

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій  
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

другий (магістерський)  
(рівень вищої освіти)

Розроблення методу управління рухом мобільної платформи  
з чотирма рулевими колесами

(тема)

Виконав:

студент 2 курсу, групи КТРСМ-21-1

Белов П.О.

(прізвище, ініціали)

Спеціальності 151 Автоматизація та  
комп'ютерно-інтегровані технології

(код і повна назва спеціальності)

Тип програми Освітньо-професійна

Освітня програма Комп'ютеризовані та  
робототехнічні системи

(повна назва освітньої програми)

Керівник проф. Новоселов С.П.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри КІТАМ

(підпис)

Невлюдов І. Ш.

(прізвище, ініціали)

2022 р.

## ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Факультет	Автоматики і комп'ютеризованих технологій
Кафедра	Комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки
Рівень вищої освіти	другий (магістерський)
Спеціальність	151 Автоматизація та комп'ютерно-інтегровані технології
Тип програми	освітньо-професійна
Освітня програма	Комп'ютеризовані та робототехнічні системи (код і повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри Невлюдов І.Ш  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2022 р.

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Белову Петру Олександровичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення методу управління рухом мобільної платформи з чотирма рулевими колесами

затверджена наказом по університету від 07.11. 2022 р. № 1462 Ст.

2. Термін подання студентом роботи до екзаменаційної комісії 23.12. 2022 р.

3. Вихідні дані до роботи кінематична модель рухомої платформи з чотирма колесами, що можуть змінювати кут повороту; кут повороту коліс: 180 градусів; кількість рулевих коліс: 4 шт;

спосіб управління: за допомогою крокових двигунів.

4. Перелік питань, що потрібно опрацювати в роботі: аналіз предметної області та технічного завдання; розробка методу управління рулевими колесами; опис розробленого методу управління рухом мобільної платформи з чотирма рулевими колесами; експериментальні дослідження.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) Демонстраційний матеріал представлений у форматі презентації PowerPoint (\*.pptx) – 17 с.

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

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

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на виконання кваліфікаційної роботи	15.09.2022	Виконано
2	Визначення актуальності роботи, мети, предмета, об'єкта, та розробка завдань для досягнення мети	22.09 – 30.09.22	Виконано
3	Аналіз літературних джерел	01.10 – 13.10.22	Виконано
4	Вибір середовища розробки	15.10 – 3.11.22	Виконано
5	Оформлення пояснювальної записки та документації	05.11 – 19.11.22	Виконано
6	Оформлення додатків	20.11 – 01.12.2	Виконано
7	Оформлення графічного та презентаційних матеріалів комп'ютерного захисту	01.12 – 06.12.22	Виконано
8	Представлення роботи на рецензування	14.12 – 19.12.22	Виконано

Дата видачі завдання 15.09.2022

Студент

(підпис)

Керівник роботи

(підпис)

Белов П.О.

( прізвище, ініціали)

проф. Новоселов С.П.

(посада, прізвище, ініціали)

## РЕФЕРАТ

Пояснювальна записка: 98 с., 2 табл., 47 рис., 3 дод., 30 джерел.

### АВТОМАТИЗАЦІЯ, МОБІЛЬНИЙ РОБОТ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, РУЛЬОВІ КОЛЕСА, КРОКОВИЙ ДВИГУН.

Мета роботи – розроблення методу визначення орієнтації коліс в залежності від траєкторії руху мобільної платформи.

Об'єкт розробки – автоматизована система керування рухом мобільної платформи.

Предмет розробки – метод визначення траєкторії руху роботу з чотирма рульовими колесами для досяжності заданої точки на карті місцевості.

Виконано огляд сучасних існуючих мобільних роботів з чотирма рулевими колесами, виявлено та проаналізовано їх недоліки та переваги, розроблено програмне забезпечення для мобільного робота з чотирма рулевими колесами.

## ABSTRACT

Explanatory note: 98 p., 2 tab., 47 fig., 3 adj., 30 sources.

AUTOMATION, MOBILE ROBOT, SOFTWARE, STEERING WHEELS, STEPPER MOTOR.

The purpose of the work is to develop a method for determining the orientation of the wheels depending on the trajectory of the mobile platform.

The object of development is an automated system for controlling the movement of a mobile platform.

The subject of development is a method of determining the trajectory of a robot with four steering wheels to reach a given point on the terrain map.

An overview of the current existing mobile robots with four steering wheels was performed, their disadvantages and advantages were identified and analyzed, and software for a mobile robot with four steering wheels was developed.

## ЗМІСТ

Перелік скорочень .....	8
Вступ.....	9
1 Аналіз предметної області та технічного завдання .....	11
1.1 Аналіз використання мобільних роботів у сучасних виробничих системах .....	11
1.2 Аналіз аналогічних конструкцій мобільних роботів.....	14
1.2.1 Rosbot Pro Nano - 4WIS .....	14
1.2.2 Naio Technologies – weeding robot Oz .....	16
1.2.3 Naio Technologies – weeding robot for large-scale vegetable crops Dino .....	17
1.3 Аналіз програмного забезпечення мобільних роботів.....	18
1.3.1 ROS.....	18
1.3.2 Arduino .....	21
1.3.3 RobotC .....	23
1.3.4 Raspberry Pi OS.....	24
1.4 Основи рулевого управління .....	26
1.4.1 Керування триколісним роботом з рулевим управлінням.....	27
1.4.2 Рулеве керування робота з причепом .....	27
1.4.3 Чотириколісний робот з двома рульові колесами .....	29
1.4.4 Непаралельні рульові колеса .....	30
1.4.5 Чотириколісний робот з чотирма рулевими колесами .....	30
1.4.6 Паралельне рульове управління .....	31
1.4.7 Непаралельне рульове управління .....	32
1.5 Системи рулевого керування .....	33
1.6 Висновки до першого розділу .....	35
2 Розробка методу управління рульовими колесами .....	36

2.1	Опис конструкції незалежних поворотних колісних модулів.....	36
2.1.1	Аналіз аналогічних рішень .....	36
2.1.2	Ескізне проектування модуля поворотних коліс.....	42
2.2	Розробка алгоритму управління поворотним колісним модулем.....	43
2.3	Розробка алгоритму синхронного управління комплексом поворотних колісних модулів .....	45
2.4	Висновки до другого розділу .....	47
3	Опис розробленого методу управління рухом мобільної платформи з чотирма рульовими колесами .....	48
3.1	Висновки до третього розділу .....	55
4	Експериментальні дослідження.....	56
4.1	Постановка задачі проведення експерименту.....	56
4.2	Опис програмного засобу для виконання експериментальних досліджень .....	56
4.3	Аналіз результатів дослідження.....	61
4.4	Висновки до четвертого розділу .....	65
5	Охорона праці.....	66
5.1	Висновки до п'ятого розділу.....	68
	Висновки .....	69
	Перелік джерел посилання .....	70
	Додаток А Лістинг коду .....	73
	Додаток Б Демонстраційний матеріал.....	88
	Додаток В Відомість кваліфікаційної роботи.....	97

## ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

4WIS – 4 Wheel Independent Suspension;

API – application programming interface;

IDE – integrated drive electronics;

IRC – Instantaneous Center of Rotation;

OC – operating system.

## ВСТУП

На даний час підрозділ робототехніки, який займається створенням мобільних роботів, швидко розвивається.

Робототехніка – це прикладна наука, яка займається розробкою автоматизованих технічних систем і є найважливішою технічною базою для інтенсифікації виробництва. В даний час велика кількість завдань пов'язана з організацією автоматичного руху робота, виконанням деяких основних дій. Серед різних класів роботів важливим сучасним класом є мобільний робот, який розміщений на мобільній платформі, що використовується в промисловості (робототехніка), екології (динамічний моніторинг навколишнього середовища, робота в екологічно шкідливих умовах), військовій сфері (розвідувальний робот), побутовий робот (автономні пирососи, газонокосарки тощо).

Однак створити роботів, які впевнено рухаються навіть на рівній поверхні, де перешкоди для них нездоланні, важко з кількох причин, включаючи недосконалість алгоритмів управління. Надійний алгоритм управління рухом електроприводів мобільних роботів є критично важливим для успішної роботи.

Завдання управління рухом мобільних колісних роботів з чотирма рулевими колесами, крім теоретичного значення, стає дедалі практичнішим у різних галузях сільського господарства, техніки та науки. Коло завдань, які потрібно вирішити, дуже широке, наприклад: робот, який прополюватиме грядки, або збиратиме врожай на полі. Високі вимоги до характеристик руху роботів вимагають використання математичних та комп'ютерних моделей на всіх етапах синтезу алгоритмів управління. Тому дослідження в цій галузі є актуальним завданням.

Мета роботи – розроблення методу визначення орієнтації коліс в

залежності від траєкторії руху мобільної платформи.

Об'єкт розробки – автоматизована система керування рухом мобільної платформи.

Предмет розробки – метод визначення траєкторії руху роботу з чотирма рульовими колесами для досяжності заданої точки на карті місцевості.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз технічного завдання та предметної області;
- провести аналіз існуючих аналогів;
- проаналізувати і написати необхідний код програми;
- експериментальних досліджень розроблено програмний засіб, що дозволяє перевірити правильність запропонованих математичних виразів та методики розрахунку куту повороту ведучих коліс в залежності від координат цільової точки на карті місцевості;
- наочно змодельовати та візуалізувати траєкторію руху мобільної платформи для досягнення вказаної точки на карті місцевості.

Робота виконана згідно [1-3].

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ТЕХНІЧНОГО ЗАВДАННЯ

## 1.1 Аналіз використання мобільних роботів у сучасних виробничих системах

Мобільний робот – це автоматична машина, яка здатна пересуватися. Мобільна робототехніка зазвичай вважається підгалуззю робототехніки та інформаційної інженерії [4].

Мобільні роботи мають можливість пересуватися в навколишньому середовищі й не прикріплені до одного фізичного місця. Мобільні роботи можуть бути «автономними», що означає, що вони здатні орієнтуватися в неконтрольованому середовищі без потреби у фізичних або електромеханічних пристроях наведення. Крім того, мобільні роботи можуть покладатися на пристрої навігації, які дозволяють їм подорожувати заздалегідь визначеним навігаційним маршрутом у відносно контрольованому просторі. Навпаки, промислові роботи зазвичай більш-менш стаціонарні, складаються із з'єднаної руки (багатоланкового маніпулятора) і вузла захоплення (або кінцевого ефектора), прикріпленого до нерухомої поверхні.

Мобільні роботи стали більш звичним явищем у комерційних і промислових умовах. Лікарні вже багато років використовують автономних мобільних роботів для переміщення матеріалів. На складах встановлено мобільні роботизовані системи для ефективного переміщення матеріалів із складських полиць до зон виконання замовлень. Мобільні роботи також є основною темою поточних досліджень, і майже в кожному великому університеті є одна або кілька лабораторій, які зосереджуються на дослідженні мобільних роботів. Мобільні роботи також зустрічаються в промислових, військових і охоронних установах.

Компонентами мобільного робота є контролер, датчики, виконавчі механізми та система живлення. Контролер, як правило, є мікропроцесором, вбудованим мікроконтролером або персональним комп'ютером (ПК). Використовувані датчики залежать від вимог робота. Вимогами можуть бути розрахунок мертвої точки, тактильне визначення та визначення наближення, триангуляційне визначення дальності, уникнення зіткнень, визначення місця розташування та інші спеціальні програми. Приводи зазвичай відносяться до двигунів, які рухають робота на колесах або ногах. Для живлення мобільного робота ми зазвичай використовуємо джерело постійного струму (тобто акумулятор) замість змінного [5].

Застосування технології чотирьох рульових механізмів на сільськогосподарському обладнанні можна простежити з початку 1920-х років. Патент описував автоматичні системи рульового керування, які могли направляти трактор слідувати за борознами через поле ферми. З розвитком обчислювальних блоків і сенсорних технологій технологія автокермування набула набагато ширшого застосування в сільському господарстві. Численні звітні дослідження. Перевірено, що автоматичне рульове управління може допомогти досягти високої точності водіння, зберігаючи високу швидкість водіння та водночас звільняючи оператора від монотонного водіння, щоб зменшити втому та покращити продуктивність водіння.

Більшість технологій автоматичного керування для звичайного сільськогосподарського обладнання базувалися на рульовому керуванні Акермана, механізмі рульового керування автомобілеподібного транспортного засобу, який зазвичай є неголономною системою. Допустимі напрямки руху обмежені для неголономних систем через їх кінематичні обмеження. Таким чином, коли враховуються обмеження навколишнього середовища, такі як обмежені шляхи руху та перешкоди на обмежених робочих майданчиках, планування шляху та маневрування обладнанням для автоматичних неголономних систем стає складним.

Мобільна платформа з чотирма рулевими колесами для використання в фруктових садах або на полях з посівами повинна ефективно та безпечно маневрувати в проходах, утворених лабіринтами з дерев, а також швидко та точно направляти всередину та з цих проходів. Однією з унікальних особливостей роботи у фруктовому саду є його обмежений простір, що робить виправлення пози на проході або керування транспортним засобом у проході без численних спроб і виправлень дуже складним, якщо обладнання використовує лише механізм керування Аккермана. Незалежне рульове керування з чотирма колесами (4WIS) – це одне з рішень, яке здатне покращити як керованість, так і маневреність садових транспортних засобів, що працюють у обмеженому просторі. При проектуванні роботизованого мобільного садового обладнання розробка потужних систем автоматичного керування для транспортних засобів за допомогою 4WIS є дуже важливою. В останні роки повідомлялося про дослідження щодо розробки необхідних технологій [6].

Також існує роботизована платформа з системою 4WIS для виявлення бур'янів. Його навігаційний контролер використовує дві контрольні точки (передню та задню частини) незалежно, щоб мінімізувати відстань, необхідну для визначення траєкторії. Також є автономний з 4WIS з різними режимами керування для реалізації відповідних стратегій керування для руху між рядами та розвороту на смузі поля [7].

Щоб створити рішення для ефективного та безпечного керування контейнерами для фруктів, дослідники розробили роботизовану систему bin-dog із системою 4WIS для транспортування контейнерів у фруктових садах. Система 4WIS здатна перевертати кожне колесо незалежно, щоб підвищити маневреність до всепрямованого шляхом належної координації обертання всіх коліс. Така здатність дозволяє роботу обертатися в будь-якому миттєвому центрі обертання і забезпечує можливість керувати контейнером у обмеженому середовищі фруктового саду. Ефективна система керування

рульовим управлінням із bin-dog відіграє важливу роль у досягненні необхідного відстеження траєкторії та ефективності маневрування контейнером, на яку впливають поздовжня швидкість руху, режими рульового керування, посилення контролю та шаблони траєкторії [8].

## 1.2 Аналіз аналогічних конструкцій мобільних роботів

В даний час представлено безліч різних мобільних роботів, які необхідні на необхідні у сільському господарстві, наприклад, для збирання врожаю на полях. Ось деякі варіанти таких роботів:

### 1.2.1 Rosbot Pro Nano - 4WIS

Rosbot Pro 4WIS (4 Wheel Independent Suspension) – ідеальна платформа для проєктів автономних мобільних роботів на відкритому повітрі. Завдяки вбудованому контролеру ROS, LiDAR і камері глибини Rosbot Pro 4WIS економить час і гроші для швидкого створення прототипів або розробки продукту. Час автономної роботи може досягати 4 год без навантаження або 2,5 год з повним навантаженням. Це серйозна професійна платформа для інтеграції з польовими роботами та сервісними роботами як для внутрішнього, так і зовнішнього середовища. Він може нести до 45 кг корисного навантаження та підніматися на схил до 30°. Rosbot Pro 4WIS — одна з найгнучкіших платформ, яка може виконувати великі роботи за низькими витратами [9].

На рис. 1.1 – зовнішній вигляд Rosbot Pro Nano – 4WIS.



Рисунок 1.1 – Rosbot Pro Nano – 4WIS [10]

Технічні характеристики робота Rosbot Pro Nano – 4WIS:

- ROS готовий до швидкого прототипування або розробки продукту;
- система незалежної підвіски 4 коліс;
- у комплекті LiDAR і камера глибини;
- максимальне корисне навантаження 45 кг;
- термін служби батареї від 2,5 год до 4 год;
- контролер ROS – Jetson Nano;
- камера глибини Orbbec;
- тверді гумові колеса – діаметр 254 мм;
- потужність сервоприводу – 100 Вт;
- вартість: 4605\$.

### 1.2.2 Naio Technologies – weeding robot Oz

Робот-прополовач Oz призначений для прополки та розпушування бур'янів. Oz повністю автономний, також можна керувати ним, за допомогою пульта керування. Oz обмежує важку роботу по господарству. Робот-прополовач бере на себе відповідальність: прополює бур'яни, може мотижити та допомагає під час збирання врожаю. Автоматизований робот для прополки бур'янів Oz самостійно прополює та обробляє урожай [11].

На рис. 1.2 – зовнішній вигляд робота Oz.



Рисунок 1.2 – Oz – помічник у сільському господарстві для трудомістких завдань [12]

Технічні характеристики робота Oz:

- довжина: 130 см;
- ширина: 47 см;
- висота: 83 см;
- драйвер двигуна: ТВ6612;
- вага: 120° Wide-angle USB Camera;
- тяга: 300 кг;
- вартість: 43000\$.

### 1.2.3 Naïo Technologies – weeding robot for large-scale vegetable crops Dino

Робот Dino розроблений, щоб легко вирощувати овочі на великих овочевих фермах. Його головна перевага полягає в тому, що він працює автономно, тож у вас є більше часу для більш цінних завдань. Dino – перший повністю автономний електричний робот для механічної прополки овочевих культур. Робот розроблений компанією Naïo technology і представлений на ринку з початку 2017 року. На сьогоднішній день вироблено понад 20 екземплярів робота Діно [13].

На рис. 1.3 – зовнішній вигляд робота Dino.



Рисунок 1.3 – Робот Naïo Technologies – Dino [14]

Технічні характеристики робота Naio Technologies – Dino:

– інструмент із керуванням камери: робота на кількох зелених і червоних культурах;

– міжцентрова відстань коліс: від 160 см до 220 см;

– автономність: від 8 год до 10 год;

– швидкість: до 4 км/год;

– вага: 1,25 т (2500 фунтів);

– вартість: 124000\$.

### 1.3 Аналіз програмного забезпечення мобільних роботів

Програмне забезпечення мобільного робота – це сукупність програм системи обробки інформації та програмних документів, необхідних для експлуатації цього робота.

Сьогодні існує безліч ОС, програм і фреймворків для того, щоб програмувати плати і мікроконтролери на різних операційних системах і мовах програмування. Нижче будуть приведені найбільш популярні з них, а також обрано ПЗ для виконання магістрської кваліфікаційної роботи.

#### 1.3.1 ROS

Роботна операційна система (ROS) – це набір проміжного програмного забезпечення для робототехніки з відкритим кодом. Незважаючи на те, що ROS не є операційною системою (ОС), а набором програмних інфраструктур для розробки програмного забезпечення роботів, вона надає послуги, розроблені для неоднорідного комп'ютерного кластера, такі як апаратна абстракція, низькорівневе керування пристроєм, реалізація загальноживаних функцій, передача повідомлень між процесами та керування пакетами. Запущені набори процесів на основі ROS представлені в графівій архітектурі,

де обробка відбувається у вузлах, які можуть отримувати, розміщувати та мультиплексувати дані датчиків, повідомлення про контроль, стан, планування, приводи та інші повідомлення. Незважаючи на важливість реактивності та низької затримки в управлінні роботом, ROS не є операційною системою реального часу (RTOS). Однак можна інтегрувати ROS з кодом реального часу. Відсутність підтримки систем реального часу було вирішено у створенні ROS 2, основного перегляду ROS API, який використовуватиме переваги сучасних бібліотек і технологій для основних функцій ROS і додасть підтримку коду реального часу та вбудованої системи обладнання [15].

Програмне забезпечення в екосистемі ROS можна розділити на три групи:

- незалежні від мови та платформи інструменти, що використовуються для створення та розповсюдження програмного забезпечення на основі ROS;
- реалізації клієнтської бібліотеки ROS, такі як `roscpp`, `rospy` та `roslisp`;
- пакети, що містять програмний код, який використовує одну або декілька клієнтських бібліотек ROS.

Як незалежні від мови інструменти, так і основні клієнтські бібліотеки (C++, Python і Lisp) випущені згідно з умовами ліцензії BSD, і як такі є програмним забезпеченням з відкритим кодом і безкоштовним як для комерційного, так і для дослідницького використання. Більшість інших пакетів ліцензовано за різними ліцензіями з відкритим кодом. Ці інші пакети реалізують такі загальноживані функції та додатки, як апаратні драйвери, моделі роботів, типи даних, планування, сприйняття, одночасну локалізацію та відображення, засоби моделювання та інші алгоритми [16].

Основні клієнтські бібліотеки ROS орієнтовані на Unix-подібну систему, головним чином через їх залежність від великих колекцій залежностей програмного забезпечення з відкритим кодом. Для цих клієнтських бібліотек Ubuntu Linux зазначено як «Підтримується», тоді як інші варіанти, такі як Fedora Linux, macOS і Microsoft Windows, позначені як «експериментальні» та

підтримуються спільнотою. Рідна клієнтська бібліотека Java ROS, rosjava, однак не поділяє цих обмежень і дозволяє писати програмне забезпечення на основі ROS для ОС Android. rosjava також дозволила інтегрувати ROS в офіційно підтримуваний інструментарій MATLAB, який можна використовувати в Linux, macOS і Microsoft Windows. Також була розроблена клієнтська бібліотека JavaScript, roslibjs, яка дозволяє інтегрувати програмне забезпечення в систему ROS через будь-який веб-браузер, сумісний зі стандартами [17].

На рис 1.4 – приклад інтерфейсу ROS.

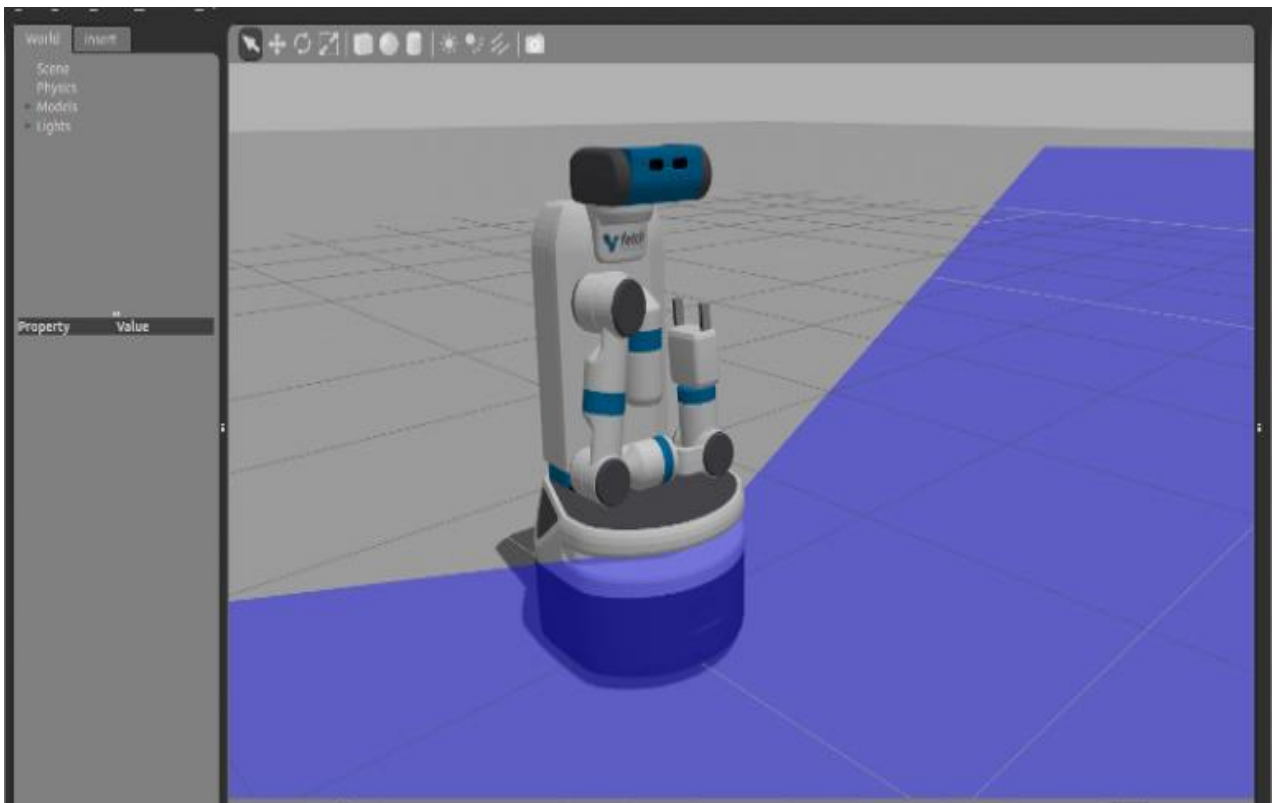


Рисунок 1.4 – Приклад інтерфейсу ROS

### 1.3.2 Arduino

Arduino – це компанія з відкритим кодом апаратного та програмного забезпечення, проект і спільнота користувачів, яка розробляє та виробляє одноплатні мікроконтролери та комплекти мікроконтролерів для створення цифрових пристроїв. Його апаратні продукти ліцензуються згідно з ліцензією CC BY-SA, тоді як програмне забезпечення ліцензується відповідно до GNU Lesser General Public License (LGPL) або GNU General Public License (GPL), що дозволяє будь-кому виготовляти плати Arduino та розповсюджувати програмне забезпечення. Плати Arduino доступні на офіційному веб-сайті або через авторизованих дистриб'юторів.

Мікроконтролери можна програмувати за допомогою мов програмування C і C++, використовуючи стандартний API, який також відомий як мова Arduino, натхненний мовою обробки та використовується з модифікованою версією IDE обробки. На додаток до використання традиційних ланцюжків інструментів компілятора, проект Arduino надає інтегроване середовище розробки (IDE) і інструмент командного рядка, розроблений у Go [18].

Програма для обладнання Arduino може бути написана на будь-якій мові програмування з компіляторами, які створюють двійковий машинний код для цільового процесора. Atmel надає середовище розробки для своїх 8-розрядних мікроконтролерів AVR і 32-розрядних ARM Cortex-M: AVR Studio (старіший) і Atmel Studio (новіший).

Інтегроване середовище розробки Arduino (IDE) – це кросплатформна програма (для Microsoft Windows, macOS і Linux), написана мовою програмування Java. Він походить від IDE для мов Processing і Wiring. Він містить редактор коду з такими функціями, як вирізання та вставлення тексту, пошук і заміна тексту, автоматичний відступ, зіставлення фігурних дужок і підсвічування синтаксису, а також забезпечує прості механізми компіляції та

завантаження програм на плату Arduino одним клацанням миші. Він також містить область повідомлень, текстову консоль, панель інструментів із кнопками для загальних функцій та ієрархію робочих меню. Вихідний код для IDE випущено під GNU General Public License, версія 2.

Arduino IDE підтримує мови C і C++ за допомогою спеціальних правил структурування коду. Arduino IDE постачає бібліотеку програмного забезпечення з проекту Wiring, яка забезпечує багато загальних процедур введення та виведення. Написаний користувачем код потребує лише двох основних функцій, для запуску ескізу та основного циклу програми, які компілюються та зв'язуються за допомогою заглушки програми `main()` у виконувану циклічну виконавчу програму з ланцюжком інструментів GNU, також включену в дистрибутив IDE. Arduino IDE використовує програму `avrdude` для перетворення виконуваного коду в текстовий файл у шістнадцятковому кодуванні, який завантажується в плату Arduino програмою-завантажувачем у мікропрограмі плати [19].

На рис. 1.5 – приклад інтерфейсу Arduino IDE.



```
Arduino - 0011 Alpha
File Edit Sketch Tools Help

Blink

/*
 * Blink
 *
 * The basic Arduino example. Turns on an LED on for one second,
 * then off for one second, and so on... We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;           // LED connected to digital pin 13

void setup()              // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()               // run over and over again
{
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000);                // waits for a second
  digitalWrite(ledPin, LOW);  // sets the LED off
  delay(1000);                // waits for a second
}

Done compiling.

Binary sketch size: 1098 bytes (of a 14336 byte maximum)
```

Рисунок 1.5 – Приклад інтерфейсу Arduino IDE

### 1.3.3 RobotC

RobotC – це інтерпретована C-подібна мова програмування, доступна для VEX PIC і VEX Cortex. Подробиці:

- підтримується VEX;
- інтерпретований байт-код;
- превентивна багатозадачність;
- власні бібліотеки;
- працює з коробки, вбудована IDE;
- замінено Midnight C.

RobotC – це інтерпретована C-подібна мова програмування, доступна для VEX PIC і VEX Cortex. Вона пропонує налагодження в режимі реального часу та відносно простий набір внутрішніх бібліотек, які переважно базуються на використанні (реальних або віртуальних) масивів [20]. Особливості:

- вбудований відладчик дозволяє призупиняти та відновлювати код користувача та відстежувати змінні в реальному часі;
- IDE об'єднує оновлення вбудованого програмного забезпечення, створення коду, компіляцію та перепрошивку в одну програму;
- вбудована примітивна багатозадачність;
- підтримка доступна на веб-сайті та від VEX;
- простий і легкий в освоєнні, користувачам не потрібно цікавити себе технічними деталями.

Недоліки:

- інтерпретатор емулює 8-розрядний процесор, суттєво сповільнюючи код користувача та запобігаючи використанню будь-яких інтенсивних чисельних алгоритмів;
- бібліотека абстрагує багато деталей і запобігає розширеному доступу або тонкому налаштуванню датчиків VEX;
- не є ані безкоштовним, ані відкритим кодом і працює лише в Windows;

- відсутність підтримки об'єктів, динамічного розподілу, широких цілих чисел або переривань сильно обмежує можливості коду;
- синтаксис, подібний до масиву, може заплутати і рясніє можливостями для індексів поза межами або переповненням;
- оптимізатор поганий, а IDE загалом низької якості, із фіксованим стилем коду та жахливим завершенням коду [21].

На рис 1.6 – приклад інтерфейсу RobotC.

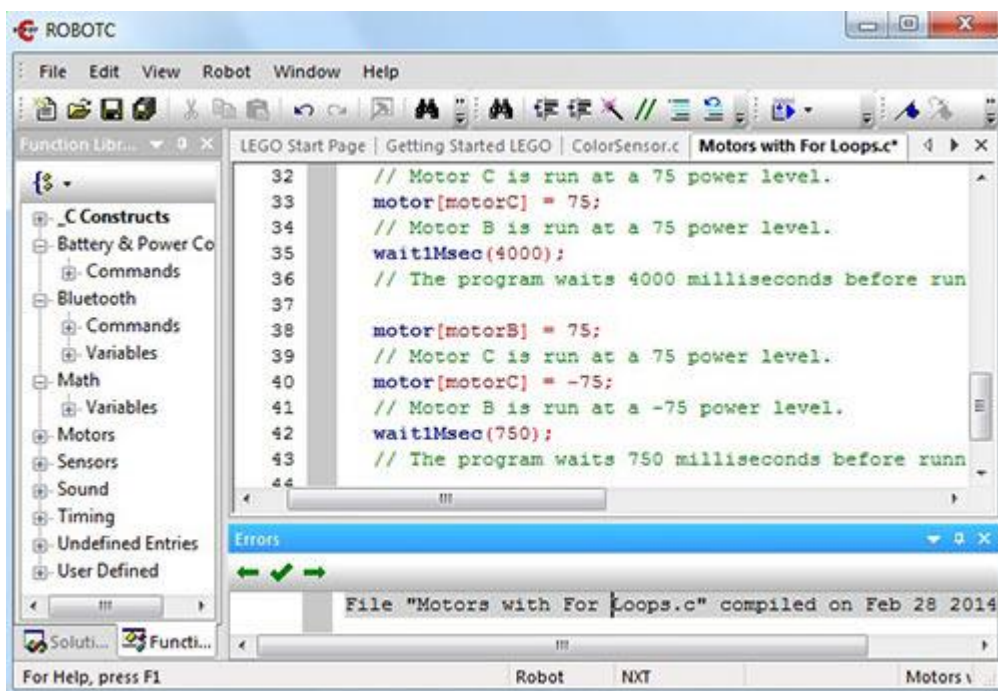


Рисунок 1.6 – Приклад інтерфейсу RobotC

### 1.3.4 Raspberry Pi OS

ОС Raspberry Pi (раніше Raspbian) – це Unix-подібна операційна система, заснована на дистрибутиві Debian Linux для сімейства компактних одноплатних комп'ютерів Raspberry Pi. Вперше розроблена незалежною системою в 2012 році, вона виробляється як основна операційна система для цих плат з 2013 року, розповсюджується Raspberry Pi Foundation.

ОС Raspberry Pi оптимізована для Raspberry Pi з процесорами ARM. Він працює на кожному Raspberry Pi, крім мікроконтролера Pico. ОС Raspberry Pi використовує модифіковане середовище робочого столу LXDE з диспетчером вікон стека Openbox разом із унікальною темою. Дистрибутив за замовчуванням постачається з копією системи комп'ютерної алгебри Wolfram Mathematica, VLC і полегшеною версією веб-браузера Chromium [22].  
Особливості:

- інтерфейс користувача. ОС Raspberry Pi має середовище робочого столу PIXEL на основі LXDE, яке виглядає схожим на багато поширених робочих столів, таких як macOS і Microsoft Windows. Робочий стіл має фонове зображення. Панель меню розташована вгорі та містить меню програми та ярлики веб-браузера (Chromium), файлового менеджера та терміналу. На іншому кінці панелі меню показано меню Bluetooth, меню Wi-Fi, регулятор гучності та годинник. Зовнішній вигляд робочого столу також можна змінити за замовчуванням, наприклад змінити положення панелі меню;

- управління пакетами. Пакети можна інстальювати через програму рекомендованого програмного забезпечення, а також за допомогою інструмента «Додати/видалити програмне забезпечення», оболонки графічного інтерфейсу.

Компоненти:

- PCManFM – це файловий браузер, який забезпечує швидкий доступ до всіх областей комп'ютера, і був перероблений у першому випуску Raspberry Pi OS Buster (2019-06-20);

- ОС Raspberry Pi спочатку поширювала веб-браузер Eripanu, але перейшла на Chromium із запуском оновленого робочого столу. Вбудований браузер поставляється з попередньо встановленими uBlock Origin і h264ify;

- ОС Raspberry Pi поставляється з багатьма IDE для початківців, такими як Thonny Python IDE, Mu Editor і Greenfoot. Він також поставляється з освітнім програмним забезпеченням, таким як Scratch і Bookshelf [23].

## 1.4 Основи рулевого управління

У цьому розділі розглядаються основи рулевого управління та концепція миттєвого центру обертання (IRC).

На рисунку 1.7 показано поточне положення чотириколісного робота.

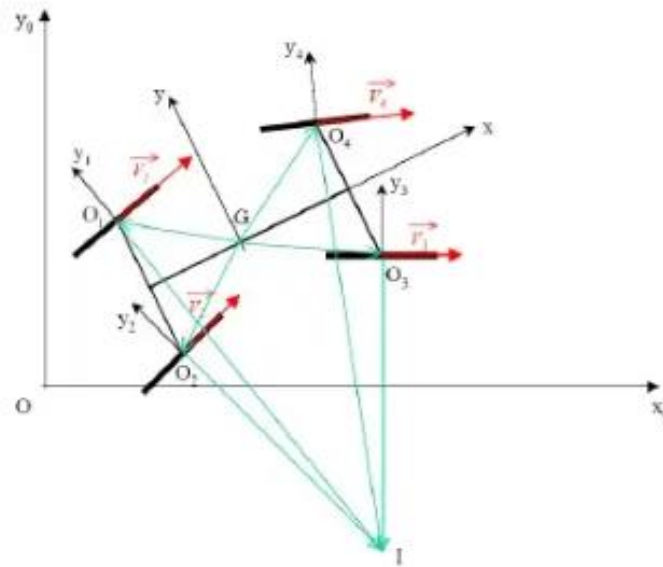


Рисунок 1.7 – IRC концепція

Осі коліс повинні перетинатися в точці, якщо немає ковзання. Ця точка, I, є центром миттєвого обертання (IRC) для руху транспортного засоба відносно поверхні.

### 1.4.1 Керування триколісним роботом з рулевим управлінням

Триколісний робот з рулевим управлінням представлений на рис. 1.8.

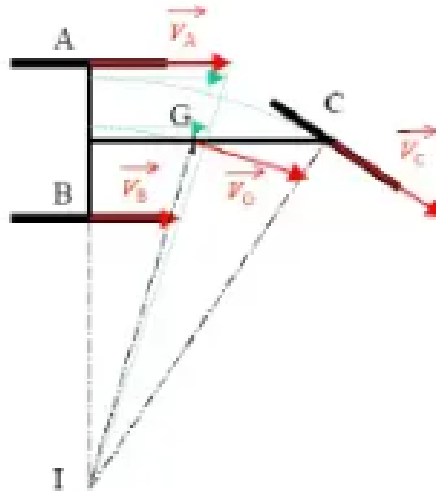


Рисунок 1.8 – Триколісний робот з рулевим управлінням

Припускаємо, що ковзання немає. Швидкість кожного колеса перпендикулярна до його осі обертання. У цьому випадку колеса А і В мають однакову вісь обертання. IRC розташований на перетині осей коліс А (або В) і С. Зауважте  $V_A \neq V_B \neq V_C$ , тому три колеса повинні бути незалежними, щоб обертатися з різними швидкостями. Це означає, що якщо триколісний робот задньопривідний, то між задніми колесами потрібен диференціал. Якщо є три ведучих колеса, потрібен другий диференціал між переднім і заднім колесами.

### 1.4.2 Рулеве керування робота з причепом

У цьому випадку колеса А – В і С – D мають однакову вісь обертання (рис. 1.9). IRC розташований на перетині осей коліс А (або В) і С (або D). Примітка:  $V_A \neq V_B \neq V_C \neq V_D$ , тому чотири колеса повинні бути незалежними, щоб мати можливість обертатися з різними швидкостями. Це означає, що якщо

транспортний засіб із заднім приводом, потрібен диференціал між задніми колесами. Якщо це повнопривідний автомобіль, потрібні ще два диференціали. Рульове керування роботом з причепом на рис. 1.9. Геометричний аналіз наведений на рис. 1.10.

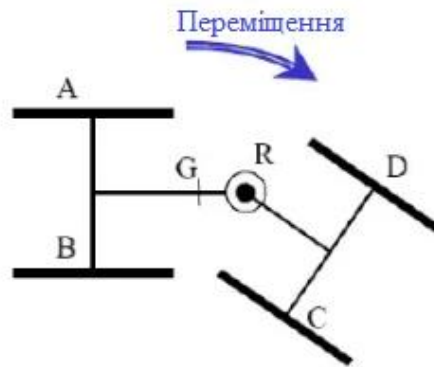


Рисунок 1.9 – Рульове керування роботом з причепом

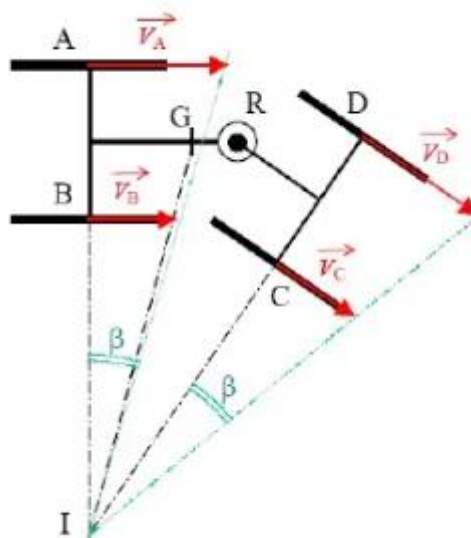


Рисунок 1.10 – Геометричний аналіз

### 1.4.3 Чотириколісний робот з двома рульові колесами

Якщо рульові колеса залишаються паралельними під час керування, як показано на рисунку 1.11, немає жодної точки перетину. Це несумісно з гіпотезою відсутності ковзання. У цьому випадку принаймні один вектор швидкості повинен мати інший напрямок. Це означає, що принаймні одне колесо пробуксовує, щоб транспортний засіб міг рухатися. Рішенням є використання різного кута повороту для кожного керма. Кут між поворотними колесами називається кутом Аккермана. Кінематичне дослідження паралельного рульового керування наведено на рисунку 1.12.

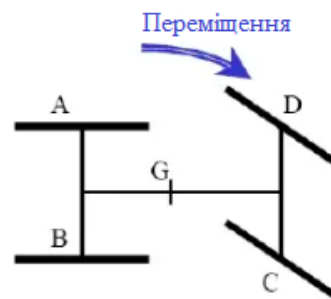


Рисунок 1.11 – Чотириколісний робот з двома рульові колесами

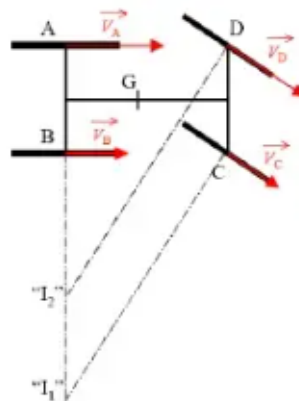


Рисунок 1.12 – Кінематичне дослідження паралельного рульового керування

#### 1.4.4 Непаралельні рульові колеса

При правильному диференціальному куті повороту IRC існує і його можна знайти так само, як і раніше. IRC розташований на перетині осей коліс A (або B) і C (або D) (рис. 1.13). Через ідеальне зчеплення диференціальний кут повороту має бути таким, щоб осі трьох коліс перетиналися в одній точці (IRC). Примітка: 4 колеса повинні бути незалежними, щоб мати можливість обертатися з різними швидкостями. Це означає, що якщо транспортний засіб має задній привід, потрібен диференціал між задніми колесами. Якщо він має повний привід, потрібні ще два диференціали.

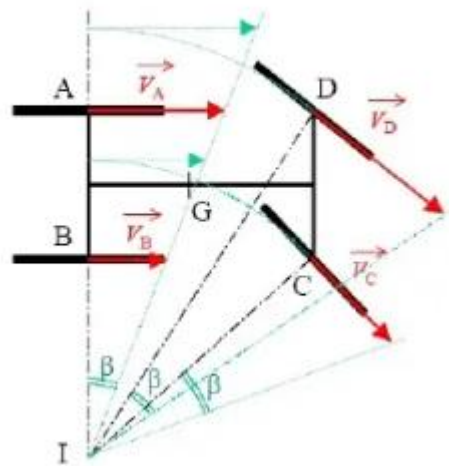


Рисунок 1.13 – Непаралельне рульове управління

#### 1.4.5 Чотириколісний робот з чотирма рулевими колесами

Схему чотириколісного робота з чотирма рулевими колесами наведено на рис. 1.14.

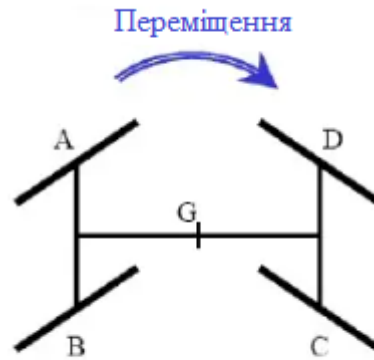


Рисунок 1.14 – Чотириколісний робот з чотирма рулевими колесами

#### 1.4.6 Паралельне рульове управління

Якщо рульові колеса залишаються паралельними під час кермування (рис. 1.15), єдиної точки перетину немає. Це несумісне з гіпотезою відсутності ковзання. У цьому випадку принаймні два вектори швидкості повинні мати різний напрямок. Це означає, що принаймні два колеса пробуксовують, щоб транспортний засіб міг рухатися. Можна використовувати рішення, що використовує різний кут повороту для кожного колеса. Кут між рульовими колесами називається кутом Акермана.

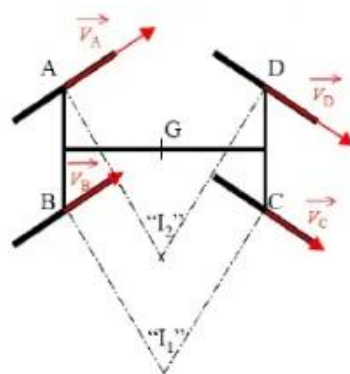


Рисунок 1.15 – Паралельне рульове управління

### 1.4.7 Непаралельне рульове управління

При правильних кутах повороту IRC існує і його можна знайти. У порівнянні з попереднім випадком, такий тип керування (рис. 1.16) дозволяє розташувати IRC у будь-якій точці літака. Знову ж таки, диференціальний кут повороту має бути правильним. Довільне розташування IRC для непаралельного керування наведено на рисунку 1.17.

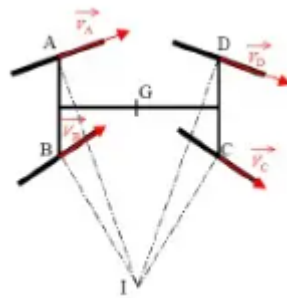


Рисунок 1.16 – Чотириколісний транспортний засіб з непаралельним рульовим управлінням

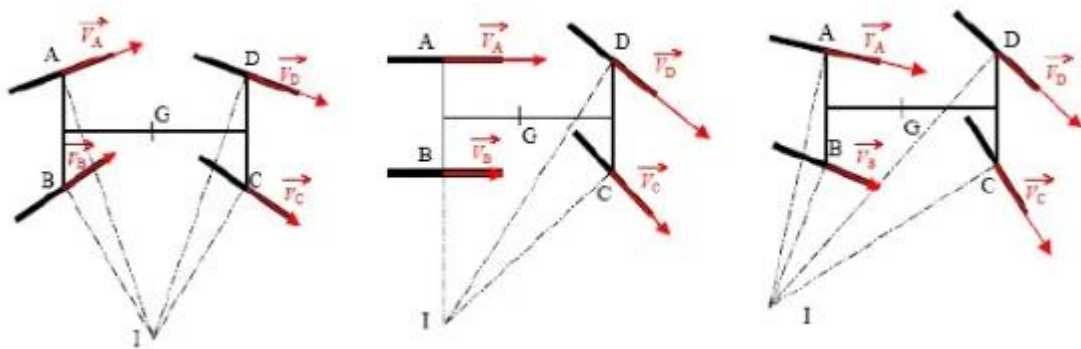


Рисунок 1.17 – Довільне розташування IRC для непаралельного керування

Є деякі особливі випадки, які заслуговують на окрему увагу (рис. 1.18).

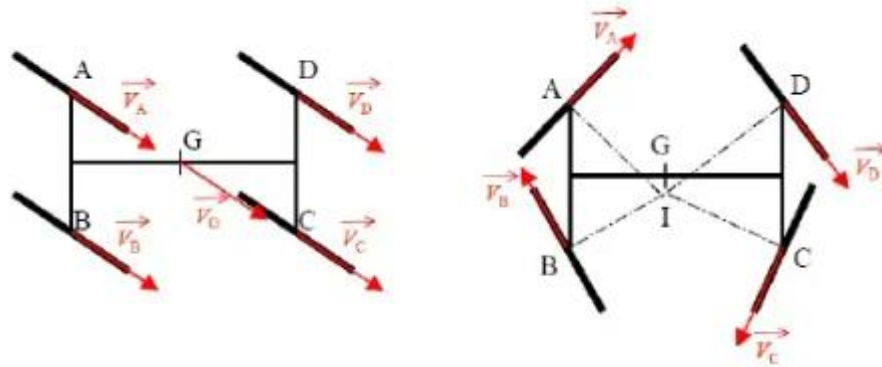


Рисунок 1.18 – Окремі випадки непаралельного керування

### 1.5 Системи рульового керування

Незалежне (явне) рульове керування: незалежне рульове керування явно налаштовує кожне з коліс на бажаний курс. Окрім проблем складності приведення в дію та точності координаційного керування, ця схема забезпечує переваги щодо маневреності мобільних роботів, особливо тих, що працюють на непідготовленій місцевості. Поширеною різновидом незалежного керування всіма колесами, недоступним іншими схемами, є крабове керування, при якому всі колеса повертаються на однакову величину в одному напрямку. В результаті робот може рухатися вбік. Координація водіння та рульового керування дозволяє ефективно маневрувати та зменшує вплив внутрішніх втрат через бій приводів [24].

Рульове керування Акермана: найпоширенішим типом рульового керування на легкових автомобілях є рульове управління Акермана, яке механічно координує кут передніх двох коліс. Щоб підтримувати всі колеса в стані чистого кочення під час повороту, колеса повинні слідувати вигнутим траєкторіям із різними радіусами, що виходять із спільного центру. Переваги чіткого рульового управління включають більш агресивне рульове управління з кращим розрахунком на місці (завдяки меншому пробуксовці коліс) і менше

споживання електроенергії. Недоліком явного рульового керування є більша кількість приводів, кількість частин і необхідний об'єм обертання. Шарнірне рульове управління поширене у великому землерийному обладнанні. Курс автомобіля змінюється шляхом складання шарнірних вузлів шасі. Для великих транспортних засобів рульове керування з шарнірною рамою має перевагу, оскільки дозволяє транспортному засобу бути значно маневренішим, ніж транспортний засіб із скоординованим рульовим керуванням. Рульове керування з шарнірною рамою має перевагу перед поворотним керуванням у тому, що під час повороту зберігається максимальна тяга, яку забезпечують тягові елементи [25].

Шарнірне керування: шарнірне керування може бути компактним, легким, вимагати невеликої кількості деталей і виявляти маневреність від повороту до прямолінійного руху, використовуючи лише рухи, компоненти та об'єм стріловидності, необхідні для прямолінійного руху. Поворотне керування досягається шляхом створення диференціальної тяги між лівою та правою сторонами транспортного засобу, що спричиняє зміну курсу. Це ефективне та просте рішення для керування роботом. Однак він не такий точний, як інші методи керування; деякі характеристики, включаючи тертя, пробуксовку коліс та інші непередбачувані атрибути, можуть спричинити проблеми. Ця конфігурація рульового управління є особливим випадком, коли бісектриси коліс не перетинаються, і ковзання коліс використовується для обертання робота. Недоліком є те, що занос викликає непередбачувані потреби в потужності через нерівності рельєфу та нелінійну взаємодію шини з ґрунтом. За допомогою міні-кермування також не вдається досягти максимально агресивного кермування, якого можна досягти за допомогою явного кермування, оскільки максимальна тяга вперед не підтримується під час повороту [26].

На рис. 1.19 – кінематика основних видів рульового керування.

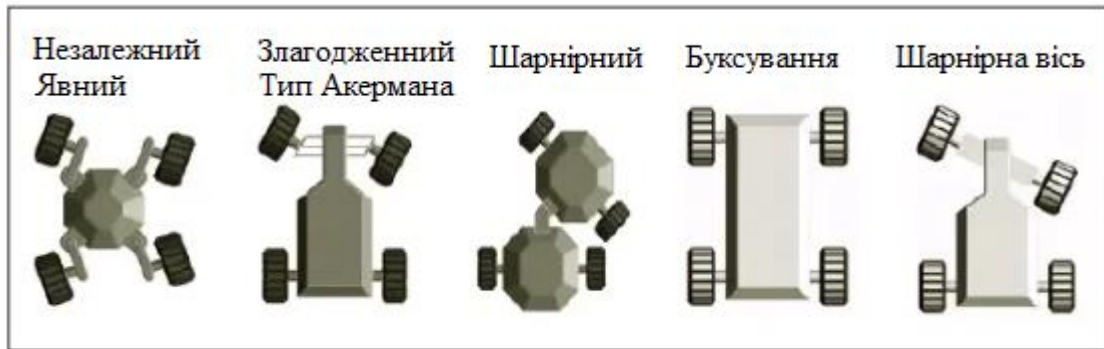


Рисунок 1.19 – Кінематика основних видів рульового керування

Пасивно шарнірна вісь: цей тип керування виконується шляхом додавання вільного шарніру до однієї з осей автомобіля. Його зазвичай можна знайти у вагонах або візках. Одним із недоліків одновісного керування є те, що під час поворотів колеса рухаються в окремих коліях. За складних ґрунтових умов це вимагає посиленого приводу, оскільки кожне колесо рухається по свіжій місцевості. Переваги включають механічну простоту, відносно низьку потужність рульового керування та помірну маневреність [27].

## 1.6 Висновки до першого розділу

У першому розділі було проведено аналіз предметної області та технічного завдання, під час якого розглянуто такі теми:

- аналіз використання мобільних роботів у сучасних виробничих системах;
- аналіз аналогічних конструкцій мобільних роботів;
- аналіз програмного забезпечення мобільних роботів;
- основи рулевого управління;
- системи рульового керування.

## 2 РОЗРОБКА МЕТОДУ УПРАВЛІННЯ РУЛЬОВИМИ КОЛЕСАМИ

### 2.1 Опис конструкції незалежних поворотних колісних модулів

#### 2.1.1 Аналіз аналогічних рішень

Аналіз аналогічних рішень показав, що конструкція поворотних коліс має схожі конструктивні рішення. Відмінність полягає в габаритах (рис. 2.1) [28] та застосуванні амортизаторів, як, наприклад показано в [29] (рис. 2.2).



Рисунок 2.1 – Агро-робот SMASH

Приклад реалізації поворотних коліс в агро-роботах SMASH від фірми Yanmar показано на рисунку 2.2.



Рисунок 2.2 – Приклад реалізації поворотних коліс в агро-роботах SMASH

В даній конструкції не реалізовано засобів амортизації руху, та використовуються потужні двигуни для руху платформи, а також виконання функція повороту. Це пов'язано з областю використання пристрою – агропромисловий сектор, де потрібно вирішувати завдання перевезення значних вантажів, та переміщуватись пересічною місцевістю.

Для складських приміщень та для цехового застосування підходять роботи класу FRC Team 1640 [30].

Реалізація колісного блоку таких роботів відрізняється меншим діаметром коліс, менш потужними двигунами та наявністю системи амортизації для забезпечення плавності переміщення вантажів.

Існує декілька варіантів конструкції поворотного механізму. Перша версія конструкції механізму показана на рисунку 2.3.



Рисунок 2.3 – Конструкція поворотного механізму

В даній версії для повороту засовується двигун постійного струму та енкодер для визначення кута повороту колеса. Система ременів з передає зусилля від тягового двигуна на вісь, що є основою кріплення мотора та колеса, та на окремий модуль з енкодером.

На рисунку 2.4 показано 3D-модель поворотного вузлу, з якої можна бачити принцип кріплення всього механізму то мобільної платформи. Для руху колеса застосовується окремий двигун. Оберти з цього двигуна передаються на колесо через ланцюгову передачу (рис. 2.4).

Така конструкція забезпечує дуже легку заміну двигуна та дозволяє застосовувати різні типи двигунів для виконання руху – крокові, або безколекторні, постійного струму.

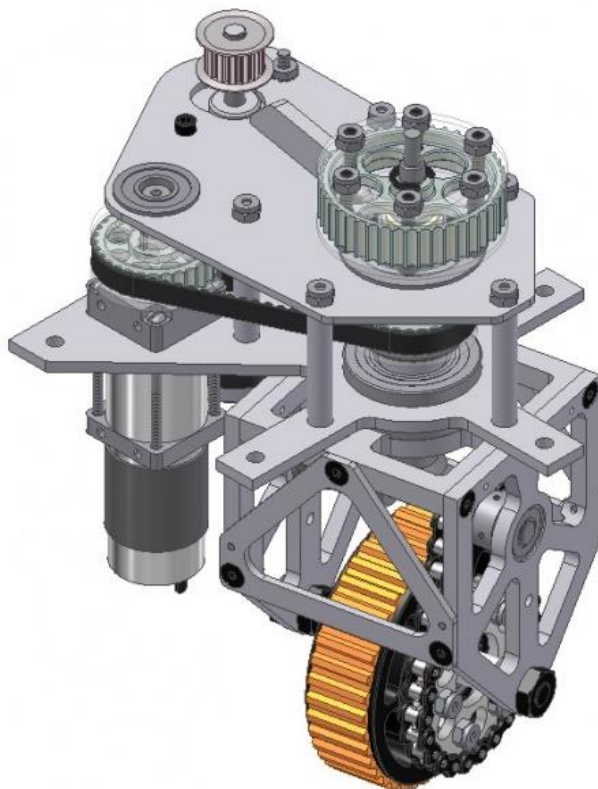


Рисунок 2.4 – 3D-модель поворотного вузлу FRC Team 1640

Розробники досягли цього, зробивши модуль, що включає шарнірний механізм, приводний двигун CIM, двигун рульового управління та коробку передач, датчик кута та трансмісію, що з'єднує ці компоненти. Шарнірний модуль можна замінити, видаливши болти та електричні з'єднання[30].

Конструкція лівого і правого модулів відрізняється дзеркальним відображенням її компонентів. В роботі застосовуються два типи модулів. Усі ліві модулі ідентичні та взаємозамінні, як і всі праві модулі. Ідентичне також калібрування кута датчика.

Для рульового керування були обрані мотори Vanebots RS-540 з 4-ступінчастими планетарними редукторами Vanebots 256:1. Вал коробки передач приводить в рух 32-зубовий шків синхронного ременя HTD5, а також жорстко з'єднаний з датчиком кута повороту (кут валу двигуна рульового керування використовується як посередник для визначення кута повороту).

Поворотна труба зовнішнього діаметра має 32-зубчастий синхронний шків HTD5. Ремінь HTD5 360 мм шириною 15 мм з'єднує ведучий і ведений рульові шківви 32Т (1:1).

Для визначення кута оберту використовується датчик кута та калібрування. В модулі встановлений датчик Vishay (на оригінальних нейлонових опорах). Цей датчик є абсолютним енкодером Vishay 981HE0B4WA1F16. Датчики Vishay мають "D"-вали, які жорстко з'єднані з валами коробки передач рульового керування.

На даний час існує два варіанта механізму повороту коліс від компанії FRC Team 1640, що показані на рисунку 2.5.

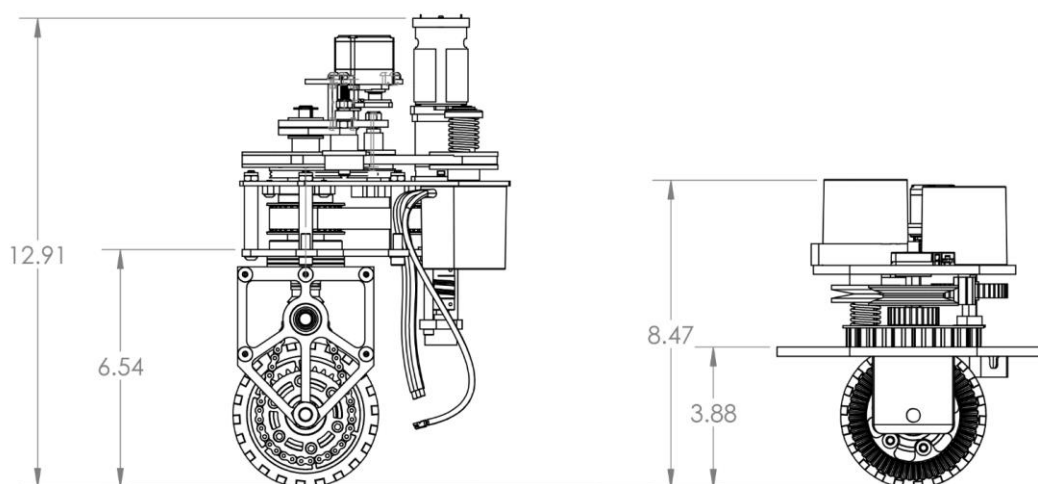


Рисунок 2.5 – Два варіанта механізму повороту коліс від компанії FRC Team 1640

Ключовими перевагами нового модуля є зменшення маси, низькі профілі та покращена надійність безступінчатої коробки передач (CVT). Це результат проекту розвитку команди, який проводився протягом останніх двох років.

Зменшена висота з'єднання шасі з 6,54 до 3,88 дюйма дозволяє створити простіший і легший каркас шасі. Електронну панель тепер можна під'єднати

безпосередньо до нижньої поверхні рами шасі. Зменшення загальної висоти модуля звільняє простір у кутах робота для механізмів мобільної платформи.

На рисунку 2.6 показана модель поворотного механізму, що виконано на сервоприводах.



Рисунок 2.6 – Модель поворотного механізму, що виконано на сервоприводах

Основна зміна полягає в досягненні зниження CVT. Використовується той самий шків CVT, що й у попередніх моделях, але замість нього використовується пара сервоприводних «нажимних шківів», які натягують клиновий ремінь CVT, щоб вплинути на зміну редуктора (розсуваючи половини шківа CVT), відстань між осями ремінної петлі CVT змінюється шляхом переміщення положення осі приводного двигуна/шківа CVT. Це зміщення положення керується сервоприводом. Механізм «прижимного шківа» був крихким і піддавався поломці.

### 2.1.2 Ескізне проектування модуля поворотних коліс

Виходячи з аналізу аналогічних рішень пропонується спрощена конструкція поворотного колісного модуля для вирішення задачі довільного управління рухом мобільної платформи.

На рисунку 2.7 представлено ескіз механізму руху та поворотів.

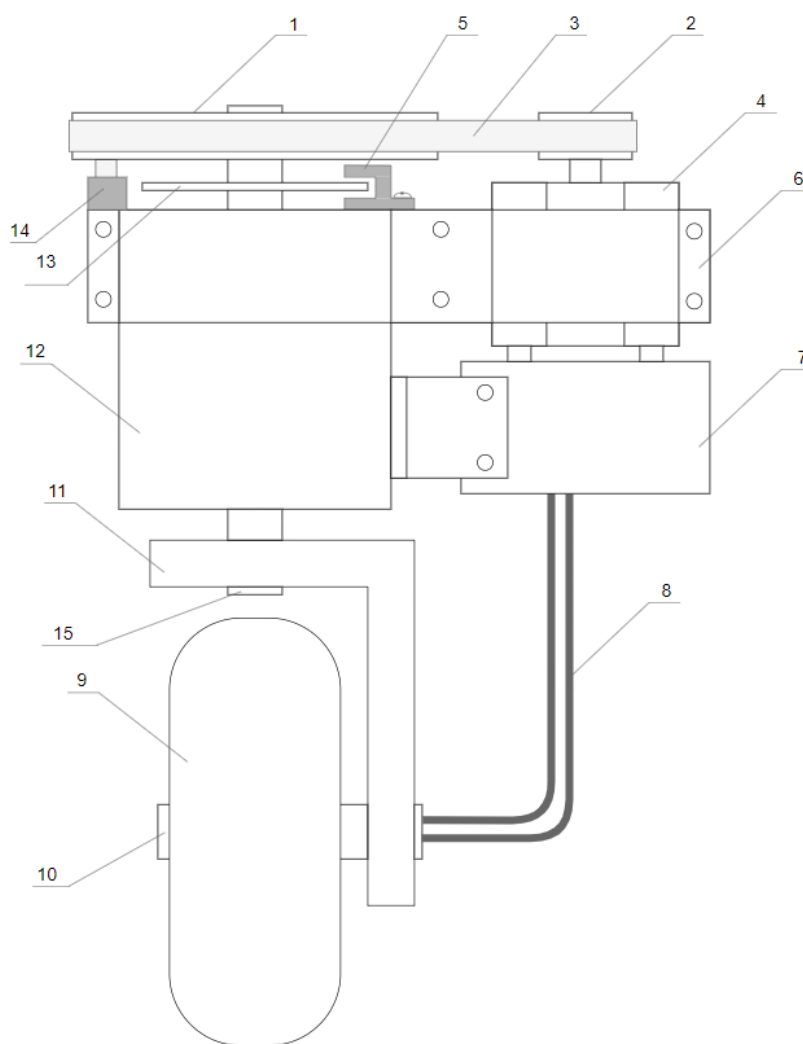


Рисунок 2.7 – Ескіз механізму руху та поворотів

Пропонована конструкція є спрощеним рішенням створення модуля повороту коліс та являє собою комбінацію розглянутих конструктивних рішень.

Основа 12 містить в собі опорні підшипники та є кріпленням всієї конструкції до шасі мобільної платформи. За допомогою кронштейну 6 до основи кріпиться кроковий двигун 4.

Кроковий двигун 4 управляє кутом повороту колеса 9 відносно вертикальної осі 15. Колесо кріпиться на «Г» образний тримач за допомогою втулки 10. В даній конструкції використовується мотор-колесо, таким чином, додаткового двигуна для приводу коліс не використовується.

Управління поворотом колеса відбувається кроковим двигуном за допомогою ремня 3, що передає оберти з ведучого шківу 2 до веденого 1. Завдяки різному діаметру організовано понижуюча передача 1:4.

Контролер 7, що керує роботою даного колісного модуля кріпиться до основи 12 за допомогою кутового кронштейну.

Контроль кута обертв відбувається за кількістю кроків, що робить двигун від його кінцевого положення до потрібного значення. Додатково в конструкції передбачено наявність енкодеру 13, що кріпиться на вісь 15. Дані з енкодеру зчитуються за допомогою оптичного датчику 5. Кінцевий датчик 15 призначено для виявлення кінцевого положення центральної осі при старті роботи модулю повороту коліс.

## 2.2 Розробка алгоритму управління поворотним колісним модулем

Схема алгоритму управління поворотним колісним модулем показана на рисунку 2.8.

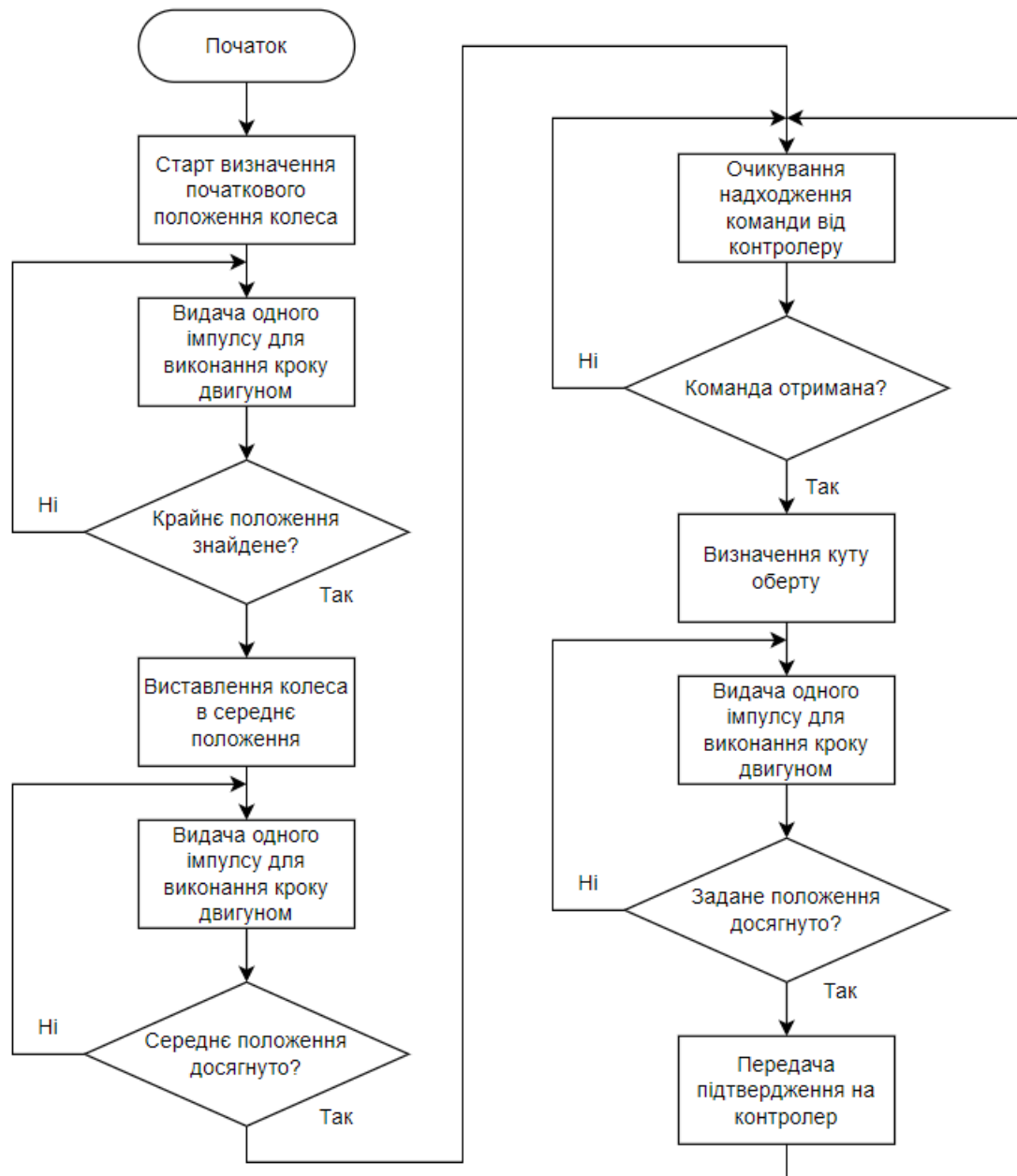


Рисунок 2.8 – Алгоритм управління поворотним колісним модулем

На початку роботи відбувається виставлення колеса в початкове положення. Початковим вважається таке положення, коли спрацював кінцевий датчик, що встановлено на базовій основі. Від цього положення відлічуються всі інші положення колеса.

Після знаходження початкового положення, колесо переводиться в середнє положення. Середнім вважається таке положення, при якому колесо орієнтовано строго паралельно напрямку руху мобільної платформи.

Коли модуль завершує етап калібрування, модуль управління його роботою переходить в режим очікування команд від головного контролера.

При надходженні команди від контролера відбувається її обробка та визначається заданий кут повороту колеса.

Після визначення заданого кута запускається процес обертів повороту колеса в задане положення. Це положення визначається за кількістю імпульсів до крокового двигуна від модуля управління, а також контролюється за даними від енкодера.

Після досягнення заданого положення кроковий двигун зупиняється та утримує колесо в заданому положенні, а модуль керування передає на головний контролер пакет даних для підтвердження операції виконання орієнтації колеса.

Після цього модуль управління переходить в режим очікування нової команди від головного контролера.

### 2.3 Розробка алгоритму синхронного управління комплексом поворотних колісних модулів

Схема алгоритму синхронного управління комплексом поворотних колісних модулів показана на рисунку 2.9.

На початку роботи виконується ініціалізація початкового положення коліс. Для цього видається відповідна команда, що розпізнається кожним окремим модулем повороту. Кожен модуль незалежно один від одного виконують команду ініціалізації та повідомляють про це головний модуль, на його запит.

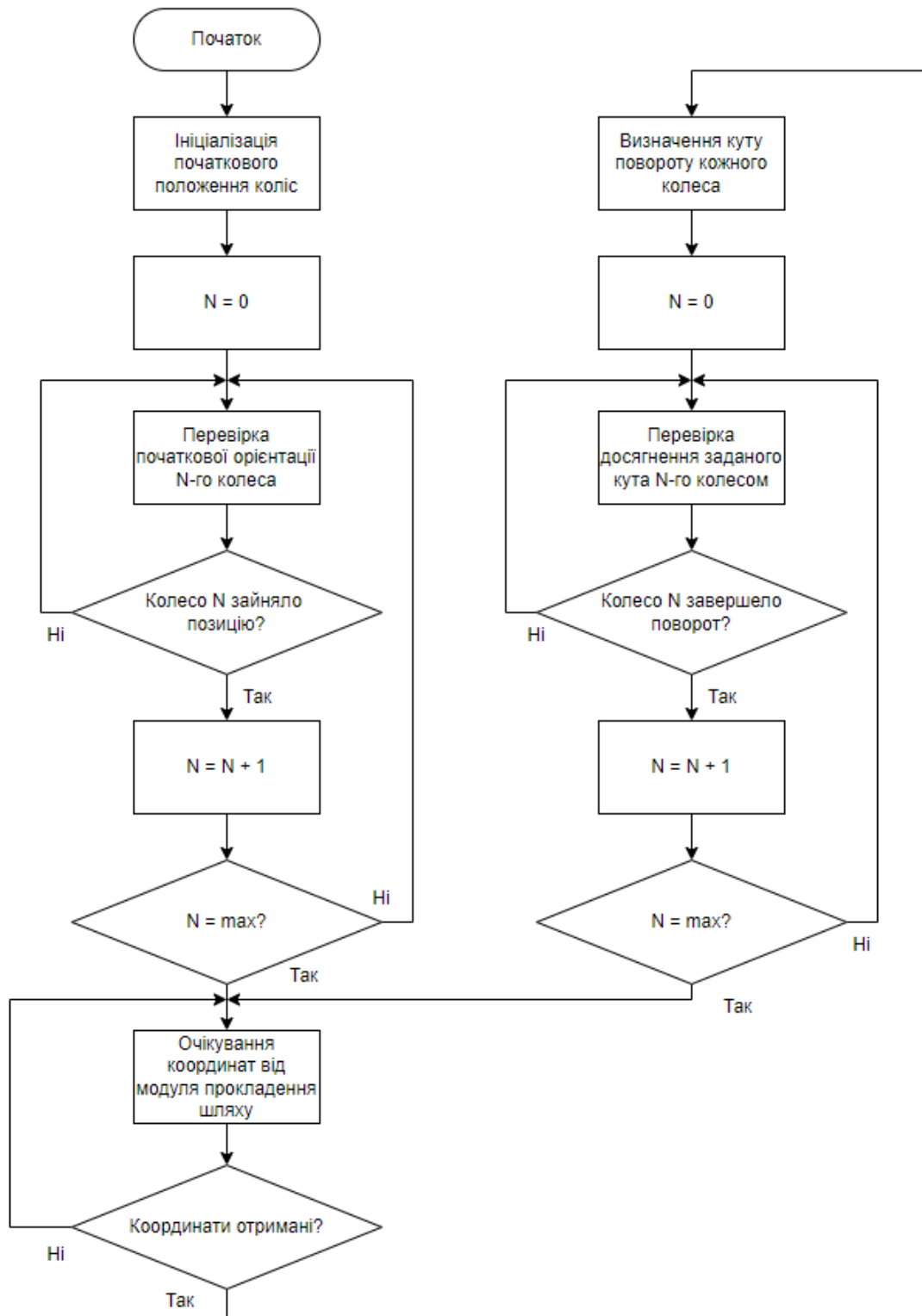


Рисунок 2.9 – Схема алгоритму синхронного управління комплексом поворотних колісних модулів

Головний модуль послідовно опитує кожен модуль та перевіряє досягнення їм початкового положення.

Тільки після заняття заданого положення програма переходить до етапу виконання безпосередньо управління рухом мобільної платформи.

Визначенням певного кута повороту кожного колеса відбувається на основі даних від модуля прокладення шляху. Зазвичай такий модуль виконується на базі міні-комп'ютеру Raspberry PI.

На основі отриманої цільової координати та поточної точки в просторі за методом, що запропоновано в даній роботі визначається кут повороту кожного колеса.

Перед початком руху, головний контролер надсилає команду кожному модулю керування відповідні дані та контролює завершення процесу заняття колесами заданої орієнтації.

Після завершення операції повороту коліс модуль переходить в режим очікування нових координат.

## 2.4 Висновки до другого розділу

У другому розділі було проведено розробку методу управління рульовими колесами, в цьому розділі розглянуто такі теми:

- аналіз аналогічних рішень;
- ескізне проектування модуля поворотних коліс;
- розробка алгоритму управління поворотним колісним модулем;
- розробка алгоритму синхронного управління комплексом поворотних колісних модулів.

### 3 ОПИС РОЗРОБЛЕНОГО МЕТОДУ УПРАВЛІННЯ РУХОМ МОБІЛЬНОЇ ПЛАТФОРМИ З ЧОТИРМА РУЛЬОВИМИ КОЛЕСАМИ

В процесі руху мобільної платформи постають задачі зміни прямолінійного напрямку руху. Для цього використовуються різні принципи побудови рульових механізмів. В нашій роботі розглядається принцип керування із чотирма незалежними поворотними колесами.

Задача, що вирішується, має наступні початкові вимоги. Для платформи, що знаходиться в точці  $M(x_0, y_0)$ , та на поточний момент має вектор швидкості  $v$  (рис. 3.1), потрібно визначити такий кут повороти для кожного колеса, щоб вона потрапила в точку  $T(x_1, y_1)$ .

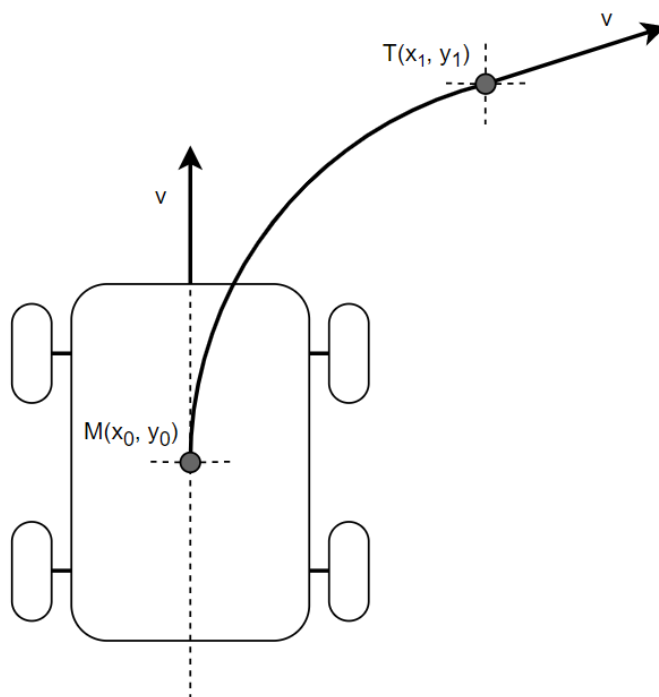


Рисунок 3.1 – Траєкторія руху мобільної платформи

Для рішення поставленої задачі запропоновано наступний метод.

На карті місцевості визначаються координати розташування мобільної платформи та бажаної точки, в яку необхідно перемістити цю платформу.

Якщо цільова точка не знаходиться на вісі вектору швидкості  $v$ , то необхідно знайти таке положення кожного рульового колеса, щоб забезпечити плавний поворот платформи в процесі руху до цієї точки.

На наступному етапі необхідно знайти координати  $C(x, y)$  центра кола на якому лежить траєкторія руху платформи (рис. 3.2).

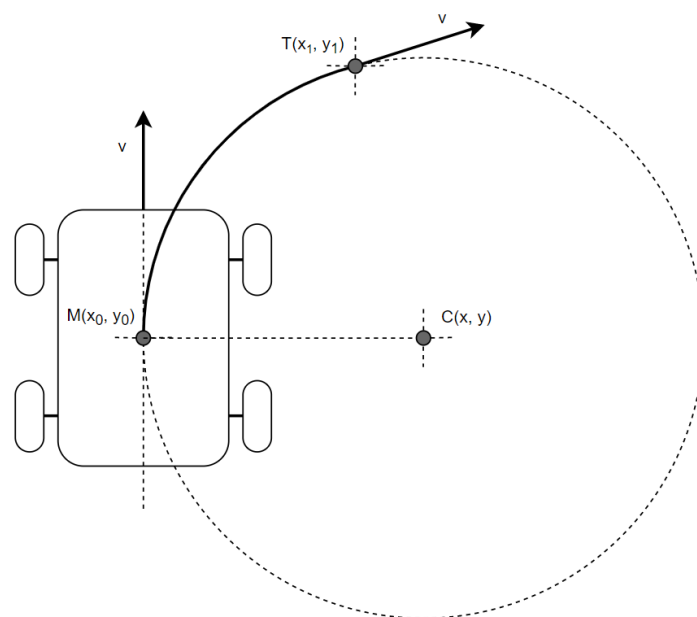


Рисунок 3.2 – Координати  $C(x, y)$  центра кола на якому лежить траєкторія руху платформи

При визначенні центра кола приймається до уваги, що його центр завжди знаходиться на вісі, що виходить з геометричного центру платформи (рис. 3.2) та перпендикулярна до вектору швидкості  $v$ .

Для розрахунку значення радіусу  $R$  необхідно визначити кут нахилу траєкторії руху  $\alpha$  (відрізок  $MT$  на рисунку 3.3).

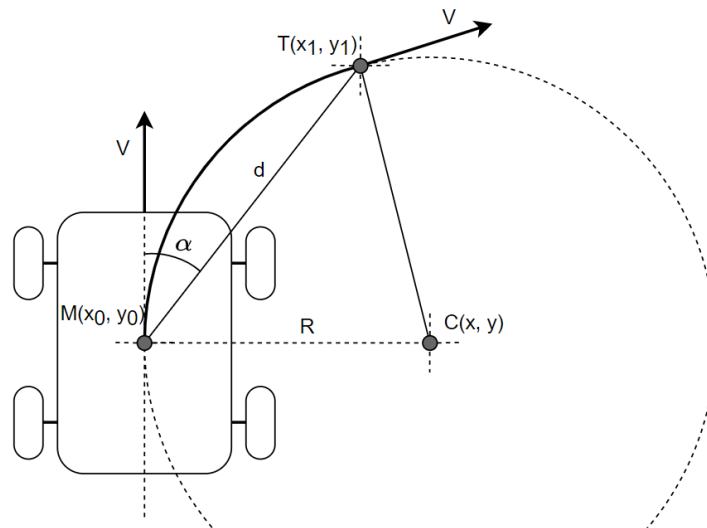


Рисунок 3.3 – Розрахунок радіусу R

За теоремою косинусів отримуємо:

$$d^2 = 2R^2(1 - \cos(2\alpha)), \quad (3.1)$$

де  $d$  – відстань між центром розташування мобільної платформи та цільовою точкою;

$R$  – радіус кола, на межі якого лежать точки  $M$  та  $T$ ;

$\alpha$  – кут між вектором швидкості  $v$  та прямою  $d$ .

Перепишемо вираз (3.1) таким чином:

$$d^2 = 2R^2(1 - \cos(2\alpha)) = 4R^2 \sin^2 \alpha. \quad (3.2)$$

Тепер з виразу (3.2) знайдемо  $R$ :

$$R = \frac{c}{(2 \sin \alpha)}. \quad (3.3)$$

Для того, щоб платформа потрапила в точку  $T$ , знаючи радіус кола, що є траєкторією її руху, необхідно визначити кут повороту кожного колеса.

На рисунку 3.4 показана кінематична модель рухомої платформи з чотирма колесами, що можуть змінювати кут повороту.

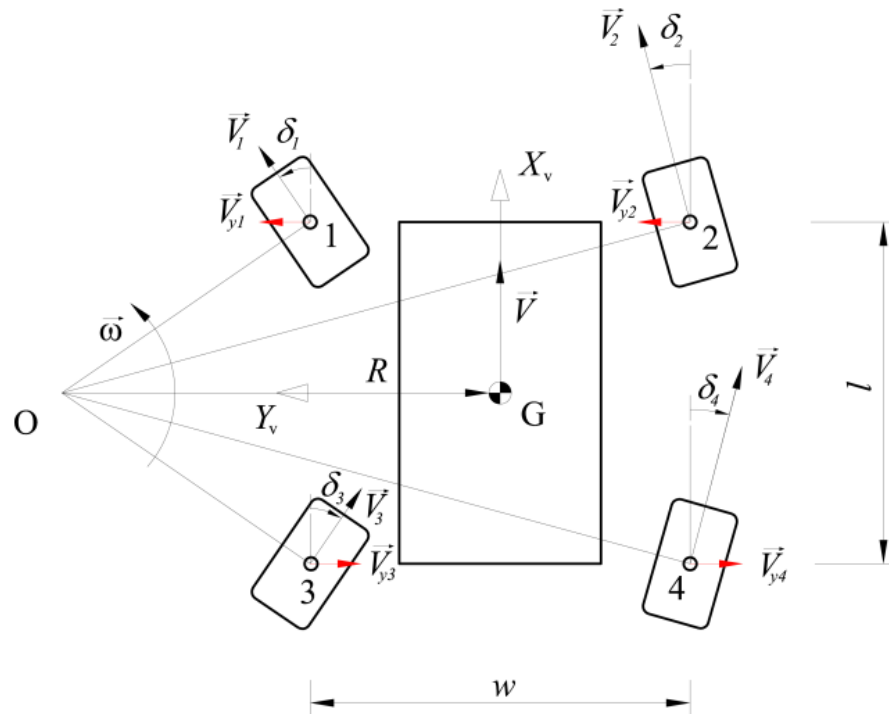


Рисунок 3.4 – Кінематична модель рухомої платформи з чотирма колесами, що можуть змінювати кут повороту

Виходячи із рисунка 3.4 розглянемо порядок дій для визначення куту повороту кожного колеса.

Для цього побудуємо геометричну модель, що описує поведінку колеса при його повороті.

Така модель для одного колеса показана на рисунку 3.5.

Основна задача в даному випадку знайти координати центру колеса  $W_1$  по відношенню до координати центру мобільної платформи  $M$ .

Коли буде відома ця координата, то можна буде провести відрізок  $W_1C$  від центру кола траєкторії руху платформи.

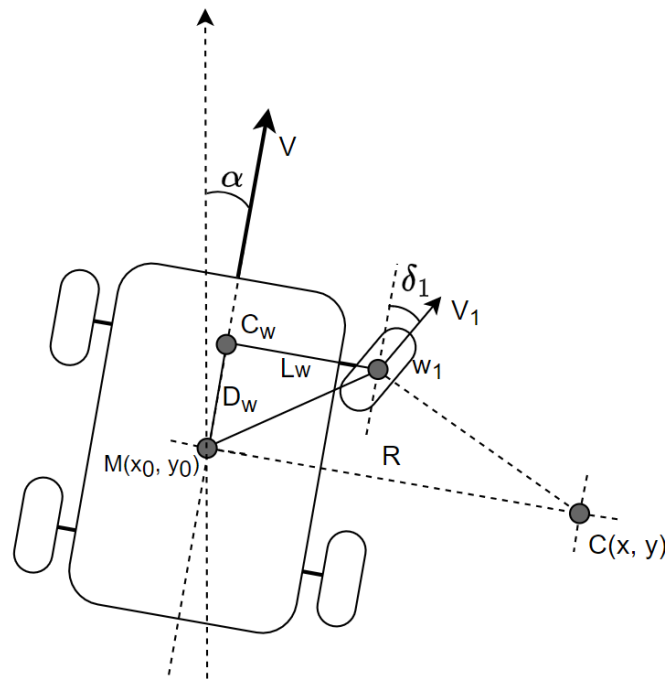


Рисунок 3.5 – Геометрична модель, що описує поведінку колеса

Напрямок повороту колеса буде нормаль до відрізка  $W_1C$ .

В загальному випадку платформа буде знаходитись в довільному положенні по відношенню до центральної вісі, що є базовою для карти місцевості та паралельна однієї з осей системи координат  $x$  або  $y$ . В такому випадку необхідно враховувати кут повороту всієї платформи  $\alpha$  (рис. 3.5).

На рисунку 3.5  $D_w$  – відстань від центру платформи  $M$  до лінії, що поєднує центри передніх коліс (рис. 3.5).  $L_w$  – відстані між центральною віссю  $C_w$  платформи та центром колеса  $W_1$ .

Координата  $C_x$  центру відрізка, від колеса  $W_1$  до центральної осі розраховується за наступною формулою:

$$C_x = \cos(\alpha)D_w, \quad (3.4)$$

де  $\alpha$  – кут повороту мобільної платформи відносно базової осі;

$D_w$  – відстань від центру платформи М до лінії, що поєднує центри передніх коліс.

Координата  $C_y$  розраховується аналогічно, але замість косинуса використовуємо синус:

$$C_y = \sin(\alpha)D_w. \quad (3.5)$$

Остаточно знайти координату центру колеса  $W_{1,x}$  можна за наступною формулою:

$$W_{1,x} = C_x - \cos(\alpha - 90)L_w, \quad (3.6)$$

де  $L_w$  – відстані між центральною віссю  $C_w$  платформи та центром колеса  $W_1$ .

Координату центру колеса  $W_{1,y}$  знаходимо за наступною формулою:

$$W_{1,y} = C_y - \sin(\alpha - 90)L_w. \quad (3.7)$$

Кут повороту колеса знаходиться як нормаль до радіусу, що йде від центру колеса  $W_{1,(x,y)}$  до центру кола  $C_{(x,y)}$ .

Для знаходження нормалі нам потрібно знати діаметр колеса. На рисунку 3.6 показано схематично основні параметри мобільної платформи для розрахунку куту повороту колеса.

Координати точки  $D_{wa}$  розраховуються за наступними формулами:

$$D_{wa} = W_{1,x} - D_w \frac{(C_y - W_{1,y})}{\sqrt{(C_x - W_{1,x})^2 + (C_y - W_{1,y})^2}}, \quad (3.8)$$

$$D_{wb} = W_{1,y} - D_w \frac{(C_x - W_{1,x})}{\sqrt{(C_x - W_{1,x})^2 + (C_y - W_{1,y})^2}}, \quad (3.9)$$

де  $D_w$  – діаметр колеса;

$W_{1,x}$  – x координата центру колеса;

$W_{1,y}$  – y координата центру колеса;

$C_x$  – x координата центру кола траєкторії;

$C_y$  – y координата центру кола траєкторії.

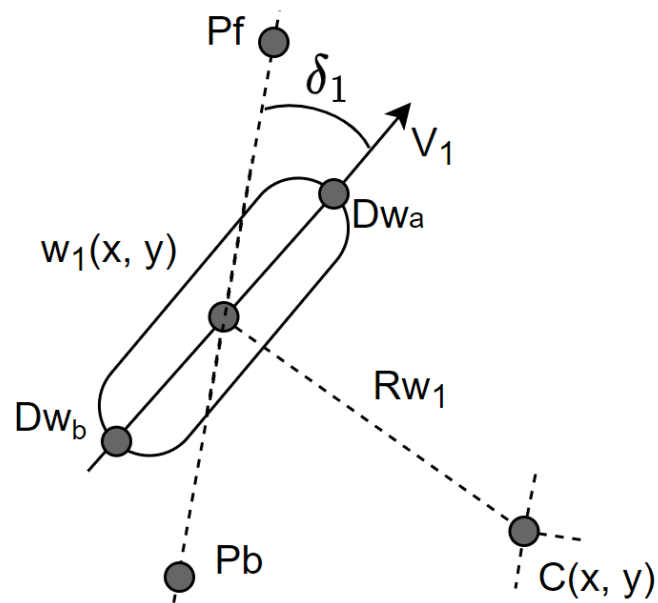


Рисунок 3.6 – Розрахунок куту повороту колеса

Кут повороту колеса  $\delta_1$  розраховується як кут між центральною віссю мобільної платформи, яка представлена точками  $P_f$  та  $P_b$  та вектором  $v_1$ , що поєднує точки  $D_{wa}$  та  $D_{wb}$ .

$$\theta_1 = \tan^{-1}(D_{wa.y} - D_{wb.y}, D_{wa.x} - D_{wb.x}), \quad (3.10)$$

$$\theta_2 = \tan^{-1}(P_{f.y} - P_{b.y}, P_{f.x} - P_{b.x}), \quad (3.11)$$

де  $D_{wa}$ ,  $D_{wb}$  – точки що визначають діаметр колеса;

$P_{f.y}$  та  $P_{f.x}$  – точки що визначають довжину платформи вздовж головної вісі.

Модуль різниці між значеннями  $\theta_1$  та  $\theta_2$  обчислюється за наступною формулою:

$$d_\theta = |\theta_1 - \theta_2|. \quad (3.12)$$

Кут повороту колеса  $\delta$  розраховується за наступною формулою:

$$\delta_1 = \min \{d_0, |180 - d_0|\}. \quad (3.13)$$

### 3.1 Висновки до третього розділу

В третьому розділі було проведено опис розробленого методу управління рухом мобільної платформи з чотирма рульовими колесами, і він присвячений кінематичним формулам, математичним формулам та розрахункам, необхідним для продовження подальшої роботи.

Наведені такі формули, як: формула розрахунку кута повороту коліс, формула знаходження центру колеса, формули знаходження координат  $C_w$  та  $C_y$ , та інші.

## 4 ЕКСПЕРИМЕНТАЛЬНІ ДОСЛІДЖЕННЯ

### 4.1 Постановка задачі проведення експерименту

Для проведення експериментальних досліджень необхідно розробити програмний засіб, що дозволяє перевірити правильність запропонованих математичних виразів та методики розрахунку куту повороту ведучих коліс в залежності від координат цільової точки на карті місцевості, а також наочно змоделювати та візуалізувати траєкторію руху мобільної платформи для досягнення вказаної точки на карті місцевості.

Програмний засіб необхідно зробити інтерактивним з можливістю введення потрібної цільової точки та можливістю зміни орієнтації мобільної платформи в просторі.

Всі розрахункові параметри необхідно вивести на екран монітору для контролю та налізу отриманих даних.

### 4.2 Опис програмного засобу для виконання експериментальних досліджень

На рисунку 4.1 показано зовнішній вигляд основного вікна програмного засобу.

Як можна бачити з наведеного рисунку в програмі передбачено основна область де схематично відображається мобільна платформа та положення її коліс.

Положення коліс може змінюватись в залежності від положення цільової точки.

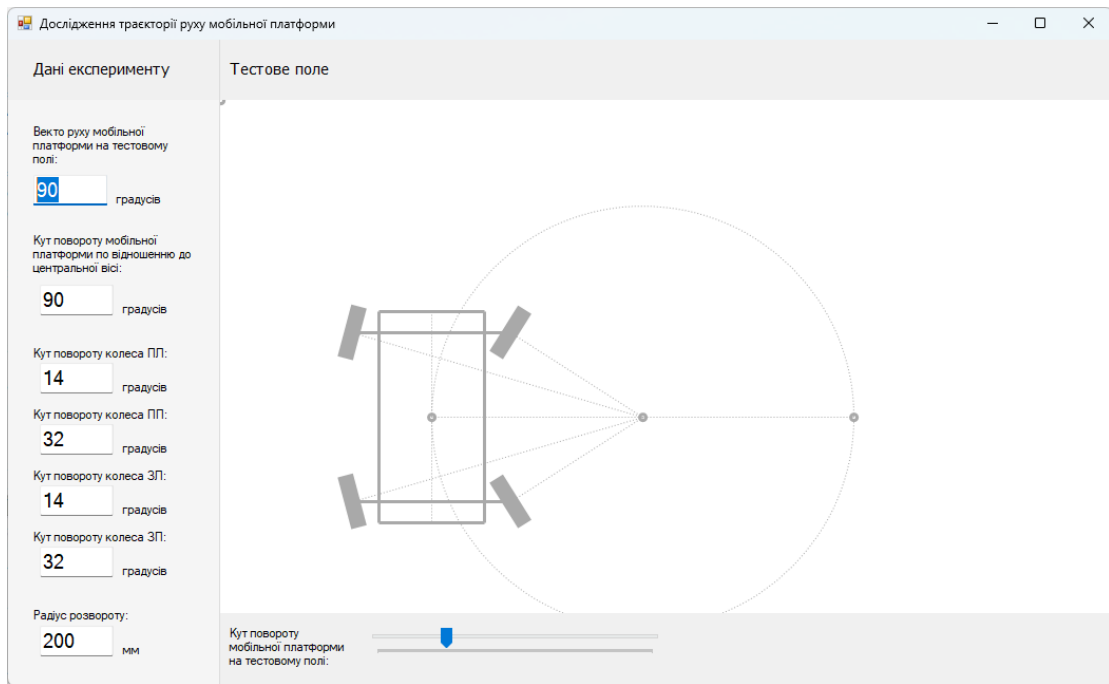


Рисунок 4.1 – Зовнішній вигляд основного вікна програмного засобу

Введення нового положення цільової точки відбувається за допомогою лівої кнопки миші.

В лівому боці програми відображаються розрахункові дані:

- вектор руху мобільної платформи на тестовому полі;
- кут між вектором руху платформи та центром кола розвороту;
- кут повороту переднього лівого колеса;
- кут повороту переднього правого колеса;
- кут повороту заднього лівого колеса;
- кут повороту заднього правого колеса;
- радіус розвороту.

За допомогою регулятора в нижній частині вікна можна регулювати кут повороту мобільної платформи на тестовому полі.

Для відстеження події натискання лівою кнопкою миші на тестовому полі створена відповідна функція, приклад якої показано на рисунку 4.2.

Ссылка: 1

```
private void myPanel1_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Right)
    {
        Target = new Point();
        Destination.X = e.X;
        Destination.Y = e.Y;

        destination_vector.p1 = Destination;
        destination_vector.p2 = Mobile;

        double theta1 = Math.Atan2(mobile_vector.p1.Y - mobile_vector.p2.Y,
            mobile_vector.p1.X - mobile_vector.p2.X);
        double theta2 = Math.Atan2(destination_vector.p1.Y - destination_vector.p2.Y,
            destination_vector.p1.X - destination_vector.p2.X);
        double diff = Math.Abs(theta1 - theta2);
        double angle = Math.Min(diff, Math.Abs(180 - diff));
        double dAngle = angle * 180 / Math.PI;
        angle_Destination = (int)dAngle;

        textBox5.Text = angle_Destination.ToString();

        myPanel1.Refresh();
    }
}
```

Рисунок 4.2 – Фрагмент функції обробки натискання лівою кнопкою миші на тестовому полі

В даній функції, по-перше, визначаються координати натискання  $x$  та  $y$ . Ці координати заносяться в змінну «Destination».

Наступним кроком формується орієнтований вектор «destination\_vector» до якого заносяться дві токи: координати центру мобільної платформи «Mobile» та нової координати «Destination».

Далі за формулами (3.10) – (3.13) визначаються значення  $\theta_1$  та  $\theta_2$  та модуль різниці між ними. Отримані дані виводяться в текстове поле «Кут між вектором руху платформи та центром кола розвороту»

Отримані попередні дані використовується для розрахунку куту повороту коліс.

Визначення куту між векторами зустрічається декілька разів в програмному кодї, тому набір основних операторів оформлено у вигляді функції getAngle, зміст якої показано на рисунку 4.3.

Ссылка: 2

```
private int getAngle(myLine L1, myLine L2)
{
    double theta1 = Math.Atan2(L1.p1.Y - L1.p2.Y, L1.p1.X - L1.p2.X);
    double theta2 = Math.Atan2(L2.p1.Y - L2.p2.Y, L2.p1.X - L2.p2.X);
    double diff = Math.Abs(theta1 - theta2);
    double angle = Math.Min(diff, Math.Abs(180 - diff));
    double dAngle = angle * 180 / Math.PI;
    angle_Destination = (int)dAngle;

    return angle_Destination;
}
```

Рисунок 4.3 – Зміст функції getAngle

Функція приймає два вектори L1 та L2, що містять по дві координати, які лежать на векторі, проводить розрахунок та повертає ціле значення куту.

На рисунку 4.4 наведена функція визначення куту повороту коліс.

Ссылка: 4

```
private void drawWhell(Point startPoint, Point endPoint, int whellDiameter, Graphics g, TextBox textBox)
{
    int x1 = (int)(endPoint.X - whellDiameter * ((startPoint.Y - endPoint.Y) /
    (Math.Sqrt(Math.Pow((startPoint.X - endPoint.X), 2) + Math.Pow((startPoint.Y - endPoint.Y), 2))));
    int y1 = (int)(endPoint.Y + whellDiameter * ((startPoint.X - endPoint.X) /
    (Math.Sqrt(Math.Pow((startPoint.X - endPoint.X), 2) + Math.Pow((startPoint.Y - endPoint.Y), 2))));
    Point normalPoint1 = new Point(x1, y1);

    int x2 = (int)(endPoint.X + whellDiameter * ((startPoint.Y - endPoint.Y) /
    (Math.Sqrt(Math.Pow((startPoint.X - endPoint.X), 2) + Math.Pow((startPoint.Y - endPoint.Y), 2))));
    int y2 = (int)(endPoint.Y - whellDiameter * ((startPoint.X - endPoint.X) /
    (Math.Sqrt(Math.Pow((startPoint.X - endPoint.X), 2) + Math.Pow((startPoint.Y - endPoint.Y), 2))));
    Point normalPoint2 = new Point(x2, y2);

    Pen whellPen = new Pen(Color.DarkGray, 15);
    g.DrawLine(whellPen, normalPoint1, normalPoint2);

    //get angle whell rotate
    myLine L1 = new myLine();
    L1.p1 = normalPoint1;
    L1.p2 = normalPoint2;

    myLine L2 = new myLine();
    L2.p1 = mobile_vector.p2;
    L2.p2 = mobile_vector.p1;

    int angelWhell = getAngle(L1, L2);
    textBox.Text = angelWhell.ToString();
}
```

Рисунок 4.4 – Функція визначення куту повороту коліс

Дана функція приймає в якості вхідних даних:

- розрахований центр кола траєкторії слідування до вказаної цільової точки (рис. 4.5);
- координати кріплення колеса (центр колеса);
- діаметр колеса;
- посилання на графічний контекст;
- посилання на текстовий блок для виведення розрахункових даних.

На рисунку 4.5 показана побудована розрахункова схема траєкторії руху мобільної платформи.

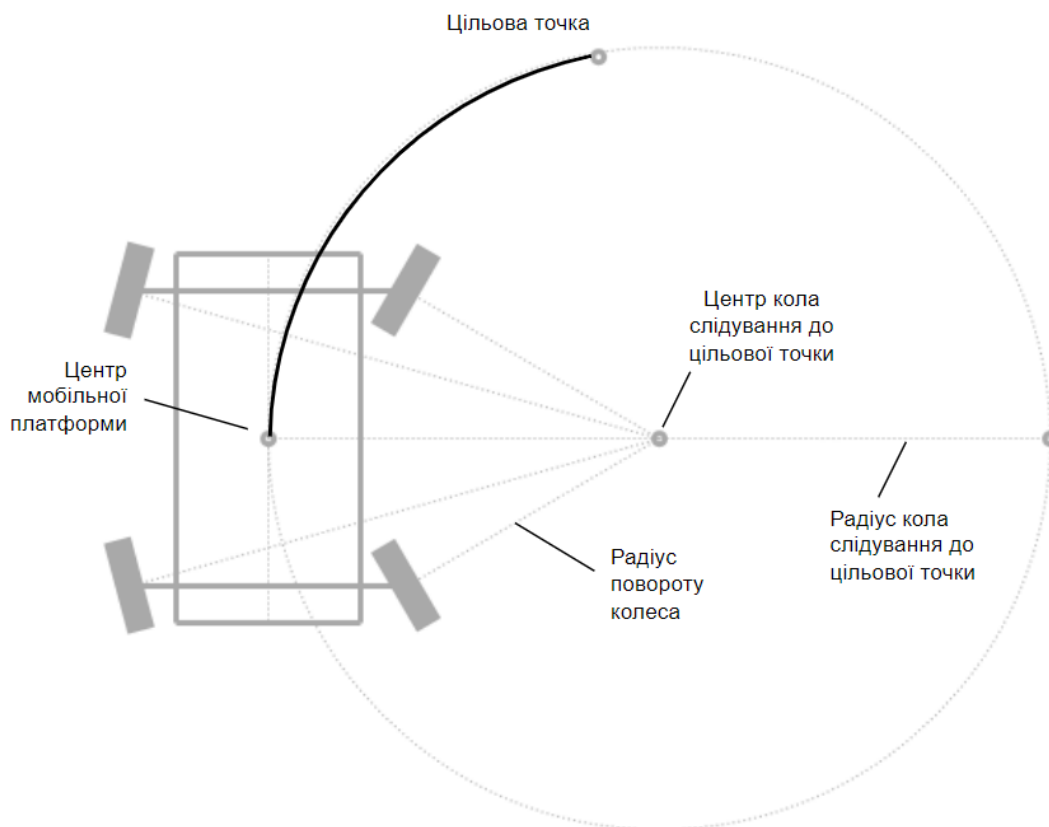


Рисунок 4.5 – Побудована розрахункова схема траєкторії руху мобільної платформи

На даному рисунку позначені радіуси повороту коліс, радіус кола розвороту платформи, розрахунковий центр траєкторії руху.

Кут повороту коліс визначається, як нормаль до радіусу від центру кола траєкторії руху до центру кріплення коліс на мобільній платформі.

Функція `drawWhell` викликається чотири рази, по кількості коліс у мобільній платформі.

На рисунку 4.6 показано приклад викликання цієї функції.

```
drawWhell(centerDestination, point_0sF_Left,
          (int)(whellDiameter / 2), e.Graphics, textBox1);
drawWhell(centerDestination, point_0sF_Right,
          (int)(whellDiameter / 2), e.Graphics, textBox2);
drawWhell(centerDestination, point_0sB_Left,
          (int)(whellDiameter / 2), e.Graphics, textBox3);
drawWhell(centerDestination, point_0sB_Right,
          (int)(whellDiameter / 2), e.Graphics, textBox4);
```

Рисунок 4.6 – Приклад застосування функції розрахунку куту повороту коліс

### 4.3 Аналіз результатів дослідження

На початку проведення дослідження платформа займає початкове положення, що становить  $90^\circ$  по відношенню до координат тестового поля.

Для позначення цільової точки необхідно натиснути в будь-якому місці тестового поля.

Після цього автоматично запускається процес розрахунку положення коліс та будується нова траєкторія руху роботу для досягнення вказаної точки.

На рисунку 4.8 показано приклад розрахунку куту повороту коліс для досягнення точки C1.

З наведеного рисунку можна бачити, що кут повороту переднього лівого колеса становить  $14^\circ$ ; кут повороту переднього правого колеса становить  $32^\circ$ ; кут повороту заднього лівого колеса становить  $14^\circ$ ; кут повороту заднього правого колеса становить  $32^\circ$ .

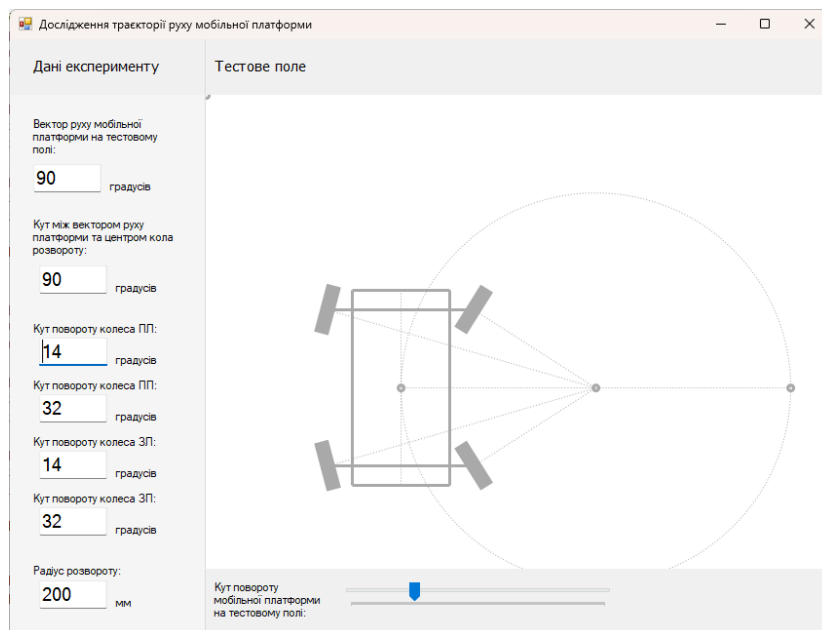


Рисунок 4.7 – Початкове положення платформи

Також розраховано радіус розвороту для досягнення вказаної точки. Цей радіус становить 415 мм.

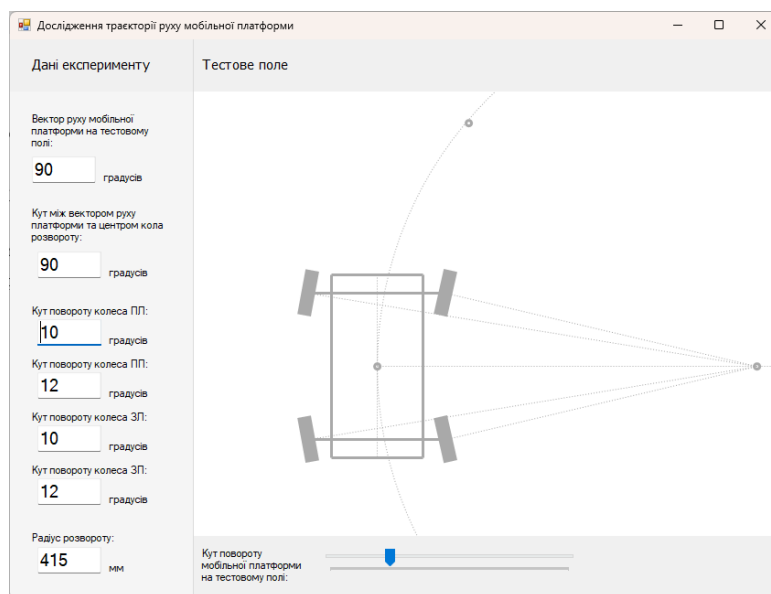


Рисунок 4.8 – Приклад розрахунку куту повороту коліс для досягнення точки

На рисунку 4.9 показано варіант зміни куту повороту коліс, у випадку зміни орієнтації мобільної платформи.

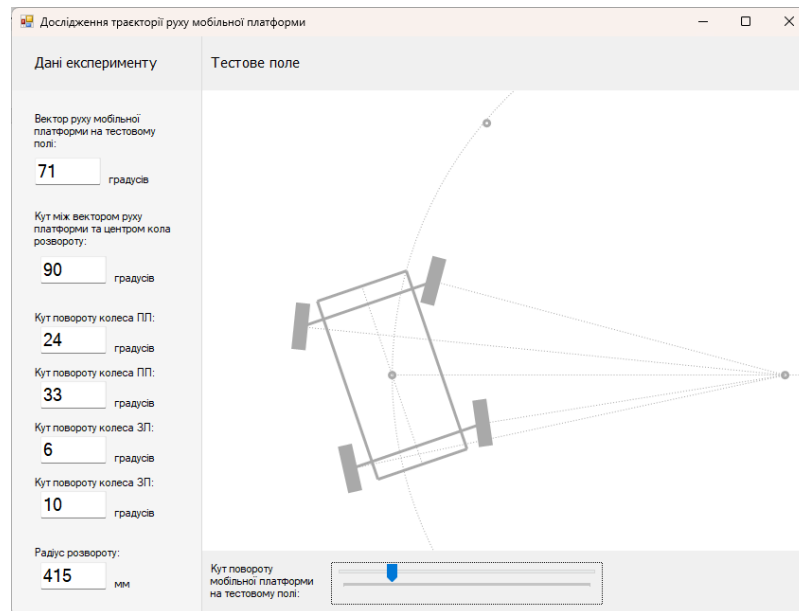


Рисунок 4.9 – Варіант зміни куту повороту коліс, у випадку зміни орієнтації мобільної платформи

Таке положення платформи є неприродним для розрахунку траєкторії руху, тому що завжди автоматично обирається оптимальний кут до центральної точки кола розвороту.

Оптимальним вважається перпендикуляр до головної центральної осі платформи.

Але розроблена програма дає можливість побачити та промоделювати різні варіанти поведінки мобільного робота.

За допомогою даної програми можливо знайти мінімальний радіус розвороту платформи.

Є такі значення радіусу розвороту, при яких колеса розташовуються майже перпендикулярно напрямку руху, а це не можливо реалізувати на практиці. На рисунку 4.10 показано такий випадок.

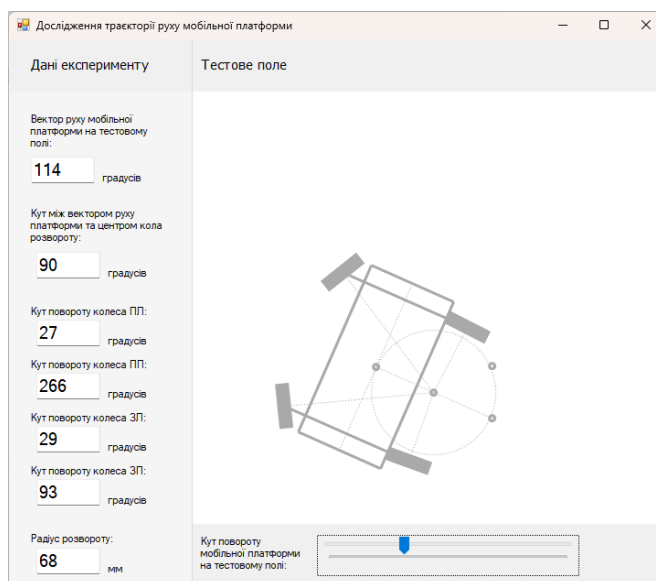


Рисунок 4.10 – Мінімальний радіус розвороту платформи

В даному було отримано мінімальний радіус в 68 мм.

Розроблена програма також дозволяє вказувати довільні координати центру розвороту.

Такі координати задаються правою кнопкою миші. На рисунку 4.11 показано приклад завдання координат майже у центрі платформи.

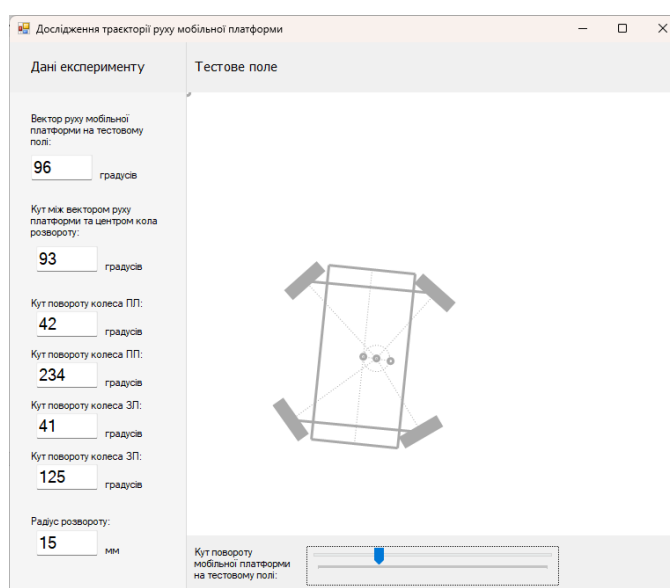


Рисунок 4.11 – Приклад розвороту платформи відносно вертикальної осі

Такий приклад можна застосувати на практиці для виконання розвороту в обмеженому просторі.

#### 4.4 Висновки до четвертого розділу

В четвертому розділі було проведено опис експериментальні дослідження, поставлено завдання проведення експерименту, проведено опис програмного засобу для виконання експериментальних досліджень, представлено аналіз результатів дослідження, наведено фрагменти коду програми а також скріншоти розробленого ПЗ.

## 5 ОХОРОНА ПРАЦІ

Розробка програмного забезпечення повинна здійснюватися у спеціально відведених місці. Він повинен мати такі параметри: вільний простір для кількох робочих місць, відповідну кількість ПК (з РК-моніторами). Характеристика електричної мережі повинна мати такі характеристики: трифазна чотирипровідна мережа змінного струму напругою 380/220 В частотою 50 Гц з незаземленою нейтраллю.

Крім того, щороку необхідно проводити контроль ізоляції при відключенні електроенергії відповідно до вимог ПУЕ-2011. Приклад функціональної схеми готового робочого місця представлено на рис. 5.1.

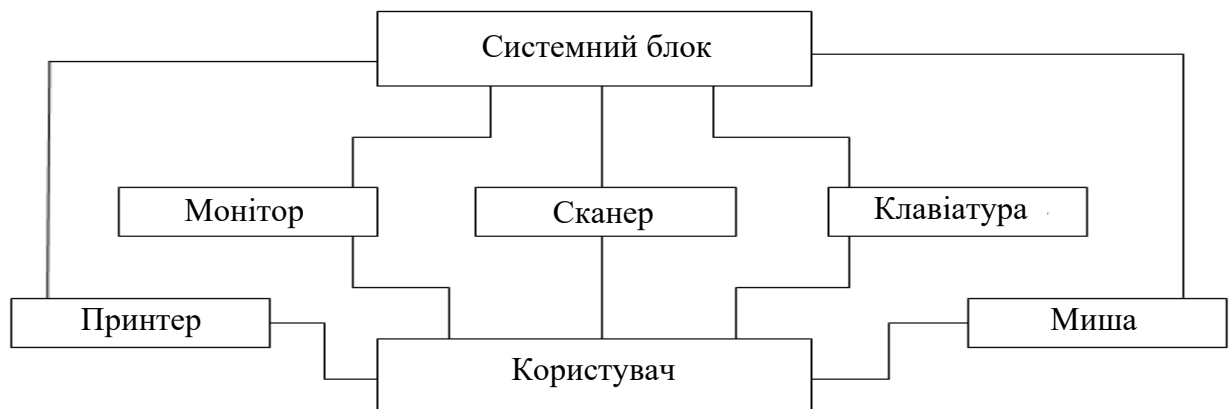


Рисунок 5.1 – Функціональна схема робочого місця

Згідно ДСН 3.3.6.042-99, є обов'язковим для встановлення на об'єкті певних метеорологічних засобів. (табл. 5.1).

Таблиця 5.1 – Норми мікроклімату для приміщень з ПК

Пора року	Температура повітря, °С	Відносна вологість повітря, %	Швидкість руху повітря, м/с
Холодна	від 21 до 24	від 40 до 60	0,1
Тепла	від 22 до 25	від 40 до 60	від 0,1 до 0,2

Відповідно до ДБН В.25-28-2006 для освітлення приміщень застосовують штучне та природне освітлення. При цьому рівень штучного освітлення повинен бути від 300 до 500 лк [31].

Відповідно до вимог НПАОП 0.00-1.28-10, ДСН 3.3.6.037-99 рівень шуму в приміщенні не повинен перевищувати значення 50 дБ.

Приміщення відділу розробки програмного забезпечення належить до об'єктів без підвищеного ризику ураження електричним струмом згідно з НПАОП 40.1-1.21-98. Це тому, що немає умов, які створюють підвищену небезпеку чи особливу небезпеку.

Також в приміщенні повинен бути електрощит, на якому будуть встановлені пристрої захисту від струму. Всі розетки мають застережливий напис «220». З робітниками проводяться інструктажі з охорони праці відповідно до НПАОП 0.00-4.12-05.

Згідно з вимогами НПАОП 0.00-4.12-05 обов'язково проводиться вступний, первинний, повторний, а за необхідності – позаплановий і цільовий інструктаж на виробництві.

Розрахунок занулення має на меті визначити умови, при яких воно надійно виконує покладені на нього завдання – швидко відключає пошкоджену установку від мережі і в той же час забезпечує безпеку дотику людини до занулення корпусу в аварійний період [32].

Схема занулення представлена на рис. 5.2, параметри цієї схеми відображені в табл. 5.2.

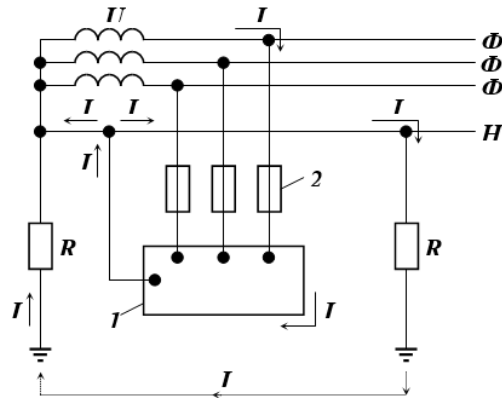


Рисунок 5.2 – Принципова схема занулення в трифазній мережі до 1000 В

Таблиця 5.2 – Параметри схеми занулення в трифазній мережі до 1000 В

$1$	корпус електроустановки;
$2$	апарати захисту від струмів короткого замикання (КЗ)
$\Phi$	фазний провід;
$НЗ$	нульовий захисний провідник;
$R_o$	опір заземлення нейтралі обмотки джерела струму;
$R_n$	опір повторного заземлення нульового захисного провідника;
$I_k$	ток КЗ;
$I_n$	частина струму КЗ, що протікає через нульовий захисний провідник;
$I_3$	частина струму КЗ, що протікає через землю.

### 5.1 Висновки до п'ятого розділу

У п'ятому розділі представлено функціональну схему робочого місця, наведено таблицю норми мікроклімату для приміщень з ПК, принципова схема занулення у трифазній мережі до 1000 В, а також розділ містить інформацію про загальну безпеку під час роботи з ПК.

## ВИСНОВКИ

В ході виконання роботи було проведено аналіз предметної області та технічного завдання. Було вивчено системи управління рухом мобільної платформи з чотирма рулевими колесами. Були виконані наступні завдання:

- проаналізовано використання мобільних роботів з чотирма рулевими колесами у сучасних виробничих системах;
- вивчена предметна область вже існуючих конструкцій мобільних роботів з чотирма рулевими колесами;
- проведено аналіз програмного забезпечення мобільних роботів з чотирма рулевими колесами;
- проведено аналіз основи рулевого управління та існуючих систем рульового управління;
- зроблено ескізне проектування модуля поворотних коліс;
- розроблено алгоритм управління поворотним колісним модулем;
- розроблено алгоритм синхронного управління комплексом поворотних колісних модулів;
- проведено тестування розробленого програмного забезпечення для мобільного робота.

У результаті було досягнуто основна мети роботи: розроблено метод визначення орієнтації коліс в залежності від траєкторії руху мобільної платформи.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки з підготовки та захисту кваліфікаційної роботи здобувачами другого (магістерського) рівня вищої освіти спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології, освітньо-професійних програм: «Автоматизоване управління технологічними процесами», «Комп'ютерно-інтегровані технологічні процеси і виробництва», «Комп'ютеризовані та робототехнічні системи» / Упоряд. І. Ш. Невлюдов, Р. В. Артюх, В. В. Безкоровайний, Н. П. Демська, В. В. Євсєєв, О. І. Филипенко, О. М. Цимбал. Харків: ХНУРЕ, 2021. 55 с.
2. ДСТУ 3008: 2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлення. К.: ДП “УкрНДНЦ”. 2016. 30 с.
3. Дипломне проектування для студентів усіх форм навчання спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології»: довід. / І. Ш. Невлюдов, А. О. Андрусевич, О. В. Токарева, Г. В. Пономарьова. Київ, 2018. 320 с.
4. Будова та система керування мобільної платформи на базі робота павука / Бєлов П. О., Мандрікін М. С., Катков О. В., Цимбал О. М. Харків: Topical issues of modern science, society and education. Proceedings of the 6th International scientific and practical conference, 2021. 343 с.
5. Основы робототехники / Роберт Букарев. Харків. 2017. 416 с.
6. K. Tchoń, K. Arent, M. Janiak, and Ł. Juskiewicz, “Motionplanning for the mobile platform Rex”, in Recent Advances in Automation, Robotics and Measuring Techniques, P. 497–506. 2017.
7. T. Chen et al. Simultaneous path following and lateral stability control of 4WD-4WS autonomous electric vehicles with actuator saturation Adv. Eng. Soft. 2019. P. 423.
8. R. Seeber et al. Conditioned Super-twisting algorithm for systems with saturated control action Automatica. 2020. P. 234.

9. X. Tu et al. Robust navigation control of a 4WD/4WS agricultural robotic vehicle Comput. Electron. Agric. 2019. P. 127.
10. Офіційний сайт Robotshop // Сайт Robotshop. URL: <https://www.robotshop.com/en/rosbot-plus-nano.html> (дата звернення: 18.10.2022).
11. Офіційний сайт Naio Technologies // Сайт Naio Technologies. URL: <https://www.naio-technologies.com/en/oz/> (дата звернення: 18.10.2022).
12. Офіційний сайт Naio Technologies // Сайт Naio Technologies. URL: <https://www.naio-technologies.com/en/dino/> (дата звернення: 19.10.2022).
13. Офіційний сайт Naio Technologies // Сайт Naio Technologies. URL: <https://www.naio-technologies.com/en/> (дата звернення: 29.10.2022).
14. Офіційний сайт Naio Technologies // Сайт Naio Technologies. URL: <https://www.naio-technologies.com/en> (дата звернення: 29.10.2022).
15. A Gentle Introduction to ROS / Джейсон М. О'кейн. 2018. с. 166.
16. Lentin Joseph. Robot Operating System (ROS) for Absolute Beginners: Robotics Programming Made Easy. Apress, 2018. P. 207.
17. Wyatt Newman. A Systematic Approach to Learning Robot Programming with ROS. Chapman and Hall/CRC. 2017. P. 376.
18. Programming Arduino: Getting Started with Sketches / Монк С. 2018. 192 с.
19. Jeremy Blum; Exploring Arduino: Tools and Techniques for Engineering Wizardry. 2nd ed. Wiley. P. 512. 2019.
20. Learning to Program with Robotc / Алекс Кроу. 2020. 70 с.
21. RobotC – кросс-робототехнический язык программирования / Векслер В. А. 2019. С. 10-20.
22. Practical Linux with Raspberry Pi OS / Ашвін П. 2021. 176 с.
23. Adventures In Raspberry Pi / Carrie A. P. 2019. P. 256.
24. Decoupling control of steering and driving system for in-wheel-motor-drive electric vehicle / H. Zhang. 2018. С. 15-30.

25. Mechatronic design of a four wheel steering mobile robot with fault-tolerant odometry feedback / Oftadeh R., Aref M.M., Ghabcheloo R., Mattila J. 2018. P. 663-669.

26. Path tracking controller design of four wheel independent steering automatic guided vehicle / Setiawan Y.D., Nguyen T.H., Pratama P.S., Kim H.K., Kim S.B. 2016. P. 10-27.

27. Design of the dual offset active caster wheel for holonomic omnidirectional mobile robots / Chung W., bae Moon C., Jung C., Jin J. 2020. P. 26-37.

28. Yanmar leads agricultural robotics project to develop // «SMASH» system  
URL: <https://roboticsandautomationnews.com/2020/04/20/yanmar-leads-agricultural-robotics-project-to-develop-smash-system/31807/> (дата звернення: 30.11.2022).

29. Hua, Fei & Li, Guoli & Liu, Fang & Liu, Yongbin. Mechanical design of a four-wheel independent drive and steering mobile robot platform. 2016. P. 235-238.

30. FRC Team 1640 // Сайт FRC Team. URL: [http://team1640.com/wiki/index.php/FRC\\_Team\\_1640](http://team1640.com/wiki/index.php/FRC_Team_1640) (дата звернення: 30.11.2022).

31. Управління охороною праці та ризиком за міжнародними стандартами / Гогіташвілі Г. Г., Карчевські Є.-Т., Лапін В. М. К. 2017. 367 с.

32. Охорона праці: Навч. посіб. / Я. І. Бедрій, Є. О. Геврик, І. Я. Кіт, О. С. Мурін, В. М. Єнкало. 2020. 280 с.