

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)

Система моніторингу характеристик руху мобільного
колісного робота на базі датчику одометру

(тема)

Виконав:

студент II курсу, групи СПМ-21-2
Жуков С.В.
(прізвище, ініціали)

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: проф. Каргін А.О.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри ЕОМ

(підпис)

Коваленко А.А.

(прізвище, ініціали)

2023 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

студенту _____ Жукову Станіславу Валерійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи Система моніторингу характеристик руху мобільного колісного робота на базі датчику одометру

затверджена наказом по університету від “ 03 ” квітня 2023 р. № 318 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 17 травня 2023 р

3. Вхідні дані до роботи _____

Моделі програмного управління роботами

Методи обробки даних від сенсорів в робототехніці

4. Перелік питань, що потрібно опрацювати у роботі _____

Огляд існуючих аналогів мобільних роботів

Логічна модель системи

Концептуальна модель системи

Фізична модель системи

Реалізація системи

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

Презентація доповіді - 11 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	Термін виконання етапів роботи	Примітка
1	Аналіз предметної області	03.04.23 – 07.04.23	
2	Розробка методів	07.04.23 – 14.04.23	
3	Проектування системи	14.04.23 – 23.04.23	
4	Моделювання та реалізація	23.04.23 – 01.05.23	
5	Оформлення матеріалів кваліфікаційної роботи	01.05.23 – 07.05.23	
6	Підготовка та редагування звіту	07.05.23 – 10.05.23	
7	Подання кваліфікаційної роботи керівникові	10.05.23 – 14.05.23	
8	Подання кваліфікаційної роботи на рецензування	14.05.23 – 17.05.23	

Дата видачі завдання 03 квітня 2023 р

Студент _____
(підпис)

Керівник роботи _____
(підпис)

проф. Каргін А.О.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 73 с., 16 рис., 1 табл., 2 дод., 21 джерел.

СИСТЕМА МОНІТОРИНГУ, МОБІЛЬНИЙ КОЛІСНИЙ РОБОТ, ДАТЧИК ОДОМЕТРУ, ХАРАКТЕРИСТИКИ РУХУ, ARM, МАТЛАВ, МКР, ПРОГРАМНЕ УПРАВЛІННЯ, КОНТРОЛЕР РУХУ, ОПТИМІЗАЦІЯ РУХУ.

Метою кваліфікаційної роботи є створення системи моніторингу характеристик руху мобільного колісного роботу на базі датчику одометру.

У ході виконання кваліфікаційної роботи було проведено аналіз існуючих систем та мобільних колісних роботів. Здійснено порівняння існуючих аналогів мобільних колісних роботів. Опис аналогів датчика одометра. Проведено аналіз існуючих датчиків одометра. Виконання системи моніторингу характеристик руху мобільного колісного робота на базі датчику одометру.

ABSTRACT

Master's thesis: 73 pages, 16 figures, 1 table, 2 appendices, 21 sources.

MONITORING SYSTEM, MOBILE WHEEL ROBOT, ODOMETER SENSOR, MOTION CHARACTERISTICS, ARM, MATLAB, MCR, SOFTWARE CONTROL, MOTION CONTROLLER, MOTION OPTIMIZATION.

The major goal of this thesis is to create a system for monitoring the characteristics of the movement of a mobile wheeled robot based on the odometer sensor.

In the course of the qualification work, an analysis of existing systems and mobile wheeled robots was carried out. A comparison of existing analogs of mobile wheeled robots was made. Description of analogues of the odometer sensor. The existing odometer sensors were analyzed. Implementation of a system for monitoring the movement characteristics of a mobile wheeled robot based on an odometer sensor.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП	8
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ПІДХОДІВ	9
1.1 Загальний опис предметної області	9
1.2 Огляд існуючих аналогів мобільних роботів	11
1.3 Датчик одометра.....	14
1.4 Постанова задачі.....	20
2 АНАЛІЗ ТЕХНОЛОГІЙ ДЛЯ СИСТЕМИ МОНІТОРИНГУ ХАРАКТЕРИСТИК РУХУ МОБІЛЬНОГО КОЛІСНОГО РОБОТУ НА БАЗІ ДАТЧИКУ ОДОМЕТРУ	21
2.1 Логічна модель	21
2.2 Концептуальна модель	28
2.3 Фізична модель системи.....	33
2.4 Моделювання в MatLab методів керування переміщенням роботу	38
3 РЕАЛІЗАЦІЯ СИСТЕМИ.....	46
3.1 Реалізація системи.....	46
3.2 Опис використаних компонентів.....	49
3.3 Опис компонентів реалізації.....	56
ВИСНОВКИ.....	59
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	60
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	62
ДОДАТОК Б	69
КОД СИСТЕМИ	69
ДОДАТОК В.....	72
РЕАЛІЗАЦІЯ ДОДАТКОВИХ ДАТЧИКІВ	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

ШІ – Штучний інтелект

МБ – Мобільний робот

СМР – Система моніторингу руху

ПУ – Програмне управління

ШО – Штучне оточення

ШІВ – Штучний інтелект що відчуває

ВСТУП

Система моніторингу характеристик руху мобільного колісного робота на базі датчика одометра є важливою компонентою у багатьох робототехнічних додатках, де потрібне точне визначення позиції та швидкості робота. Датчик одометра призначений вимірювання переміщення робота шляхом виявлення обертання коліс.

Система моніторингу може складатися з декількох компонентів, таких як мікроконтролер, датчик одометра, програмне забезпечення обробки даних та інтерфейс для взаємодії з оператором або іншими системами.

При роботі системи датчик одометра збирає дані про обертання коліс та передає їх на мікроконтролер, де ці дані обробляються для визначення поточної позиції та швидкості робота. Програмне забезпечення може використовуватися для запису та аналізу цих даних, а також для визначення додаткових характеристик руху, таких як прискорення та кут повороту [1].

Однією з основних переваг системи моніторингу характеристик руху на базі датчика одометра є її висока точність та надійність.[3] Крім того, вона може бути використана для підвищення безпеки та ефективності робототехнічних додатків, наприклад, для автономного руху робота або визначення оптимального маршруту руху.

Два найпростіші варіанти використання системи моніторингу характеристик руху на базі датчика одометра можуть бути: використання її для визначення позиції робота на карті (наприклад, у приміщенні), та використання її для вимірювання швидкості руху робота.

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА ОГЛЯД ПІДХОДІВ

1.1 Загальний опис предметної області

Аналіз галузі застосування роботів демонструє широкий спектр використання роботів у різних сферах життя. В сучасному світі роботи використовуються в індустрії, медицині, науці, архітектурі, автомобільній та авіаційній промисловості та інших галузях.

В автомобільній промисловості роботи широко використовуються в процесі виготовлення автомобілів, зокрема в збірці елементів кузова і двигуна, а також в роботі на лініях ремонту і обслуговування автомобілів. Роботи також застосовуються для автоматизації складських процесів та транспортування товарів.

У медицині роботи використовуються для проведення операцій, реабілітації пацієнтів, дослідження і тестування ліків. В науці роботи застосовуються для дослідження космосу, океану, геологічних формацій і т.д.

У сфері обслуговування роботи використовуються в готелях, ресторанах, банках і магазинах для обслуговування клієнтів, очищення приміщень і навколишнього середовища, а також для охорони і безпеки.

У загальному, роботи відіграють важливу роль в автоматизації процесів та полегшенні праці людей в різних сферах діяльності. Розвиток технологій дозволяє розширювати можливості застосування роботів та вдосконалювати їхню роботу, забезпечуючи більш точний та ефективний процес виробництва та обслуговування[13]. Крім того, роботизація виробництва є однією з найважливіших тенденцій у промисловості. Завдяки використанню роботів, виробничі процеси можуть бути автоматизовані, що дозволяє підвищувати ефективність та якість продукції, скорочувати терміни виготовлення та знижувати витрати. В цьому контексті мобільні колісні роботи, які можуть пересуватися по виробничих приміщеннях, є дуже корисними.

Система моніторингу характеристик руху мобільного колісного роботу на базі датчика одометру може бути застосована в різних галузях, включаючи промисловість, транспорт, медицину та науку. У промисловості вона може бути використана для контролю за рухом роботів на виробничій лінії, забезпечення точності та якості виготовленої продукції. У транспорті така система може допомогти водієві автомобіля контролювати швидкість руху та пройденої відстань, що знижує ризик аварії та покращує безпеку на дорозі.

У медицині система моніторингу може бути використана для контролю за рухом медичних роботів, що забезпечує точність та ефективність операцій. У науці така система може бути використана для дослідження руху та поведінки роботів у різних умовах, що допомагає удосконалювати їхні характеристики та функціональні можливості.

Отже, система моніторингу характеристик руху мобільного колісного роботу на базі датчика одометру має широкий спектр застосування та є дуже актуальною в сучасному світі, де індустрія робототехніки і автономних систем стає все більш розвиненою. Ця тема має великий потенціал у багатьох галузях, де використовуються мобільні роботи.

1) Промисловість: у великих промислових комплексах та заводах мобільні роботи використовуються для транспортування матеріалів, відходів, обладнання та іншого вантажу.[1] Система моніторингу характеристик руху може допомогти у покращенні роботи таких роботів, збільшенні їхньої продуктивності та надійності.

2) Сільське господарство: мобільні роботи використовуються в сільському господарстві для автоматизації процесів поливу, внесення добрив, збирання врожаю та інших робіт. Система моніторингу характеристик руху може допомогти в покращенні точності та ефективності цих процесів.

3) Медицина: мобільні роботи можуть використовуватися в медицині для транспортування зразків, ліків та іншого медичного обладнання. Система моніторингу характеристик руху може допомогти у

забезпеченні точності та швидкості доставки, що особливо важливо у випадках надзвичайних ситуацій.

1.2 Огляд існуючих аналогів мобільних роботів

Сьогодні мобільні колісні роботи знаходять широке застосування в різних галузях, таких як промисловість, медицина, автомобільна та військова промисловість, логістика та інші. У зв'язку з цим, на ринку існує велика кількість аналогів роботів з різними характеристиками та можливостями.

Одним з найбільш відомих і успішних колісних роботів є робот-пилосос Roomba виробництва компанії iRobot. Він має високу мобільність та можливість автономного навігації у просторі, що дозволяє йому з легкістю очищати підлогу в приміщеннях.[9] Робот оснащений різноманітними датчиками, такими як датчик оптичної навігації, датчик відстані, датчик акустичної навігації та інші.

Також існують колісні роботи для військових цілей. Один з них - це робот SWORDS (Special Weapons Observation Reconnaissance Detection System), розроблений компанією Foster-Miller. Цей робот має можливість самостійно рухатися на важкому бездоріжжі, пересуватися під водою та навіть лазити по сходах. Робот SWORDS (Special Weapons Observation Reconnaissance Detection System) є мобільним колісним роботом, який був розроблений для використання в армії США. Він призначений для проведення розвідки, збору інформації та моніторингу ситуації на полі бою.

SWORDS складається з двох частин: основного робота та операторської консолі. Оператор може керувати роботом з безпечної відстані за допомогою джойстика та інших управляючих пристроїв, а також отримувати відео-інформацію та дані з різних датчиків.

SWORDS обладнаний різноманітними сенсорами, включаючи камери та мікрофони для збору звукової та візуальної інформації, а також датчиками нахилу, щоб виявляти поштовхи та інші рухи. Робот також може бути

оснащений різними видами зброї та іншої бойової техніки, що дозволяє йому вести вогонь на ворога, який піддається обстрілу з безпечної відстані.



Рисунок 1.1 – Колісний робот SWORDS

SWORDS може використовуватися в різних сценаріях, включаючи розвідку під час бойових дій, пошук та рятування, демонтаж вибухових пристроїв, моніторинг небезпечних зон та інше. Цей робот допомагає збільшити безпеку солдатів та дозволяє отримувати важливу інформацію без необхідності відправляти людей у небезпечні зони.[18]

Також, робот SWORDS знайшов застосування в мисливстві та вибуховій справі. Він може бути використаний для збору інформації про

диких тварин, таких як кабани або борсуки, що допомагає зберегти життя мисливців, а також може бути корисним у вибуховій справі, де він може бути використаний для оцінки ризику для людей і матеріальних цінностей.

Ще одним відомим прикладом використання роботів є Boston Dynamics Spot, який був використаний в Нью-Йорку в ході пандемії COVID-19. Spot був використаний для зняття температури та моніторингу симптомів COVID-19 у пацієнтів, що дозволило медичному персоналу зберігати безпеку від зараження вірусом.

У сучасному світі, роботи знаходять все більше застосувань у різних галузях, таких як медицина, промисловість, логістика та військова справа. Розробка систем моніторингу характеристик руху мобільного колісного роботу на базі датчика одометру може значно покращити продуктивність та безпеку роботів у цих галузях, а також в інших, де вони знайшли своє застосування.

Окрім цього, існують роботи, призначені для виконання конкретних завдань в різних сферах, таких як медицина, промисловість, транспорт, логістика та інші. Наприклад, у медицині роботи використовуються для проведення складних операцій, у промисловості - для збирання, упаковки та сортування товарів, а в логістиці - для доставки товарів.



Рисунок 1.2 – Мобільний колісний робот

Одним з найбільш популярних роботів в світі є мобільний колісний

робот, який може переміщатися в будь-якому середовищі. Такі роботи використовуються в багатьох галузях, наприклад, у військовій справі, логістиці, медицині та багатьох інших.

Однак, деякі проблеми щодо моніторингу характеристик руху мобільних колісних роботів залишаються невирішеними. До цих проблем можна віднести питання точності вимірювання параметрів руху та можливості збору даних в режимі реального часу[20].

Тому, система моніторингу характеристик руху мобільного колісного робота на базі датчика одометру є важливим напрямком досліджень у галузі робототехніки та автоматизації процесів. Вона дозволяє вимірювати параметри руху робота з високою точністю та збирати дані в режимі реального часу. Це дозволяє ефективніше контролювати рух робота та забезпечувати його безпеку під час виконання завдань в різних галузях.

1.3 Датчик одометра

Для вимірювання відстані і швидкості руху колісного робота можна використовувати різні датчики, зокрема:

Одометр - датчик, який вимірює відстань, пройдену роботом за допомогою вимірювання кількості обертів коліс і діаметру колеса. Одометр є найбільш поширеним датчиком для вимірювання відстані і швидкості руху в колісних роботах. [14]

Лазерний дальномір - датчик, який використовує лазерний промінь для вимірювання відстані до об'єкта. Лазерний дальномір може бути використаний для вимірювання відстані до стін, перешкод або інших об'єктів.

Ультразвуковий дальномір - датчик, який використовує ультразвукові хвилі для вимірювання відстані до об'єкта. Ультразвуковий дальномір може бути використаний для вимірювання відстані до перешкод, які можуть бути недоступні для лазерного дальноміра.

GPS - система глобального позиціонування, яка використовує сигнали з супутників для визначення місця розташування робота і його швидкості руху.

Інфрачервоний датчик - датчик, який вимірює відстань до об'єкта за допомогою інфрачервоних хвиль. Інфрачервоні датчики можуть бути використані для вимірювання відстані до об'єктів в помірних умовах освітлення.

Крім використання датчиків відстані, таких як лазерні дальноміри, є інші методи вимірювання відстані, що можуть бути використані для моніторингу руху мобільного колісного робота.

Один з таких методів - використання ультразвукових датчиків. Вони працюють на основі відбивання звукових хвиль від поверхні об'єкта, і дозволяють вимірювати відстань до цього об'єкта. Ультразвукові датчики часто використовуються в робототехніці для визначення відстані до стін, перешкод або інших об'єктів.[11]

Інший метод - використання інфрачервоних датчиків. Ці датчики також вимірюють відстань на основі відбивання, але від інфрачервоних променів. Вони часто використовуються в робототехніці для визначення відстані до стін, перешкод або інших об'єктів, а також для визначення положення робота в просторі. [15]

Також існують системи вимірювання відстані на основі GPS (глобальної системи позиціонування). Вони дозволяють визначати координати робота в реальному часі, що дає можливість моніторити його рух на відстані до кількох метрів.

Кожен з цих методів має свої переваги та недоліки, і вибір методу вимірювання відстані залежить від конкретної задачі та умов, в яких працює робот. У деяких випадках може бути вигідно використовувати не тільки один датчик, а кілька, щоб отримати більш точні дані про рух робота. Наприклад, використання інфрачервоного датчика додатково до датчика одометру може допомогти визначити точне положення робота, коли він знаходиться в

темному середовищі, де датчик одометру може не працювати належним чином.

Також, для вимірювання відстані можуть бути використані датчики відстані на основі ультразвуку, радарні датчики, лазерні датчики тощо. Кожен з цих датчиків має свої переваги та недоліки і може бути більш ефективним у певних умовах.

Наприклад, ультразвуковий датчик може бути корисним для вимірювання відстані до об'єктів, що перебувають на невеликій відстані від робота, а радарний датчик може бути більш ефективним для вимірювання відстані до об'єктів на великій відстані. Лазерний датчик може бути корисним для точного вимірювання відстані до об'єктів з високою точністю та швидкістю вимірювання.

Важливо враховувати, що використання декількох датчиків може призвести до більш складної обробки даних та збільшення складності програмного забезпечення робота. Тому перед вибором датчиків для вимірювання відстані необхідно зробити аналіз вимог до системи та обрати найбільш оптимальні датчики для певних умов та завдань, що має виконувати робот. [14]

Схема одометра - це електрична схема, яка використовується для вимірювання відстані, яку пройшов рухомий об'єкт. Одометр може бути механічним, електронним або гібридним (Електромеханічний). У випадку з механічним одометром, рухомий об'єкт приводить в рух механізм, що виконує роботу зі збереження вимірювання відстані. В електронному випадку, датчик вимірює відстань, а потім електронна схема перетворює ці дані в вимірювання відстані. У гібридному одометрі комбінуються обидві схеми.[9]

У системах моніторингу характеристик руху мобільного колісного робота на базі датчика одометру, схема одометра складається з двох головних компонентів: датчика руху та електронної схеми. Датчик руху вимірює кількість обертів коліс робота, а електронна схема перетворює ці

дані в відстань, яку рухомий об'єкт проходить. Такі системи дозволяють точно відстежувати рух робота та контролювати його швидкість і відстань, пройдену за певний час.

Після отримання сигналу від датчика, він передається до мікроконтролера, який обробляє ці дані. Мікроконтролер обчислює відстань, пройдену роботом, на основі даних, отриманих від датчика одометра, і зберігає цю інформацію у внутрішній пам'яті.

Для використання отриманої інформації про пройдену відстань, мікроконтролер може бути підключений до інших систем робота, наприклад, системи автоматичного управління, що дозволяє забезпечити точне рух робота на задану відстань.

Після цього сигнал з датчика подається на схему обчислення одометра, яка може бути реалізована як електронна схема або програмно [14] на мікроконтролері. Схема обчислення використовує вхідний сигнал від датчика і виконує алгоритм обчислення пройденної відстані.

Одним з найпростіших методів обчислення пройденної відстані є метод різниці кількості імпульсів між двома моментами часу. При цьому вимірюється час між двома пульсами і знаходиться середнє значення часу для одного оберту колеса. Потім використовуючи відомий діаметр колеса, розраховується відстань, яку пройшов робот.

Проте цей метод має деякі недоліки, зокрема, він не враховує зносу коліс і деформації шин, що може призвести до неточностей в вимірюваннях. Для усунення цих недоліків можна використовувати більш складні алгоритми, які враховують фактичний діаметр колеса залежно від його стану і стиснення шин.[17]

Однак, важливо зазначити, що точність вимірювання відстані за допомогою одометру може бути порушена через ряд факторів. Наприклад, зношеність шин або нерівності на дорозі можуть спричинити зміну діаметра колеса, що змінить коефіцієнт вимірювання відстані. Також, відмінності в поверхні дороги, нахилу та інші фактори можуть вплинути на точність

вимірювання відстані, отриманої за допомогою одометру.

Одним з рішень цих проблем є використання калібровки датчика одометру, що дозволяє визначити точний коефіцієнт вимірювання відстані в залежності від впливу різних факторів. Також, можуть бути використані додаткові датчики, такі як датчики інклінометра, щоб врахувати нахил дороги та інші фактори впливу на точність вимірювання відстані.

В даний час на ринку доступні різноманітні датчики одометра з різними характеристиками та варіантами підключення.[14] Деякі з них можуть бути підключені безпосередньо до мікроконтролера робота, інші - через додатковий інтерфейс, такий як CAN або RS-485.

Одним з рішень для вирішення проблем точності вимірювання відстані датчиком одометра є калібрування. Калібрування датчика одометра дозволяє встановити точний коефіцієнт вимірювання відстані, що залежить від впливу різних факторів, таких як розмір колеса, стан поверхні, зношеність шин і т.д. Це дозволяє забезпечити більш точні результати вимірювання. Крім того, для поліпшення точності вимірювання можуть використовуватися додаткові датчики, такі як датчики інклінометра. Такий підхід дозволяє покращити точність вимірювання, особливо на нерівних або нахилених поверхнях.

На ринку доступні різноманітні датчики одометра з різними характеристиками та варіантами підключення. [15] Деякі датчики можуть бути підключені безпосередньо до мікроконтролера робота, що спрощує процес інтеграції. Інші датчики можуть використовувати додаткові інтерфейси, такі як CAN (Controller Area Network) або RS-485, що дозволяє передавати дані вимірювання на відстань до мікроконтролера. Вибір конкретного датчика одометра залежить від потреб і вимог конкретної системи моніторингу руху робота

Також існують датчики, які вимірюють кут обертання колеса. Ці датчики вимірюють кут обертання від осі колеса та відносять його до діаметра колеса, щоб визначити пройденої відстань. Основною частиною датчика одометру є датчик руху, що вимірює кількість

обертів колеса або шини, і мікроконтролер, що обробляє отримані дані і обчислює пройдений шлях. Ось проста схема датчика одометру:



Рисунок 1.3 – Схема обробки датчика одометру

Датчик руху може бути магнітним, оптичним або індуктивним. Зазвичай він встановлюється поруч зі зубчастим диском або магнітним ротором, який закріплюється на валу колеса або шини. Коли ротор обертається, датчик руху виконує вимірювання та відправляє сигнали мікроконтролеру.

1.4 Постановка задачі

Метою роботи є проведення аналізу предметної області, визначити основні проблеми та напрями вдосконалення існуючих систем моніторингу руху транспортних засобів. Зробити порівняльний аналіз розробленої системи з існуючими аналогами, визначити переваги та недоліки.

Основною метою дослідження є виявлення можливості використання датчика одометру для збору і аналізу даних про характеристики руху мобільного колісного роботу, такі як швидкість, відстань, час руху, пройдений шлях та кут повороту.

Для досягнення поставленої мети потрібно вирішити наступні завдання:

- дослідити можливості датчика одометру для збору даних про рух мобільного колісного роботу;
- провести огляд наукових джерел та існуючих систем моніторингу руху мобільних роботів, що використовують датчики одометру;
- визначити та описати параметри руху роботу, що будуть моніторингуватись;
- дослідити принцип роботи датчика одометру та його взаємодію з колісною системою робота.

Отже, в результаті виконання дипломної роботи очікується отримання моніторингу характеристик руху мобільного колісного роботу, яка дозволить отримувати інформацію про швидкість, пройдений шлях, прискорення та інші параметри руху з використанням датчика одометру.

2 АНАЛІЗ ТЕХНОЛОГІЙ ДЛЯ СИСТЕМИ МОНІТОРИНГУ ХАРАКТЕРИСТИК РУХУ МОБІЛЬНОГО КОЛІСНОГО РОБОТУ НА БАЗІ ДАТЧИКУ ОДОМЕТРУ

2.1 Логічна модель

У сучасному світі робототехніка відіграє значну роль в різних сферах життя, включаючи промисловість, медицину, автономну навігацію та багато інших. Розробка ефективних систем штучного інтелекту (ШІ), здатних самостійно приймати рішення та взаємодіяти з навколишнім середовищем, є актуальним та цікавим напрямом досліджень.

Одним з ключових аспектів успішної реалізації ШІ є архітектура системи, яка забезпечує організацію та взаємодію компонентів. Однією з поширених архітектур є шарова архітектура, що дозволяє розділити функціональні компоненти системи на рівні за їх функціональним значенням.

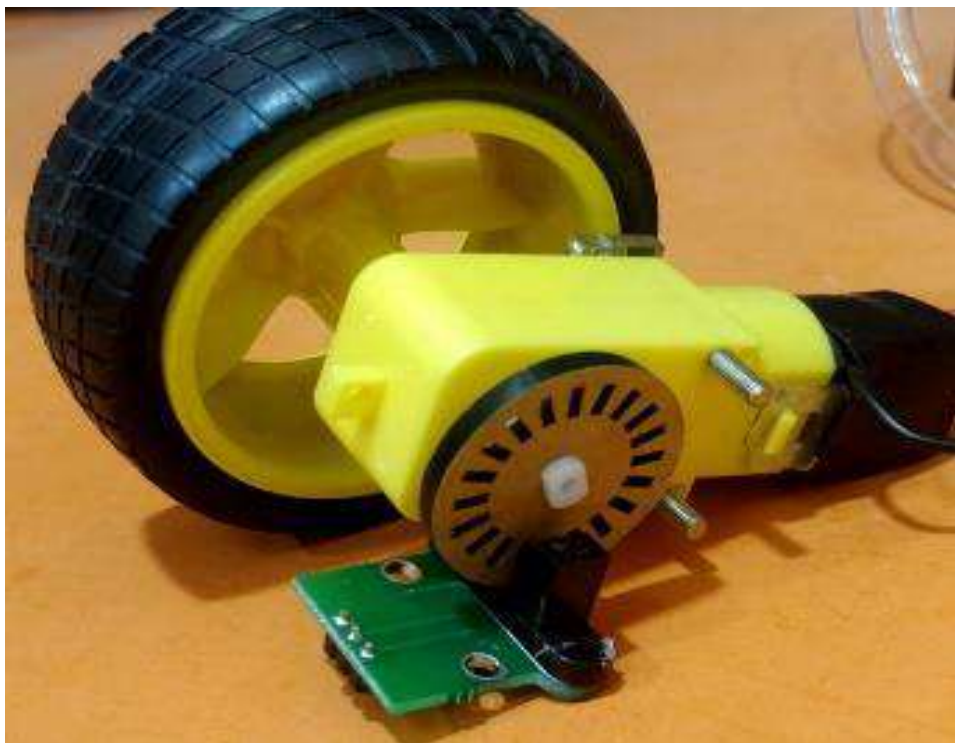
Розглядається шарова архітектура ШІ для робота, який має завдання самостійно вести себе в певному штучному оточенні. Дана архітектура складається з чотирьох шарів, кожен з яких виконує свої функції та співпрацює з іншими шарами для досягнення бажаних результатів.

ШІ має шарову архітектуру. До нижнього першого шару належать програмні інтерфейси з апаратним забезпеченням АІБС, тобто драйвери сенсорів й актуаторів. Для штучного оточення що розглядається в даній роботі це драйвери датчику одометра, ультразвукового датчику відстані та матриці датчиків відображення що влаштовані на днищі шасі робота, а також драйвери сервопривіду поворотної платформи та приводів двигунів Motor Shield. До другого шару належать контролери руху. Це контролер руху вздовж розмітки на підлозі (K1), контролер руху вздовж штучної огорожі (K2), контролер руху до візуального маркера по вільній траєкторії (K3) та контролери відстані (S) й швидкості (V) руху. До третього шару належить

контролер програмного управління рухом що реалізує план (програму в термінах управління роботами) досягнення кінцої цільової точки. І на верхньому четвертому шарі знаходяться інтелектуальні компоненти прийняття рішень в умовах неповної інформації що застосовують когнітивні моделі сприйняття даних від сенсорів та емоційної оцінки ситуації для прийняття рішень.

Перелічені вище контролери розроблені на мові C++ в середовищі Arduino IDE. Роль драйверів відіграють вбудовані функції та бібліотеки Arduino IDE. Розглянемо організацію драйверів й контролерів на прикладі одометра. При переміщеннях колесного роботу одометр збирає дані про обертання колес. Дані обробляються мікроконтролером для визначення пройденої відстані та швидкості робота.

Серед трьох типів одометрів (механічні, електромеханічні та електронні) обрано електронний що представляє собою інфрачервоний датчик H206 та диск з 20 отворами, який кріпиться на вісь двигуна



Рисинок 2.1 – Апаратна реалізація датчика одометра

Диск розміщується між інфрачервоним випромінювачем та приймачем (датчик H206) і при кожному проходженні отворів диска датчик реагує на зміну його сигналу. Оскільки отворів у диска 20, а датчик реагує на обидві зміни сигналу, то за час повного обороту колеса МР мікроконтролер нарахує 40 переривань. Відстань, пройдена роботом за один оберт колеса, дорівнюватиме довжині його кола. Для радіусу колеса r пройдена відстань обчислюється за формулою:

$$S = 2\pi r * (n/40) \quad (2.1)$$

де n – кількість переривань.

Контролер швидкості підтримує поточну лінійну швидкість руху роботу рівною заданому значенню v^* . Використовується функція `millis()`, значення якої є кількість мілісекунд починаючи з часу подачі живлення на плату Arduino. Підраховується кількість мілісекунд m на кожний повний оберт колеса й обчислюється миттєва швидкість v в одиницях м/сек по формулі:

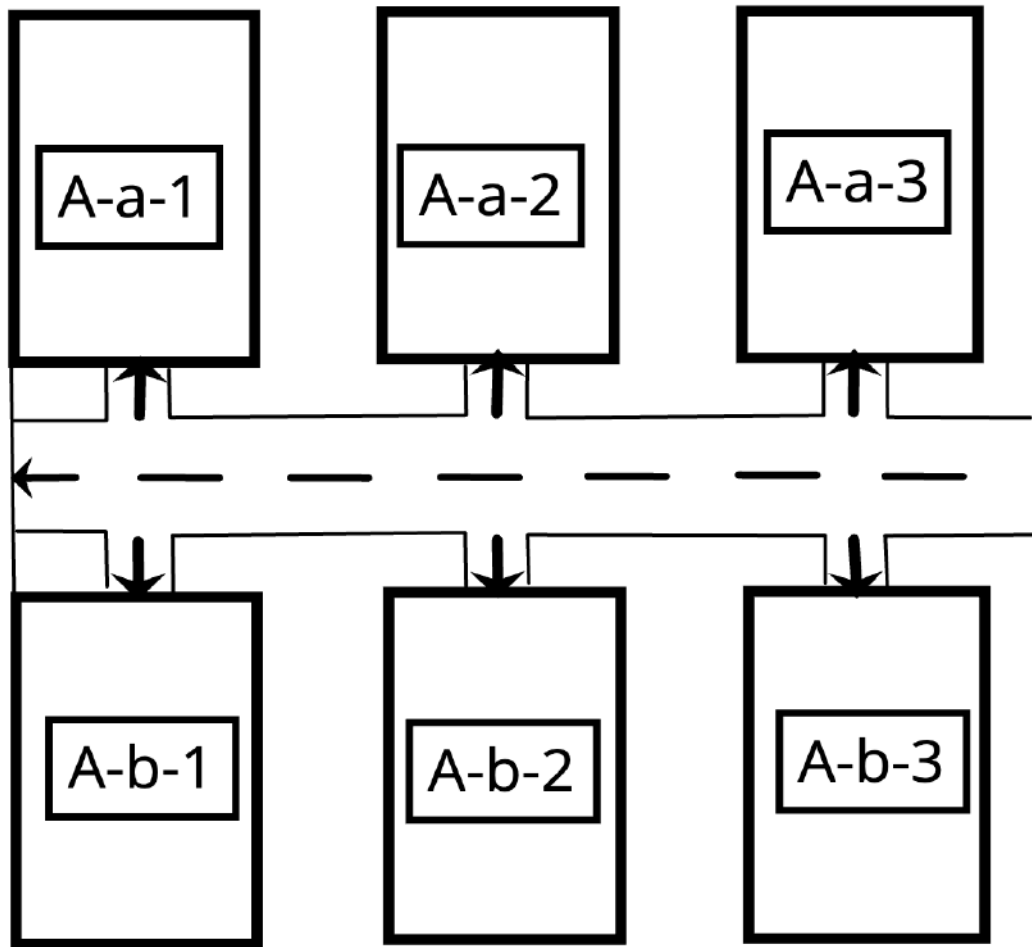
$$v = 2\pi r * 1000/m \quad (2.2)$$

Знаходиться похибка управління

$$e = v * -v \quad (2.3)$$

Застосовуючи один з алгоритмів управління зі зворотними зв'язками корегуються оберти колес.

Взаємозв'язки різного рівня контролерів при реалізації МР функції контролю в штучному оточені «Склад контейнерів» розглянемо на наступному прикладі. Нехай при виконанні УП після завершення поточного етапу МР знаходиться у позиції А-а-3



Рисинок 2.2 – Приклад маршруту переміщення

Планувальник сгенерував УП переміщення МР до вантажної позиції А-в-1. Приклад трьох типів УП наведено в таблиці 2.1. Методи програмного управління роботом демонструються на прикладі обмеженого набору команд управління: вперед, праворуч, ліворуч. В методі жорсткого програмного управління видача команди викликає початок відповідного руху. Цей рух триває до тих пір, поки не буде видана наступна команда вимкнути попередньо увімкнутий рух. Для методу жорсткого програмного управління вважається, що переміщення здійснюються з постійними швидкостями й тривають константний час. Наприклад, t_{A-a-3} секунд потрібно часу для переміщення МР від місця моніторингу номеру контейнера на позиції А-а-3 (рисинок 2.2) до лінії

розмітки А-а. До складу команд методу жорсткого програмного управління (інтерпретація програми управління роботом - жорстка послідовність виконання команд: і-а по порядку команда в програмі не може почати виконуватися, доки не буде завершено виконання і-1-ї команди) обов'язково входить затримки, наприклад, затримка t_{A-a-3} з параметром часу. Інтерпретатор команд контролеру при обробці цієї команди затримує на t_{A-a-3} одиниць часу початок виконання наступної команди.

Таблиця 2.1 - Програмне управління

Жорстке програмне управління	Гнучке програмне управління	Гнучке ситуаційне програмне управління
<i>вперед</i>	<i>вперед</i> $K1 S=120$	<i>вперед</i> $K1 R=f, S=50, V=1$
<i>затримка t_{A-a-3}</i>	<i>праворуч $\gamma=90^\circ$</i>	<i>праворуч $\gamma=90^\circ$</i>
<i>вимк_вперед</i>	<i>вперед</i> $K1 S=500$	<i>вперед</i> $K2 R=h, S=50, V=2$
<i>праворуч</i>	<i>ліворуч $\gamma=90^\circ$</i>	<i>ліворуч $\gamma=90^\circ$</i>
<i>затримка t_{90°</i>	<i>вперед</i> $K1 S=120$	<i>вперед</i> $K1 R=k, S=50, V=1$
<i>вимк_праворуч</i>	<i>стоп</i>	<i>стоп</i>
<i>вперед</i>		
<i>затримка t_{1-3}</i>		
<i>вимк_вперед</i>		
<i>ліворуч</i>		
<i>затримка t_{90°</i>		
<i>вперед</i>		
<i>затримка t_{A-b-1}</i>		
<i>вимк_вперед</i>		
<i>стоп</i>		

Метод жорсткого програмного управління вимогливий до попереднього впорядкування середовища: обов'язкові жорсткі вимоги до швидкості переміщення і миттєві режими розгону/гальмування робота. Ці вимоги для колесних роботів практично ніколи не можливо виконати. У методі програмного управління із зворотними зв'язками (гнучкого програмного управління) для ідентифікації події виходу робота на певні позиції використані дані від датчиків, а не інтервали часу. В таблиці 2.1

наведено приклад УП, в якій використовуються дані від одометрів (колес роботу ($S=120\text{см}$) і сервоприводу розвороту ($\gamma=90^\circ$). В УП указано що використовується контролер управління K1 (рух вздовж розмітки на підлозі на підставі даних від датчиків відображення). Умова вимикання команди задається конкретним значенням пройденої МР відстані, наприклад, $S=120\text{см}$ у першій команді вперед в таблиці 2.1. Контролер K1, отримуючи у реальному часі дані від драйверу одометра, розраховує поточну відстань (таблиця 2.1) й видає сигнал `вимкнути_вперед`, коли значення відстані дорівнює чи більше вказаного, тобто. $S \geq 120$.

В методі гнучкого ситуаційного програмного управління є можливість задавати будь яку ситуацію й будь який контролер. В прикладі (таблиця 2.1) на першому етапі УП використовується контролер K1, третій – контролер K2, п'ятий - знову K1. При реалізації першого етапу, крім контролера K1, паралельно працюють контролери швидкості ($V=1$) й відстані ($S=50$) та контролер руху вздовж смуги, який додатково відстежує наявність маркеру f на підлозі, що позначає перехрестя лінії розмітки від контейнера А-а-3 (рисунок 2.3) та лінії розмітки А-а у цьому. Ця умова указана в команді у вигляді $R=f$.

Також у системі моніторингу характеристик руху мобільного колісного роботу на базі датчика одометру H206 можна використовувати алгоритми розрахунку пройденого шляху та швидкості руху робота. Ці алгоритми можуть бути реалізовані на мікроконтролері, який підключений до датчика H206. При цьому, використовуючи дані, отримані від датчика, мікроконтролер може розраховувати координати розташування робота на площині.

Крім того, в системі можна використовувати алгоритми для оптимізації траєкторії руху робота на основі даних про пройдений шлях. Також можна реалізувати алгоритми для визначення кута повороту колеса робота, який допоможе вирішувати задачі навігації на поворотах та зміну напрямку руху робота.

Існує кілька алгоритмів для оптимізації траєкторії руху робота на основі даних про пройдений шлях. Нижче ми розглянемо кілька з них:

- алгоритм трьох точок - це один з найпростіших і найбільш ефективних алгоритмів для оптимізації траєкторії руху робота. Він використовує даний шлях, щоб визначити оптимальну траєкторію на основі трьох точок: поточної позиції робота, початкової точки та кінцевої точки маршруту;
- алгоритм кутів використовує даний шлях для визначення кутів між кожною парою точок. Далі він розраховує кут повороту робота на кожному етапі руху, щоб визначити оптимальну траєкторію;
- алгоритм мінімальної кривини - це алгоритм, який використовує даний шлях, щоб знайти оптимальну траєкторію з мінімальною кривинністю. Він розраховує кривину кожного сегменту шляху та вибирає траєкторію з найменшою кривинністю.

Алгоритми полягаються у використанні алгоритму A^* (A-star) для знаходження оптимальної траєкторії. A^* - це алгоритм пошуку шляху, що використовує оцінки відстані для знаходження найкоротшого шляху між вихідним вузлом та цільовим вузлом. В даному випадку вихідним вузлом є поточне положення робота, а цільовим вузлом може бути кінцева точка маршруту або точка, яка потребує особливої уваги, наприклад, перешкода на дорозі.

Алгоритм A^* використовує дві оцінки відстані: фактичну відстань від початку до поточної точки (g-функція), та передбачувану відстань від поточної точки до цільової точки (h-функція). Кожна точка має свій g- та h-значення, а ціна кожного можливого маршруту розраховується як сума цих значень.

Для використання алгоритму A^* потрібно мати граф, в якому кожна вершина - це можлива позиція робота на дорозі, а кожне ребро - це відрізок

між двома вершинами, який відповідає відстані між ними. Граф можна побудувати на основі даних, отриманих від датчика одометру.

Отримані дані про шлях та швидкість руху робота також можуть бути використані для визначення зносу коліс та вирішення питання заміни їх вчасно. Крім того, такі дані можуть бути корисними для діагностики проблем з роботом та попередження несправностей.

2.2 Концептуальна модель

Концептуальна модель системи моніторингу характеристик руху мобільного колісного роботу на базі датчика одометру - це відображення ключових елементів та взаємозв'язків між ними з точки зору виконання функцій системи. Вона базується на зборі та обробці даних від датчика одометру для вимірювання відстані, швидкості та інших параметрів руху мобільного колісного роботу.

Концептуальна модель - це опис взаємозв'язку елементів системи відповідно до її функцій та цілей. У контексті розробки системи моніторингу характеристик руху мобільного колісного роботу на базі датчика одометру, концептуальна модель описує загальну структуру та роботу системи.

Першим елементом системи є сам колісний робот, який рухається на певній відстані, та в якому встановлено датчик одометру, що фіксує кількість обертів колеса та на їх основі вимірює відстань, пройдену роботом.

Далі, інформація, зібрана датчиком, передається до мікроконтролера, який обробляє дані та розраховує поточну відстань та швидкість руху робота. При досягненні встановленої відстані, контролер видає сигнал, що робот повинен зупинитися.

У системі також є контролери руху, що реалізують управління переміщеннями роботу. Контролер вздовж розмітки на підлозі використовує дані від драйверу матриці датчиків, щоб обчислити відхилення робота від лінії розмітки, тобто похибку управління, та на її основі знаходить значення

управляючого сигналу. Аналогічно, два інших контролерів руху обчислюють похибки управління на основі даних від ультразвукового датчика та відео камери відповідно. Контролери реалізують управління зі зворотними зв'язками застосовуючи налаштовані ПІД алгоритми.

Датчик одометру H206 - це інфрачервоний датчик, який встановлюється на колесо робота та слугує для вимірювання довжини шляху, пройденого роботом. Він має вбудований енкодер, який дозволяє точно вимірювати кількість обертів колеса та відстань, пройдену роботом.

Датчик H206 має високу точність вимірювання відстані та швидкості, а також може працювати в широкому діапазоні температур. Для забезпечення надійної роботи датчика на колесі робота може бути встановлена спеціальна гумова обідна стрічка, яка забезпечує зчиплення колеса з поверхнею та запобігає зсуванню датчика з місця.

Для отримання даних з датчика H206 необхідно підключити його до контролера або мікроконтролера робота за допомогою спеціальних проводів. За допомогою програмного забезпечення можна зчитувати дані з датчика та аналізувати характеристики руху робота, такі як швидкість, відстань, кут повороту та інші параметри.

У концептуальній моделі на тему "системи моніторингу характеристик руху мобільного колісного роботу на базі датчику одометру" датчик H206 може бути використаний як основний датчик для вимірювання довжини шляху та швидкості руху робота. Дані з датчика можуть бути передані на контролер робота, де за допомогою алгоритмів обробки даних можуть бути отримані додаткові характеристики руху, наприклад, прискорення, траєкторія руху та інші параметри.

Отримані дані про пройдений шлях та швидкість руху робота за допомогою датчика одометру H206 можуть бути використані для контролю руху робота, оптимізації траєкторії руху та поліпшення точності навігації.

Діаграми варіантів використання (use case) допомагають візуалізувати основні сценарії взаємодії з системою. Основні варіанти використання

системи моніторингу можуть включати:

- 1) запуск системи та ініціалізація датчика одометру;
- 2) зчитування даних про пройдений шлях та обробка цих даних;
- 3) відображення характеристик руху, таких як швидкість, відстань та напрямок;
- 4) калібрування датчика одометру для підвищення точності вимірювання відстані;
- 5) збереження даних про рух для подальшого аналізу та використання;
- 6) оптимізація траєкторії руху на основі зібраних даних;
- 7) взаємодія з іншими системами або пристроями, які використовують дані про рух робота;
- 8) діагностика та обслуговування датчика одометру для забезпечення його надійності та точності вимірювання;
- 9) забезпечення безпеки руху робота шляхом контролю швидкості та вживання необхідних заходів.

Ці діаграми варіантів використання допомагають зрозуміти, як користувачі будуть взаємодіяти з системою та як вона виконуватиме свої функції. Вони можуть служити основою для подальшої розробки системи, уточнення вимог до неї та планування роботи з розробниками.

Ця концептуальна модель системи моніторингу характеристик руху мобільного колісного роботу на базі датчика одометру надає загальне уявлення про структуру та функціонування системи, її можливості та варіанти використання. Деталізація та реалізація системи можуть вимагати подальших технічних розробок, але ця модель служить вихідним пунктом для їхнього проведення.

Крім того, важливо врахувати аспекти обслуговування системи моніторингу характеристик руху. Для забезпечення надійності та точності вимірювань датчика одометру, необхідно регулярно проводити калібрування.

Цей процес включає визначення точного коефіцієнта вимірювання відстані залежно від впливу різних факторів, таких як стан дороги, знос шин, відхилення датчика тощо.

Крім того, додаткові датчики, такі як датчики інклінометра, можуть бути використані для врахування нахилу дороги та інших факторів, які можуть впливати на точність вимірювання відстані. Це дозволяє отримати більш точні дані про рух робота та покращити оптимізацію траєкторії.

На ринку доступні різноманітні датчики одометра з різними характеристиками та варіантами підключення. Деякі з них можуть бути підключені безпосередньо до мікроконтролера робота, що спрощує процес інтеграції. Інші датчики можуть використовувати додаткові інтерфейси, такі як CAN або RS-485, для передачі даних до системи моніторингу.

Усі ці можливості системи моніторингу характеристик руху мобільного колісного роботу на базі датчика одометру роблять її корисною для широкого спектру застосувань. Наприклад, вона може бути використана для автоматичного керування транспортними засобами, моніторингу вантажоперевезень, контролю за станом автопарку, відстеження маршрутів та багатьох інших сфер діяльності, де важлива точна інформація про рух мобільного робота. Така система може допомогти підвищити ефективність руху робота, зменшити витрати палива, збільшити точність навігації та забезпечити безпеку руху.

Для використання системи моніторингу характеристик руху необхідно розробити відповідне програмне забезпечення. Це може включати в себе розробку алгоритмів для зчитування даних з датчика одометру, їх обробки та аналізу. Також потрібно розробити інтерфейс користувача, який дозволить оператору взаємодіяти з системою, переглядати і аналізувати дані про рух робота.

Одним з варіантів використання системи може бути оптимізація траєкторії руху. На основі отриманих даних про пройдений шлях та інші характеристики руху, можна розробити алгоритми для вибору оптимальної

траєкторії з метою зменшення часу руху, економії палива або інших важливих факторів. Такі алгоритми можуть враховувати перешкоди на шляху, обмеження швидкості або інші фактори, що впливають на рух робота.

Для кращого розуміння можливостей системи моніторингу характеристик руху, нижче наведена діаграма варіантів використання:

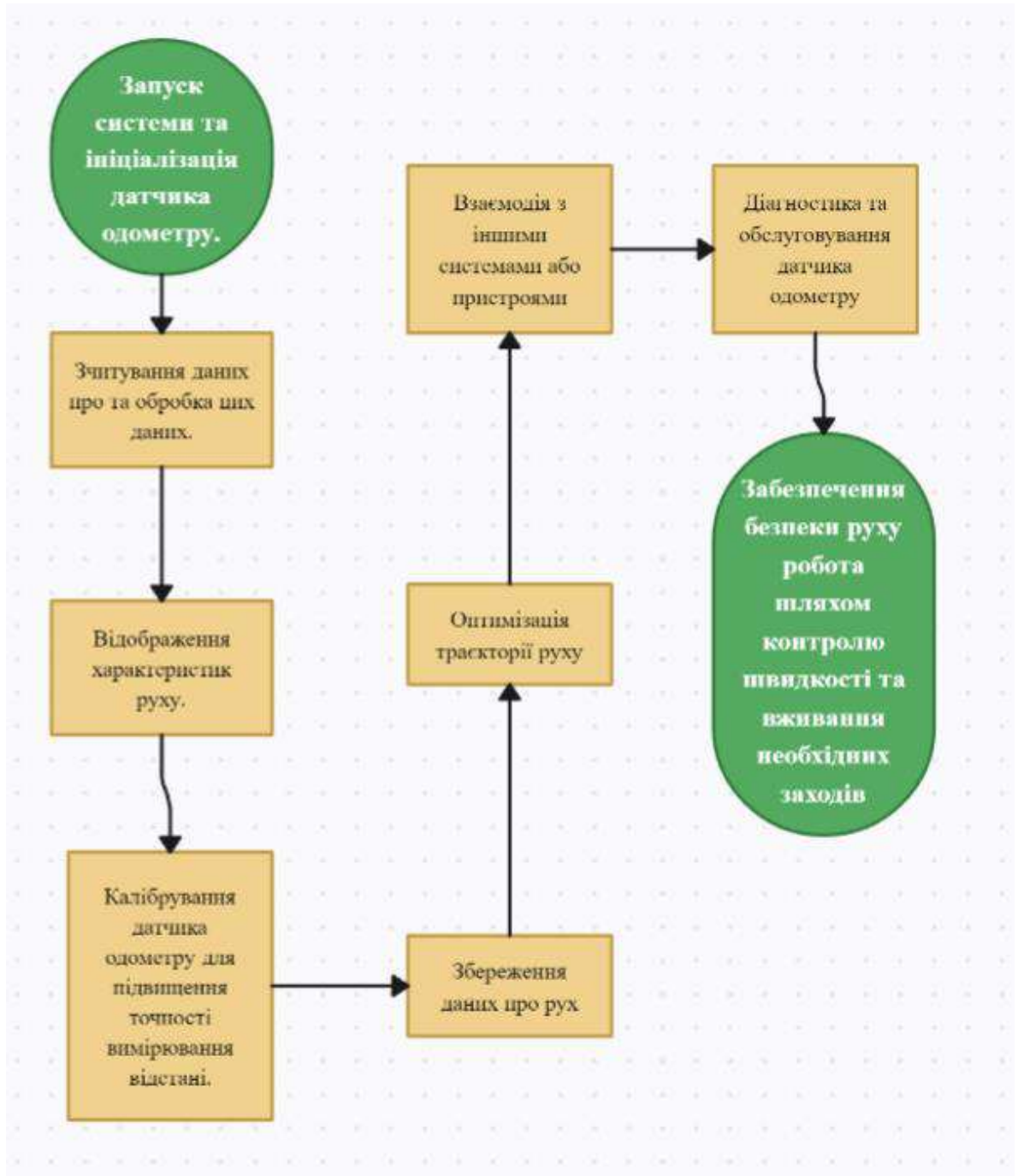


Рисунок 2.3 – Діаграма варіантів використання

Система моніторингу характеристик руху мобільного колісного робота на базі датчика одометру забезпечує оператора або систему автоматичного керування з потрібною інформацією про рух робота. Це дозволяє вдосконалити процес управління, підвищити продуктивність та забезпечити ефективну роботу робота в різних сферах застосування.

Діаграма варіантів використання відображає послідовність дій системи, починаючи з вимірювання даних про рух робота за допомогою датчика одометру. Потім дані обробляються та аналізуються для визначення характеристик руху, таких як шлях, швидкість, прискорення та інші. На основі цих даних виконуються алгоритми оптимізації траєкторії, які визначають оптимальний шлях руху робота з урахуванням важливих факторів. Нарешті, система виконує команди для керування роботом з метою проходження обраної траєкторії. Вона може бути використана як основа для подальшої розробки технічного проекту та впровадження системи у практичній діяльності.

2.3 Фізична модель системи

У фізичній моделі можуть бути включені схеми з платами, які використовуються для обробки сигналів з датчика одометру, розрахунків алгоритмів керування та взаємодії з іншими компонентами робота. Наприклад, плата мікроконтролера або FPGA може бути використана для збору даних, обчислень та передачі сигналів між різними компонентами системи.

У фізичній моделі можна враховувати такі компоненти як мотори, шасі робота, колеса, сенсори, контролери та інші елементи, які взаємодіють між собою для забезпечення руху робота та збору даних про його характеристики.

Фізична модель системи моніторингу руху робота на базі датчика одометру в середовищі MatLab є апроксимацією реальної системи і

використовується для аналізу та валідації алгоритмів керування та системи моніторингу. Ця модель дозволяє вірно відтворити основні принципи роботи системи, її взаємодію з навколишнім середовищем та виконання рухових завдань.

Датчик H206 підключається до піну переривання (цифрові піни 2 та 3 на рисунку 2.4) і його показання фіксує вбудована функція `attachInterrupt()`. Таким чином роль драйвера датчика одометр виконує ця функція.

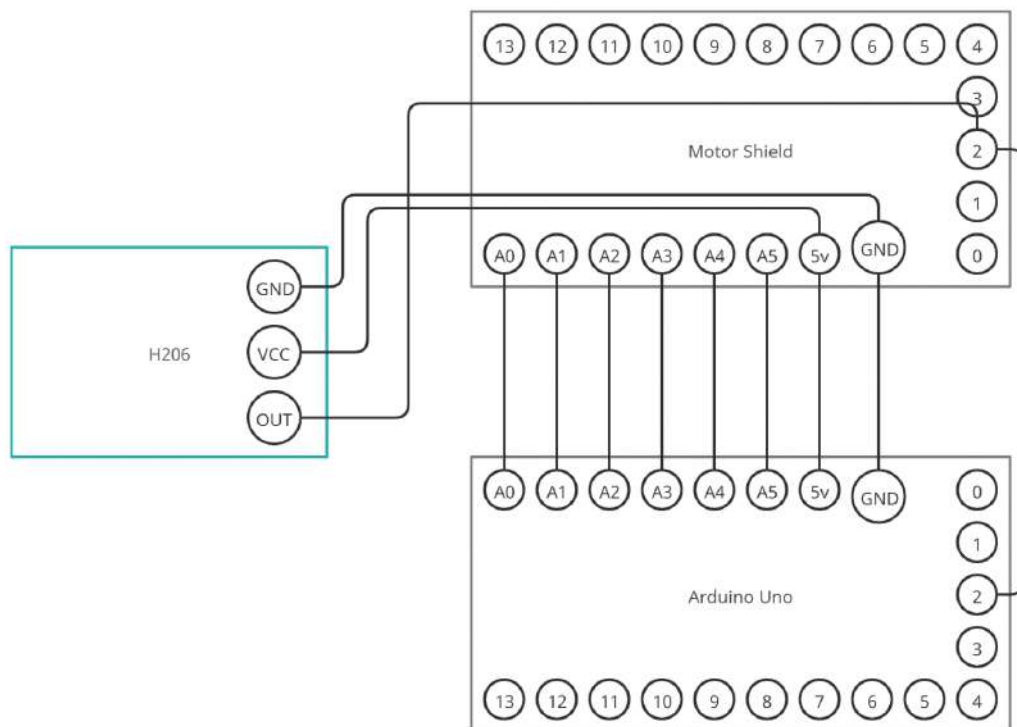


Рисунок 2.4 – Схема з'єднання компонентів на базі датчика H206

Контролер відстані видає по запиту поточне значення пройденої відстані (1) в метрах, реалізує функції обнулення $n=0$ в (1) та переривання при досягненні значення відстані S^* , яке задається.

Датчик одометру H206 - це електромеханічний датчик, який використовується для вимірювання відстані, пройденої роботом. Він зазвичай розташований на колесі робота і використовується для визначення кількості обертів колеса або переміщення робота на основі вимірювання кількості зроблених кроків або імпульсів. Механічна структура робота це

робот який повинен мати колеса, на яких розташований датчик одометру H206. Це можуть бути звичайні колеса або спеціальні колеса, що дозволяють точно виміряти переміщення. Мікроконтролер або пристрій управління. Для обробки сигналів, що надходять від датчика одометру, і виконання розрахунків щодо руху робота необхіден мікроконтролер або спеціалізований пристрій управління. Він отримує дані від датчика одометру і виконує розрахунки, такі як визначення відстані, швидкості, напрямку руху тощо.

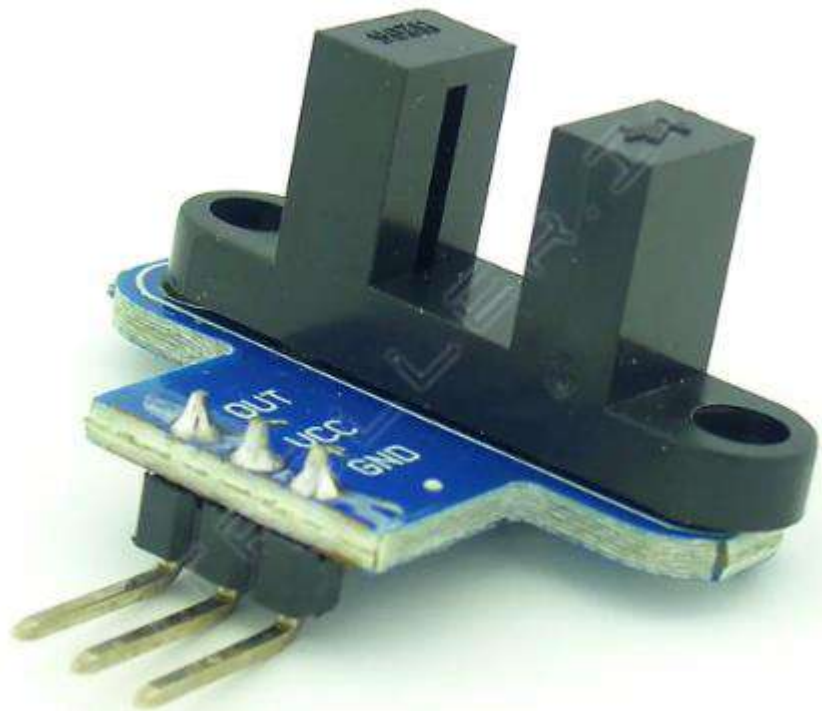


Рисунок 2.5 – Датчик H206



Рисунок 2.6 – Мікроконтролер Arduino UNO



Рисунок 2.7 – Мікроконтролер motor shield

Фізична модель системи моніторингу характеристик руху мобільного колісного робота з датчиком одометру H206 заснована на взаємодії цих компонентів. Датчик одометру H206 фіксує кількість обертів колеса і передає цю інформацію до мікроконтролера або пристрою управління. Мікроконтролер або пристрій управління виконує обробку даних і розрахунки щодо руху, враховуючи параметри датчика. Результати обробки даних, такі як відстань, швидкість і напрямок руху, можуть бути використані для моніторингу та керування рухом робота.

Ця фізична модель є важливим елементом системи моніторингу характеристик руху робота і може застосовуватись в різних сферах, включаючи робототехніку, автономну навігацію, мапінг та локацію, системи відслідковування тощо. Використання датчика одометру H206 дозволяє точно виміряти переміщення робота і забезпечує надійну інформацію для подальшого аналізу та керування рухом.

Установка датчиків (H206) трохи примхлива. Вони можуть встановлюватися тільки на двигуни, які мають вісь, що виступають з обох

кінців двигуна. Оскільки при такому кріпленні виходить, що і колесо, і пластина з поділками датчика змонтовані на одній осі, це позначатиме, що вони обертаються з однаковою швидкістю. Тобто, вимірюючи швидкість обертання пластини, ми можемо визначити швидкість обертання колеса. При установці переконайтеся в тому, що отвори (поділу) пластини потрапляють в дію інфрачервоного датчика, оскільки тільки він зможе визначити кількість отворів, що пройшло через нього. За бажання ви можете на свій смак модифікувати механічну частину проекту – головне, щоб дотримувалися зазначених умов.

Пластина з сітчастим градуюванням має 20 поділів (слотів, отворів). Це означає, що за час повного обороту колеса інфрачервоний датчик зможе виявити 20 пропусків (отворів).

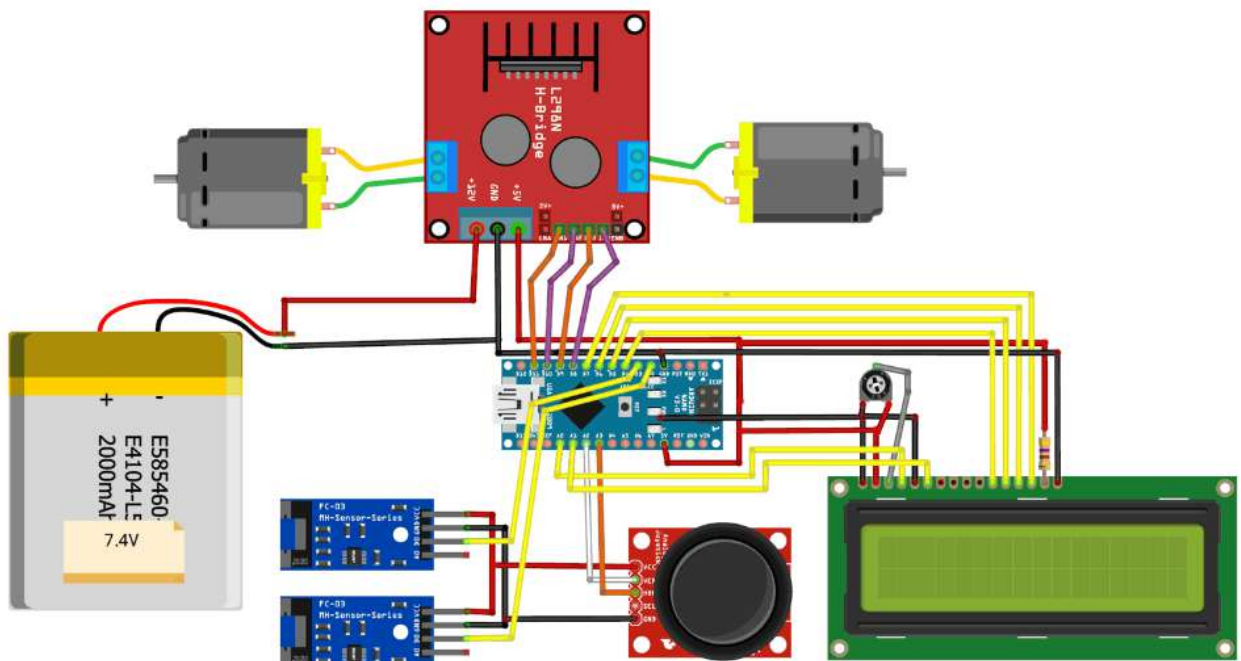


Рисунок 2.8 – Фізична модель мобільного роботу

Схема запитується від літійового елемента 7.4V. Ці 7.4V подаються на контакт 12V драйвера двигуна, а регулятор напруги драйвера двигуна перетворює їх на стабілізовану напругу +5V, яка використовується для живлення. Оскільки, як було вже зазначено, датчик H206 формус

керуючий вплив кожного разу, коли виявляється отвір у пластині з поділками. Для підвищення точності визначення швидкості ці управляючі дії будуть подаватись на входи зовнішніх переривань на контактах 2 та 3 плати Arduino.

2.4 Моделювання в MatLab методів керування переміщенням роботу

Моделювання керування переміщенням робота в системі моніторингу характеристик руху можна виконати в середовищі MatLab за допомогою різних методів.

Один з таких методів - це метод алгоритму. Він заснований на механізмі еволюції та використовується для знаходження оптимального рішення у великих просторах параметрів. Застосування алгоритмів може допомогти у виборі найкращих траєкторій руху робота.

Інший метод - це метод керування за допомогою PID-регулятора. Цей метод заснований на відслідковуванні заданої траєкторії та коригуванні помилок. Він може допомогти у підтримці постійної швидкості та уникненні зіткнень з перешкодами.

Також можна використовувати метод маршрутизації на основі алгоритму A* для знаходження найкоротшого шляху руху робота.

Для моделювання цих методів керування можна використовувати різні інструменти середовища MatLab, такі як Simulink, Robotics System Toolbox та інші. З їх допомогою можна створити фізичну модель робота з використанням датчика одометру та реалізувати різні методи керування переміщенням.

У середовищі MatLab можна створити фізичну модель системи моніторингу руху мобільного колісного робота на базі датчика одометру шляхом інтеграції датчика, керувальних алгоритмів та фізичних властивостей робота.

Одним з перших кроків є моделювання фізичної структури робота в

середовищі MatLab. Це можна зробити за допомогою 3D-моделювання або створенням просторової моделі з використанням математичних рівнянь. У цій моделі слід враховувати фізичні параметри робота, такі як розміри, маса, розташування коліс та датчика одометру.

Далі потрібно включити датчик одометру в модель. Це можна зробити, використовуючи математичні моделі для вимірювання відстані, яку пройшов робот, на основі обертів коліс та їх радіусів. Залежно від характеристик конкретного датчика одометру, можна використовувати математичні формули для перетворення обертів коліс у відстань.

Після цього слід враховувати алгоритми керування переміщенням робота. Наприклад, якщо використовується PID-регулятор, потрібно включити його модель, що враховує вхідні сигнали (поточну швидкість, помилку відслідковування траєкторії) та виконує розрахунки для керування швидкістю або напрямком руху робота.

Крім того, можна створити діаграму варіантів використання, яка ілюструє різні сценарії руху робота залежно від вхідних даних та керування. Наприклад, робот може рухатись прямо, поворотом на місці, зміною швидкості або зупинкою залежно від отриманих від датчика одометру даних та виконуваних алгоритмів керування.

Для моделювання таких алгоритмів керування в середовищі MatLab можна використовувати різні підходи, такі як символічне моделювання з використанням математичних рівнянь, числове моделювання з використанням алгоритмів та обчислень або гібридний підхід, комбінуючи обидва методи.

Залежно від конкретних вимог і цілей системи моніторингу руху робота, можна враховувати різні фактори, такі як динаміка руху, вплив зовнішніх сил, реакцію на перешкоди та інші аспекти.

При моделюванні в середовищі MatLab можна використовувати різні інструменти та пакети, які надаються для створення фізичної моделі та симуляції робота. Наприклад, MatLab Robotics System Toolbox надає

розширений функціонал для моделювання та симуляції роботів, включаючи рух колісних роботів.

Серед можливих кроків для моделювання в середовищі MatLab входять:

- 1) додавання датчика одометру;
- 2) розробка алгоритмів керування;
- 3) симуляція робота;
- 4) оптимізація та налаштування алгоритмів;
- 5) валідація моделі;
- 6) розширення функціоналу.

В рамках дослідження були реалізовані різні методи управління для системи моніторингу характеристик руху мобільного колісного роботу на базі датчика одометру. Основні методи управління включають жорстке програмне управління, гнучке програмне управління та гнучке ситуаційне програмне управління.

Жорстке програмне управління використовує фіксовані послідовності команд для керування роботом. У цьому методі робот виконує певні дії, такі як рух уперед, поворот на право або ліво, затримки на певний час і зупинка. Цей метод має просту реалізацію і підходить для простих завдань, де не потрібно враховувати зовнішні умови.

Лістинг 2.1 – Код у MATLAB для жорсткого програмного управління

```
function rigidControl()
    % Вперед
    moveForward();
    pause(tA-a-3);
    stopMotion();

    % Праворуч
    moveRight();
    pause(t90);
    stopMotion();
```

```

    % Вперед
    moveForward();
    pause(t1-3);
    stopMotion();

    % Ліворуч
    moveLeft();
    pause(t90);
    stopMotion();

    % Вперед
    moveForward();
    pause(tA-b-1);
    stopMotion();

    % Стоп
    stopMotion();
end

function moveForward() % Код для руху вперед
end

function moveRight() % Код для руху праворуч
end

function moveLeft() % Код для руху ліворуч
end

function stopMotion() Код для зупинки руху
end

```

Цей код реалізує жорстке програмне управління рухом мобільного колісного робота. Його основна мета полягає у керуванні рухом робота, виконуючи певну послідовність команд.

У функції `rigidControl` виконується послідовність рухів, яка описана коментарями в кодї. Наприклад, робот спочатку рухається вперед, потім повертає праворуч, знову рухається вперед, повертає ліворуч і так далі. Кожен рух виконується за допомогою виклику відповідних функцій руху, таких як `moveForward()`, `moveRight()`, `moveLeft()`. Після кожного руху робот зупиняється за допомогою функції `stopMotion()`.

Загальний алгоритм цього коду передбачає виконання послідовних рухів робота з періодами затримки `pause()`, які визначені в змінних `tA`, `t1`, `t90`, `a` і `b`.

Опис жорсткого програмного управління яке було реалізовано:

- 1) рух вперед здійснюється протягом певного часу, визначеного параметром t_A -а-3;
- 2) після цього рух вперед вимикається;
- 3) поворот праворуч здійснюється на кут 90° , затримка для якого встановлена параметром t_{90° ;
- 4) після повороту праворуч вимикається рух вправо;
- 5) цей процес повторюється для руху вперед, повороту ліворуч та зупинки руху згідно вказаних параметрів.

Лістинг 2.2 – Код у MATLAB для гнучкого програмного управління

```
function flexibleControl()
    % Вперед
    moveForwardK1(S=120);
    % Праворуч
    turnRight( $\gamma$ =90);
    % Вперед
    moveForwardK1(S=500);
    % Ліворуч
    turnLeft( $\gamma$ =90);
    % Вперед
    moveForwardK1(S=120);
    % Стоп
    stopMotion();
end
function moveForwardK1(S)
    % Код для руху вперед з параметром S
end
function turnRight( $\gamma$ )
    % Код для повороту праворуч з кутом  $\gamma$ 
end
function turnLeft( $\gamma$ )
    % Код для повороту ліворуч з кутом  $\gamma$ 
end
function stopMotion()
    % Код для зупинки руху
end
```

Даний код реалізує гнучке програмне управління рухом мобільного колісного робота. Його основна мета полягає у керуванні рухом робота, виконуючи певну послідовність команд.

У функції `flexibleControl` виконується послідовність рухів, описаних коментарями в кодї. Наприклад, робот спочатку рухається вперед з певною швидкістю (параметр S), потім повертає праворуч на певний кут (параметр γ), знову рухається вперед з іншою швидкістю, повертає ліворуч та знову рухається вперед. Кожен рух виконується за допомогою виклику відповідних функцій руху, таких як `moveForwardK1(S)`, `turnRight(γ)`, `turnLeft(γ)`. Після кожного руху робот зупиняється за допомогою функції `stopMotion()`.

Загальний алгоритм цього коду передбачає гнучке управління рухом робота з параметрами S і γ , які визначають швидкість руху та кут повороту відповідно.

Опис гнучкого програмного управління:

- 1) рух вперед здійснюється з встановленою швидкістю $S=120$;
- 2) поворот праворуч здійснюється на кут 90° ;
- 3) рух вперед здійснюється з встановленою швидкістю $S=500$;
- 4) поворот ліворуч здійснюється на кут 90° ;
- 5) рух вперед здійснюється з встановленою швидкістю $S=120$;
- 6) процес завершується зупинкою руху.

Лістинг 2.3 – Код у MATLAB для гнучкого ситуаційного програмного управління

```
function situationControl()
    % Вперед
    moveForwardK1(R=f, S=50, V=1);
    % Праворуч
    turnRight( $\gamma=90$ );
    % Вперед
    moveForwardK2(R=h, S=50, V=2);
    % Ліворуч
    turnLeft( $\gamma=90$ );
    % Вперед
    moveForwardK1(R=k, S=50, V=1);
    % Стоп
    stopMotion();
end
function moveForwardK1(R, S, V)
    % Код для руху вперед з параметрами R, S, V
end
function moveForwardK2(R, S, V)
```

```

    % Код для руху вперед з параметрами R, S, V
end
function turnRight( $\gamma$ )
    % Код для повороту праворуч з кутом  $\gamma$ 
end
function turnLeft( $\gamma$ )
    % Код для повороту ліворуч з кутом  $\gamma$ 
end
function stopMotion()
    % Код для зупинки руху
End

```

Наведений код реалізує програмне управління рухом мобільного колісного робота в різних ситуаціях. Кожна ситуація описана послідовністю команд руху, які виконуються роботом.

У функції `situationControl` робот виконує послідовність дій, описаних коментарями в коді. Наприклад, робот спочатку рухається вперед з певними параметрами (R , S , V), потім повертає праворуч на певний кут (γ), знову рухається вперед з іншими параметрами, повертає ліворуч та знову рухається вперед. Кожен рух виконується за допомогою виклику відповідних функцій руху, таких як `moveForwardK1(R, S, V)`, `moveForwardK2(R, S, V)`, `turnRight(γ)`, `turnLeft(γ)`. Після кожного руху робот зупиняється за допомогою функції `stopMotion()`.

Загальний алгоритм цього коду передбачає програмне управління рухом робота в різних ситуаціях, залежно від значень параметрів R , S , V та γ .

Опис гнучкого ситуаційного програмного управління:

- 1) рух вперед здійснюється залежно від ситуації з параметрами $R=f$, $S=50$ та $V=1$;
- 2) поворот праворуч здійснюється на кут 90° ;
- 3) рух вперед здійснюється залежно від ситуації з параметрами $R=h$, $S=50$ та $V=2$;
- 4) поворот ліворуч здійснюється на кут 90° ;
- 5) рух вперед здійснюється залежно від ситуації з параметрами $R=k$, $S=50$ та $V=1$;
- б) процес завершується зупинкою руху.

Для реалізації цих управляючих програм було використано середовище MatLab. У коді програми були визначені різні кроки руху, повороти та затримки на встановлені часові проміжки. Кожен крок виконувався з відповідними параметрами, такими як швидкість руху (S), кут повороту (γ) та інші. Код був написаний з урахуванням потрібних логічних операторів та команд для управління роботом.

Після реалізації управляючих програм було проведено моделювання цих методів у середовищі MatLab. За допомогою моделювання було встановлено, як робот виконує кожну команду та як впливають різні параметри на його рух.

Отримані результати моделювання підтвердили коректність реалізації управляючих програм. Кожен метод демонстрував очікувану поведінку робота, здатність виконувати різні рухи та змінювати параметри руху залежно від ситуації.

3 РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Реалізація системи

Система моніторингу характеристик руху мобільного колісного роботу на базі датчика одометру включає наступні компоненти:

- 1) датчик одометру, що фіксує кількість обертів колеса;
- 2) мікроконтролер, що збирає дані з датчика одометру та інших датчиків, відповідальних за моніторинг руху та його характеристик, та обробляє їх;
- 3) додаткові датчики, такі як акселерометр, гіроскоп, компас, тискометр тощо, для збору додаткової інформації про рух транспортного засобу;
- 4) акумулятор або джерело живлення, яке забезпечує енергію для всіх компонентів системи;
- 5) бездротовий модуль передачі даних, який забезпечує зв'язок між системою моніторингу та зовнішнім обладнанням, таким як мобільний додаток або сервер;
- 6) комп'ютер або мобільний пристрій, на якому відображається інформація про характеристики руху транспортного засобу;
- 7) програмне забезпечення, що забезпечує обробку та відображення даних з системи моніторингу.

Логічна модель системи моніторингу характеристик руху мобільного колісного роботу на базі датчика одометру може бути представлена у вигляді блок-схеми, яка ілюструє взаємодію компонентів системи та логіку обробки та передачі даних. Наприклад, на блок-схемі можна виділити окремі блоки для збору даних з датчиків, їх обробки та аналізу, передачі даних,

відображення даних тощо. Кожен блок може бути детальніше розглянутий, вказавши конкретні функції та алгоритми, що виконуються в процесі роботи системи моніторингу.

- 1) Датчик одометра - виконує функцію вимірювання кількості обертів колеса. Для цього він використовує вбудований енкодер або оптичний датчик, який фіксує кількість обертів колеса та передає ці дані мікроконтролеру.
- 2) Мікроконтролер - виконує функцію обробки даних, що надходять від датчика. Він зчитує сигнал від датчика, розраховує відстань та зберігає дані в пам'яті. Також мікроконтролер може мати вбудований модуль Bluetooth або Wi-Fi для передачі даних на зовнішній пристрій.
- 3) Дисплей - виконує функцію відображення вимірної відстані та інших параметрів руху, які можуть бути розраховані на основі даних з датчика одометра.
- 4) Живлення - забезпечує живлення мікроконтролера та дисплея. Зазвичай використовуються літій-іонні або нікель-кадмієві акумулятори.
- 5) Конденсатор - використовується для стабілізації напруги в колі живлення. Він забезпечує постійну напругу для мікроконтролера та дисплея, що забезпечує їх правильну роботу.
- 6) Резистор - використовується для обмеження струму в колі живлення та забезпечення безпеки пристрою.

Усі ці блоки разом утворюють логічну модель системи моніторингу характеристик руху мобільного колісного роботу на базі датчика одометра. Дана модель може бути розширена шляхом додавання нових компонентів для забезпечення додаткових функцій, наприклад, вимірювання швидкості руху, визначення напрямку руху, аналіз відхилень від заданого маршруту та б.і.

Одним зі способів покращення функціональності системи моніторингу руху є додавання GPS-модуля для визначення географічних координат руху транспортного засобу. Це дозволить не тільки визначити точні координати транспортного засобу, але й визначити швидкість руху та прокласти оптимальний маршрут.

Також можна використовувати додаткові сенсори, такі як датчики тиску в шинах, датчики температури двигуна, датчики вібрації, для моніторингу стану транспортного засобу та вчасного виявлення потенційних проблем.

Контролери руху реалізують управління переміщеннями роботу окремих етапів плану переміщення. Контролери руху вздовж розмітки на підлозі використовує дані від драйверу матриці датчиків відображення на їх підставі обчислюється відхилення робота від лінії розмітки, тобто похибка управління, й на її основі знаходить значення управляючого сигналу. Аналогічно два інших контролерів руху обчислюються похибки управління на основі даних від ультразвукового датчику й відео камери, відповідно. Контролери реалізують управління зі зворотними зв'язками застосовуючи налаштовані ПД алгоритм чи нечіткі алгоритми, такі як нечіткий лінійний статичний із зоною нечутливості й насиченням, нечіткий трьохпозиційний, нечіткий нелінійний із гістерезисом й насиченням, нечіткий ПД.

Контролер програмного управління виконує функцію менеджера окремих етапів плану досягнення цільової позиції. Маршрут від поточної точки місцезнаходження робота до цільової розбивається на окремі етапи, послідовність яких уявляє управляючу програму роботу. Контролер програмного управління активізує етапи управляючої програми шляхом активізації команди що вказана, контролює завершення виконання етапу (успішне чи аврійне), та приймає рішення щодо активізації наступного етапу. В управляючій програмі (УП) можуть бути етапи з різними методами управління, на відміну від класичного програмного управління. Етапи що реалізують жорстке програмне управління, коли умовою завершення етапу є

час (в опису етапа УП задається інтервал часу) можуть перемішуватися з етапами що реалізують гнучке програмне управління, коли умовою завершення етапу є пройдена відстань, чи відстань до перешкоди, чи наявність певного маркера що ідентифікується камерою зору. Контролер цього рівня виконує моніторинг даних від відповідних датчиків (одометр, ультразвуковий та відео) й порівнює поточні дані з заданими в УП. При виконанні умов, вказаних в описі етапу, ініціюється завершення етапу УП. Жорстке програмне управління порівняно з гнучким менш ресурсоємне але може застосовуватися в умовах відсутності перешкод за підтримки контролерів нижчого шару що мають високі значення критерію якості управління.

Обов'язковою компонентою що належить до четвертого шару компонент зі штучним інтелектом є планувальник завдань, який в залежності від стану (ситуації) штучного оточення генерує УП, як послідовність етапів досягнення цілі. Планувальник завдань реалізує метод безперевного планування за прецедентами. Безперевне планування передбачає що після завершення кожного етапу УП (успішно чи аварійно) планувальник виходячи з поточної ситуації по новому генерує план досягнення цілі. Такий підхід забезпечує високий рівень автономності прийняття рішень в умовах збурень та неповної інформації.

3.2 Опис використаних компонентів

Розробка алгоритмів для зчитування даних з датчика одометру, їх обробки та аналізу. Одним з варіантів використання системи це оптимізація траєкторії руху. На основі отриманих даних про пройдений шлях та інші характеристики руху, можна розробити алгоритми для вибору оптимальної траєкторії з метою зменшення часу руху, економії палива або інших важливих факторів. Такі алгоритми можуть враховувати перешкоди на шляху, обмеження швидкості або інші фактори, що впливають на рух робота.

Для кращого розуміння можливостей системи моніторингу характеристик руху, нижче наведена діаграма варіантів використання:



Рисунок 3.1 – Діаграма використання компонентів та обробки системи

Ця діаграма починається з вимірювання даних про рух робота, після чого дані піддаються обробці та аналізу. На основі цих даних


```

function rigidControl()

    % Під'їзд до рядка A-b,
    if ~strcmp(currentRow, 'A-b')
        moveToRow('A-b');
        currentRow = 'A-b';
    end

    % Перевірка ситуації у контейнері A-b-7
    if strcmp(currentContainer, 'A-b-7')
        disp('Робот знаходиться у контейнері A-b-7');
        return;
    end

    % Визначення розмірів контейнера A-b-7 (в метрах)
    containerLength = 4;
    distanceBetweenContainers = 2;

    % Визначення відстані до цільового контейнера A-b-7
    targetDistance = (7 - str2double(currentContainer(end))) *
        (containerLength + distanceBetweenContainers);

    % Жорсткі кроки для під'їзду до контейнера A-b-7
    moveForward(targetDistance);      % Рух вперед на відстань
targetDistance
    delay('tA-b-7');                 % Затримка на tA-b-7
    stopMotion();                     % Зупинка руху
    turnRight();                      % Рух праворуч
    delay(90);                        % Затримка на 90 градусів
    moveForward(containerLength);     % Рух вперед на довжину
контейнера
    delay(2);                          % Затримка на 2 секунди
    stopMotion();                     % Зупинка руху

    % Стоп
    stopMotion();

end

% Функція для руху вперед на задану відстань
function moveForward(distance)
    % Код для руху вперед на задану відстань
end

% Функція для затримки на певний час (в секундах)
function delay(time)
    % Код для затримки
end

% Функція для руху праворуч
function turnRight()
    % Код для руху праворуч
End

```

У цьому прикладі використовуються функції `moveForward`, `delay`, `turnRight`, `turnLeft` і `stopMotion`, які виконують відповідні дії. Кроки руху включають рух вперед на відстань `targetDistance`, затримку `tA-b-7`, зупинку руху, рух праворуч, затримку на 90 градусів, рух вперед на довжину контейнера `containerLength`, затримку на 2 секунди та зупинку руху. У цьому прикладі використовуються функції `moveForward`, `delay`, `turnRight`, `turnLeft` і `stopMotion`, які виконують відповідні дії. Кроки руху включають рух вперед на відстань `targetDistance`, затримку `tA-b-7`, зупинку руху, рух праворуч, затримку на 90 градусів, рух вперед на довжину контейнера `containerLength`, затримку на 2 секунди та зупинку руху.

`moveForward(distance)`: функція для руху робота вперед на задану відстань `distance`. У реальному виконанні ця функція повинна бути реалізована залежно від апаратного забезпечення робота.

`Delay(time)`: функція для затримки виконання програми на певний час `time` (у секундах). `turnRight()`: функція для руху робота праворуч. У реальному виконанні ця функція повинна бути реалізована залежно від апаратного забезпечення робота. `turnLeft()`: функція для руху робота ліворуч. У реальному виконанні ця функція повинна бути реалізована залежно від апаратного забезпечення робота. `stopMotion()`: функція для зупинки руху робота. У реальному виконанні ця функція повинна бути реалізована залежно від апаратного забезпечення робота.

Таким чином мобільний робот успішно виконув усі дії які були зазначені методом жорсткого програмного управління для під'їзду.

Такоже було розглянуто гнучке програмне управління для під'їзду робота до контейнера A-c-1 у середовищі Matlab з використанням розмірів контейнера та вказаними кроками.

Лістинг 3.2 – Гнучне програмне управління для під'їзду робота до контейнера А-с-1

```

% Гнучке програмне управління для під'їзду робота до контейнера
А-с-1

% Визначення параметрів руху
moveForwardK1 = 4; % Відстань для руху вперед (в метрах)
S = 2; % Затримка після руху вперед (в секундах)
γ = 90; % Кут повороту (в градусах)

% Початок програми
disp('startdprog');

% Рух вперед
disp('Рух вперед. ');
moveForward(впередK1);

% Затримка після руху вперед
disp('Затримка після руху вперед. ');
delay(S);

% Поворот праворуч або ліворуч (залежно від значення γ)
if γ >= 0
    disp(['Поворот праворуч на ', num2str(γ), ' градусів.']);
    turnRight(γ);
else
    disp(['Поворот ліворуч на ', num2str(abs(γ)), '
градусів.']);
    turnLeft(abs(γ));
end

% Зупинка руху
disp('Зупинка руху. ');
stopMotion();

% Кінець програми
disp('Кінець програми. ');

```

Цей гнучкий алгоритм програмного управління дозволяє роботу ефективно під'їжджати до контейнера А-с-1 з урахуванням вказаних параметрів руху.

Розгляд гнучкого ситуативного програмне управління для під'їзду робота до контейнера А-а-2 у середовищі Matlab з використанням розмірів контейнера та вказаними кроками.

Лістинг 3.3 – Гнучне ситуаційне програмне управління для під'їзду робота до контейнера А-а-2 описано нижче

```
% Задані параметри руху
впередK1 = 4; % Відстань для руху вперед (4 метра)
міжКонтейнерами = 2; % Відстань між контейнерами (2 метра)
S = 2; % Затримка після руху (2 секунди)
V = 0.5; % Швидкість руху (0.5 метра/секунда)

% Початок програми
disp('Початок програми');

% Функція руху вперед на задану відстань
moveForward = @(distance) distance / V;

% Рух до контейнера А-с-1
впередK1_R = moveForward(впередK1 + міжКонтейнерами);
disp(['Рух вперед на відстань ', num2str(впередK1_R), '
секунд']);

% Затримка після руху вперед
disp(['Затримка на ', num2str(S), ' секунд']);
pause(S);

% Зупинка руху
disp('Зупинка руху');

% Кінець програми
disp('Кінець програми');
```

Це використовується для зазначення параметрів руху та розраховується також час руху робота до контейнера А-с-1. Після досягнення цільової точки виконується затримка, а потім рух зупиняється. Зазначений код вище демонструє приклад реалізації програмного управління у MatLab з використанням вказаних параметрів.

Отже, було проведено три перевірки пересування робота у задану точку. Всі дії за методами управління були успішно проведені та виконані. Реалізація жорсткого, гнучкого та гноткого ситуативного програмного управління були задіяні.

3.3 Опис компонентів реалізації

Програмна реалізація складається з двох частин і призначена для локалізації мобільного робота на одновимірній карті. Основна частина коду реалізована в середовищі MATLAB на комп'ютері. Друга частина коду написана в середовищі Arduino IDE і завантажена на мобільну платформу.

Алгоритм руху мобільного робота по лінії та передача сигналів з датчиків виконується в другій частині коду на мобільній платформі. Для перевірки працездатності алгоритму була створена карта контейнерів, що описують одновимірний простір, по якому рухається мобільний робот.

Опис програми методу програмного управління переміщеннями робота розділяється на три компоненти і реалізований за допомогою MATLAB та Arduino IDE.

1) Підготовка програмного коду управління на персональному комп'ютері для подальшого використання.



```
sketch_dec06a $
#include <AFMotor.h>

AF_DCMotor motor1(1);
AF_DCMotor motor2(2);
AF_DCMotor motor3(3);
AF_DCMotor motor4(4);

void setup() {
  motor1.setSpeed(200);
  motor2.setSpeed(200);
  motor3.setSpeed(200);
  motor4.setSpeed(200);

  motor1.run(FORWARD);
  motor2.run(FORWARD);
  motor3.run(FORWARD);
  motor4.run(FORWARD);

  delay(4000);

  motor1.run(RELEASE);
  motor2.run(RELEASE);
  motor3.run(RELEASE);
  motor4.run(RELEASE);
}
```

Рисунок 3.3 – Програмний код

2) Завантаження кодів програми управління роботом в пам'ять мікроконтролера Arduino.

У процесі реалізації програмного управління роботом, спочатку ми маємо завантажити відповідні коди програми управління в пам'ять мікроконтролера Arduino. Це виконується за допомогою середовища розробки Arduino IDE.

Arduino IDE є спеціальним інтегрованим середовищем розробки, яке дозволяє писати, збирати та завантажувати програми управління на мікроконтролери Arduino. Це дозволяє нам створити програму для робота, яка буде виконувати певні дії та реагувати на зовнішні події.

Після встановлення Arduino IDE та підключення мікроконтролера Arduino до комп'ютера за допомогою USB-кабелю, ми можемо завантажити програму управління на мікроконтролер Arduino. Для цього ми обираємо потрібну платформу Arduino та порт, до якого підключений мікроконтролер, та натискаємо кнопку "Завантажити". Arduino IDE виконає процес завантаження програми управління на мікроконтролер.



Рисунок 3.4 – Завантаження коду

3) Розроблена програма взаємодіє з іншими пристроями апаратного комплексу через систему підготовки кодів керуючої програми. Для формування коду керуючої програми використовується середовище Arduino та відповідні бібліотеки. Бібліотека ultrasonic була вибрана як відповідна та буда описана нижче.

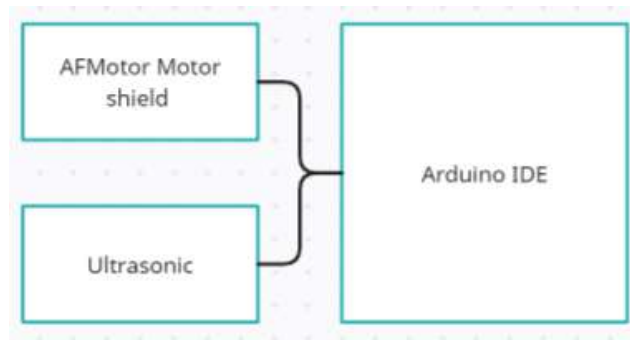


Рисунок 3.5 – Бібліотеки

Основні функції бібліотеки Ultrasonic.h:

- оголошуємо об'єкт control с параметрами;
- оголошуємо об'єкт control() з одним параметром;
- Controlodm, control(),situationControlrigidControl, flexibleControl();
- читаємо значення відстані;
- `int dist = ultrasonic.distanceRead();`
- `int dist = ultrasonic.distanceRead(INC).`

Бібліотека AFMotor Motor shield надає набір основних функцій для керування двигунами з використанням AFMotor Motor shield. Основні функції цієї бібліотеки включають:

- `AF_DCMotor`: Ця функція ініціалізує об'єкт для керування одним змінним постійним струмом (DC) двигуном, підключеним до AFMotor Motor shield;
- `setSpeed`: Ця функція встановлює швидкість обертання двигуна.
- `run`: Ця функція визначає напрямок обертання двигуна. Доступні значення параметра включають FORWARD (вперед), BACKWARD (назад) і RELEASE (вимкнено);
- `runSingle`: Ця функція дозволяє керувати кожним окремим мотором.

ВИСНОВКИ

У порівнянні з існуючими аналогами, система моніторингу характеристик руху мобільного колісного роботу на базі датчика одометру має свої переваги, такі як висока точність вимірювань, незалежно від ходу.

Була проаналізована апаратна база для виконання розробки та визначено необхідні компоненти для її реалізації. Також було проведено аналіз технологій для вирішення поставленої задачі, визначено переваги та недоліки різних методів збору, обробки та аналізу даних.

У ході аналізу методологічного підґрунтя було визначено, що для успішної реалізації системи моніторингу потрібно враховувати фізичні властивості руху мобільного колісного роботу та використовувати різні методи обробки та аналізу даних для отримання точної та повної інформації про рух.

Отже, можна зробити висновок, що система моніторингу характеристик руху мобільного колісного роботу на базі датчика одометру є важливою технологією для вимірювання та аналізу рухових параметрів роботів. Для успішної реалізації системи потрібно враховувати апаратну базу, технології збору, обробки та аналізу даних, а також фізичні властивості руху мобільного колісного роботу. Правильне використання цієї системи може допомогти в підвищенні продуктивності та ефективності роботів, а також покращенні безпеки праці та економії ресурсів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Крупський А.В. Робот як об'єкт вивчення та інструмент пізнання у навчальному процесі з інформатики // Освітній вісник «Свідомість». 2018. №10
2. Роботизація: огляд сфери та досвід впровадження в компаніях [Електронний ресурс].
3. Алгоритми локальної навігації і картографії для бортової системи управління автономного мобільного робота 2012 / Кучерської Роман Володимирович, Манько Сергій Вікторович
4. Роботи. Велика енциклопедія / - Ексмо, 2017. - 144с.
5. Лук'янов Андрій Анатолійович Методи управління рухом експериментального мобільного сервісного робота // Вісник ІРДТУ. 2014. №3
6. Ельперін І. В. Автоматизація виробничих процесів. Ліра-К, 2017. 378 с.
7. Ata A. A. Autonomous mobile robot for mine detection. Alexandria University, 2015. P. 607–608.
8. Ричіхіна Е. Н. Показники ефективності управлінського моніторингу муніципальної освіти. Ухта : УГТУ, 2007. 47 с.
9. Функціональне призначення та класифікація складів в логістиці // WareTeKa News, 13.11.2020, <https://wareteka.com.ua/uk/blog/klasyfikaciya-skladiv-v-logistyci/>
10. M. Czerwinski, J. Hernandez, D. McDuff, "Building an AI that feels," Appl. Sci., vol.11, 4920, Apr. 2021, doi:10.3390/app11114920.
11. T. Zhang et al., "Current trends in the development of intelligent unmanned autonomous systems," Frontiers Inf. Technol. Electron. Eng., vol. 18, Feb. 2017, pp. 68–85, doi: 10.1631/FITEE.1601650.
12. Абделькефі, А., та Маареф, Х. (2017). Проектування та управління автономним мобільним роботом з використанням одометрії та сенсорного злиття. Вісник управління, автоматики та електротехніки, 28(1), 36-49.

13. Чен, Л., Чжао, Ж., та Чжу, Г. (2018). Планування траєкторії мобільного робота на основі одометрії та сенсорного злиття. Вісник фізики: Збірник наукових праць, 1059(1), 012101.
14. "Introduction to Odometry" на веб-сайті RobotShop: <https://www.robotshop.com/community/tutorials/show/introduction-to-odometry>
15. How to Use Encoder and Odometer on DFRobot 4WD Arduino-Compatible Platform: <https://www.dfrobot.com/blog-369.html>
16. Siegwart, R., & Nourbakhsh, I. R. (2004). Introduction to Autonomous Mobile Robots. MIT Press.
17. Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic Robotics. MIT Press.
18. Wang, S., & Mei, T. (2015). Visual odometry for mobile robot navigation: A survey. Journal of Sensors, 2015, 595171.
19. Zhang, Y., Zhang, J., & Zhu, Y. (2019). A robust odometry estimation method for mobile robots based on the extended Kalman filter. IEEE Access, 7, 4864-4874.
20. Бронніков А. І., Руденко І. В. Система моніторингу автоматизованих систем на підприємстві. Виробництво & Мехатронні Системи 2022: матеріали -ої Міжнародної VI конференції : Міжнар. конф., м. Харків, 21 жовт. 2022 р. 2022. С. 71.
21. Розумний Інтернет речей – хто він та з чим його їдять? [Електронний ресурс].–Режим доступу до ресурсу: <https://harabhabr.ua/post/259243/>