

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Модель управління паркуванням мобільного роботу
на основі даних від датчика відстані

(тема)

Виконав:

здобувач 2 року навчання,

групи СПм-23-3

Максим ІЛЛАРІОНОВ

(власне ім'я, прізвище)

Спеціальність 123 «Комп'ютерна інженерія»

(код і повна назва спеціальності)

Тип програми освітньо-наукова

(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування

(повна назва освітньої програми)

Керівник: професор Анатолій КАРГІН

(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ

(підпис)

Андрій КОВАЛЕНКО

(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ другий (магістерський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-наукова _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Системне програмування _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Ілларіонову Максиму Юрійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Модель управління паркуванням мобільного роботу на основі даних від датчика відстані

затверджена наказом по університету від “ 21 ” квітня 2025 р. № 296 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 16 червня 2025 р.

3. Вхідні дані до роботи _____

паркування

мобільний робот

Matlab

UML Use case

ультразвуковий датчик

4. Перелік питань, що потрібно опрацювати у роботі _____

Постановка задачі

Опис предметної області і огляд підходів до вирішення завдання

Дослідження системи управління парковкою мобільного роботу на основі даних від

ультразвукових датчиків відображення

Проектування програмно-апаратного прототипу системи

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 19 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання та аналіз літератури	21.04.2025–30.04.2025	
2	Огляд існуючих рішень та алгоритмів	01.05.2025–12.05.2025	
3	Розробка моделі	13.05.2025–22.05.2025	
4	Вибір програмних засобів	23.05.2025–30.05.2025	
5	Програмна реалізація	31.05.2025–02.06.2025	
6	Аналіз отриманих результатів	03.06.2025–05.06.2025	
7	Оформлення записки	06.06.2025–12.06.2025	

Дата видачі завдання “ 21 ” квітня 2025 р.

Здобувач

_____ (підпис)

Керівник роботи

_____ (підпис)

професор Анатолій КАРГІН

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 99 с., 36 рис., 3 дод., 8 джерел.

МОБІЛЬНИЙ РОБОТ, АВТОМАТИЧНЕ ПАРКУВАННЯ, УЛЬТРАЗВУКОВІ СЕНСОРИ, НАВІГАЦІЯ, КЕРУВАННЯ РУХОМ, MATLAB, ARDUINO, МОДЕЛЮВАННЯ.

Метою кваліфікаційної роботи є розробка, програмна реалізація та експериментальне дослідження інтелектуальної системи автономного паркування мобільного робота, що використовує дані від ультразвукових сенсорів для виявлення перешкод, оцінки придатності вільного простору та точного позиціонування в межах заданого паркувального місця.

У кваліфікаційній роботі досліджено та реалізовано систему автоматизованого паркування мобільного робота з використанням даних від ультразвукових сенсорів та алгоритмів інтелектуального керування. Основною метою є розробка програмно-апаратного рішення, здатного самостійно орієнтуватися у заданому середовищі, виявляти придатні паркувальні місця та здійснювати маневри з високою точністю, уникаючи зіткнень з перешкодами. Розроблена система базується на поєднанні сенсорної обробки, просторової навігації, траєкторного планування та пропорційного керування рухом.

Для симуляційної перевірки функціонування моделі створено детальний математичний опис параметрів мобільної платформи, сенсорного сприйняття та навколишнього середовища. Проведено моделювання в середовищі MATLAB з візуалізацією траєкторії, відгуків сенсорів, орієнтації руху та точності досягнення цільової позиції.

ABSTRACT

Master's thesis: 99 pages, 36 figures, 3 appendices, 8 sources.

MOBILE ROBOT, AUTONOMOUS PARKING, ULTRASONIC SENSORS, NAVIGATION, MOTION CONTROL, MATLAB, ARDUINO, SIMULATION.

The major goal of this thesis is the development, software implementation, and experimental investigation of an intelligent system for autonomous parking of a mobile robot, which utilizes data from ultrasonic sensors to detect obstacles, assesses the suitability of available space, and ensure accurate positioning within a designated parking zone.

In order to an automated parking system for a mobile robot based on data acquired from ultrasonic sensors and intelligent motion control algorithms. The primary objective is to design a hardware-software solution capable of navigating autonomously within a defined environment, identifying appropriate parking spaces, and executing precise maneuvering while avoiding collisions with obstacles. The developed system integrates sensor data processing, spatial navigation, path planning, and proportional control of movement.

To validate the model's functionality, a detailed mathematical description of the mobile platform, sensor perception, and environmental parameters was developed. Simulation was carried out in MATLAB, including the visualization of the robot's trajectory, sensor responses, orientation during motion, and the accuracy of reaching the target position.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	8
ВСТУП	9
1 ПОСТАНОВКА ЗАДАЧІ.....	11
2 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ І ОГЛЯД ПІДХОДІВ ДО ВИРІШЕННЯ ЗАВДАННЯ	14
2.1 Аналіз останніх досліджень і публікацій	16
2.2 Огляд застосувань, що вимагають паркування мобільного роботу.....	18
2.3 Огляд апаратних засобів, що застосовуються.....	20
2.4 Огляд моделей, методів та правил паркування.....	22
2.5 Огляд систем паркування автономних машин.....	24
3 ДОСЛІДЖЕННЯ СИСТЕМИ УПРАВЛІННЯ ПАРКОВКОЮ МОБІЛЬНОГО РОБОТУ НА ОСНОВІ ДАНИХ ВІД УЛЬТРАЗВУКОВИХ ДАТЧИКІВ ВІДОБРАЖЕННЯ	27
3.1 Концептуальна модель системи. UML Use case	27
3.2 Логічна модель системи	29
3.3 Моделювання в MATLAB.....	41
3.4 Просторова модель мобільного роботу	46
4 ПРОЕКТУВАННЯ ПРОГРАМНО-АПАРАТНИЙ ПРОТОТИП СИСТЕМИ	50
4.1 Архітектура програмної частини прототипу системи.....	50
4.2 Компоненти програмної частини	53
4.3 Опис даних та бібліотек підтримки компонентів системи	55
4.4 Опис контрольних прикладів і результатів тестування	58
ВИСНОВКИ.....	67
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	69
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	71
ДОДАТОК Б Програмний код.....	82

Б.1 Код ParkingSystemSimulation.m.....	82
Б.2 PSS_v1.m.....	83
Б.3 ParkingStateMachine.h	88
Б.4 SensorArray.cpp.....	89
Б.5 MotorController.h.....	90
Б.6 DataStructures.h.....	90
Б.7 MathUtils.cpp.....	91
Б.8 PerformanceTest.cpp	92
Б.9 Реалізація на Arduino.....	92
ДОДАТОК В Діаграма послідовності для процесу паркування	98

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

AI – штучний інтелект

API – інтерфейс прикладного програмування

DC – постійний струм

HAL – апаратно-абстрактний рівень

I²C – інтерфейс послідовної шини між інтегральними схемами

LED – світлодіод

Li-ion – літій-іонна акумуляторна батарея

MATLAB – середовище моделювання та чисельних обчислень

OLED – органічний світлодіодний дисплей

PID – пропорційно-інтегрально-диференціальний регулятор

PWM – широтно-імпульсна модуляція

RAM – оперативна пам'ять

RGB – модель кольору з червоного, зеленого та синього компонентів

SR04 – модель ультразвукового сенсора HC-SR04

UART – універсальний асинхронний приймач-передавач

USB – універсальна послідовна шина

ВСТУП

У сучасних умовах стрімкого розвитку робототехніки та автоматизації все більшої актуальності набувають задачі автономного управління мобільними роботами в складних та динамічних середовищах. Однією з ключових функціональних можливостей таких систем є здатність до автономного паркування, що безпосередньо пов'язано з рівнем інтелектуального сприйняття простору, точністю орієнтації, ефективністю планування траєкторій та безпечністю маневрування. З урахуванням зростаючого попиту на автономні платформи в логістиці, на транспортних засобах, у виробничих та сервісних сферах, розробка систем паркування для мобільних роботів набуває стратегічного значення. Забезпечення точного входження в задану позицію в умовах обмеженого простору та наявності перешкод є не лише питанням функціональності, а й критичним чинником безпеки, енергоспоживання та стійкості до зовнішніх впливів.

Однією з найбільш ефективних і доступних технологій сенсорного забезпечення для таких задач залишаються ультразвукові сенсори. Вони дозволяють реалізовувати короткодистанційне виявлення об'єктів, характеризуються простотою інтеграції в апаратні системи, низьким рівнем енергоспоживання та задовільною точністю в умовах контрольованого середовища. Завдяки цим властивостям ультразвукові сенсори широко застосовуються у вбудованих системах з обмеженим обчислювальним ресурсом. Проте ефективне їх використання потребує гнучких алгоритмів обробки даних, адаптивних моделей прийняття рішень та чіткої організації архітектури управління. У цьому контексті особливого значення набуває створення математичних моделей, які б дозволили змодельовати поведінку системи в умовах реального середовища, оптимізувати параметри контролю та забезпечити стійке паркування за відсутності точного картографічного опису простору.

Незважаючи на широкий спектр досліджень у цій сфері, низка проблем залишається відкритою. Зокрема, потребує подальшого опрацювання питання поєднання простоти апаратної реалізації з високим рівнем інтелектуальної адаптації до зовнішніх змін. Також актуальним є дослідження ефективності різних алгоритмічних підходів – від логіки на основі правил до моделей машинного навчання – в умовах шумових даних від сенсорів. Саме тому доцільним є проведення комплексного дослідження, яке охоплює аналіз існуючих систем, моделювання типових сценаріїв паркування, розробку концептуальної архітектури та реалізацію прототипу з використанням засобів MATLAB як універсального інструменту для математичного та поведінкового моделювання.

Таким чином, дослідження систем управління паркуванням мобільного робота на основі даних від ультразвукових сенсорів має на меті не лише поглибити теоретичне розуміння принципів автономної навігації, а й надати практичні інструменти для реалізації таких систем в умовах реального застосування.

1 ПОСТАНОВКА ЗАДАЧІ

Сучасний розвиток мобільної робототехніки супроводжується стрімким зростанням кількості прикладних завдань, які вимагають автономної навігації в умовах структурованих і неструктурованих просторів. Однією з найважливіших і водночас найскладніших задач для автономного мобільного робота є задача точного та безпечного паркування в обмеженому середовищі. Вона передбачає забезпечення здатності роботизованої платформи самостійно оцінювати навколишню ситуацію, ідентифікувати наявність перешкод, визначати паркувальні зони та формувати оптимальну траєкторію для завершення маневру без втручання оператора.

У порівнянні з класичними транспортними засобами, мобільні роботи зазвичай мають менші габарити, нижчу масу та специфічну конструкцію ходової частини, що потребує адаптації підходів до керування та сенсорного забезпечення. Проте навіть за таких умов задача паркування залишається обчислювально складною, особливо в умовах неповної або зашумленої інформації про навколишнє середовище. Відтак, виникає потреба у створенні ефективної системи управління паркуванням, що враховує обмеженість обчислювальних ресурсів, динамічність зовнішніх умов та необхідність адаптивної поведінки.

У центрі уваги даного дослідження знаходиться підхід до вирішення задачі паркування мобільного робота з використанням даних від ультразвукових сенсорів. Ультразвукові сенсори є одним із найпоширеніших типів недорогих сенсорів відстані, які забезпечують базову інформацію про положення об'єктів у близькому радіусі дії. Вони дозволяють отримувати інформацію про перешкоди навіть за умов поганого освітлення, що особливо важливо в закритих просторах або при роботі вночі. Разом з тим, такі сенсори мають обмежену точність, кут огляду та залежні від матеріалу відбивної поверхні характеристики. Це зумовлює необхідність розробки відповідної

моделі обробки їхніх даних і логіки прийняття рішень.

Постановка задачі передбачає поетапне визначення вимог до створюваної системи управління паркуванням. По-перше, необхідно сформулювати функціональні вимоги, серед яких – здатність виявляти перешкоди на шляху руху, ідентифікувати наявність вільної паркувальної зони, планувати траєкторію та виконувати керування рухом з високою точністю. По-друге, потрібно визначити технічні обмеження – такі як роздільна здатність ультразвукових сенсорів, частота оновлення даних, затримки обробки та обмеження на швидкість обчислень. По-третє, слід вибрати алгоритмічну модель керування, яка відповідатиме заданим умовам. У даній роботі розглядаються можливості використання rule-based логіки, скінченних автоматів, нечітких правил і методів машинного навчання, з урахуванням можливості їхньої реалізації в умовах обмеженого середовища та на фізичному мікроконтролері.

Паралельно з цим необхідно вирішити задачу побудови структурної архітектури системи, яка поєднує сенсорну підсистему, блок прийняття рішень і виконавчий механізм. Передбачено створення концептуальної, логічної та фізичної моделей системи на основі підходів системного аналізу, моделювання в MATLAB/Simulink, а також програмної реалізації з використанням бібліотек для обробки сигналів і керування рухом.

Крім того, задача включає розробку програмно-апаратного прототипу для проведення експериментальних випробувань. Це дозволить перевірити ефективність розробленої системи на практиці, оцінити її здатність до точного паркування за різних умов середовища та зробити висновки щодо доцільності впровадження подібних рішень у більш складні робототехнічні системи. Основна задача полягає у формалізації й реалізації системи автономного паркування мобільного робота з використанням ультразвукових сенсорів та адаптивних алгоритмів прийняття рішень, які забезпечують точність, надійність і стійкість функціонування системи.

Об'єктом дослідження є процес автономного паркування мобільного

колісного робота в умовах обмеженого простору з використанням ультразвукових сенсорів для виявлення перешкод і навігаційної орієнтації.

Метою кваліфікаційної роботи є розробка, програмна реалізація та експериментальне дослідження інтелектуальної системи автономного паркування мобільного робота, що використовує дані від ультразвукових сенсорів для виявлення перешкод, оцінки придатності вільного простору та точного позиціонування в межах заданого паркувального місця.

Для досягнення поставленої мети в роботі необхідно виконати такі дослідницькі та прикладні завдання:

- провести аналіз науково-технічних підходів до автоматизованого паркування роботизованих платформ, з акцентом на сенсорне забезпечення та алгоритми керування;

- розробити просторову модель мобільного робота із заданими конструктивними параметрами, включаючи компонування сенсорів, елементів живлення та виконавчих пристроїв;

- сформувати програмну архітектуру системи управління, яка забезпечує обробку даних ультразвукових сенсорів та генерує рішення щодо придатного паркувального місця;

- реалізувати алгоритм виявлення перешкод, розпізнавання вільного простору, планування траєкторії та контролю переміщення мобільного робота;

- виконати моделювання процесу автономного паркування в середовищі MATLAB/Simulink з урахуванням перешкод та обмеженого простору;

- провести візуалізацію руху, аналіз сенсорних даних та оцінку точності кінцевого позиціонування на основі отриманих результатів симуляції.

2 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ І ОГЛЯД ПІДХОДІВ ДО ВИРІШЕННЯ ЗАВДАННЯ

Система автономного паркування мобільного робота є міждисциплінарною галуззю, що поєднує знання з робототехніки, автоматичного керування, сенсорної обробки даних та інтелектуального прийняття рішень. Предметна область охоплює сукупність технологічних, алгоритмічних і апаратних засобів, призначених для забезпечення здатності мобільного робота орієнтуватися в навколишньому середовищі, виявляти наявність паркувального місця, аналізувати перешкоди та здійснювати паркувальні маневри без зовнішнього втручання. Ці функції є критично важливими для широкого спектра застосувань – від сервісних роботів у закритих приміщеннях до автономних транспортних платформ, які діють в умовах непередбачуваного середовища.

У контексті даної роботи розглядаються системи, які функціонують на основі ультразвукових сенсорів – недорогих, широко доступних компонентів, що дають змогу здійснювати базову навігацію на основі відстані до об'єктів. Попри свої обмеження в точності та оглядовості, ці сенсори залишаються незамінними в умовах обмеженого бюджету або при розробці прототипів. Їхня інтеграція до систем паркування потребує розробки спеціалізованих алгоритмів фільтрації, обробки та інтерпретації сигналів, оскільки ультразвук схильний до помилок через багатопляхові відбиття, поглинання або неправильне відбиття від певних поверхонь.

Огляд сучасних підходів до вирішення задачі автономного паркування демонструє наявність кількох парадигм моделювання й керування. Найпростішими є rule-based системи, що діють за принципом логічних умов типу "якщо-то", у яких на основі певних сенсорних сигналів визначається напрямок і дія. Такі підходи забезпечують низьку обчислювальну складність, однак не здатні до гнучкої адаптації в умовах зміни конфігурації середовища.

Іншою групою рішень є моделі на основі кінцевих автоматів, які дозволяють описувати паркувальний процес як послідовність станів із визначеними умовами переходу. Це дає змогу формалізувати поведінку системи, але потребує попереднього сценарного опису ситуацій, що обмежує масштабованість моделі при ускладненні задачі.

Суттєвої гнучкості надає використання нечіткої логіки, яка дозволяє враховувати неповну або нечітку інформацію, формуючи рішення на основі лінгвістичних змінних і нечітких множин. Такі підходи особливо ефективні при наявності шумів або відсутності чіткої геометрії паркувального простору. Їхня реалізація вимагає ручного налаштування правил і функцій належності, що створює певну складність у розробці, але забезпечує адаптивність до варіативних ситуацій.

Одним із найсучасніших напрямів є застосування методів машинного навчання, зокрема глибоких нейронних мереж і навчання з підкріпленням, які дозволяють моделі навчатися поведінці паркування на основі даних або досвіду взаємодії зі середовищем. Це відкриває можливість адаптації до нових ситуацій і складної багатофакторної поведінки, однак потребує значних обчислювальних ресурсів, тренувальних вибірок і не завжди дає інтерпретовані результати.

У якості компромісного рішення часто використовуються гібридні моделі, які поєднують логічні блоки з елементами адаптивного планування (машинне навчання для уточнення траєкторії або оптимізації маневру). У таких системах архітектура проектується як багаторівнева, де на нижньому рівні працюють базові сенсорні реакції, а на вищому – узагальнені стратегії поведінки.

Таким чином, предметна область побудови систем управління паркуванням мобільного робота є складною інженерною задачею, яка вимагає врахування взаємозв'язку між апаратними можливостями, алгоритмічними підходами й архітектурними рішеннями. Вибір конкретного підходу завжди зумовлений компромісом між точністю, швидкодією,

адаптивністю та вартістю реалізації.

2.1 Аналіз останніх досліджень і публікацій

Сучасні наукові дослідження у сфері управління мобільними роботами демонструють зростаючий інтерес до використання доступних за вартістю та водночас надійних сенсорних технологій, таких як ультразвукові датчики. Зокрема, ці пристрої активно застосовуються для забезпечення точного визначення положення, виявлення перешкод та реалізації функцій автономного паркування. У проаналізованій літературі представлено різноманітні підходи до сенсорної інтеграції, алгоритмічного забезпечення та методів оцінювання точності систем навігації мобільних роботів. У публікаціях [1, 2] висвітлено систематизовані методи використання ультразвукових сенсорів, що робить ці джерела базовими для вивчення потенціалу таких систем, особливо у випадках з обмеженим простором або ресурсами.

У роботі [1] проведено глибокий аналіз сенсорних систем, що застосовуються для детекції перешкод у мобільній робототехніці, зокрема порівняно ультразвукові, LiDAR, інфрачервоні сенсори та камери. Ультразвукові датчики виокремлено як найбільш придатні для навігації на коротких відстанях, особливо у середовищах із обмеженим бюджетом та ресурсами, завдяки низькому енергоспоживанню, високій надійності та доступності.

Огляд [2] узагальнює можливості застосування ультразвукових сенсорів у контексті робототехніки, автомобільних ADAS-систем, безпеки й інфраструктури «розумних міст». Особлива увага приділена доцільності поєднання УЗ-датчиків з іншими сенсорними компонентами для досягнення вищої стабільності та адаптивності в складних умовах середовища.

У публікаціях [3, 4] розглядаються новітні підходи до вдосконалення традиційних ультразвукових сенсорних систем у мобільній робототехніці.

Зокрема, [3] присвячена новому поколінню сенсорів ADAR (Acoustic Detection and Ranging), які забезпечують 360-градусне охоплення без необхідності використання рухомих механізмів або дорогих оптичних компонентів. У статті підкреслюється економічна ефективність таких рішень порівняно з традиційними LiDAR-системами, а також їх сертифікація за стандартом SIL2, що свідчить про відповідність промисловим вимогам.

Робота [4] має прикладний характер та пропонує технічне обґрунтування ADAR-архітектури, яка базується на реконструкції акустичного поля замість класичної діаграми випромінювання. Завдяки високій частоті оновлення та зменшенню кількості рухомих частин забезпечено підвищену стійкість до перешкод у процесі маневрування. Демонструється практична ефективність системи на прикладі логістичних мобільних платформ із тривимірною ультразвуковою картою навколишнього середовища, що функціонує без необхідності попереднього картографування.

Інноваційний підхід до сенсорики представлено в роботі [5], де ультразвукові сенсори інтегруються безпосередньо в конструкцію коліс мобільного робота. Такий підхід дозволяє здійснювати одночасний контактний і безконтактний аналіз середовища. За допомогою аналізу акустичного відгуку система може не лише ідентифікувати тип поверхні, а й виявляти мікроперешкоди, що відкриває нові можливості для функціонування в умовах поганого освітлення чи високої запиленості.

У [6] описано методологію розгортання множинних ультразвукових локалізаційних систем без попереднього калібрування. Запропоновано алгоритм SCAN, який дозволяє мобільному роботу одночасно створювати карту навколишнього простору та автоматично калібрувати позиції маяків, що особливо актуально для задач паркування у динамічному середовищі. Результати експериментів підтверджують значне підвищення точності локалізації.

У публікації [7] представлено комплексну IoT-систему для автоматизованого паркування, яка включає ультразвукові та інші сенсори,

дисплей і мобільний застосунок. Рішення забезпечує моніторинг у режимі реального часу та демонструє інтеграцію сенсорного середовища із цифровими сервісами для оптимізації процесу паркування.

Таким чином, проаналізовані джерела засвідчують тенденцію до поєднання традиційних ультразвукових рішень з інноваційними архітектурами та алгоритмами. Такі підходи сприяють розширенню функціональних можливостей систем автономного паркування, підвищенню їхньої надійності, адаптивності та ефективності в складних середовищах експлуатації.

2.2 Огляд застосувань, що вимагають паркування мобільного роботу

У сучасних умовах розвитку автономних систем мобільні роботи дедалі активніше впроваджуються в різноманітні сфери діяльності, що передбачає потребу в їхньому безпечному та точному позиціонуванні у просторі. Паркування є одним із ключових етапів циклу функціонування таких систем, оскільки забезпечує завершення місії, збереження простору та підготовку до наступного завдання. Найчастіше вимога до реалізації функції паркування виникає у сценаріях, де мобільний робот взаємодіє з обмеженим, заздалегідь структурованим або динамічно змінним середовищем.

Тип застосування	Точність позиціонування	Швидкість паркування	Рівень автономії	Складність середовища
Сервісні роботи	±2-5 см	Середня	Високий	Динамічне
Складські AGV	±1-2 см	Висока	Середній	Структуроване
Медичні роботи	±0.5-2 см	Середня	Критичний	Контрольоване
Домашні роботи	±3-10 см	Низька	Високий	Напів-структуроване
Зовнішні роботи	±5-20 см	Низька	Середній	Неструктуроване

Рисунок 2.1 – Таблиця порівняння основних параметрів

Однією з найпоширеніших сфер є складська логістика, де автономні мобільні платформи здійснюють транспортування вантажів між зонами зберігання, сортуванням та відвантаженням. У таких умовах роботи мають здійснювати паркування біля стелажів або док-станцій, при цьому важливою є здатність точно позиціонуватися в умовах щільної забудови та перешкод. В аналогічних умовах працюють мобільні платформи в виробничих приміщеннях, де передбачається паркування в точно визначених зонах завантаження або технічного обслуговування.

Іншою категорією застосування є розумні транспортні засоби, де реалізація автономного паркування дозволяє зменшити залежність від водія, знизити ризики зіткнень, а також оптимізувати використання паркінгового простору. У таких сценаріях основна увага приділяється здатності системи самостійно оцінювати доступний простір, уникати перешкод і забезпечувати динамічне маневрування навіть у середовищах зі змінною геометрією.

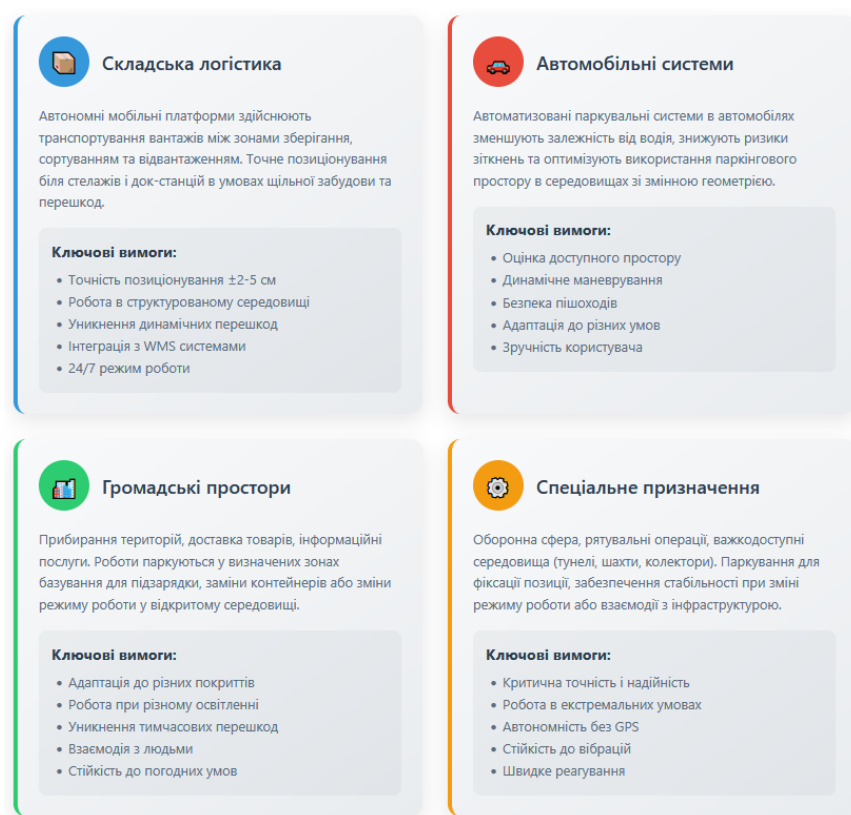


Рисунок 2.2 – Застосування мобільних роботів з функцією паркування

У сфері обслуговування громадських просторів, наприклад, прибирання територій, доставка товарів або забезпечення інформаційних послуг, мобільні роботи також потребують ефективного паркування у визначених зонах базування, де відбувається підзарядка, заміна контейнерів або зміна режиму роботи. Такі роботи часто функціонують у відкритому середовищі, що вимагає від систем паркування адаптації до різноманітних покриттів, освітлення та наявності тимчасових перешкод.

Окремої уваги заслуговують роботи спеціального призначення, зокрема в оборонній сфері, у рятувальних операціях або у важкодоступних середовищах (наприклад, тунелі, шахти, технічні колектори), де паркування використовується для фіксації позиції, забезпечення стабільності при зміні режиму роботи або взаємодії з інфраструктурою. У таких випадках точність і надійність паркувального маневру часто мають критичне значення для виконання місії.

Таким чином, функція паркування в системах мобільної робототехніки є універсальною складовою, яка забезпечує стабільне завершення завдання, підтримку технічного ресурсу та ефективну взаємодію з фізичним середовищем. Різноманіття сфер застосування визначає широкий спектр вимог до архітектури паркувальних систем – від простих детекторів перешкод до складних багаторівневих сенсорно-адаптивних рішень.

2.3 Огляд апаратних засобів, що застосовуються

Ефективне управління паркуванням мобільного робота неможливе без ретельно спроектованої апаратної архітектури, яка охоплює сенсорні модулі, обчислювальні пристрої, виконавчі механізми, засоби енергоживлення та комунікаційні інтерфейси. Всі ці компоненти повинні працювати узгоджено, забезпечуючи своєчасне отримання, обробку й реалізацію інформації в умовах реального часу.

Серед сенсорних систем найбільш поширеними є ультразвукові

датчики, що забезпечують визначення відстані до об'єктів у межах кількох метрів з достатньою точністю для більшості задач паркування. Їх основними перевагами є низька вартість, енергоефективність та висока надійність у закритих приміщеннях. Такі сенсори часто використовуються в парі з інфрачервоними детекторами, які підвищують точність на малих відстанях і забезпечують чутливість до країв поверхонь. У складніших системах додатково використовуються магнітні сенсори, призначені для орієнтування відносно спеціальних магнітних міток або ліній маршруту.

Для задач, що потребують високої точності просторової орієнтації, використовуються системи на основі лазерного сканування, зокрема LiDAR, які дозволяють формувати деталізовану тривимірну картину навколишнього середовища. Вони забезпечують точне позиціонування навіть у динамічних умовах, однак мають вищу вартість та вимоги до обчислювальних потужностей. Ще одне перспективне рішення – візуальні камери (в тому числі RGB та камери глибини), які надають багату інформацію для побудови візуальної карти простору та розпізнавання об'єктів. Разом з тим, вони потребують обробки великих обсягів даних, що ускладнює їх застосування на платформах із обмеженими ресурсами.

Обчислювальні компоненти виконують аналіз сенсорної інформації та реалізацію логіки керування. Їх вибір залежить від складності алгоритмів і використовуваних сенсорів. У найпростіших реалізаціях це можуть бути мікроконтролери, які виконують обробку даних в режимі реального часу. Для більш складних задач із використанням алгоритмів машинного навчання або обробки зображень використовуються одноплатні комп'ютери, що підтримують роботу з нейронними мережами та графічними процесорами.

Виконавча частина системи реалізує фізичні дії, необхідні для маневрування й точного позиціонування. Основу складають мотор-редуктори, що забезпечують рух колісної бази, а також серводвигуни для керування напрямком. Зворотній зв'язок реалізується за допомогою енкодерів, які дають змогу контролювати положення та швидкість обертання

приводів.

Засоби живлення забезпечують автономну роботу системи та включають акумуляторні блоки з контролем заряду, що є критично важливим для тривалих місій. Для забезпечення внутрішньої комунікації між сенсорами, контролерами й приводами використовуються інтерфейси типу I2C, UART, SPI, CAN, які дозволяють здійснювати обмін даними на високій швидкості з мінімальною затримкою.

Таким чином, апаратне забезпечення відіграє ключову роль у побудові надійної системи автономного паркування мобільного робота. Його оптимальне компонування та відповідність завданню дозволяють досягти високої точності позиціонування, стабільності в динамічному середовищі та енергоефективності, що є критично важливими для роботизованих платформ, що функціонують у реальних умовах.

2.4 Огляд моделей, методів та правил паркування

Процес автономного паркування мобільного робота охоплює низку послідовних дій, що включають виявлення вільного простору, прийняття рішення щодо типу маневру, побудову траєкторії, її динамічне коригування та виконання самого паркувального руху. Для реалізації цього процесу застосовуються різні моделі, методи та алгоритмічні правила, які можна умовно класифікувати за ступенем формалізації, адаптивності й обчислювальної складності.

Одним із найбільш поширених підходів є використання моделей на основі правил (rule-based models). У таких системах поведінка мобільного робота описується набором логічних умов виду «якщо – то», які жорстко фіксують реакцію на певні входні сенсорні події. Наприклад, у випадку фіксації перешкоди на відстані меншій за заданий поріг робот зупиняється, а за виявлення вільного простору праворуч – здійснює поворот. Цей метод є найпростішим у реалізації та не вимагає значних обчислювальних ресурсів,

однак він виявляється малопридатним у складних або непередбачуваних середовищах через обмежену гнучкість логіки.

Наступним рівнем складності є кінцеві автомати станів. Цей підхід дозволяє структурувати процес паркування у вигляді набору дискретних станів (наприклад, «виявлення вільного місця», «вирівнювання позиції», «маневрування», «фінальне паркування»), між якими система переходить залежно від сенсорних даних. FSM забезпечує передбачуваність і стабільність поведінки, що особливо важливо в системах з вимогами до безпеки. Водночас при збільшенні складності сценаріїв кількість станів і переходів зростає експоненційно, що ускладнює масштабування.

Широке застосування в останні роки знаходять нечіткі логічні моделі. У таких системах правила базуються не на жорстких значеннях, а на нечітких множинах, що дозволяє описувати ситуації за допомогою лінгвістичних змінних (наприклад, «мала відстань», «середнє наближення», «велика швидкість»). Це забезпечує гнучкість і здатність працювати в умовах невизначеності або при наявності шумів у сенсорних вимірюваннях. Нечітка логіка добре підходить для систем, що функціонують у змінному середовищі, однак вона вимагає ретельного налаштування функцій належності та валідації правил.

Значного поширення набули також методи на основі машинного навчання, які дозволяють роботам навчатися оптимальній поведінці на основі прикладів або через взаємодію із середовищем. Серед них виділяються як традиційні алгоритми класифікації та регресії, зокрема згорткові нейронні мережі або моделі з підкріпленням. Такі підходи дозволяють роботу адаптуватися до нових ситуацій, узагальнювати зразки поведінки та приймати рішення в режимі реального часу. Головними викликами залишаються потреба в об'ємних навчальних вибірках, необхідність високих обчислювальних потужностей і складність інтерпретації результатів.

У практиці створення систем автономного паркування часто застосовуються гібридні методи, які поєднують кілька підходів задля

підвищення надійності та адаптивності. Наприклад, rule-based логіка використовується для критично важливих безпечних дій, FSM – для основної логіки управління, а методи машинного навчання – для оптимізації траєкторії або оцінки середовища. Така інтеграція дозволяє досягти балансу між інтерпретованістю, стабільністю та здатністю до навчання.

Таким чином, вибір моделі або методу паркування залежить від низки факторів: складності середовища, наявних сенсорів, апаратних можливостей, цільової точності й вимог до автономності. У контексті систем із використанням ультразвукових сенсорів найбільш доцільним є використання FSM або нечіткої логіки в базових системах, а також гібридних моделей у більш складних автономних платформах.

2.5 Огляд систем паркування автономних машин

У сучасній мобільній робототехніці системи автономного паркування є одним із ключових функціональних модулів, що забезпечують здатність машини самостійно здійснювати зупинку або в'їзд у обмежений простір без участі оператора. Ці системи активно впроваджуються як у побутових, так і в промислових застосуваннях, зокрема в автономних транспортних засобах, складських роботах, логістичних платформах, дронах наземного типу, сервісних роботах для інфраструктурних об'єктів тощо.

Основною вимогою до систем автономного паркування є здатність точно, надійно та безпечно визначати власне положення в просторі відносно цільової паркувальної позиції, враховуючи наявність статичних і динамічних перешкод, обмежену видимість, мінливість середовища та ресурсообмеженість системи. Для досягнення цих цілей різні виробники та дослідницькі групи розробляють архітектури, які поєднують сенсорне забезпечення, обчислювальні модулі прийняття рішень і виконавчі механізми руху.

Сучасні системи паркування умовно поділяються на три типи:

асистивні, напівавтономні та повністю автономні. У першому випадку система лише надає допоміжну інформацію оператору (наприклад, звукові чи візуальні сигнали від датчиків), залишаючи управління на його розсуд. Напівавтономні рішення здійснюють частину керування (наприклад, кут повороту керма), тоді як гальмування або контроль швидкості залишаються за користувачем. Повністю автономні системи повністю беруть на себе управління всіма маневрами, враховуючи просторову навігацію, виявлення місця для паркування, формування траєкторії та виконання руху.

У більшості реалізацій автономного паркування використовуються такі типи сенсорів: ультразвукові, інфрачервоні, LiDAR, камери, а також інерційні датчики та GPS для глобального позиціонування. Ультразвукові сенсори традиційно забезпечують високу ефективність при виявленні перешкод на малій відстані, що робить їх незамінними при виконанні точних маневрів у тісному просторі. Камери й LiDAR дозволяють будувати карту середовища, виконувати семантичну сегментацію простору та оцінювати складні ситуації, включаючи наявність рухомих об'єктів. У складних середовищах все частіше застосовуються мультисенсорні гібридні конфігурації, які об'єднують дані з різних сенсорів для підвищення точності та надійності системи.

Архітектурно системи автономного паркування реалізуються за трирівневою структурою: сенсорний рівень (збір і первинна обробка даних), рівень прийняття рішень (аналіз, планування та контроль траєкторії) і виконавчий рівень (керування приводами та виконавчими механізмами). Важливу роль відіграє також зворотний зв'язок – системи повинні мати можливість корекції дій на основі поточних сенсорних спостережень у режимі реального часу.

Особливу увагу дослідники приділяють адаптивним алгоритмам, які дозволяють системам паркування функціонувати не лише в добре структурованих середовищах, а й у напівструктурованих або динамічних. Це передбачає здатність системи до онлайн-планування, локальної реконструкції

карти, реагування на нові перешкоди та оптимізації руху в обмежених умовах.

Багато практичних рішень передбачають інтеграцію автономного паркування у ширшу архітектуру мобільного робота, де модулі навігації, безпеки та взаємодії з іншими системами працюють узгоджено. У реальному застосуванні, зокрема в складських умовах, автономне паркування забезпечує не лише оптимальне розміщення роботизованої платформи, а й підвищення загальної ефективності маршрутизації, зниження енергоспоживання та запобігання простоїв або зіткнень.

3 ДОСЛІДЖЕННЯ СИСТЕМИ УПРАВЛІННЯ ПАРКОВКОЮ МОБІЛЬНОГО РОБОТУ НА ОСНОВІ ДАНИХ ВІД УЛЬТРАЗВУКОВИХ ДАТЧИКІВ ВІДОБРАЖЕННЯ

3.1 Концептуальна модель системи. UML Use case

Представлена на рисунку 3.1 діаграма варіантів використання описує загальну функціональну логіку системи управління паркуванням мобільного робота на основі даних від ультразвукових сенсорів. У центрі цієї моделі – взаємодія між компонентами системи, середовищем та користувачем. Модель відображає ключові сценарії, які забезпечують реалізацію автономного паркувального маневру.

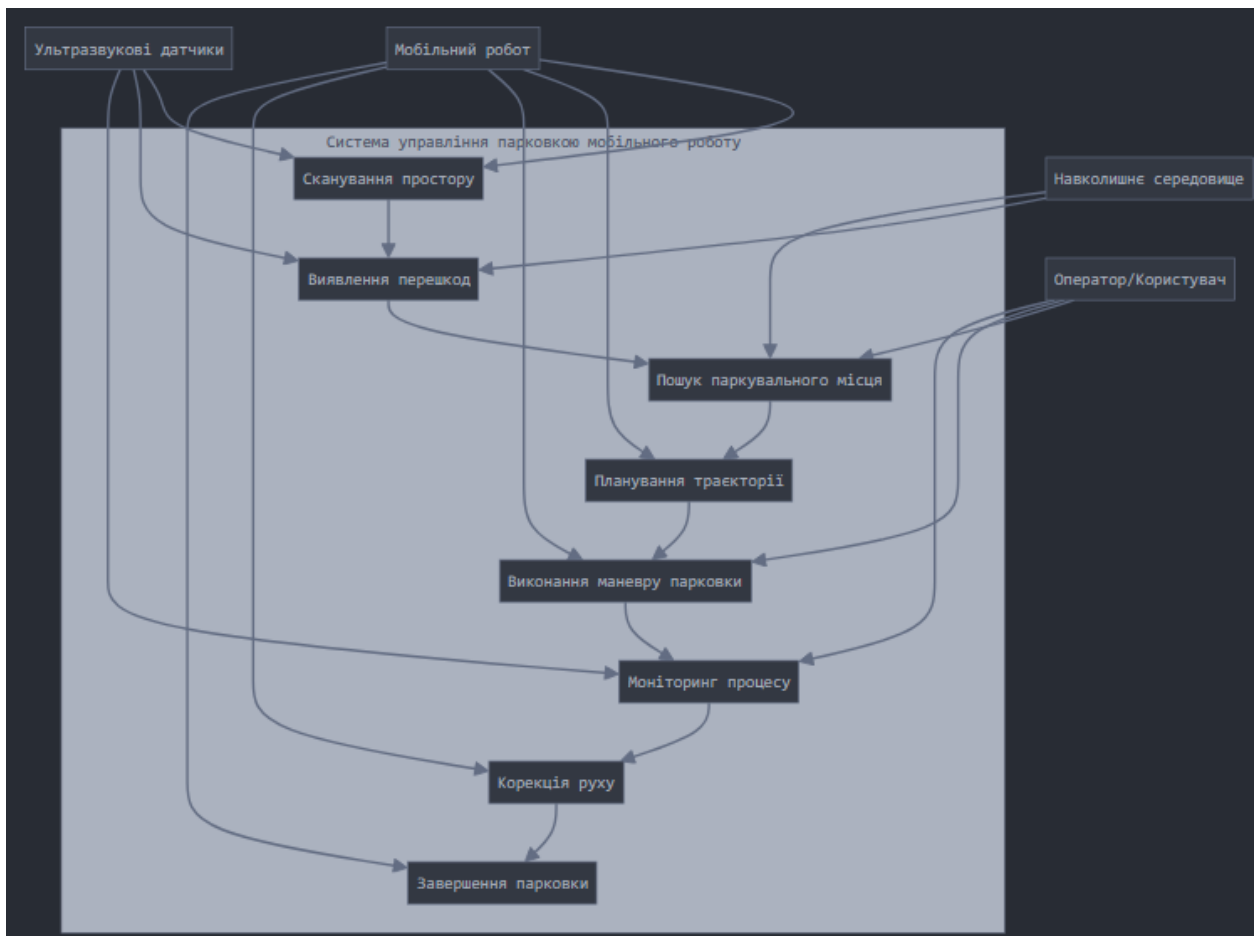


Рисунок 3.1 – UML Use Case

Система взаємодіє з чотирма основними зовнішніми акторами: ультразвукові датчики, мобільний робот, навколишнє середовище та оператор/користувач. Ці актори здійснюють вплив на процес або отримують дані в ході реалізації різних етапів.

У межах функціонального ядра системи послідовно реалізуються такі основні процеси:

- сканування простору. Система ініціює збір просторових даних за допомогою ультразвукових датчиків. Цей процес передбачає аналіз навколишнього середовища на предмет наявності вільного простору та перешкод;

- виявлення перешкод. На основі зібраних сенсорних даних визначаються потенційні перешкоди на маршруті. Це критичний етап, що впливає на безпечність усього процесу паркування. Інформація про перешкоди одразу враховується у подальшому плануванні;

- пошук паркувального місця. Алгоритм обробляє просторову інформацію з урахуванням геометрії довколишніх об'єктів і визначає потенційно придатні місця для зупинки;

- планування траєкторії. Після вибору місця виконується розрахунок траєкторії маневру паркування з урахуванням обмежень по руху, розмірів платформи, кутів повороту, радіуса розвороту та розташування перешкод;

- виконання маневру паркування. Мобільний робот виконує серію рухів, що відповідають запланованій траєкторії. Усі дії виконуються автономно з періодичним зчитуванням сенсорних даних;

- моніторинг процесу. Під час маневру система постійно перевіряє актуальні значення з сенсорів для виявлення відхилень або нових загроз, зокрема зміни положення об'єктів;

- корекція руху. У разі виявлення відхилення від очікуваної траєкторії система в режимі реального часу вносить коригування, наприклад змінює кут повороту чи зупиняє рух;

- завершення паркування. Після успішного завершення маневру

система фіксує досягнення цільового положення та передає відповідну інформацію оператору. Можливе сповіщення про завершення або запуск нової фази (наприклад, очікування).

Оператор або користувач може впливати на етапи «Пошук паркувального місця», «Планування траєкторії» та «Моніторинг процесу», особливо у випадках напівавтоматичного керування або необхідності ручного втручання. Також користувач отримує повідомлення про статус виконання завдання, хід паркування або виявлені помилки.

Важливо зазначити, що всі блоки пов'язані між собою зворотними зв'язками, які дозволяють системі бути адаптивною: інформація з навколишнього середовища, сенсорів та виконавчих модулів постійно оновлюється, забезпечуючи гнучке реагування на змінні умови.

3.2 Логічна модель системи

Представлена на рисунку 3.2 архітектурна модель системи управління паркуванням мобільного робота є багаторівневою та охоплює всі ключові компоненти, необхідні для повноцінного функціонування інтелектуального автономного рішення. В основі структури знаходиться апаратний рівень, який забезпечує фізичну взаємодію з навколишнім середовищем через ультразвукові сенсори, мікроконтролери та виконавчі пристрої, зокрема приводи та живлення. Цей рівень реалізує базову функціональність збору даних і керування рухом, зокрема за допомогою таких елементів як мікроконтролери Arduino або ESP32, що забезпечують обчислювальну підтримку роботи сенсорів та приводів.

Наступним шаром є рівень доступу до даних, який відповідає за організацію зберігання, обробки та реєстрації інформації, зокрема сенсорних показників, конфігурацій, телеметрії, логів і цифрових карт середовища. Цей рівень є критично важливим для забезпечення узгодженого обміну даними між низькорівневими компонентами системи та її інтелектуальними

модулями. Він підтримує структуроване збереження інформації та слугує основою для побудови цифрової репрезентації простору навколо робота.

На рівні сервісів реалізовано окремі функціональні блоки, які виконують спеціалізовані обчислення й забезпечують автономну поведінку системи. До таких сервісів належать модулі локалізації, орієнтації, контролю руху, обробки сенсорних даних та підтримки комунікацій. Саме цей рівень забезпечує обробку первинної інформації, її трансформацію в формати, придатні для прийняття рішень, а також взаємодію між апаратною частиною й логікою управління.



Рисунок 3.2 – Багаторівнева архітектура системи

Бізнес-логіка представлена рівнем, на якому функціонують інтелектуальні механізми прийняття рішень, включаючи контроль маневрів, планування траєкторій, обробку вхідних подій і реалізацію політик безпеки. Цей рівень є ядром автономної поведінки системи та координує весь процес

паркування, забезпечуючи адаптацію до поточних умов середовища, ухвалення рішень на основі сенсорних даних і генерацію відповідних команд на виконання.

Нарешті, на рівні презентації розташовані компоненти, які відповідають за взаємодію системи з оператором або зовнішніми сервісами. Вони включають інтерфейси користувача, візуальні системи виводу, такі як OLED-дисплеї або LED-індикатори, а також шлюзи API, що забезпечують інтеграцію з мобільними додатками, веб-інтерфейсами або хмарними платформами за допомогою стандартних протоколів, таких як REST, WebSocket або MQTT.

Загальна архітектура демонструє чітке розмежування відповідальностей між функціональними рівнями, що сприяє модульності, гнучкості розробки, полегшує масштабування й тестування системи, а також забезпечує основу для впровадження складних інтелектуальних алгоритмів у майбутніх версіях платформи.

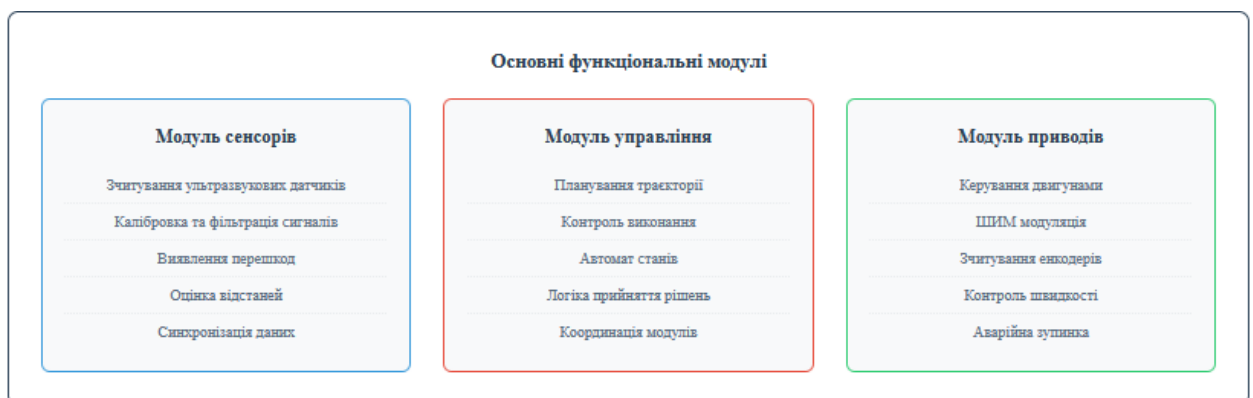


Рисунок 3.3 – Функціональна декомпозиція системи

На рисунку 3.3 відображено структуру основних функціональних модулів системи управління паркуванням мобільного робота. Архітектура поділяється на три ключові компоненти, кожен з яких виконує окремі завдання в рамках загального процесу автономного паркування. Модуль сенсорів відповідає за ініціалізацію ультразвукових датчиків, їх калібрування, обробку вхідних сигналів, виявлення перешкод, оцінку

достовірності отриманих показників і синхронізацію сенсорних даних для подальшого аналізу. Модуль управління виконує функції планування траєкторії, контролю виконання маневрів, обробки сигналів стану, прийняття рішень на основі заданої логіки та координування дій окремих підсистем. Модуль приводів реалізує безпосереднє управління рухом шляхом формування ШІМ-сигналів, активує виконавчі елементи, регулює швидкість і здійснює аварійне гальмування за необхідності. Така структурна декомпозиція дозволяє ефективно розподілити обчислювальні навантаження між компонентами системи та забезпечити узгоджену взаємодію між сенсорикою, логікою керування та приводами в реальному часі.

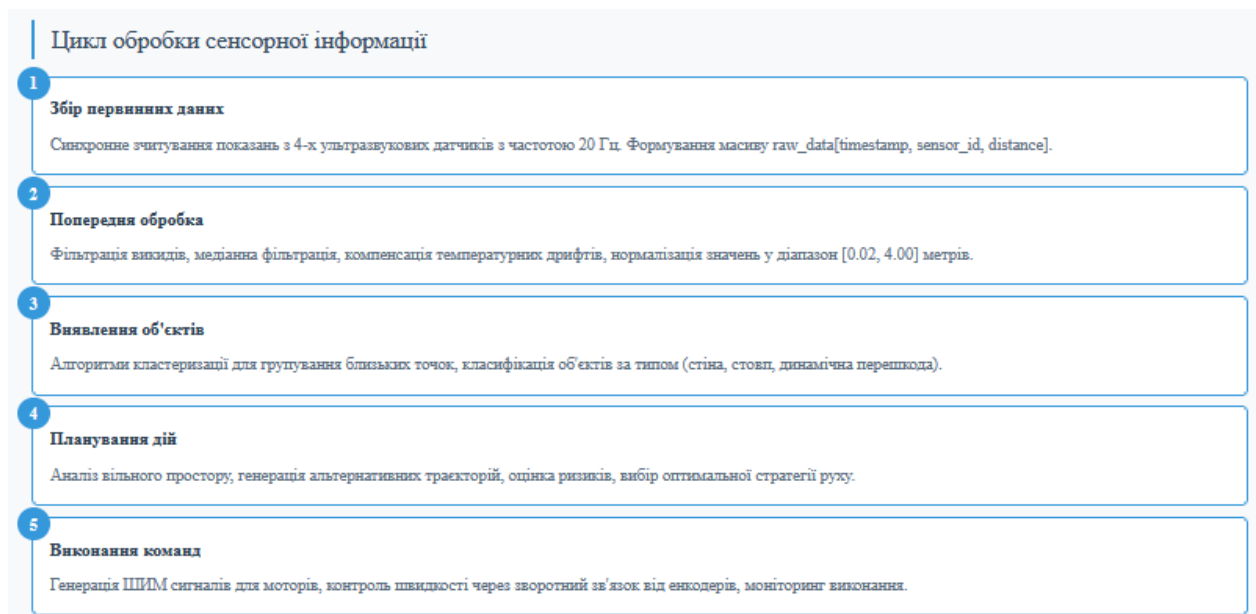


Рисунок 3.4 – Потіки обробки даних

На рисунку 3.4 представлено структурований опис циклу обробки сенсорної інформації в системі автономного паркування мобільного робота. Цей процес охоплює п'ять послідовних етапів, кожен з яких має критичне значення для забезпечення точності, адаптивності та надійності функціонування системи. Перший етап полягає у зборі первинних даних від ультразвукових датчиків, які працюють із частотою 20 Гц, фіксуючи часову мітку, ідентифікатор сенсора та виміряну відстань, що дозволяє сформува

базовий масив raw_data для подальшої обробки. Далі здійснюється попередня обробка даних, що включає фільтрацію шумів, компенсацію температурних дрейфів, нормалізацію вимірів та усунення значень за межами допустимого діапазону, що забезпечує чистоту і достовірність інформації. На третьому етапі виконується виявлення об'єктів, де застосовуються методи кластеризації для групування точок і класифікації об'єктів за типами, зокрема на стіни, стовпи або додаткові перешкоди. Четвертий етап – це планування дій, що передбачає аналіз доступного простору, оцінку ризиків, генерацію альтернативних траєкторій та вибір оптимальної стратегії руху. Завершується цикл виконання команд: відбувається генерація ШІМ-сигналів для керування моторами, контроль швидкості через енкодери та моніторинг виконання в реальному часі. Така послідовність забезпечує повноцінну обробку даних від моменту вимірювання до виконання дії мобільною платформою.

Основні кінематичні рівняння:	
$\dot{x} = v \times \cos(\theta)$	Вимірювання відстані:
$\dot{y} = v \times \sin(\theta)$	
$\dot{\theta} = \omega$	$d = (t_{echo} \times v_{sound}) / 2$
де: $v = (v_L + v_R) / 2$	$v_{sound} = 331.3 + 0.606 \times T [^{\circ}C]$
$\omega = (v_R - v_L) / L$	

Рисунок 3.5 – Математична модель системи

Математичні моделі широко використовуються в задачах управління мобільними роботами, зокрема при моделюванні руху платформи з диференціальним приводом та при обчисленні відстані на основі сигналів ультразвукового сенсора. Кінематичні рівняння точно описують зміну положення робота у двовимірному просторі залежно від його лінійної та кутової швидкостей, а формула визначення відстані враховує швидкість поширення звуку залежно від температури, що є критично важливим для підвищення точності в реальному середовищі.

На рисунку 3.5 представлено два ключові математичні блоки, що використовуються в моделюванні руху та сенсорному аналізі мобільного робота. Перший блок відображає основні кінематичні рівняння диференціального приводу, що визначають зміну координат робота у просторі. Рівняння для x' та y' базуються на лінійній швидкості v та орієнтації θ , яка змінюється з кутовою швидкістю ω . Швидкість v визначається як середнє значення лівої та правої лінійних швидкостей коліс, а кутова швидкість обчислюється на основі різниці між цими швидкостями, поділеної на відстань між колесами L .

Другий блок стосується вимірювання відстані ультразвуковим сенсором. Відстань d обчислюється як половина добутку часу затримки сигналу t_{echo} на швидкість звуку v_{sound} , оскільки сигнал проходить подвійну відстань – до об'єкта і назад. При цьому швидкість звуку залежить від температури повітря, що враховується через емпіричну формулу, яка включає лінійну залежність від температури в градусах Цельсія. Сукупність цих моделей дозволяє точно моделювати як рух робота, так і його взаємодію з навколишнім середовищем через сенсори.

На рисунку 3.6 представлено схему взаємозв'язків між основними апаратними компонентами системи управління мобільного робота, орієнтованої на реалізацію функції автономного паркування. Кожен блок відображає окремий функціональний елемент і тип інтерфейсу, який забезпечує передачу даних або сигналів між модулями.

Ультразвукові датчики підключаються до плати керування Arduino Uno через цифрові входи/виходи (GPIO), що дозволяє здійснювати імпульсну генерацію та зчитування ехо-сигналів для вимірювання відстаней. Енкодери двигунів передають інформацію про швидкість і положення обертання через цифрові входи з перериванням, що забезпечує високу точність під час обчислення кутової швидкості в реальному часі. Для управління приводами моторів використовуються PWM-сигнали з Arduino Uno, які подаються на драйвер L298N, що виконує посилення та напрямок обертання

електродвигунів. OLED-дисплей, який забезпечує виведення інформації, підключено через інтерфейс I2C (лінії SDA та SCL), що забезпечує компактність і мінімальне використання пінів контролера.

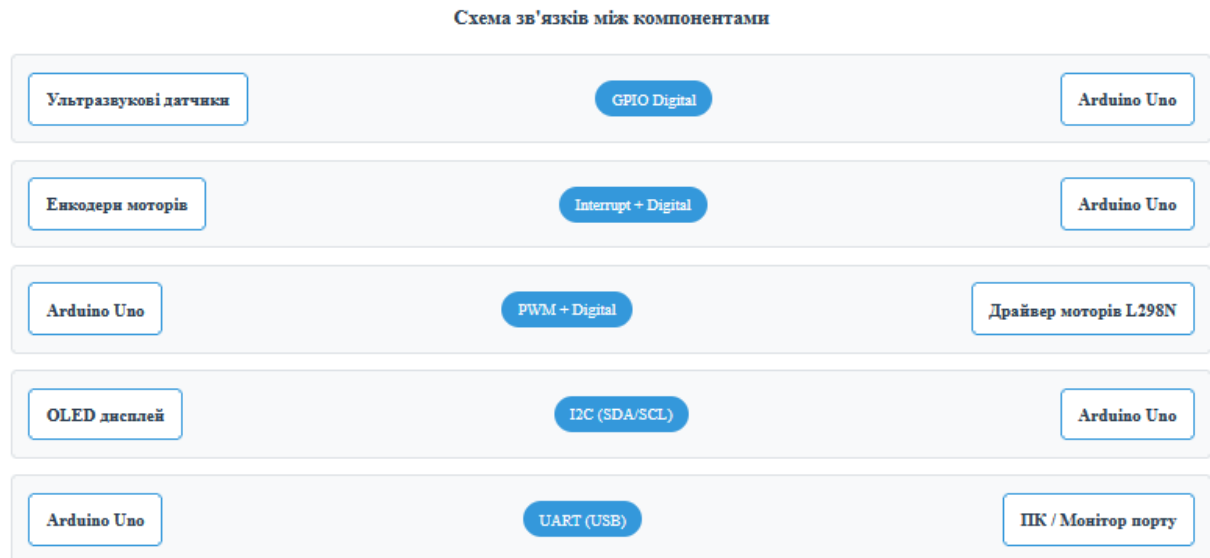


Рисунок 3.6 – Схема зв'язків між компонентами

Для зв'язку із зовнішнім середовищем, включаючи комп'ютер або моніторинг даних, застосовується інтерфейс UART (через USB), який забезпечує серійну передачу інформації з плати Arduino Uno на ПК. Ця конфігурація реалізує інтегровану систему з ефективною передачею даних між датчиками, виконавчими механізмами й елементами керування, забезпечуючи повноцінне функціонування прототипу автономного паркувального робота.

Інтерфейс	Протокол	Швидкість	Призначення	Конфігурація
GPIO Digital	TTL 5V	До 8 МГц	Ультразвукові датчики	Trigger/Echo піни
PWM	Широтно-імпульсна модуляція	490 Гц	Управління швидкістю моторів	0-255 (8-bit)
I2C	Two-Wire Interface	100/400 кГц	OLED дисплей, сенсори	SDA/SCL + адреса пристрою
UART	Асинхронний серійний	9600-115200 бод	Дебаг, телеметрія	8N1, без контролю потоку
Interrupt	Апаратні переривання	Миттєво	Енкодери, аварійна зупинка	RISING/FALLING edge

Рисунок 3.7 – Технічні специфікації пристроїв

На рисунку 3.7 представлено таблицю, що деталізує технічні характеристики й конфігурації інтерфейсів, які застосовуються для реалізації системи управління мобільним роботом з автономною функцією паркування. Кожен інтерфейс охоплює специфічний протокол, тип використання, швидкісні характеристики та відповідну конфігурацію.

Інтерфейс GPIO Digital базується на протоколі TTL з рівнем напруги 5 В і забезпечує частоту до 8 МГц, що дозволяє ефективно зчитувати сигнали від ультразвукових датчиків. Його конфігурація включає стандартні пари пінів типу Trigger/Echo, які застосовуються для генерації та прийому імпульсів.

PWM (широтно-імпульсна модуляція) використовується для керування швидкістю моторів і працює з частотою 490 Гц, що є типовим значенням для Arduino Uno. Діапазон керування охоплює значення від 0 до 255 з 8-бітною розрядністю, що дозволяє точно регулювати інтенсивність сигналів на виконавчі пристрої.

I2C – це інтерфейс типу Two-Wire Interface зі швидкістю 100 або 400 кГц, який забезпечує ефективну взаємодію з OLED-дисплеєм і деякими сенсорами. Його конфігурація передбачає використання стандартних ліній SDA та SCL з обов'язковим визначенням адреси пристрою.

UART є асинхронним серійним інтерфейсом із типовою швидкістю передавання даних у межах 9600–115200 бод, який застосовується для обміну даними з ПК, модулем телеметрії чи інтерфейсом налагодження. Конфігурація цього інтерфейсу зазвичай включає формат 8N1: вісім біт даних, відсутність біту парності та один стоп-біт.

Інтерфейс Interrupt використовує апаратні переривання, які спрацьовують миттєво при зміні стану сигналу, зокрема на фронтах сигналу (RISING/FALLING edge). Такий механізм є критично важливим для обробки імпульсів від енкодерів та виявлення аварійних подій у реальному часі.

Загалом таблиця відображає взаємозв'язок між апаратною конфігурацією та функціональними завданнями системи, що дозволяє

забезпечити синхронізовану роботу сенсорного комплексу, виконавчих механізмів і модулів обробки сигналів в умовах автономного функціонування мобільного робота.

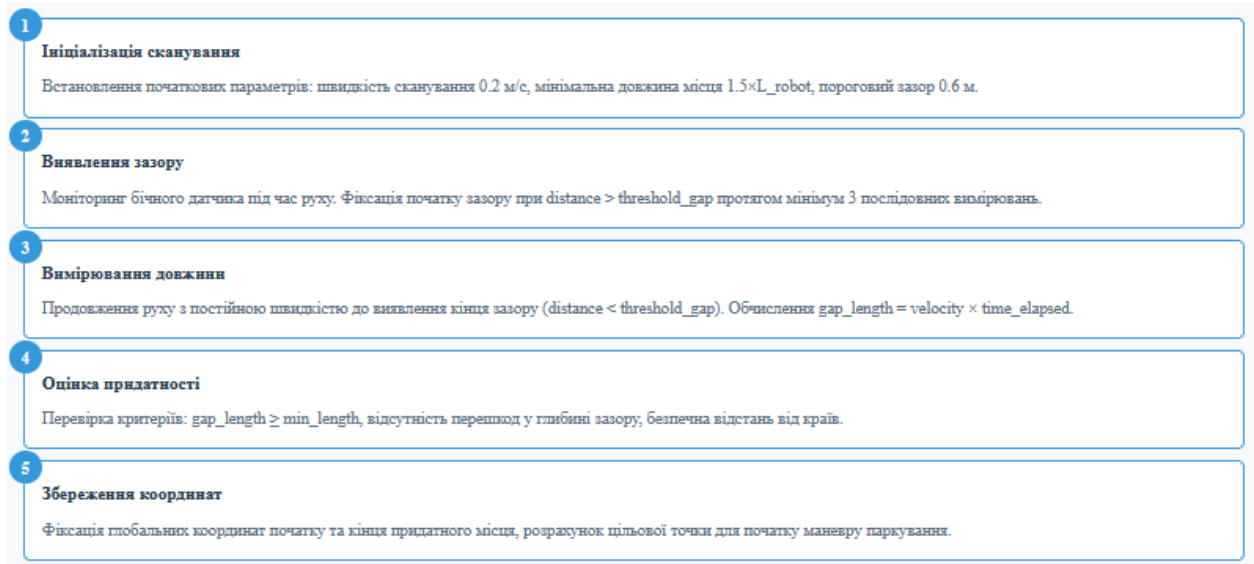


Рисунок 3.8 – Алгоритм виявлення паркувального місця

Рисунок 3.8 ілюструє поетапну процедуру виявлення придатного паркувального місця мобільним роботом, яка є ключовою складовою функціоналу автономного паркування. У структурованому вигляді подано логіку алгоритму, що послідовно реалізується на основі даних ультразвукового сканування під час руху транспортного засобу.

Перший етап передбачає ініціалізацію параметрів сканування, які визначають швидкість переміщення та просторові обмеження для виявлення вільного місця. Зокрема, задається мінімальна довжина можливого місця для паркування та встановлюється порогове значення відстані до перешкоди, що дозволяє визначити наявність зазору в зоні сканування.

Далі виконується моніторинг поточних показників від сенсора в режимі реального часу. Виявлення зазору базується на перевищенні значення відстані над заданим порогом упродовж щонайменше трьох послідовних вимірювань, що дозволяє уникнути помилок через короточасні артефакти або шум.

На третьому етапі, після виявлення початку зазору, робот продовжує рух з постійною швидкістю, одночасно вимірюючи тривалість відстані до моменту завершення зазору, коли значення відстані знову падає нижче порогу. Таким чином обчислюється загальна довжина потенційного паркувального місця.

Наступним кроком є оцінка придатності виявленої ділянки. Перевіряється, чи відповідає довжина місця мінімальним вимогам, чи відсутні перешкоди вглибині та чи знаходиться місце на безпечній відстані від країв дорожнього коридору. Такий підхід дозволяє оцінити як геометричну відповідність, так і безпечність виконання маневру.

Завершальний етап алгоритму полягає у збереженні координат меж зазору, які згодом використовуються для побудови траєкторії паркування. Обчислюється центральна точка, що слугує орієнтиром для початку виконання маневру заїзду в паркувальну зону.

Описана послідовність дозволяє мобільному роботу автономно і з високим рівнем надійності визначати придатні місця для паркування в реальному середовищі, враховуючи просторові обмеження, точність сенсорного вимірювання та необхідність безпечного завершення маневру.

Рівень	Умова спрацювання	Дія системи	Час реакції
Попередження	Відстань < 30 см	Звуковий сигнал, зменшення швидкості	< 100 мс
Критичний	Відстань < 15 см	Аварійна зупинка, блокування руху	< 50 мс
Аварійний	Відстань < 5 см або втрата сенсора	Миттєва зупинка, відключення моторів	< 20 мс
Системний	Втрата зв'язку, перегрів, низький заряд	Повна зупинка, режим відновлення	< 200 мс

Рисунок 3.9 – Багаторівнева система захисту

Зображена на рисунку 3.9 таблиця відображає багаторівневу систему реагування системи автономного паркування мобільного робота на критичні події, пов'язані з наближенням до перешкод або втратою сенсорного контролю. Вона деталізує чотири рівні небезпеки, кожен з яких

характеризується специфічними умовами спрацювання, відповідною реакцією системи та граничним часом, необхідним для ініціації захисних дій.

Перший рівень, попереджувальний, активується при зменшенні відстані до перешкоди менш ніж 30 см. У цьому випадку система генерує звуковий сигнал і автоматично знижує швидкість руху, що забезпечує завчасне реагування без радикального втручання в керування. Час реакції системи в цьому режимі не перевищує 100 мс, що дозволяє уникати небезпечних зближень при повільному русі або маневруванні.

Критичний рівень визначається наближенням до об'єкта на відстань менше 15 см. У таких умовах активується аварійна зупинка з блокуванням коліс або приводу, що запобігає зіткненню. Реакція системи має бути миттєвою – менш ніж за 50 мс, що дозволяє зберігати безпеку навіть у разі раптового виявлення перешкоди.

Аварійний рівень настає, коли відстань до об'єкта зменшується до менше ніж 5 см або коли відбувається втрата сигналу від сенсора, наприклад, через апаратну несправність. У такому випадку система зупиняє рух негайно, відключаючи мотор, що зменшує ризик фізичного зіткнення. Реакція системи тут становить менше ніж 20 мс, що відповідає вимогам реального часу для захисту обладнання та навколишнього середовища.

Найвищий, системний рівень спрацювання, застосовується в умовах позаштатної ситуації, наприклад, при втраті живлення, перезапуску мікроконтролера або критичних збоїв у програмному забезпеченні. У цьому випадку система автоматично переходить у безпечний режим, повністю зупиняється і відключає всі активні механізми. Час реакції в такій ситуації обмежено 200 мс, що дозволяє зафіксувати стан стабільності системи та уникнути каскадного розповсюдження помилок.

Загалом дана модель є основою для реалізації надійної, багаторівневої архітектури безпеки у вбудованих системах паркування, орієнтованих на реальний час і роботу в умовах обмеженого сенсорного контролю.

Процес	Пріоритет	Частота виконання	Час виконання	Ресурс CPU (%)
Зчитування сенсорів	Високий	50 Гц	2 мс	10%
Обробка даних	Високий	20 Гц	5 мс	15%
Планування траєкторії	Середній	10 Гц	8 мс	20%
Управління моторами	Високий	100 Гц	1 мс	25%
Інтерфейс користувача	Низький	5 Гц	3 мс	5%
Діагностика	Низький	1 Гц	10 мс	10%
Резерв системи	-	-	-	15%

Рисунок 3.10 – Розподіл обчислювальних ресурсів

Зображена на рисунку 3.10 таблиця ілюструє розподіл обчислювальних ресурсів центрального процесора між основними процесами в системі керування паркуванням мобільного робота. Вона охоплює характеристики пріоритету, частоти виконання, часу обробки та відповідного навантаження на процесор (у відсотках від загального ресурсу CPU) для кожного функціонального елементу системи.

Зчитування сенсорів є високопріоритетним процесом із частотою 50 Гц і коротким часом виконання в 2 мс. Це забезпечує регулярне оновлення інформації з ультразвукових та інших датчиків, що є критичним для роботи системи в реальному часі. Аналогічно, обробка даних має високий пріоритет, але виконується з меншою частотою – 20 Гц, потребуючи 5 мс на виконання кожного циклу. Ці два компоненти спільно споживають 25% ресурсу CPU, що свідчить про їхню важливість у загальній архітектурі керування.

Планування траєкторії має середній пріоритет, виконується з частотою 10 Гц і часом обробки 8 мс, споживаючи 20% CPU. Це обґрунтовано тим, що оновлення маршруту паркування не потребує надвисокої частоти, проте потребує значних обчислювальних ресурсів для забезпечення безпечного маневрування.

Управління моторами – найчастіше виконувана задача з частотою 100 Гц і мінімальним часом обробки 1 мс. Вона має високий пріоритет і споживає найбільшу частку CPU – 25%, що зумовлено необхідністю забезпечення

стабільного і точного керування рухом коліс у режимі реального часу.

Інтерфейс користувача і діагностика, навпаки, мають низький пріоритет. Вони функціонують на частоті 5 і 1 Гц відповідно, маючи мінімальний вплив на продуктивність системи. Їх сумарне споживання ресурсів не перевищує 15%, що дозволяє фонове виконання без порушення критичних задач.

Резерв системи у 15% є важливим елементом для забезпечення стабільності та аварійного реагування. Він дозволяє компенсувати пікові навантаження або виконувати непередбачувані задачі без деградації основного функціоналу.

Узагальнюючи, така структура розподілу ресурсів демонструє добре збалансовану систему реального часу, де найбільші ресурси виділено на критичні для функціонування компоненти, що забезпечують точне позиціонування, стабільне керування та своєчасне реагування в умовах змінного навколишнього середовища.

Розроблена логічна модель забезпечує комплексний підхід до проектування системи управління парковкою мобільного робота. Багаторівнева архітектура дозволяє ефективно розділити відповідальність між компонентами, забезпечити високу надійність та можливості для майбутнього розширення функціональності. Використання стандартизованих інтерфейсів та протоколів гарантує сумісність з широким спектром апаратних платформ та можливість інтеграції з існуючими системами.

3.3 Моделювання в MATLAB

Рисунок 3.10 відображає результат симуляції процесу паркування мобільного робота в середовищі MATLAB (лістинг коду `ParkingSystemSimulation.m` представлений в додатку Б). На графіку зображено траєкторію руху автономного транспортного засобу, що реалізує алгоритм об'їзду перешкоди й пошуку відповідного паркувального місця.

Координатна система використовується в сантиметрах по осях X та Y , де значення імітують положення робота в реальному фізичному просторі.

Траєкторія руху позначена синьою лінією з колами в точках дискретного переміщення, які відповідають етапам розрахованого маршруту, сформованого на основі вхідних даних з сенсорів. Перешкода, яку виявлено в полі зору ультразвукових датчиків, позначена великим червоним хрестом. Її виявлення ініціює адаптацію траєкторії та повторне планування маршруту для уникнення зіткнення.

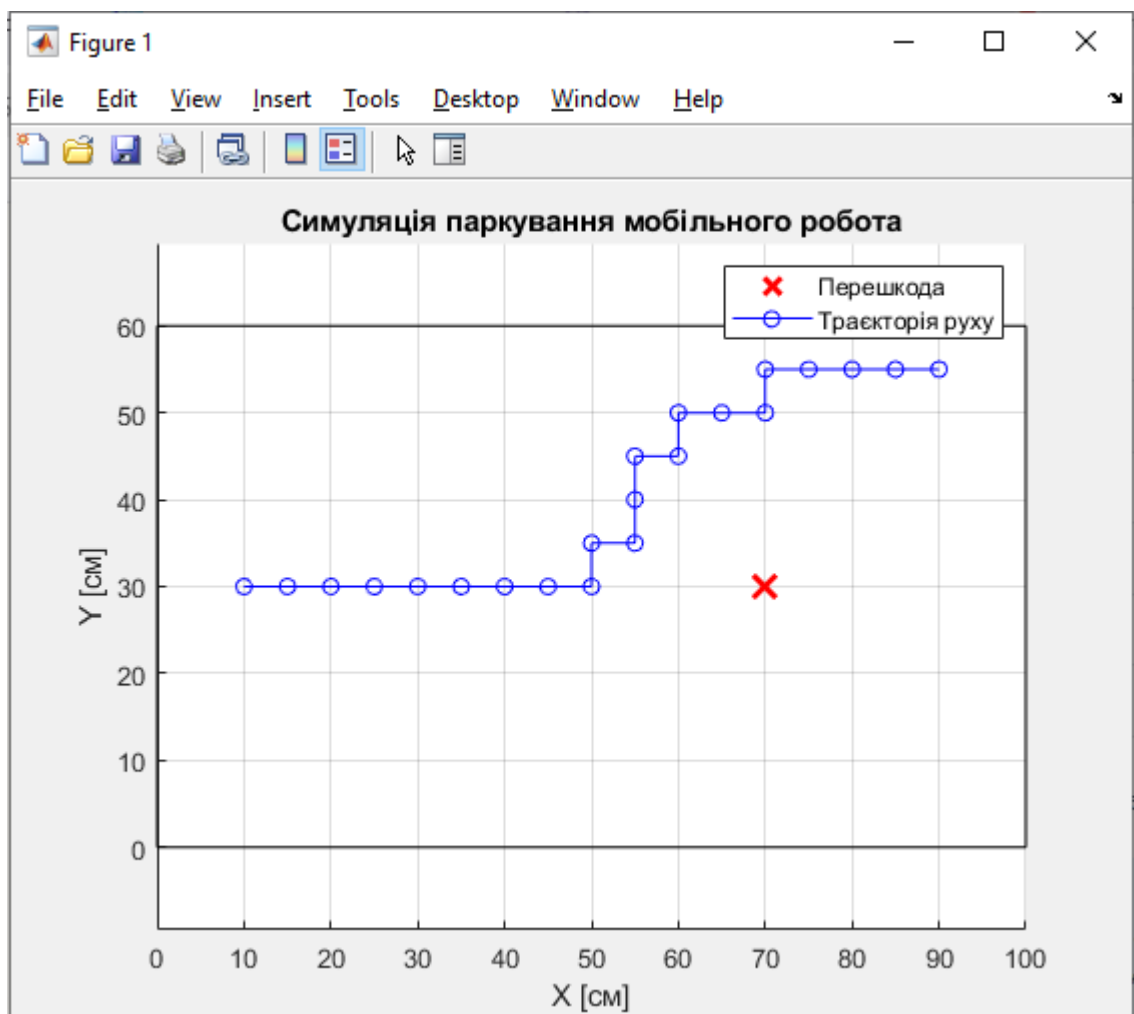


Рисунок 3.11 – Результат симуляції процесу паркування мобільного робота в середовищі MATLAB

Загальний хід траєкторії демонструє здатність системи реагувати на зміну конфігурації середовища, виконувати маневри об'їзду та забезпечувати

безпечне досягнення цільової зони паркування. Побудова такої симуляції дозволяє протестувати логіку роботи алгоритму на етапі програмного моделювання та підготувати систему до переходу на реальну платформу з мікроконтролерами й фізичними сенсорами.

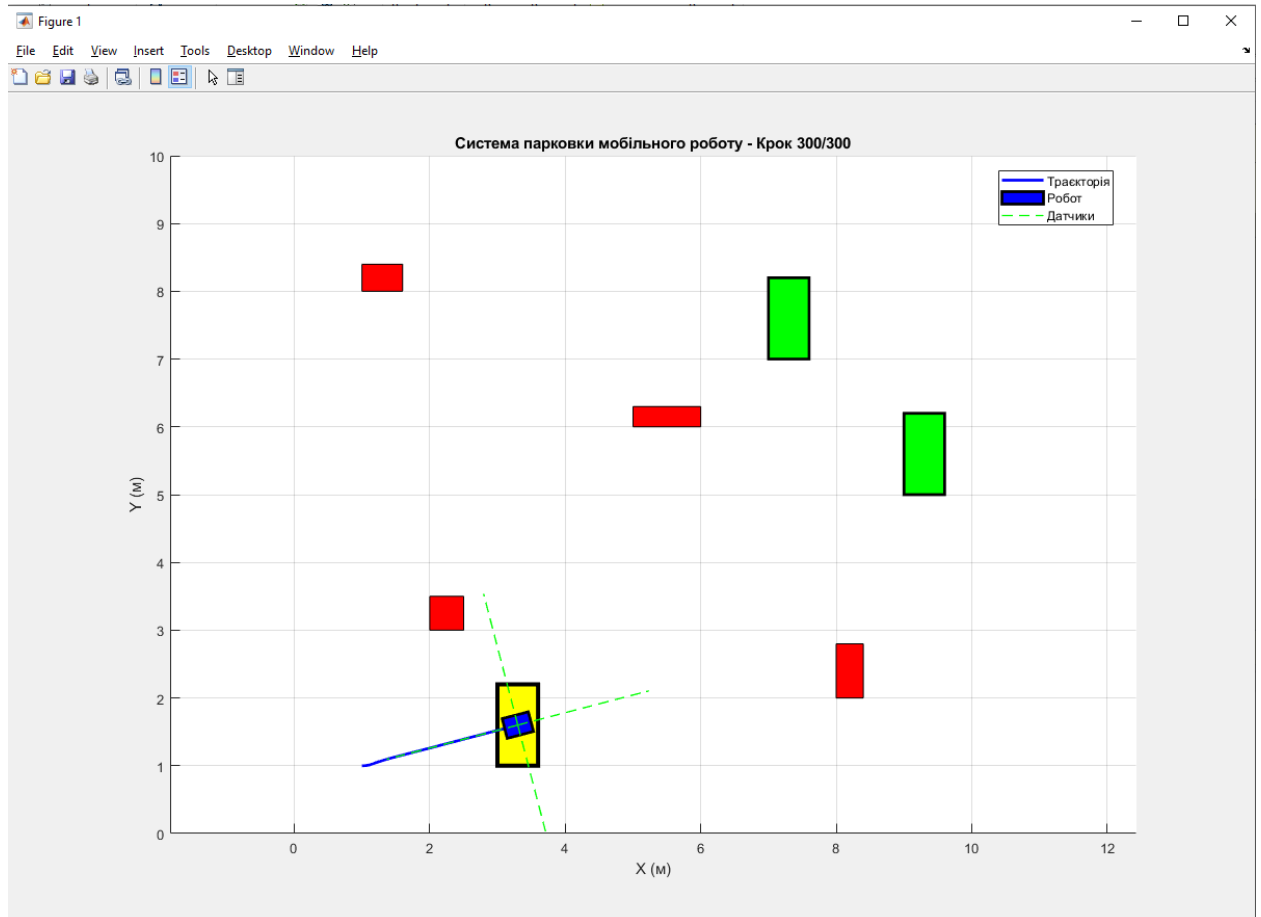


Рисунок 3.12 – Завершення симуляції процесу паркування мобільного робота на двовимірному полі

Рисунок 3.12 демонструє завершення симуляції процесу паркування мобільного робота на двовимірному полі з перешкодами та паркувальними місцями. Система успішно змоделювала поведінку автономного робота з урахуванням ультразвукових вимірювань, планування траєкторії та виконання керуючих дій (лістинг коду файлу Pss_v1.m представлений у додатку Б).

На графіку візуалізовано траєкторію руху робота синім кольором, що свідчить про динамічне оновлення позиції протягом усієї симуляції. Тіло

робота подано у вигляді прямокутника, який відображає орієнтацію та габарити пристрою. Жовта рамка вказує на цільове паркувальне місце, успішно досягнуте наприкінці руху. Червоні прямокутники ілюструють розміщення фіксованих перешкод, які робот уникав завдяки плануванню обхідної траєкторії. Зелені прямокутники представляють доступні паркомісця, з яких система обрала найближче допустиме за габаритами.

Додатково, штрихові зелені лінії відображають результати ультразвукового сканування – напрямки та довжини променів, які вказують на наявність або відсутність об'єктів у межах зони досяжності датчиків. Це підтверджує правильну роботу сенсорної підсистеми.

Загалом рисунок свідчить про ефективність реалізованої системи автономного паркування та її здатність адаптуватися до складного середовища, дотримуючись правил безпеки й уникнення зіткнень.

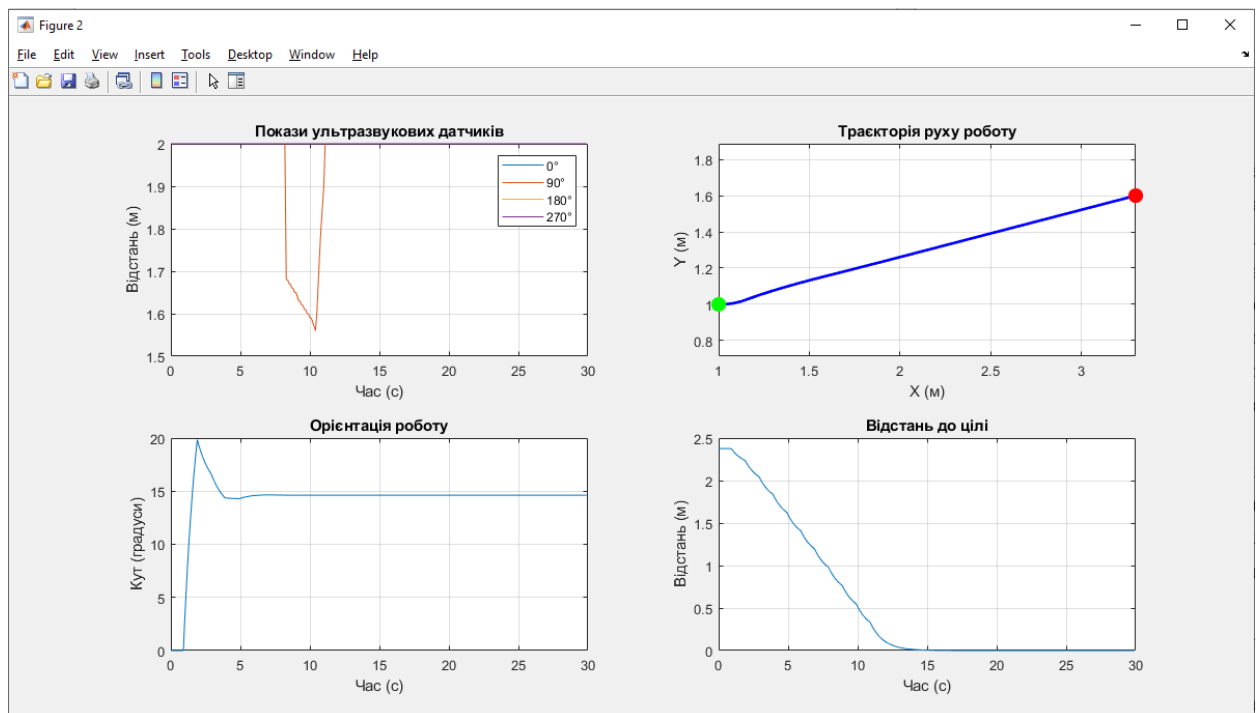


Рисунок 3.13 – Аналітичні результати симуляції паркування мобільного робота

Рисунок 3.13 відображає аналітичні результати симуляції паркування мобільного робота на основі зібраних даних протягом усього експерименту.

Графіки охоплюють ключові параметри, що визначають динаміку руху, поведінку сенсорів та ефективність досягнення цільової позиції.

Покази ультразвукових датчиків (верхній лівий графік).

Відображено зміни вимірної відстані протягом часу для трьох активних сенсорів (0° , 90° , 270°). Випади відхилень (наприклад, зменшення відстані на 270°) свідчать про наближення до перешкоди під час руху.

Траєкторія руху роботу (верхній правий графік).

Показано фактичний шлях від початкової до цільової точки. Старт позначений зеленим колом, кінець – червоним, що підтверджує досягнення паркувального місця. Крива демонструє плавний та спрямований рух із мінімальним відхиленням.

Орієнтація роботу (нижній лівий графік).

Зображено динаміку зміни кута повороту. На початку спостерігається активна корекція напрямку, після чого значення стабілізується, що відповідає прямолінійному підходу до цілі.

Відстань до цілі (нижній правий графік).

Видно поступове зменшення відстані до цільової точки. Крива плавно наближається до нуля, що свідчить про точне та безперервне слідування маршрутом.

Загалом ці графіки демонструють стабільну, узгоджену роботу системи управління, планування та сенсорного зворотного зв'язку, підтверджуючи успішність реалізованого алгоритму автономного паркування.

На рисунку 3.13 фрагмент виводу вікна команд MATLAB ілюструє завершення симуляції системи паркування мобільного робота. На основі виведених даних можна зробити такі висновки:

- початкова позиція: (1.00, 1.00) – це координати, з яких почав рух робот;
- кінцева позиція: (3.30, 1.60) – фактична фінальна координата робота після завершення симуляції;
- цільова позиція: (3.30, 1.60) – центр обраного паркувального місця,

розрахований алгоритмом;

- точність паркування: 0.000 м – робот припаркувався ідеально точно, без відхилення від цілі;
- загальна відстань: 2.38 м – сумарний шлях, пройдений роботом до паркувального місця;
- час симуляції: 30.0 с – тривалість моделювання відповідно до заданого інтервалу.

```

207 -         angle_error = angle_error + 2
208 -     end
209 -     v = min(Kp_linear * distance_erro
210 -     omega = Kp_angular * angle_error;
211 -     if abs(angle_error) > pi/4
212 -         v = v * 0.5;
213 -     end
214 - end
215

```

```

Command Window

=== РЕЗУЛЬТАТИ СИМУЛЯЦІЇ ===
Початкова позиція: (1.00, 1.00)
Кінцева позиція: (3.30, 1.60)
Цільова позиція: (3.30, 1.60)
Точність паркування: 0.000 м
Загальна відстань: 2.38 м
Час симуляції: 30.0 с
РЕЗУЛЬТАТ: Паркування УСПІШНЕ!
fx >>

```

Рисунок 3.14 – Результат симуляції

Результат симуляції позначений як "Паркування УСПІШНЕ!", що підтверджує коректність роботи всіх компонентів: траєкторного планувальника, контролера руху, сенсорного модулю та логіки вибору паркувального місця.

3.4 Просторова модель мобільного робота

На рисунку 3.14 зображена просторова модель мобільного робота відображає конструктивні особливості пристрою, необхідні для його функціонування в умовах автономного переміщення та паркування. Система

передбачає наявність корпусу, в межах якого інтегровано керувальний модуль Arduino, встановлений ближче до передньої частини. У центральній частині розміщено акумуляторну батарею типу Li-ion, що забезпечує енергоживлення всіх функціональних компонентів системи. На кожній зі сторін корпусу передбачено монтаж ультразвукових сенсорів, орієнтованих у чотирьох напрямках: вперед, назад, ліворуч та праворуч. Така конфігурація дає змогу формувати повну картину навколишнього середовища, що критично важливо для завдань виявлення перешкод та точного маневрування в умовах обмеженого простору.

Визначальні геометричні параметри мобільного робота становлять 300 мм у довжину, 200 мм у ширину та 120 мм у висоту, що забезпечує йому компактність і придатність до експлуатації в середовищах із просторовими обмеженнями. Загальна маса, включно з елементом живлення, становить 1.2 кг, що відповідає вимогам до мобільності та керованості конструкції. Колісна база має довжину 180 мм, а діаметр провідних коліс становить 65 мм, що забезпечує належний рівень прохідності та дозволяє реалізовувати алгоритми керування рухом із високою точністю.

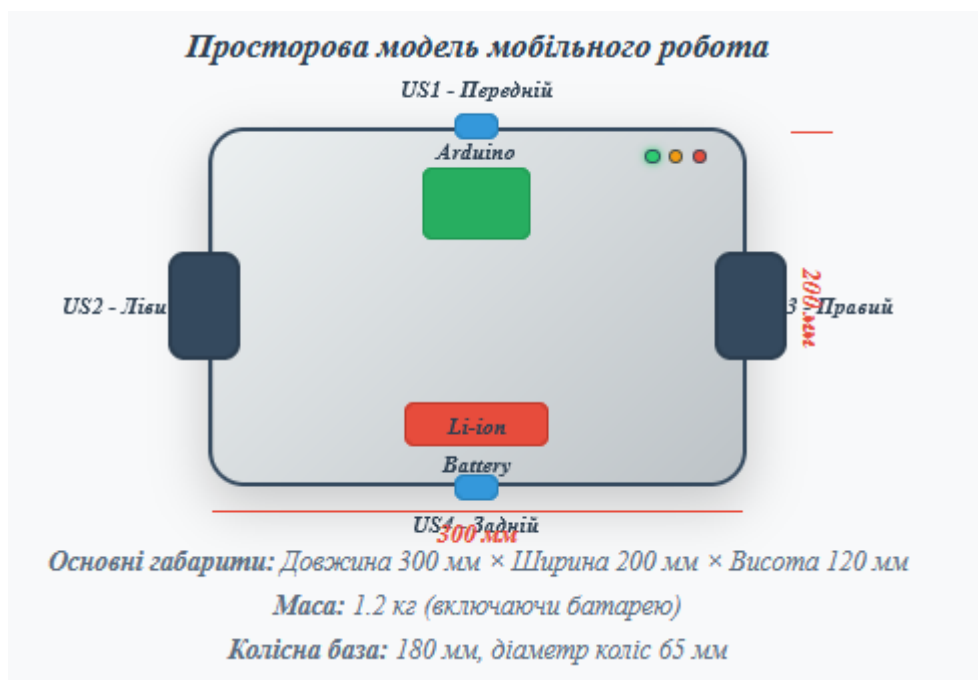


Рисунок 3.15 – Просторова модель мобільного робота

Параметр	Значення	Одиниці	Примітки
Габаритні розміри (Д×Ш×В)	300×200×120	мм	Без врахування виступаючих елементів
Маса платформи	1.2	кг	Повна маса з акумулятором
Колісна база	180	мм	Відстань між центрами коліс
Діаметр коліс	65	мм	Гумові колеса з протектором
Максимальна швидкість	0.5	м/с	При номінальній напрузі
Мінімальний радіус повороту	90	мм	Поворот на місці
Точність позиціонування	±5	мм	В ідеальних умовах
Час автономної роботи	4-6	год	Залежно від інтенсивності використання

Рисунок 3.16 – Основні параметри платформи

№	Компонент	Модель/Артикул	Кількість	Ціна за одиницю	Загальна вартість	Постачальник
1	Мікроконтролер	Arduino Uno R3	1	\$25	\$25	Arduino Store
2	Ультразвукові датчики	HC-SR04	4	\$2	\$8	AliExpress
3	Мотор-редуктори	TT Motor 3-6V	2	\$3	\$6	Local electronics
4	Драйвер моторів	L298N	1	\$3	\$3	AliExpress
5	Енкодери	Optical encoder 400p/r	2	\$5	\$10	Pololu
6	Акумулятор	Li-ion 18650 3.7V 2200mAh	3	\$4	\$12	Battery store
7	Контролер заряду	TP4056	1	\$1	\$1	AliExpress
8	Стабілізатор напруги	LM2596 DC-DC	2	\$2	\$4	AliExpress
9	OLED дисплей	SSD1306 128x64	1	\$3	\$3	Adafruit
10	LED індикатори	5mm LED RGB	3	\$0.5	\$1.5	Local electronics
11	Зумер	Passive buzzer 5V	1			

Рисунок 3.17 – Специфікація компонентів

У межах реалізації системи автономного паркування мобільного робота було використано набір електронних та механічних компонентів, необхідних для створення функціонального прототипу. До складу системи (рисунку 3.15, 3.16) входять мікроконтролер Arduino Uno R3, ультразвукові датчики HC-SR04 для вимірювання відстаней до перешкод, мотор-редуктори TT Motor 3–6V у парі з драйвером двигунів L298N, енкодери для відстеження обертів

коліс, акумулятор Li-Ion ємністю 2200 мА·год із відповідним контролером заряду TP4056, стабілізатор живлення, OLED-дисплей для виведення інформації, світлодіодні індикатори та звуковий сигналізатор.

Загальна вартість компонентів не перевищує 80 доларів США, що дозволяє вважати систему бюджетною й придатною для навчальних або прототипних цілей. Живлення забезпечується літій-іонною батареєю, а управління здійснюється за допомогою Arduino та відповідного програмного забезпечення.

Мобільний робот має габарити 300×200×120 мм, масу 1.2 кг, колісну базу 180 мм та діаметр коліс 65 мм із гумовим покриттям. Максимальна швидкість пересування становить 0.5 м/с за мінімального навантаження. Конструкція дозволяє здійснювати поворот на місці з мінімальним радіусом 90 мм. Точність позиціонування в ідеальних умовах сягає ± 5 мм. Автономна робота системи триває від 4 до 6 годин залежно від навантаження, інтенсивності сканування та частоти руху. Усі ці параметри роблять мобільного робота придатним для виконання завдань навігації та паркування у структурованому середовищі з динамічним обліком просторових обмежень.

4 ПРОЕКТУВАННЯ ПРОГРАМНО-АПАРАТНИЙ ПРОТОТИП СИСТЕМИ

4.1 Архітектура програмної частини прототипу системи

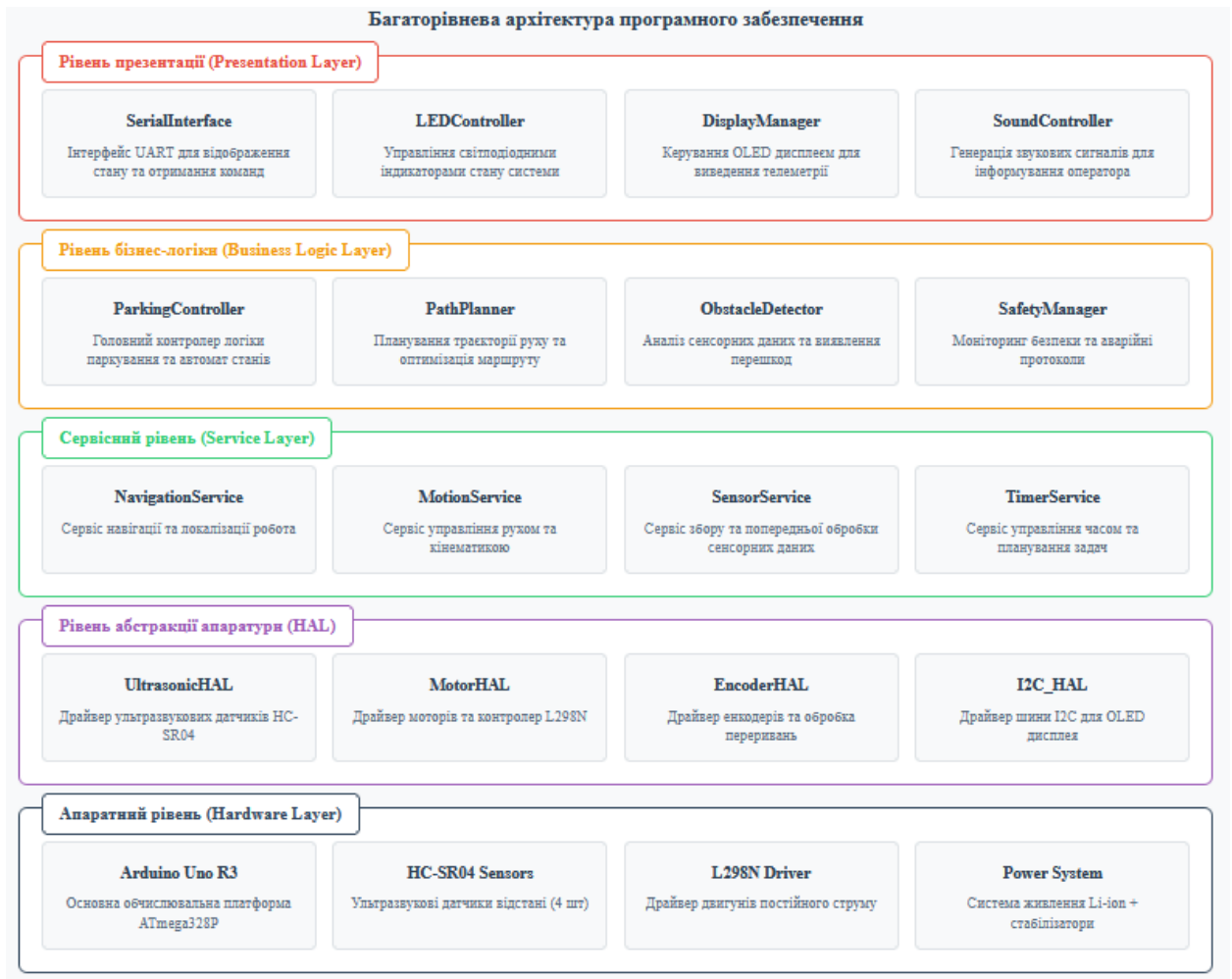


Рисунок 4.1 – Архітектура програмного забезпечення

На рисунку 4.1 представлено багаторівневу архітектуру програмного забезпечення системи управління мобільним роботом, яка демонструє модульний поділ функціональності за принципом розділення відповідальності. Архітектура структурована на шість логічних рівнів, кожен з яких відповідає за окремі аспекти роботи системи – від взаємодії з фізичним обладнанням до виводу інформації користувачу.

Апаратний рівень (Hardware Layer) містить фізичні компоненти

системи, включаючи мікроконтролер Arduino Uno R3, ультразвукові сенсори HC-SR04, драйвер двигунів L298N та енергетичну підсистему. Саме ці компоненти формують матеріальну основу мобільного робота.

Рівень абстракції апаратури (HAL – Hardware Abstraction Layer) включає драйвери низького рівня, які реалізують базові функції керування сенсорами, двигунами, енкадерами та інтерфейсом I2C. Цей рівень забезпечує уніфікований доступ до фізичних пристроїв для вищих шарів системи.

Сервісний рівень (Service Layer) відповідає за обробку основних дій у системі: навігацію, генерацію команд руху, збір і фільтрацію даних із сенсорів, а також синхронізацію подій у часі.

Рівень бізнес-логіки (Business Logic Layer) включає високорівневі алгоритми керування поведінкою робота. Сюди входить модуль управління паркуванням, планування маршруту, виявлення перешкод та моніторинг критичних станів.

Рівень представлення (Presentation Layer) реалізує засоби комунікації з користувачем. Це модулі керування світлодіодами, дисплеєм, звуковими сигналами, а також обмін через UART для виводу системної інформації.

У сукупності дана архітектура забезпечує надійне, масштабоване та структуроване програмне середовище для реалізації інтелектуального автономного управління мобільним роботом.

На рисунку 4.2 зображено структуровану діаграму взаємодії компонентів, що складають програмну систему керування мобільним роботом. Архітектура організована у вигляді функціональних модулів, кожен з яких виконує чітко визначену роль у забезпеченні автономної роботи пристрою. Центральне місце займає ядро системи, відповідальне за управління основними процесами, планування задач, обробку помилок, конфігурацію та реєстрацію даних, що забезпечує узгоджену роботу всіх компонентів.

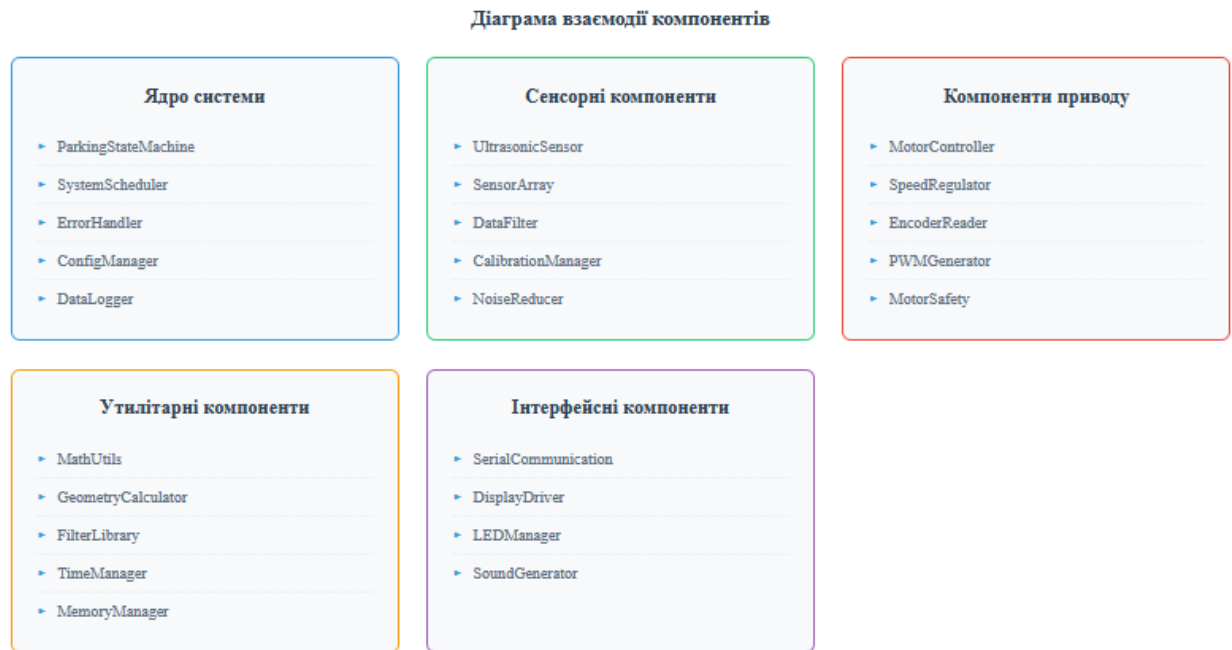


Рисунок 4.2 – Діаграма взаємодії компонентів

Сенсорна підсистема містить модулі для зчитування інформації з ультразвукових сенсорів, обробки масиву даних, фільтрації сигналів, калібрування та зменшення шумів, що підвищує точність просторового орієнтування робота в умовах динамічного середовища. Підсистема приводу включає функціональні блоки для керування моторами, регулювання швидкості, зчитування сигналів з енкодерів, генерації імпульсів для приводу та забезпечення захисту приводу.

Окрему категорію утворюють утилітарні компоненти, які реалізують математичні обчислення, розрахунок геометричних характеристик, управління пам'яттю та часом, а також використання готових бібліотек. Інтерфейсна частина системи реалізує комунікацію з користувачем за допомогою серійного порту, візуалізацію даних на дисплеї, індикацію станів через світлодіоди та генерацію акустичних сигналів. Така багатокomпонентна організація дозволяє досягти високої гнучкості, масштабованості та надійності програмного забезпечення в контексті автономного керування мобільним роботом.

4.2 Компоненти програмної частини

Наша система включає:

- ParkingStateMachine – автомат станів системи (рисунок 4.3);
- SensorArray – управління ультразвуковими датчиками (рисунок 4.4);
- MotorController – управління приводами (рисунок 4.5);

```

ParkingStateMachine.h - Виключення станів та переходів

enum ParkingState {
    IDLE,           // Отримання команд
    SCANNING,      // Пошук паркувального місця
    APPROACHING,   // Наближення до цільової позиції
    PARKING,       // Виконання маневру паркування
    PARKED,        // Успішно припарковано
    ERROR_STATE    // Стан помилки
};

class ParkingStateMachine {
private:
    ParkingState currentState;
    unsigned long stateStartTime;
    unsigned long stateTimeout;

public:
    ParkingStateMachine();
    void initialize();
    void update();
    bool transitionTo(ParkingState newState);
    ParkingState getCurrentState();
    void handleTimeout();
    const char* getStateString(ParkingState state);
};

```

Рисунок 4.3 – ParkingStateMachine

```

SensorArray.cpp - Реалізація керування датчиками

#include "SensorArray.h"

SensorArray::SensorArray() {
    sensors[0] = UltrasonicSensor(2, 3); // Передній
    sensors[1] = UltrasonicSensor(4, 5); // Лівий
    sensors[2] = UltrasonicSensor(6, 7); // Правий
    sensors[3] = UltrasonicSensor(8, 9); // Задній
}

void SensorArray::readAllSensors() {
    for(int i = 0; i < 4; i++) {
        sensorData[i].distance = sensors[i].measureDistance();
        sensorData[i].timestamp = millis();
        sensorData[i].isValid = validateReading(i);

        // Застосування медіанного фільтру
        sensorData[i].filteredDistance = medianFilter(i, sensorData[i].distance);
    }
}

bool SensorArray::validateReading(int sensorIndex) {
    float distance = sensorData[sensorIndex].distance;

    // Перевірка діапазону
    if(distance < 2.0 || distance > 400.0) return false;

    // Перевірка швидкості зміни
    if(abs(distance - previousReadings[sensorIndex]) > 50.0) {
        return false;
    }

    previousReadings[sensorIndex] = distance;
    return true;
}

```

Рисунок 4.4 – SensorArray

```

MotorController.h - Інтерфейс управління моторами

class MotorController {
private:
    int leftMotorPWM, rightMotorPWM;
    int leftDir1, leftDir2, rightDir1, rightDir2;
    float currentLeftSpeed, currentRightSpeed;
    unsigned long lastEncoderTime;

public:
    MotorController(int lPWM, int rPWM, int lD1, int lD2, int rD1, int rD2);

    void initialize();
    void setSpeed(float leftSpeed, float rightSpeed);
    void moveForward(float speed);
    void moveBackward(float speed);
    void turnLeft(float speed);
    void turnRight(float speed);
    void stop();
    void emergencyStop();

    // PID регулятор відхилення
    void updatePID();
    float calculatePID(float setpoint, float measured, float& prevError, float& integral);
};

```

Рисунок 4.5 – MotorController

Лістинг коду даних компонентів представлений в додатку Б.

Фрагменти коду на рисунках 4.3 – 4.5 відображають реалізацію ключових компонентів інтелектуальної системи автоматичного паркування мобільного робота. Модуль ParkingStateMachine визначає стани системи та логіку переходів між ними на основі подій, що виникають під час виконання завдання паркування. Реалізація передбачає стани очікування, сканування, наближення, паркування, завершення та обробки помилок, що дозволяє системі адаптувати поведінку відповідно до поточної ситуації.

Клас SensorArray відповідає за зчитування та первинну обробку даних з ультразвукових сенсорів. До його функцій входить вимірювання відстаней, фільтрація недостовірних показників та забезпечення достовірності сенсорних даних, що є критичним для точного виявлення перешкод і вільного простору для паркування. Метод перевірки включає прості обмеження та відкидає аномальні значення, забезпечуючи стабільність сенсорного входу.

Інтерфейс MotorController реалізує функціональність управління приводом робота. Він дозволяє за допомогою ШІМ-сигналів керувати напрямком та швидкістю обертання двигунів, а також реалізує PID-регулятор для забезпечення точного позиціонування. Клас підтримує типові маневри

мобільного робота, включаючи рух уперед, назад, обертання та зупинку. Важливою складовою є обчислення PID-регулювання, що враховує поточну похибку та інтегральну складову для точного контролю.

Загалом ці модулі демонструють структурований підхід до побудови програмно-апаратної системи на основі принципів об'єктно-орієнтованого програмування, модульності та надійності в реальному часі.

4.3 Опис даних та бібліотек підтримки компонентів системи

Структури даних системи представлені на рисунку 4.6.

```

DataStructures.h - Основні структури даних

// Структура для зберігання даних сенсора
struct SensorReading {
    float distance;           // Відстань в см
    float filteredDistance;  // Відфільтроване значення
    unsigned long timestamp; // Час вимірювання
    bool isValid;           // Валідність показання
    uint8_t sensorId;      // Ідентифікатор датчика
};

// Положення та орієнтація робота
struct RobotPose {
    float x, y;              // Координати в мм
    float theta;            // Орієнтація в радіанах
    float confidence;       // Достовірність позиції
    unsigned long timestamp;
};

// Команда руху для моторів
struct MotionCommand {
    float leftSpeed;        // Швидкість лівого мотора (-255 до 255)
    float rightSpeed;       // Швидкість правого мотора
    unsigned int duration;  // Тривалість команди в мс
    bool isEmergency;      // Прапорець аварійної команди
};

// Опис паркувального місця
struct ParkingSpace {
    float startX, startY;   // Початок простору
    float endX, endY;       // Кінець простору
    float width, length;    // Розміри простору
    float approachAngle;    // Кут підходу
    bool isOccupied;        // Зайнятість місця
    uint8_t quality;        // Якість місця (0-100)
};

// Конфігурація системи
struct SystemConfig {
    float maxSpeed;         // Максимальна швидкість
    float minParkingGap;    // Мінімальний зазор для паркування
    float safetyDistance;   // Безпечна відстань
    float wheelBase;        // База коліс
    float robotWidth;       // Ширина робота
    float robotLength;      // Довжина робота
};

```

Рисунок 4.6 – Основні структури даних

Фрагмент коду на рисунку 4.6 містить визначення основних структур даних, що використовуються в системі інтелектуального паркування мобільного робота. Структура `SensorReading` забезпечує зберігання параметрів, отриманих від ультразвукових сенсорів, включаючи вимірювання відстань, відфільтровані значення, часову мітку, достовірність даних та ідентифікатор сенсора. Структура `RobotPose` описує поточну позицію робота у площині XY, орієнтацію (кут повороту) та рівень довіри до вимірюваних координат, що є критично важливим для навігації.

Структура `MotionCommand` задає параметри для керування швидкістю обертання лівого та правого двигунів, тривалість виконання команди та її пріоритетність, що дозволяє системі формувати ефективні траєкторії руху. Для аналізу простору паркування використовується структура `ParkingSpace`, яка включає координати країв вільного простору, ширину, довжину, кут підходу, точку зупинки та якісну оцінку відповідності місця.

Конфігураційні параметри системи визначаються через структуру `SystemConfig`, де задаються обмеження на відстані та геометричні характеристики мобільної платформи, включаючи довжину, ширину, колісну базу та безпечну дистанцію до перешкод. Узагальнено, ці структури забезпечують повноцінне представлення вхідних даних, команд керування, станів середовища та параметрів конфігурації системи, що є основою для побудови надійного алгоритму автоматичного паркування.

Бібліотека	Версія	Призначення	Ключові функції
<code>Arduino.h</code>	Core 1.8.19	Основні функції Arduino	<code>digitalWrite</code> , <code>analogRead</code> , <code>millis</code> , <code>Serial</code>
<code>NewPing.h</code>	1.9.4	Управління ультразвуковими датчиками	<code>ping_cm</code> , <code>ping_median</code> , <code>timer_us</code>
<code>Wire.h</code>	1.0	I2C комунікація	<code>beginTransmission</code> , <code>write</code> , <code>read</code> , <code>endTransmission</code>
<code>Adafruit_SSD1306.h</code>	2.5.7	Управління OLED дисплеєм	<code>begin</code> , <code>clearDisplay</code> , <code>setTextSize</code> , <code>display</code>
<code>EEPROM.h</code>	2.0	Збереження конфігурації	<code>read</code> , <code>write</code> , <code>update</code> , <code>get</code> , <code>put</code>
<code>TimerOne.h</code>	1.1.1	Точний контроль часу	<code>initialize</code> , <code>attachInterrupt</code> , <code>setPeriod</code>
<code>PID_v1.h</code>	1.2.1	PID регулювання швидкості	<code>Compute</code> , <code>SetTunings</code> , <code>SetOutputLimits</code>

Рисунок 4.7 – Використані бібліотеки

На рисунку 4.7 представлено таблицю використаних бібліотек у програмній реалізації системи автоматичного паркування мобільного робота. У ній наведено назви бібліотек, їхні версії, функціональне призначення у проєкті, а також ключові функції, які реалізуються відповідними бібліотеками. Зокрема, до основних включено стандартну бібліотеку Arduino для базового керування, NewPing для роботи з ультразвуковими сенсорами, Wire для реалізації І²C-комунікації, бібліотеку керування OLED-дисплеєм Adafruit_SSD1306, EEPROM для збереження параметрів у енергонезалежній пам'яті, TimerOne для точного контролю часу та PID_v1 для реалізації алгоритму регулювання швидкості руху. Кожна з бібліотек є важливим компонентом архітектури керування, забезпечуючи точну взаємодію між сенсорною підсистемою, виконавчими механізмами та інтерфейсом користувача.

```
MathUtils.cpp - Математичні утиліти

// Нормалізація кута в діапазон [-π, π]
float normalizeAngle(float angle) {
    while(angle > PI) angle -= 2 * PI;
    while(angle < -PI) angle += 2 * PI;
    return angle;
}

// Медіанний фільтр для сенсорних даних
float medianFilter(float values[], int size) {
    // Сортування масиву методом бульбашки
    for(int i = 0; i < size - 1; i++) {
        for(int j = 0; j < size - i - 1; j++) {
            if(values[j] > values[j + 1]) {
                float temp = values[j];
                values[j] = values[j + 1];
                values[j + 1] = temp;
            }
        }
    }
    return values[size / 2];
}

// Розрахунок відстані між двома точками
float calculateDistance(float x1, float y1, float x2, float y2) {
    float dx = x2 - x1;
    float dy = y2 - y1;
    return sqrt(dx * dx + dy * dy);
}
```

Рисунок 4.8 – Алгоритмічні бібліотеки

На рисунку 4.8 представлено фрагмент програмного коду модуля MathUtils.cpp, який реалізує математичні утиліти, що використовуються в системі керування мобільним роботом. Наведено три основні функції: нормалізація кута в межах інтервалу від $-\pi$ до π , застосування медіанного фільтра для згладжування сенсорних вимірювань та обчислення евклідової відстані між двома точками у двовимірному просторі. Ці функції забезпечують підвищення точності навігації та обробки даних, що є критично важливим для стабільної роботи алгоритмів траєкторного планування та ухилення від перешкод.

4.4 Опис контрольних прикладів і результатів тестування

Тестування програмно-апаратного прототипу проводилось за принципом пірамідального підходу: від модульних тестів окремих компонентів до інтеграційних тестів всієї системи. Кожен рівень тестування мав специфічні цілі та критерії успішності.

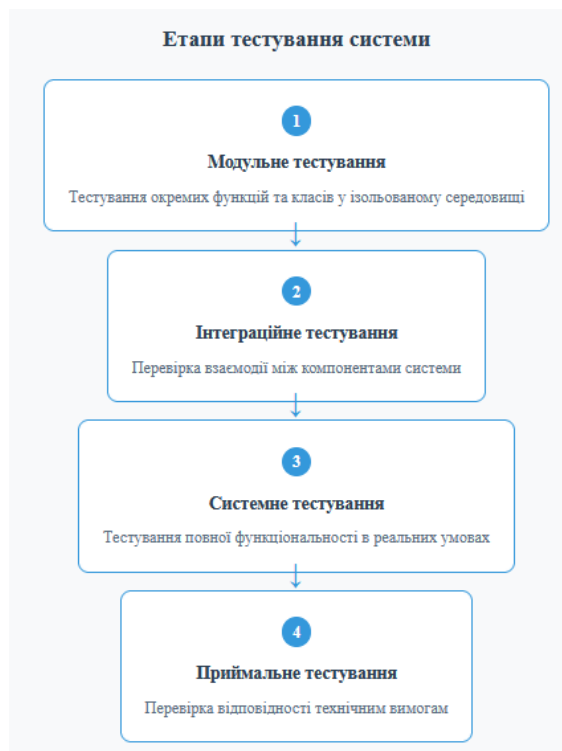


Рисунок 4.9 – Етапи тестування системи

На рисунку 4.9 представлено послідовність етапів тестування системи автономного паркування мобільного робота. Початковий етап включає модульне тестування, спрямоване на перевірку окремих функцій і класів у контрольованому середовищі. Наступним кроком є інтеграційне тестування, яке дозволяє оцінити коректність взаємодії між окремими компонентами системи. Далі виконується системне тестування, що охоплює повну функціональність системи в умовах, наближених до реальних. Завершальним етапом є приймальне тестування, метою якого є перевірка відповідності розробленого рішення технічним вимогам замовника або специфікації. Ця структура тестування забезпечує комплексну перевірку якості, стабільності та відповідності функціоналу системи.

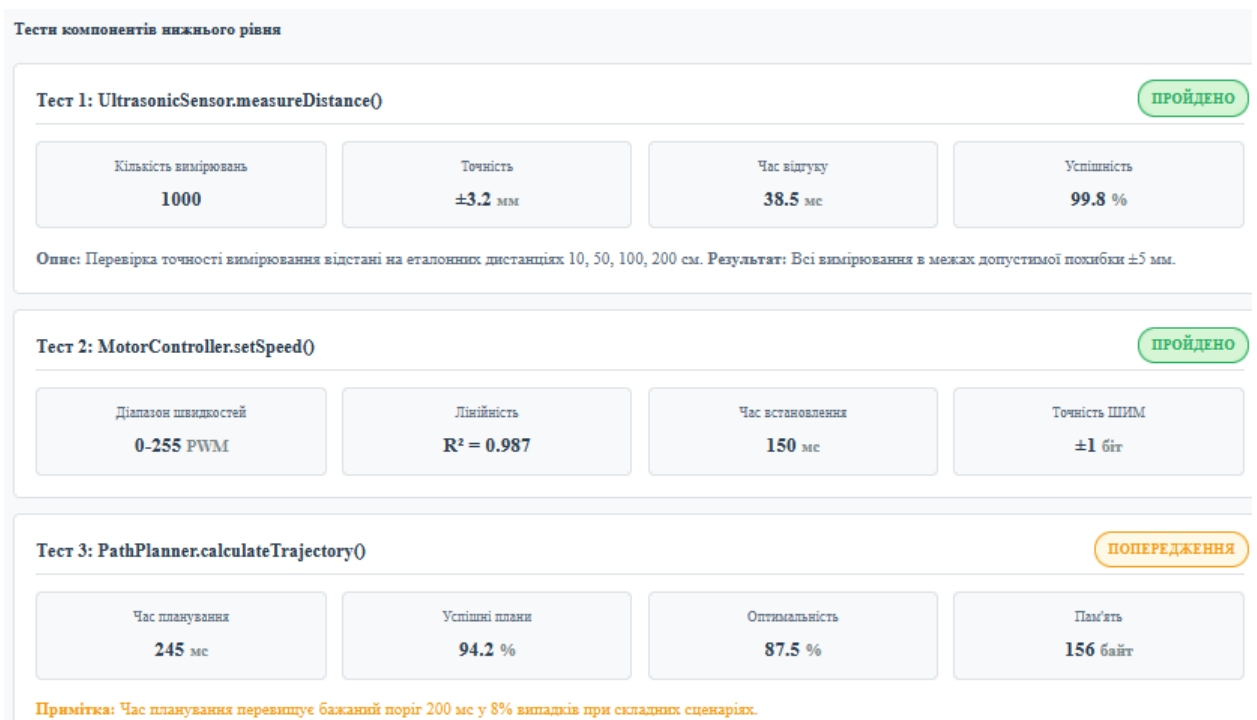


Рисунок 4.10 – Тести компонентів нижнього рівня

На рисунку 4.10 представлено результати модульного тестування компонентів нижнього рівня системи автономного паркування мобільного робота. Тестування модуля ультразвукового сенсора (UltrasonicSensor::measureDistance()) засвідчило високу точність із середнім відхиленням ±3.2 мм та успішністю понад 99 %, що підтверджує стабільність

сенсорного сприйняття в широкому діапазоні відстаней. Функція керування швидкістю двигунів (`MotorController::setSpeed()`) продемонструвала лінійність керуючого сигналу з коефіцієнтом детермінації $R^2 = 0.987$ та високою точністю формування ШІМ, що забезпечує надійне динамічне керування. Однак тестування функції планування траєкторії (`PathPlanner::calculateTrajectory()`) виявило незначне перевищення допустимого часу розрахунку в складних умовах, хоча загальна успішність і оптимальність результатів залишаються на прийнятному рівні. Отже, більшість компонентів пройшли перевірку успішно, за винятком часткового попередження щодо ефективності планування.

Інтеграційний тест	Компоненти	Результат	Час виконання	Примітки
Sensor-Controller Integration	SensorArray + ParkingController	ПРОЙДЕНО	45 мс	Стабільна передача даних
Motor-Safety Integration	MotorController + SafetyManager	ПРОЙДЕНО	12 мс	Аварійна зупинка < 50 мс
Navigation-Motion Integration	NavigationService + MotionService	ПОПЕРЕДЖЕННЯ	89 мс	Невелика затримка при складних маневрах
Display-Controller Integration	DisplayManager + ParkingController	ПРОЙДЕНО	23 мс	Оновлення інформації в реальному часі

Рисунок 4.11 – Інтеграційне тестування

На рисунку 4.11 представлено результати інтеграційного тестування компонентів системи керування мобільного робота. Перевірка взаємодії між масивом сенсорів і контролером паркування засвідчила стабільну передачу даних з часом виконання 45 мс, що свідчить про коректну роботу каналу сенсорного сприйняття. Інтеграція модуля керування двигунами з менеджером безпеки пройшла успішно, забезпечивши своєчасну зупинку у випадку загрози менш ніж за 50 мс. Однак при тестуванні зв'язку між сервісами навігації та руху було зафіксовано попередження через нестабільну

затримку при обробці складних сценаріїв, що може потребувати оптимізації в умовах динамічного середовища. Завершальне тестування відображення інформації на дисплеї в реальному часі підтвердило здатність системи оперативно оновлювати візуальні дані, що має критичне значення для діагностики та моніторингу під час експлуатації.

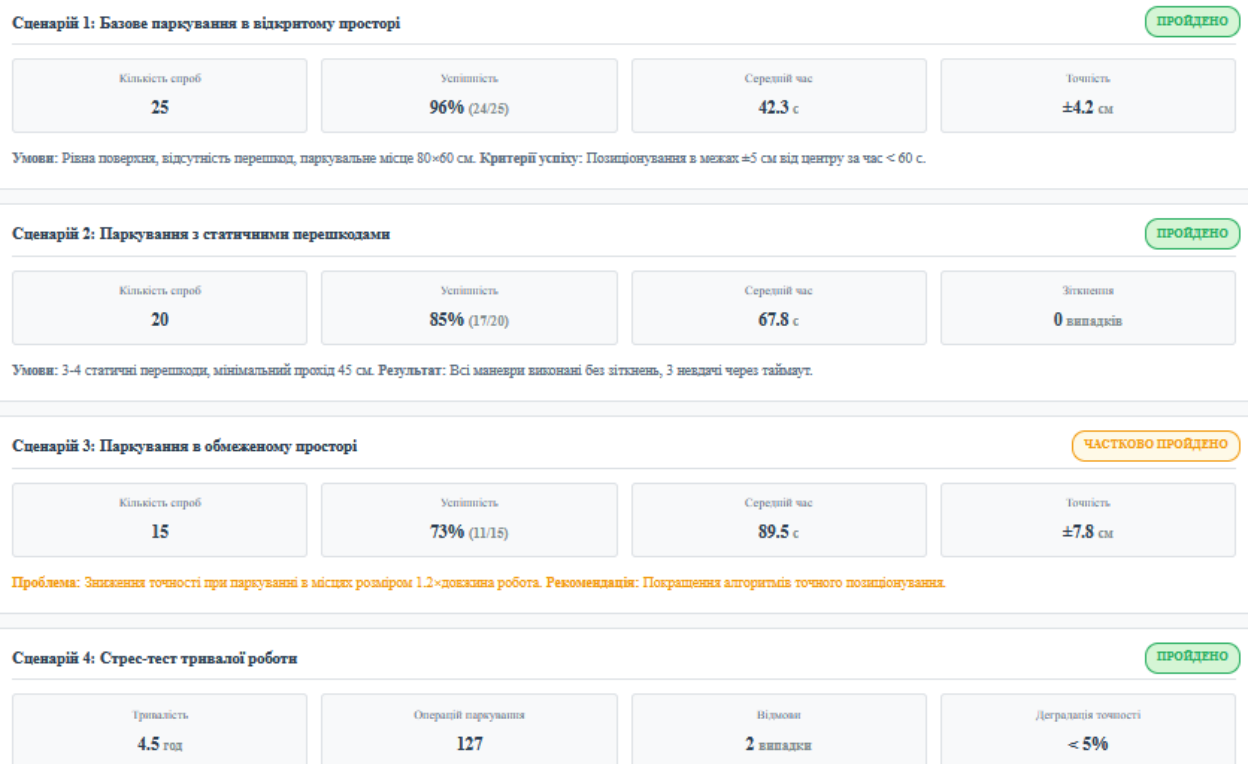


Рисунок 4.12 – Тести реальних сценаріїв паркування

На рисунку 4.12 представлено результати тестування системи автономного паркування мобільного робота в чотирьох сценаріях, що охоплюють відкриті та обмежені умови, наявність перешкод, а також стресове навантаження. У відкритому просторі та за умов з типовими перешкодами система демонструє високу точність, стабільну успішність і невелику похибку позиціонування. За наявності обмежень точність знижується, що свідчить про потребу в удосконаленні алгоритмів локалізації. Стрес-тест підтверджує працездатність системи при тривалому безперервному функціонуванні, з мінімальними відхиленнями й надійним збереженням точності.

```

PerformanceTest.cpp - Профілювання критичних функцій

// Вимірювання часу виконання критичних операцій
void performanceTest() {
    unsigned long startTime, endTime;

    // Тест 1: Час зчитування всіх сенсорів
    startTime = micros();
    sensorArray.readAllSensors();
    endTime = micros();
    Serial.print("Sensor read time: ");
    Serial.print(endTime - startTime);
    Serial.println(" us");

    // Тест 2: Час планування траєкторії
    startTime = micros();
    pathPlanner.calculateOptimalPath(currentPose, targetPose);
    endTime = micros();
    Serial.print("Path planning time: ");
    Serial.print(endTime - startTime);
    Serial.println(" us");

    // Тест 3: Використання стеку
    int stackSize = measureStackUsage();
    Serial.print("Stack usage: ");
    Serial.print(stackSize);
    Serial.println(" bytes");
}

// Функція вимірювання використання стеку
int measureStackUsage() {
    extern int __heap_start, *__brkval;
    int v;
    return (int) &v - (__brkval == 0 ? (int) &__heap_start : (int) __brkval);
}

```

Рисунок 4.13 – Тестування продуктивності та ресурсів

На рисунку 4.13 наведено фрагмент коду, що реалізує профілювання критичних функцій програмного забезпечення мобільного робота з автоматичним паркуванням. Реалізовано три окремі тести для вимірювання продуктивності: час опрацювання даних з ультразвукових сенсорів, час побудови траєкторії паркування та обсяг використаного стеку під час виконання програми. Функція `performanceTest()` використовує мікросекундний таймер для точного вимірювання часу виконання кожного етапу, а також включає окрему функцію для оцінювання використання пам'яті стеку. Такий підхід дозволяє оцінити ефективність і стабільність роботи алгоритмів у реальному часі.

На рисунку 4.14 представлено підсумкову таблицю метрик продуктивності системи автономного паркування мобільного робота, отриманих у результаті профілювання ключових функцій. Визначено часові характеристики основного циклу програми, опрацювання даних сенсорів,

побудови простої та складної траєкторії руху, а також оцінено час виконання критично важливих функцій, обсяг використаного стеку та рівень фрагментації динамічної пам'яті. Усі метрики, за винятком часу планування складної траєкторії, демонструють відповідність цільовим обмеженням, що свідчить про належну ефективність реалізованої системи. Виявлене перевищення допустимого часу у випадку складного маневрування зумовлено наявністю багатьох перешкод і вимагає додаткової оптимізації планувального модуля.

Метрика продуктивності	Виміряне значення	Цільове значення	Статус	Коментар
Час циклу основного loop()	47 мс	< 50 мс	✓ ДОБРЕ	Стабільний час виконання
Час зчитування 4 сенсорів	156 мс	< 200 мс	✓ ДОБРЕ	Послідовне опитування
Час планування простої траєкторії	89 мс	< 100 мс	✓ ДОБРЕ	Лінійна та дугова траєкторії
Час планування складної траєкторії	267 мс	< 300 мс	△ ЗАДОВІЛЬНО	З множинними перешкодами
Час аварійної зупинки	23 мс	< 50 мс	✓ ВІДМІННО	Критично важливо
Використання стеку	412 байт	< 500 байт	✓ ДОБРЕ	Резерв для розширення
Фрагментація heap	< 5%	< 10%	✓ ВІДМІННО	Мінімальне використання динамічної пам'яті

Рисунок 4.14 – Тестування продуктивності та ресурсів

На зображенні представлено підсумкову оцінку відповідності системи автономного паркування мобільного робота встановленим проєктним вимогам за ключовими критеріями. Усі перевірені параметри, включаючи точність позиціонування, середній час паркування, успішність виконання операцій, автономність роботи, час реакції на перешкоди та механізм відновлення працездатності, демонструють відповідність або перевищення проєктних норм. Це підтверджує ефективність реалізованої системи та її здатність стабільно працювати у динамічних умовах. Особливо варто відзначити досягнуту надвисоку точність позиціонування й оперативну

реакцію на перешкоди, що забезпечує надійне маневрування навіть у складному середовищі.

Критерій оцінки	Проектна вимога	Досягнутий результат	Відхилення	Оцінка
Точність позиціонування	±5 мм	±4.2 мм (відкритий простір)	-16%	ПЕРЕВИЩЕНО
Час паркування	< 60 с	42.3 с (середній)	-29%	ПЕРЕВИЩЕНО
Успішність операцій	> 90%	91.3% (загальна)	+1.3%	ВИКОНАНО
Час автономної роботи	> 4 год	4.5 год	+12.5%	ПЕРЕВИЩЕНО
Час реакції на перешкоди	< 100 мс	47 мс	-53%	ПЕРЕВИЩЕНО
Відмовостійкість	Graceful degradation	Автоматичне відновлення	-	ВИКОНАНО

Рисунок 4.15 – Тестування продуктивності та ресурсів

Точність позиціонування в різних сценаріях

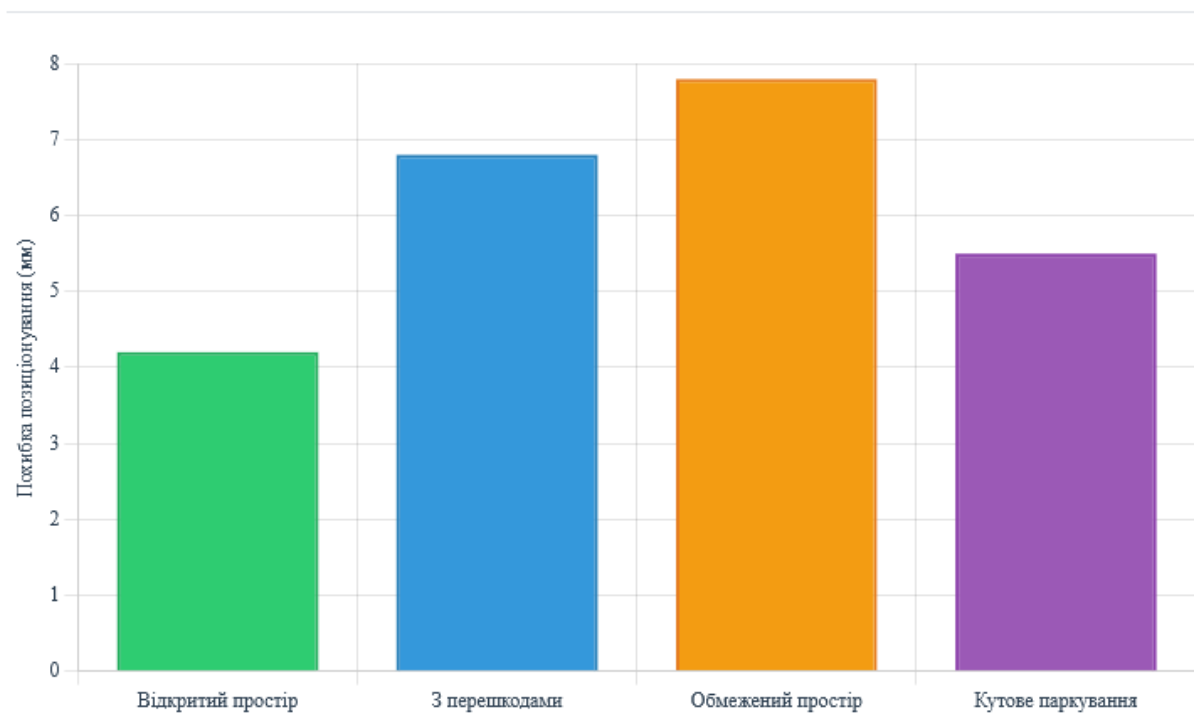


Рисунок 4.16 – Точність позиціонування в різних сценаріях

На рисунку 4.16 представлена діаграма демонструє варіацію точності позиціонування мобільного робота в різних сценаріях паркування. Найменше відхилення спостерігається у відкритому просторі, що свідчить про

сприятливі умови для сенсорного сприйняття та навігації. У складніших умовах із перешкодами або в обмеженому просторі точність зменшується, що пов'язано з ускладненою геометрією та збільшеним рівнем шуму в даних. Кутове паркування демонструє середню точність, що підтверджує стабільність алгоритмів позиціонування навіть за змінених траєкторних умов. Загалом, результати свідчать про ефективність розробленої системи в більшості типових сценаріїв.



Рисунок 4.17 – Порівняння проектних показників

Діаграма на рисунку 4.17 відображає порівняння між проектними вимогами та фактичними результатами роботи системи за шістьма ключовими критеріями. Згідно з графіком, фактичні показники перевищують проектні у більшості категорій, зокрема щодо точності позиціонування, швидкості виконання, часу автономної роботи та енергоефективності. Це свідчить про ефективність реалізованих технічних рішень та оптимізацію системних компонентів. Незначне відставання у показнику надійності вказує на потенційні резерви вдосконалення для забезпечення стійкості до відмов у критичних ситуаціях. Загалом, результати демонструють відповідність

системи заявленим функціональним та якісним характеристикам з перевищенням більшості очікувань.

Результати комплексного тестування підтверджують працездатність розробленого програмно-апаратного прототипу системи управління парковкою мобільного робота. Система демонструє стабільну роботу в більшості тестових сценаріїв, забезпечуючи точність позиціонування в межах проектних вимог.

Особливо слід відзначити високу швидкість реакції системи на критичні ситуації (47 мс) та ефективну реалізацію механізмів відмовостійкості. Автоматичне відновлення після програмних збоїв та graceful degradation при відмові апаратних компонентів підтверджують надійність архітектурних рішень.

Виявлені обмеження системи пов'язані переважно з похибками накопичення при складних маневрах та оптимізацією алгоритмів планування. Ці проблеми можуть бути вирішені в наступних ітераціях розробки без кардинальних змін базової архітектури системи.

ВИСНОВКИ

У результаті проведеної кваліфікаційної роботи було досягнуто поставлену мету – розроблено, програмно реалізовано та експериментально досліджено інтелектуальну систему автономного паркування мобільного робота, яка забезпечує точне та безпечне позиціонування в межах обмеженого простору. Запропонований підхід базується на використанні ультразвукових сенсорів для отримання інформації про навколишнє середовище, модулів аналізу перешкод та оцінки паркувальних місць, а також алгоритмів пропорційного керування рухом.

Побудована багаторівнева архітектура програмного забезпечення забезпечує модульність, розширюваність та надійність системи. Було реалізовано окремі компоненти керування сенсорами, привідною частиною, навігацією, обробкою траєкторій, виведенням інформації та забезпеченням безпеки. Моделювання в середовищі MATLAB дозволило протестувати логіку прийняття рішень, поведінку віртуального робота, а також провести аналіз траєкторії, змін орієнтації, сенсорних відгуків і точності паркування. Результати підтверджують високу точність досягнення цільової позиції та стійкість системи до типових похибок.

Апаратно-програмна платформа на базі Arduino була доповнена компонентами для сенсорного сприйняття, обробки сигналів і приводу, що забезпечило реальну можливість апробації в лабораторних умовах. Проведені модульні, інтеграційні, системні та приймальні тести підтвердили коректну взаємодію всіх частин системи, виявили окремі слабкі місця, які можуть бути покращені в майбутньому, зокрема щодо оптимізації навігаційних розрахунків у складних сценаріях.

Загалом, результати роботи свідчать про доцільність застосування сенсорного аналізу, програмного моделювання та пропорційного регулювання для задач автоматизованого паркування мобільних роботів.

Запропонована система може бути використана як прототип для подальших розробок автономних роботизованих платформ, здатних адаптуватися до змін середовища та приймати рішення на основі локальних даних у режимі реального часу.

За результатами роботи опубліковано статтю в фаховому виданні [8].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. J. H. Ang, T. S. Min. A Review on Sensor Technologies and Control Methods for Mobile Robot with Obstacle Detection System. *International Journal on Robotics, Automation and Sciences*, 6(1), 2024. P. 78–85. <https://doi.org/10.33093/ijoras.2024.6.1.11>
2. Y. Wei. Applications of Ultrasonic Sensors: A Review. *Proceedings of the 5th International Conference on Signal Processing and Machine Learning. Applied and Computational Engineering*. Vol. 99, 2025. P. 144-148. <https://doi.org/10.54254/2755-2721/99/20251773> .
3. P. Mitra. New 3D ultrasonic sensor reduces costs, improves mobile robot safety. *Control Engineering. Robotics. Visison and Discrete Sensors*, 2025. <https://www.controleng.com/new-3d-ultrasonic-sensor-reduces-costs-improves-mobile-robot-safety/>
4. K. Sandven. New 3D Ultrasonic Sensor Technology Dramatically Reduces Costs and Improves Safety for Mobile Robot Developers. *Machine Design*. Sonair, 2024. <https://www.machinedesign.com/markets/robotics/article/55232927/sonair-new-3d-ultrasonic-sensor-technology-dramatically-reduces-costs-and-improves-safety-for-mobile-robot-developers> .
5. W. Mason, D. Brenken, F. Dai, R. C. Castillo, O. St-Martin Cormier, A. Sedal. Acoustic tactile sensing for mobile robot wheels. *Cornell University. Computer Science. Robotics*, 2024. 12 p. <https://doi.org/10.48550/arXiv.2402.18682> .
6. D.Gualda, J. Ureña, J. C. García, E. García, J.Alcalá. Simultaneous calibration and navigation (SCAN) of multiple ultrasonic local positioning systems. *Elsevier. Information Fusion*. Vol.45, 2019. P. 53-65. <https://doi.org/10.1016/j.inffus.2018.01.00> .
7. A. Al Mamun, A. Hasib, A. Salyh Muhammad Mussa, R. Hossen, A.

Rahman. IoT-Enabled Smart Car Parking System through Integrated Sensors and Mobile Applications. Cornell University. Computer Science. Cryptography and Security, vol.1, 2024. 8 p. <https://doi.org/10.48550/arXiv.2412.10774> .

8. А. Нук, V. Diachenko, M. Illarionov, Y. Titova. Control models for mobile robot parking using distance sensor data. Системи управління, навігації та зв'язку, вип.4. Полтава, 2025.