

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут автоматики та електротехніки**

Кафедра комп'ютеризованих систем управління

«Допущений до захисту»

Завідувач кафедри

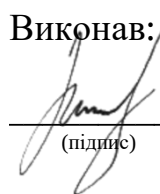
 Черно О.О.
«16» 06 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

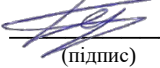
на тему: «Розробка системи керування роботом-маніпулятором»

Виконав: студент 4347ст групи

 **Ткач В.В.**
(підпис)

Керівник роботи:

доцент кафедри КСУ, к.т.н., доцент
(посада, науковий ступень вчене звання)

 **Герасін О.С.**
(підпис)

м. Миколаїв – 2025 р.

Національний університет кораблебудування імені адмірала Макарова

ННІ автоматики і електротехніки

Кафедра комп'ютеризованих систем управління

Рівень вищої освіти бакалавр

Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка

(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри КСУ

 **Черно О.О.**
« » 2025 р.

ЗАВДАННЯ **НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Т к а ч у В і т а л і ю В а л е р і й о в и ч у

(прізвище, ім'я, по батькові)

1. Тема роботи: розробка системи керування роботом-маніпулятором

керівник роботи Герасін Олександр Сергійович, к.т.н., доцент

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

затверджені наказом вищого навчального закладу від "23" 04 2025 р. № УЧ-343

2. Строк подання студентом роботи 09.06.2025

3. Вихідні дані до роботи Розробити систему керування роботом-

маніпулятором з базовим інтерфейсом користувача та режимами ручного керування,
автоматичного пошуку для захоплення об'єктів маніпулювання та переміщення у
вказані координати.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Аналіз проблем керування сучасними роботами-маніпуляторами.

Розробка математичної моделі робота-маніпулятора.

Програмна реалізація системи керування роботом-маніпулятором.

Експериментальне дослідження розробленої системи.

Охорона праці.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Сучасні тенденції автоматизації.

Щільність автоматизації промисловості 2013-2022 (середньосвітове значення).

Щільність автоматизації промисловості станом на 2023 рік.

Комплекс завдань що можуть виконуватися роботами-маніпуляторами.

Переваги паралелограмних механізмів.

Мета і основні задачі роботи.

Варіанти конфігурації Five-Bar Linkage.

Математична модель досліджуваного механізму (пряма задача кінематики).

Зворотна задача кінематики.

<i>Порівняння робочої зони планарних маніпуляторів</i>
<i>Методи інтерполяції траєкторії.</i>
<i>Порівняння методів інтерполяції траєкторії.</i>
<i>Принципова схема системи.</i>
<i>Програмні модулі системи.</i>
<i>Зворотна задача кінематики.</i>
<i>Порівняння робочої зони планарних маніпуляторів</i>
<i>Експериментальне дослідження системи.</i>
<i>Демонстраційне відео роботи системи.</i>
<i>Висновки.</i>

6. Консультанти розділів роботи

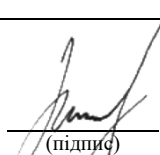
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
<i>Охорона праці</i>	<i>Герасін О.С., доцент каф. КСУ</i>	<i>31.03.2025</i>	<i>19.05.2025</i>

7. Дата видачі завдання: «15» 02 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів підготовки кваліфікаційної роботи	Строк виконання	Примітка
1.	Аналіз проблем керування сучасними роботами-маніпуляторами	10.03.2025	виконано
2.	Розробка математичної моделі робота-маніпулятора	31.03.2025	виконано
3.	Програмна реалізація системи керування роботом-маніпулятором	21.04.2025	виконано
4.	Експериментальне дослідження розробленої системи	12.05.2025	виконано
5.	Охорона праці	19.05.2025	виконано
6.	Оформлення пояснювальної записки кваліфікаційної роботи	02.06.2025	виконано

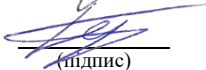
Студент


(підпис)

Ткач В.В.

(прізвище та ініціали)

Керівник роботи


(підпис)

Герасін О.С.

(прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота присвячена розробці системи керування роботом-маніпулятором із паралелограмним механізмом для забезпечення високої точності позиціонування та автоматичного пошуку об'єктів.

Розроблено математичну модель робота з механізмом Five-Bar Linkage (3 ступені вільності), розв'язано задачі кінематики та визначено робочий простір. Досліджено методи планування траєкторій з використанням трапецієподібної, поліноміальної та сплайнової інтерполяції. Створено програму керування з ручним та автоматичним режимами, включаючи алгоритм пошуку об'єктів з ультразвуковим датчиком.

Експериментально підтверджено працездатність системи: похибка позиціонування ± 2 мм (X,Y), $\pm 3,4$ мм (Z), ефективність пошуку об'єктів 98%. Результати застосовні для автоматизації промислових процесів та навчальних цілей.

Ключові слова: робот-маніпулятор, паралелограмний механізм, система керування, кінематика, планування траєкторій, автоматичний пошук об'єктів, позиціонування.

ABSTRACT

This thesis focuses on developing a control system for a robot manipulator with parallelogram-type mechanism to ensure high positioning accuracy and efficient automatic object search.

A mathematical model of the Five-Bar Linkage robot with 3 degrees of freedom was developed, direct and inverse kinematics problems were solved, and workspace was determined. Motion trajectory planning methods using trapezoidal, polynomial, and spline interpolation were investigated. Control software with manual and automatic modes was created, including automatic object search algorithm using ultrasonic sensor.

Experimental studies confirmed system operability: positioning error ± 2 mm (X, Y axes), ± 3.4 mm (Z axis), object search efficiency 98%. Results are applicable for industrial automation and educational purposes.

Keywords: robot manipulator, parallelogram mechanism, control system, kinematics, trajectory planning, automatic object search, positioning.

ЗМІСТ

ВСТУП	8
Розділ 1. АНАЛІЗ ПРОБЛЕМ КЕРУВАННЯ СУЧАСНИМИ РОБОТАМИ- МАНІПУЛЯТОРАМИ	11
1.1. Огляд конструктивних особливостей роботів-маніпуляторів із механізмом паралелограмного типу	11
1.2. Аналіз систем керування промисловими роботами	16
1.3. Методи визначення положення та орієнтації захвата робота- маніпулятора	24
1.4. Висновки за розділом	27
Розділ 2. РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ РОБОТА-МАНІПУЛЯТОРА	28
2.1. Кінематичний аналіз механізму паралелограмного типу	28
2.2. Математичне моделювання планування траєкторії маніпулятора	41
2.3. Висновки за розділом	51
Розділ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ КЕРУВАННЯ РОБОТОМ- МАНІПУЛЯТОРОМ	53
3.1. Розробка програми ручного керування	53
3.2. Програмування алгоритму поліноміальної інтерполяції	55
3.3. Програмування алгоритму розв’язання зворотної задачі кінематики... ..	57
3.4. Програмування алгоритму встановлення кута серводвигунів	58
3.5. Програмування алгоритму вимірювання дистанції	60
3.6. Складання та реалізація алгоритму автоматичного пошуку об'єктів маніпулювання	61
3.7. Інтеграція режимів керування та розробка інтерфейсу користувача	64
3.8. Висновки за розділом	68

Розділ 4. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РОЗРОБЛЕНОЇ СИСТЕМИ	70
4.1. Методика проведення експериментальних досліджень.....	70
4.2. Аналіз точності позиціонування захвату робота-маніпулятора	71
4.3. Оцінка ефективності алгоритму пошуку об'єктів роботом	73
4.4. Висновки за розділом	74
Розділ 5. ОХОРОНА ПРАЦІ	75
5.1. Загальні поняття охорони праці	75
5.2. Аналіз небезпечних та шкідливих факторів, що діють на оператора системи керування роботом-маніпулятором.....	75
5.3. Заходи з техніки безпеки для зменшення або усунення впливу небезпечних та шкідливих факторів при обслуговуванні запропонованої системи	77
5.4. Висновки за розділом	86
ВИСНОВКИ.....	88
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	90
ДОДАТОК А.....	91
ДОДАТОК Б	93

ВСТУП

Сучасні технологічні процеси в промисловості та науці характеризуються зростаючою потребою в автоматизації виробництва. Особливо актуальним напрямком розвитку робототехніки є вдосконалення роботів-маніпуляторів, які здатні виконувати широкий спектр задач з високою точністю, швидкістю та надійністю. Серед різноманітних конструкцій роботів-маніпуляторів особливу увагу привертають механізми паралелограмного типу, які мають суттєві переваги перед традиційними послідовними кінематичними конфігураціями.

Актуальність теми дослідження зумовлена необхідністю розробки ефективних систем керування роботами-маніпуляторами з механізмом паралелограмного типу, які б забезпечували високу точність позиціонування, оптимальне планування траєкторій та можливість автоматичного пошуку об'єктів маніпулювання. Використання таких систем сприяє підвищенню продуктивності праці, зниженню виробничих витрат та підвищенню якості продукції в різних галузях промисловості.

В останні роки значна увага приділяється дослідженню кінематики та динаміки роботів паралелограмного типу. Зокрема, в роботах вітчизняних та закордонних вчених, значна увага приділяється розв'язанню прямої та зворотної задач кінематики, аналізу сингулярностей та робочого простору маніпуляторів. Сучасні дослідження спрямовані на розробку алгоритмів інтерполяції траєкторій, що забезпечують плавний рух робота-маніпулятора, мінімізацію ривків та енергоспоживання.

Метою роботи є розробка та експериментальне дослідження системи керування роботом-маніпулятором із механізмом паралелограмного типу, що забезпечує високу точність позиціонування та ефективний автоматичний пошук об'єктів маніпулювання.

Для досягнення поставленої мети сформульовано наступні завдання:

- проаналізувати конструктивні особливості роботів-маніпуляторів із механізмом паралелограмного типу та методи керування ними.
- Розробити математичну модель кінематики та динаміки робота-маніпулятора паралелограмного типу.
- Розробити алгоритми планування траєкторій руху маніпулятора з використанням різних методів інтерполяції.
- Створити програмну реалізацію системи керування роботом-маніпулятором із режимами ручного та автоматичного керування.
- Розробити алгоритм автоматичного пошуку об'єктів маніпулювання з використанням ультразвукового датчика.
- Провести експериментальне дослідження розробленої системи для визначення точності позиціонування та ефективності алгоритму пошуку об'єктів.
- Розробити заходи з охорони праці для забезпечення безпечної експлуатації системи керування роботом-маніпулятором.

Об'єктом дослідження є процес автоматизованого керування роботом-маніпулятором із механізмом паралелограмного типу.

Предметом дослідження є методи та алгоритми керування роботом-маніпулятором, що забезпечують високу точність позиціонування та автоматичний пошук об'єктів маніпулювання.

Практичне значення одержаних результатів полягає в створенні програмно-апаратного комплексу для керування роботом-маніпулятором, який може бути використаний у навчальному процесі, дослідницькій роботі та в промисловості для виконання задач автоматизованого захоплення та переміщення об'єктів. Розроблена система забезпечує як ручне керування через

інтуїтивно зрозумілий інтерфейс, так і автоматичне виконання операцій пошуку та захоплення об'єктів.

РОЗДІЛ 1

АНАЛІЗ ПРОБЛЕМ КЕРУВАННЯ СУЧАСНИМИ РОБОТАМИ- МАНІПУЛЯТОРАМИ

1.1. Огляд конструктивних особливостей роботів-маніпуляторів із механізмом паралелограмного типу

Роботи-маніпулятори класифікуються за кінематичною структурою на послідовні та паралельні.

Послідовні маніпулятори складаються з ланок, з'єднаних послідовно за допомогою шарнірів. Кожен суглоб додає один ступінь свободи, що дозволяє досягати високої гнучкості, але може знижувати жорсткість конструкції. Наприклад – PUMA 6R (Рис. 1.1):

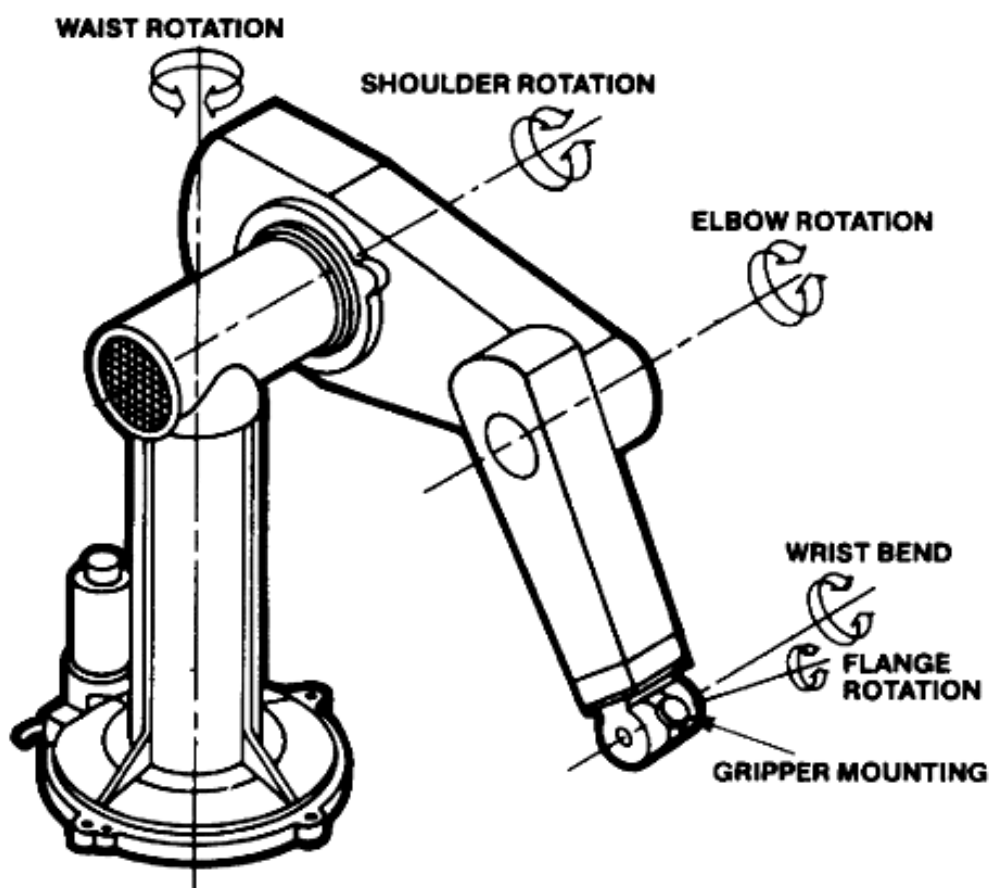


Рис. 1.1. Маніпулятор PUMA 6R

Паралельні маніпулятори мають рухому платформу, з'єднану з нерухомою основою через кілька незалежних кінематичних ланцюгів. Вони забезпечують високу жорсткість і точність, але мають обмежену робочу область. Приклад – платформа Стюарта (Рис. 1.2).

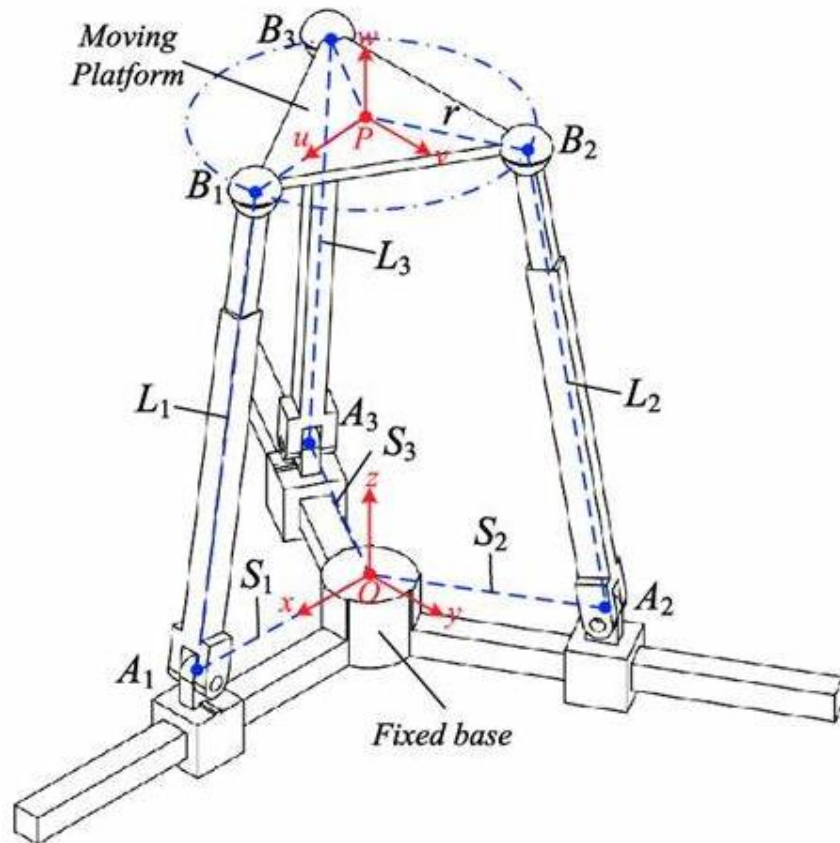


Рис. 1.2. Маніпулятор типу «платформа Стюарта»

Одним із різновидів паралельних маніпуляторів є маніпулятори з механізмом паралелограмного типу. Вони використовують кінематичні ланцюги у формі паралелограмів для забезпечення синхронного руху ланок, що покращує стабільність і точність. До таких конструкцій належать Five-Bar Parallel Manipulator (Рис. 1.3) та Parallelogram-Based Robotic Arm (Рис. 1.4).

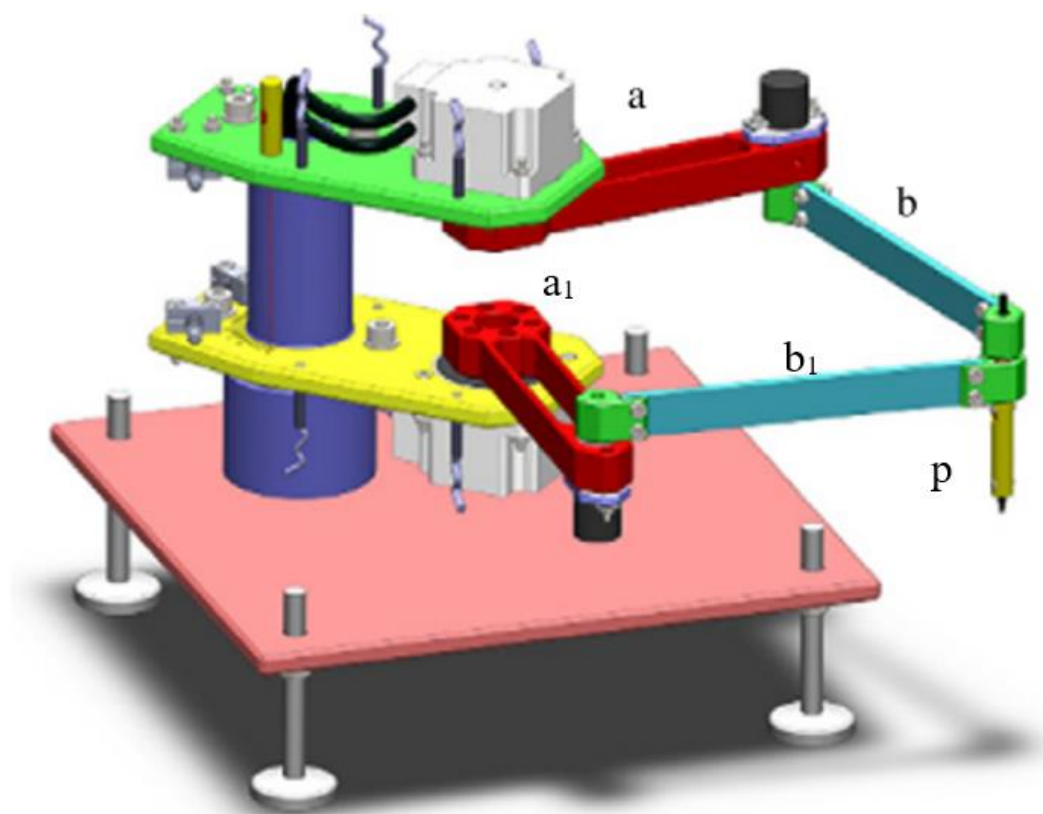


Рис. 1.3. Робот Five-Bar Parallel Manipulator

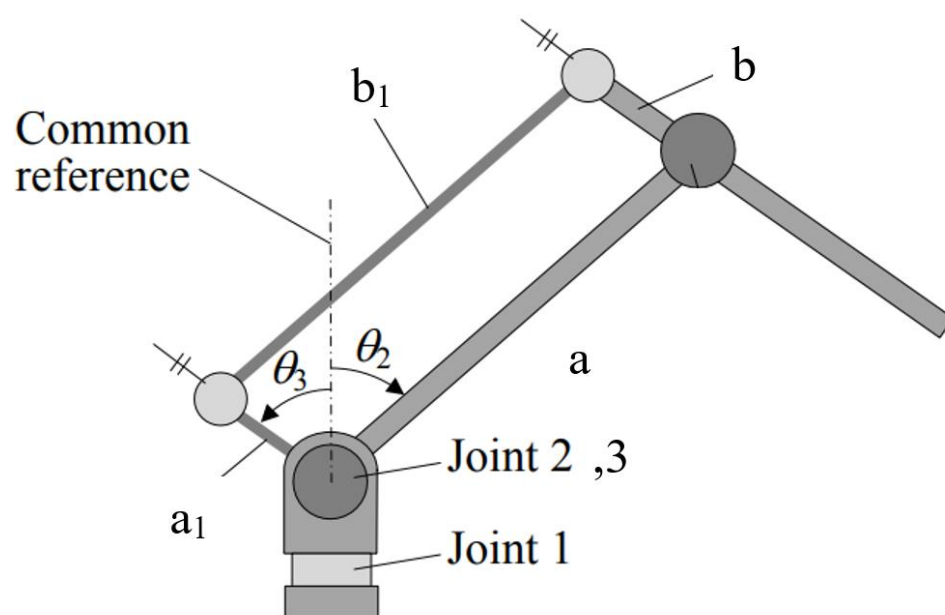


Рис. 1.4. Робот Parallelogram-Based Robotic Arm

Розглянемо особливості кожної з конфігурацій:

Five-Bar Parallel Manipulator. Маніпулятор з механізмом типу 5R (5 Revolute linkages – 5 обертальних з'єднань) має два ступені свободи та планарну робочу зону:

- база – нерухома основа, до якої кріпляться два приводи;
- дві керовані ланки (a , a_1) – з'єднані з приводами на базі через обертові з'єднання (активні суглоби);
- дві пасивні ланки (b , b_1) – з'єднані з робочим органом, та суглобами активних ланок;
- робочий орган (p) – кінцева точка механізму яка виконує певну дію.

Parallelogram-Based Robotic Arm [1] є одним з різновидів такого маніпулятора:

- перше з'єднання (joint 1) – з'єднання руки робота з платформою;
- друге з'єднання (joint 2) – з'єднує активну ланку (a) та двигун на платформі, який відповідає за зміну кута θ_2 ;
- третє з'єднання (joint 3) – з'єднує активну ланку (a_1) та двигун на платформі, який відповідає за зміну кута θ_3 . не пов'язаний механічно з другим з'єднанням;
- робочий орган (p) в такому випадку монтується на вільний кінець пасивної ланки (b).

Область застосування таких конфігурацій досить широка, і окрім класичних задач з маніпуляціями над об'єктами, можна виділити застосування їх в якості нижніх кінцівок поширених нині чотирилапих роботів (Рис. 1.5-1.6):

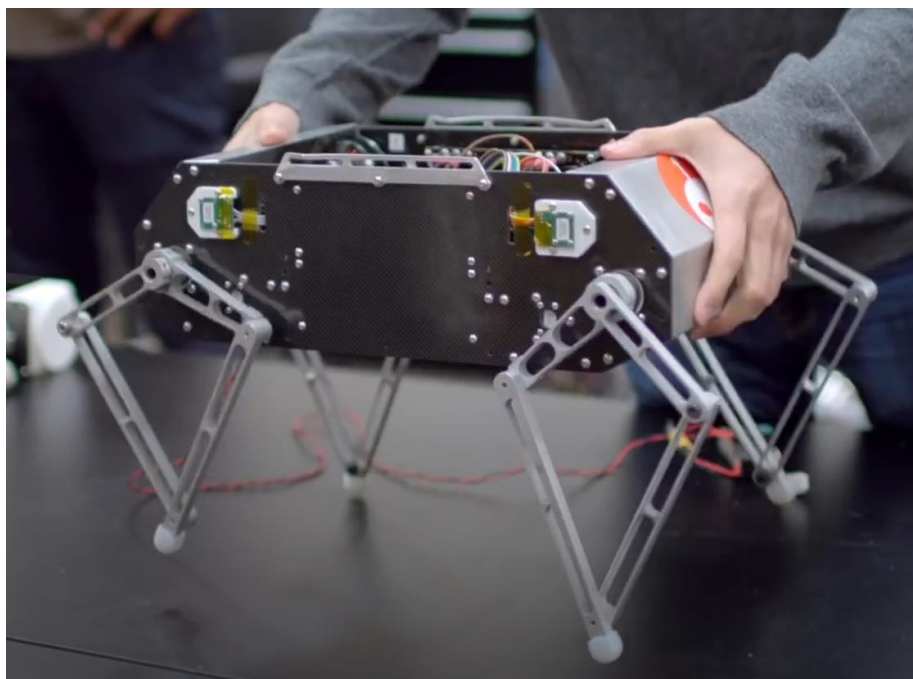


Рис. 1.5. Stanford's Doggo

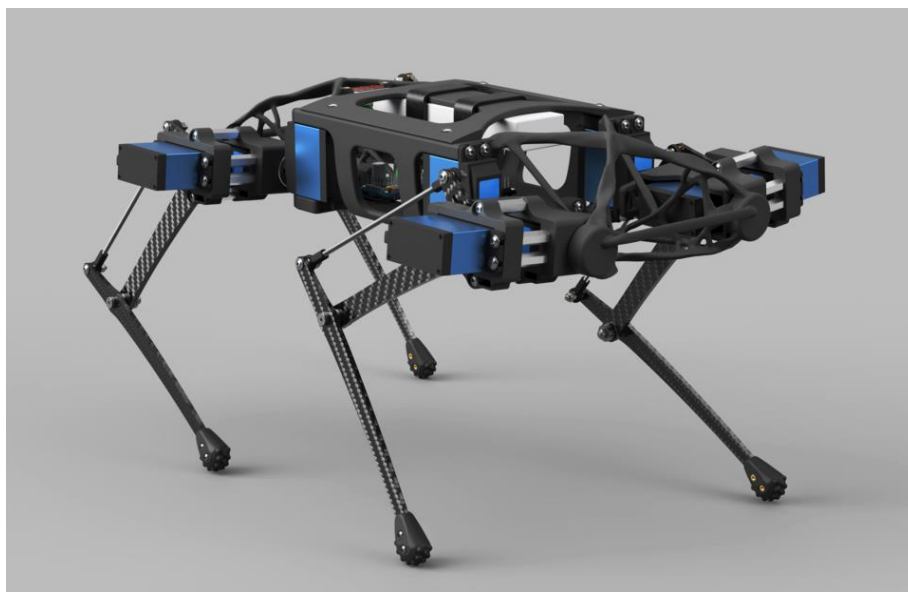


Рис. 1.6. CAD модель прототипу чотириноного робота

Таким чином, паралельний маніпулятор (Five-Bar Parallel Manipulator) відзначається високою жорсткістю конструкції завдяки закритим кінематичним ланцюгам та демонструє відмінні показники точності позиціонування.

Рівномірний розподіл навантаження між приводами та можливість використання нерухомо встановлених приводів зменшує навантаження на ланки та інерцію системи. Проте ця конфігурація має суттєві обмеження: робочий простір значно менший порівняно з послідовними маніпуляторами, а складна кінематика та особливості конструкції створюють ризик виникнення сингулярностей [3], хоча, існують методи їх запобігання [4]. Взаємозалежність рухів ускладнює керування, а маневреність у певних областях робочого простору обмежена. Додатково виникають складнощі з масштабуванням конструкції, здебільшого через зростання обчислювальної складності програми.

Маніпулятори на основі паралелограмних механізмів (Parallelogram-Based Robotic Arm) пропонують збалансоване поєднання характеристик послідовних та паралельних систем. Вони забезпечують кращу жорсткість порівняно з чисто послідовними конфігураціями та простішу компенсацію гравітаційних впливів [1]. Робочий простір таких маніпуляторів більший, ніж у чисто паралельних систем. Водночас вони поступаються паралельним конфігураціям у жорсткості та мають складнішу конструкцію порівняно з послідовними маніпуляторами. Паралелограмні механізми вносять додаткові обмеження на рухи ланок та створюють труднощі у проектуванні й виготовленні точних механізмів.

Компромісний підхід до проектування маніпуляторів, що поєднує елементи послідовних та паралельних конфігурацій, дозволяє досягти оптимального балансу між робочою зоною, жорсткістю, вантажопідйомністю та складністю управління для конкретних задач.

1.2. Аналіз систем керування промисловими роботами

Система керування промислового робота — це комплекс апаратних і програмних засобів, що реалізує алгоритм керування відповідно до заданих параметрів траєкторії, швидкості, прискорення та навантаження. Вона

забезпечує замкнений контур регулювання з використанням зворотного зв'язку за положенням, швидкістю та моментом.

Основними функціями системи керування є:

- генерація траєкторії – формування керуючих впливів для досягнення заданих координат;
- інтерполяція руху – визначення проміжних точок траєкторії між заданими ключовими положеннями;
- контроль динаміки системи – забезпечення стійкості та коректної роботи привідної системи;
- корекція помилок – обробка зворотного зв'язку з використанням під-регуляторів або адаптивних алгоритмів.

Системи керування промисловими роботами класифікують за кількома критеріями: архітектурою, принципом керування, рівнем автоматизації та способом формування керуючих впливів.

За структурою керування:

- Централізовані системи - всі функції керування виконуються одним центральним контролером. Прикладами таких систем є:
 - KUKA KR200 з контролером KR C1 (рис. 1.7), де один потужний контролер керує всіма рухами та функціями робота;
- Децентралізовані (розподілені) системи - функції розподілені між кількома контролерами. Наприклад:
 - система керування зварювальною роботизованою лінією, де окремі контролери керують зварювальними роботами, конвеєрною стрічкою та системою подачі деталей;
 - виробнича лінія з декількома роботами, де кожен робот має власний контролер, але вони об'єднані в мережу для синхронізації (рис. 1.8).



Рис. 1.7. Система KUKA KR200 + KR C1

- Ієрархічні системи - багаторівнева структура з підпорядкуванням нижчих рівнів вищим:
 - автоматизована виробнича лінія з головним контролером (верхній рівень), PLC-контролерами (середній рівень) та локальними контролерами роботів (нижній рівень);
 - система керування автомобільною складальною лінією Toyota, де багаторівнева система контролює роботу десятків роботів і конвеєрів (рис. 1.8).

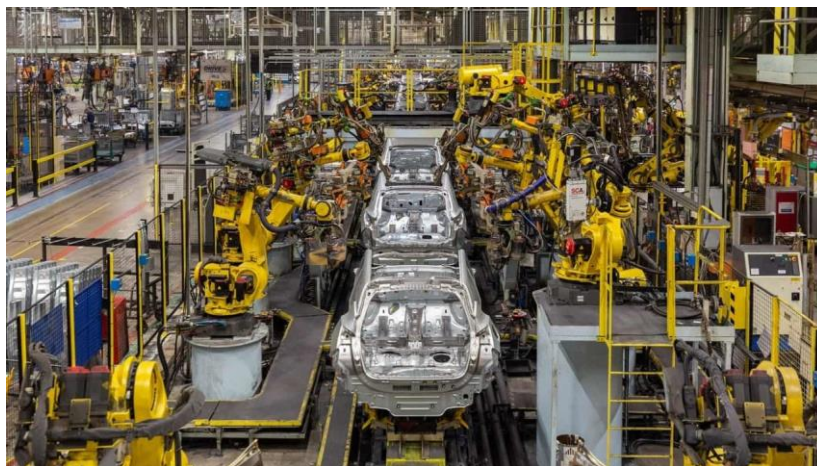


Рис. 1.8. Автоматизована виробнича лінія

За типом керування:

- Програмні (жорсткі) - виконують заздалегідь запрограмовану послідовність дій:
 - роботи для точкового зварювання в автомобільній промисловості, які виконують однакові рухи для кожної деталі;
 - роботи-пакувальники, що виконують циклічні операції пакування за заданою програмою (рис. 1.9).



Рис. 1.9. Робот що пакує продукцію у пластикові контейнери.

- Адаптивні - можуть змінювати свою поведінку залежно від зміни робочого середовища:
 - робот ABB FlexPicker з системою машинного зору, який визначає положення об'єктів на конвеєрі та адаптує свої рухи (рис. 1.10).

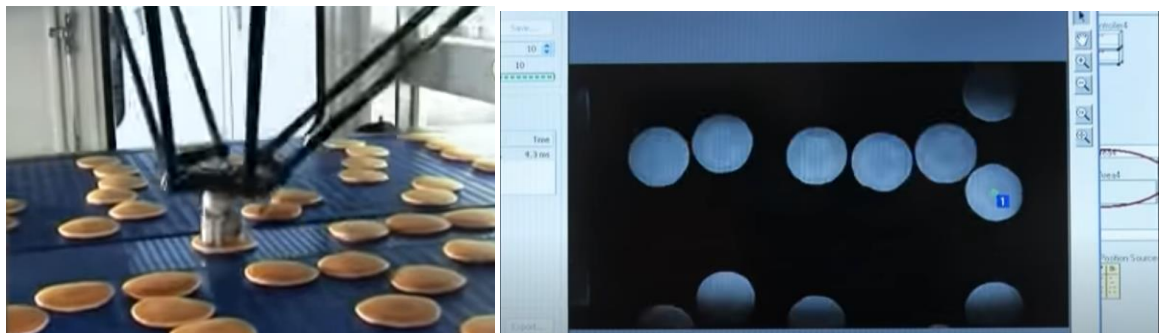


Рис. 1.10. FlexPicker складує оладки на середню лінію та кадр з камери робота (зліва направо)

- Інтелектуальні - використовують елементи штучного інтелекту для прийняття рішень:
 - роботи-сортувальники з ШІ-системою, що розпізнають та класифікують різні види відходів на лінії переробки;
 - роботи Figure Helix з розширеними можливостями штучного інтелекту, які працюють поруч з людьми та навчаються нових задач (рис. 1.11).

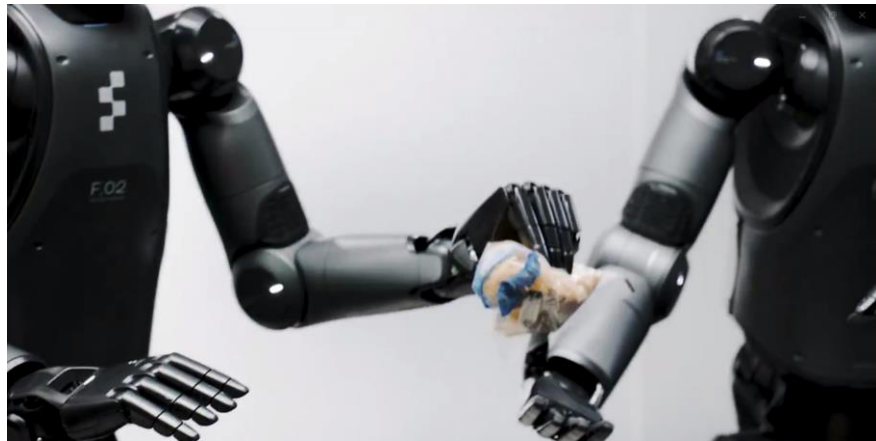


Рис. 1.11. Гуманоїд передає предмет іншому гуманоїду в рамках спільного виконання задачі поставленої людиною

За способом програмування:

- За допомогою навчального пульта (teach pendant):
 - промисловий робот Kawasaki RS010L, який програмується оператором через пульт керування E01 (рис. 1.12).



Рис. 1.12. Система «робот + контролер» з навчальним пультом на передньому плані

- Програмування мовами робототехніки (Robot Programming Language):
 - роботи ABB, що програмуються мовою RAPID;
 - роботи KUKA, що використовують мову KRL (KUKA Robot Language);
 - роботи FANUC, що програмуються на мові Karel або TP.
- Offline-програмування з використанням CAD/CAM систем
 - роботи для фрезерування складних деталей, запрограмовані через систему Siemens NX CAM (рис. 1.13);
 - зварювальні роботи, траєкторії яких створені в системах Robotmaster або RoboDK на основі 3D-моделей.

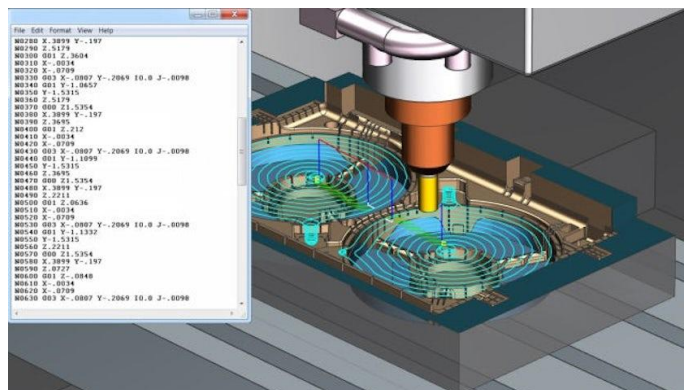


Рис. 1.13. Скріншот симуляції обробки деталі в NX CAM

- Програмування через демонстрацію (Programming by Demonstration):
 - роботи Franka Emika Panda, які запам'ятовують рухи після фізичного проведення їх маніпулятором;
 - робот Rethink Robotics Sawyer, який може навчатись нових дій шляхом демонстрації оператором (рис. 1.14).



Рис. 1.14. Оператор в режимі Zero gravity mode направляє маніпулятор в необхідну йому точку

За рівнем зворотного зв'язку:

- системи з відкритим контуром (без зворотного зв'язку)
 - будь-які роботи, що не містять саморегулювання. Наприклад, базові роботи для нанесення клею з попередньо запрограмованими траєкторіями.
- Системи із замкненим контуром (зі зворотним зв'язком)
 - будь-які роботи, система керування яких передбачає саморегулювання. Наприклад, роботи-маніпулятори.
- Комбіновані системи

- робот для зварювання з елементами машинного зору і датчиками тиску;
- роботи для складання електроніки з тактильними датчиками і елементами машинного зору.

За типом приводів:

- Гідравлічні системи керування
 - важкі роботи-маніпулятори для будівництва і роботи з великими вантажами.
- Пневматичні системи керування
 - роботи-маніпулятори Festo для швидкої обробки легких деталей (рис. 1.15);
 - пакувальні роботи з пневматичними захватами.



Рис. 1.15. Маніпулятор Festo

- Електричні системи керування
 - більшість сучасних промислових роботів (ABB, KUKA, FANUC, Yaskawa).
- Комбіновані системи
 - роботи з електричними приводами та пневматичними захватами;

- гібридні системи з електричними приводами для прецизійних рухів та гідравлічними для роботи з великим навантаженням (перші моделі Atlas, створені до 2024 року).

1.3. Методи визначення положення та орієнтації захвата робота-маніпулятора

Далі доцільно розглянути приклад маніпулятора з трьома ланками.

Визначення положення та орієнтації робочого органа на основі заданих значень узагальнених координат (кутів обертання або лінійних переміщень ланок) – вирішує пряма задача кінематики.

Відомо про два аналітичних методи вирішення цієї задачі:

- матричний – полягає у визначенні «матриці маніпулятора» [2]. Згідно з цим методом спочатку, складаються матриці елементарних поворотів $R_{x,\alpha}, R_{y,\phi}, R_{z,\theta}$, перемноженням яких отримується матриця повороту R :

$$R_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}, \quad (1.1)$$

$$R_{y,\phi} = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}, \quad (1.2)$$

$$R_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1.3)$$

$$R = R_{x,\alpha} R_{y,\phi} R_{z,\theta} . \quad (1.4)$$

Після цього отримують матриці перетворень, так, положення i -ї ланки описується через матрицю:

$$T = \begin{bmatrix} R_{3 \times 3} & p_{3 \times 1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \text{Rotation matrix} & \text{position vector} \\ \text{perspective transformation} & \text{scaling} \end{bmatrix}. \quad (1.5)$$

А загальна матриця перетворення для n -ланкового маніпулятора:

$$T_i^{i-1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & r_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & r_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.6)$$

Таким чином, підставивши у цю матрицю параметри Денавіта-Хартенберга (Табл. 1.1), її елементи будуть вказувати на орієнтацію та положення робочого органа:

$$T = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1.7)$$

де $\bar{n}$ – вектор нормалі до робочого органа, $\bar{s}$ – дотичний вектор робочого органа, $\bar{a}$ – вектор підходу робочого органа, $\bar{p}$ – вектор положення робочого органа.

Таблиця 1.1 - Параметри Денавіта-Хартенберга

θ_i	обертання навколо осі Z_{i-1}
d_i	зміщення вздовж осі Z_{i-1}
$a_i(r_i)$	довжина ланки вздовж осі x_i
α_i	нахил між осями Z_{i-1} і Z_i .

- Тригонометричний – вирішення тригонометричних рівнянь.

Для тригонометричного методу робота поміщають у 2-вимірну систему координат. За допомогою тригонометричних функцій та відомих значень кутів в суглобах, можна знайти положення робочого органа.

Щоб знайти координати робочого органа, необхідно визначити проекцію кожної ланки на площини XY , YZ (рис. 1.16). Це можна зробити з допомогою обчислення проєкцій ланок на кожну з цих площин, l_1 – може обертатися тільки навколо осі z_0 , тобто нахил відносно інших осей абсолютних координат завжди буде дорівнювати 0, а значить і проєкція l_1 на ці осі буде дорівнювати нулю.

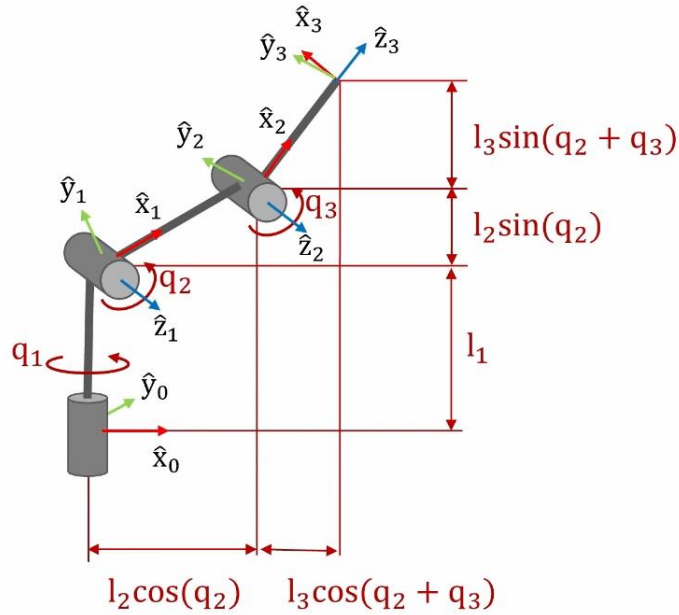


Рис. 1.16. Кінематична модель послідовного маніпулятора

Обчислюємо висоту робочого органа:

$$z' = l_1, \quad (1.8)$$

$$z'' = l_2 \sin q_2, \quad (1.9)$$

$$z''' = l_3 \sin(q_2 + q_3), \quad (1.10)$$

$$z = z' + z'' + z''' . \quad (1.11)$$

Також розраховується дистанцію робочого органа на площині XY :

$$d' = l_2 \cos q_2, \quad (1.12)$$

$$d'' = l_3 \cos(q_2 + q_3), \quad (1.13)$$

$$d = x' + x''. \quad (1.14)$$

Обчисливши проєкції d на вісь $\hat{x}_0$ та $\hat{y}_0$, отримуються точки (рис. 1.17):

$$x = d \cos q_1, \quad (1.15)$$

$$y = d \sin q_1. \quad (1.16)$$

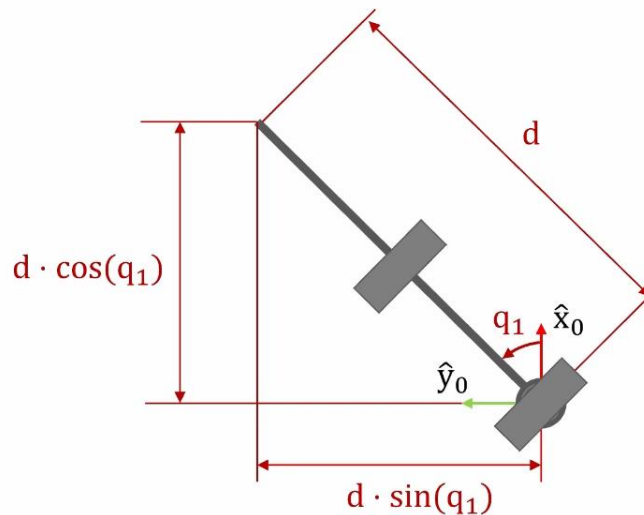


Рис. 1.17. Кінематична модель послідовного маніпулятора на площині XY

1.4. Висновки за розділом

Вибір між послідовною та паралельною конфігурацією при проєктуванні робота-маніпулятора є ключовим рішенням, що визначає його функціональні можливості, експлуатаційні характеристики та сфери застосування. Правильно вибрана конфігурація дозволяє оптимізувати роботу маніпулятора відповідно до конкретних задач та умов експлуатації.

Комплексний аналіз умов експлуатації, вимог до точності, динамічних характеристик, вантажопідйомності та економічних показників дозволяє обрати оптимальну конфігурацію маніпулятора, яка забезпечить найкраще співвідношення експлуатаційних характеристик та вартості для конкретного застосування.

РОЗДІЛ 2

РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ РОБОТА-МАНІПУЛЯТОРА

2.1. Кінематичний аналіз механізму паралелограмного типу

Структурний аналіз механізму. Механізм, що досліджується у даній роботі, фактично є однією з конфігурацій Five-Bar Linkage, яка складається з п'яти ланок, з'єднаних між собою п'ятьма обертальними кінематичними парами. Точки A і E представляють собою нерухомі опори (стояки), закріплені на фіксованій відстані l_5 одна від одної. Ланки l_1 і l_4 є вхідними (ведучими) ланками, які приводяться в рух двома окремими приводами у точках A і E відповідно. Ланки l_2 і l_3 є шатунами, які з'єднуються в точці C , l_6 – є продовженням ланки l_3 до точки F , яка є робочим органом маніпулятора.

Структура механізму має такі геометричні особливості:

- ланки l_1 і l_4 кріпляться до нерухомих опор в точках A і E відповідно, і можуть обертатися навколо цих точок;
- ланки l_2 і l_3 з'єднуються з ланками l_1 і l_4 в точках B і D відповідно;
- ланки l_2 і l_3 з'єднуються між собою в точці C ;
- відрізок CF – це одна ланка що розділена з'єднанням D на l_3 та l_6 ;
- усі кінематичні пари є обертальними (клас V).

Для даного механізму, згідно з критерієм Грюблера-Куцбаха [5], кількість ступенів вільності визначається як:

$$DOF = 3(n - 1) - 2p_5 - p_4, \quad (2.1)$$

де n – кількість ланок, p_5 - кількість кінематичних пар V класу, p_4 - кількість кінематичних пар IV класу.

У нашому випадку: $n = 5$ (п'ять рухомих ланок), $p_5 = 5$ (п'ять обертальних кінематичних пар), $p_4 = 0$, тому

$$DOF = 3(5 - 1) - 2 \times 5 - 0 = 2. \quad (2.2)$$

Таким чином, механізм має 2 ступені вільності. При цьому, механізм встановлено на платформу що може обертатися, тому попереднє твердження справедливе тільки щодо механізму що розглядається, сам маніпулятор має 3 ступені вільності.

Кінематична схема механізму. На рис. 2.1 представлена кінематична схема досліджуваного паралельного маніпулятора з позначенням усіх ланок та кінематичних пар. Механізм складається з п'яти ланок з довжинами l_1, l_2, l_3, l_4, l_5 та l_6 , де l_5 – фіксована відстань між нерухомими опорами A і E , а l_6 – продовження ланки l_3 до точки F .

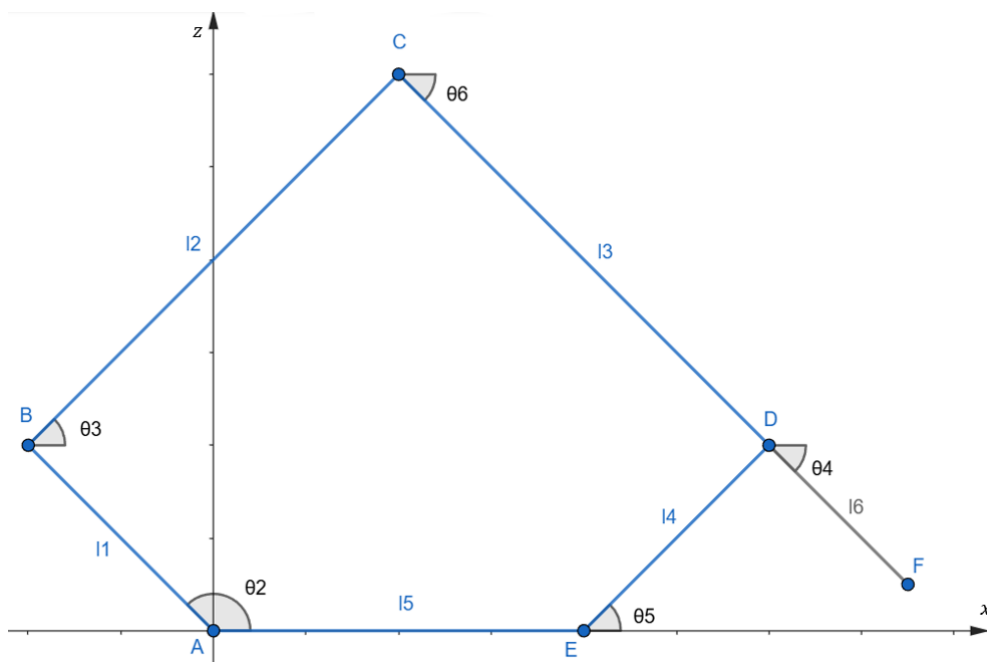


Рис. 2.1. Кінематична модель маніпулятора що досліджується на площині XZ (пропорції спотворено для демонстрації геометричних особливостей конфігурації)

Для аналізу приймаємо наступні параметри механізму:

- $l_0 = 135$ мм – висота платформи на яку встановлений механізм;
- $l_1 = 40$ мм - довжина ланки 1 (AB);
- $l_2 = 160$ мм - довжина ланки 2 (BC);

- $l_3 = 40$ мм - довжина ланки 3 (CD);
- $l_4 = 160$ мм - довжина ланки 4 (DE);
- $l_5 = 0$ мм - фіксована відстань між опорами A і E ;
- $l_6 = 160$ мм – довжина ланки 6 (DF).

Кут θ_1 визначає поворот всього механізму навколо осі z . Кути θ_2 і θ_5 визначають положення ведучих ланок 1 і 4 відносно осі x . Кути θ_3 і θ_4 визначають положення шатунів 2 і 3 відносно осі x .

Визначення положень ланок механізму. Пряма задача кінематики. Пряма кінематична задача полягає у визначенні положення робочої точки $F(x, y, z)$ при заданих кутах θ_2 та θ_5 ведучих ланок, так кута повороту бази θ_1 .

Введемо локальну систему координат з початком у точці A . Положення точки E визначається як: $(l_5, 0)$.

Спустивши перпендикуляри на осі x та z , розраховуємо координати для точок B, D :

$$x_B = l_1 \cos \theta_2, \quad (2.3)$$

$$z_B = l_1 \sin \theta_2, \quad (2.4)$$

$$x_D = l_5 + l_4 \cos \theta_5, \quad (2.5)$$

$$z_D = l_4 \sin \theta_5. \quad (2.6)$$

Знаючи координати B та D , можна розрахувати положення C, F :

$$x_C = x_B + l_2 \cos \theta_3, \quad (2.7)$$

$$z_C = z_B + l_2 \sin \theta_3, \quad (2.8)$$

$$x_F = x_D + l_2 \cos \theta_6, \quad (2.9)$$

$$z_C = z_F + l_2 \sin \theta_6. \quad (2.10)$$

На основі виконаних обчислень, та властивостей паралелограма ($l_1 \parallel l_3$, та $l_2 \parallel l_4$), отримуємо L – проєкцію EF на площину XY (рис. 2.2):

$$\begin{aligned} L &= l_4 \cos \theta_5 + l_6 \cos \theta_4 = l_4 \cos(\theta_5) + l_6 \cos(\theta_2 - \pi) = \\ &= l_4 \cos \theta_5 + l_6(-\cos \theta_2). \end{aligned} \quad (2.11)$$

Таким чином, з'являється можливість розрахувати координати точки F , враховуючи висоту платформи l_0 :

$$x = L \cos \theta_1, \quad (2.12)$$

$$y = L \sin \theta_1, \quad (2.13)$$

$$\begin{aligned} z &= l_0 + l_4 \sin \theta_5 + l_6 \sin \theta_6 = l_0 + l_4 \sin \theta_5 + l_6 \sin(\theta_2 + \pi) = \\ &= l_0 + l_4 \sin \theta_5 + l_6(-\sin \theta_2). \end{aligned} \quad (2.14)$$

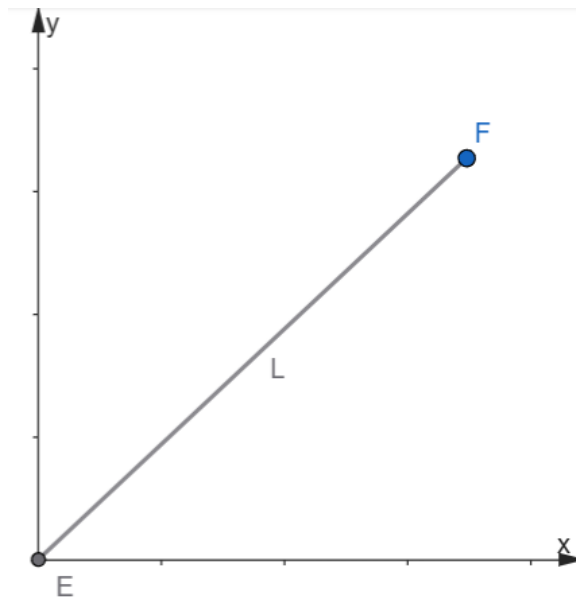


Рис. 2.2. Спрощена кінематична модель паралельного маніпулятора на площині XY

Обернена задача кінематики. Обернена кінематична задача полягає у визначенні кутів θ_2 та θ_5 ведучих ланок та кута θ_1 повороту бази, які забезпечують задане положення робочої точки $F(x, y, z)$.

Враховуючи що плечовий суглоб знаходиться вище осі x - переміщуємо абсолютну систему координат на $-l_0$ по осі z :

$$z = z - l_0. \quad (2.15)$$

Після чого одразу можна розрахувати кут повороту платформи через координати (x, y) :

$$\theta_1 = \text{atan2}(x, y). \quad (2.16)$$

Розраховуємо проекцію відрізка EF що падає на площину XY :

$$L = \sqrt{x^2 + y^2}. \quad (2.17)$$

Для спрощення подальших розрахунків, уявляємо що механізм $l_1, l_2, l_3, l_4, l_5, l_6$ – планарний (рис. 2.3). За відомим значенням L – можна розрахувати довжину EF :

$$h = \sqrt{L^2 + z^2}. \quad (2.18)$$

Після чого обчислюється кут що утворюється між h та площиною XY

$$\alpha = \text{atan} \frac{z}{L}. \quad (2.19)$$

Та кут що утворюється між l_4 і h :

$$\beta = \text{acos} \left(\frac{\frac{h}{2}}{l_4} \right). \quad (2.20)$$

За (2.18) та (2.19) можна обчислити кут θ_5 що лежить між l_4 і площиною XY , та зовнішній кут q що лежить між l_6 та віссю x в точці D :

$$\theta_5 = \alpha + \beta, \quad (2.21)$$

$$q = \alpha - \beta. \quad (2.22)$$

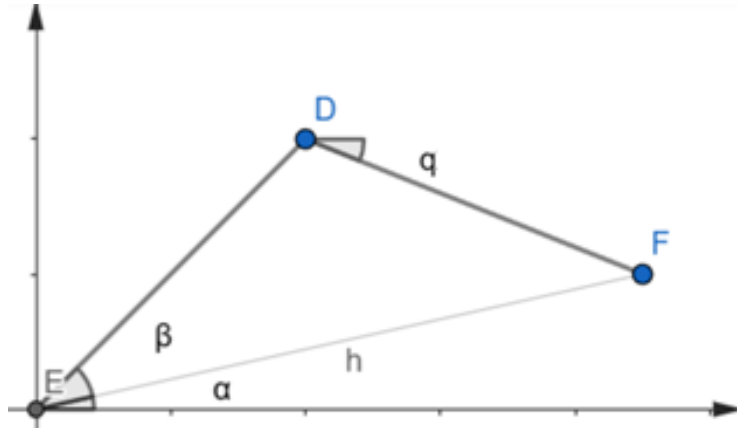


Рис. 2.3. Спрощена кінематична модель паралельного маніпулятора на площині яку утворює механізм $ABCDEF$

Далі, обчислюємо положення точки C на площині в якій лежить механізм. Таким чином, задача приводиться до пошуку кута θ_2 планарного Five-Bar Manipulator, з робочим органом у точці C :

$$x_C = L - (l_6 + l_3) \cos q, \quad (2.23)$$

$$z_C = z - (l_6 + l_3) \sin q. \quad (2.24)$$

Позначимо відрізок $EC = d$:

$$d = \sqrt{x_C^2 + z_C^2}. \quad (2.25)$$

З теореми косинусів випливає що:

$$l_2^2 = l_1^2 + d^2 - 2l_1d \cos p, \quad (2.26)$$

де p – кут ABC .

З (2.24) визначимо p :

$$\cos p = \frac{l_1^2 - l_2^2 + d^2}{2l_1d}. \quad (2.27)$$

$$p = \arccos(\cos p). \quad (2.28)$$

Тепер, можемо вивести кут θ_2 :

$$\theta_2 = p + m, \quad (2.29)$$

де $m = \text{atan2}(xC, zC)$.

Цікаво зазначити, що в [6] пропонується пошук θ_2 таким чином:

$$\theta_2 = 2 \text{atan} \left(\frac{-b - \sqrt{a^2 + b^2 - c^2}}{-a - c} \right), \quad (2.30)$$

де a – положення C на осі абсцис, b – положення на осі ординат (z), а c – проєкція d на вісь, що утворюється прямою яка проходить через точки A і B :

$$c = d \cos p = \frac{l_1^2 - l_2^2 + a^2 + b^2}{2l_1}. \quad (2.31)$$

У порівнянні з (2.23-27), (2.28) проста в обчисленні та дає більш точний результат.

Аналіз швидкостей. Для визначення швидкостей використовуємо матрицю Якобі для маніпулятора. Приймаємо, що ланки 1 і 4 обертаються з постійними кутовими швидкостями $\omega_1 = \omega_2 = \omega_5 = 5,24$ рад/с [7].

Диференціюємо рівняння кінематики (2.10-12):

$$\frac{dx}{d\theta_1} = -L \sin \theta_1. \quad (2.32)$$

$$\frac{dy}{d\theta_1} = L \cos \theta_1. \quad (2.33)$$

$$\frac{dz}{d\theta_1} = 0. \quad (2.34)$$

$$\frac{dx}{d\theta_2} = -\cos \theta_1 \times l_6 \sin \theta_2. \quad (2.35)$$

$$\frac{dy}{d\theta_2} = -\sin \theta_1 \times l_6 \sin \theta_2. \quad (2.36)$$

$$\frac{dz}{d\theta_2} = l_6 \times -\cos \theta_1. \quad (2.37)$$

$$\frac{dx}{d\theta_5} = -\cos \theta_1 \times l_4 \sin \theta_5. \quad (2.38)$$

$$\frac{dy}{d\theta_5} = -\sin \theta_1 \times l_4 \sin \theta_2. \quad (2.39)$$

$$\frac{dz}{d\theta_5} = l_4 \cos \theta_5. \quad (2.40)$$

Отже, матриця Якобі:

$$J = \begin{pmatrix} \frac{dx}{d\theta_1} & \frac{dx}{d\theta_2} & \frac{dx}{d\theta_5} \\ \frac{dy}{d\theta_1} & \frac{dy}{d\theta_2} & \frac{dy}{d\theta_5} \\ \frac{dz}{d\theta_1} & \frac{dz}{d\theta_2} & \frac{dz}{d\theta_5} \end{pmatrix} = \begin{pmatrix} -L \sin \theta_1 & -\cos \theta_1 l_6 \sin \theta_2 & -\cos \theta_1 l_4 \sin \theta_5 \\ L \cos \theta_1 & -\sin \theta_1 l_6 \sin \theta_2 & -\sin \theta_1 l_4 \sin \theta_2 \\ 0 & -\cos \theta_1 l_6 & l_4 \cos \theta_5 \end{pmatrix}. \quad (2.41)$$

Тепер, щоб знайти швидкості у просторі узагальнених координат, використаємо рівняння швидкостей:

$$\dot{\mathbf{x}} = J\dot{\mathbf{q}}, \quad (2.42)$$

де:

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} \quad \dot{\mathbf{q}} = \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_5 \end{pmatrix} = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_5 \end{pmatrix}. \quad (2.43)$$

Аналіз сингулярностей механізму. Сингулярність - це особлива точка в робочому просторі робота, яка призводить до втрати роботом одного або більше ступенів свободи. Коли центральна точка робочого органу робота переміщується в сингулярність або поблизу неї, робот припиняє рух або рухається в несподіваному напрямку. Джерелами сингулярності є:

- фізичні обмеження роботів як фізичних пристроїв. Наприклад, кожен з двигунів робота має максимальну швидкість.
- Рухами робота керують алгоритми та математика, які не мають фізичних обмежень. Наприклад, з математичної точки зору, допустимою є швидкість повороту ланок, що дорівнює «нескінченності».

Конфлікт між цими двома фактами може спричинити безліч проблем під час програмування робота. Якщо не враховувати це під час проектування, алгоритм керування може наказати двигунам робота виконати фізично неможливий рух. Це, по суті, відбувається, коли маніпулятор стикається з сингулярністю.

Сингулярності існують для певної конфігурації робота, якщо визначник матриці Якобі - Якобіан, дорівнює нулю:

$$Jacobian = \det(J). \quad (2.44)$$

Якщо $\det(J) > 0$ то механізм знаходиться в конфігурації "ліктем вгору", а при $\det(J) < 0$ то механізм знаходиться в конфігурації "ліктем вниз".

Найчастіше визначають такі типи сингулярностей:

- сингулярність зап'ястя виникає, коли ланки зап'ястя вирівнюються так, що один із суглобів втрачає свою ефективність (виникає у роботів з трьома осями обертання зап'ястя). Коли друга та третя осі зап'ястя вирівнюються, кінцева точка втрачає ступінь свободи.
- Ліктьові сингулярності виникають, коли лікоть повністю випрямлений або зігнутий до крайнього положення. Коли лікоть повністю розігнутий, і система не може визначити, чи згинати його вперед чи назад.
- Сингулярність плеча виникає, коли вісь обертання першого суглоба (плеча) збігається з віссю передпліччя або всі осі обертання суглобів плеча вирівнюються. Якщо перша ланка робота (від основи до плеча) повністю

витагнута, маніпулятор може втратити можливість рухати кінцеву точку у певних напрямках.

Інший спосіб класифікувати сингулярності роботів - розділити їх на дві групи:

- внутрішні сингулярності робочої області. Цей тип сингулярностей виникає, коли центральна точка інструменту робота потрапляє в межі робочого простору робота. Вони виникають, коли дві або більше суглобових осей робота вирівнюються одна з одною. Сингулярності зап'ястя і плеча є прикладами внутрішніх сингулярностей робочого простору. Їх часто важко уникнути, оскільки не одразу зрозуміло, де вони знаходяться в робочій зоні робота.
- Сингулярності на межі робочого простору. Цей тип сингулярностей виникає на межі робочої зони робота. Кожного разу, коли центральна точка інструменту робота наближається до межі, існує ризик потрапляння в сингулярність. Сингулярності ліктя є прикладом сингулярності на межі робочого простору.

Отже, Якобіан:

$$\begin{aligned}
 \det(J) &= \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = \\
 &= a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - \\
 &\quad - a_{13}a_{22}a_{31} - a_{12}a_{21}a_{33} - a_{11}a_{23}a_{32}, \tag{2.45} \\
 \det(J) &= (-L \sin \theta_1)(-\sin \theta_1 l_6 \sin \theta_2)(l_4 \cos \theta_5) + \\
 &\quad + (-\cos \theta_1 l_6 \sin \theta_2)(-\sin \theta_1 l_4 \sin \theta_2)(0) + \\
 &\quad + (-\cos \theta_1 l_4 \sin \theta_5)(L \cos \theta_1)(-\cos \theta_1 l_6) - \\
 &\quad - (-\cos \theta_1 l_4 \sin \theta_5)(-\sin \theta_1 l_6 \sin \theta_2)(0) - \\
 &\quad - (-\cos \theta_1 l_6 \sin \theta_2)(\cos \theta_1)(l_4 \cos \theta_5) -
 \end{aligned}$$

$$-(-L \sin \theta_1)(-\sin \theta_1 l_4 \sin \theta_2)(-\cos \theta_1 l_6). \quad (2.46)$$

Прибираємо члени що рівні нулю:

$$\begin{aligned} \det(J) = & (-L \sin \theta_1)(-\sin \theta_1 l_6 \sin \theta_2)(l_4 \cos \theta_5) + \\ & + (-\cos \theta_1 l_4 \sin \theta_5)(L \cos \theta_1)(-\cos \theta_1 l_6) - \\ & - (-\cos \theta_1 l_6 \sin \theta_2)(\cos \theta_1)(l_4 \cos \theta_5) - \\ & - (-L \sin \theta_1)(-\sin \theta_1 l_4 \sin \theta_2)(-\cos \theta_1 l_6). \end{aligned} \quad (2.47)$$

Визначення робочої зони маніпулятора. Оцінюючи робота з точки зору його робочого простору, ми зазвичай обчислюємо об'єм або область простору, яка може бути досягнута його робочим органом, з урахуванням геометричних обмежень робота, тобто обмежень суглобів, діапазону спрацьовування, самозіткнень, сингулярних конфігурацій.

Тобто, робочий простір W (рис. 2.5-2.7) складається з множини всіх можливих точок P , які одночасно належать множині допустимих положень суглобів G_j та знаходяться в діапазоні приводу G_r , з яких виключено всі точки, що потрапляють у множину самозіткнень G_c та множину сингулярностей G_s (рис. 2.4):

$$W = (P \cap G_j \cap G_r) \setminus (G_c \cup G_s). \quad (2.48)$$

Отже, в нашому випадку, треба лишити тільки положення при $(\det(J) \neq 0)$, адже конфігурація «ліктем вниз» механічно недоступна. Також, необхідно вказати діапазони зміни кутів $\theta_1, \theta_2, \theta_5$ під час обчислення алгоритму прямої кінематики, разом з встановленням діапазону допустимої відстані EF, і L – таким чином запобігаючи колізіям.

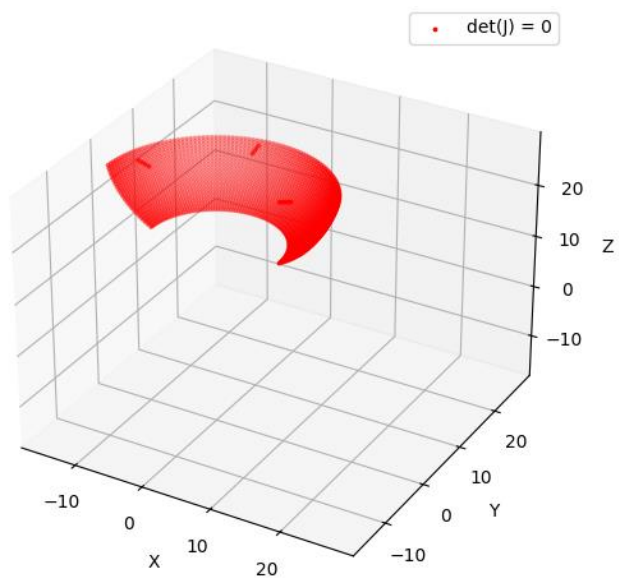


Рис. 2.4. Точки у просторі що відповідають сингулярним положенням

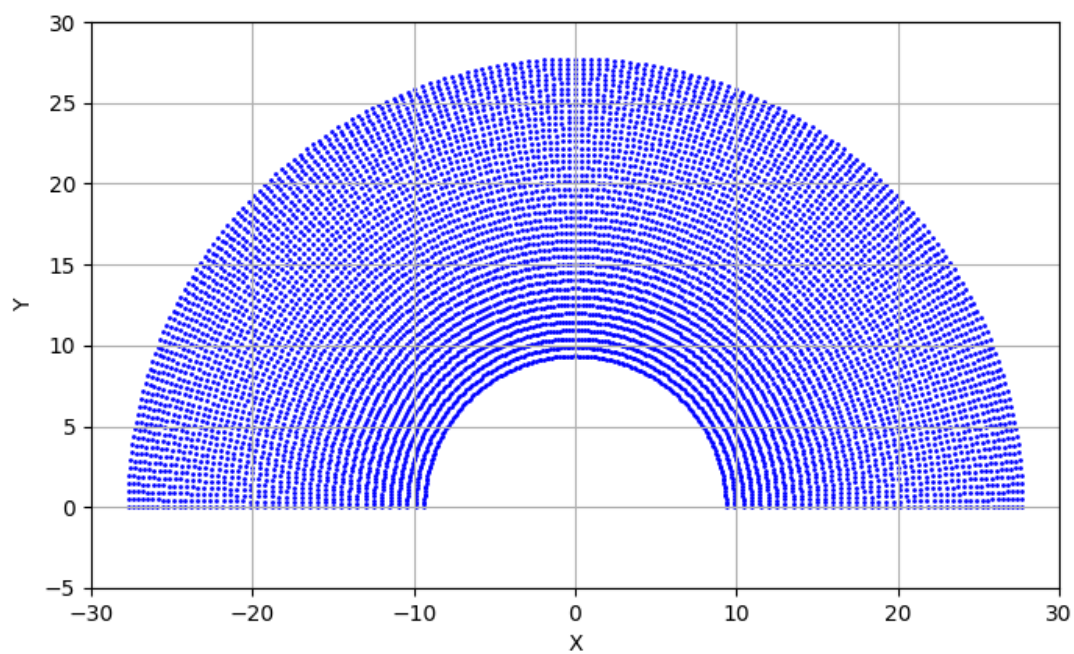


Рис. 2.5. Усі доступні положення при застосуванні обмежень (XY)

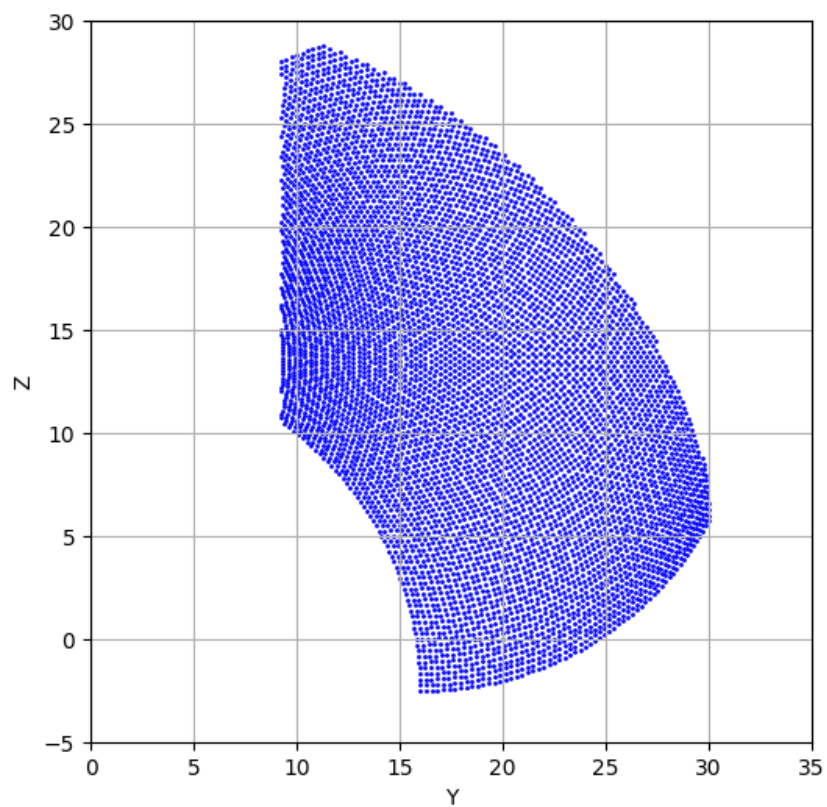


Рис. 2.6. Усі доступні положення при застосуванні обмежень (YZ)

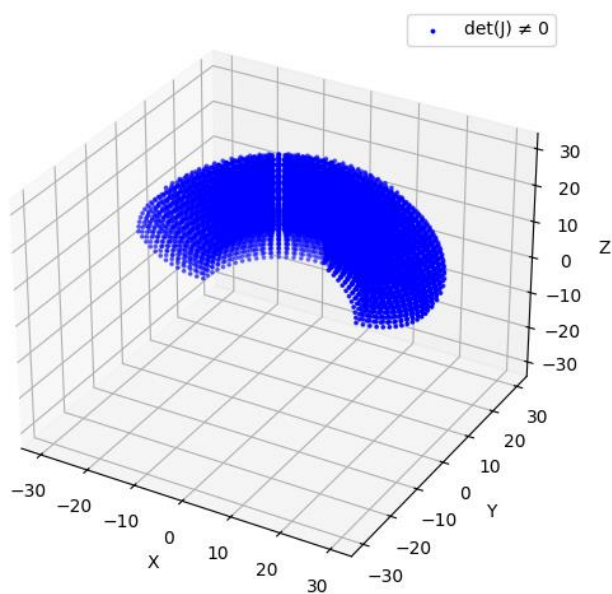


Рис. 2.7. Усі доступні положення при застосуванні обмежень
(ізометрична проєкція)

2.2. Математичне моделювання планування траєкторії маніпулятора

Планування траєкторії – одна з проблем, які виникають під час планування руху. Типова ієрархія планування руху виглядає наступним чином:

- планування завдання - розробка набору цілей високого рівня, наприклад, «підійти і забрати об'єкт, що лежить перед вами».
- Планування шляху - створення можливого шляху від початкової точки до кінцевої точки. Шлях зазвичай складається з набору з'єднаних між собою маршрутних точок.
- Планування траєкторії - створення часового графіка проходження шляху з урахуванням таких обмежень, як положення, швидкість і прискорення.
- Дотримання траєкторії - після того, як вся траєкторія спланована, необхідна система керування, яка може виконати траєкторію досить точно.

Зазвичай, для автоматизації планування шляху використовують алгоритми пошуку. Найбільш розповсюджені з них – A^* (A-star), D^* (Dynamic A^*), алгоритм Дейкстри, генетичні алгоритми.

Припустимо, що набір точок шляху вже доступний, і ми хочемо згенерувати траєкторію для маніпулятора, який буде слідувати за цими точками у часі (тут, і далі у тексті використовується нормалізований час (2.51), та лінійна інтерполяція (2.53) – якщо не вказано інше).

Отже, постає вибір, у якому з просторів буде згенерована траєкторія: робочий простір (Task space) або простір зчленувань (Joint space), відмінності між якими описані в таблиці 2.1, та продемонстровані на рис. 2.8-2.12.

Таблиця 2.1 – Порівняння Task space та Joint space

	Простір зчленувань (Joint space)	Робочий простір (Task space)
Планується:	Значення кутів або положень приводів у кожен момент часу	Положення та орієнтація кінцевої точки в фізичному просторі у кожен момент часу
Інтерполяція	Між кутами суглобів	Між точками у фізичному просторі
Переваги	Простота реалізації, висока швидкість обчислень (обернена кінематика обчислюється тільки в двох точках). Рухи двигуна більш плавні.	Більший контроль над фактичною траєкторією кінцевої точки. Краща обробка перешкод та самозіткнень.
Недоліки	Проміжні точки не завжди враховуватимуть обмеження суглобів або перешкоди.	Вимагає обчислення оберненої кінематики для кожної точки шляху, що впливає на кількість обчислювальних ресурсів або час виконання програми. Рухи двигуна не завжди плавні.

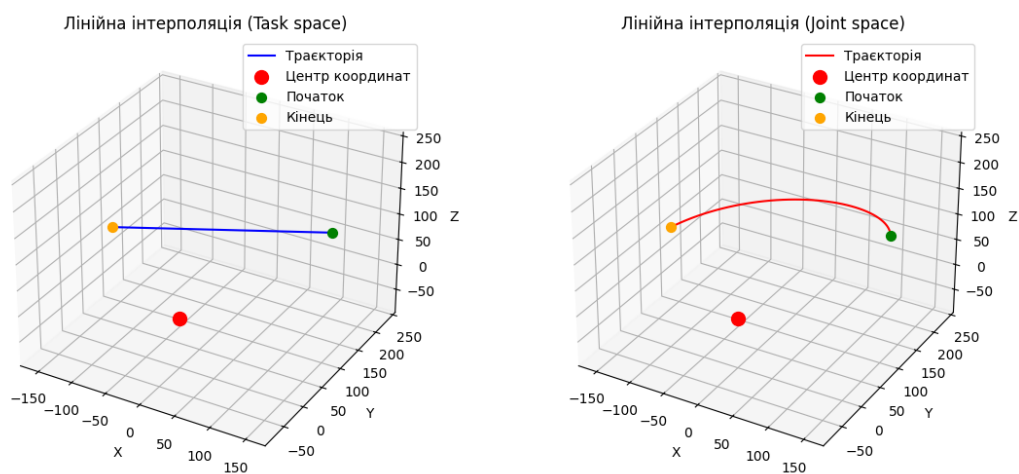


Рис. 2.8. Траєкторії інтерпольовані в Task space та Joint space

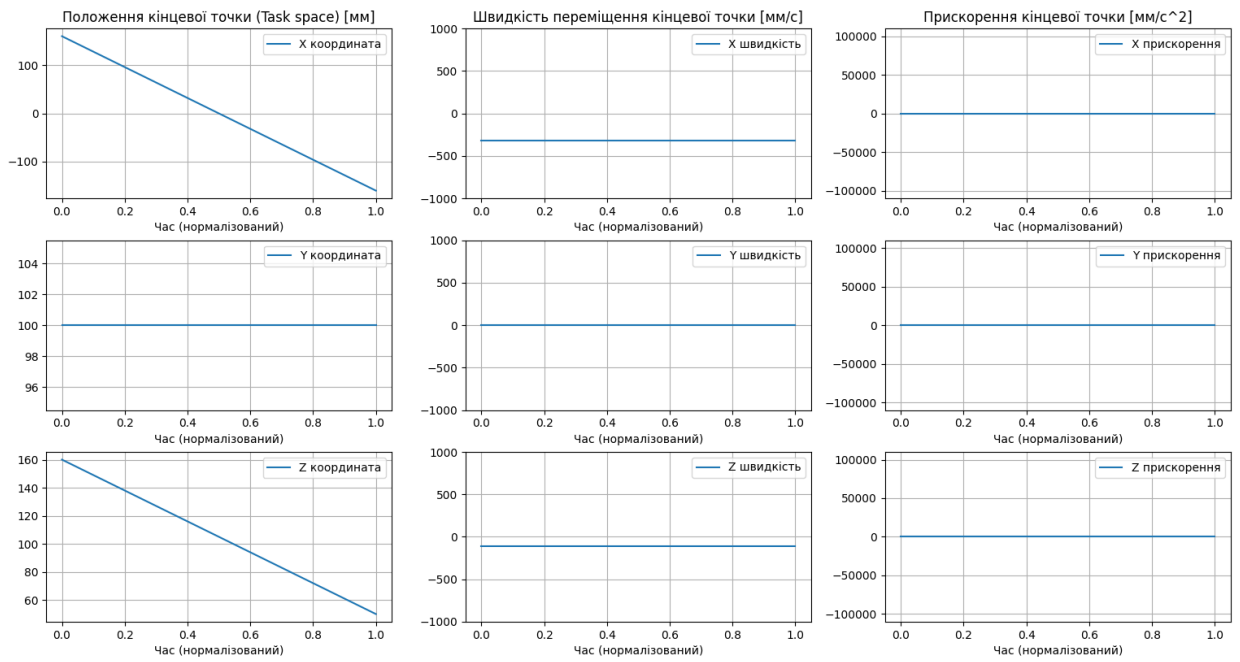


Рис. 2.9. Динамічні показники лінійного руху у Task space

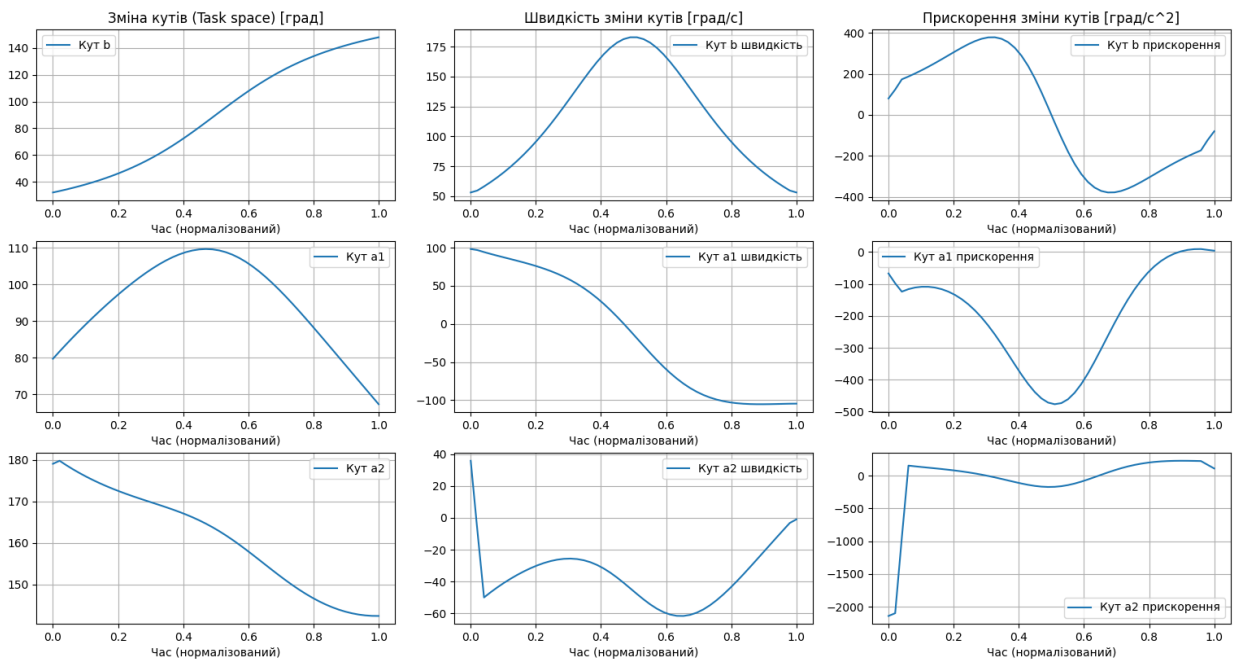


Рис. 2.10. Динамічні показники зміни кутів під час лінійного руху (Task space)

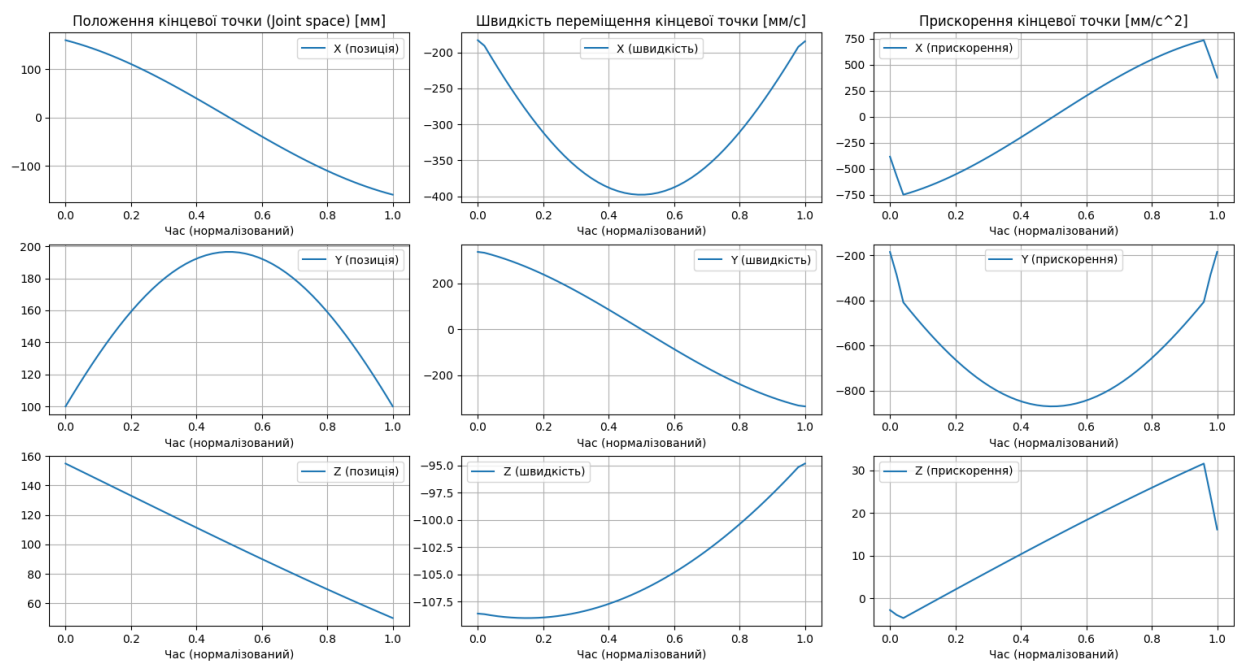


Рис. 2.11. Динамічні показники лінійного руху у Joint space

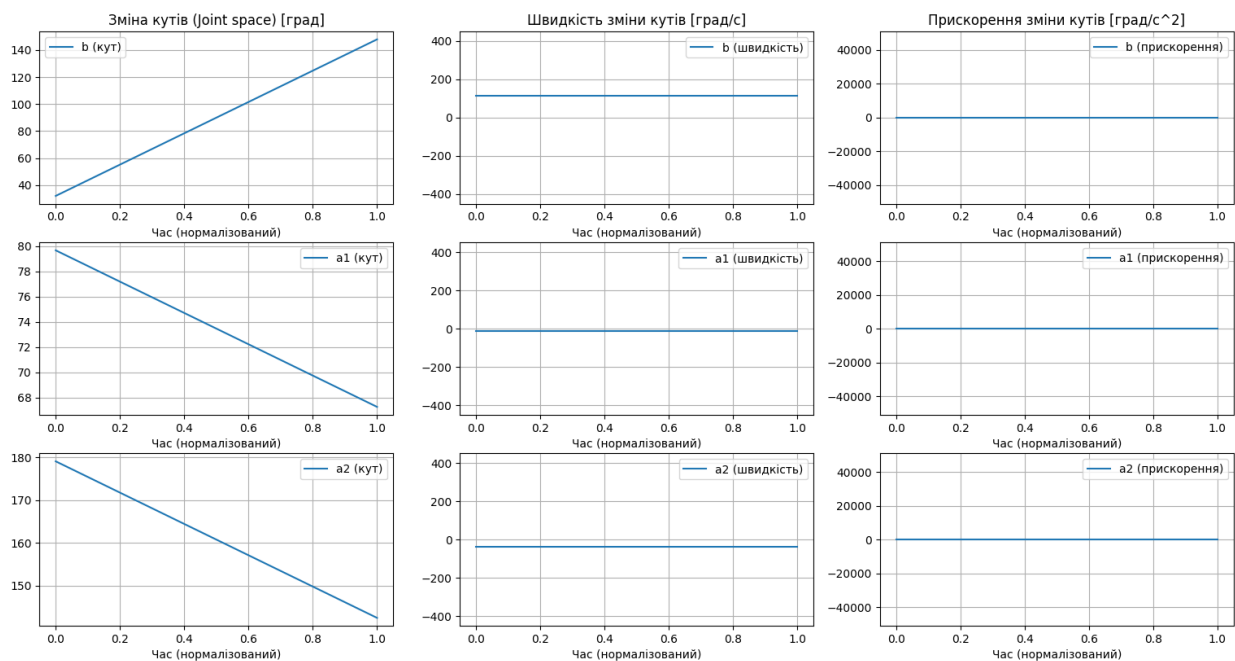


Рис. 2.12. Динамічні показники зміни кутів під час лінійного руху (Joint space)

На рис. 2.9-2.10, помітно що при постійному русі в Task space з нульовим прискоренням, зміни в Joint space можуть бути нелінійними. При переміщенні робота чи механізму в декартовому просторі з постійною лінійною швидкістю ($\dot{x} = \text{const}$), відповідні кутові швидкості суглобів визначаються через матрицю Якобі системи:

$$\dot{q} = J^{-1} \dot{x}. \quad (2.49)$$

На рис. 2.11-2.12 спостерігається симетрична ситуація. Якщо двигуни рухаються з постійними кутовими швидкостями ($\dot{q} = \text{const}$), рух кінцевої точки в декартовому просторі (Task space) може бути нерівномірним (2.40).

Проблема швидкості переміщення стає більш явною, коли маніпулятор проходить шлях що складається з декількох точок (рис. 2.13). В обох випадках (рис. 2.14-2.15) і (рис. 2.16-2.17), швидкість обертання двигунів змінюється ривком, тобто, кути мають дуже велике прискорення в цей момент.

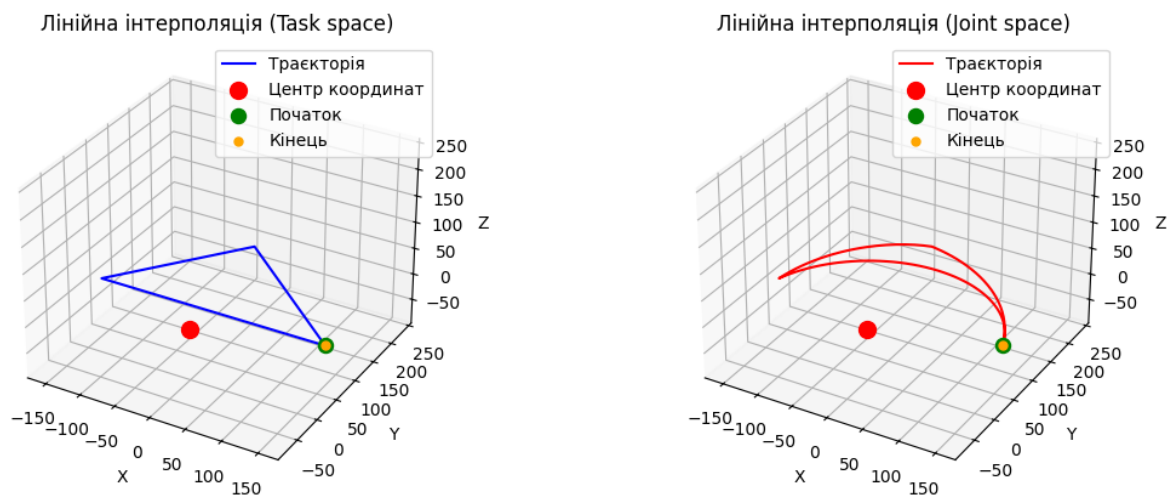


Рис. 2.13. Траєкторії, інтерпольовані в Task space та Joint space

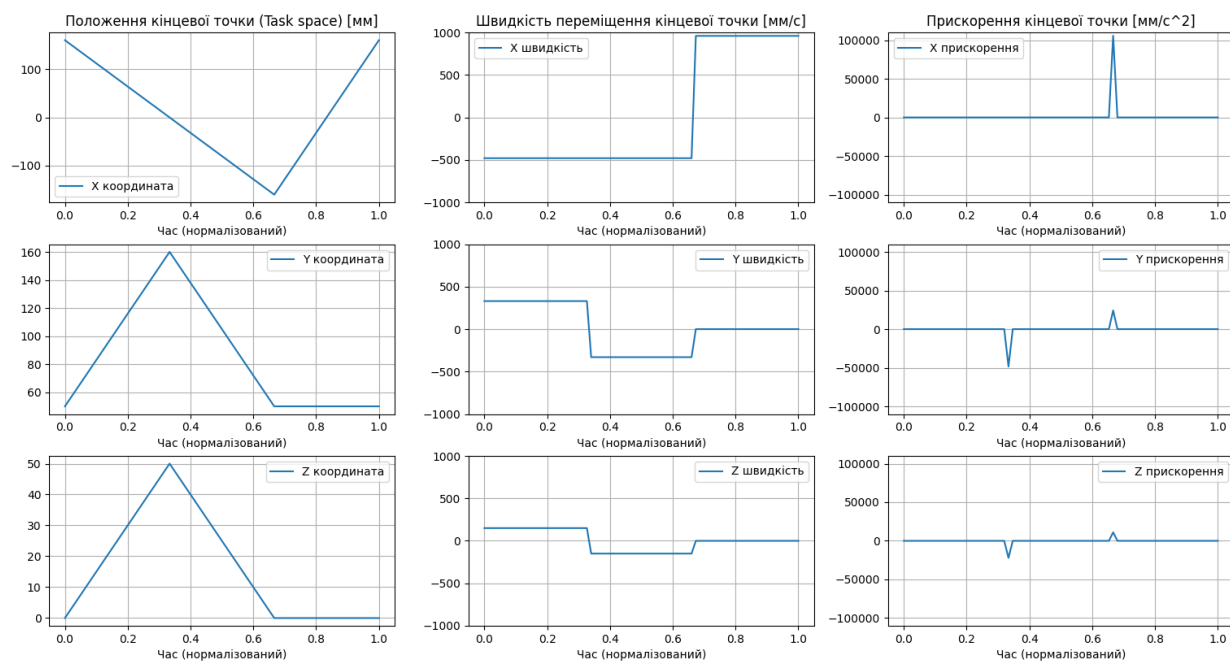


Рис. 2.14. Динамічні показники лінійного руху у Task space

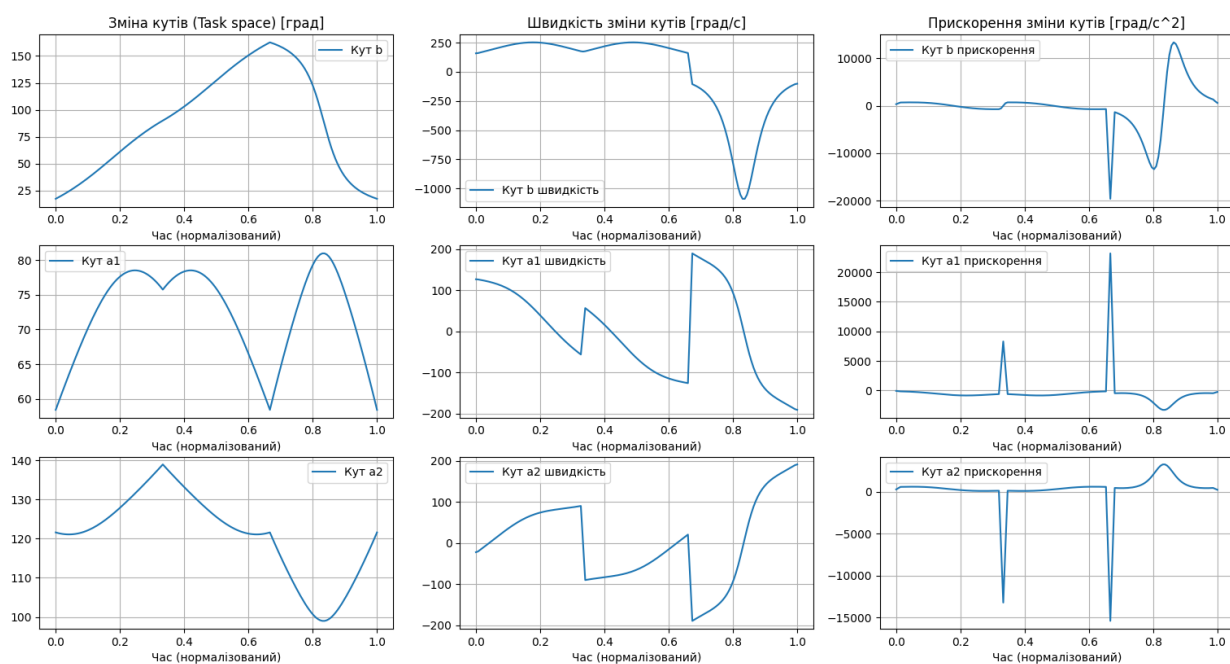


Рис. 2.15. Динамічні показники зміни кутів під час лінійного руху (Task space)

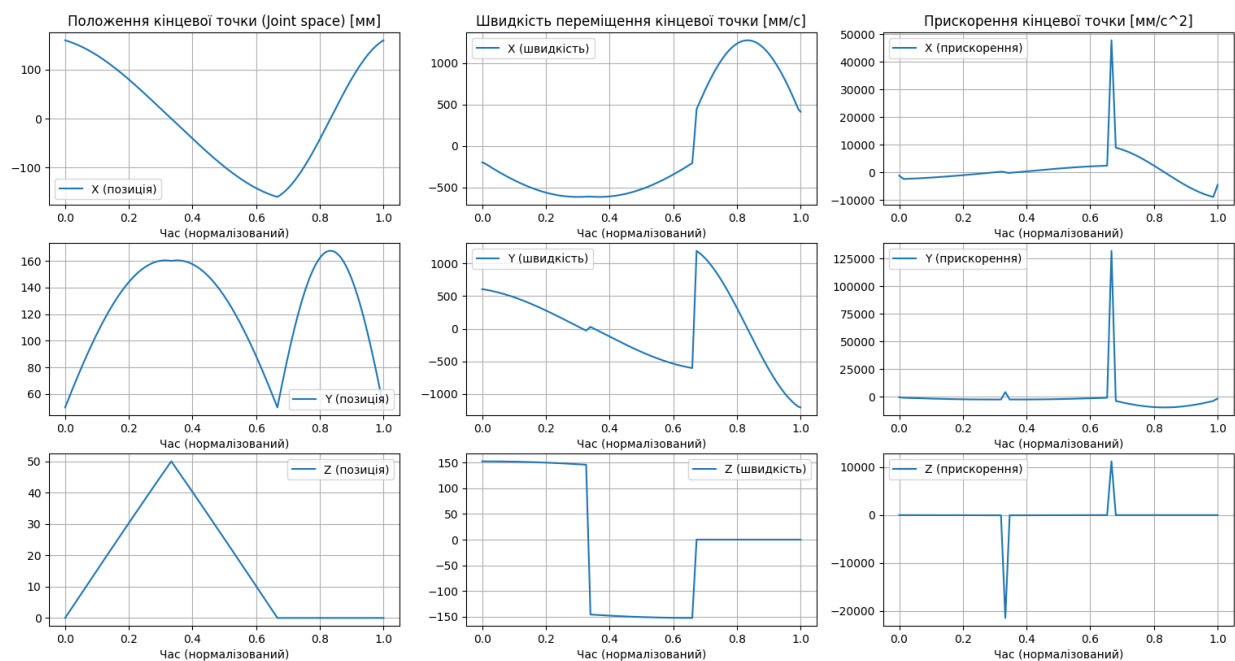


Рис. 2.16. Динамічні показники лінійного руху у Joint space

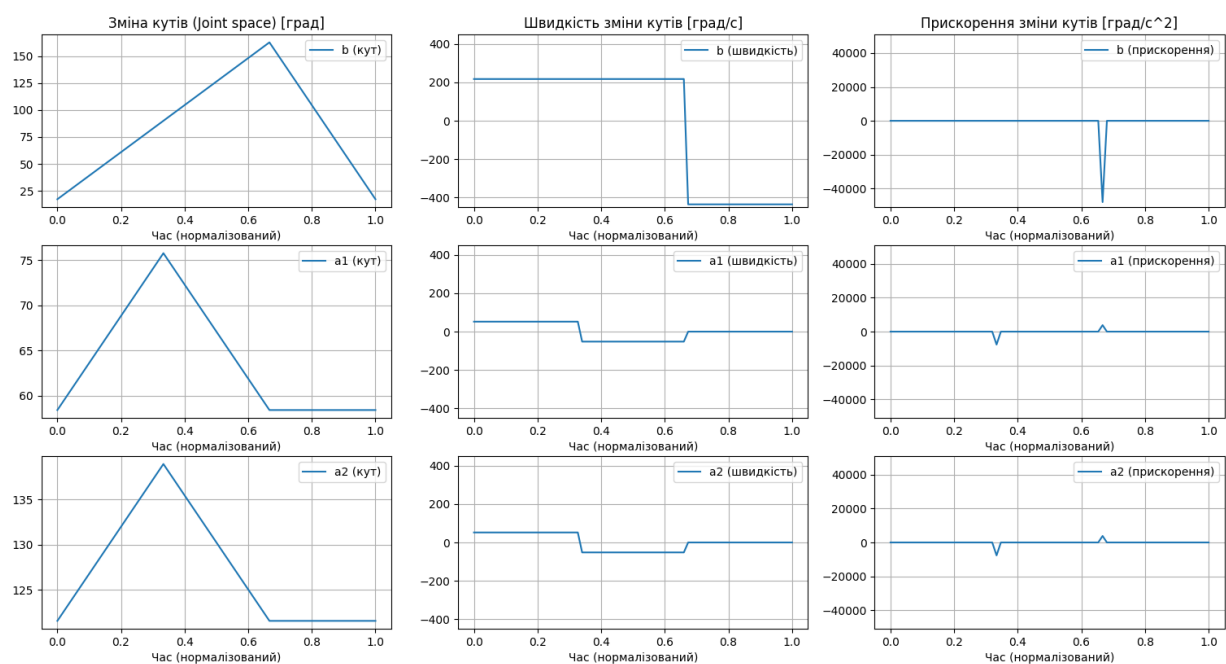


Рис. 2.17. Динамічні показники зміни кутів під час лінійного руху (Joint space)

Для забезпечення більш плавної зміни кутів, доцільно розглянути інші методи інтерполяції:

- трапецієподібні траєкторії швидкості - це кусково-задані траєкторії з постійним прискоренням, нульовим прискоренням і постійним уповільненням. Такі траєкторії досягаються за допомоги інтерполяції траєкторії по кривій що утворюється «лінійним сегментом з параболічною сумішшю» (LSPB) або сигмоїдальною функцією.

В моделюванні використовується крива LSPB – $s(t)$:

$$s(t) = \begin{cases} \frac{4.5t^2}{2}, & 0 \leq t \leq \frac{1}{3} \\ \frac{4.5t}{3} - 0.25, & \frac{1}{3} \leq t \leq \frac{2}{3} \\ 4.5 \left(t - \frac{t^2}{2} \right) - 1.25, & \frac{2}{3} \leq t \leq 1 \end{cases} \quad (2.50)$$

де t – нормалізований час:

$$t_i = \frac{i}{N-1}, \quad i \in \{0, 1, \dots, N-1\}, \quad (2.51)$$

а N – кількість точок з яких буде складатися крива;

- поліноміальна – досягається інтерполюванням траєкторії по кривій $s(t)$ що утворюється поліномом, найчастіше – третього або п'ятого порядків.

$$s(t) = 3t^2 - 2t^3. \quad (2.52)$$

Для обох випадків, $s(t)$ підставляється у функцію лінійної інтерполяції p_i :

$$p_i = p_{start} + s(t_i)(p_{end} - p_{start}), \quad (2.53)$$

де p_{start} , p_{end} – початкова та кінцева точка траєкторії.

- сплайни – також є кусковими комбінаціями поліномів, але на відміну від поліноміальних траєкторій, які є поліномами в часі (тобто один поліном для кожного відрізка), сплайни є поліномами в просторі, які можна

використовувати для створення складних форм. Часовий аспект полягає в тому, щоб слідувати за отриманими сплайнами з рівномірною швидкістю.

В моделюванні використовується Catmull-Rom сплайн:

$$p_i(t) = c_0 p_{i-1} + c_1 p_i + c_2 p_{i+1} + c_3 p_{i+2}, \quad (2.54)$$

де p_i – послідовність точок (початкова, контрольні та кінцева точки), а:

$$c_0 = -\alpha t^3 + 2\alpha t^2 - \alpha t, \quad (2.55)$$

$$c_1 = (2 - \alpha)t^3 + (\alpha - 3)t^2 + 1, \quad (2.56)$$

$$c_2 = (\alpha - 2)t^3 + (3 - 2\alpha)t^2 + \alpha t, \quad (2.57)$$

$$c_3 = \alpha t^3 - \alpha t^2, \quad (2.58)$$

де α – параметр натягнутості (чим більше значення – тим менш натягнутий сплайн).

Порівняння траєкторій руху робочого органу з використанням різних інтерполяцій, та їх динамічні показники представлено на рис. 2.19-2.20. Траєкторії у декартовому просторі майже ідентичні, особливо, коли сплайн сильно натягнутий ($\alpha \leq 0.1$). Профілі швидкості кутів теж мають незначні відмінності, але прискорення кутів помітно відрізняються: при застосуванні LSPB помітно що воно змінюється рівнями (майже так само як і прискорення положень на рис. 2.18)

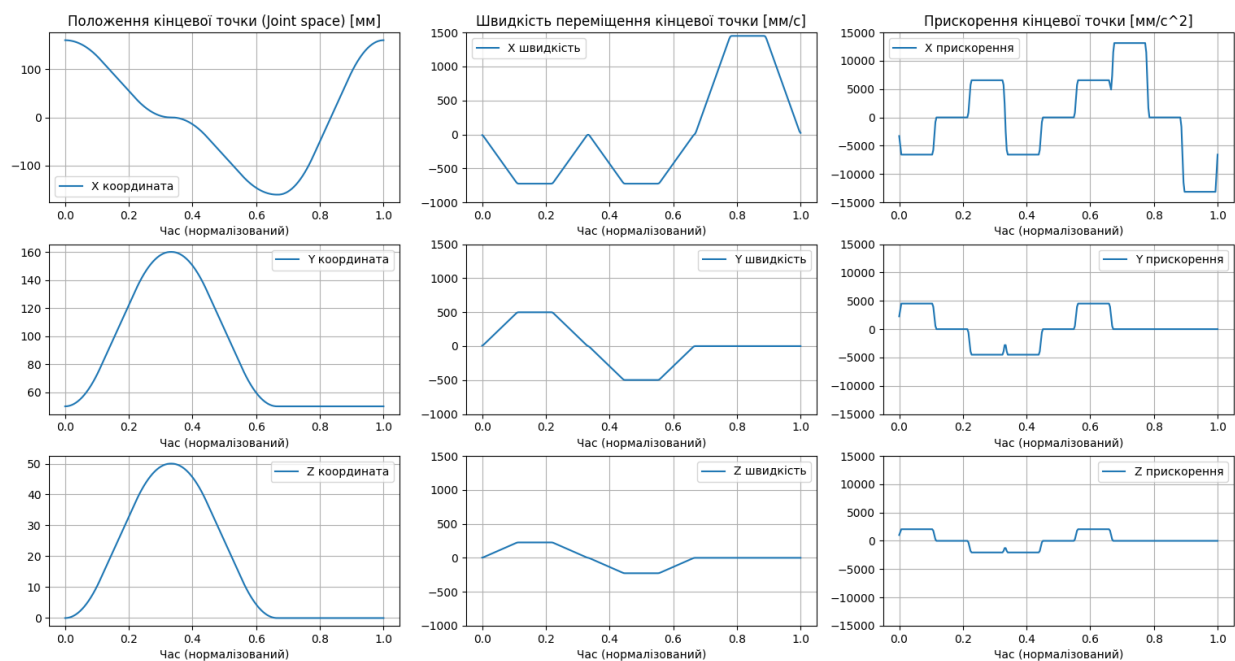


Рис. 2.18. Трапецієподібні траєкторії швидкості

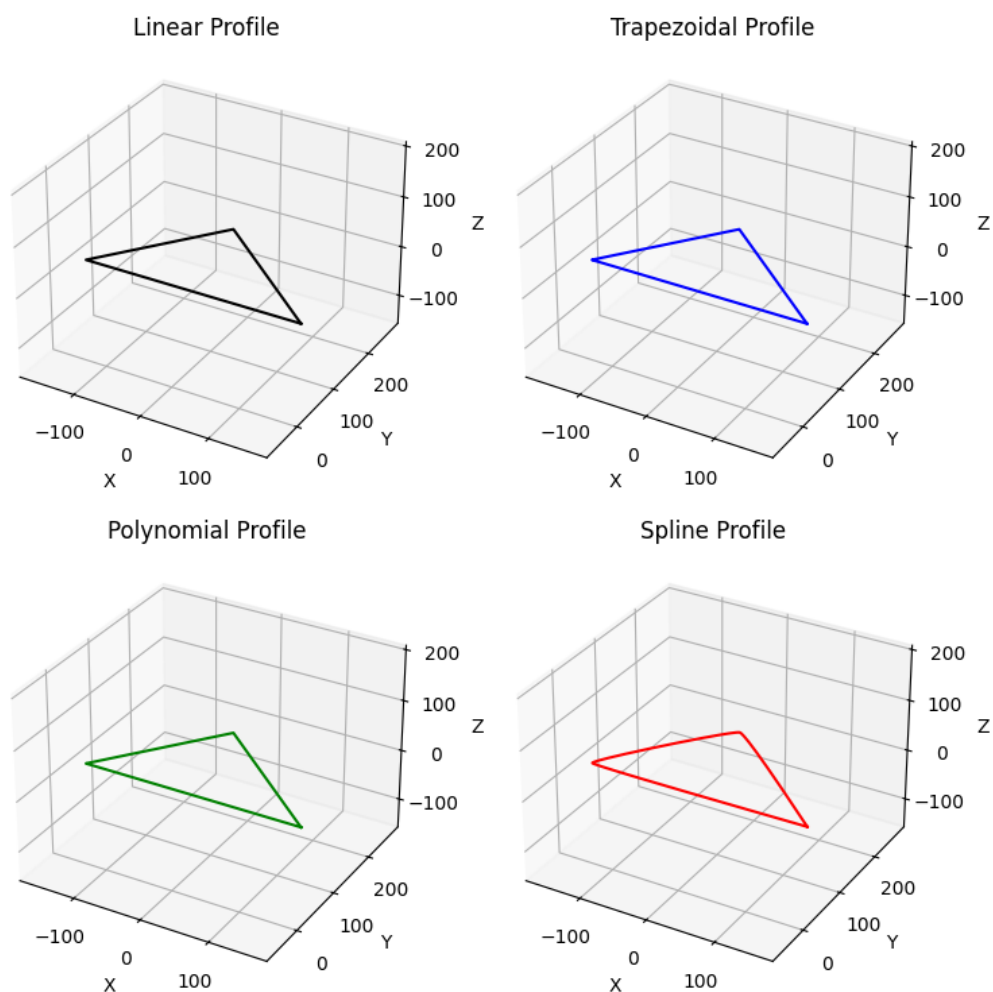


Рис. 2.19. Порівняння траєкторій при різних функціях інтерполяції

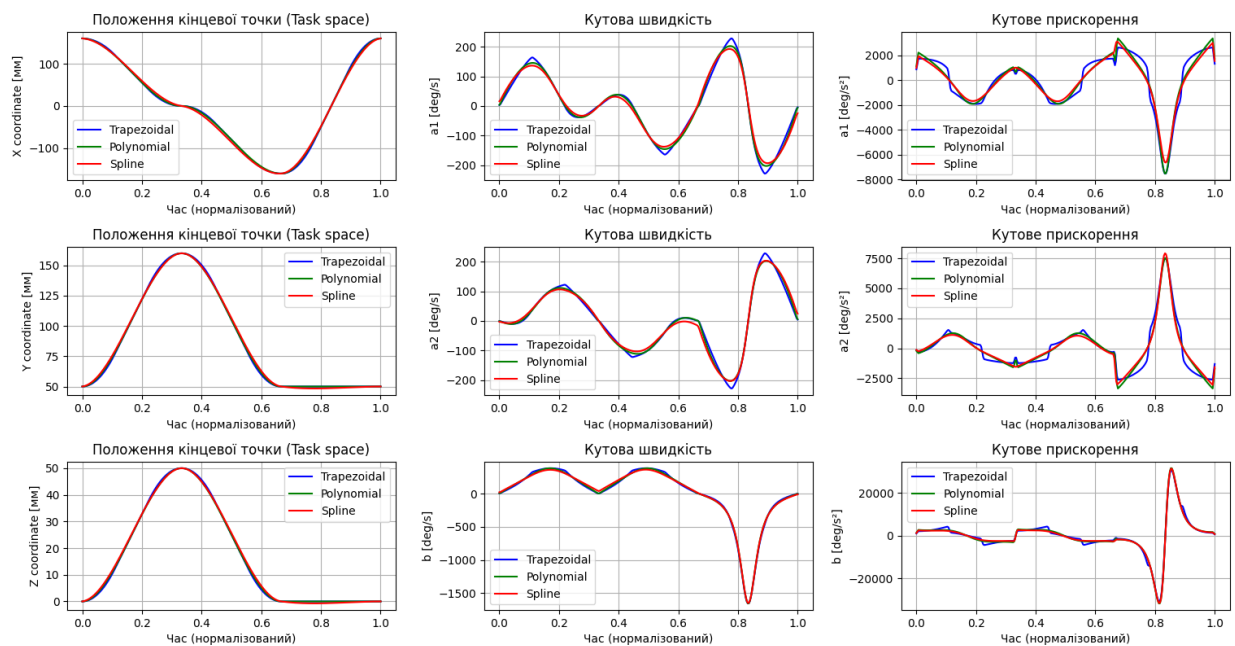


Рис. 2.20. Динамічні показники для кожної з траєкторій робота

2.3. Висновки за розділом

Розроблена математична модель робота-маніпулятора паралелограмного типу дозволяє детально описати його кінематичні властивості та динаміку руху. Структурний аналіз виявив, що досліджуваний механізм є конфігурацією Five-Bar Linkage з 2 ступенями вільності, який встановлений на обертову платформу, що надає системі загалом 3 ступені вільності.

Для механізму сформульовано та розв'язано пряму та обернену задачі кінематики, що забезпечує визначення положення робочого органу за кутами приводів і навпаки. Матриця Якобі пов'язує швидкості в просторі узагальнених координат із швидкостями в робочому просторі, а її визначник дозволяє виявити сингулярні конфігурації маніпулятора, при яких механізм втрачає один або більше ступенів вільності.

Робочий простір маніпулятора визначено з урахуванням геометричних обмежень, діапазону приводів, самозіткнень та сингулярностей. Порівняння

планування траєкторій у робочому просторі та просторі зчленувань виявило їх характерні особливості: Task space забезпечує кращий контроль над траєкторією кінцевої точки, а Joint space – більш плавний рух двигунів.

Дослідження різних методів інтерполяції траєкторій (трапецієподібних, поліноміальних, сплайнових) показало, що для забезпечення плавного руху маніпулятора доцільно використовувати трапецієподібні траєкторії швидкості, поліноміальну інтерполяцію або Catmull-Rom сплайни з відповідними параметрами натягнутості, що дозволяє мінімізувати різкі зміни прискорення при проходженні проміжних точок.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ КЕРУВАННЯ РОБОТОМ-МАНІПУЛЯТОРОМ

3.1. Розробка програми ручного керування

У комплектації робота, вже наявний пульт керування, що складається з двох джойстиків, та двох тактових кнопок (рис. 3.1).

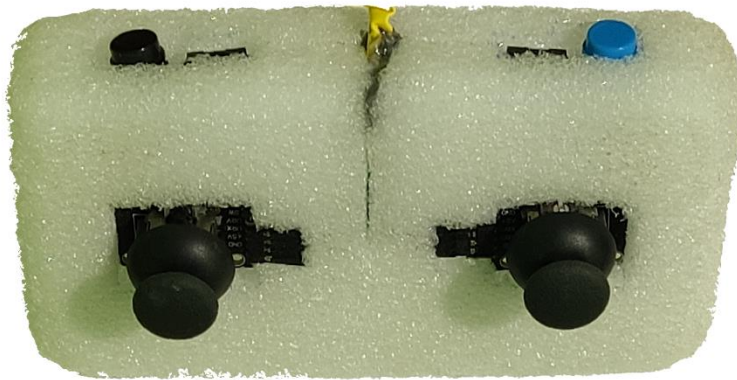


Рис. 3.1. Пульт керування

Враховуючи що є потреба керувати п'ятьма сервоприводами, необхідно визначити, чим буде керувати кожен елемент пульта. Найбільш зручно керувати хватом за допомогою кнопок, а двигунами – за допомогою джойстиків. Блок-схему алгоритму зображено на рис. 3.2.

Базове управління хватом (Gripper):

Читання аналогових значень із кнопок `openButtonPin` та `closeButtonPin`.

Умовні перевірки, які активують сервохват (`gripservo`) у певну позицію: `gripservo.write(0)` – для розімкнення, `gripservo.write(67)` – для зімкнення.

Зчитування потенціометрів:

Введено `previousMillis` і `currentMillis`, щоб функція не оновлювала значення аналогових входів на кожному циклі, а лише кожні 10 мс. Це дозволило реалізувати регулярне оновлення значень без затримок (`delay()`), зберігаючи плавність керування.

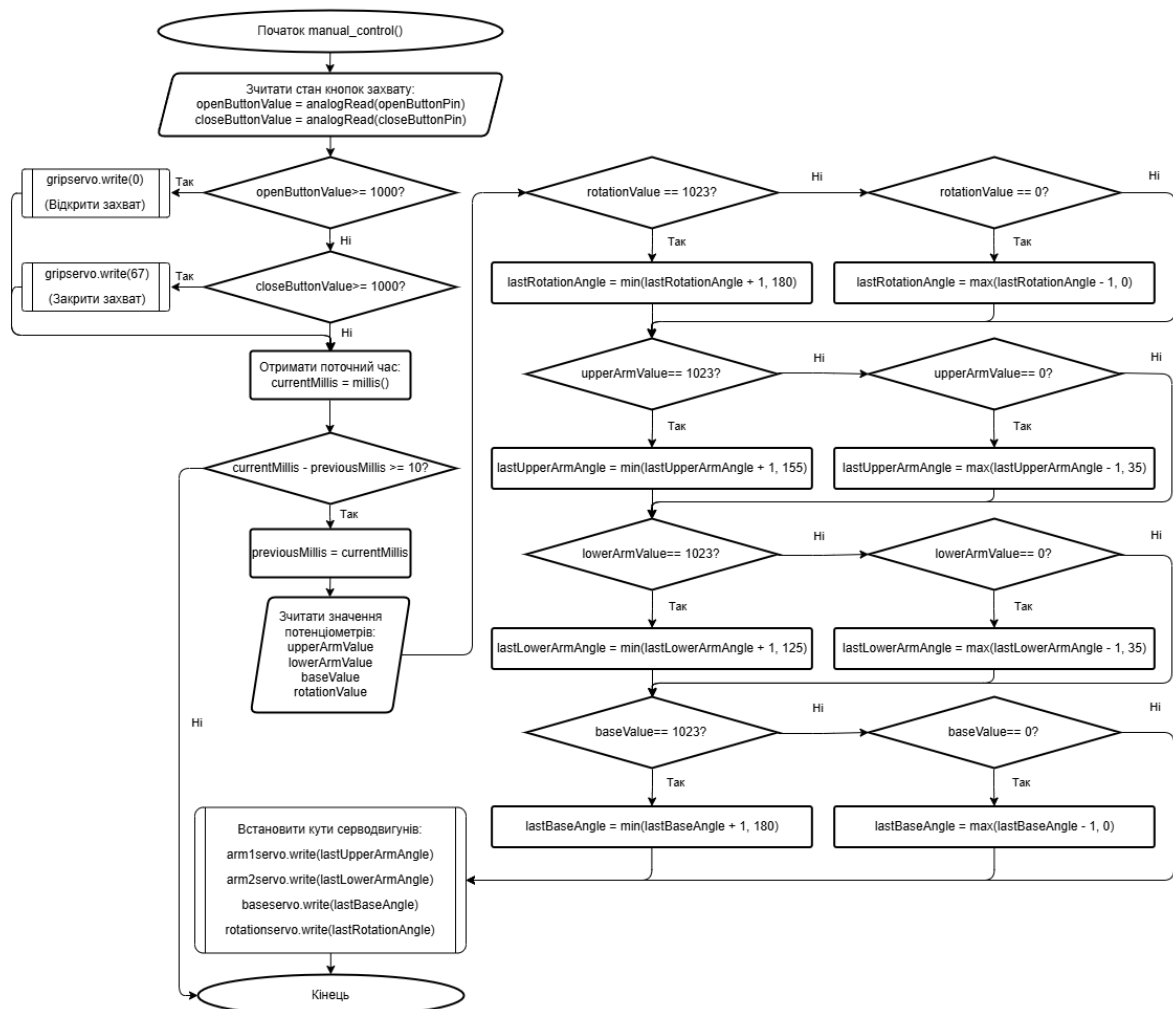


Рис. 3.2. Блок-схема роботи функції `manual_control()`

Читання аналогових значень із `upperArmPotPin`, `lowerArmPotPin`, `basePotPin`, `rotationPotPin`. Значення порівнюються з 0 або 1023, тобто з крайніми положеннями потенціометрів.

Замість прямого перетворення значень у кути, реалізовано поступову зміну кута: кут збільшується або зменшується на 1° в кожному циклі, поки значення на вході не зміниться. Це дало змогу досягти плавного управління замість стрибків.

Передача команд до сервоприводів:

Команди `servo.write(angle)` для кожного з сервомоторів. Значення `angle` передаються лише після перетворення та обмеження, забезпечуючи безпечний рух для механізму та двигунів.

3.2. Програмування алгоритму поліноміальної інтерполяції

Блок-схему алгоритму зображено на рис. 3.3.

Ініціалізація:

Зчитуються початкові координати, задані глобальними змінними `currentX`, `currentY`, `currentZ` (за замовчуванням початкове положення робота).

Встановлюється кількість кроків інтерполяції - `steps = 100` (чим більше значення – тим більш плавне переміщення).

Цикл обчислення проміжних точок:

Використовується нормалізований час (2.51).

Поліноміальний профіль S-кривої обчислюється за (2.52).

Обчислення проміжних координат (x , y , z) виконується за (2.53)

Передача обчислених точок до алгоритму розв'язання зворотної задачі кінематики – `moveToPos(x, y, z)`.

Затримка для обмеження швидкості переміщення.

Оновлення поточної позиції (`currentX`, `currentY`, `currentZ`):

Встановлення поточного положення як початкового для коректного обчислення траєкторії до наступної точки.

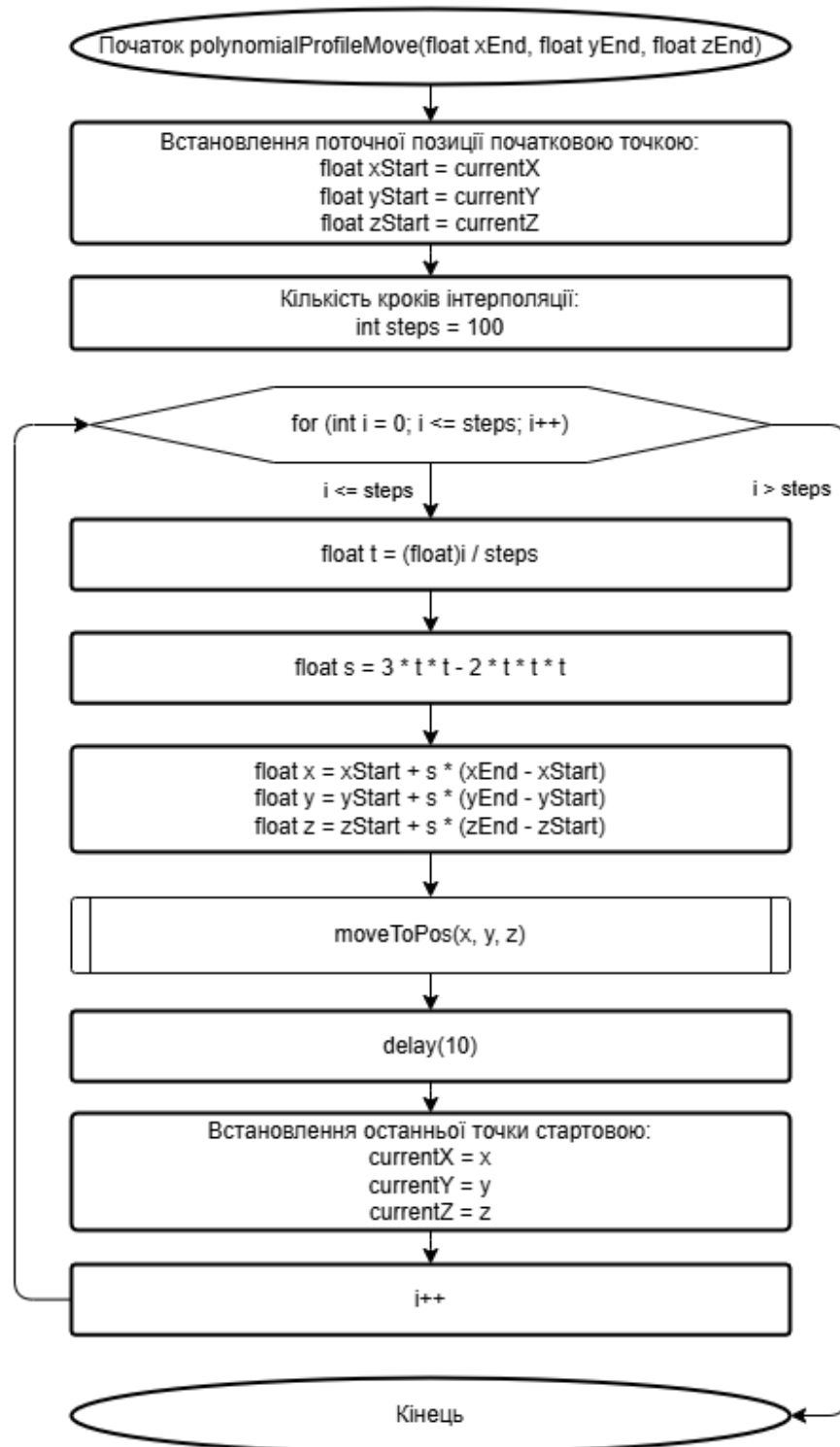


Рис. 3.3. Блок-схема роботи функції polynomialProfileMove()

3.3. Програмування алгоритму розв'язання зворотної задачі кінематики

Блок-схему алгоритму зображено на рис. 3.4.

Обчислення базових геометричних параметрів:

b – поворот платформи навколо вертикальної осі в градусах, визначається через (2.16).

l – горизонтальна відстань до цілі (в площині XY), визначається через (2.17).

h – пряма відстань до цілі у 3D (від бази до точки (x, y, z)), визначається через (2.18).

Перевірка на досяжність:

Якщо точка за межами робочої області (занадто близько чи далеко), рух не виконується.

Обчислення кутів для плеча та ліктя:

α - кут нахилу до точки в вертикальній площині (2.19).

β - кут у плечовому суглобі, для двох однакових сегментів по 160 мм (2.20).

α_1 - кут для приводу плеча в градусах (2.21).

Обчислення координат згину та другого кута:

Обчислюється положення точки C при довжині ліктьового сегмента довжиною 200 мм (2.22-24).

α_2 - кут для приводу ліктя в градусах (2.30-31).

Передача обчислених кутів до алгоритму перевірки, перетворення та встановлення кутів – moveToAngle().

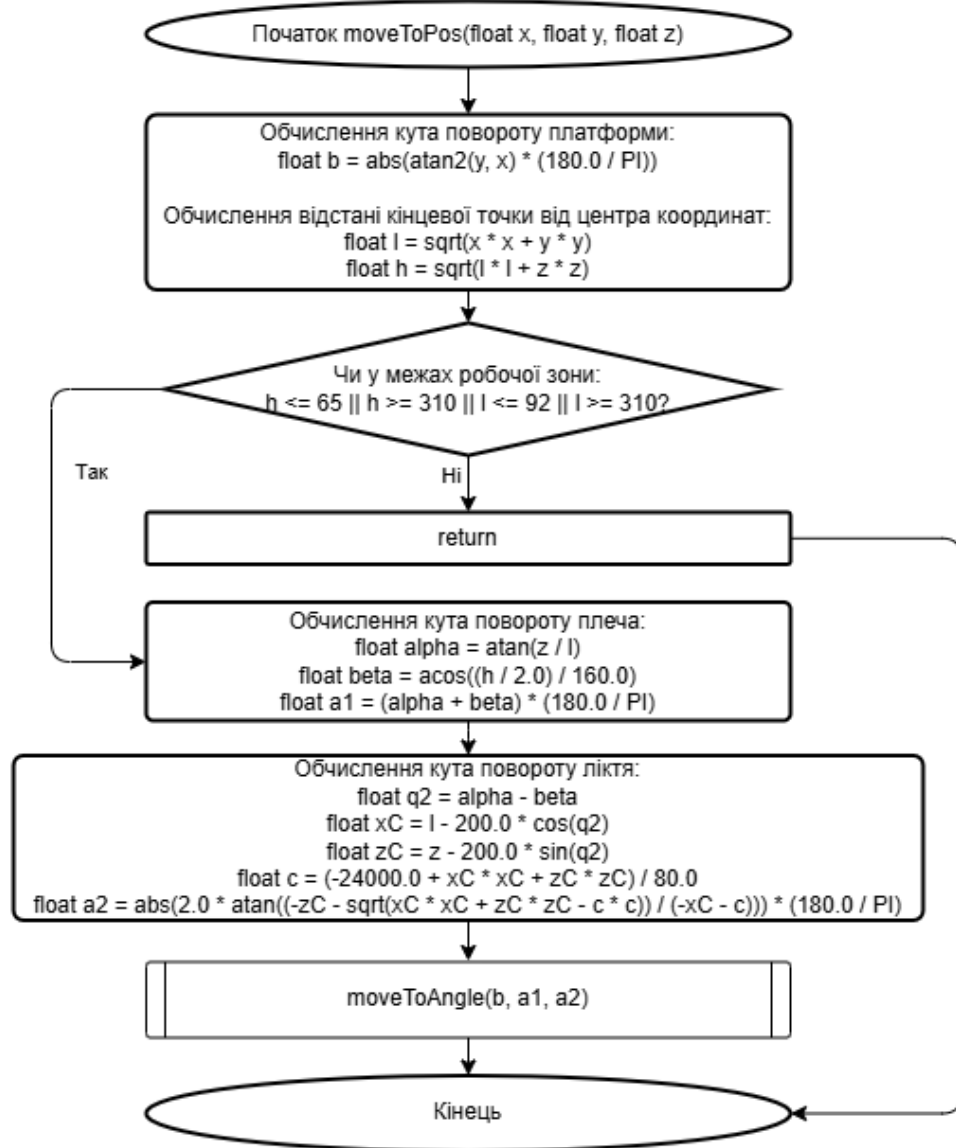


Рис. 3.4. Блок-схема роботи функції moveToPos()

3.4. Програмування алгоритму встановлення кута серводвигунів

Блок-схему алгоритму зображено на рис. 3.5.

Перевірка діапазону:

Перевіряє, чи кути в межах: $a1$ - від 0° до 120° , $a2$ - від 70° до 180° .

Якщо умова не виконується, виконується гілка else з повідомленням про помилку.

Перетворення кутів з урахуванням фізичного розташування сервоприводів:

map() - перетворює діапазон обчислених кутів у робочий діапазон серводвигуна: $a1$ мапується на діапазон 35-155, $a2$ мапується на діапазон 125-35.

Фізичне керування сервоприводами:

servo.write() – перетворює обчислені кути у ШІМ сигнал та надсилає до сервоприводів: arm1servo - плече, arm2servo - лікоть, baseservo - обертання платформи.

Обробка помилок:

Якщо кути поза межами, виводиться повідомлення про помилку зі значеннями кутів через UART.

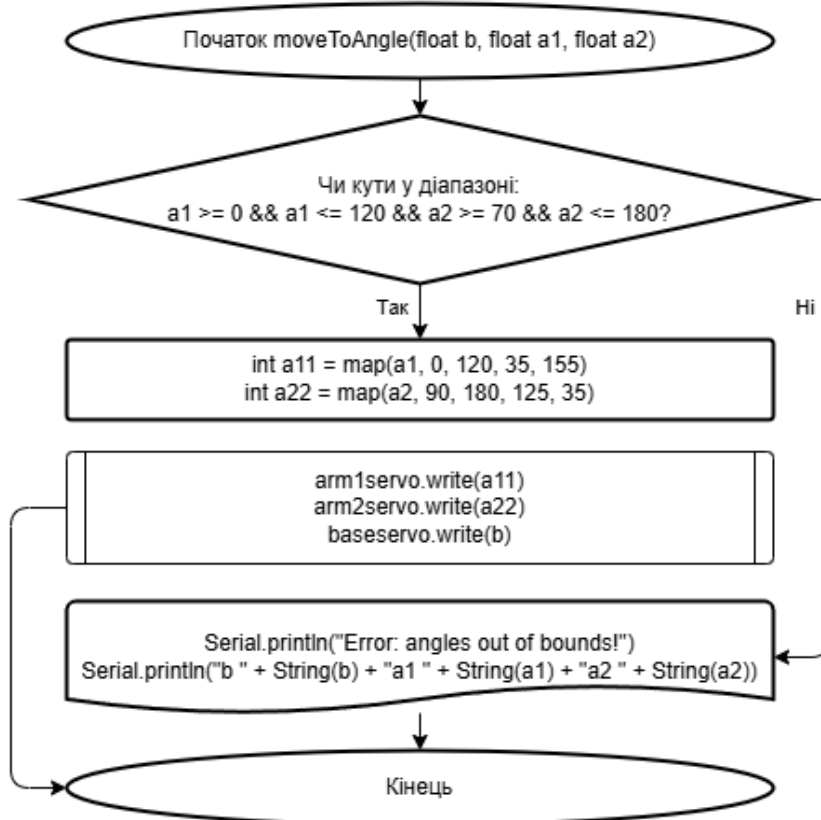


Рис. 3.5. Блок-схема роботи функції moveToAngle()

3.5. Програмування алгоритму вимірювання дистанції

Блок-схему алгоритму зображено на рис. 3.6.

Генерація ультразвукового імпульсу:

Подається імпульс тривалістю 10 мікросекунд на TRIGGER_PIN, що викликає генерацію ультразвукової хвилі датчиком HC-SR04.

Очікування відбитого сигналу:

Вимірюється затримка на ECHO_PIN - час, за який хвиля долетіла до об'єкта та повернулася назад (у мікросекундах).

Обчислення відстані:

Затримка множиться на швидкість звуку у повітрі (см/мкс), та ділиться навпіл, отриманий результат переводиться у міліметри.

Повернення результату функцією.

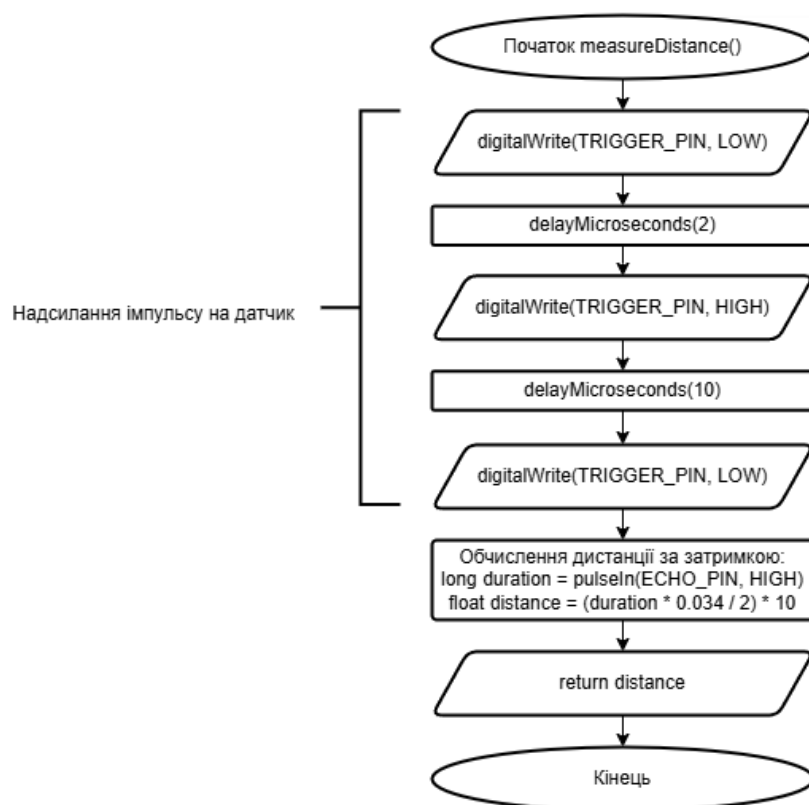


Рис. 3.6. Блок-схема роботи функції measureDistance()

3.6. Складання та реалізація алгоритму автоматичного пошуку об'єктів маніпулювання

Блок-схему алгоритму зображено на рис. 3.7.

Ініціалізація та підготовка:

`currentPosition` - зберігає поточну точку сканування (статична, щоб не скидалася при кожному виклику).

`num_points` - кількість точок на півколі по якому буде рухатися робоча точка під час сканування.

l – радіус півкола за яким буде рухатися робоча точка (100 мм).

`min_radius` – мінімальний радіус півкола (потрібен для корекції точок апроксимації півкола).

Переміщення до обчисленої позиції після циклу підпрограми.

Рух по півколу:

Кожен крок `currentPosition` дає новий кут `theta` по дузі (від π до 0).

Обчислюються координати точки сканування на колі з радіусом l .

Якщо точка ближча, ніж дозволено (`min_radius`), її віддаляють до мінімального радіуса.

Оновлюється координата точки (на фіксованій висоті -80 мм від висоти платформи).

Отримання відстані до об'єкта:

`distance` - вимірює відстань до об'єкта в напрямку цієї точки.

Виявлення об'єкта та захоплення:

Якщо об'єкт виявлено у зоні досяжності, обчислюється реальне положення об'єкта ($objX$, $objY$) шляхом додавання виміряної відстані вздовж напрямку руки. Z-координата не змінюється.

Рука наближається до об'єкта, хапає його, переносить в зону скидання, відпускає, повертається назад, і цикл сканування починається знову.

Перехід до наступної точки:

Якщо об'єкт не знайдено – індекс поточного положення переходить до наступної точки.

Якщо пройдено всі точки - індекс скидається і сканування починається знову.

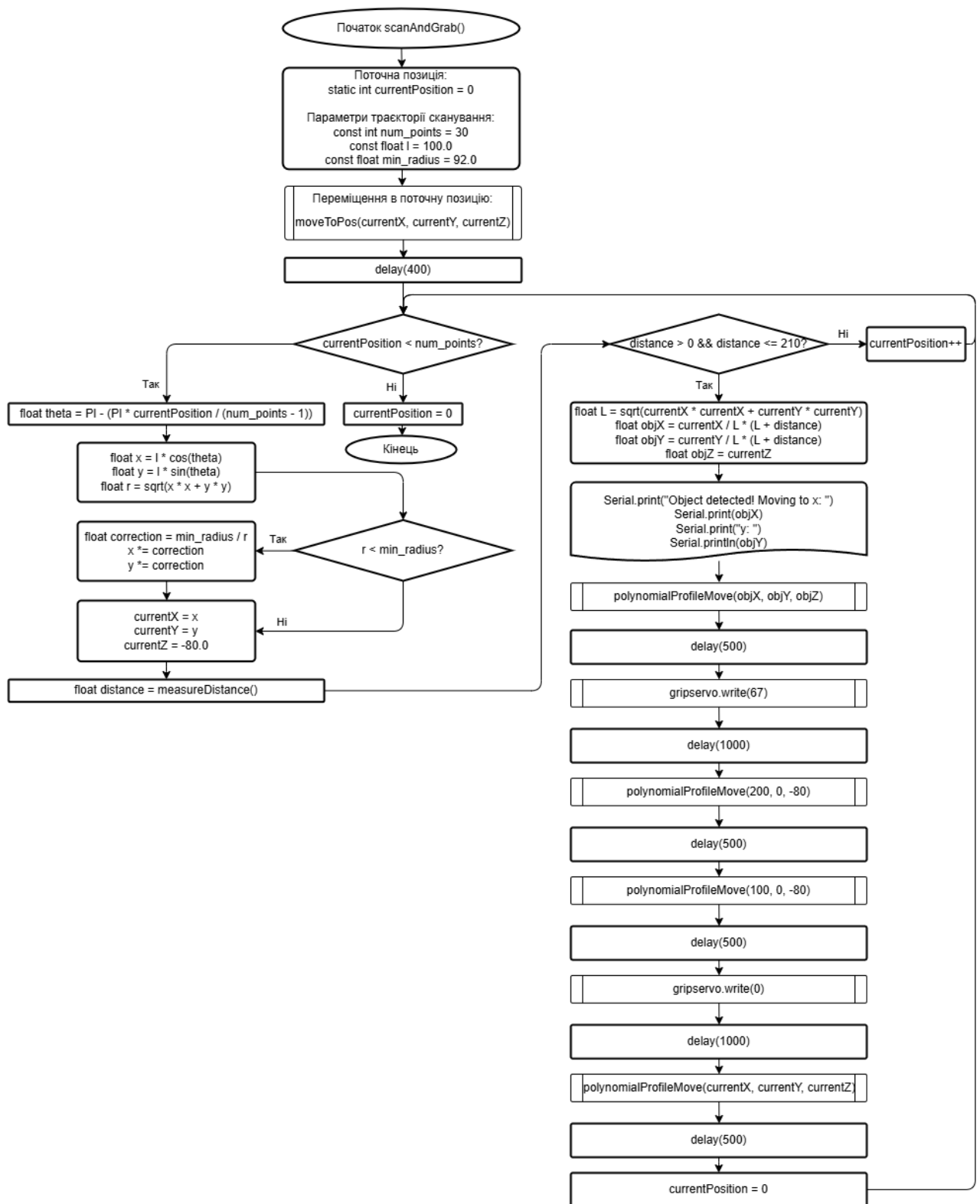


Рис. 3.7. Блок-схема роботи функції scanAndGrab()

3.7. Інтеграція режимів керування та розробка інтерфейсу користувача

Взаємодія користувача з системою може відбуватися не тільки за допомогою пульта керування, а й через UART. Так, користувач може надсилати команди керування системою у вигляді рядків.

SerialEvent() - викликається автоматично (рис. 3.8), коли надходять дані через USB (Serial). Функція читає вхідний текст, символ за символом:

Збирає вхідний рядок у змінну inputString.

Якщо зустрічає символ нового рядка (\n), позначає, що рядок завершено: stringComplete = true.

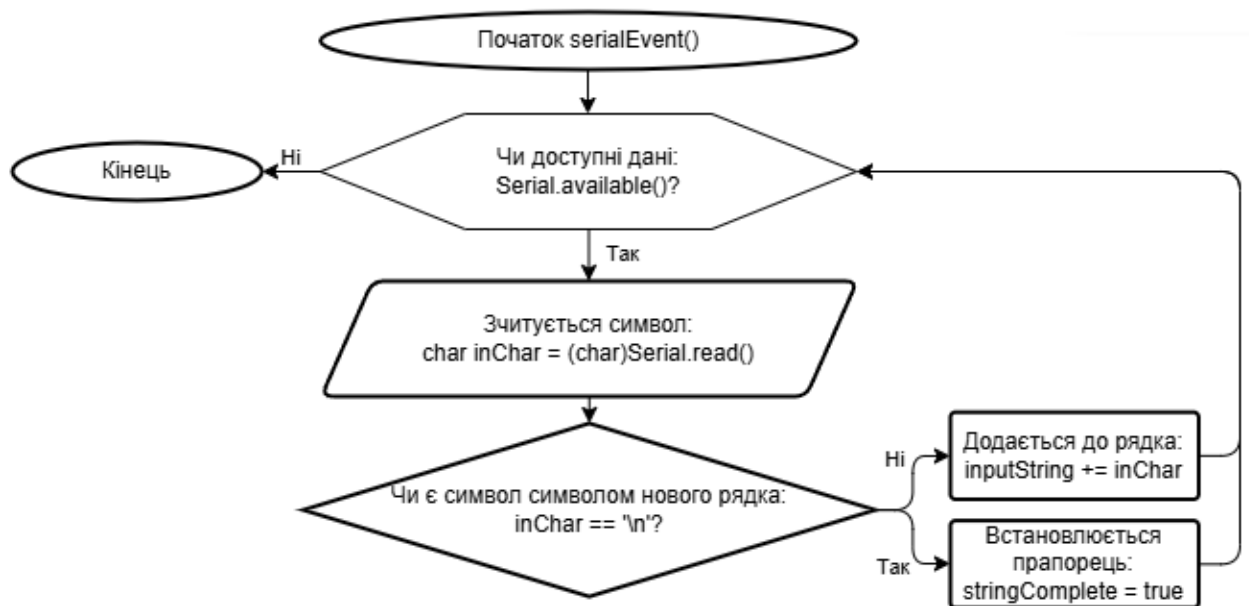


Рис. 3.8. Блок-схема роботи функції serialEvent()

ProcessSerialCommand() - викликається в основному циклі loop(), щоб обробити готову команду, якщо рядок завершено (рис. 3.9):

Перевірка завершення вводу.

Вивід прийнятої команди.

Розбір команди на числа:

Розбиває рядок на до 4 числових аргументи. Наприклад, введення “3 100 0 -80” розпізнається як: `values[0] = 3` - команда, `values[1..3]` параметри (підказки щодо формату параметрів для кожної команди виводяться на UART).

Обробка команд (через `switch(values[0])`):

- case 1 - Ручний режим. До виходу (доки не натиснуто обидві кнопки для виходу), керування відбувається вручну;
- case 2 - Автоматичне сканування і захоплення. Автоматично сканує й хапає об'єкти (доки не натиснуто обидві кнопки для виходу);
- case 3 - Переміщення до вказаних координат. Переміщення руки в координати, задані як `values[1]`, `values[2]`, `values[3]`. Також, відбувається перевірка допустимого діапазону по осі Z ($-135 \leq z \leq 160$);
- case 4 - Керування захопленням. Наприклад, 4 67 - закрити схват на кут 67, 4 0 – відкрити;
- default – обробка невідомої команди. Виводить повідомлення про те що система не може розпізнати введену команду.

Завершення: Після обробки, все скидається для прийому нової команди.

Таким чином, у головній функції (рис. 3.10) відбувається лише ініціалізація периферійних пристроїв, переміщення на початкову позицію, виведення інтерфейсу користувача через UART, та сповіщення звуковим сигналом про готовність до роботи (рис 3.9).

У головному циклі (рис. 3.11) проходять наступні дії:

- виклик функції обробки команд `processSerialCommand()`,
- перевірка умови одночасного натиснення обох кнопок пульта керування – якщо так, то програвється звуковий сигнал вимкнення, виводиться повідомлення про перезапуск, та встановлюється watchdog-таймер на 15 мс, що спричиняє м'яке перезавантаження контролера, а якщо ні – переміщення на початок циклу.

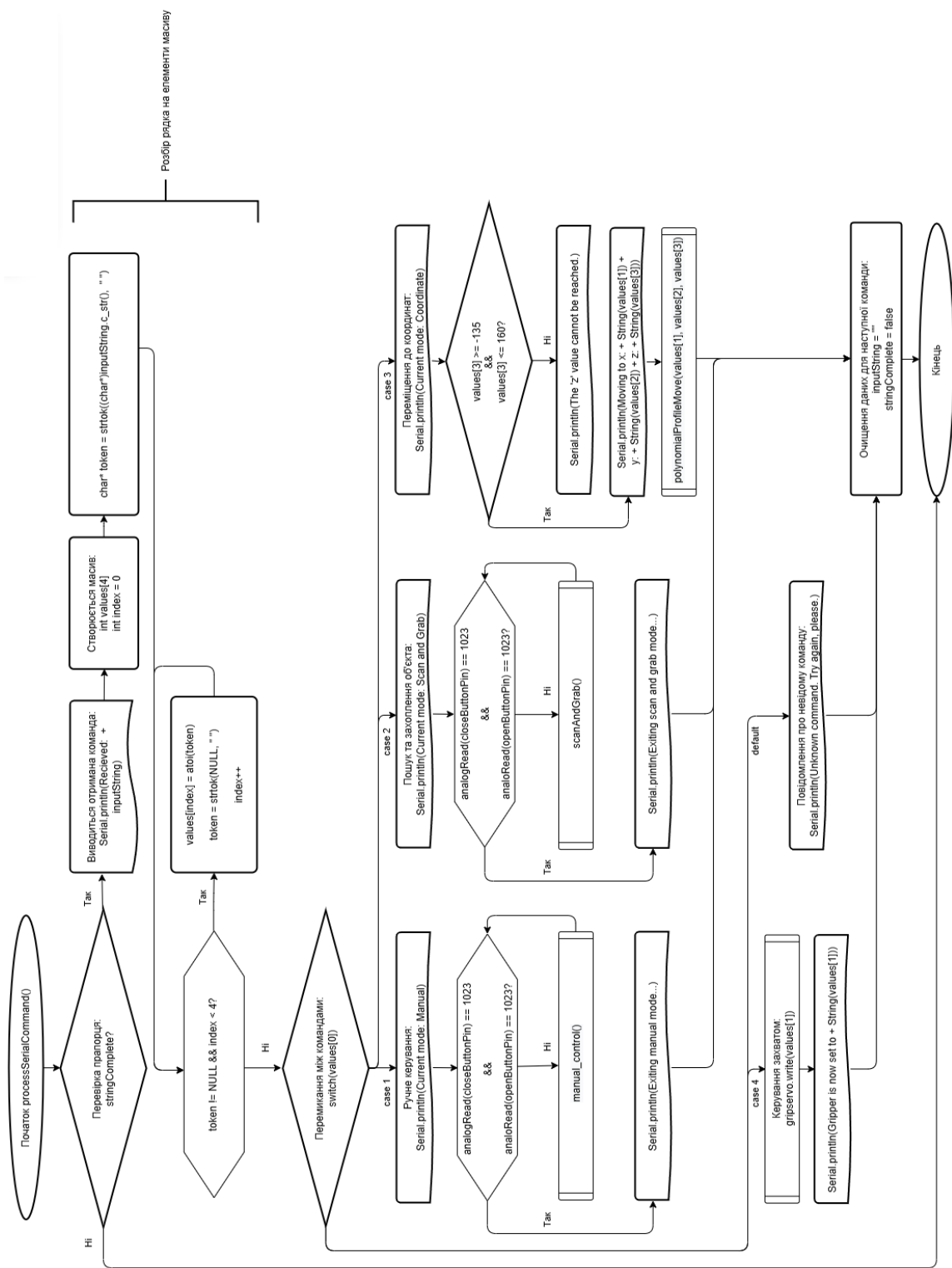


Рис. 3.9. Блок-схема роботи функції processSerialCommand()

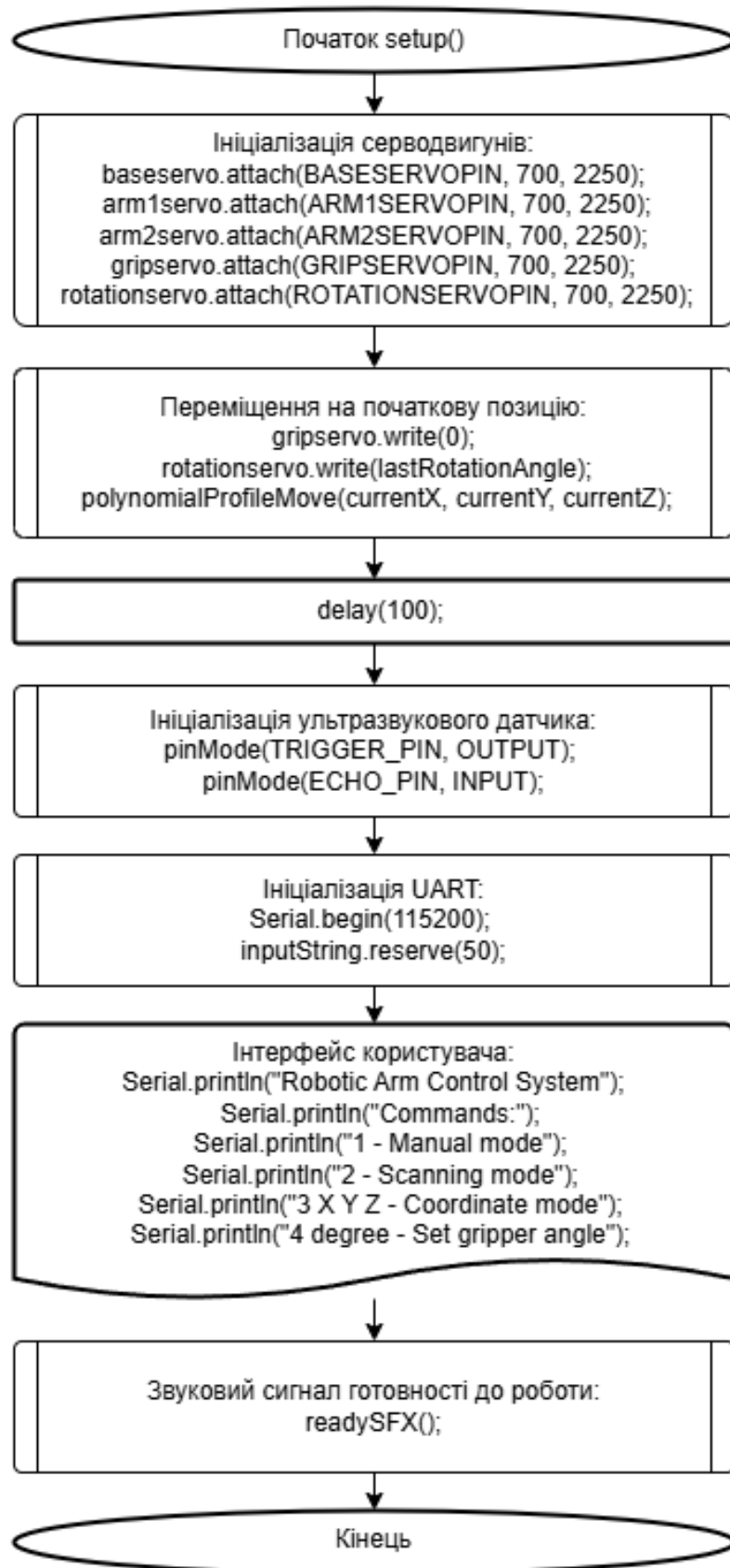


Рис. 3.10. Блок-схема роботи головної функції

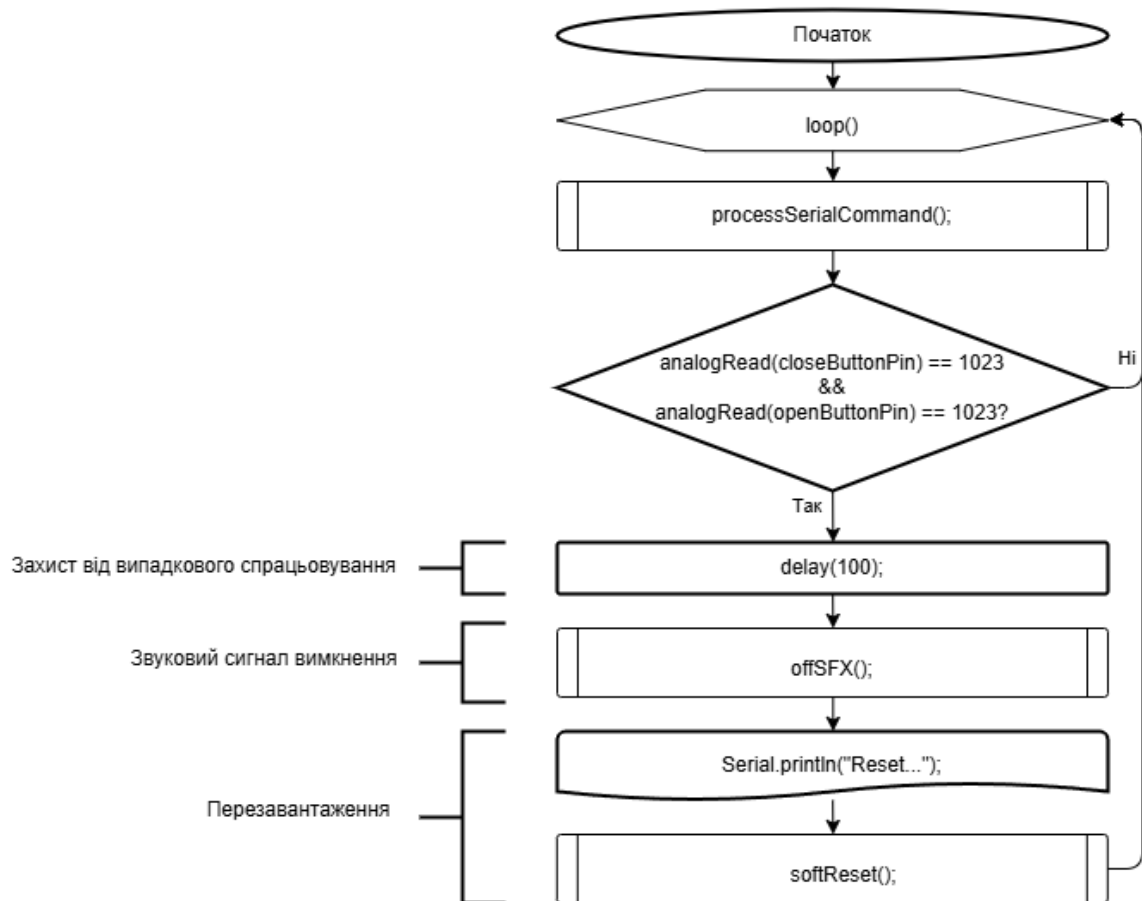


Рис. 3.11. Блок-схема роботи головного циклу

3.8. Висновки за розділом

У контексті виконаної розробки, можна зробити кілька узагальнених і технічно обґрунтованих висновків щодо ефективності, логічної структури та функціональної повноти програмного забезпечення, розробленого для даної системи.

Передусім, варто зазначити, що програмна частина керування реалізована у вигляді взаємопов'язаних модулів, кожен з яких виконує окрему логічну функцію в загальній архітектурі системи. Особливо важливу роль відіграє механізм приймання та обробки команд, який забезпечує інтерфейс між користувачем (або зовнішньою програмою) та роботом-маніпулятором. Це дозволяє здійснювати динамічну зміну режимів роботи пристрою без потреби у

перепрошиванні мікроконтролера. Таким чином, реалізовано гнучке інтерактивне управління, що є необхідним для систем, які функціонують у змінному або частково непередбачуваному середовищі.

У межах програмної реалізації впроваджено декілька режимів роботи маніпулятора: ручний, автоматичний (режим сканування та захоплення), а також режим координатного позиціонування. Кожен з цих режимів охоплює відповідні функціональні блоки, що забезпечують як низькорівневе управління сервоприводами, так і високорівневу логіку взаємодії із зовнішніми об'єктами. Наприклад, функція автоматичного пошуку та захоплення реалізує циклічне сканування простору з подальшим визначенням положення об'єкта за допомогою ультразвукового датчика, що свідчить про інтеграцію сенсорного контролю та зворотного зв'язку в логіку роботи системи.

Особливу увагу в програмній реалізації приділено обмеженню робочого простору маніпулятора. Перевірка діапазонів кутів та координат під час обчислення положення руки дозволяє уникнути виходу за фізичні межі руху, що зменшує ризик пошкодження конструктивних елементів. Таким чином, програмно забезпечено захист від помилкових дій, що підвищує надійність та довговічність роботи пристрою.

Загалом, розроблене програмне забезпечення демонструє продуману архітектуру та відповідає вимогам до керування реальними технічними об'єктами. Воно поєднує в собі простоту у використанні, адаптивність до різних режимів експлуатації та базовий рівень безпеки. Отже, програмна реалізація виступає як ключовий компонент системи керування роботом-маніпулятором, забезпечуючи її функціональну завершеність та практичну придатність до використання у прикладних задачах автоматизації.

РОЗДІЛ 4

ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

4.1. Методика проведення експериментальних досліджень

Для оцінки функціональних характеристик розробленого робота-маніпулятора (рис. 4.1) було застосовано два основні методи експериментального дослідження: перевірка точності позиціонування та випробування ефективності алгоритму автоматичного виявлення об'єктів. У дослідженні точності позиціонування використовувався метод порівняння заданих координат цільової точки з фактичними координатами положення захвату після виконання переміщення. Результати фіксувалися шляхом візуального або контактного вимірювання, з подальшим обчисленням абсолютної похибки за кожною віссю та в просторі, а також статистичних показників, таких як середньоквадратичне відхилення і повторюваність. Для перевірки роботи алгоритму виявлення об'єктів застосовувався метод радіального сканування з використанням ультразвукового датчика, в межах якого захват переміщувався по півколу з фіксованим радіусом і кутовим кроком, здійснюючи серію вимірювань відстані до потенційних об'єктів. У процесі дослідження вимірювалися час сканування, час руху до об'єкта після виявлення, точність визначення координат, а також фіксувалися випадки виявлення чи пропуску об'єктів для подальшого розрахунку ймовірності виявлення. Усі експерименти проводилися в стабільному середовищі для забезпечення повторюваності умов.

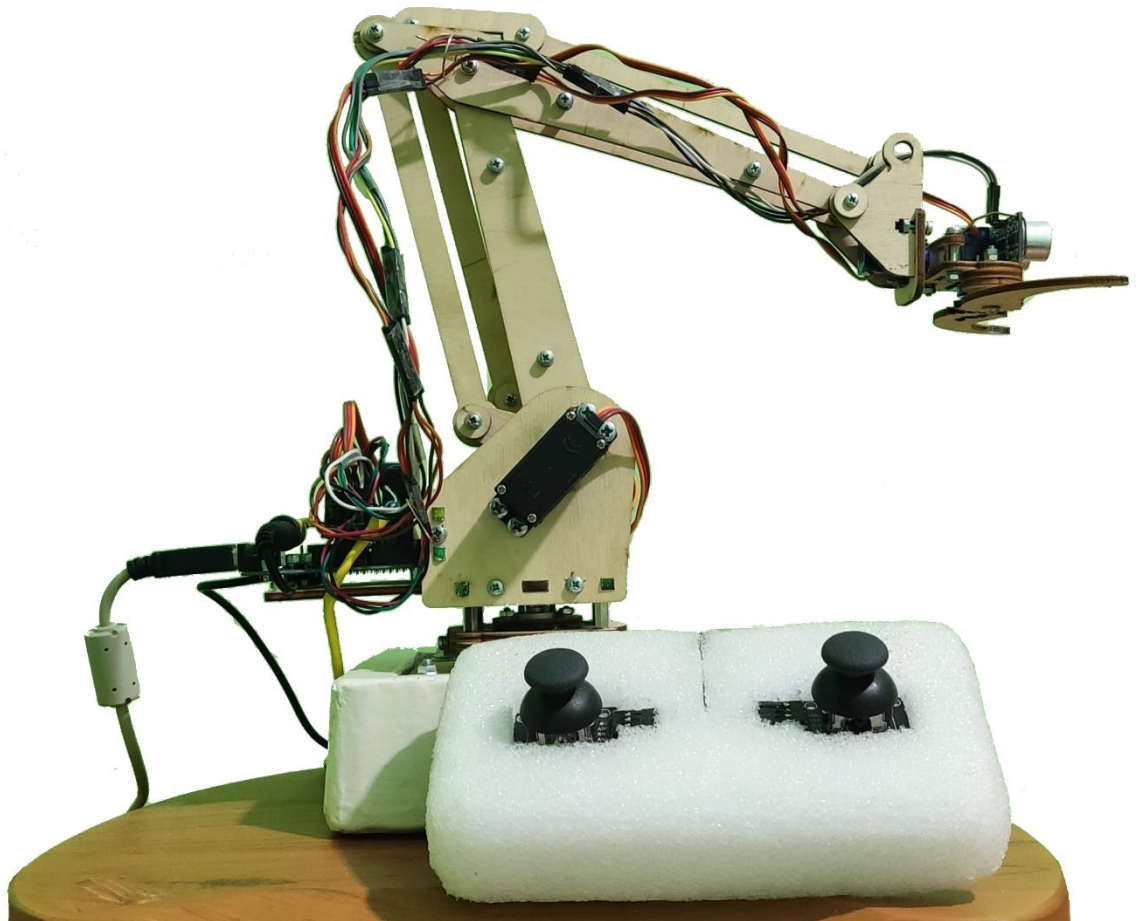


Рис. 4.1. Робот-маніпулятор з пультом керування

4.2. Аналіз точності позиціонування захвату робота-маніпулятора

Точність позиціонування є одним із ключових параметрів ефективності роботи маніпулятора. Під час експерименту робот отримував координати цільової точки, після чого переміщував захват у зазначене положення.

Для кожного тесту фіксувалось:

- задане положення (X_0, Y_0, Z_0),
- фактичне положення захвату (X, Y, Z),
- абсолютна похибка позиціонування за кожною віссю,
- середнє квадратичне відхилення.

Абсолютна похибка для крайніх положень на кожній осі окремо:

При $X_0 = 310, Y_0 = 0, Z_0 = 0$:

$$\Delta X = |X_0 - X| = |310 - 307| = 3 \text{ мм}, \quad (4.1)$$

При заданій точці $X_0 = 0, Y_0 = 310, Z_0 = 0$:

$$\Delta Y = |Y_0 - Y| = |310 - 307| = 3 \text{ мм}, \quad (4.2)$$

При заданій точці $X_0 = 0, Y_0 = 0, Z_0 = 160$:

$$\Delta Z = |Z_0 - Z| = |160 - 166| = 6 \text{ мм}. \quad (4.3)$$

Абсолютна похибка у тривімірному просторі:

При заданій точці $X_0 = 219, Y_0 = 219, Z_0 = 0$:

$$\Delta X = |X_0 - X| = |219 - 220.5| = 1,5 \text{ мм}, \quad (4.4)$$

$$\Delta Y = |Y_0 - Y| = |219 - 220.5| = 1,5 \text{ мм}, \quad (4.5)$$

$$\Delta Z = |Z_0 - Z| = |0 - 3| = 3 \text{ мм}, \quad (4.6)$$

$$\Delta_{abs} = \sqrt{(\Delta X)^2 + (\Delta Y)^2 + (\Delta Z)^2}, \quad (4.7)$$

$$\Delta_{abs} = \sqrt{(1,5)^2 + (1,5)^2 + (3)^2} = 3,674 \text{ мм}. \quad (4.8)$$

Середнє квадратичне відхилення:

$$\sigma_{max} = \sqrt{\frac{1}{N} \sum_{i=0}^N (P_i + P_{mean})}, \quad (4.9)$$

де N – кількість вимірювань, P_i – масив виміряних значень, P_{mean} – середнє арифметичне з усіх P_i .

Максимальна повторюваність:

$$R = 2\sigma_{max}. \quad (4.10)$$

Результати експериментів показали, що середня похибка позиціонування становить $\pm 2,012$ мм по осі X , $\pm 2,012$ мм по осі Y , та $\pm 3,37$ мм по осі Z .

Найбільші відхилення були зафіксовані при роботі в крайніх положеннях робочої зони, що пояснюється в першу чергу люфтом з'єднань механізму та використанням потенціометричних датчиків у зворотному зв'язку сервоприводів, а також кумулятивною похибкою обертання ланок, впливом гравітації.

4.3. Оцінка ефективності алгоритму пошуку об'єктів роботом

У розробленій системі пошук об'єктів здійснюється за допомогою алгоритму радіального сканування в полярній системі координат. Робот-маніпулятор виконує послідовне переміщення захвату по півколу з фіксованим радіусом та кутовим кроком. На кожній позиції вимірюється відстань до потенційного об'єкта за допомогою ультразвукового датчика.

Ефективність алгоритму оцінювалася за наступними критеріями.

Час сканування: в середньому 13,5 секунд.

Час руху до об'єкта з моменту виявлення: в середньому 2,5 секунди.

Імовірність виявлення: до 98% для об'єктів з коефіцієнтом поглинання звуку (40 кГц) $< 0,2$.

Похибка визначення координат: не перевищує ± 6 мм при стабільній роботі ультразвукового датчика.

Стійкість до перешкод: алгоритм демонструє стабільну роботу за умови відсутності фонового шуму та при достатньому розмірі об'єкта.

Обмеження.

Система погано розрізняє об'єкти розмірами менше 3–4 см.

Виявлення об'єктів можливе лише в межах сектору 180° та радіусом до 310 мм.

Вимірювання глибини залежить від якості сигналу ультразвукового датчика та кута падіння хвиль, що може викликати хибні спрацьовування на крайніх позиціях.

4.4. Висновки за розділом

Проведені експериментальні дослідження підтвердили працездатність та ефективність розробленої системи робота-маніпулятора. Аналіз точності позиціонування показав, що середня абсолютна похибка перебуває в межах допустимих значень для подібного класу пристроїв, становлячи приблизно ± 2 мм по осях X і Y та $\pm 3,4$ мм по осі Z . Найбільші відхилення спостерігалися поблизу меж робочої області, що обумовлено конструктивними особливостями та точністю зворотного зв'язку. Показник повторюваності вказує на достатню стабільність системи при виконанні однакових рухів у схожих умовах.

Алгоритм автоматичного виявлення об'єктів на основі радіального сканування в полярній системі координат продемонстрував високу ефективність при виконанні задач виявлення й захоплення цілей. Зокрема, ймовірність успішного виявлення об'єктів з низьким коефіцієнтом поглинання ультразвуку сягала 98%. Водночас були виявлені обмеження, пов'язані з мінімальними розмірами об'єктів, кутами падіння хвиль та особливостями відбиття сигналу.

Загалом, результати експериментів засвідчили, що реалізована система здатна успішно виконувати базові завдання позиціонування та автоматичного пошуку об'єктів. Надалі можливе вдосконалення точності та надійності роботи шляхом оптимізації механічної частини та застосування більш досконалих сенсорів.

РОЗДІЛ 5

ОХОРОНА ПРАЦІ

5.1. Загальні поняття охорони праці

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Метою охорони праці є створення безпечних, нешкідливих та комфортних умов праці, запобігання виробничому травматизму, професійним захворюванням і аваріям. Правову основу охорони праці в Україні становлять Конституція України, Закон України "Про охорону праці", Кодекс законів про працю України та інші нормативно-правові акти.

Основними принципами державної політики в галузі охорони праці є:

- пріоритет життя і здоров'я працівників;
- повна відповідальність роботодавця за створення належних, безпечних і здорових умов праці;
- підвищення рівня промислової безпеки;
- комплексне розв'язання завдань охорони праці;
- соціальний захист працівників, які потерпіли від нещасних випадків на виробництві.

5.2. Аналіз небезпечних та шкідливих факторів, що діють на оператора системи керування роботом-маніпулятором

При роботі з системою керування роботом-маніпулятором оператор піддається впливу низки небезпечних та шкідливих виробничих факторів, які можна класифікувати за наступними категоріями:

Фізичні фактори:

- 1) механічні небезпеки:
 - ризик травмування рухомими частинами робота-маніпулятора при його налаштуванні або обслуговуванні;
 - небезпека затискання між рухомими та нерухомими частинами обладнання;
 - можливість падіння деталей, що переміщуються роботом.
- 2) Електричні небезпеки:
 - ризик ураження електричним струмом при контакті з електричними компонентами системи керування;
 - небезпека короткого замикання та можливість виникнення пожежі.
- 3) Вплив шуму та вібрації:
 - підвищений рівень шуму від електроприводів робота-маніпулятора (до 45-48 дБА);
- 4) Електромагнітне випромінювання:
 - вплив електромагнітних полів від електронних компонентів системи керування.
- 5) Освітлення:
 - недостатня або надмірна освітленість робочої зони (фактична освітленість часто коливається в межах 200-300 лк при нормі 400 лк);
 - нерівномірність освітлення, прямі та відбиті відблиски, що можуть викликати зорове стомлення.

Психофізіологічні фактори:

- 1) нервово-психічні навантаження:
 - напруженість уваги при контролі за роботою системи (до 75% робочого часу);
 - необхідність швидкого прийняття рішень в аварійних ситуаціях (час реакції має становити менше 0,5 секунди);

- монотонність роботи при тривалому спостереженні за технологічним процесом (до 60% робочого часу).
- 2) Фізичні перевантаження:
- тривале перебування у вимушеній робочій позі (до 80% робочого часу);
 - напруження зору при роботі з дисплеями та панелями керування (безперервна тривалість зосередження погляду до 45 хвилин).

Хімічні фактори (вплив хімічних речовин):

- можливий вплив мастильних матеріалів, використовуваних для обслуговування робота-маніпулятора (мінеральні масла, концентрація в повітрі до 5 мг/м³);
- випари від технологічних процесів, що виконуються за допомогою робота (залежно від специфіки виробництва).

5.3. Заходи з техніки безпеки для зменшення або усунення впливу небезпечних та шкідливих факторів при обслуговуванні запропонованої системи

Організаційні заходи:

- 1) навчання та інструктаж персоналу:
- проведення первинного інструктажу з охорони праці тривалістю не менше 8 годин з практичними заняттями;
 - щоквартальні періодичні інструктажі тривалістю 2-4 години;
 - спеціальне навчання роботі з системами керування роботами-маніпуляторами - 20 годин теоретичних занять та 30 годин практики;
 - регулярна перевірка знань правил безпеки праці: щорічна атестація з тестуванням та практичними завданнями;

- розробка та видача персоналу детальних інструкцій з безпечної експлуатації системи (обсягом не менше 15 сторінок).
- 2) Режим праці та відпочинку:
- тривалість безперервної роботи з системою керування - не більше 2 годин;
 - короткі регламентовані перерви: 15 хвилин після кожних 2 годин роботи;
 - спеціальна перерва для виконання комплексу вправ для очей - 5 хвилин щогодини;
 - комплексна перерва для фізичних вправ та відновлення - 30 хвилин в середині зміни;
 - обмеження загальної тривалості роботи з системою до 6 годин за 8-годинну зміну;
 - ротація працівників між різними видами діяльності через кожні 4 години.
- 3) Організація робочого місця:
- ергономічна організація пульта керування згідно з ДСТУ ISO 9241-5:2004;
 - регульоване робоче крісло з підтримкою поперекового відділу хребта та підлокітниками;
 - розміщення основних органів керування на відстані 25-45 см від оператора;
 - розміщення засобів відображення інформації під кутом 15-30° нижче рівня очей оператора;
 - встановлення пультів керування на висоті 72-78 см від підлоги;
 - забезпечення простору для ніг висотою не менше 60 см, шириною не менше 70 см, глибиною на рівні колін не менше 45 см.

Технічні заходи:

- 1) захист від механічних небезпек:

- встановлення захисних огорожень робочої зони роботи маніпулятора (розрахунок за формулами 6.1-6.5);
- оснащення системи датчиками присутності людини з радіусом дії не менше 0.5 м та швидкодією 0,1 сек;
- встановлення оптичних бар'єрів із частотою сканування не менше 25 Гц;
- наявність кнопок аварійної зупинки на відстані не більше 0,5 м від периметру робочої зони.

2) Електробезпека:

- використання захисного заземлення обладнання з опором не більше 8 Ом;
- застосування захисного відключення системи з часом спрацьовування не більше 0,05 с при струмі витoku 30 мА;
- ізоляція струмоведучих частин з опором ізоляції не менше 1 МОм;
- встановлення подвійної ізоляції для ручних органів управління;
- регулярна перевірка опору ізоляції та контуру заземлення (не рідше 1 разу на 6 місяців);
- використання розділових трансформаторів для електронних компонентів системи керування з коефіцієнтом трансформації не менше 1:1.

3) Зниження шуму та вібрації:

- встановлення віброізоляційних опор для обладнання із коефіцієнтом передачі не більше 0,2;
- застосування звукопоглинаючих матеріалів із коефіцієнтом поглинання не менше 0,7 у діапазоні частот 250-4000 Гц;
- використання шумопоглинаючих екранів висотою не менше 1 м з ефективністю зниження шуму на 12-15 дБА;
- встановлення амортизуючих прокладок між вібруючими елементами та опорними конструкціями;

- регулярне технічне обслуговування електроприводів та механічних вузлів (не рідше 1 разу на місяць).
- 4) Оптимізація мікроклімату:
- забезпечення локальної вентиляції робочої зони з кратністю повітрообміну не менше 1.3 об'ємів за годину;
 - підтримання температури повітря в межах 22-24°C в теплий період року та 21-23°C в холодний період;
 - забезпечення відносної вологості повітря в межах 40-60%;
 - обмеження швидкості руху повітря до 0,1-0,2 м/с;
 - встановлення систем кондиціонування із функцією очищення повітря від пилу ефективністю не менше 90%.
- 5) Покращення освітлення:
- забезпечення рівномірного освітлення робочої зони з коефіцієнтом пульсації не більше 10%;
 - підтримання рівня освітленості 400-500 лк на робочих поверхнях;
 - встановлення місцевого освітлення з можливістю регулювання інтенсивності в межах 300-500 лк;
 - використання світильників з розсіяним світлом та захисним кутом не менше 40°;
 - застосування антиблікових фільтрів для екранів відображення інформації.

Розрахунки технічних заходів безпеки.

- 1) Розрахунок захисного огороження робочої зони.

Вихідні дані:

- максимальна довжина вильоту маніпулятора: $L = 0,44$ м;
- максимальна швидкість руху робочого органу: $V = 1,62$ м/с;
- маса деталей, що переміщуються: $m = 0.140$ кг.

Розрахунок безпечної відстані розташування огороження.

Безпечна відстань розраховується за формулою:

$$S = L + \Delta L + L_p + S_0, \quad (5.1)$$

де ΔL - додаткова відстань, враховуюча можливі відхилення від програмної траєкторії (приймаємо $\Delta L = 0,005$ м); L_p - довша сторона найбільшого предмету що може бути захоплений маніпулятором (приймаємо $L_p = 0,15$ м); S_0 - мінімальна допустима відстань між огородженням і найбільш висунутою частиною робота (приймаємо $S_0 = 0,15$ м).

Таким чином, безпечна відстань складає:

$$S = 0,44 + 0,005 + 0,15 + 0,15 = 0,745 \text{ м.}$$

Розрахунок висоти огородження.

Висота огородження повинна бути такою, щоб виключити можливість перекидання деталей через верхню частину огорожі. Розрахункова висота огородження:

$$H = h_1 + h_2, \quad (5.2)$$

де h_1 - максимальна висота підйому робочого органу маніпулятора (приймаємо $h_1 = 0,335$ м); h_2 - додаткова висота, що враховує можливе випадіння деталей (приймаємо $h_2 = 0,134$ м).

Таким чином, висота огородження складає:

$$H = 0,335 + 0,134 = 0,469 \text{ м.} \quad (5.3)$$

Вибір матеріалу огородження.

Для захисту від механічних впливів огородження повинно витримувати енергію удару від можливого зіткнення з рухомими частинами робота-маніпулятора. Енергія удару розраховується за формулою:

$$E = \frac{m \times V^2}{2}, \quad (5.4)$$

$$E = \frac{0,140 \times 1,2^2}{2} = 0,1008 \text{ Дж.} \quad (5.5)$$

Виходячи з розрахунку енергії удару, в якості матеріалу огороження рекомендується використовувати металеву сітку з товщиною дроту не менше 1 мм і розміром чарунок не більше 10×10 мм, закріплену на металевому каркасі, або листи полікарбонату товщиною не менше 1 мм.

Вимоги до конструкції огороження:

- огороження повинно бути стаціонарним з блокувальними пристроями на дверях доступу;
- двері повинні бути оснащені електромеханічним блокуванням, що відключає робота при відкритті;
- на огороженні повинні бути розміщені попереджувальні знаки про небезпеку.

2) Розрахунок системи заземлення.

Вихідні дані:

- напруга живлення обладнання: $U = 220 \text{ В}$;
- потужність системи: $P = 15 \text{ Вт}$;
- питомий опір ґрунту: $\rho = 100 \text{ Ом} \cdot \text{м}$;
- тип заземлювача: сталевий стрижень діаметром $d = 16 \text{ мм}$, довжиною $l = 2,5 \text{ м}$.

Розрахунок опору одиночного заземлювача.

Опір одиночного вертикального заземлювача розраховується за формулою:

$$R_{\text{в}} = \frac{\rho}{2\pi \cdot l \cdot \left(\ln\left(2 \cdot \frac{l}{d}\right) + 0.5 \cdot \ln\left(\frac{4 \cdot t + l}{4 \cdot t - l}\right) \right)}, \quad (5.6)$$

де t - відстань від поверхні землі до середини заземлювача, м (приймаємо $t = 1,5 \text{ м}$).

$$R_B = \frac{100}{2 \cdot 3,14 \cdot 2,5 \cdot \left(\ln\left(2 \cdot \frac{2,5}{0,016}\right) + 0,5 \cdot \ln\left(\frac{4 \cdot 1,5 + 2,5}{4 \cdot 1,5 - 2,5}\right) \right)} = 1,03 \text{ Ом.} \quad (5.7)$$

Згідно з правилами улаштування електроустановок, опір заземлювального пристрою, до якого приєднано нейтраль джерела живлення або виводи джерела однофазного струму, у будь-яку пору року не повинен перевищувати 2, 4 і 8 Ом відповідно для лінійних напруг 660, 380 і 220 В джерела трифазного струму або 380, 220 і 127 В джерела однофазного струму. Отже, заземлення відповідає вимогам.

3) Розрахунок необхідного повітрообміну.

Вихідні дані:

- об'єм приміщення: $V = 150 \text{ м}^3$;
- кількість працюючих: $n = 2$ особи;
- категорія робіт: середньої важкості Пб.

Розрахунок повітрообміну за вмістом CO_2 .

Необхідний повітрообмін за вмістом вуглекислого газу визначається за формулою:

$$L = \frac{G \cdot n}{K_B - K_{\Pi}}, \quad (5.8)$$

де G - кількість CO_2 , що виділяється однією людиною при фізичному навантаженні, $\text{м}^3/\text{год}$ (для категорії робіт Пб приймаємо $G = 58 \frac{\text{л}}{\text{год}} = 0,058 \text{ м}^3/\text{год}$); K_B - гранично допустима концентрація CO_2 у повітрі робочої зони ($K_B = 1000 \text{ ppm}$); K_{Π} - концентрація CO_2 у приточному повітрі ($K_{\Pi} = 400 \text{ ppm}$).

$$L = \frac{0,058 \cdot 2}{0,001 - 0,0004} = 193,33 \frac{\text{м}^3}{\text{год}}. \quad (5.9)$$

Кратність повітрообміну.

Кратність повітрообміну визначається за формулою:

$$K = \frac{L}{V}, \quad (5.10)$$

$$K = \frac{193,33}{150} = 1,289 \approx 1,3. \quad (5.11)$$

Таким чином, необхідно забезпечити кратність повітрообміну не менше 1,3 об'ємів приміщення за годину.

Вибір вентиляційного обладнання.

Для забезпечення розрахованого повітрообміну рекомендується встановити припливно-витяжну вентиляційну систему з продуктивністю не менше 195 м³/год. Система повинна бути обладнана фільтрами для очищення повітря від пилу та регуляторами швидкості для підтримки оптимального мікроклімату.

4) Розрахунок освітлення.

Вихідні дані:

- розміри приміщення: довжина $A = 10$ м, ширина $B = 5$ м;
- висота приміщення: $H = 3$ м;
- висота робочої поверхні: $h = 0,8$ м;
- нормована освітленість: $E_n = 400$ лк;
- коефіцієнт запасу: $K_z = 1,5$.

Визначення висоти підвісу світильників.

Висота підвісу світильників над робочою поверхнею:

$$h = H - h_p - h_c, \quad (5.12)$$

де H - висота приміщення, м; h_p - висота робочої поверхні, м; h_c - відстань від стелі до світильників, м (приймаємо $h_c = 0,2$ м).

$$h = 3 - 0,8 - 0,2 = 2 \text{ м}. \quad (5.13)$$

Визначення індексу приміщення:

$$i = \frac{A \cdot B}{h \cdot (A+B)}, \quad (5.14)$$

$$i = \frac{10 \cdot 5}{2 \cdot (10+5)} = 1,67. \quad (5.15)$$

Визначення коефіцієнтів відбиття:

- коефіцієнт відбиття стелі: $\rho_{\text{п}} = 70\%$;
- коефіцієнт відбиття стін: $\rho_{\text{с}} = 50\%$;
- коефіцієнт відбиття робочої поверхні: $\rho_{\text{р}} = 30\%$.

Визначення коефіцієнта використання світлового потоку (η).

За індексом приміщення $i = 1,5$ та коефіцієнтами відбиття $\rho_{\text{п}} = 70\%$, $\rho_{\text{с}} = 50$, $\rho_{\text{р}} = 30\%$ для світлодіодних світильників коефіцієнт використання світлового потоку $\eta = 0,584$.

Розрахунок необхідного світлового потоку однієї лампи:

$$\Phi = \frac{E_{\text{н}} \cdot S \cdot K_{\text{з}} \cdot Z}{N \cdot \eta}, \quad (5.16)$$

де $E_{\text{н}}$ - нормована освітленість, лк; S - площа приміщення, м²; $K_{\text{з}}$ - коефіцієнт запасу; Z - коефіцієнт нерівномірності освітлення ($Z = 1,1$); N - кількість світильників, шт.; η - коефіцієнт використання світлового потоку.

Для приміщення розміром 10×5 м при відстані між світильниками 2 м кількість світильників складе:

$$N = \left(\frac{10}{2}\right) \cdot \left(\frac{5}{2}\right) = 5 \cdot 2,5 = 12,5 \approx 13 \text{ шт.} \quad (5.17)$$

$$\Phi = \frac{400 \cdot 50 \cdot 1,5 \cdot 1,1}{13 \cdot 0,584} = 4346,7 \text{ лм.} \quad (5.18)$$

За розрахованим світловим потоком обираємо світлодіодні світильники типу SL зі світловим потоком 4500 лм.

Перевірка.

Відхилення фактичної освітленості від нормованої:

$$\Delta E = \left(\frac{\Phi_{\text{факт}} - \Phi_{\text{розрах}}}{\Phi_{\text{розрах}}} \right) \cdot 100\%, \quad (5.19)$$

де $\Phi_{\text{факт}}$ - фактичний світловий потік обраної лампи, лм; $\Phi_{\text{розрах}}$ - розрахований світловий потік, лм.

5.4. Висновки за розділом

Проведений аналіз потенційних небезпек та шкідливих факторів при експлуатації системи керування роботом-маніпулятором дозволив виявити основні ризики для оператора та технічного персоналу. До найбільш суттєвих небезпек відносяться: механічний вплив рухомих частин робота, небезпека ураження електричним струмом, шум та вібрація від роботи електроприводів, психофізіологічні навантаження при керуванні системою.

Для зниження або усунення виявлених ризиків розроблено комплекс організаційних та технічних заходів з охорони праці. Особлива увага приділена технічним заходам безпеки, для яких проведено детальні розрахунки:

- розрахунок захисного огороження робочої зони робота-маніпулятора, що забезпечує безпечну відстань 0,745 м і висоту 0,469 м;
- розрахунок системи заземлення вертикальним електродом та з'єднувальною смугою, що забезпечує опір заземлення 1,03 Ом (при нормі 8 Ом);
- розрахунок необхідного повітрообміну, що показав потребу в кратності повітрообміну не менше 1,3 об'ємів приміщення за годину;
- розрахунок системи освітлення з використанням 13 світлодіодних світильників, що забезпечує нормовану освітленість 400 лк.

Розроблено також детальний режим праці та відпочинку для операторів системи керування роботом-маніпулятором, що включає обмеження тривалості безперервної роботи до 2 годин та регламентовані перерви різної тривалості та призначення.

Запропоновані заходи з охорони праці відповідають вимогам чинного законодавства України та дозволяють забезпечити безпечні умови праці для обслуговуючого персоналу системи керування роботом-маніпулятором. При дотриманні всіх розроблених рекомендацій можливо суттєво знизити ризики виробничого травматизму та професійних захворювань.

Особлива увага приділена організації робочого місця оператора з урахуванням ергономічних вимог, що сприяє зниженню втоми та підвищенню ефективності праці. Встановлено конкретні параметри робочого місця: висота пультів керування (72-78 см), оптимальні зони розміщення органів керування (25-45 см від оператора), кути розміщення засобів відображення інформації (15-30° нижче рівня очей).

Комплекс технічних заходів, включаючи системи блокування, датчики присутності, системи вентиляції та освітлення, розроблено з урахуванням специфіки системи керування роботом-маніпулятором та відповідає сучасним вимогам до забезпечення безпечних умов праці.

Розрахунки систем безпеки виконано згідно з діючими нормативними документами та стандартами, що гарантує їх ефективність при практичному застосуванні. Впровадження розроблених заходів дозволить створити безпечні та комфортні умови праці для операторів системи керування роботом-маніпулятором, забезпечити тривале збереження їх здоров'я та високу продуктивність праці.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено та експериментально досліджено систему керування роботом-маніпулятором із механізмом паралелограмного типу. В ході дослідження отримано наступні результати:

Проведено ґрунтовний аналіз конструктивних особливостей роботів-маніпуляторів із механізмом паралелограмного типу, що дозволило визначити його переваги порівняно з послідовними кінематичними структурами, зокрема, підвищену жорсткість конструкції, вищу точність позиціонування та кращі динамічні характеристики. Встановлено, що для вибору оптимальної конфігурації маніпулятора необхідно враховувати умови експлуатації, вимоги до точності, динамічні характеристики та вантажопідйомність.

Розроблено математичну модель робота-маніпулятора паралелограмного типу, яка дозволяє описати його кінематичні властивості та динаміку руху. Визначено, що досліджуваний механізм є конфігурацією Five-Bar Linkage з 2 ступенями вільності, встановленою на обертову платформу, що надає системі загалом 3 ступені вільності. Сформульовано та розв'язано пряму та обернену задачі кінематики, визначено робочий простір маніпулятора з урахуванням геометричних обмежень, діапазону приводів, самозіткнень та сингулярностей.

Досліджено різні методи інтерполяції траєкторій (трапецієподібні, поліноміальні, сплайнові) та встановлено, що для забезпечення плавного руху маніпулятора доцільно використовувати трапецієподібні траєкторії швидкості, поліноміальну інтерполяцію або Catmull-Rom сплайни з відповідними параметрами натягнутості, що дозволяє мінімізувати різкі зміни прискорення.

Розроблено програмну реалізацію системи керування роботом-маніпулятором, яка включає взаємопов'язані модулі ручного керування, поліноміальної інтерполяції, розв'язання зворотної задачі кінематики, встановлення кута серводвигунів, вимірювання дистанції та автоматичного

пошуку об'єктів. Реалізовано декілька режимів роботи: ручний, автоматичний (режим сканування та захоплення) та режим координатного позиціонування.

Створено алгоритм автоматичного пошуку об'єктів маніпулювання на основі радіального сканування в полярній системі координат з використанням ультразвукового датчика, що забезпечує високу ймовірність виявлення об'єктів (до 98% для об'єктів з низьким коефіцієнтом поглинання ультразвуку).

Проведено експериментальні дослідження розробленої системи, які підтвердили її працездатність та ефективність. Встановлено, що середня абсолютна похибка позиціонування становить приблизно ± 2 мм по осях X і Y та $\pm 3,4$ мм по осі Z, що є прийнятним для даного класу пристроїв. Виявлено, що найбільші відхилення спостерігаються поблизу меж робочої області, що обумовлено конструктивними особливостями та точністю зворотного зв'язку.

Розроблено комплекс організаційних та технічних заходів з охорони праці, який включає розрахунок захисного огородження робочої зони робота-маніпулятора, системи заземлення, необхідного повітрообміну та системи освітлення. Встановлено, що при дотриманні розроблених рекомендацій можливо суттєво знизити ризики виробничого травматизму та професійних захворювань при експлуатації системи керування роботом-маніпулятором.

Практична значущість розробленої системи полягає в можливості її використання для автоматизації процесів захоплення та переміщення об'єктів у різних галузях промисловості, що сприятиме підвищенню продуктивності праці та якості продукції.

Перспективними напрямками подальших досліджень є вдосконалення системи керування роботом-маніпулятором шляхом впровадження методів машинного навчання для оптимізації траєкторій руху, розробки більш досконалих алгоритмів розпізнавання об'єктів з використанням комп'ютерного зору та підвищення точності позиціонування за рахунок вдосконалення механічної частини та застосування більш точних сенсорів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ahn K.-H., Lee W.-B., Song J.-B. Reduction in Gravitational Torques of an Industrial Robot Equipped with 2 DOF Passive Counterbalance Mechanisms // *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*. Daejeon, Korea, October 9–14, 2016. – DOI: 10.1109/IROS.2016.7759639.
2. Черно О. О. *Системи управління маніпуляційних роботів: конспект лекцій*. Миколаїв: Національний університет кораблебудування імені адмірала Макарова, 2024.
3. Wang L., Zhang Z., Shao Z., Tang X. Analysis and optimization of a novel planar 5R parallel mechanism with variable actuation modes // *Mechanism and Machine Theory*. – 2016. – Vol. 99. – P. 124–139. – DOI: 10.1016/j.rcim.2018.09.010.
4. Erwin-Christian L., Valentin C., Tivadar D., Alexandru O., Elida-Gabriela T., Melania-Olivia S. Optimal Synthesis of Five-Bar Linkage Based on Singularity-Free Workspaces with Predefined Shapes // *Robotics*. – 2024. – Vol. 13(2). – P. 55. – DOI: 10.3390/robotics13120173.
5. Rapid Prototyping and Mechanical Design [Електронний ресурс]. – Режим доступу: <https://www.cs.cmu.edu/~rapidproto/mechanisms/chpt4.html> (дата звернення 17.05.2025).
6. Vathan L. B., Kumar H., John B. I. Kinematic Analysis of Five-Bar Mechanism in Industrial Robotics // *International Journal of Applied Engineering Research*. – 2015. – Vol. 10(24). – P. 20971-20976.
7. MG995 Servo Datasheet [Електронний ресурс]. – Режим доступу: <https://www.alldatasheet.com/datasheet-pdf/view/1132435/ETC2/MG995.html> (дата звернення 17.05.2025).

ДОДАТОК А

ПРИНЦИПОВА СХЕМА СИСТЕМИ КЕРУВАННЯ РОБОТОМ-МАНІПУЛЯТОРОМ

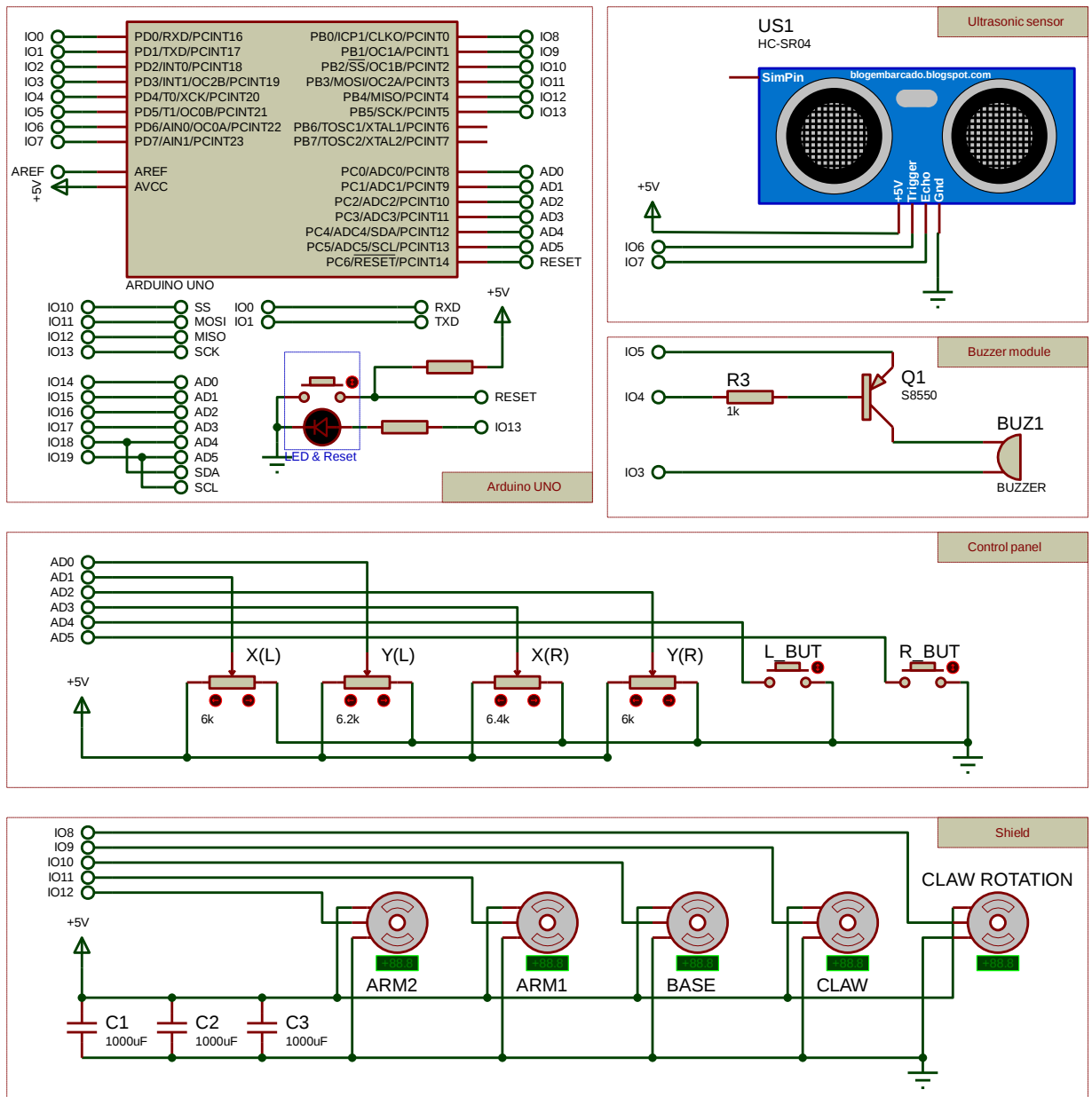


Рис. А1. Принципова схема системи керування роботом-маніпулятором

- ATmega328P (Arduino UNO)
- Модуль звукової індикації MH-FMD (Buzzer module)
- Модуль для живлення периферійних компонентів (Shield)
- Ультразвуковий датчик HC-SR04 (Ultrasonic sensor)

- Сервомотор SG90 (CLAW, CLAW ROTATION)
- Сервомотор MG995 (BASE, ARM1, ARM2)
- Джойстик KY-023 (X(L), X(R), Y(L), Y(R))
- Кнопка тактова (L_BUT, R_BUT)

ДОДАТОК Б

КОД ПРОГРАМИ СИСТЕМИ КЕРУВАННЯ РОБОТОМ- МАНІПУЛЯТОРОМ

Цифрові та аналогові входи/виходи мікроконтролера визначено на основі принципової електричної схеми (див. додаток А).

```
#include <Servo.h> // Бібліотека для роботи з сервоприводами
#include <avr/wdt.h> // Бібліотека для роботи з watchdog таймером

// Визначення пінів для сенсорів та серводвигунів
#define TRIGGER_PIN 6 // Пін тригера ультразвукового датчика
#define ECHO_PIN 7 // Пін ехо ультразвукового датчика
#define ROTATIONSERVOPIN 8 // Пін серводвигуна обертання захвату
#define GRIPSERVOPIN 9 // Пін серводвигуна захвату
#define BASESERVOPIN 10 // Пін серводвигуна основи
#define ARM1SERVOPIN 11 // Пін серводвигуна верхньої частини руки
#define ARM2SERVOPIN 12 // Пін серводвигуна нижньої частини руки

// Аналогові входи для потенціометрів та кнопок
const int lowerArmPotPin = A0; // Потенціометр нижньої частини руки
const int basePotPin = A1; // Потенціометр основи
const int upperArmPotPin = A2; // Потенціометр верхньої частини руки
const int rotationPotPin = A3; // Потенціометр обертання
const int openButtonPin = A4; // Кнопка відкриття захвату
const int closeButtonPin = A5; // Кнопка закриття захвату

// Створення об'єктів серводвигунів
Servo arm1servo; // Сервопривод верхньої частини руки
Servo arm2servo; // Сервопривод нижньої частини руки
Servo baseservo; // Сервопривод основи
Servo gripservo; // Сервопривод захвату
Servo rotationservo; // Сервопривод обертання

// Змінні для ручного керування
int lastRotationAngle = 80; // Останній кут повороту
int lastUpperArmAngle = 125; // Останній кут верхньої частини руки
int lastLowerArmAngle = 35; // Останній кут нижньої частини руки
int lastBaseAngle = 90; // Останній кут основи
unsigned long previousMillis = 0; // Час останнього оновлення кутів

// Змінні для координатного керування - поточна позиція робочого органу
```

```

float currentX = 0;
float currentY = 160;
float currentZ = 160;

String inputString = ""; // Рядок для зберігання команд з послідовного порту
bool stringComplete = false; // Прапорець завершення введення рядка

/*
 * Функція ручного керування маніпулятором через потенціометри
 * Зчитує стани кнопок та потенціометрів і застосовує відповідні кути до серводвигунів
 */
void manual_control() {
    // Зчитування станів кнопок для керування захватом
    int openButtonValue = analogRead(openButtonPin);
    int closeButtonValue = analogRead(closeButtonPin);

    // Керування захватом на основі натискання кнопок
    if (openButtonValue >= 1000) {
        gripservo.write(0); // Відкрити захват
    } else if (closeButtonValue >= 1000) {
        gripservo.write(67); // Закрити захват
    }

    // Перевірка часу для обмеження частоти зчитувань з потенціометрів
    unsigned long currentMillis = millis();
    if (currentMillis - previousMillis >= 10) { // Оновлення кожні 10 мс
        previousMillis = currentMillis;

        // Зчитування значень потенціометрів
        int upperArmValue = analogRead(upperArmPotPin);
        int lowerArmValue = analogRead(lowerArmPotPin);
        int baseValue = analogRead(basePotPin);
        int rotationValue = analogRead(rotationPotPin);

        // Оновлення кутів на основі зчитаних значень потенціометрів
        // Для кожного з'єднання перевіряємо крайні значення потенціометра
        // та інкрементуємо або декрементуємо відповідний кут

        // Обертання
        if (rotationValue == 1023) {
            lastRotationAngle = min(lastRotationAngle + 1, 180); // Збільшуємо з обмеженням до 180
        } else if (rotationValue == 0) {
            lastRotationAngle = max(lastRotationAngle - 1, 0); // Зменшуємо з обмеженням до 0
        }
    }
}

```

```

// Верхня частина руки
if (upperArmValue == 1023) {
    lastUpperArmAngle = min(lastUpperArmAngle + 1, 155); // Збільшуємо з обмеженням до 155
} else if (upperArmValue == 0) {
    lastUpperArmAngle = max(lastUpperArmAngle - 1, 35); // Зменшуємо з обмеженням до 35
}

// Нижня частина руки
if (lowerArmValue == 1023) {
    lastLowerArmAngle = min(lastLowerArmAngle + 1, 125); // Збільшуємо з обмеженням до 125
} else if (lowerArmValue == 0) {
    lastLowerArmAngle = max(lastLowerArmAngle - 1, 35); // Зменшуємо з обмеженням до 35
}

// Основа
if (baseValue == 1023) {
    lastBaseAngle = min(lastBaseAngle + 1, 180); // Збільшуємо з обмеженням до 180
} else if (baseValue == 0) {
    lastBaseAngle = max(lastBaseAngle - 1, 0); // Зменшуємо з обмеженням до 0
}

// Застосування оновлених кутів до серводвигунів
arm1servo.write(lastUpperArmAngle);
arm2servo.write(lastLowerArmAngle);
baseservo.write(lastBaseAngle);
rotation servo.write(lastRotationAngle);
}
}

/*
 * Функція обробки подій послідовного порту
 * Викликається автоматично при надходженні даних через послідовний порт
 */
void serialEvent() {
    while (Serial.available()) {
        char inChar = (char)Serial.read();
        if (inChar == '\n') { // Якщо отримано символ нового рядка - команда завершена
            stringComplete = true;
        } else {
            inputString += inChar; // Додаємо символ до рядка команди
        }
    }
}
}

```

```

/*
 * Функція обробки команд з послідовного порту
 * Розбирає отриману команду та викликає відповідні дії
 */
void processSerialCommand() {
    // Якщо рядок завершився (отримано символ нового рядка)
    if (stringComplete) {
        Serial.println("Recieved: " + inputString);

        // Розбити рядок на частини за пробілами
        int values[4]; // масив для збереження числових параметрів команди
        int index = 0;
        char* token = strtok((char*)inputString.c_str(), " ");
        while (token != NULL && index < 4) {
            values[index] = atoi(token); // Перетворення текстового фрагмента в число
            token = strtok(NULL, " ");
            index++;
        }

        // Обробка команди залежно від першого числа в рядку (код команди)
        switch (values[0]) {
            case 1: // Ручний режим
                Serial.println("Current mode: Manual");
                while (true) {
                    // Перевіряємо умову виходу з режиму - натиснуті обидві кнопки
                    if (analogRead(closeButtonPin) == 1023 && analogRead(openButtonPin) == 1023) {
                        Serial.println("Exiting manual mode...");
                        break;
                    }
                    manual_control(); // Виконуємо ручне керування
                }
                break;
            case 2: // Режим сканування та захоплення
                Serial.println("Current mode: Scan and Grab");
                while (true) {
                    // Перевіряємо умову виходу з режиму - натиснуті обидві кнопки
                    if (analogRead(closeButtonPin) == 1023 && analogRead(openButtonPin) == 1023) {
                        Serial.println("Exiting scan and grab mode...");
                        break;
                    }
                    scanAndGrab(); // Виконуємо сканування та захоплення
                }
                break;
            case 3: // Режим координатного керування
                Serial.println("Current mode: Coordinate");
                // Перевіряємо чи входить координата Z у допустимий діапазон

```

```

        if (values[3] >= -135 && values[3] <= 160) {
            Serial.println("Moving to x:" + String(values[1]) + " y:" + String(values[2]) + " z:" +
String(values[3]));
            // Рух до вказаних координат з використанням поліноміальної інтерполяції
            polynomialProfileMove(values[1], values[2], values[3]);
        } else {
            Serial.println("The 'z' value cannot be reached.");
        }
        break;
    case 4: // Керування кутом захвату
        gripservo.write(values[1]);
        Serial.println("Gripper is now set to " + String(values[1]));
        break;
    default: // Невідома команда
        Serial.println("Unknown command. Try again, please.");
        break;
}

// Очищаємо буфер для наступної команди
inputString = "";
stringComplete = false;
}
}

```

```

/*
 * Функція руху до заданої точки з використанням поліноміальної інтерполяції
 * Забезпечує плавний рух від поточної позиції до цільової
 * xEnd - Цільова координата X
 * yEnd - Цільова координата Y
 * zEnd - Цільова координата Z
 */
void polynomialProfileMove(float xEnd, float yEnd, float zEnd) {
    // Початкові координати - поточна позиція маніпулятора
    float xStart = currentX;
    float yStart = currentY;
    float zStart = currentZ;

    int steps = 100; // Кількість кроків для інтерполяції

    // Розрахунок проміжних точок та рух до них
    for (int i = 0; i <= steps; i++) {
        float t = (float)i / steps; // Нормалізований час (0..1)

        // Поліном 3-го ступеня для плавного прискорення/сповільнення
        // s = 3t2 - 2t3 (плавний S-подібний профіль)
    }
}

```

```

float s = 3 * t * t - 2 * t * t * t;

// Лінійна інтерполяція з використанням S-подібного профілю
float x = xStart + s * (xEnd - xStart);
float y = yStart + s * (yEnd - yStart);
float z = zStart + s * (zEnd - zStart);

moveToPos(x, y, z); // Рух до розрахованої проміжної точки
delay(10); // Затримка для плавності руху

// Оновлення поточної позиції
currentX = x;
currentY = y;
currentZ = z;
}
}

/*
* Функція розв'язання зворотної задачі кінематики
* Обчислює необхідні кути серводвигунів для досягнення заданої позиції в декартових координатах
* x - Координата X цілі
* y - Координата Y цілі
* z - Координата Z цілі
*/
void moveToPos(float x, float y, float z) {
    // Кут повороту основи (в градусах)
    float b = abs(atan2(y, x) * (180.0 / PI));

    // Відстань від центру основи до проекції цільової точки на горизонтальну площину
    float l = sqrt(x * x + y * y);

    // Відстань від центру основи до цільової точки у 3D просторі
    float h = sqrt(l * l + z * z);

    // Перевірка, чи знаходиться точка в межах робочого простору
    if (h <= 65 || h >= 310 || l <= 92 || l >= 310) {
        return; // Точка поза робочим простором, вихід з функції
    }

    // Розрахунок кутів ланок з використанням тригонометрії
    float alpha = atan(z / l); // Кут нахилу від горизонталі до лінії від основи до цілі
    float beta = acos((h / 2.0) / 160.0); // Половина кута дуги для двох однакових ланок

    // Кут першої ланки від горизонталі (в градусах)
    float a1 = (alpha + beta) * (180.0 / PI);

```

```

// Розрахунок кута другої ланки (в градусах)
float q2 = alpha - beta;
float xC = 1 - 200.0 * cos(q2); // X-координата з'єднання ланок
float zC = z - 200.0 * sin(q2); // Z-координата з'єднання ланок
float c = (-24000.0 + xC * xC + zC * zC) / 80.0;
float a2 = abs(2.0 * atan((-zC - sqrt(xC * xC + zC * zC - c * c)) / (-xC - c))) * (180.0 / PI);

// Застосування обчислених кутів
moveToAngle(b, a1, a2);
}

/*
* Функція встановлення кутів серводвигунів
* Перевіряє обмеження кутів та переводить абстрактні кути в реальні кути серводвигунів
* b - Кут основи
* a1 - Кут першої ланки
* a2 - Кут другої ланки
*/
void moveToAngle(float b, float a1, float a2) {
    // Перевірка, чи входять кути в допустимі діапазони
    if (a1 >= 0 && a1 <= 120 && a2 >= 70 && a2 <= 180) {
        // Перетворення абстрактних кутів у реальні кути серводвигунів
        int a11 = map(a1, 0, 120, 35, 155); // Перша ланка
        int a22 = map(a2, 90, 180, 125, 35); // Друга ланка

        // Встановлення кутів серводвигунів
        arm1servo.write(a11);
        arm2servo.write(a22);
        baseservo.write(b); // Кут основи не потребує перетворення
    } else {
        // Виведення помилки, якщо кути виходять за допустимі межі
        Serial.println("Error: angles out of bounds!");
        Serial.println("b " + String(b) + " a1 " + String(a1) + " a2 " + String(a2));
    }
}

/*
* Функція вимірювання відстані з використанням ультразвукового датчика
* distance - Відстань у міліметрах
*/
float measureDistance() {
    // Генерація ультразвукового імпульсу
    digitalWrite(TRIGGER_PIN, LOW);

```

```

delayMicroseconds(2); // Коротка пауза для стабілізації
digitalWrite(TRIGGER_PIN, HIGH);
delayMicroseconds(10); // Тривалість імпульсу
digitalWrite(TRIGGER_PIN, LOW);

// Вимірювання часу до отримання відбитого сигналу
long duration = pulseIn(ECHO_PIN, HIGH);

// Розрахунок відстані в міліметрах
// Швидкість звуку 0.034 см/мкс, ділимо на 2 (туди-назад)
// Множимо на 10 для переведення у міліметри
float distance = (duration * 0.0343 / 2) * 10;
return distance;
}

/*
 * Функція сканування та захоплення об'єктів
 * Виконує радіальне сканування простору з використанням ультразвукового датчика
 * При виявленні об'єкта виконує його захоплення
 */
void scanAndGrab() {
    static int currentPosition = 0; // Поточна позиція сканування
    const int num_points = 30; // Кількість точок сканування
    const float l = 100.0; // Радіус сканування
    const float min_radius = 92.0; // Мінімальний радіус

    // Переміщення до поточної позиції
    moveToPos(currentX, currentY, currentZ);
    delay(400); // Затримка для завершення переміщення

    if (currentPosition < (num_points - 3)) {
        // Розрахунок кута для радіального сканування (рівномірно розподілені точки)
        float theta = PI - (PI * currentPosition / (num_points - 1));

        // Розрахунок координат X і Y
        float x = l * cos(theta);
        float y = l * sin(theta);
        float r = sqrt(x * x + y * y);

        // Корекція позиції, якщо вона виходить за мінімальний радіус
        if (r < min_radius) {
            float correction = min_radius / r;
            x *= correction;
            y *= correction;
        }
    }
}

```

```

// Оновлення поточної позиції для сканування
currentX = x;
currentY = y;
currentZ = -80.0; // Фіксована висота сканування

// Вимірювання відстані до об'єкта
float distance = measureDistance();

// Перевірка, чи виявлено об'єкт в межах досяжності
if (distance > 0 && distance <= 210) {
    // Розрахунок координат об'єкта на основі вимірної відстані
    float L = sqrt(currentX * currentX + currentY * currentY);
    float objX = currentX / L * (L + distance);
    float objY = currentY / L * (L + distance);
    float objZ = currentZ;

    // Вивід інформації про виявлений об'єкт
    Serial.print("Object detected! Moving to x: ");
    Serial.print(objX);
    Serial.print(" y: ");
    Serial.println(objY);

    // Послідовність дій для захоплення об'єкта
    polynomialProfileMove(objX, objY, objZ); // Рух до об'єкта
    delay(500);

    gripservo.write(85); // Закриття захвату
    delay(1000);

    polynomialProfileMove(objX, objY, 160); // Підйом об'єкта
    delay(500);

    polynomialProfileMove(200, 0, 160); // Рух до зони вивантаження
    delay(500);

    polynomialProfileMove(200, 0, -80); // Спуск об'єкта
    delay(500);

    gripservo.write(0); // Відкриття захвату (випуск об'єкта)
    delay(1000);

    polynomialProfileMove(100, 0, -80); // Рух до центру
    delay(500);

    polynomialProfileMove(currentX, currentY, currentZ); // Повернення до позиції сканування

```

```

        delay(1000);

        currentPosition = 0; // Скидання позиції для початку нового циклу сканування
    }

    currentPosition++; // Перехід до наступної позиції сканування
} else {
    currentPosition = 0; // Скидання позиції після завершення повного циклу
}
}

/*
 * Функція для відтворення звукового сигналу готовності
 * Використовує цифрові піни для генерації звукового сигналу
 */
void readySFX() {
    pinMode(5, OUTPUT);
    pinMode(4, OUTPUT);
    pinMode(3, OUTPUT);
    digitalWrite(5, HIGH);
    digitalWrite(3, LOW);

    // Генерація послідовності коротких сигналів
    digitalWrite(4, LOW);
    delay(80);
    digitalWrite(4, HIGH);
    delay(80);
    digitalWrite(4, LOW);
    delay(80);
    digitalWrite(4, HIGH);
    delay(80);
    digitalWrite(4, LOW);
    delay(80);
    digitalWrite(4, HIGH);
    delay(80);
}

/*
 * Функція для відтворення звукового сигналу вимкнення
 * Використовує цифрові піни для генерації звукового сигналу
 */
void offSFX() {
    pinMode(5, OUTPUT);
    pinMode(4, OUTPUT);

```

```

pinMode(3, OUTPUT);
digitalWrite(5, HIGH);
digitalWrite(3, LOW);

// Генерація послідовності довгих сигналів
digitalWrite(4, LOW);
delay(180);
digitalWrite(4, HIGH);
delay(120);
digitalWrite(4, LOW);
delay(180);
digitalWrite(4, HIGH);
}

/*
 * Функція для програмного перезавантаження мікроконтролера
 * Використовує watchdog таймер для виконання скидання
 */
void softReset() {
    wdt_enable(WDTO_15MS); // Увімкнути watchdog на 15 мс
    while (1) {}           // Очікувати скидання (зациклювання викличе спрацювання watchdog)
}

/*
 * Функція ініціалізації, викликається один раз при запуску
 * Налаштовує серводвигуни, сенсори та послідовний порт
 */
void setup() {
    // Ініціалізація серводвигунів з налаштуванням імпульсів для точного контролю
    baseservo.attach(BASESERVOPIN, 700, 2250);
    arm1servo.attach(ARM1SERVOPIN, 700, 2250);
    arm2servo.attach(ARM2SERVOPIN, 700, 2250);
    gripservo.attach(GRIPSERVOPIN, 700, 2250);
    rotationservo.attach(ROTATIONSERVOPIN, 700, 2250);

    // Початкове позиціонування
    gripservo.write(0); // Відкритий захват
    rotationservo.write(lastRotationAngle); // Встановлення початкового кута обертання
    polynomialProfileMove(currentX, currentY, currentZ); // Рух у початкову позицію
    delay(100);

    // Ініціалізація ультразвукового датчика
    pinMode(TRIGGER_PIN, OUTPUT);
    pinMode(ECHO_PIN, INPUT);

```

```

// Ініціалізація послідовного порту
Serial.begin(115200); // Висока швидкість для швидкого обміну даними
inputString.reserve(50); // Резервування пам'яті під командний рядок

// Виведення інформації про доступні команди
Serial.println("Robotic Arm Control System");
Serial.println("Commands:");
Serial.println("1 - Manual mode");
Serial.println("2 - Scanning mode");
Serial.println("3 X Y Z - Coordinate mode");
Serial.println("4 degree - Set gripper angle");

readySFX(); // Відтворення звукового сигналу готовності
}

/*
 * Головний цикл програми, виконується нескінченно
 */
void loop() {
    processSerialCommand(); // Обробка команд з послідовного порту

    // Перевірка комбінації кнопок для перезавантаження
    if (analogRead(closeButtonPin) == 1023 && analogRead(openButtonPin) == 1023) {
        delay(100); // Захист від випадкового спрацювання
        offSFX(); // Звуковий сигнал вимкнення
        Serial.println("Reset...");
        softReset(); // Програмне перезавантаження мікроконтролера
    }
}

```