)		
	1	2	3
	Бали:		
1. Технічна здійсненність концепції	3	3	4
2. Ринкові переваги (наявність аналогів)	2	2	3
3. Ринкові переваги (ціна продукту)	3	2	3
4. Ринкові переваги (технічні властивості)	3	3	4
5. Ринкові переваги (експлуатаційні витрати)	3	3	3
6. Ринкові перспективи (розмір ринку)	2	3	2
7. Ринкові перспективи (конкуренція)	2	2	3

Продовження таблиці 4.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки експертами

8. Практична здійсненність (наявність фахівців)	3	4	3
9. Практична здійсненність (наявність фінансів)	2	3	3
10. Практична здійсненність (необхідність нових матеріалів)	4	4	4
11. Практична здійсненність (термін реалізації)	3	3	3
12. Практична здійсненність (розробка документів)	4	4	4
Сума балів	34	36	39
Середньоарифметична сума балів CB_c	36,3		

За результатами розрахунків, наведених в таблиці 4.1, зробимо висновок щодо науково-технічного рівня і рівня комерційного потенціалу розробки. При цьому використаємо рекомендації, наведені в [22].

Згідно проведених досліджень рівень комерційного потенціалу розробки за темою «Інформаційно - вимірювальна система визначення перешкод та їх оминання» становить 36,3 бала, що, відповідно до [22], свідчить про комерційну важливість проведення даних досліджень (рівень комерційного потенціалу розробки вище середнього).

4.2 Розрахунок витрат на проведення науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи на тему «Інформаційно - вимірювальна система визначення перешкод та їх оминання», під час планування, обліку і калькулювання собівартості науково-дослідної роботи групуємо за відповідними статтями.

4.2.1 Витрати на оплату праці

Основна заробітна плата дослідників

Витрати на основну заробітну плату дослідників (Z_o) розраховуємо у відповідності до посадових окладів працівників, за формулою 4.2 [22].

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.2)$$

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.

$$Z_o = 20000,00 \cdot 10 / 22 = 9090,91 \text{ (грн.)}$$

Проведені розрахунки зведемо до таблиці 4.2.

Таблиця 4.2 – Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн	Число днів роботи	Витрати на заробітну плату, грн
Науковий керівник	20000,00	909,09	10	9090,91
Виконавець (аспірант)	12000,00	545,45	30	16363,64
Всього				25454,55

Основна заробітна плата робітників

Витрати на основну заробітну плату робітників (Z_p) за відповідними найменуваннями робіт НДР на тему «Інформаційно - вимірювальна система визначення перешкод та їх оминання» розраховуємо за формулою 4.3.

$$Z_p = \sum_{i=1}^n C_i \cdot t_i, \quad (4.3)$$

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 4.4.

$$C_i = \frac{M_M * K_i * K_c}{T_p * t_{зм}}, \quad (4.4)$$

де M_M – розмір мінімальної місячної заробітної плати, прийmemo $M_M = 8000,00$ (грн.);

K_i – коефіцієнт міжкваліфікаційного співвідношення [22];

K_c – мінімальний коефіцієнт співвідношень місячних тарифних ставок;

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

$t_{зм}$ – тривалість зміни, год.

$$C_j = 8000,00 \cdot 1,50 \cdot 1,5 / (22 \cdot 8) = 102,27 \text{ (грн.)}$$

$$Z_{pl} = 102,27 \cdot 3,20 = 327,26 \text{ (грн.)}$$

Проведені розрахунки зведемо до таблиці 4.3.

Таблиця 4.3 – 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
Апаратна реалізація та збирання дослідного зразка рухомої платформи	3,20	4	1,50	102,27	327,26
Розробка та інтеграція програмного забезпечення керування системою	10,00	3	1,35	92,04	920,40
Тестування моделі рухомої платформи	7,00	4	1,50	102,27	715,89
Формування бази даних результатів вимірювань	5,00	3	1,35	92,04	460,20
Всього					2423,75

Додаткова заробітна плата дослідників та робітників

Додаткову заробітну плату розраховуємо як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою 4.5.

$$Z_{\text{дод}} = (Z_o + Z_p) \cdot \frac{H_{\text{дод}}}{100\%}, \quad (4.5)$$

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

$$Z_{\text{дод}} = (25454,55 + 2423,75) \cdot 11 / 100\% = 3066,61 \text{ (грн.)}$$

4.2.2 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 4.6.

$$З_n = (З_o + З_p + З_{дод}) \cdot \frac{H_{zn}}{100\%}, \quad (4.6)$$

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

$$З_n = (25454,55 + 2423,75 + 3066,61) \cdot 22 / 100\% = 6807,88 \text{ (грн.)}$$

4.2.3 Сировина та матеріали

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою 4.7.

$$M = \sum_{j=1}^n H_j \cdot Ц_j \cdot K_j - \sum_{j=1}^n B_j \cdot Ц_{ej}, \quad (4.7)$$

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;

$Ц_{ej}$ – вартість відходів j -го найменування, (грн./кг).

$$M_1 = 2 \cdot 222,00 \cdot 1,1 - 0 \cdot 0 = 488,4 \text{ (грн.)}$$

Проведені розрахунки зведемо до таблиці 4.4.

Таблиця 4.4 – 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 Quedi Everyday 500/80	222,00	2	0	0	488,40
Канцелярське приладдя BIC-master	190,00	1	0	0	209,00
Всього					697,40

4.2.4 Розрахунок витрат на комплектуючі

Витрати на комплектуючі (K_6), які використовують при проведенні НДР на тему «Інформаційно - вимірювальна система визначення перешкод та їх оминання», розраховуємо, згідно з їхньою номенклатурою, за формулою 4.8.

$$K_6 = \sum_{j=1}^n H_j \cdot C_j \cdot K_j, \quad (4.8)$$

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

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

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

$$K_6 = 1 \cdot 120,00 \cdot 1,1 = 132,00 \text{ (грн.)}$$

Проведені розрахунки зведемо до таблиці 4.5.

Таблиця 4.5 – Витрати на комплектуючі

Найменування комплектуючих	Кількість, шт.	Ціна за штуку, грн	Сума, грн
НС-SR04 – ультразвуковий сенсор	1	120,00	132,00
Сервомотор	3	220,00	726,00
Кроковий мотор	2	800,00	1760,00
Плата – основа для апаратної реалізації	1	300,00	330,00
Рухомі колеса	4	200,00	880,00
PCB_V1.0 – плата підключення	1	150,00	165,00
PCB_V2.0 – плата підключення	1	200,00	220,00
Дріт - перемичка F/F(4)	1	50,00	55,00
ХН-2.54-5Pin - кабель	1	60,00	66,00
M1.4*4 – саморізний гвинт	12	5,00	66,00
M2.4*4 - гвинт	5	5,00	27,50
M3*6 - гвинт	5	10,00	55,00
M2*10 - гвинт	5	8,00	44,00
M3*14 - гвинт	4	12,00	52,80
M2 - гайка	5	5,00	27,50
M3 - гайка	4	8,00	35,20
Raspberry Pi 5 – мікрокомп'ютер	1	3883,00	4271,30
Varta Longlife Power AA - батарейка	2	159,00	349,80
Всього			9363,10

4.2.5 Спецустаткування для наукових (експериментальних) робіт

Витрати за статтею «Спецустаткування для наукових (експериментальних) робіт» відсутні.

4.2.6 Програмне забезпечення для наукових (експериментальних) робіт

Витрати за статтею «Програмне забезпечення для наукових (експериментальних) робіт» відсутні.

4.2.7 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 4.9 [22].

$$A_{обл} = \frac{Ц_б}{T_г} \cdot \frac{t_{вик}}{12}, \quad (4.9)$$

де $Ц_б$ – балансова вартість обладнання, програмних засобів, приміщень тощо, які використовувались для проведення досліджень, (грн.);

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

$T_г$ – строк корисного використання обладнання, програмних засобів, приміщень тощо, років.

$$A_{обл} = (5300,00 \cdot 1) / (3 \cdot 12) = 147,22 \text{ (грн.)}$$

Проведені розрахунки зведемо до таблиці 4.6.

Таблиця 4.6 – Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Windows 11 Home	5300,00	3	1	147,22
Microsoft Office 2021 Home & Student (Одноразова покупка)	3500,00	3	1	97,22
Ноутбук Dell G5 5590	29000,00	3	1	805,56
Приміщення лабораторії досліджень	450000,00	30	1	1250,00
Робоче місце інженера-розробника	8250,00	5	1	137,50
Всього				2437,50

4.2.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховуємо за формулою 4.10.

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{внi}}{\eta_i}, \quad (4.10)$$

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.); прийmemo $C_e = 10,98$ (грн.);

$K_{\text{внi}}$ – коефіцієнт, що враховує використання потужності, $K_{\text{внi}} < 1$;

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

$$B_e = 0,10 \cdot 190,0 \cdot 10,98 \cdot 0,95 / 0,97 = 204,34 \text{ (грн.)}$$

Проведені розрахунки зведемо до таблиці 4.7.

Таблиця 4.7 – Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Ноутбук Dell G5 5590	0,10	190,00	204,34
Робоче місце інженера - розробника	0,20	190,00	408,43
Всього			612,77

4.2.9 Службові відрядження

Витрати за статтею «Службові відрядження» відсутні.

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

Витрати на роботи, які виконують сторонні підприємства, установи і організації відсутні.

4.2.11 Інші витрати

Витрати за статтею «Інші витрати» розраховуємо як 50...100% від суми основної заробітної плати дослідників та робітників за формулою 4.11 [22].

$$I_{\text{в}} = (Z_o + Z_p) \cdot \frac{H_{\text{ив}}}{100\%}, \quad (4.11)$$

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

$$I_{\text{в}} = (25454,55 + 2423,75) \cdot 50 / 100\% = 13939,15 \text{ (грн.)}$$

4.2.12 Накладні (загальновиробничі) витрати

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуємо як 100...150% від суми основної заробітної плати дослідників та робітників за формулою 4.12.

$$B_{\text{нзв}} = (Z_o + Z_p) \cdot \frac{H_{\text{нзв}}}{100\%}, \quad (4.12)$$

де $H_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати», прийmemo $H_{\text{нзв}} = 110\%$.

$$B_{\text{нзв}} = (25454,55 + 2423,75) \cdot 110 / 100\% = 30666,13 \text{ (грн.)}$$

Витрати на проведення науково-дослідної роботи на тему «Інформаційно - вимірювальна система визначення перешкод та їх оминання» розраховуємо як суму всіх попередніх статей витрат за формулою 4.13.

$$B_{\text{заг}} = Z_o + Z_p + Z_{\text{од}} + Z_n + M + K_{\text{в}} + B_{\text{спец}} + B_{\text{прг}} + \\ + A_{\text{обл}} + B_e + B_{\text{св}} + B_{\text{сн}} + I_{\text{в}} + B_{\text{нзв}}, \quad (4.13)$$

$$B_{\text{заг}} = 25454,55 + 2423,75 + 3066,61 + 6807,88 + 697,40 + 9363,10 + 0,00 + 0,00 \\ + 2437,50 + 612,77 + 0,00 + 0,00 + 13939,15 + 30666,13 = 95468,84 \text{ (грн.)}$$

Загальні витрати ZB на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховується за формулою 4.14.

$$ZB = \frac{B_{заг}}{\eta}, \quad (4.14)$$

де η - коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи, прийmemo $\eta = 0,5$.

$$ZB = 95468,84 / 0,5 = 190937,68 \text{ (грн.)}$$

4.3 Оцінювання важливості та наукової значимості науководослідної роботи фундаментального чи пошукового характеру

Для обґрунтування доцільності виконання науково-дослідної роботи використовуємо спеціальний комплексний показник, що враховує важливість, результативність роботи, можливість впровадження її результатів у виробництво, величину витрат на роботу [22].

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

$$K_p = \frac{I^n * T_c * R}{B * t}, \quad (4.15)$$

де I – коефіцієнт важливості роботи, $I = 2...5$;

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;

T_C – коефіцієнт складності роботи, $T_C = 1...3$;

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 $R = 1$;

B – вартість науково-дослідної роботи, (тис. грн.);

t – час проведення дослідження, років.

$$K_p = \frac{2^3 * 2 * 3}{101,2 * 0,3} = 1,5$$

Оскільки $K_p = 1,5 > 1$, науково-дослідна робота є ефективною, має достатній рівень науково-технічної результативності та обґрунтовану практичну цінність при помірних витратах.

4.4 Розрахунок економічної ефективності науково-технічної розробки при її можливій комерціалізації потенційним інвестором

Результати дослідження проведені за темою «Інформаційно - вимірювальна система визначення перешкод та їх оминання» передбачають комерціалізацію протягом 4-х років реалізації на ринку і відповідають практичному застосуванню за напрямом - розробка чи суттєве вдосконалення машини (механізму, приладу, пристрою) для використання кінцевими споживачами [22].

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних, що наведено в таблиці 4.8

ΔN – збільшення кількості споживачів пристрою, у періоди часу, що аналізуються, від покращення його певних характеристик;

Таблиця 4.8 – Показники економічного ефекту

Показник	1-й рік	2-й рік	3-й рік	4-й рік
Збільшення кількості споживачів, осіб	250	500	600	350

N – кількість споживачів які використовували аналогічний пристрій у році до впровадження результатів нової науково-технічної розробки, прийmemo 10000 осіб;

C_o – вартість пристрою у році до впровадження результатів розробки, прийmemo 10000,00 (грн.);

$\pm \Delta C_o$ – зміна вартості пристрою від впровадження результатів науково-технічної розробки, прийmemo 800,00 (грн.)

Можливе збільшення чистого прибутку у потенційного інвестора $\Delta \Pi_i$ для кожного із 4-х років, протягом яких очікується отримання позитивних результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховуємо за формулою 4.16 [22].

$$\Delta \Pi_i = (\pm \Delta C_o \cdot N + C_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (4.16)$$

де λ – коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2025 році ставка податку на додану вартість складає 20%, а коефіцієнт $\lambda = 0,8333$;

ρ – коефіцієнт, який враховує рентабельність інноваційного продукту).
Прийmemo $\rho = 35\%$;

ϑ – ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2025 році $\vartheta = 18\%$;

Збільшення чистого прибутку 1-го року:

$$\begin{aligned} \Delta \Pi_1 &= (800,00 * 10000,00 + 10800,00 * 250) * 0,83 * 0,35 * \\ &* (1 - 0,18/100\%) = 3102754,97 \text{ (грн.)} \end{aligned}$$

Збільшення чистого прибутку 2-го року:

$$\Delta\Pi_2 = (800,00 * 10000,00 + 10800,00 * 750) * 0,83 * 0,35 * \\ * (1 - 0,18/100\%) = 4668631,31 \text{ (грн.)}$$

Збільшення чистого прибутку 3-го року:

$$\Delta\Pi_3 = (800,00 * 10000,00 + 10800,00 * 1350) * 0,83 * 0,35 * \\ * (1 - 0,18/100\%) = 6551176,62 \text{ (грн.)}$$

Збільшення чистого прибутку 4-го року:

$$\Delta\Pi_4 = (800,00 * 10000,00 + 10800,00 * 1700) * 0,83 * 0,35 * \\ * (1 - 0,18/100\%) = 7643796,36 \text{ (грн.)}$$

Приведена вартість збільшення всіх чистих прибутків $\Pi\Pi$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки розраховується за формулою 4.17 [22].

$$\Pi\Pi = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.17)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-технічної розробки, (грн.);

T – період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-технічної розробки, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,15$;

t – період часу (в роках) від моменту початку впровадження науково-технічної розробки до моменту отримання потенційним інвестором додаткових чистих прибутків у цьому році.

$$\begin{aligned} III &= 3102754,97 / (1 + 0,15)^1 + 4668631,31 / (1 + 0,15)^2 + 6551176,62 / (1 + 0,15)^3 \\ &+ 7643796,36 / (1 + 0,15)^4 = 2698047,80 + 3530156,00 + 4307504,97 + 4370365,37 \\ &= 14906074,14 \text{ (грн.)} \end{aligned}$$

Величина початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки розраховується за формулою 4.18.

$$PV = k_{инв} \cdot ZB, \quad (4.18)$$

де $k_{инв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію, приймаємо $k_{инв} = 2$;

ZB – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, приймаємо 190937,68 (грн.).

$$PV = k_{инв} \cdot ZB = 2 \cdot 190937,68 = 381875,36 \text{ (грн.)}$$

Абсолютний економічний ефект $E_{абс}$ для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки розраховується за формулою 4.19 та становитиме:

$$E_{абс} = III - PV, \quad (4.19)$$

де III – приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, 14906074,14 (грн.);

PV – теперішня вартість початкових інвестицій, 381875,36 (грн.)

$$E_{абс} = 14906074,14 - 381875,36 = 14524198,78 \text{ (грн.)}$$

Внутрішня економічна дохідність інвестицій E_6 , які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки розраховується за формулою 4.20.

$$E_6 = \sqrt[T_{ж}]{\left(1 + \frac{E_{абс}}{PV}\right)} - 1, \quad (4.20)$$

де $E_{абс}$ – абсолютний економічний ефект вкладених інвестицій, 14501211,66 (грн.);

PV – теперішня вартість початкових інвестицій, 381875,36 (грн.);

$T_{ж}$ – життєвий цикл науково-технічної розробки, тобто час від початку її розробки до закінчення отримання позитивних результатів від її впровадження, 4 роки.

$$E_6 = \sqrt[4]{1 + \frac{E_{абс}}{PV}} - 1 = (1 + 14524198,78 / 381875,36)^{1/4} = 2,49.$$

Мінімальна внутрішня економічна дохідність вкладених інвестицій $\tau_{мін}$ розраховується за формулою 4.21.

$$\tau_{мін} = d + f, \quad (4.21)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2025 році в Україні $d = 0,12$;

f – показник, що характеризує ризикованість вкладення інвестицій, прийmemo 0,25.

$\tau_{мін} = 0,12 + 0,25 = 0,37 < 2,49$ свідчить про те, що внутрішня економічна дохідність інвестицій E_6 , вища мінімальної внутрішньої дохідності. Тобто інвестувати в науково-дослідну роботу за темою «Інформаційно - вимірвальна система визначення перешкод та їх оминання» доцільно [22].

Період окупності інвестицій $T_{ок}$ які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки розраховується за формулою 4.22.

$$T_{ок} = \frac{1}{E_6}, \quad (4.22)$$

де E_6 – внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = 1 / 2,49 = 0,4 \text{ р.}$$

$T_{ок} < 3$ -х років, що свідчить про комерційну привабливість науково-технічної розробки і може спонукати потенційного інвестора профінансувати впровадження даної розробки та виведення її на ринок.

4.5 Висновки до розділу

Згідно з результатами проведених досліджень, рівень комерційного потенціалу розробки за темою «Інформаційно-вимірювальна система визначення перешкод та їх оминання» становить 36,3 бала, що свідчить про високу комерційну значимість цієї науково-технічної розробки, оскільки рівень комерційного потенціалу перевищує середній. Це підтверджує, що проведення досліджень є актуальним і здатні забезпечити реальні економічні та практичні результати.

Крім того, термін окупності проекту складає 0,4 року, що значно менше порогового значення у 3 роки. Такий короткий період окупності свідчить про високу привабливість розробки з комерційної точки зору. Це означає, що потенційний інвестор має всі підстави для того, щоб профінансувати впровадження системи.

Таким чином, на основі проведеного аналізу можна зробити обґрунтований висновок про доцільність і важливість проведення науково-дослідної роботи за темою «Інформаційно-вимірювальна система визначення перешкод та їх оминання», адже вона поєднує значний комерційний потенціал із високою економічною ефективністю та перспективністю для впровадження у практичну діяльність.

ВИСНОВКИ

У результаті виконання магістерської роботи було створено повноцінну інформаційно-вимірювальну систему для визначення та оминання перешкод мобільною платформою, яка поєднала теоретичні засади сучасних сенсорних технологій з практичною реалізацією апаратно-програмного комплексу. Проведений у першому розділі аналіз існуючих сенсорних систем та підходів до побудови архітектур уникнення перешкод дозволив сформуванню комплексне уявлення про актуальний стан галузі, сильні та слабкі сторони доступних технологій, а також визначити оптимальні технічні рішення для побудови власної системи.

Розроблена в другому розділі структурна та функціональна модель системи ґрунтувалася на раціональному виборі ультразвукового сенсора як основного засобу вимірювання відстані, механізмів забезпечення руху та повороту платформи, а також детальному опрацюванні схеми взаємодії між апаратними модулями. Завдяки модульному підходу вдалося забезпечити гнучкість конструкції та перспективність її подальшої модернізації, включно із заміною сенсорів, розширенням функціоналу або інтеграцією більш складних алгоритмів обробки даних.

Під час моделювання та практичної реалізації, розглянутих у третьому розділі, було проведено налагодження апаратної частини, розробку програмного забезпечення, а також комплексне тестування. Система продемонструвала стабільну роботу, високу точність визначення відстані та надійність алгоритмів ухилення від перешкод. Підбір параметрів сервоприводів, оптимізація логіки руху та калібрування сенсорів забезпечили швидку реакцію платформи на зміну навколишніх умов, що стало важливою передумовою підвищення ефективності та енергоощадності всієї системи.

Особливої уваги заслуговує програмна частина, яка була розроблена з урахуванням можливості масштабування та врахування майбутніх оновлень. Алгоритмічна логіка дозволяє оптимізувати траєкторію руху у режимі реального

часу, забезпечує багатоваріантність прийняття рішень та створює основу для подальшого впровадження елементів машинного навчання або більш складних систем навігації.

Економічна частина роботи, представлена у четвертому розділі, підтвердила доцільність проведених досліджень та раціональність обраних технічних рішень. Проведений науковий, комерційний та технологічний аудит дозволив комплексно оцінити рівень наукової новизни, практичну значущість та технічну реалізованість розробки.

Розрахунок витрат на виконання науково-дослідної роботи дав можливість визначити структуру та обсяг необхідних ресурсів, а також оцінити економічні параметри створеного апаратно-програмного комплексу. Аналіз економічної ефективності продемонстрував, що використані технічні компоненти та програмні рішення є обґрунтованими з погляду вартості, а загальна модель проекту відповідає принципам оптимального ресурсного планування.

Отримані результати підтверджують ефективність запропонованої архітектури, правильність вибору технічних засобів та доцільність розробленої програмної логіки. Система повністю відповідає поставленій меті та виконаним завданням, демонструючи високу якість реалізації та перспективність подальшого розвитку. Вона може бути використана як основа для створення автономних робототехнічних пристроїв, навчальних та дослідницьких комплексів у сфері інтелектуальних систем керування, а також як прототип для промислових рішень із навігації та моніторингу середовища.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Liu, Y., Wang, S., Xie, Y., Xiong, T., & Wu, M. «A Review of Sensing Technologies for Indoor Autonomous Mobile Robots». *Sensors*, 2024, 24(4), 1222.
- 2 Huang, J.; Junginger, S.; Liu, H.; Thurow, K. Indoor Positioning Systems of Mobile Robots: A Review. *Robotics* 2023, 12, 47.
- 3 Zeng, H.; Song, X.; Jiang, S. Multi-Object Navigation Using Potential Target Position Policy Function. *IEEE Trans. Image Process.* 2023, 32, 2608–2619.
- 4 Qin, H.; Shao, S.; Wang, T.; Yu, X.; Jiang, Y.; Cao, Z. Review of Autonomous Path Planning Algorithms for Mobile Robots. *Drones* 2023, 7, 211.
- 5 Zhou, H.; Yao, Z.; Zhang, Z.; Liu, P.; Lu, M. An Online Multi-Robot SLAM System Based on Lidar/UWB Fusion. *IEEE Sens. J.* 2022, 22, 2530–2542.
- 6 Xu, W.; Cai, Y.; He, D.; Lin, J.; Zhang, F. FAST-LIO2: Fast Direct LiDAR-Inertial Odometry. *IEEE Trans. Robot.* 2022, 38, 2053–2073.
- 7 Krishnamurthi, R., Kumar, A., Gopinathan, D., Nayyar, A., & Qureshi, B. «An Overview of IoT Sensor Data Processing, Fusion, and Analysis Techniques». *Sensors (Basel)*, 2020, 20(21), 6076.
- 8 Katona, K., Neamah, H. A., & Korondi, P. «Obstacle Avoidance and Path Planning Methods for Autonomous Navigation of Mobile Robot». *Sensors*, 2024, 24(11), 3573.
- 9 Zhai, S.; Pei, Y. The Dynamic Path Planning of Autonomous Vehicles on Icy and Snowy Roads Based on an Improved Artificial Potential Field. *Sustainability* 2023, 15, 15377.
- 10 Wang, X.; Yin, Y.; Jing, Q. Maritime Search Path Planning Method of an Unmanned Surface Vehicle Based on an Improved Bug Algorithm. *J. Mar. Sci. Eng.* 2023, 11, 2320.
- 11 Houshyari, H.; Sezer, V. A new gap-based obstacle avoidance approach: Follow the obstacle circle method. *Robotica* 2022, 40, 2231–2254.

12 Schoener, M.; Coyle, E.; Thompson, D. An anytime Visibility–Voronoi graph-search algorithm for generating robust and feasible unmanned surface vehicle paths. *Auton. Robot.* 2022, 46, 911–927.

13 Jang, D.u.; Kim, J.s. Development of Ship Route-Planning Algorithm Based on Rapidly-Exploring Random Tree (RRT*) Using Designated Space. *J. Mar. Sci. Eng.* 2022, 10, 1800.

14 Luo, S.; Zhang, M.; Zhuang, Y.; Ma, C.; Li, Q. A survey of path planning of industrial robots based on rapidly exploring random trees. *Front. Neurorobot.* 2023, 17, 1268447.

15 Rafai, A.N.A.; Adzhar, N.; Jaini, N.I. A review on path planning and obstacle avoidance algorithms for autonomous mobile robots. *J. Robot.* 2022, 2022, 2538220.

16 Sousa, L. C.; Silva, Y. M. R.; Schettino, V. B.; Santos, T. M. B.; Zachi, A. R. L.; Gouvêa, J. A.; Pinto, M. F. Obstacle Avoidance Technique for Mobile Robots at Autonomous Human-Robot Collaborative Warehouse Environments. *Sensors*, 2025, 25(8), 2387.

17 Wang, S.; Liu, Q.; Zhang, J.; Huang, F. Autonomous Navigation by Mobile Robot with Sensor Fusion Based on Deep Reinforcement Learning. *Sensors*, 2024, 24, 3895.

18 Misir, O.; Koc, A.; Demir, B. Visual-Based Obstacle Avoidance Method Using Advanced Deep Learning Techniques. *Engineering Applications of Artificial Intelligence*, 2025, 134, 107123.

19 Chen, Z.; Li, Y.; Hu, M.; Zhao, K. A Comprehensive Review of Obstacle Avoidance for Autonomous Systems: Sensor Fusion, LiDAR, Cameras, and Beyond. *Artificial Intelligence in Engineering*, 2025, 7, 100215.

20 Rani, B. S.; Kumar, S.; Prasad, V. R. An Innovative Approach for Obstacle Avoidance and Path Planning of Mobile Robots Using Reinforcement Learning. *Knowledge-Based Systems*, 2025, 293, 111037.

21 Zhang, L.; Xu, H.; Zhao, P. Intelligent Obstacle Avoidance Algorithm for Mobile Robots Based on Improved Fuzzy Neural Network. Computational Intelligence and Neuroscience, 2022, 2022, 8954060.

22 Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад. : В. О. Козловський, О. Й. Лесько, В. В. Кавецький. – Вінниця : ВНТУ, 2021. – 42 с

Додаток А
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНО - ВИМІРЮВАЛЬНА СИСТЕМА ВИЗНАЧЕННЯ ПЕРЕШКОД ТА ЇХ ОМИНАННЯ

Виконав: студент 2-го курсу, групи КІВТ-24м
спеціальності 175 – Інформаційно - вимірювальні
технології

(шифр і назва напрямку підготовки, спеціальності)

 Гайдамашко А.О.
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. ІРТС

 Притула М.О.
(прізвище та ініціали)

«15» 12 2025 р.

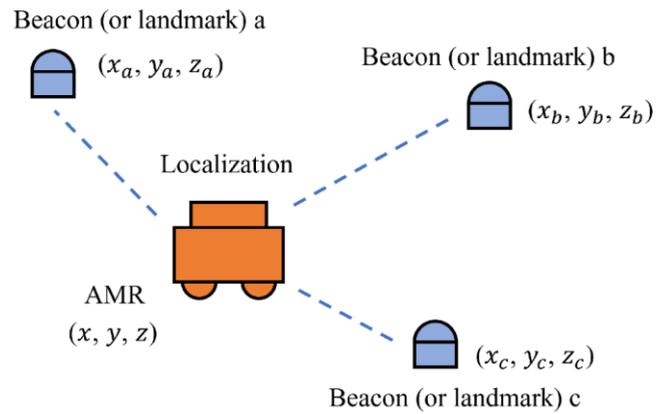


Рисунок 1 - Схематична діаграма методу на основі активного маяка

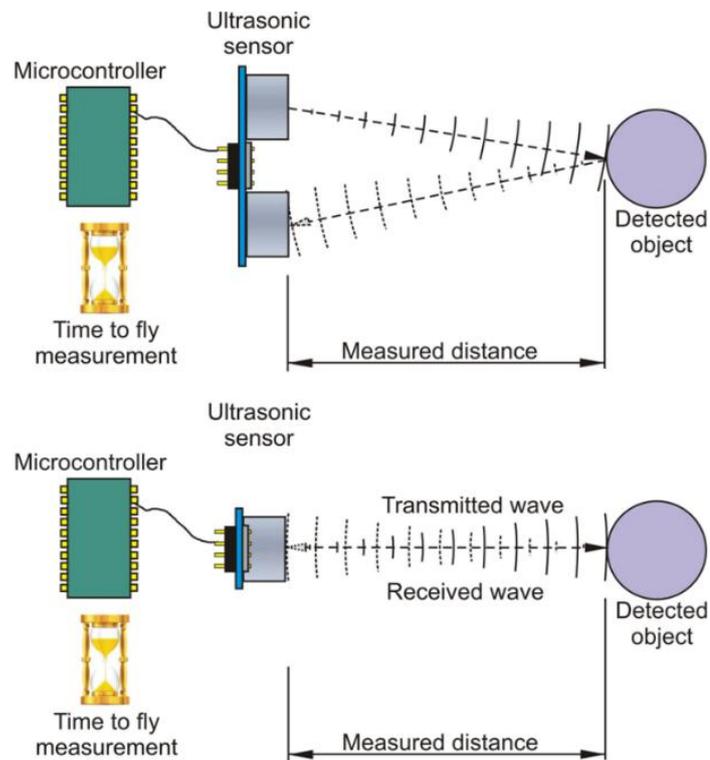


Рисунок 2 - Методи вимірювання відстані сенсором: ехолокаційний метод та односпрямований метод

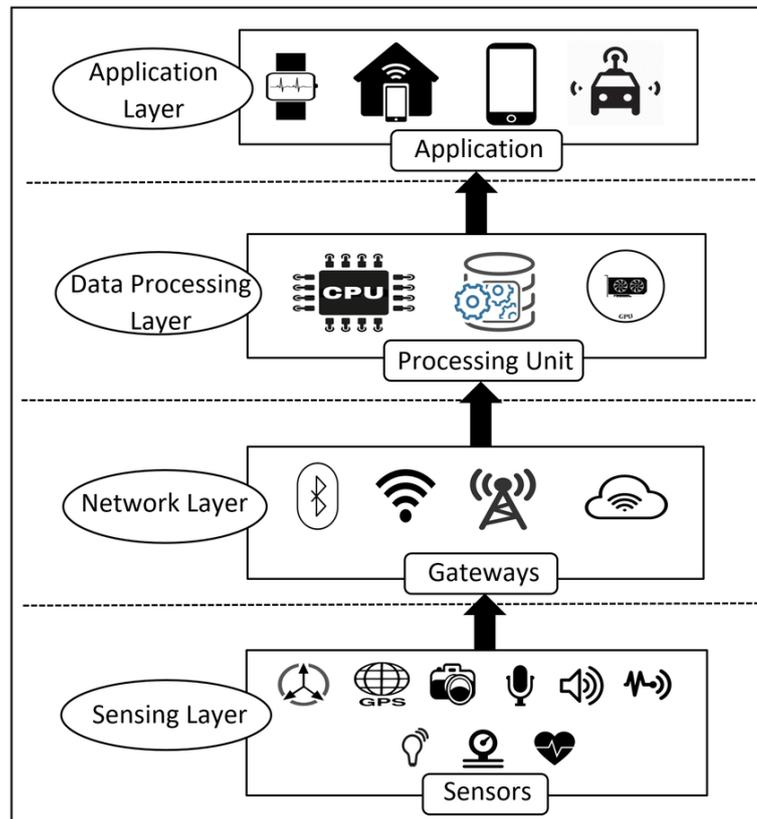


Рисунок 3 – Схематичне зображення шарів базової архітектури

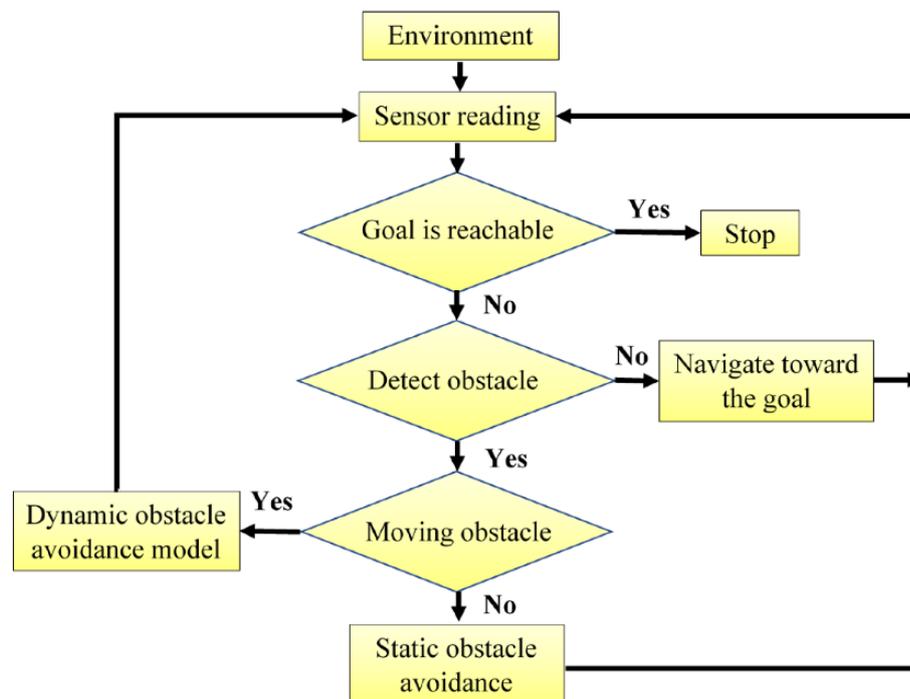


Рисунок 4 – Процедура оминання перешкод

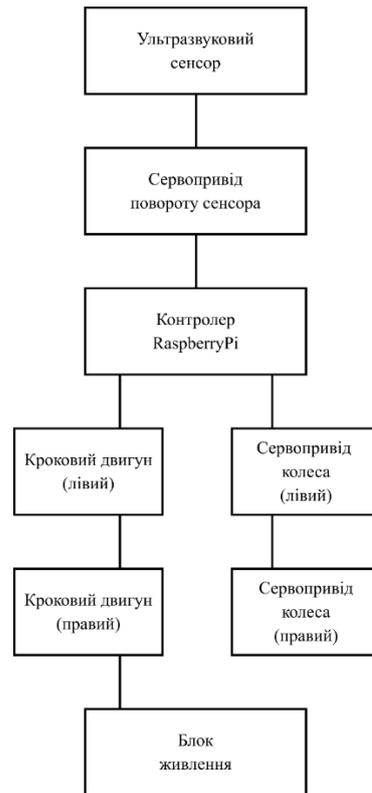


Рисунок 5 – Спрощена структурна схема логічних взаємозв'язків рухомої платформи

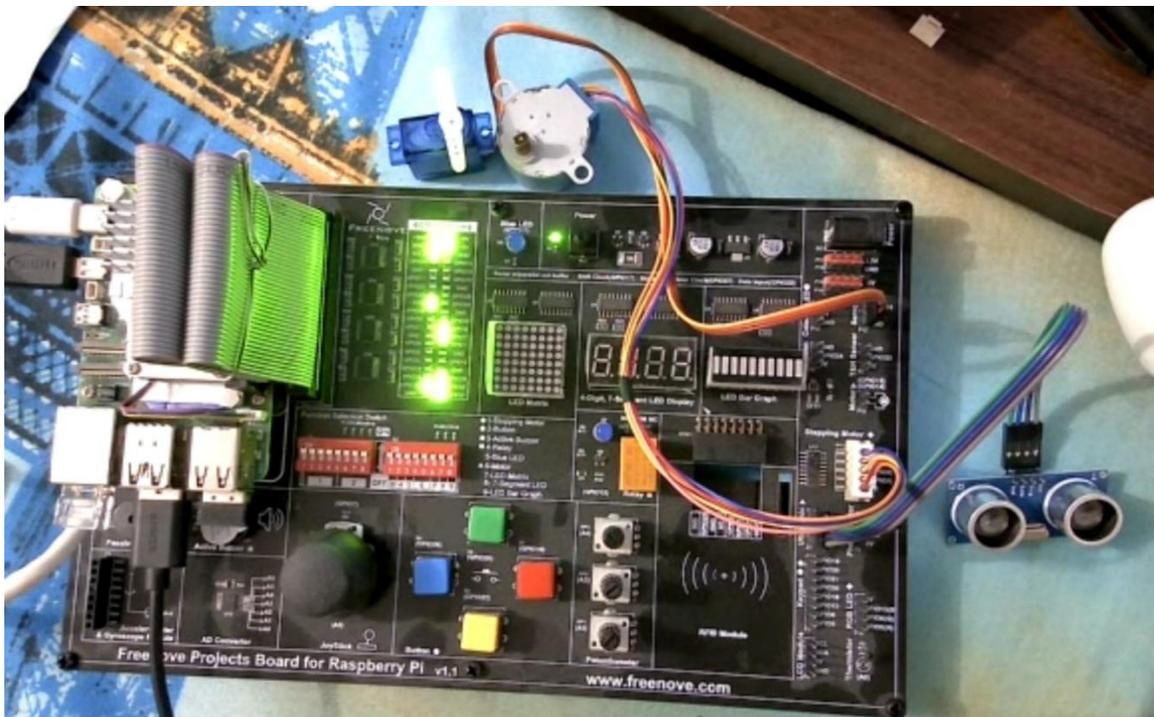


Рисунок 6 – Дослідний зразок готовий до тестування



Рисунок 7 – Перевірка точності сенсору за допомогою лінійки та перешкоди

```

Shell
>>> %FastDebug diploma.py
Робот стартує...
Відстань: 21.7 см
Все чисто, сервомотор не рухається.
Кроковий двигун рухається...
  
```

Рисунок 8 – Результати сенсору при порівнянні

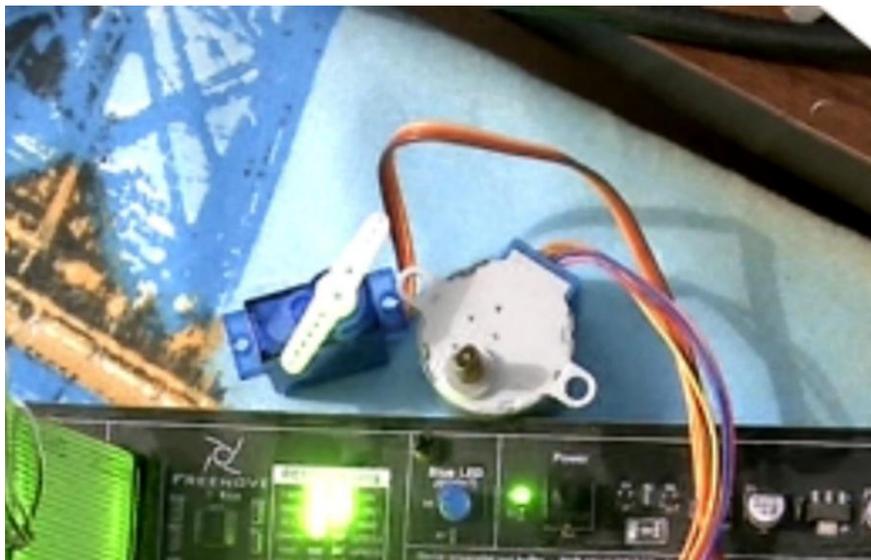


Рисунок 9 – Рух сервомотору на 45°

Додаток Б
(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ РОБОТИ

**ІНФОРМАЦІЙНО - ВИМІРЮВАЛЬНА СИСТЕМА ВИЗНАЧЕННЯ
ПЕРЕШКОД ТА ЇХ ОМИНАННЯ**

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: «Інформаційно-вимірювальна система визначення
перешкод та їх оминання»

Тип роботи: Магістерська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)
Підрозділ Кафедра ІРТС
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі
системою StrikePlagiarism 0,32%

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

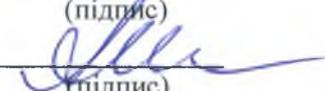
- Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення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

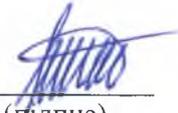
Експертна комісія:

Притула М.О. – к.т.н., доцент каф. ІРТС
(прізвище, ініціали, посада)

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


(підпис)


(підпис)

Особа, відповідальна за перевірку 
(підпис)

Семенов А.О.
(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник 
(підпис)

Притула М.О. – к.т.н., доцент каф. ІРТС
(прізвище, ініціали, посада)

Здобувач 
(підпис)

Гайдамашко А.О.
(прізвище, ініціали)

Додаток В (Довідниковий)

Лістинг програми для системи оминання перешкод

```

from gpiozero import AngularServo, OutputDevice,
DistanceSensor
from time import sleep
import random
import warnings

warnings.filterwarnings("ignore")

front_left_servo = AngularServo(17, min_angle=0,
max_angle=180)
front_right_servo = AngularServo(18, min_angle=0,
max_angle=180)

rear_motor_pins = (19, 13, 6, 5) # GPIO для крокових моторів
rear_motors = [OutputDevice(pin) for pin in rear_motor_pins]

CCWStep = (0x01, 0x02, 0x04, 0x08)
CWStep = (0x08, 0x04, 0x02, 0x01)

def moveOnePeriod(direction, ms):
    if ms < 3:
        ms = 3
    for j in range(4):
        for i in range(4):
            if direction == 1:
                rear_motors[i].on() if (CCWStep[j] == 1 << i)
            else rear_motors[i].off()
            else:
                rear_motors[i].on() if (CWStep[j] == 1 << i)
            else rear_motors[i].off()
        sleep(ms * 0.001)

```

```

def moveSteps(direction, ms, steps):
    for _ in range(steps):
        moveOnePeriod(direction, ms)

def stopRearMotors():
    for m in rear_motors:
        m.off()

sensor_servo = AngularServo(27, min_angle=0, max_angle=180) #
змінити GPIO під ваш пін

trigPin = 14
echoPin = 15
sensor = DistanceSensor(echo=echoPin, trigger=trigPin,
max_distance=3)

# Функції керування рухом

def moveForward(steps=512):
    front_left_servo.angle = 90 # прямиий рух
    front_right_servo.angle = 90
    moveSteps(1, 3, steps)
    stopRearMotors()

def turn(direction='left', steps=256):
    if direction == 'left':
        front_left_servo.angle = 45
        front_right_servo.angle = 135
        moveSteps(1, 3, steps)
    else:
        front_left_servo.angle = 135
        front_right_servo.angle = 45
        moveSteps(1, 3, steps)
    stopRearMotors()
    front_left_servo.angle = 90

```

```

front_right_servo.angle = 90

def scanObstacle():

    # Повертаємо сенсор вперед і ліво/право для перевірки

    sensor_servo.angle = 90
    sleep(0.5)
    front_distance = sensor.distance * 100

    sensor_servo.angle = 45
    sleep(0.5)
    left_distance = sensor.distance * 100

    sensor_servo.angle = 135
    sleep(0.5)
    right_distance = sensor.distance * 100

    sensor_servo.angle = 90
    sleep(0.2)
    return front_distance, left_distance, right_distance

def loop():
    while True:
        front, left, right = scanObstacle()
        print(f"Front: {front:.1f}cm | Left: {left:.1f}cm |
Right: {right:.1f}cm")

        if front > 20: # якщо попереду більше ніж 20 см,
рухаємося вперед
            moveForward(steps=256)
        else:
            # Перешкода попереду – обираємо сторону для обходу
            if left > right:
                turn('left', steps=128)
            else:

```

```
        turn('right', steps=128)
        moveForward(steps=128)

def destroy():
    stopRearMotors()
    front_left_servo.close()
    front_right_servo.close()
    sensor_servo.close()
    sensor.close()

# Точка входу

if __name__ == '__main__':
    print("Рухома платформа стартує...")
    try:
        loop()
    except KeyboardInterrupt:
        print("Завершення програми")
    finally:
        destroy()
```