

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет електроенерготехніки та автоматики

(повне найменування інституту, факультету)

Кафедра автоматизації електромеханічних систем та електроприводу

(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

_____ **Сергій КОВБАСА**
(підпис)

“ ____ ” _____ 20__ р.

Дипломний проект

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Електромеханічні системи
автоматизації, електропривод та електромобільність»
спеціальності 141 «Електроенергетика, електротехніка та електромеханіка»**
(код і назва)

на тему: **«Стенд для досліджень кінематики руху роботів-маніпуляторів»**

Виконав: студент 4 курсу, групи _____ **ЕП-02**
(шифр групи)

_____ **Черняєв Микита Олегович** _____
(прізвище, ім'я, по батькові) (підпис)

Керівник доц., к.т.н. Стаценко Олексій Володимирович _____
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по-батькові) (підпис)

Консультант _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ім'я, по-батькові) (підпис)

Рецензент асист. Ігнатюк Є.С. _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по-батькові) (підпис)

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студент (-ка) _____
(підпис)

Київ – 2024 р.

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЕКТУ

[illegible]

				141.0013.025.БР		
	ПІБ	Підп.	Дата			
Розробн.	Черняєв М.О.			Відомість дипломного проекту	Лист	Листів
Керівн.	Стаценко О.В.				2	
Консуль т.					КПІ ім. Ігоря Сікорського Каф. АЕМС-ЕП Гр. ЕП-02	
Н/контр.						
Зав.каф.	Ковбаса С.М.					

Пояснювальна записка до дипломного проекту

на тему: Стенд для досліджень кінематики руху роботів-маніпуляторів

Київ – 2024 р.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет електроенерготехніки та автоматики
(повна назва)

Кафедра автоматизації електромеханічних систем та електроприводу
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 141 Електроенергетика, електротехніка та електромеханіка
(код і назва)

Освітньо-професійна програма – Електромеханічні системи автоматизації, електропривод та електромобільність

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Сергій КОВБАСА
(підпис)

«__» _____ 20__ р.

ЗАВДАННЯ
на дипломний проект студенту

Черняєву Микиті Олеговичу
(прізвище, ім'я, по батькові)

1. Тема проекту Стенд для досліджень кінематики руху роботів-маніпуляторів,

керівник проекту Стаценко Олексій Володимирович, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__» _____ 20__ р. № _____

2. Термін подання студентом проекту 15.06.2024.

3. Вихідні дані до проекту: стенд повинен забезпечувати відпрацювання траєкторій руху при позиційному керуванні маніпулятором, завдання якої здійснюється через узагальнені координати програмним шляхом після розв'язку задач кінематики.

4. Зміст пояснювальної записки: 1 Аналітичний огляд роботів-маніпуляторів та стендів для дослідження їх роботи; 2. Структура та вибір складових лабораторного стенду; 3. Алгоритм роботи з лабораторним стендом та опис програмних складових; 4. Математична модель робота-маніпулятора у середовищі Simulink.

5. Перелік графічного матеріалу (із зазначенням обов’язкових креслеників, плакатів, презентацій тощо): Креслення загального виду стенду – А1; Схема функційна стенду – А1; Результати досліджень роботи стенду – А1.

6. Консультанти розділів проекту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

Якщо визначені консультанти. Консультантом не може бути зазначено керівника дипломного проекту.

7. Дата видачі завдання 15.04.2024

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Аналітичний огляд роботів-маніпуляторів та стендів для дослідження кінематики	30.04.2024	
2	Розробка структури та вибір складових лабораторного стенду	05.05.2024	
3	Розробка алгоритму роботи з лабораторним стендом та розробка програмних складових	20.05.2024	
4	Розробка математичної моделі робота-маніпулятора у середовищі Simulink та дослідження роботи	31.05.2024	
5	Оформлення пояснювальної записки, графічних матеріалів та підготовка презентації	15.06.2024	

Студент

(підпис)

Микита ЧЕРНЯЄВ

Керівник проекту

(підпис)

Олексій СТАЦЕНКО

РЕФЕРАТ

Дипломний проект виконаний на 66 сторінках, вміщує 45 рисунків, 3 таблиці та 3 графічних матеріалів у форматі А1.

Метою бакалаврської роботи є розробка лабораторного стенду для дослідження кінематики руху роботів маніпуляторів.

Було проведено аналітичний огляд вітчизняних та зарубіжних лабораторних стендів, що використовуються для досліджень та навчальних цілях. Було створено структурну схему лабораторного стенда на основі робота-маніпулятора.

На основі структурної схеми було обрано складові лабораторного стенду та створено програмне забезпечення для кожної з частини лабораторного стенду. Розроблено методичні вказівки для роботи з стендом.

Базуючись на структурній схемі робота-маніпулятора було розроблено математичну модель лабораторного стенду у середовищі Simulink.

Ключові слова: ЛАБОРАТОРНИЙ СТЕНД, РОБОТ МАНІПУЛЯТОР, КІНЕМАТИКА РОБОТІВ-МАНІПУЛЯТОРІВ, ОПТОРОВ'ЯЗКА, СЕРВОПРИВОД, RASPBERRYPI ZERO, SSH, ОДНОПЛАТНИЙ КОМП'ЮТЕР, STM32, UART, PWM, МОДЕЛЮВАННЯ, МАТЕМАТИЧНА МОДЕЛЬ, MATLAB, SIMULINK.

					141.0013.025.БР				
Змн.	Арк.	№ докум.	Підпис	Дата	Стенд для досліджень кінематики руху роботів-маніпуляторів	Літ.	Арк.	Акрушів	
Розроб.		Черняєв М.О.							
Перевір.		Стаценко О.В.							
Реценз.		Ігнатюк Є.С.				НТУУ «КПІ ім. Ігоря Сікорського» ФЕА, кафедра АЕМС-ЕП, м. Львів			
Н. Контр.									
Затверд.									

SUMMARY

The diploma project is 66 pages long, contains 45 figures, 3 tables and 3 graphic materials in A1 format.

The purpose of the bachelor's thesis is to develop a laboratory bench for studying the kinematics of robot manipulators.

An analytical review of domestic and foreign laboratory stands used for research and educational purposes was conducted. A structural diagram of a laboratory bench based on a robot manipulator was created.

Based on the structural diagram, the components of the laboratory bench were selected and software was created for each part of the laboratory bench. Guidelines for working with the stand were developed.

Based on the structural diagram of the robot manipulator, a mathematical model of the laboratory bench was developed in the Simulink environment.

Keywords: LABORATORY BENCH, ROBOT MANIPULATOR, KINEMATICS OF ROBOT MANIPULATORS, OPTICAL INTERCONNECTION, SERVO, RASPBERRYPI ZERO, SSH, SINGLE-BOARD COMPUTER, STM32, UART, PWM, MODELING, MATHEMATICAL MODEL, MATLAB, SIMULINK.

					141.0013.025.БР				
Змн.	Арк.	№ докум.	Підпис	Дата	Стенд для досліджень кінематики руху роботів-маніпуляторів				
Розроб.		Черняєв М.О.							
Перевір.		Стаценко О.В.							
Реценз.		Ігнатюк Є.С.							
Н. Контр.									
Затверд.					Літ. Арк. Акрушів НТУУ «КПІ ім. Ігоря Сікорського» ФЕА, кафедра АЕМС-ЕП, зп. ЕП-02				

ЗМІСТ

ВСТУП	10
ПЕРЕЛІК СКОРОЧЕНЬ.....	12
1 АНАЛІТИЧНИЙ ОГЛЯД РОБОТІВ-МАНІПУЛЯТОРІВ ТА СТЕНДІВ ДЛЯ ДОСЛІДЖЕННЯ ЇХ РОБОТИ	13
1.1 Загальна інформація про маніпулятори та промислові роботи.....	13
1.2 Основні етапи розвитку робототехніки	13
1.3 Класифікація маніпуляторів та промислових роботів	16
1.4 Зарубіжні роботи-маніпулятори що використовуються для досліджень та навчання	19
1.5 Роботи-маніпулятори що використовуються для досліджень та навчання в ЗВО України	23
1.6 Постановка задач дипломного проекту	27
2 СТРУКТУРА ТА ВИБІР СКЛАДОВИХ ЛАБОРАТОРНОГО СТЕНДУ	29
2.1 Принцип дії та загальна структурна схема лабораторного стенду	29
2.2 Електромеханічна частина лабораторного стенду	31
2.2.1 Сервопривод MG996R.....	32
2.2.2 Сервопривод DS3218MG.....	34
2.3 Пристрій керування сервоприводами	36
2.4 Блок розрахунку траєкторій.....	39
2.5 Гальванічна розв'язка	40
3 АЛГОРИТМ РОБОТИ З ЛАБОРАТОРНИМ СТЕНДОМ ТА ОПИС ПРОГРАМНИХ СКЛАДОВИХ	43
3.1 Програмне забезпечення для МК STM32.....	43
3.2 Застосунок для обробки вхідних даних	48
3.3 Програмне забезпечення для одноплатного ПК RaspberryPi	54
3.4 Інструкція до використання стенду.....	57

4 МАТЕМАТИЧНА МОДЕЛЬ РОБОТА-МАНІПУЛЯТОРА У СЕРЕДОВИЩІ SIMULINK	58
4.1 Математичний опис робота маніпулятора	58
4.2 Дослідження роботи робота-маніпулятора з використанням математичної моделі.....	62
ЗАГАЛЬНІ ВИСНОВКИ	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	70

ВСТУП

Роботи-маніпулятори відіграють важливу роль у сучасному виробництві, забезпечуючи високу точність, ефективність та безпеку робочих процесів. Вони здатні виконувати складні завдання з великою швидкістю і безперервністю, що значно підвищує продуктивність і якість вироблення продукції. Застосування роботів-маніпуляторів дозволяє автоматизувати такі процеси, як зварювання, складання, фарбування, упаковка, сортування та багато інших. Це знижує витрати на виробництво, мінімізує помилки, спричинені людським фактором, і забезпечує більш гнучке реагування на змінні вимоги ринку.

Водночас, навчальні роботи-маніпулятори є важливими для освіти і досліджень, сприяючи розвитку навичок і знань студентів у галузі робототехніки та автоматизації. Вони дозволяють студентам отримувати практичний досвід, експериментувати з різними алгоритмами керування, програмуванням та інтеграцією систем. Такі роботи часто використовуються в університетах та технічних закладах освіти для підготовки інженерів і техніків, які будуть працювати з промисловими роботами у майбутньому.

Навчальні роботи-маніпулятори дозволяють навчати робітників, інженерів і техніків роботі з сучасними автоматизованими системами, що є важливим аспектом у підтримці високих стандартів виробництва. Крім того, такі роботи можуть використовуватися для підвищення кваліфікації працівників і адаптації їхніх навичок до нових технологічних вимог.

У сфері наукових досліджень навчальні роботи-маніпулятори відкривають нові можливості для інновацій. Вони використовуються для розробки нових технологій, тестування експериментальних рішень і проведення наукових досліджень у галузі робототехніки. Це сприяє швидкому розвитку нових методів та технологій, які можуть бути впроваджені в промислове виробництво, підвищуючи його ефективність і конкурентоспроможність.

Таким чином можна стверджувати, що роботи-маніпулятори є невід'ємною частиною сучасного виробництва, забезпечуючи високий рівень автоматизації та ефективності. Їх застосування у навчальних цілях є важливим для розвитку фахівців, інновацій і наукових досліджень. Вони дозволяють підготувати кваліфікованих працівників і стимулювати технологічний прогрес, що є ключовим фактором успіху в умовах швидко змінюваного світу.

Тому основною метою даного дипломного проєкту є розробка стенду для дослідження кінематики руху роботів-маніпуляторів. Для досягнення поставленої мети необхідно вирішити ряд задач, а саме: провести аналітичний огляд роботів маніпуляторів та навчальних стендів на їх основі, розробити структуру та провести вибір складових лабораторного стенду, розробити програмне забезпечення для кожної з частин лабораторного стенду.

Вирішення цих задач дозволить створити лабораторний стенд, який може бути використаний для перевірки правильності вирішення задач кінематики руху роботів-маніпуляторів.

ПЕРЕЛІК СКОРОЧЕНЬ

РО – Робочий орган.

РМ – Робот-маніпулятор.

МК – Мікроконтролер.

ПЗ – Програмне забезпечення.

RPZ – RaspberryPi Zero.

1 АНАЛІТИЧНИЙ ОГЛЯД РОБОТІВ-МАНІПУЛЯТОРІВ ТА СТЕНДІВ ДЛЯ ДОСЛІДЖЕННЯ ЇХ РОБОТИ

1.1 Загальна інформація про маніпулятори та промислові роботи

Сучасні роботи-маніпулятори є одними з основних компонентів виробничих систем, підйомно-транспортного обладнання та логістичних систем.

На сучасних виробництвах все найчастіше знаходять своє використання промислові роботи, які здатні незалежно від людини виконувати досить складні завдання, що пов'язані з переміщенням та транспортуванням різних об'єктів, виконанням технологічних функцій, здійсненням функцій контролю та обслуговування.

Промисловий робот визначається як багатоцільовий маніпуляційний робот, до складу якого входять механічний маніпулятор перепрограмованої системи керування, який застосовується для переміщення об'єктів в просторі трьох і більше координат.

Згідно з ДСТУ2879-94 промисловий робот – це автоматична машина, стаціонарна чи пересувна, з виконавчим пристроєм у вигляді маніпулятора, який має декілька ступенів рухомості, і перепрограмовуваним пристроєм програмного керування для виконання у виробничому процесі рухових і керувальних функцій.

Маніпулятор – це керований пристрій або машина для виконання рухових функцій, аналогічних функціям руки людини при переміщенні об'єктів в просторі, оснащене робочим органом [1].

1.2 Основні етапи розвитку робототехніки

Ідеї створення машин, подібних до живих істот, можна простежити ще з часів стародавніх цивілізацій. Стародавні греки, єгиптяни та китайці створювали міфи та будували перші маніпулятори. Наприклад, давньогрецький інженер Герон Александрійський створив механічні пристрої, що працювали від води, вітру чи пари, одним з яких був

автоматизований ляльковий театр, здатний розігравати п'єсу тривалістю 10 хвилин. Аналогічно, у Стародавньому Китаї винахідник і математик Лян Цзи побудував рухому фігуру у формі людини для імператора.

Значний стрибок у концепції робототехніки зробив Леонардо да Вінчі близько 1495 року. Він сконструював механічного лицаря, що наведено на рис.1.1, відомого як робот Леонардо, здатного стояти, сидіти, рухати руками і відкривати щелепу. Цей роботизований лицар приводився в рух за допомогою серії шківів і кабелів, демонструючи інноваційне мислення епохи Відродження у створенні людиноподібних машин [2].

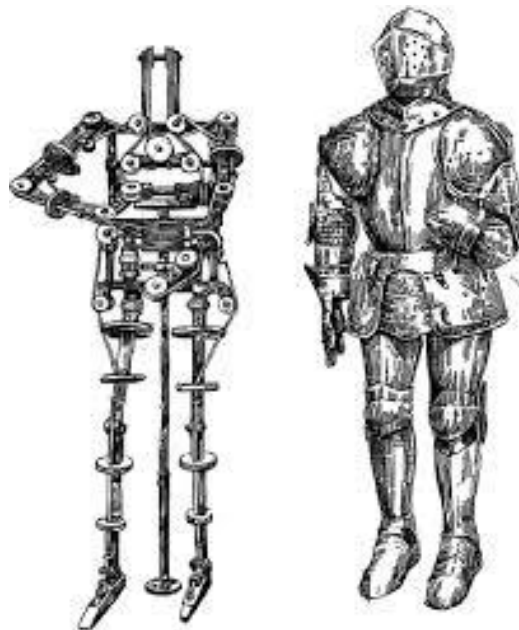


Рисунок 1.1 – Робот Леонардо

Промислова революція ознаменувала поворотний момент переходу від ручної праці до механізації, створивши підґрунтя для розвитку сучасної робототехніки. Саме в цю епоху був закладений фундамент сучасних роботів, коли були винайдені машини, призначені для автоматизації завдань, які раніше виконувалися вручну.

Сам термін "робот" був введений в публічний лексикон у 1920 році чеським письменником Карелом Чапеком у його п'єсі "R.U.R." ("Універсальні роботи Россам"). У цій п'єсі, що змальовувала майбутнє, штучні люди, яких називали "роботами", виконували роботу для людей.

Слово "робот" походить від чеського слова "robota", що означає "примусова праця", відображаючи теми дегуманізації та експлуатації роботів у п'єсі.

Поява перших програмованих машин у середині 20-го століття ознаменувала справжній початок робототехніки. Ці машини, хоча і були примітивними за сьогоdnішніми стандартами, заклали основу для складних роботів, яких ми маємо сьогодні.

Перший промисловий робот Unimate, що показано на рис.1.2, з'явився на конвеєрі General Motors у 1961 році. Розроблений для виконання небезпечних та повторюваних завдань, він поклав початок роботизації виробництва. Успіх Unimate призвів до широкого впровадження роботів у різних галузях промисловості [2].

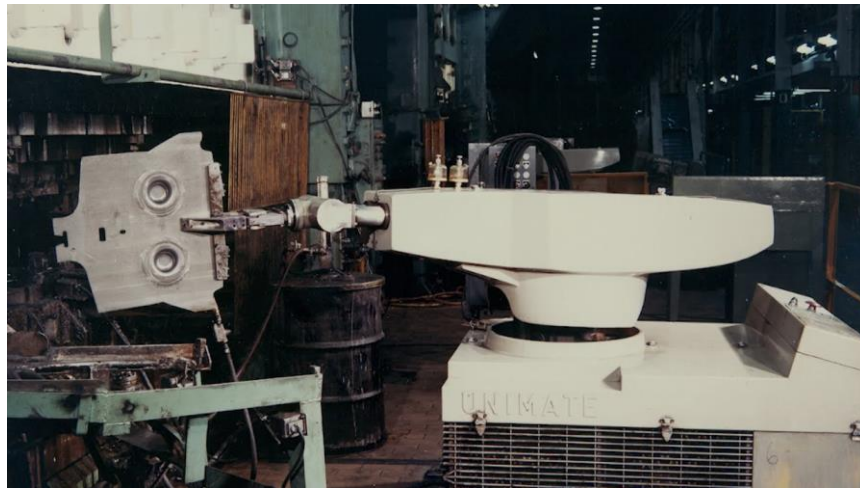


Рисунок 1.2 – Перший індустріальний робот

Досягнення в галузі інформатики та штучного інтелекту в другій половині 20-го століття сприяли подальшому розвитку робототехніки. Роботи стали більш досконалими, здатними виконувати складні завдання з більшою автономністю. Дослідження космосу також отримало користь від робототехніки: такі роботи, як зонди НАСА "Вояджер" і марсохід "Соджорнер", розширили наше розуміння Всесвіту.

Інтеграція штучного інтелекту з робототехнікою призвела до революційних розробок у 21 столітті. Роботи тепер можуть навчатися, адаптуватися та взаємодіяти з навколишнім середовищем у спосіб, який раніше було неможливо уявити. Людиноподібні роботи, автономні

транспортні засоби та дрони – це вже не наукова фантастика, а частина нашої реальності, яка виконує різні функції: від спілкування і транспортування до спостереження і доставки.

Роль робототехніки в охороні здоров'я була особливо визначною: хірургічні роботи, реабілітаційні пристрої та автономні системи моніторингу покращують догляд за пацієнтами. Роботи також стали частиною повсякденного життя, допомагаючи у виконанні різноманітних завдань - від пирососу до сфери розваг [2].

1.3 Класифікація маніпуляторів та промислових роботів

За ступенем досконалості і типами систем керування промислові роботи поділяються на три покоління:

- Перше покоління – програмні роботи (промислові), які працюють за жорсткими програмами. Вони широко застосовуються в машинобудуванні.
- Друге покоління – адаптивні роботи, які працюють за гнучкими програмами і змінюють характер робіт в залежності від зміни параметрів зовнішнього середовища.
- Третє покоління – інтелектуальні роботи, які здатні сприймати і логічно оцінювати навколишні обставини і визначити рухи, необхідні для досягнення заданої мети роботи [1].



Рисунок 1.3 – Ознаки класифікації промислових роботів

Промислові роботи класифікуються за такими вказаними ознаками, що наведені на рис.1.3:

- 1) Характер виконуваних операцій:
 - a) технологічні (виробничі);
 - b) допоміжні (підйомно-транспортні);
 - c) універсальні.
- 2) Ступінь спеціалізації:
 - a) спеціальні;
 - b) спеціалізовані;
 - c) багатоцільові;
- 3) Галузь застосування і вид виробництва:
 - a) ливарні;
 - b) нанесення покриттів;
 - c) ковальсько-пресові;
 - d) автоматичного контролю;
 - e) зварювальні;
 - f) складальні;
 - g) механічної обробки;
 - h) транспортно-складські роботи;
 - i) термообробка;
 - j) інші;
- 4) Системи основних координатних переміщень:
 - a) прямокутні (плоскі і просторові);
 - b) полярні (плоскі, циліндричні чи сферичні);
 - c) ангулярні (плоскі, циліндричні чи сферичні);
- 5) Кількість ступенів рухомості:
 - a) роботи з двома ступенями рухомості;
 - b) роботи з трьома ступенями рухомості;
 - c) роботи з чотирма ступенями рухомості;
 - d) роботи з ступенями рухомості більше чотирьох;
- 6) Вантажопідйомність:

- a) надлегкі (до 10Н);
- b) легкі (до 100Н);
- c) середні (до 2000Н);
- d) важкі (до 1000 Н);
- e) надважкі (понад 10000Н);

7) Мобільність:

- a) стаціонарні (убудовані в обладнання, підлогові і підвісні);
- b) пересувні (підлогові і підвісні);

8) Конструктивне виконання:

- a) вбудовані в устаткування;
- b) підлогові;
- c) підвісні;

9) Тип приводу:

- a) електромеханічні;
- b) пневматичні;
- c) гідравлічні;
- d) комбіновані;

10) Схема розташування приводів:

- a) в єдиному блоці;
- b) на виконавчих органах;

11) Вид відпрацьовування програми:

- a) жорстко програмувальні;
- b) адаптивні;
- c) гнучко програмувальні;

12) Вид систем керування:

- a) позиційні;
- b) контурні;
- c) комбіновані [1].

1.4 Зарубіжні роботи-маніпулятори що використовуються для досліджень та навчання

Окремо варто виокремити категорію роботів-маніпуляторів, що використовуються для освітніх цілей. Зробивши аналітичний огляд цієї сфери, можна виокремити 7 найрозповсюдженіших зарубіжних роботів-маніпуляторів:

1. Universal Robots UR5



Рисунок 1.4 – Зовнішній вигляд робота UR5

UR5 від компанії Universal Robots (Данія), був створений в 2005 році. Компанія спеціалізується на розробці та виробництві промислових роботів-маніпуляторів. UR5 використовується для роботи з маленькими предметами, монтажу, упаковці, обробці матеріалів, а також у наукових дослідженнях та в освітніх цілях. В залежності від модифікації робот коштує від \$35,000 до \$50,000. До переваг цього робота можна віднести низьку вагу, високу гнучкість, точність та безпечність [3].

2. ABB IRB 120



Рисунок 1.5 – Зовнішній вигляд робота IRB 120

Виготовлений у Швейцарії, компанією ABB, робот маніпулятор IRB 120 використовується для складних завдань, таких медичні операції та наукові дослідження. До переваг можна віднести компактний розмір, швидкість та точність, можливість монтажу на стіни або стельові конструкції. Однак можливі проблеми з технічним обслуговуванням через складність конструкції. Ціна цього робота становить \$25,000 - \$40,000 [4].

3. Fanuc LR Mate 200iD



Рисунок 1.6– Зовнішній вигляд робота

Робот-маніпулятор LR Mate 200iD від Fanuc Corporation виготовлений у Японії. Цей робот використовується в імітаційних наукових дослідках. До переваг LR Mate 200iD можна віднести швидкість та точність, компактний розмір, можливість роботи в обмеженому просторі. Ціна маніпулятора становить приблизно 30,000\$ [5].

4. KUKA KR6R700



Рисунок 1.7 – Зовнішній вигляд робота

Робот-маніпулятор KR 6 R700 від KUKA Robotics Corporation виготовлений у Німеччині. Цей робот використовується для завдань, які вимагають високу прецизійність. Цю модель застосовують у конвеєрних лініях на заводах BMW для монтажу та обробки матеріалів та для наукових досліджень у галузі матеріалознавства. До переваг KR 6 R700 можна віднести високий клас точності, надійність, легкий монтаж та просте програмування. Однак робот має високу ціну, що становить \$40,000 - \$60,000 [6].

5. Yaskawa Motoman MH5

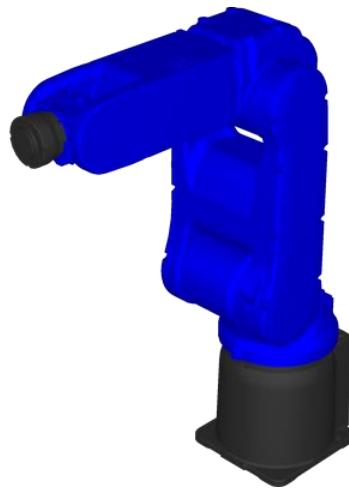


Рисунок 1.8 – Зовнішній вигляд робота

Робот-маніпулятор MH5 від Yaskawa Electric Corporation виготовлений у Японії. Цей робот використовується для досліджень у галузі моторобудування. До переваг MH5 відносять точність та стабільність [7].

6. Denso Vs-050



Рисунок 1.9 – Зовнішній вигляд робота

Робот-маніпулятор VS-050 від DENSO Corporation виготовлений у Японії. Маніпулятор використовується для моделювання кінематики руху при проведенні досліджень у галузі робототехніки. Цей маніпулятор зарекомендував себе як надійна та довговічна експериментальна машина, однак через свою ціну (30,000\$) він не дуже розповсюджена [8].

7. Mitsubishi MELFA RV-35



Рисунок 1.10 – Зовнішній вигляд робота

Робот-маніпулятор RV-3S від Mitsubishi Electric Corporation виготовлений у Японії. Маніпулятор здебільшого використовують у галузі фармацевтики, а саме для автоматизації виробництва ліків та фармакологічних дослідженнях. Недоліками цього робота є складність у програмуванні та висока ціна, що робить цей маніпулятор досить нішевим рішенням [9].

Як ми бачимо, усі наявні зарубіжні роботи-маніпулятори є надзвичайно дорогими та дуже складними в налаштуванні та експлуатації. Також варто відзначити, що на зарубіжному ринку не існує готового та комплектного лабораторного стенду з роботом-маніпулятором, що можна використовувати для навчання. Тому розглянемо які лабораторні стенди для навчання використовуються в ЗВО України.

1.5 Роботи-маніпулятори що використовуються для досліджень та навчання в ЗВО України

На кафедрі автоматизації, електротехнічних та комп'ютерно-інтегрованих технологій Національного університету водного господарства та природокористування м. Рівне створено стенд, що зображено на рис.1.11.

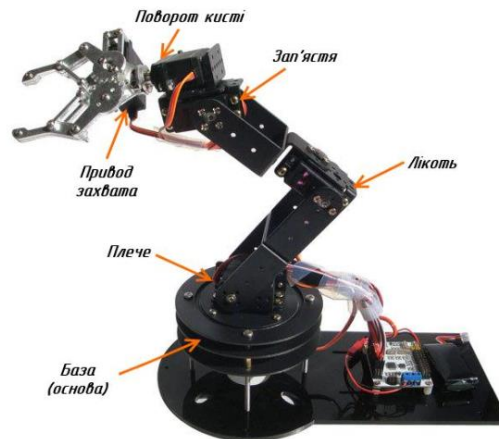


Рисунок 1.11 – Зовнішній вигляд маніпулятора

Маніпулятор має 6 ступенів свободи. Для приведення в рух ланок маніпулятора в кожному із суглобів встановлено сервопривід MG996R. Перший ступінь рухомості забезпечує основа маніпулятора, другий – плече, третій – лікоть, четвертий – обертання кисті, п'ятий – поворот кисті, шостий – захват. Система управління роботом-маніпулятором реалізована на основі мікроконтролерної плати Arduino Mega 2560. Однак можливості цього лабораторного стенду є досить обмеженими, що дозволяє лише обертати окремі кінематичні ланки від початкового положення до кінцевого [10].

На кафедрі факультету комп'ютерно-інтегрованих технологій, мехатроніки і робототехніки Державного університету «Житомирська політехніка» працює лабораторний стенд на базі робота моделі TinkerKit Braccio.

TinkerKit Braccio – це робот ангулярної системи, маніпуляційна система (МС) якого має 5 ступенів рухомості, що реалізуються 6 рухомими ланками від L2 до L6, які приводяться в рух відповідними двигунами від M1 до M6. Загальний вигляд робота наведено на рис.1.12.

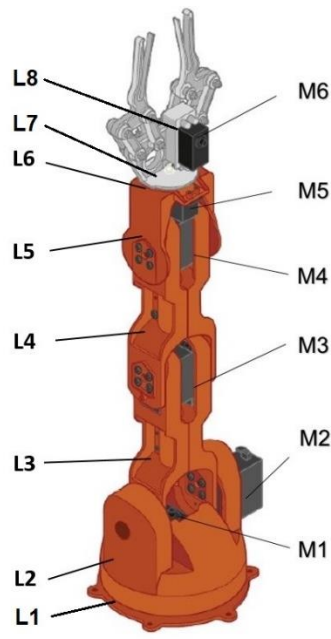


Рисунок 1.12 – Загальний вигляд робота TinkerKit Braccio

На рис.1.12 позначено:

- L1 – Deatl (основа);
- L2 – Base (основа, приводиться в рух двигуном M1 відносно ланки L1);
- L3 – Shoulder (плече, приводиться в рух двигуном M2);
- L4 – Elbow (лікоть, приводиться в рух двигуном M3);
- L5 – Vertical wrist (вертикальний зап'ясток, приводиться в рух двигуном M4);
- L6 – Rotary wrist (обертальний зап'ясток, приводиться в рух двигуном M5);
- L7 – конструктивна константа;
- L2 – Gripper (схват, затискний пристрій, приводиться в рух двигуном M6);

Конфігурацію робота можна змінювати, зменшуючи кількість рухомих ланок його МС. На робот встановлений класичний двопальцевий схват (gripper), який при бажанні можна замінити на легкий інструмент або інший прилад, наприклад, камеру, ліхтар тощо.

Для функціонування робота необхідно встановити спеціальні бібліотеки на ПЗ *Arduino IDE*. Програма стандартно компілюється на комп'ютері, а потім записується у мікроконтролер *Arduino Uno*. Модуль *Braccio Shield* може керувати 6 серводвигунами (M1 – M6) одночасно.

Також є можливість підключити до модуля 6 датчиків (в роз'єми *INPUTS* за рис. 1.13). Дана функція розширює можливості застосування робота.

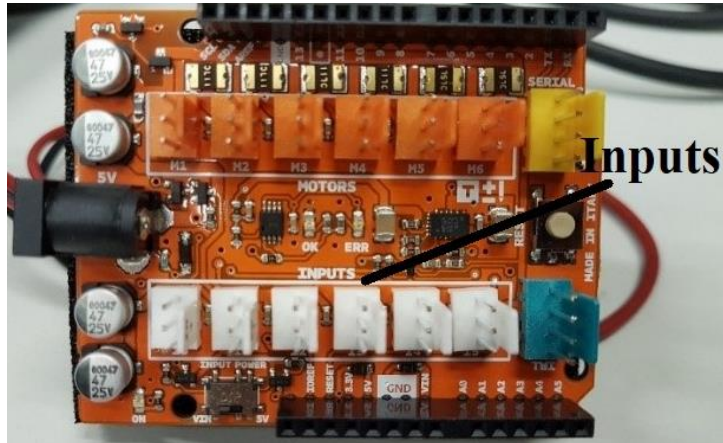


Рисунок 1.13 – Загальний вигляд плати розширення *Braccio Shield*

Лабораторний стенд складається з таких складових груп:

- елементи управління;
- виконавчі механізми;
- допоміжні компоненти.

В групу *елементи управління* входять наступні компоненти:

- плати *Arduino Uno* (2 шт.);
- драйвери *Braccio Shield* (2 шт.);
- панель управління (для ручного керування 1 шт.).

До групи виконавчі механізми входить два робота моделі *TinkerKit Braccio*.

Група допоміжні компоненти складається з ТО (технологічних об'єктів) і станини, на якій встановлені роботи і елементи управління.

На рис. 1.14 зображено лабораторний стенд з розміщенням його складових елементів.

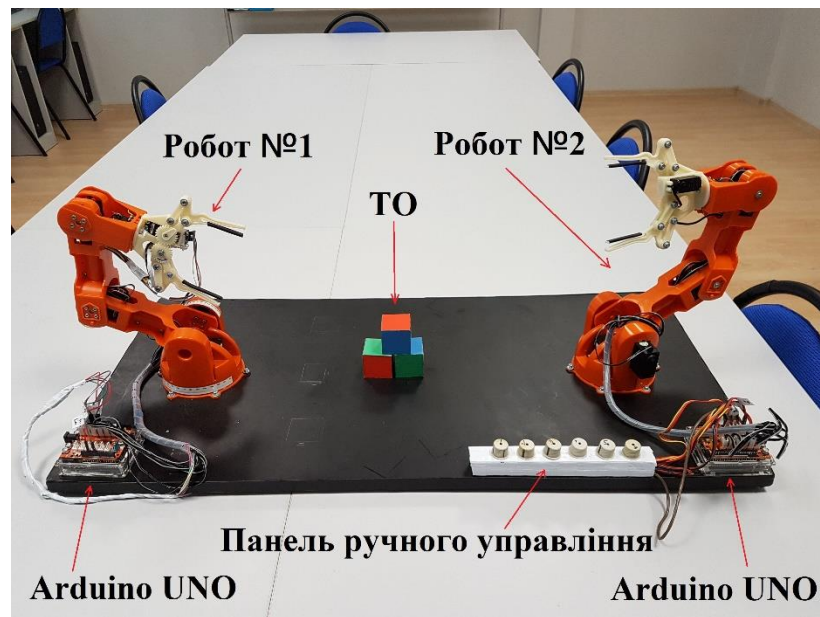


Рисунок 1.14 – Лабораторний стенд

Перевагами цього лабораторного стенду є можливість симуляції великої кількості сценаріїв руху роботів, що може бути дуже корисним при налаштуванні нової промислової лінії автоматизації. Проте цей лабораторний стенд має обмежені можливості у відпрацюванні траєкторій руху та з точки зору програмування [11].

У 2020 році на базі національного ТУ «Дніпровська політехніка» спільно з компанією Fischertechnik було відкрито лабораторію мехатроніки та робототехніки. Лабораторія оснащена діючими макетами технологічного обладнання німецької компанії Fischertechnik, до якого входить 3D-робот із механічним захватом

Виконавчі механізми стенду працюють за керуючими сигналами, які надходять з елементів управління, а саме підсилювачів, за допомогою драйверів, керуючий сигнал від контролера Arduino Uno до серводвигунів. Стенд призначено для виконання простих дій та захвату об'єкту за заданим алгоритмом, що зображено на рис.1.15.

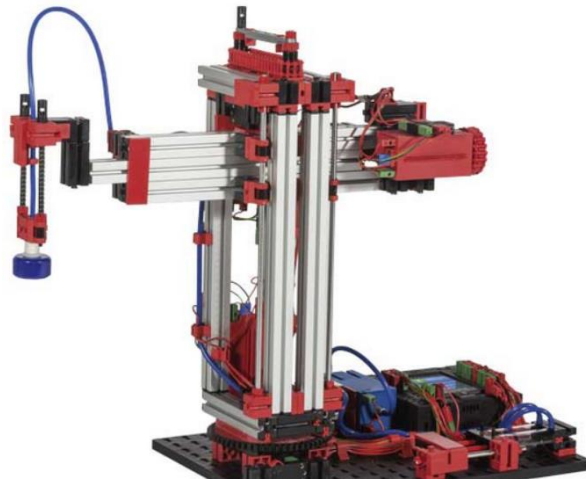


Рисунок 1.15 – Зовнішній вигляд лабораторного станду

Керування стандом здійснюється від програмованого контролера ROBOTICS TXT, що програмується в середовищі ROBO Pro або мовами C, C++, C# и VB. Контролер працює під керуванням ОС Linux. Оснащений кольоровим сенсорним дисплеєм 2,4 дюйма. Головний процесор ARM Cortex A8 (два ядра, 32 біт, 600 МГц), периферійний — Cortex M3. Універсальні входи: 8 штук (цифрові/аналогові 0-9 В або 0-5 кОм). Лічильні входи: 4 штуки. Підключення до ПК: USB/mini-USB. Модель призначена для захоплення і переміщення заготовок між виробничими ділянками. Захоплення наступної заготовки здійснюється після повернення маніпулятора в початкове положення [12].

Даний лабораторний стенд має великий потенціал з боку лабораторних досліджень, однак через свою конфігурацію кінематичних пар та складність програмування, робить його досить складним рішенням для навчання.

1.6 Постановка задач дипломного проекту

В результаті проведеного аналізу були визначені переваги та недоліки існуючих лабораторних стендів, які використовуються в університетах країн Європи та Україні. Найкращі роботи-маніпулятори характеризуються високою вартістю, а більш дешеві рішення обмежені у своєму функціоналі та мають складнощі під час експлуатації, що не дозволяю повноцінно виконувати дослідження кінематики їх руху.

Тому розробка лабораторного стенду для дослідження кінематики руху роботів-маніпуляторів є актуальним завданням. Такий стенд має забезпечити: формування складних траєкторій руху при позиційному керуванні роботом; інтерфейс для завдання цих траєкторій через узагальнені координати, отримані в результаті попереднього вирішення задач кінематики. Досягнення зазначеної цілі передбачає вирішення ряду задач:

- розробка структури стенду;
- визначення вимог до окремих складових структури та вибір їх реалізації;
- розробка апаратних та програмних складових;
- розробка математичної моделі стенду та перевірка її працездатності;
- складання вказівок для роботи зі стендом.

Висновки до розділу 1

В першому розділі дипломного проекту було проведено загальний аналітичний огляд роботів маніпуляторів. Було виокремлено основні етапи розвитку РМ, їх класифікацію та сфери застосування.

Виконавши огляд зарубіжних та вітчизняних лабораторних стендів на основі РМ, що використовуються для освітніх цілей та наукових досліджень, визначили переваги та недоліки кожного з стендів. Як результат, було сформовано завдання до дипломного проекту.

2 СТРУКТУРА ТА ВИБІР СКЛАДОВИХ ЛАБОРАТОРНОГО СТЕНДУ

2.1 Принцип дії та загальна структурна схема лабораторного стенду

Головне завдання розроблюваного лабораторного стенду – це переміщення РО від однієї точки до іншої за заданий період часу, або, іншими словами, лабораторний стенд повинен мати можливість позиційного керування. Важливим моментом є те, що в кожній точці РО повинен робити зупинку.

За основу візьмемо робот-маніпулятор, який має 5 ступенів свободи руху, та математичний опис якого здійснюється в ангулярній системі координат. Структура механічної частини робота наведена на рис.2.1. Саме така конфігурація РМ дозволяє створити велику кількість траєкторій для відпрацювання.

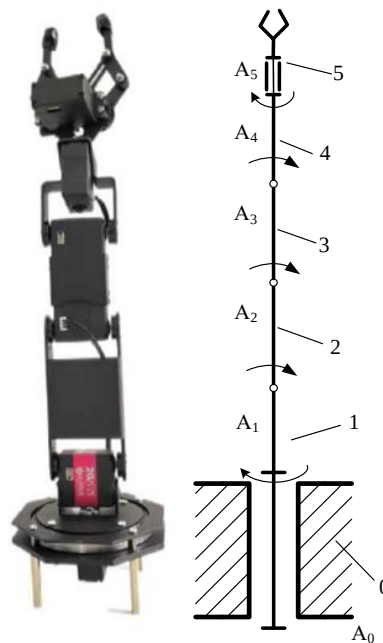


Рисунок 2.1 - Зовнішній вигляд та структурна схема робота-маніпулятора

Використовуючи формулу Сомова-Малишева знайдемо ступінь рухливості маніпулятора:

$$W = 6n - \sum_{i=1}^{i=5} ip_i = 6n - 5p_5 = 6 \cdot 5 - 5 \cdot 5 = 30 - 25 = 5 \quad (2.1)$$

де W - число ступенів рухливості просторового ВМ; n - кількість рухомих ланок; i - клас кінематичної пари; p_i - кількість кінематичних пар i -го класу.

Так як у маніпулятор на рис.2.1 є роботом з послідовною кінематикою, то кожній рухомій ланці відповідає одна кінематична пара п'ятого класу, тобто $n = p_5$. Відповідно, $W = 6n - 5n = n$ або $W = p_5$, тобто число ступенів рухливості в таких маніпуляторах дорівнює числу рухомих ланок і кінематичних пар.

Керування переміщення кінематичних пар РМ має виконуватися за допомогою сервоприводів, що в свою чергу керуються за допомогою пристрою керування сервоприводами робота. Розрахунок проміжних точок виконується блоком обчислення траєкторії руху. Для обміну інформацією між пристроєм керування сервоприводами та блоком обчислення траєкторії може потребувати використання гальванічної розв'язки. Ввід даних, які характеризують траєкторію руху: узагальнені координати основних позицій, часові інтервали руху між ними та час пауз в кожній позиції, доцільно виконувати в програмному застосунку, виконуваному на персональному комп'ютері.

Таким чином, можемо зобразити загальну структурну схему лабораторного стенду.

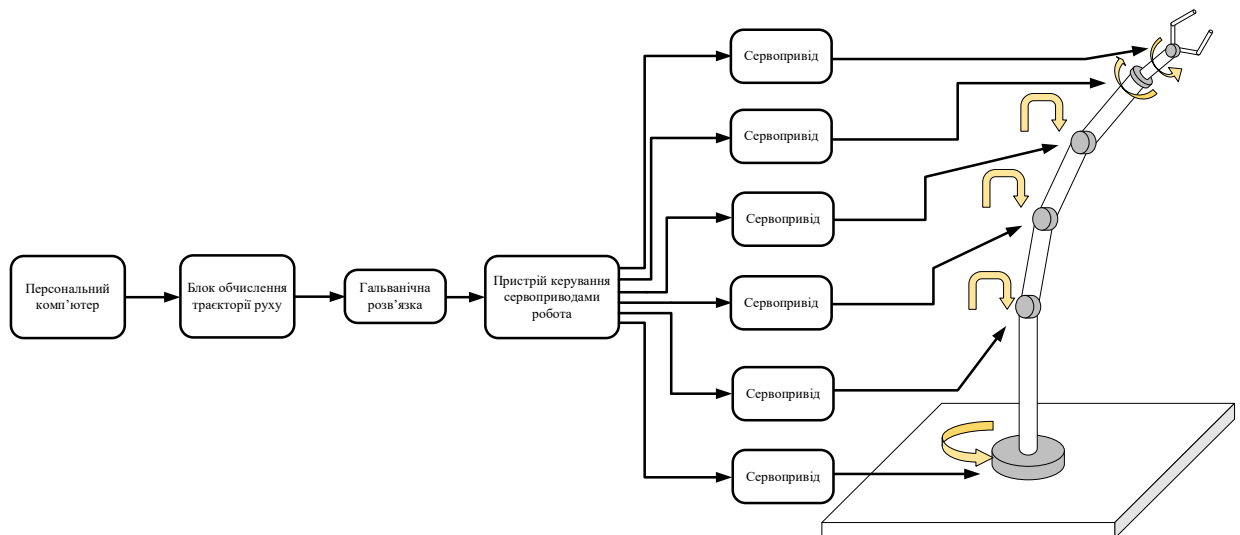


Рисунок 2.2 – Загальна структурна схема лабораторного стенду

Далі необхідно обрати реалізацію кожного з компонентів структури.

2.2 Електромеханічна частина лабораторного стенду

У якості електромеханічної частини лабораторного стенду використаємо сервоприводи. Для вибору сервоприводів необхідно розрахувати максимальне зусилля, яке повинно бути створено. Аналізуючи схему механічної складової, можна зробити висновок, що максимальне статичне зусилля буде прикладатися до сервоприводу першої кінематичної пари, який забезпечує механічний рух між першою та другою ланками маніпулятора, коли друга, третя, четверта та п'ята ланки розміщені горизонтально (рис.2.3).

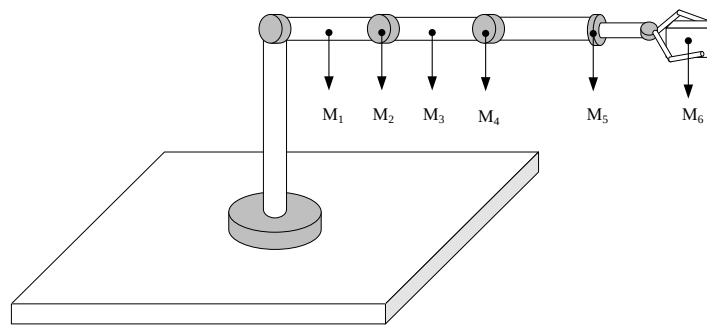


Рисунок 2.3 – Схема РМ для розрахунку максимального зусилля

Враховуючи структурну схему РМ необхідно обрати 6 сервоприводів. Знайдемо максимальне зусилля, що повинен витримувати сервопривід між першою та другою ланками маніпулятора за допомогою наступного виразу:

$$\begin{aligned}
 M &= g(m_1 l_1 + m_2 l_2 + m_3 l_3 + m_4 l_4 + m_5 l_5 + m_6 l_6) = \\
 &= 9.8(0.07 \cdot 0.06 + 0.06 \cdot 0.115 + 0.05 \cdot 0.14 + \\
 &0.06 \cdot 0.165 + 0.16 \cdot 0.22 + 0.03 \cdot 0.35) = 0.722 \text{ Н} \cdot \text{м} = 7.36 \text{ кгс} \cdot \text{см}
 \end{aligned}
 \tag{2.2}$$

де $m_1, m_2, m_3, m_4, m_5, m_6$ – маси відповідних частин РМ (на основі проведеного аналітичного огляду, прийmemo вагу сервоприводів 60 г), $l_1, l_2, l_3, l_4, l_5, l_6$ – довжини відповідних ділянок у РМ.

Як ми бачимо, нам необхідно використати сервоприводи що можуть витримати статичний момент 0.722 Н·м, або 7.36 кгс·см. Однак вибір

серводвигунів потрібно провести із запасом, для забезпечення динаміки руху кінематичних пар та збільшення надійності лабораторного стенду. Також, заради забезпечення уніфікації, будемо вважати, що статичний момент на інших ділянках РМ рівний статичному моменту з виразу (2.2). Таким чином, оберемо сервопривод DS3218MG (з номінальним моментом 19 кгс·см) для 1, 2, 3 та 4-ої кінематичних пар, а для нульової кінематичної пари та захвату можна використати більш прості сервоприводи MG996R (з номінальним моментом 9.4 кгс·см).

2.2.1 Сервопривод MG996R

Високомоментний цифровий сервопривод MG996R оснащений металевою шестернею, що забезпечує надзвичайно високий крутний момент у малому корпусі. MG996R має покращену протиударну конструкцію та перероблену друковану плату та мікросхему що робить його надзвичайно точним. Редуктор і двигун були модернізовані для покращення зони нечутливості та центрування [13].

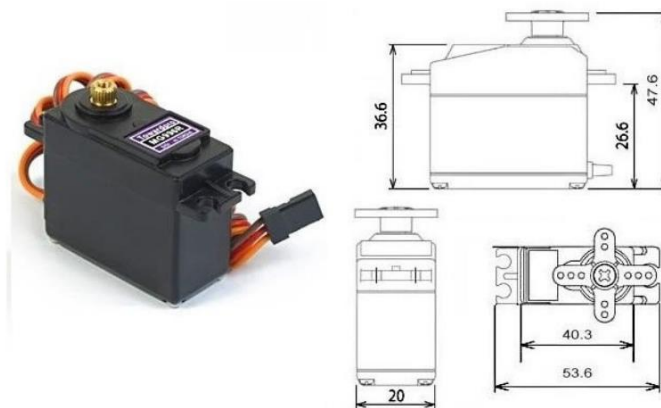


Рисунок 2.4 – Зовнішній вигляд та геометричні розміри сервопривода MG996R

Цей стандартний сервопривід з високим крутним моментом може обертатися приблизно на 180 градусів (по 90 в кожному напрямку).

Керування положенням сервопривода відбувається за допомогою ШІМ сигналу, частотою 50 Гц та часом імпульса 20 мс. Принцип керування сервоприводом наведено на рис.2.5.

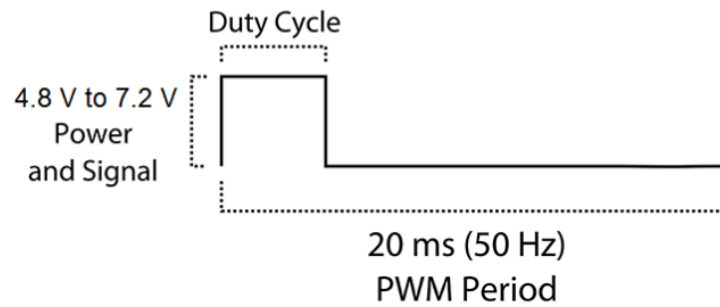


Рисунок 2.5– Принцип дії керування сервоприводом MG996R

Внутрішню структуру сервопривода MG996R можна зобразити наступним чином [14]:

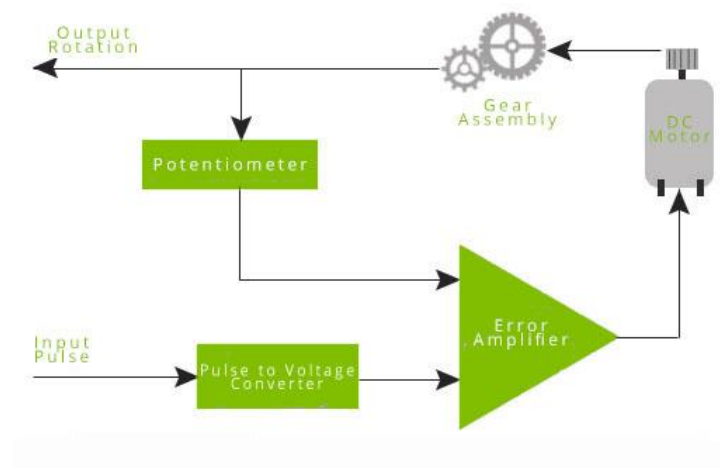


Рисунок 2.6 – Внутрішня структура сервоприводу MG996R

Двигун приймає імпульси від підсилювача та забезпечує обертання. Коли двигун постійного струму зупиняється, це означає, що вал двигуна і ручка потенціометра знаходяться в положенні рівноваги. Вхідні сигнали підсилювача рівні, і двигун не приводиться в рух.

Коли подається імпульс, сигнали на входах підсилювача не є рівними, і таким чином підсилювач приводить в дію двигун. Коли двигун обертається, у рух приводиться шестерня і, таким чином, також обертає потенціометр.

Потенціометр (через дільник напруги) генерує напругу, еквівалентну вхідному імпульсу, яка зупиняє двигун постійного струму, вказуючи на рівновагу.

До складу сервопривода входять:

1. Двигун (4.4-6V DC Brushed Motor)
2. Драйвер двигуна KC9102
3. MOSFET збірки 9926 та 4953
4. Потенціометр
5. Чотири передатних шестерні

Таблиця 2.1 – Технічні характеристики сервопривода MG996R [13]

Вага, г	55
Габарити, мм	40.7x19.7x42.9
Крутний момент, кгс·см	9.4 (4.8 В), 11 (6 В)
Швидкість обертання, с/60°	0.17 (4.8 В), 0.14 (6 В)
Робоча напруга, В	4.8 – 7.2
Струм, мА	500 – 900 (6В)
Струм короткого замикання, А	2.5 (6 В)
Ширина зони нечутливості, мкс	5
Робоча температура, °С	0-55

2.2.2 Сервопривод DS3218MG

Сервопривод DS3218MG представляє собою мотор зі спеціальним редуктором, що здатний повертати вихідний вал у задане положення (кут) і утримувати його там із певним зусиллям. Ця модель має металеві шестерні та водонепроникний корпус, зусилля тяги - 18 кг. Передача зусилля здійснюється за допомогою "гойдалки" - важеля з отворами для закріплення елементів механізації: тяг керма висоти/елеронів, колісних осей, ніг робота, замків тощо. Внутрішня структура сервопривода подібна до тої, що зображена на рис.2.6 [15].

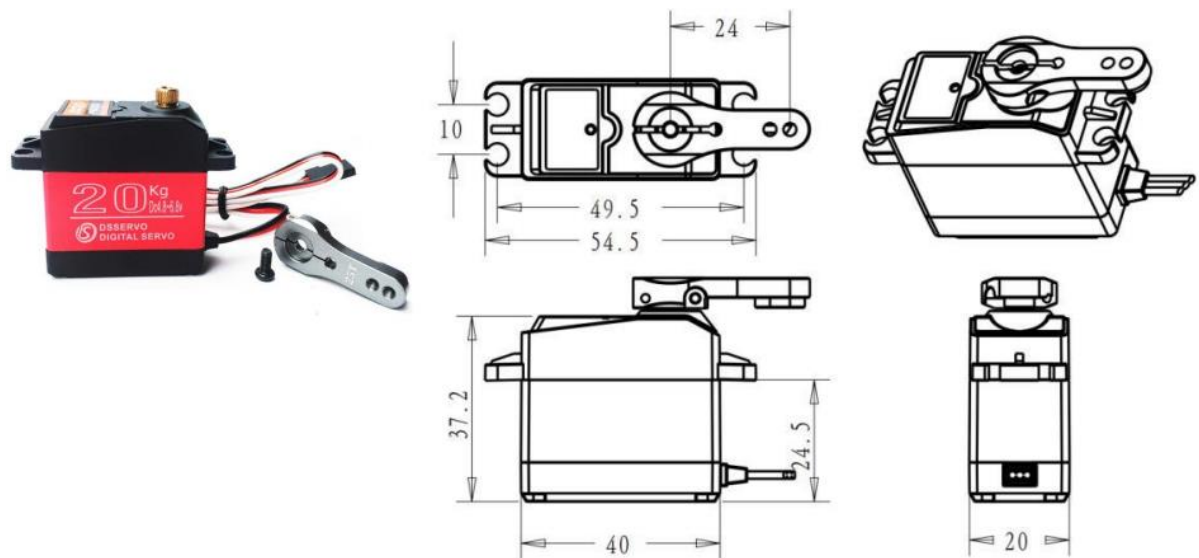


Рисунок 2.7– зовнішній вигляд та габаритні розміри сервопривода DS3218MG

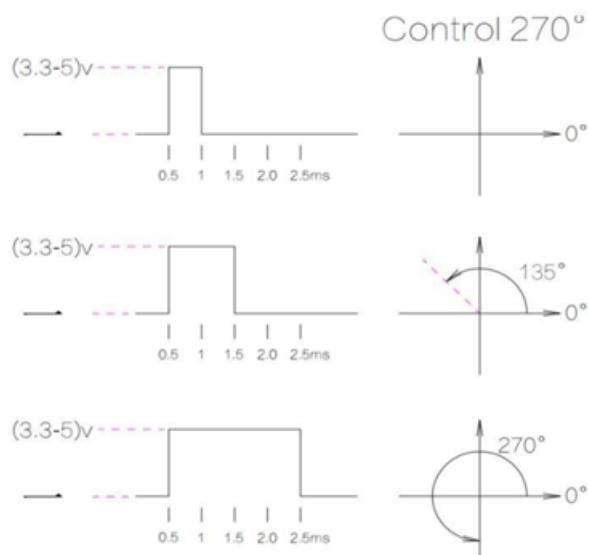


Рисунок 2.8– Принцип дії керування сервоприводом DS3218MG

Таблиця 2.2- Технічні характеристики сервопривода DS3218MG

Загальні характеристики	
Температура зберігання	-20°C~60°C
Робоча температура	-10°C~50°C
Робоча напруга	4.8-6.8 В
Механічні характеристики	
Розмір	40x20x40.5 мм

Вага	60 г	
Передавальне число	250	
Підшипник	Двійний підшипник	
З'єднувальний провід	400±5 мм	
Двигун	трьохполюсний	
Стандарт водозахисту	IP66	
Електричні характеристики		
Робоча напруга	5 В	6.8 В
Струм холостого ходу	4 мА	5 мА
Робоча швидкість	0.16 с/60°	0.14 с/60°
Крутний момент у заблокованому стані	19 кг·см	21.5 кг·см
Струм у заблокованому стані	1.5 А	1.8 А
Особливості керування		
Система управління	ШИМ	
Діапазон ширини імпульсу	500~2500 мкс	
Нульове положення	1500 мкс	
Максимальний кут повороту	270° (при 500~2500 мкс)	
Ширина зони нечутливості	3 мкс	
Робоча частота	50-330 Гц	
Напрямок обертання	За годинниковою стрілкою (при 500~2500 мкс)	

2.3 Пристрій керування сервоприводами

Задача пристрою керування сервоприводами – прийом проміжних розрахунків від одноплатного комп'ютера та керування сервоприводами. У якості МК використаємо STM32F1038T6, що встановлено на плату розширення ZL-KPZ32, що зображено на рис.2.8. Вибір саме цього МК обумовлено можливістю формування 6 ШІМ сигналів одночасно та наявністю декількох протоколів передачі даних, включаючи UART, SPI, I2C та інші.

STM32F103C8T6 – це мікроконтролер з серії STM32 від компанії STMicroelectronics. Цей мікроконтролер має високу продуктивність та низьке споживання енергії, що робить його популярним для широкого спектру застосувань від промислових контролерів до електроніки споживання.

Мікроконтролер STM32F103C8T6 включає в себе:

1. Ядро Arm® 32-bit Cortex®-M3 CPU:
 - максимальна частота 72 МГц,
 - має однотактне множення та апаратне ділення.
2. Пам'ять:
 - 64 або 128 Кбайт флеш-пам'яті,
 - 20 Кбайт SRAM.
3. Керування синхронізацією, скиданням і живленням:
 - живлення входів/виходів від 2,0 до 3,6 В,
 - POR, PDR і програмований детектор напруги (PVD),
 - кварцевий резонатор від 4 до 16 МГц,
 - внутрішній RC 8 МГц з заводською підстроюванням,
 - внутрішній RC 40 кГц,
 - PLL для тактової частоти процесора,
 - 32 кГц генератор для RTC з калібруванням.
4. Низьке енергоспоживання:
 - режими сну, зупинки та очікування,
 - живлення V_{BAT} для RTC і резервних регістрів.
5. Два 12-бітних A/D перетворювача (до 16 каналів):
 - діапазон перетворення: від 0 до 3,6 В,
 - можливість подвійної вибірки і утримання,
 - датчик температури.
6. DMA:
 - 7-канальний контролер DMA,
 - Підтримувані периферійні пристрої: таймери, АЦП, SPI, I²C і USART.

7. До 80 швидких портів вводу/виводу.
8. Режим налагодження:
 - Послідовне налагодження SWD та JTAG інтерфейси.
9. Сім таймерів:
 - Три 16-бітних таймерів, кожен з яких має до 4 IC/OC/PWM або лічильник імпульсів та квадратурний (інкрементний) енкодерний вхід,
 - 16-розрядний ШІМ-таймер для керування двигуном з генерацією часу очікування та аварійною зупинкою,
 - Два сторожових таймера (незалежний і віконний),
 - 24-бітний SysTick таймер.
10. До дев'яти інтерфейсів зв'язку:
 - до двох інтерфейсів I2C (SMBus/PMBus®),
 - до трьох інтерфейсів USART,
 - до двох SPI (18 Мбіт/с),
 - інтерфейс CAN (2.0B Active),
 - повношвидкісний інтерфейс USB 2.0 [16].

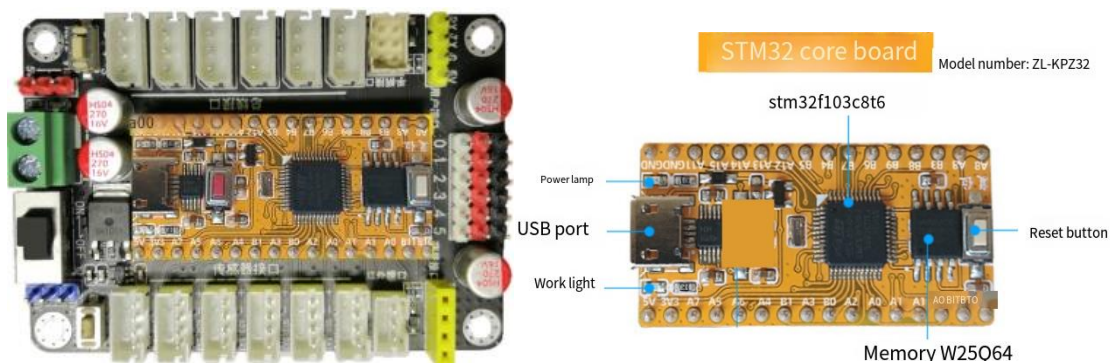


Рисунок 2.8 – Зовнішній вигляд плати ZL-KPZ32

Плата управління ZL-KPZ32 розроблена компанією Hangzhou Zhongling Technology Co, LTD, та містить наступні прилади:

1. Два світлодіоди,
2. USB порт,
3. Запам'ятовуючи пристрій W25064,
4. 2 клавіші користувача, перемикач живлення, кнопка збросу,

5. 6 виводів для ШІМ сигналів,
6. виводи для UART та Bluetooth інтерфейсів,
7. входи для підключення додаткових пристроїв [17].

2.4 Блок розрахунку траєкторій

Задача цього пристрою – прийом вхідних даних, розрахунок проміжних точок та відправка результатів розрахунку до МК. У лабораторному стенді використаємо одноплатний комп'ютер Raspberry Pi Zero. Вибір цього пристрою продиктовано значною обчислювальною здатністю, наявністю декількох послідовних інтерфейсів для передачі даних, та можливістю прийому/передачі даних через Wi-Fi, що є зручним для прийому вхідних даних від ПК.

Raspberry Pi Zero – це одноплатний комп'ютер, що виконано на базі чіпа Broadcom BCM2835, кристал якого містить у собі процесор CPU ARM1176JZ-Z-F розігнаний до частоти 1 ГГц і графічний двоядерний співпроцесор GPU VideoCore IV з частотою 400 МГц. Зверху над ним розташований чіп пам'яті Elpida B4432BBPA-10-F 512MB.

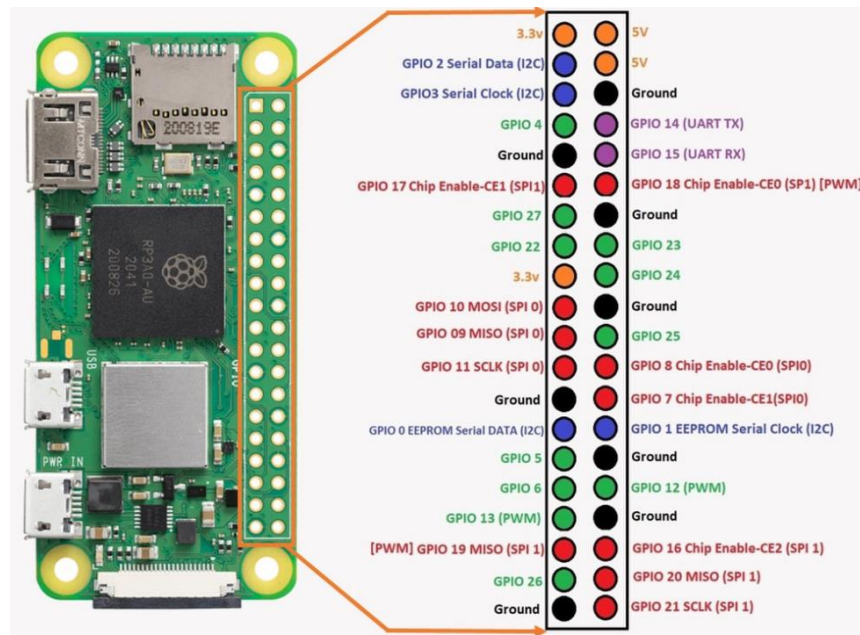


Рисунок 2.9 – Зовнішній вигляд та призначення виводів одноплатного комп'ютера Raspberry Pi Zero

Основні характеристики Raspberry Pi Zero:

- Чип: Broadcom BCM2835 з CPU і GPU;
- Процесор CPU: ARM1176JZ-F з тактовою частотою 1 ГГц;
- Графічний співпроцесор GPU: VideoCore IV з тактовою частотою 400 МГц;

- RAM-пам'ять: 512 МБ;
- Порти Mini HDMI і Micro USB с OTG;
- Живлення від Micro USB ;
- Стандартний 40-піновий роз'єм для плат розширення;
- Роз'єми копитного відео та скидання;
- CSI (Camera Serial Interface) роз'єм камери;
- Карта пам'яті: MicroSD;
- Габарити: 66×32×5 мм.

Характеристики бездротової комунікації (CYW43438):

- Частотний діапазон: 2,4 ГГц;
- Wi-Fi: 802.11 b/g/n wireless LAN;
- Bluetooth 4.1;
- Bluetooth Low Energy (BLE) [18].

2.5 Гальванічна розв'язка

Оскільки МК та одноплатний комп'ютер можуть мати різні джерела живлення, необхідно передбачити гальванічну розв'язку, що дозволить безпечно передавати значення узагальнених координат для проміжних точок на траєкторії. Для цього використаємо модуль опторозв'язки TLP281.

Модуль опторозв'язки TLP281 цифрових сигналів 4-х канальний призначений для гальванічної розв'язки ланцюгів з різним потенціалом загального проводу, різною напругою логічних рівнів або для розв'язки від "шумних" ланцюгів, таких як електродвигуни, соленоїди і т.д. . Схема цього модулю наведена на рис 2.10 [19].

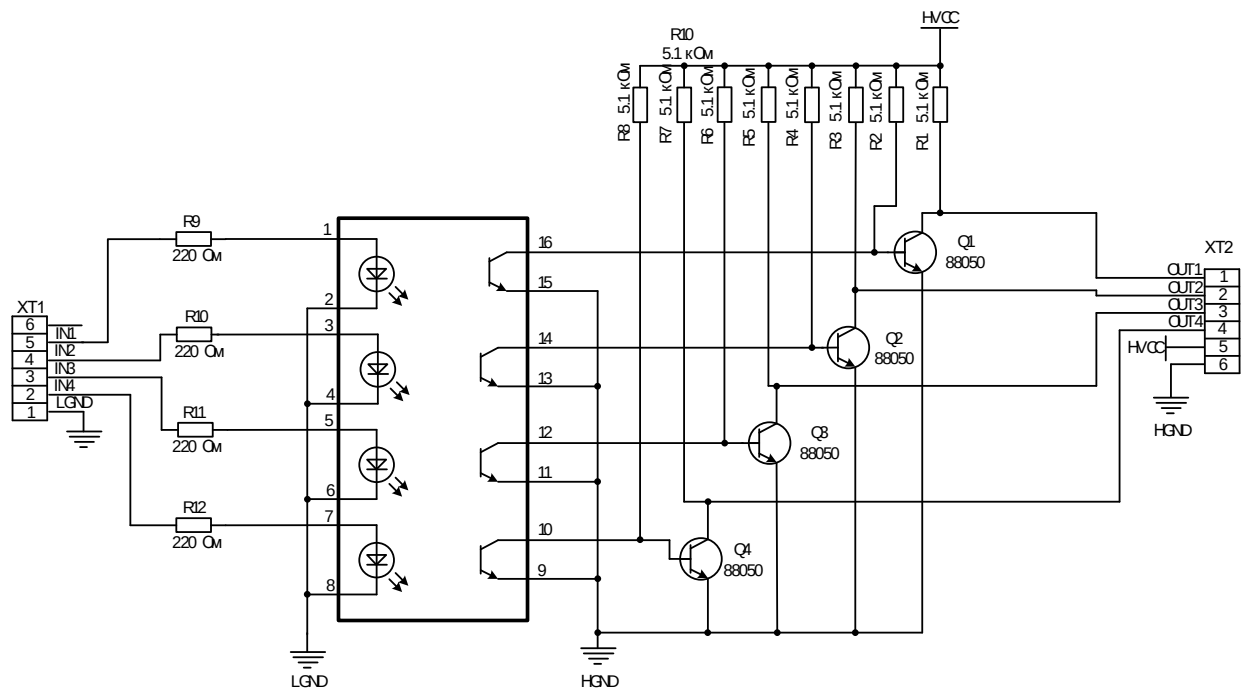


Рисунок 2.10 – Схема модулю TLP281

Характеристики:

- Кількість каналів: 4;
- Максимальна напруга розв'язки: 2500 В;
- Максимальний вхідний струм: 50 мА;
- Максимальна вихідна напруга: 24 В;
- Максимальний вихідний струм кожного каналу: 500мА;
- Розміри плати: 25 x 24 мм [19].

Висновки до розділу №2

У другому розділі дипломного проекту було виконано аналіз принципу роботи лабораторного стенду, за результатами якого була складена структурна схема стенду робота маніпулятора, до складу якої входить: персональний комп'ютер, блок обчислення траєкторії руху, гальванічна розв'язка, пристрій керування сервоприводами робота та сервоприводи.

За допомогою структурної схеми розглянули завдання кожної складової лабораторного стенду та виконали підбір технічного обладнання. У якості блоку обчислення траєкторії руху було обрано одноплатний комп'ютер RaspberryPi Zero, пристроєм керування сервоприводами виступає

мікроконтролер STM32, що встановлено на плату розширення ZL-KPZ32, для реалізації гальванічної розв'язки було використано модуль TLP281, для переміщення кінематичних пар було обрано сервоприводи типу MG996R та DS3218MG.

3 АЛГОРИТМ РОБОТИ З ЛАБОРАТОРНИМ СТЕНДОМ ТА ОПИС ПРОГРАМНИХ СКЛАДОВИХ

Взаємодія між персональним комп'ютером, RaspberryPi Zero та мікроконтролером STM32 відбувається наступним чином:

- За допомогою UART протоколу, через оптопару, мікроконтролер STM32 отримує команди на зміну положення сервоприводів, що відбувається за допомогою зміни коефіцієнту заповненості ШІМ сигналів на виході мікроконтролера.

- Розрахунок траєкторії руху та відправку результатів обчислення через UART протокол відбувається з допомогою одноплатного комп'ютера RaspberryPi.

- Початкові дані (кількість точок, узагальнені координати, час переходу та паузи між точками) вводить користувач до застосунку на ПК.

- Після виконання перевірки вхідних даних застосунок виконує передачу даних за допомогою SSH запиту до пристрою обробки траєкторії руху, тобто до одноплатного комп'ютера RaspberryPi.

3.1 Програмне забезпечення для МК STM32

Програмування МК STM32F103C8T6 виконаємо на мові програмування C у програмному середовищі Keil μ Vision. Для спрощення процесу програмування використаємо стандартну бібліотеку SPL.

Бібліотека SPL (Standard Peripherals Library) – бібліотека, створена компанією STMicroelectronics мовою C для своїх мікроконтролерів сімейства STM32F10x. Містить функції, структури та макроси для полегшення роботи з периферією мікроконтролера [20].

У даному лабораторному стенді МК виконує наступні функції:

1. Прийом даних від Raspberry Pi через UART інтерфейс.
2. Обробка прийнятих даних.
3. На основі оброблених даних формування ШІМ сигналів для керування сервоприводами РМ.

Першим кроком потрібно виконати налаштування таймерів, що будуть забезпечувати формування ШІМ сигналів на виходах мікроконтролера для керування сервоприводами.

Принцип дії ШІМ на мікроконтролері можна пояснити за допомогою рис.3.1.

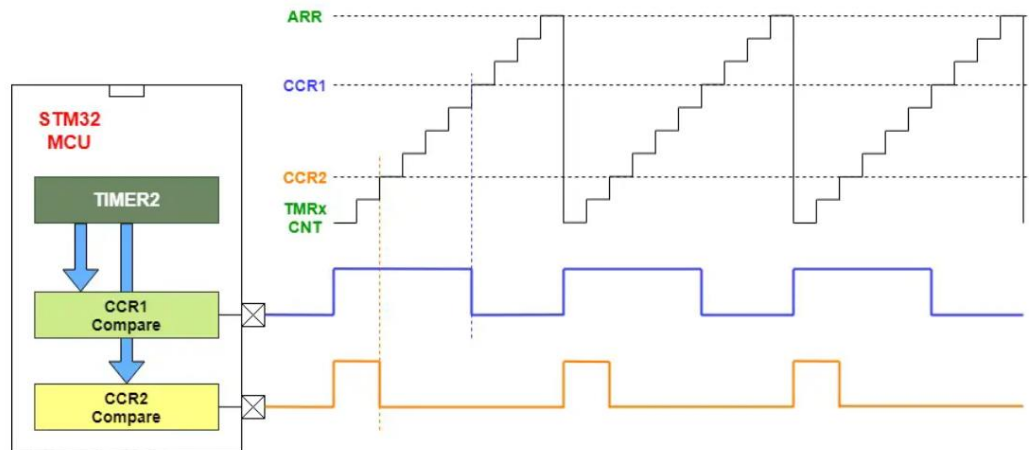


Рисунок 3.1 – Принцип дії ШІМ

Необхідні значення параметрів ШІМ сигналу можна розрахувати за наступною формулою:

$$F_{PWM} = \frac{F_{CLK}}{(ARR - 1)(PSC + 1)} \quad (3.1)$$

де F_{PWM} – частота ШІМ сигналу; F_{CLK} – частота роботи процесора (72 МГц); ARR – значення регістру, до якого рахує процесор; PSC – значення дільника частоти процесора.

Згідно з технічною документацією до сервоприводів MG996R та DS3218MG там необхідно забезпечити ШІМ сигнал з періодом 20 мс, або частотою 50 Гц. Таким чином, з формули (3.1), значення дільника частоти будуть дорівнювати $PSC = 71$, а значення регістру $ARR = 19999$ [21].

За початкове положення робота маніпулятора приймемо як показано на рис.2.1, тобто початкове положення становить 90° . З цього випливає, що рух кожного з сервоприводів може відбуватися в діапазоні $\pm 90^\circ$. Згідно з технічною документацією, для забезпечення руху сервоприводів від 0° до

180° необхідно подати ШІМ сигнал з коефіцієнтом заповнення 5% та 12.5% відповідно.

Зміна коефіцієнту заповнення відбувається за допомогою регістру CCR. Враховуючи, що для ШІМ сигналу з частотою 50 Гц значення дільника частоти дорівнює $PSC = 71$, а значення регістру $ARR = 19999$, то можемо визначити, що для початкового стану (90°) значення $CCR = 1500$, для 180° - $CCR = 2165$, а для 0° = 835.

Початкове налаштування таймерів виконаємо за наступним прикладом [21-27]:

```
void TIM3_Init(void){
    TIM_TimeBaseInitTypeDef TIM_TimeBaseInitStruct; // Оголошення структури для ініціалізації
тактування таймера
    TIM_OCInitTypeDef TIM_OCInitStruct; // Оголошення структури для ініціалізації таймера
    GPIO_InitTypeDef GPIO_InitStruct; // Оголошення структури для ініціалізації портів вводу/виводу

    GPIO_PinRemapConfig(GPIO_PartialRemap_TIM3, ENABLE); //Ввімкнення альтернативної функції
виходу

    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM3, ENABLE); //Ввімкнення тактування для
таймеру 3
    TIM_TimeBaseInitStruct.TIM_Prescaler = 71; //Встановлення дільника для таймеру 3
    TIM_TimeBaseInitStruct.TIM_Period = 19999; //Встановлення періоду ШІМ для таймеру 3
    TIM_TimeBaseInitStruct.TIM_ClockDivision = TIM_CKD_DIV1; //Значення дільника тактового
сигналу
    TIM_TimeBaseInitStruct.TIM_CounterMode = TIM_CounterMode_Up; //Вибір режиму роботи таймеру
(рахунок вгору)
    TIM_TimeBaseInit(TIM3, &TIM_TimeBaseInitStruct); //Ініціалізація параметрів таймера 3

    TIM_Cmd(TIM3, ENABLE); //Запуск таймера 3

    TIM_OCInitStruct.TIM_OCMode = TIM_OCMode_PWM1; //Вибір режиму роботи ШІМ
    TIM_OCInitStruct.TIM_OutputState = TIM_OutputState_Enable; //Вибір стану порівняння виходів
таймера
    TIM_OCInitStruct.TIM_OCPolarity = TIM_OCPolarity_High; //Вибір полярності додаткових виходів
    TIM_OCInitStruct.TIM_Pulse = 1500; //Встановлення тривалості імпульсів
    TIM_OC1Init(TIM3, &TIM_OCInitStruct); //Ініціалізація параметрів ШІМ
    TIM_OC1PreloadConfig(TIM3, TIM_OCPreload_Enable); //Ввімкнення регістру CCR

    GPIO_InitStruct.GPIO_Pin = GPIO_Pin_4; //Вибір виходу
    GPIO_InitStruct.GPIO_Mode = GPIO_Mode_AF_PP; //Вибір режиму роботи виходу
    GPIO_InitStruct.GPIO_Speed = GPIO_Speed_2MHz; //Вибір швидкості роботи виходу
    GPIO_Init(GPIOB, &GPIO_InitStruct); //Ініціалізація параметрів виходу
}
```

Оскільки період ШІМ, яким керуються сервоприводи, складає 20 мс, відправку даних будемо проводити з таким самим значенням періоду за допомогою протоколу UART.

UART (Universal Asynchronous Receiver-Transmitter) – це протокол універсального асинхронного приймача/передавача, що визначає набір правил для обміну послідовними даними між двома пристроями. В ньому дані передаються лише по двох дротах (є ще провід землі). Асинхронний означає відсутність тактового сигналу для синхронізації вихідних бітів від передавального пристрою до приймального кінця [28].

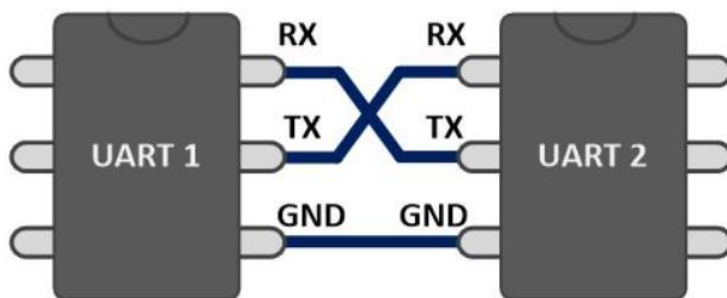


Рисунок 3.2 – Принцип взаємодії пристроїв по протоколу UART

UART протокол широко поширений в електронці, його у різних видах використовують і великі та малі комп'ютери, контролери, датчики, засоби комунікації та інші електронні пристрої.

В цьому протоколі дані передаються послідовно (у вигляді бітів логічних нулів та одиниць) та одиниця відрізняється від нуля виключно за часом між перепадами рівнів сигналу, який заздалегідь задається швидкістю передачі.

Комунікація в UART може бути симплексною (дані надсилаються лише в одному напрямку), напівдуплексною (кожна сторона передає тільки в один проміжок часу) або повнодуплексною (обидві сторони можуть передавати одночасно).

Зв'язок UART здійснюється в так званих кадрах. Кожен кадр складається з бітів – фрагментів інформації, які послідовно надсилаються по лінії. Ці біти підряд включають початковий біт, біти даних, додатковий біт парності та один або два стоп-біти. Кожен біт може мати або високу (вказує на логічну 1), або низьку напругу (вказує на логічний 0). Кожен біт кожного байта передається на рівний відведений проміжок часу (фактично, тайм-слот) [28].



Рисунок 3.3 – Принцип дії протоколу UART

Початкові налаштування UART виконаємо наступним чином [22]:

```
void USART3_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure; //Оголошення структури для налаштування порта
    USART_InitTypeDef USART_InitStructure; //Оголошення структури для налаштування USART
    NVIC_InitTypeDef NVIC_InitStructure; //Оголошення структури для налаштування преривань
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOB, ENABLE); //Ввімкнення тактування для
    порта B
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_USART3, ENABLE); //Ввімкнення тактування для
    USART3

    // Налаштування виводу PB10 -> TX USART.
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10; //Вибір ножки
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP; //Вибір режиму роботи порта
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz; //Вибір швидкості роботи
    GPIO_Init(GPIOB, &GPIO_InitStructure); //Ініціалізація налаштування ножки
    // Налаштування виводу PB11 -> RX USART.
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_11; //Вибір ножки
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING; //Вибір режиму роботи порта
    GPIO_Init(GPIOB, &GPIO_InitStructure); //Ініціалізація налаштування ножки

    USART_InitStructure.USART_BaudRate = baud; //Вибір швидкості обміну
    USART_InitStructure.USART_WordLength = USART_WordLength_8b; //Вибір довжини посилки (8
    біт)
    USART_InitStructure.USART_StopBits = USART_StopBits_1; //Ввімкнення 1 стоп-біту
    USART_InitStructure.USART_Parity = USART_Parity_No; //Вимкнення перевірки парності
    USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
    //Вимкнення апаратного контролю
    USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx; //Ввімкнення передачі
    та прийому для USART3
    USART_Init(USART3, &USART_InitStructure); //Ініціалізація параметрів для USART
    USART_ITConfig(USART3, USART_IT_RXNE, ENABLE); //Ввімкнення преривання USART
    NVIC_InitStructure.NVIC_IRQChannel = USART3_IRQn; //Ввімкнення преривання USART
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0; //Вибір пріоритету в групі
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0; //Вибір пріоритету в підгрупі
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE; //Ввімкнення дозволу для преривання
    NVIC_Init(&NVIC_InitStructure); //Ініціалізація преривання

    USART_Cmd(USART3, ENABLE); // Ввімкнення периферійних пристроїв
}
```

Прийом, обробку та запис нових значень тривалості імпульса ШІМ сигналу виконаємо за допомогою функції обробки переривань від UART[22]:

```
void USART3_IRQHandler()
```

```

{
if (USART_GetITStatus(USART3, USART_IT_RXNE))//Перевірка з якої лінії приходить преривання
{
test_baits[number_baits]=USART_ReceiveData(USART3); //Запис переданих даних у масив
number_baits++; //Збільшення лічильника отриманих даних
if (number_baits==12)
{
pulse1 = (test_baits[0]<<5)+(test_baits[1]>>1); //Перетворення отриманого двійкового числа у
десятичне для сервопривода 1
pulse2 = (test_baits[2]<<5)+(test_baits[3]>>1); //Перетворення отриманого двійкового числа у
десятичне для сервопривода 2
pulse3 = (test_baits[4]<<5)+(test_baits[5]>>1); //Перетворення отриманого двійкового числа у
десятичне для сервопривода 3
pulse4 = (test_baits[6]<<5)+(test_baits[7]>>1); //Перетворення отриманого двійкового числа у
десятичне для сервопривода 4
pulse5 = (test_baits[8]<<5)+(test_baits[9]>>1); //Перетворення отриманого двійкового числа у
десятичне для сервопривода 5
pulse6 = (test_baits[10]<<5)+(test_baits[11]>>1); //Перетворення отриманого двійкового числа у
десятичне для сервопривода 6

TIM2->CCR2 = pulse1; //Запис нового значення імпульса для сервопривода 1
TIM4->CCR3 = pulse2; //Запис нового значення імпульса для сервопривода 2
TIM4->CCR4 = pulse3; //Запис нового значення імпульса для сервопривода 3
TIM4->CCR1 = pulse4; //Запис нового значення імпульса для сервопривода 4
TIM4->CCR2 = pulse5; //Запис нового значення імпульса для сервопривода 5
TIM3->CCR1 = pulse6; //Запис нового значення імпульса для сервопривода 6
number_baits=0; // Скидання лічильника отриманих даних
}
}
}

```

3.2 Застосунок для обробки вхідних даних

Робота з лабораторним стендом починається з введення початкових даних. Для запису, перевірки та відправки вхідних даних (а саме кількість точок, узагальнені координати, час переходу та паузи між точками) було створено застосунок на мові програмування C# за допомогою середовища розробки Visual Studio та вбудованої технології Windows Forms.

Windows Forms – це платформа для створення класичних додатків Windows. Вона забезпечує один із найефективніших способів створення класичних додатків за допомогою візуального конструктора у Visual Studio. Такі функції, як розміщення візуальних елементів керування шляхом перетягування, спрощують створення класичних додатків.

У Windows Forms можна розробляти графічно складні додатки, які просто розгортати, оновлювати, і з якими зручно працювати як в автономному режимі, так і в мережі. Додатки Windows Forms можуть отримувати доступ до локального обладнання та файлової системи комп'ютера, на якому працює додаток [29].

Створений додаток користувача має наступний зовнішній вигляд:

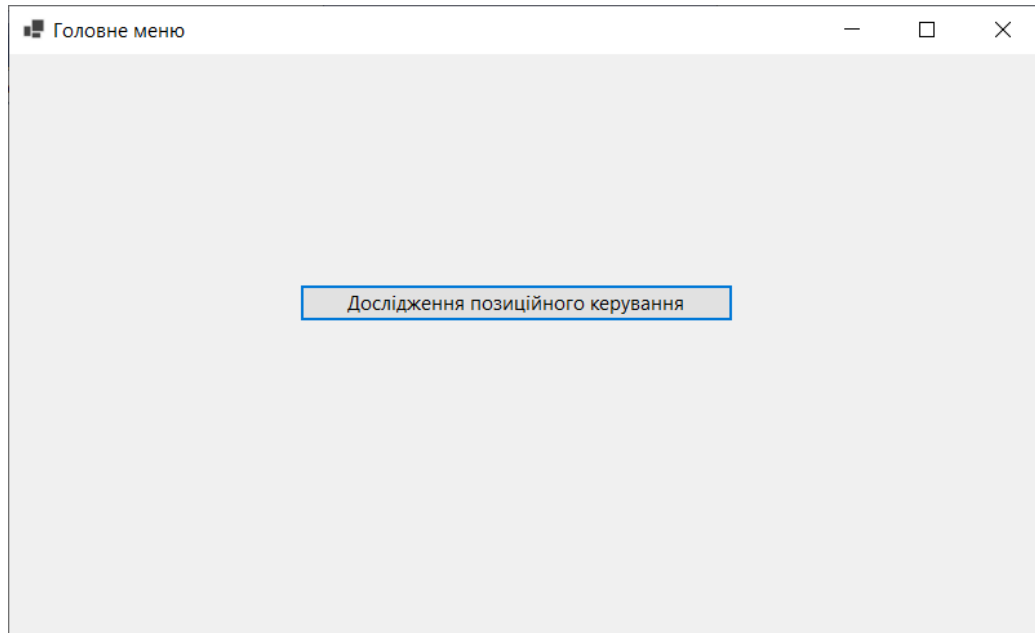


Рисунок 3.4 – Зовнішній вигляд додатку користувача

На першому етапі користувач має обрати режим роботи лабораторного стенду. Наразі доступний лише режим «Дослідження позиційного керування».

Після вибору необхідного режиму користувач потрапляє у наступне вікно, що зображено на рис.3.5.

Оберіть необхідну кількість точок

Оберіть кількість точок:

Введіть координати точок:

	t.1	t.2	t.3	t.4	t.5	t.6	t.7	t.8	t.9	t.10
q1, °	0	0	0	0	0	0	0	0	0	0
q2, °	0	0	0	0	0	0	0	0	0	0
q3, °	0	0	0	0	0	0	0	0	0	0
q4, °	0	0	0	0	0	0	0	0	0	0
q5, °	0	0	0	0	0	0	0	0	0	0

Час руху між точками:

	t(1→2)	t(2→3)	t(3→4)	t(4→5)	t(5→6)	t(6→7)	t(7→8)	t(8→9)	t(9→10)	t(k→n)
t, c	1	1	1	1	1	1	1	1	1	1

Час паузи між точками:

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
t, c	0	0	0	0	0	0	0	0	0	0

Перевірити та зберегти дані | Відправити дані | Почати виконання завдання

Видалити дані | Повернутися

Рисунок 3.5 – Вікно користувача у режимі «Дослідження позиційного керування»

На рис. 3.5 позначено:

1. Інструкція для користувача – у цьому текстовому полі будуть з’являтися підказки для роботи з застосунком.
2. Випадаючий список за допомогою якого користувач обирає кількість точок, що має відпрацювати робот (можна обрати від одної до десяти точок).
3. Блок загальних координат – поле, до якого користувач вносить значення узагальнених координат для раніше обраної кількості точок.
4. Блок для введення часу, за який робот має перемістити РО від попередньої точки до наступної. За замовчуванням у кожному полі встановлено значення 1с.
5. Блок для введення часу паузи після переходу від попередньої точки до наступної.
6. Блок керування, до якого входять 5 клавiш:
 - «Повернутися» – програма повертається на початкове вікно, що зображено на рис..
 - «Перевірити та зберегти дані» – застосунок виконує перевірку введених даних та, якщо перевірку пройдено, зберігає їх у вигляді файлу Log формату txt.
 - «Видалити дані» – застосунок видаляє введені дані у файлі Log.txt та у блоках 3, 4, 5.
 - «Відправити дані» – застосунок виконує відправку даних на Raspberry Pi через мережевий протокол SSH.
 - «Почати виконання завдання» – застосунок формує команду на запуск програми, через мережевий протокол SSH, що знаходиться на Raspberry Pi та розпочинає відпрацювання руху робота між точками. Після початку виконання програми відкривається вікно, що зображено на рис. Користувач не може його закрити поки робот маніпулятор не завершив відпрацювання руху між заданими точками.

Після вибору режиму роботи лабораторного стенду, користувач має обрати кількість точок для відпрацювання.

Дослідження позиційного керування

Оберіть необхідну кількість точок

Оберіть кількість точок: Одна точка
Дві точки
Три точки
Чотири точки
П'ять точок
Шість точок
Сім точок
Вісім точок
Дев'ять точок
Десять точок

Введіть координати точок

	т.1	т.2	т.3	т.4	т.5	т.6	т.7	т.8	т.9	т.10
q1, °	0	0								
q2, °	0	0								
q3, °	0	0								
q4, °	0	0								
q5, °	0	0								

Час руху між точками:

	t(1→1)	t(1→2)	t(2→3)	t(3→4)	t(4→5)	t(5→6)	t(6→7)	t(7→8)	t(8→9)	t(9→10)	t(k→n)
t, c	1	1	1	1	1	1	1	1	1	1	1

Час паузи між точками:

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
t, c	0	0	0	0	0	0	0	0	0	0

Перевірити та зберегти дані Відправити дані Почати виконання завдання

Видалити дані Повернутися

Рисунок 3.6 – Вибір кількості точок

Після вибору необхідної кількості точок активується необхідна кількість полів, які користувач може заповнити (усі введені числа мають бути цілими).

Дослідження позиційного керування

Введіть узагальнені координати та виконайте перевірку даних

Оберіть кількість точок: П'ять точок

Введіть координати точок

	т.1	т.2	т.3	т.4	т.5	т.6	т.7	т.8	т.9	т.10
q1, °	105	45	-10	-45	-50	0	0	0	0	0
q2, °	14	0	88	0	-14	0	0	0	0	0
q3, °	3	57	76	80	-110	0	0	0	0	0
q4, °	52	-180	-500	500	13	0	0	0	0	0
q5, °	211	1	42	50	1	0	0	0	0	0

Час руху між точками:

	t(1→1)	t(1→2)	t(2→3)	t(3→4)	t(4→5)	t(5→6)	t(6→7)	t(7→8)	t(8→9)	t(9→10)	t(k→n)
t, c	1	1	1	1	1	1	1	1	1	1	1

Час паузи між точками:

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
t, c	5	2	1	6	1	0	0	0	0	0

Перевірити та зберегти дані Відправити дані Почати виконання завдання

Видалити дані Повернутися

Рисунок 3.7 – Введення даних до форми

Після введення даних користувач має натиснути на кнопку «Перевірити та зберегти дані». Після натискання цієї клавіші виконується перевірка введених значень за двома критеріями.

Критерій №1: введені значення кутів q_1 - q_5 мають належати проміжку $-90 \leq q \leq 90$. Це обмеження викликано конфігурацією робота-маніпулятора. Якщо введене значення не задовольняє критерію №1, воно буде замінене на -90 або 90 .

Критерій №2: для кожної i -ої та $i+1$ -ої точки повинно виконуватися наступна нерівність:

$$\frac{|q_i - q_{i+1}|}{t_{\text{пер}}} \leq v_{\text{max}} \quad (3.2)$$

де q_i , q_{i+1} – узагальнені координати; $t_{\text{пер}}$ – час переходу між i -ою та $i+1$ -ою точкою; v_{max} – максимальна швидкість руху між точками (прийmemo, що $v_{\text{max}} = 60^\circ/\text{с}$). Якщо критерій №2 на будь-якому проміжку не виконується, то поле загорається червоним кольором та користувач має збільшити час переходу.

Дані введено невірно, збільшіть час переходу

Оберіть кількість точок: П'ять точок

Введіть координати точок:

	t.1	t.2	t.3	t.4	t.5	t.6	t.7	t.8	t.9	t.10
q1, °	90	45	-10	-45	-50	0	0	0	0	0
q2, °	14	0	88	0	-14	0	0	0	0	0
q3, °	3	57	76	80	-90	0	0	0	0	0
q4, °	52	-90	-90	90	13	0	0	0	0	0
q5, °	90	1	42	50	1	0	0	0	0	0

Час руху між точками:

	t(n-1)	t(1-2)	t(2-3)	t(3-4)	t(4-5)	t(5-6)	t(6-7)	t(7-8)	t(8-9)	t(9-10)	t(k-n)
t, с	1	1	1	1	1	1	1	1	1	1	1

Час паузи між точками:

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
t, с	5	2	1	6	1	0	0	0	0	0

Перевірити та зберегти дані Відправити дані Почати виконання завдання

Видалити дані Повернутися

Рисунок 3.8 – Зовнішній вигляд вікна, коли введені дані не пройшли перевірку

Якщо критерій №2 виконується, користувач може відправити дані на Raspberry Pi.

Дослідження позиційного керування

Перевірку виконано, відправляйте данні

Оберіть кількість точок: П'ять точок

Введіть координати точок

	t.1	t.2	t.3	t.4	t.5	t.6	t.7	t.8	t.9	t.10
q1, °	90	45	-10	-45	-50	0	0	0	0	0
q2, °	14	0	88	0	-14	0	0	0	0	0
q3, °	3	57	76	80	-90	0	0	0	0	0
q4, °	52	-90	-90	90	13	0	0	0	0	0
q5, °	90	1	42	50	1	0	0	0	0	0

Час руху між точками:

	t(n-1)	t(1-2)	t(2-3)	t(3-4)	t(4-5)	t(5-6)	t(6-7)	t(7-8)	t(8-9)	t(9-10)	t(k-n)
t, c						1	1	1	1	1	

Час паузи між точками:

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
t, c	5	2	1	6	1	0	0	0	0	0

Рисунок 3.9 – Зовнішній вигляд вікна, коли введені дані пройшли перевірку

Після натиснення клавіші «Відправити дані», за допомогою наступного коду, формується SSH запит та відправка файлу Log.txt [30].

```
private void button4_Click(object sender, EventArgs e)
{
    using (ScpClient client = new ScpClient(ip_address, username, password))
    {
        client.Connect();
        using (Stream localFile = File.OpenRead(localFilePath))
        {
            client.Upload(localFile, remoteFilePath);
        }
        client.Disconnect();
    }
    button5.Enabled = true;
    label41.Text = "Розпочинайте відпрацювання";
}
```

Після успішної відправки даних користувач може натиснути на клавішу «Почати виконання завдання». За допомогою наступного коду буде відкрито нове вікно та сформовано SSH запит, що розпочне роботу робота [31].

```
Private void timer1_Tick(object sender, EventArgs e)
{
    label2.Text = "Робот виконує завдання, зачекайте";
    using (var client = new SshClient(ip_address, username, password))
    {
        client.Connect();
```

```

        client.RunCommand("python3 /home/Diplom/Desktop/RobotProgram.py");
        client.Disconnect();
    }
    label2.Text = "Робот завершив виконання завдання";
    i--;
    if (i == 0) { timer1.Enabled = false; button1.Enabled = true; }
}

```

Після завершення виконання завдання користувач може ввести нові дані, або повернутися до головного меню, натиснувши кнопку «Повернутися».

3.3 Програмне забезпечення для одноплатного ПК RaspberryPi

Як вже було згадано раніше, функції RP у лабораторному стенді наступні:

1. Прийом файла Log.txt від ПК;
2. Зчитування даних з текстового файлу;
3. Прийом команди на початок відпрацювання руху між точками від ПК;
4. Розрахунок проміжних точок;
5. Відправка результатів розрахунку до МК STM32.

Програма для RaspberryPi написана на мові програмування Python за допомогою інтегрованого середовища розробки Thonny.

Зчитування та зберігання даних у масив виконується наступним чином [32]:

```

data = []
with open("log.txt") as f:
    for line in f:
        data.append([float(x) for x in line.split()])

```

Розрахунок проміжних точок виконаємо за допомогою написання функції. Оскільки швидкість переміщення змінитися миттєво не може (графік 1), потрібно використовувати лінійний закон зміни руху, як показано на рис., графік 2.

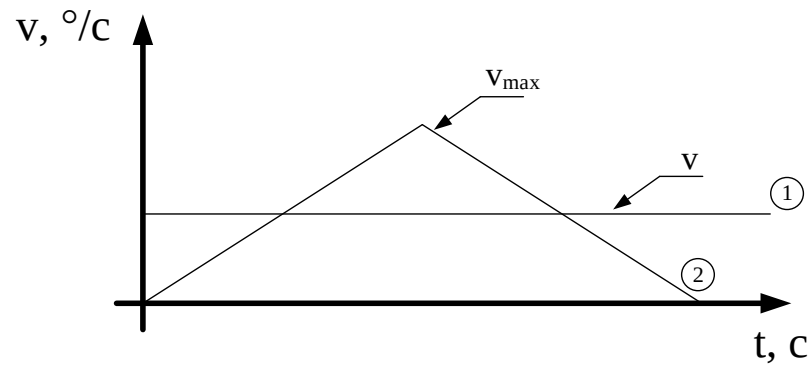


Рисунок 3.10 – Графік зміни швидкості РМ

З рис. можемо записати рівняння для зміни швидкості $v_{\max}$:

$$v(t) = \begin{cases} kt, \text{ при } t \leq \frac{t_{\text{пер}}}{2} \\ v_{\max} - k \left(t - \frac{t_{\text{пер}}}{2} \right), t > \frac{t_{\text{пер}}}{2} \end{cases} \quad (3.3)$$

де $k = \frac{v_{\max}}{t_{\text{пер}} / 2}$ – коефіцієнт нахилу, $t_{\text{пер}}$ – час переходу між точками.

Враховуючи, що $s = \int v(t) dt$ можемо записати вираз для зміни координати робота маніпулятора наступним чином:

$$s(t) = \begin{cases} \frac{kt^2}{2}, \text{ при } t \leq \frac{t_{\text{пер}}}{2} \\ \frac{v_{\max} t_{\text{пер}}}{2} + v_{\max} \left(t - \frac{t_{\text{пер}}}{2} \right) - k \frac{\left(t - \frac{t_{\text{пер}}}{2} \right)^2}{2}, t > \frac{t_{\text{пер}}}{2} \end{cases} \quad (3.4)$$

Розрахунок та відправка проміжних точок виконуємо наступним чином (відправка даних через протокол UART реалізована за допомогою бібліотеки Serial):

```
def CalculateCCR (a1,b1 ,tex):
    q11=math.ceil(a1*11.11+1500)
    q12=math.ceil(b1*11.11+1500)
    v1=(a1-b1)/tex
    vmax1=2*v1
    k1=vmax1/(tex/2)
    nt=math.ceil(tex/0.020)
    i=0
    while i<nt:
```

```

if i==0:
    ex_time=0.02
else:
    end_time= datetime.now()
    delta=end_time - start_time
    ex_time=delta.total_seconds()
if ex_time>=0.02:
    if i<=(nt/2):
        f1=math.ceil(q11-11.11*(k1*(0.020*i)*(0.020*i)/2))
        i+=1
        start_time = datetime.now()
        part11, part12= number_transform(f1)
        a=[part11,part12]
        byte_array=bytearray(a)
        x=ser.write(byte_array)
    if i>(nt/2):
        f1=math.ceil(q11-11.11*((vmax1*(0.020*nt/2))/2+vmax1*(i*0.020-0.020*nt/2)-(k1*(i*0.020-
0.020*nt/2)*(i*0.020-0.020*nt/2)/2)))
        i+=1
        start_time = datetime.now()
        part11, part12= number_transform(f1)
        a=[part11,part12]
        byte_array=bytearray(a)
        x=ser.write(byte_array)

```

Для того щоб правильно передати проміжні розрахунки та зменшити розмір посилки розіб'ємо результати обчислення функції на два двійкових числа за допомогою трьох наступних функцій:

1. Функція перетворення десяткового числа в двійкове:

```

def decimal_to_binary(num1, num2):
    binary_num1 = format(num1, '08b')
    binary_num2 = format(num2, '08b')
    shifted_binary_num1 = binary_num1[1:]+ '0'
    shifted_binary_num2 = binary_num2[1:]+ '0'
    return shifted_binary_num1, shifted_binary_num2

```

2. Функція розбиття числа на дві частини:

```

def number_transform(number):
    part1=number//64
    part2=number%64
    part1, part2=decimal_to_binary(part1, part2)
    part1, part2=decimal_to_binary(part1, part2)
    return part1, part2

```

3. Функція перетворення двійкового числа в десяткове:

```

def binary_to_decimal(num1, num2):
    decimal1 =int(num1, base=2)
    decimal2 =int(num2, base=2)
    return decimal1, decimal2

```


3.4 Інструкція до використання стенду

1. Виконати розв'язок задачі кінематики, згідно з початковими вхідними даними, що надані викладачем.
2. Ввімкнути живлення пристроїв RaspberryPi Zero та мікроконтролера STM32.
3. Створити Wi-Fi мережу, до якої під'єднати персональний комп'ютер та RaspberryPi Zero.
4. Ввести початкові дані (кількість точок, узагальнені координати, час переходу та паузи між точками) до застосунку на персональному комп'ютері, виконати перевірку даних (якщо перевірка не виконана – змінити вхідні дані).
5. Після перевірки даних виконати їх відправку та почати відпрацювання траєкторії руху робота-маніпулятора.
6. Під час пауз між переміщенням робота маніпулятора між точками виконати вимір та знайти координати положення РО.
7. Порівняти отримані дані з розрахунками з пункту 1.

Висновки до розділу 3

В даному розділі дипломного проекту було розроблено програмне забезпечення для кожної складової лабораторного стенду на основі РМ: розроблено застосунок користувача для персонального комп'ютера, що виконує запис, перевірку, відправку даних та подає команду на початок відпрацювання траєкторії руху; створено функції, що виконують зчитування вхідних даних, розрахунку проміжних точок руху РМ, перетворення та передачу результатів розрахунків для одноплатного комп'ютера RaspberryPi; розроблені підпрограми ініціалізації таймерів, USART протоколу та підпрограми запису прийнятих даних для мікроконтролера STM32. Додатково було розроблено алгоритм роботи студентів з лабораторним стендом.

4 МАТЕМАТИЧНА МОДЕЛЬ РОБОТА-МАНІПУЛЯТОРА У СЕРЕДОВИЩІ SIMULINK

4.1 Математичний опис робота маніпулятора

Згідно з структурною схемою на рис. 2.1 робот-маніпулятор складається з п'яти кінематичних обертальних пар. Для математичного опису кінематику руху маніпулятора необхідно визначити часові залежності зміни просторового положення та орієнтації РО. Для цього можна використати метод І. Денавіта та Р. Хартенберга, який був створений у 1955–1960 рр. Цей метод дозволяє визначити взаємне розташування сусідніх ланок маніпулятора. Особливістю цього метода є використовується спеціальна матриця переходу від однієї до іншої системи координат (4.1):

$$T_{i-1,i} = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.1)$$

В цій матриці перші три елементи четвертого стовпчика (a_{14} , a_{24} , a_{34}) характеризують зміщення центру координат однієї системи відносно іншої, а елементи a_{11} , a_{12} , a_{13} , a_{21} , a_{22} , a_{23} , a_{31} , a_{32} , a_{33} є косинусами кутів між кожною парою вісей обох систем координат.

Крім особливим для цього методу є і опис переходу від однієї системи координат до іншої за допомогою чотирьох елементарних рухів: двох обертальних і двох поступальних, які характеризуються двома кутами та двома лінійними відрізками (Θ_i , S_i , a_i , α_i).

Враховуючи однозначну визначеність вказаних рухів при переході від однієї системи координат до іншої, загальний вигляд матриці для будь-якої пари кінематичних ланок матиме наступний вигляд:

$$T_{i-1,i} = \begin{vmatrix} \cos(\Theta_i) & -\sin(\Theta_i)\cos(\alpha_i) & -\sin(\Theta_i)\sin(\alpha_i) & a_i \cos(\Theta_i) \\ \sin(\Theta_i) & -\cos(\Theta_i)\cos(\alpha_i) & -\cos(\Theta_i)\sin(\alpha_i) & a_i \sin(\Theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & S_i \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.2)$$

Якщо поступово розглянути кожну кінематичну пару починаючи від початкової, зв'язаної з інерціальною системою координат, і закінчуючи останньою, зв'язаною з РО маніпулятора, можна визначити всі матриці переходів. Добуток цих матриць дасть матрицю (4.3), яка визначає зв'язок між інерціальною системою координат і системою координат, пов'язаною з РО.

$$T_{0n} = \begin{vmatrix} a_{11}^{0n} & a_{13}^{0n} & a_{13}^{0n} & a_{14}^{0n} \\ a_{21}^{0n} & a_{22}^{0n} & a_{23}^{0n} & a_{24}^{0n} \\ a_{31}^{0n} & a_{32}^{0n} & a_{33}^{0n} & a_{34}^{0n} \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.3)$$

В цій матриці, перші три елементи 4-го стовпця матриці, а саме елементи a_{14}^{0n} , a_{24}^{0n} , a_{34}^{0n} є координатами центру захвату (РО), 1-й елемент 2-го стовпця і перші два елементи 3-го стовпця будуть напрямними косинусами осей z_n і y_n відносно x_0 і y_0 а саме: $\cos(x_0 \wedge z_n) = a_{13}^{0n}$; $\cos(y_0 \wedge z_n) = a_{23}^{0n}$; $\cos(x_0 \wedge y_n) = a_{12}^{0n}$. Кути між відповідними вісями можна визначити як $x_0 \wedge z_n = \arccos(a_{13}^{0n})$; $y_0 \wedge z_n = \arccos(a_{23}^{0n})$; $x_0 \wedge y_n = \arccos(a_{12}^{0n})$.

Таким чином, для отримання математичного опису кінематики руху маніпулятора достатньо скласти матриці переходу для кожної кінематичної пари та розрахувати їх добуток.

Для того щоб це зробити, необхідно спочатку розташувати системи координат на кожній ланці маніпулятора. Розташуємо вісі цих систем на кожній кінематичній парі робота-маніпулятора лабораторного стенду як показано на рис.4.1.

Наступним кроком, для опису кінематики, складемо таблицю кінематичних пар, за допомогою якої легко знайти матриці переходів. В цій таблиці проставимо значення параметрів Θ_i , S_i , a_i , α_i , а змінні параметри, які є узагальненими координатами, зобразимо буквою q_i . У досліджуваному роботі-маніпуляторі всі кінематичні пари є обертальними, а довжини ланок мають наступні значення: $L_1=8.8$ см, $L_2=10.5$ см, $L_3=7.5$ см, $L_4= 7$ см, $L_5= 10$ см.

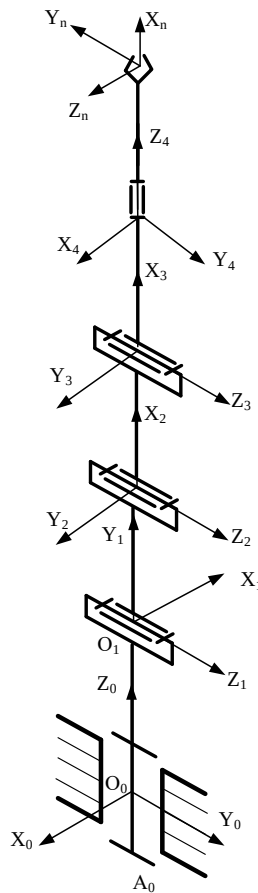


Рисунок 4.1 – Розташування систем координат робота-маніпулятора

Враховуючи правила переходу між системами координат запишемо таблицю кінематичних пар (табл.4.1).

Таблиця 4.1 – Параметри таблиці кінематичних пар для маніпулятора

Номер (i-1)-ї кінематичної пари	Тип (i-1)-ї кінематичної пари	Номер i-ої рухомої ланки	Параметри кінематичної пари			
			Θ_i	S_i	a_i	α_i
0	Обертальна	1	$q_1 + \pi/2$	L1	0	$\pi/2$
1	Обертальна	2	$q_2 + \pi/2$	0	L2	0
2	Обертальна	3	q_3	0	L3	0
3	Обертальна	4	q_4	0	L4	$-\pi/2$
4	Обертальна	5	q_5	0	L5	0

Пряма задача кінематики маніпуляторів полягає у визначенні положення його ланок у нерухомій (інерціальній) системі координат за

відомими значеннями узагальнених координат і при відомих значеннях кінематичних розмірів ланок. Це можна зробити знайшовши добуток матриць перетворень для кожної кінематичної пари.

Іншим підходом до створення математичної моделі є використання бібліотеки Simscape Multibody середовища Simulink. Ця бібліотека дозволяє створювати складні механічні системи, що складаються з окремих елементів (здебільшого представлених геометричними примітивами) та допускають відносний рух по відношенню один до одного.

При створенні математичної моделі робота-маніпулятора будемо вважати, що кожна ланка є циліндрами однакового діаметру з довжинами, що відповідають довжинам цих ланок. Створена модель наведена на рис.4.2.

Наведемо пояснення для кожної з частини схеми:

- Блок **Mechanism Configuration** використовується для налаштування напрямку і величину сили тяжіння.
- Блок **World Frame** визначає відлік координат
- Блок **Solver Configuration** використовується для налаштувань математичного розрахунку побудованої системи.
- Блок **Ground** створює «підлогу» для маніпулятора, на якій він буде знаходитися.
- Блок **Link** є стійкою маніпулятора.
- Блок **Rotate** використовується для створення приводу.
- Блок **Solid** з'єднаний з платформою, він використовується для обертального руху основної стійки.
- Блок **RigidTransform** виконує переорієнтування, тобто зміну координати розташування елемента.

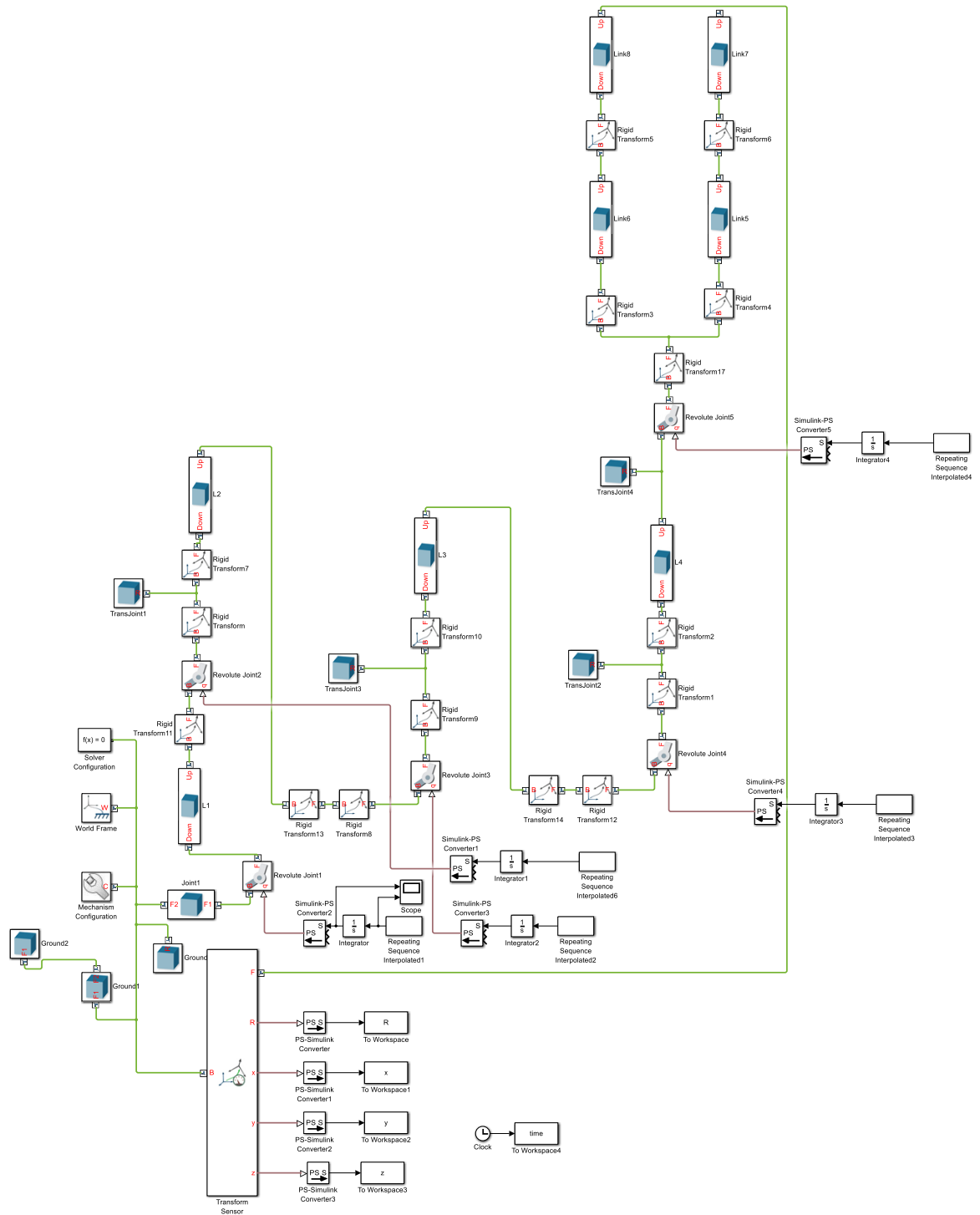


Рисунок 4.2 – Математична модель робота-маніпулятора у середовищі Simulink

4.2 Дослідження роботи робота-маніпулятора з використанням математичної моделі

У якості приладу, розглянемо перевірку розв'язку зворотної задачі кінематики для робота маніпулятора лабораторного стенду. Розв'язок задачі

полягає у знаходженні узагальнених координат маніпулятора по заданому положенню захвату. Задамося трьома точками в інерціальній системі координат: точка 1 – $x = 0.2$ см, $y = 0.12$ см, $z = 0.06$ см; точка 2 – $x = 0$ см, $y = 0.18$ см, $z = 0.13$ см; точка 3 – $x = -0.23$ см, $y = 0$ см, $z = 0.08$ см.

У результаті розв'язку даної задачі шляхом нелінійного математичного програмування отримані наступні узагальнені координати для відповідних точок: точка 1 – $q_1 = -60^\circ$, $q_2 = -20^\circ$, $q_3 = -70^\circ$, $q_4 = -45^\circ$; точка 2 – $q_1 = 0^\circ$, $q_2 = 10^\circ$, $q_3 = -65^\circ$, $q_4 = -70^\circ$; точка 3 – $q_1 = 90^\circ$, $q_2 = -40^\circ$, $q_3 = -10^\circ$, $q_4 = -90^\circ$. Варто зазначити, що значення q_5 не впливає на координати РО, а лише визначає його орієнтацію. Для перевірки отриманих даних використаємо модель, що зображена на рис.4.2. Завдання узагальнених швидкостей будемо здійснювати за прямокутним законом зміни прискорення, при якому швидкості спочатку лінійно зростають до максимального значення, а потім спадають до нульового, як показано на рис. 3.10. Величини максимальних швидкостей розраховані згідно (3.3) таким чином щоб забезпечити досягнення вищенаведених узагальнених координат в точках 1,2,3. Час переходу між кожними сусідніми точками задано рівним 1 с, а час знаходження в кожній точці – рівним 2 с. Рух маніпулятора розпочинається з початкового положення, що характеризується нульовими значеннями узагальнених координат. Графіки швидкостей наведені на рис.4.3, а зміна узагальнених координат на рис.4.4.

Результати розрахунку координат РО в інерціальній системі координат наведені на рис.4.5.

Як можна побачити з наведених графіків, в точках 1,2,3 значення координат інерціальної системи відповідають початковим значенням.

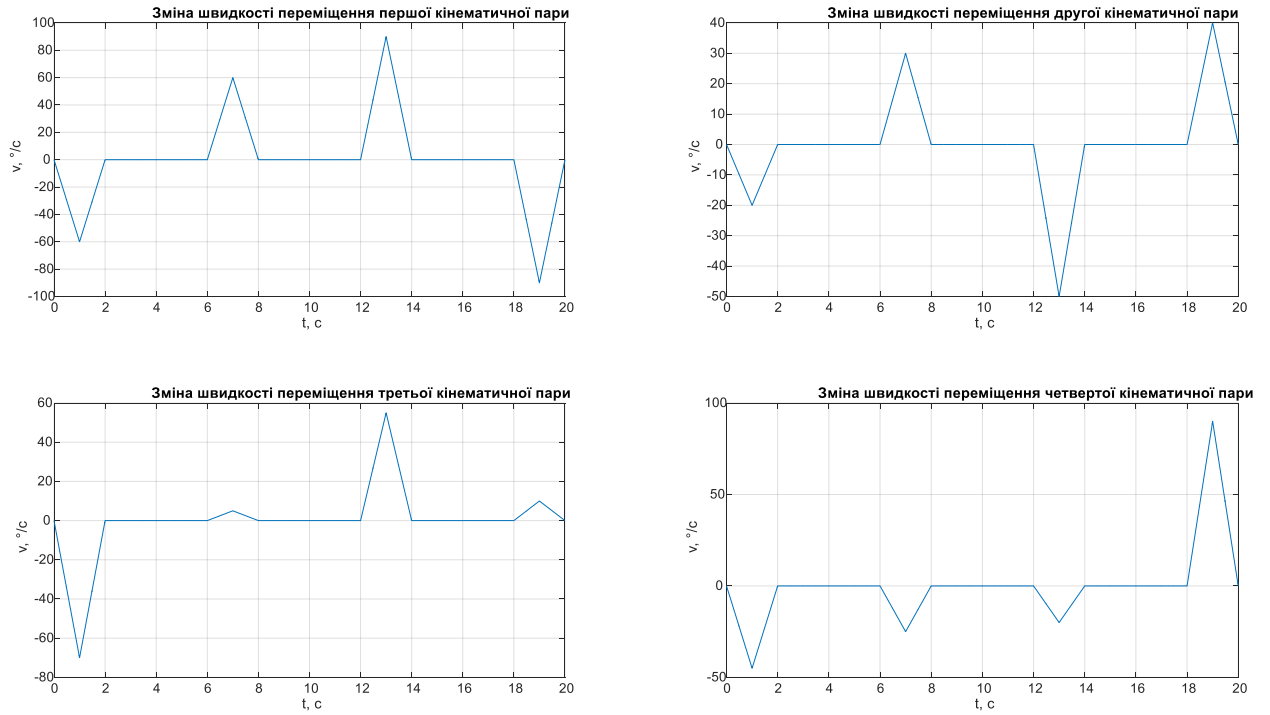


Рисунок 4.3 – Графіки зміни узагальнених швидкостей переміщення кожної з кінематичних пар

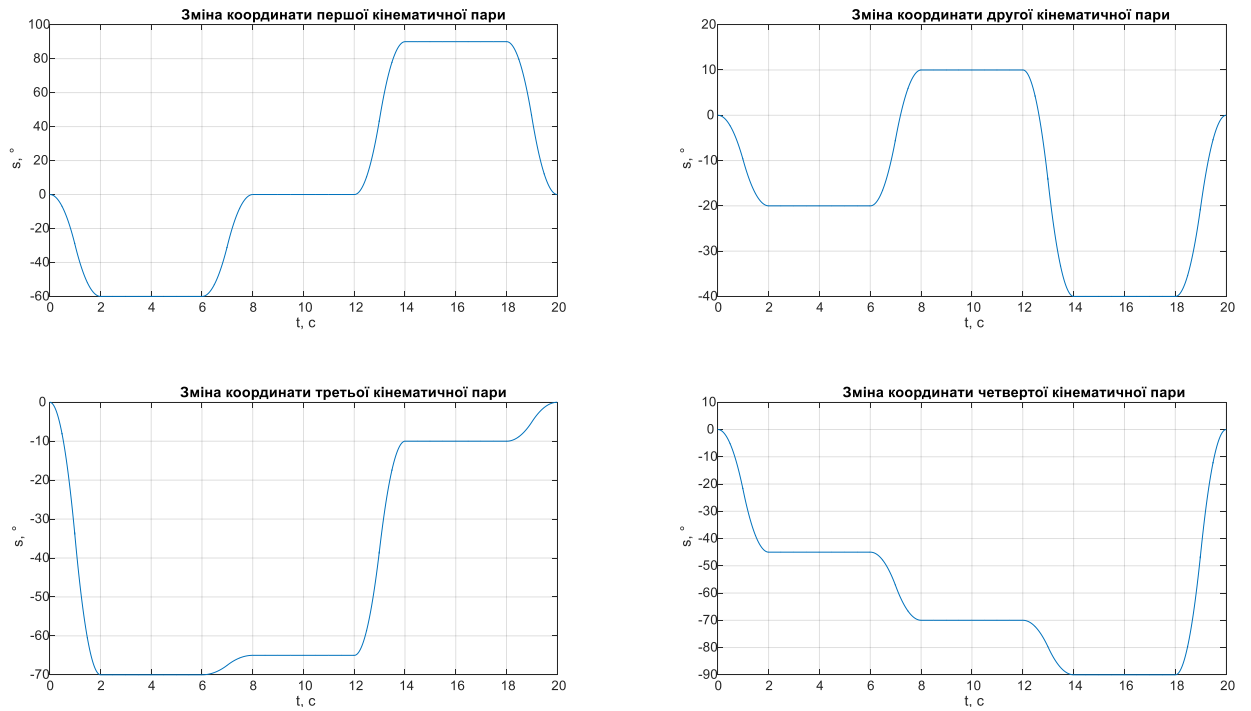


Рисунок 4.4 – Графіки зміни узагальнених координат кожної з кінематичних пар

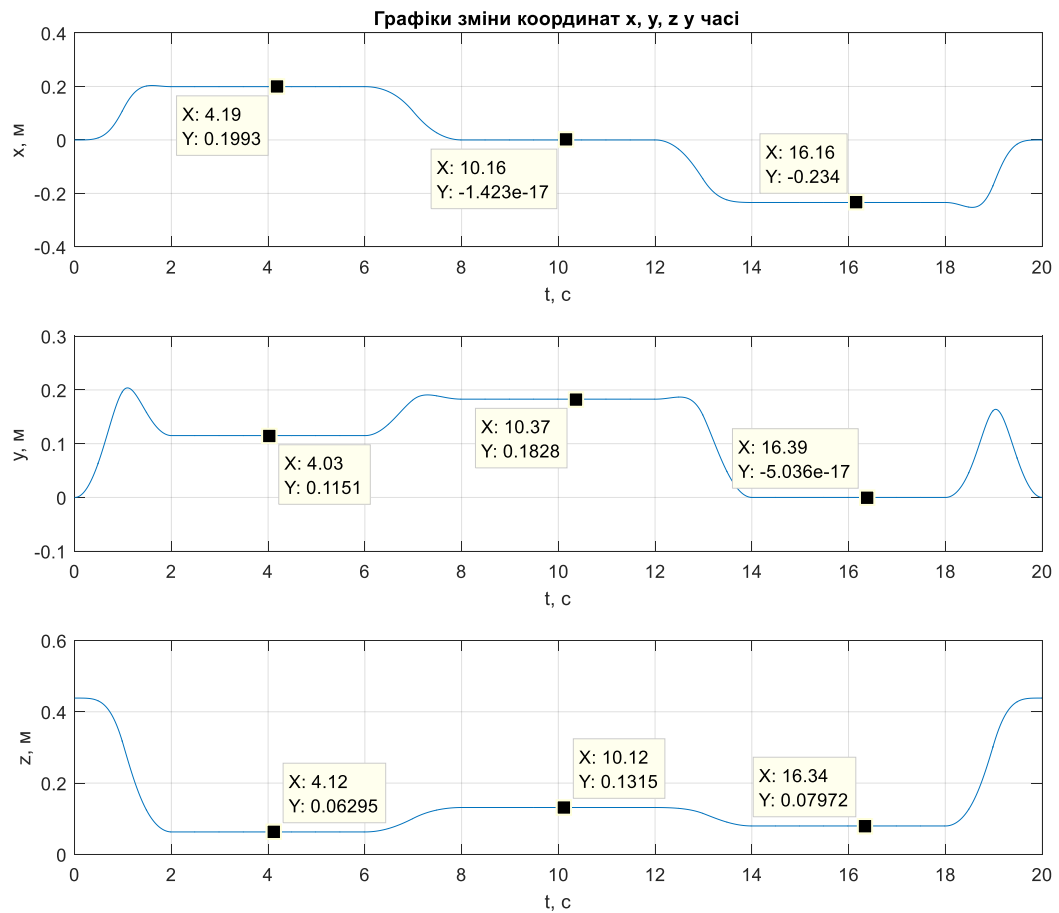


Рисунок 4.5 – Графіки зміни координат РО

Середовище Simulink надає можливості візуалізації створеної моделі та її руху при зміні узагальнених координат в часі. На рисунках 4.6-4.9 наведені скріншоти вікна Mechanic Explorer в моменти проходження маніпулятором початкової точки та точок 1,2,3.

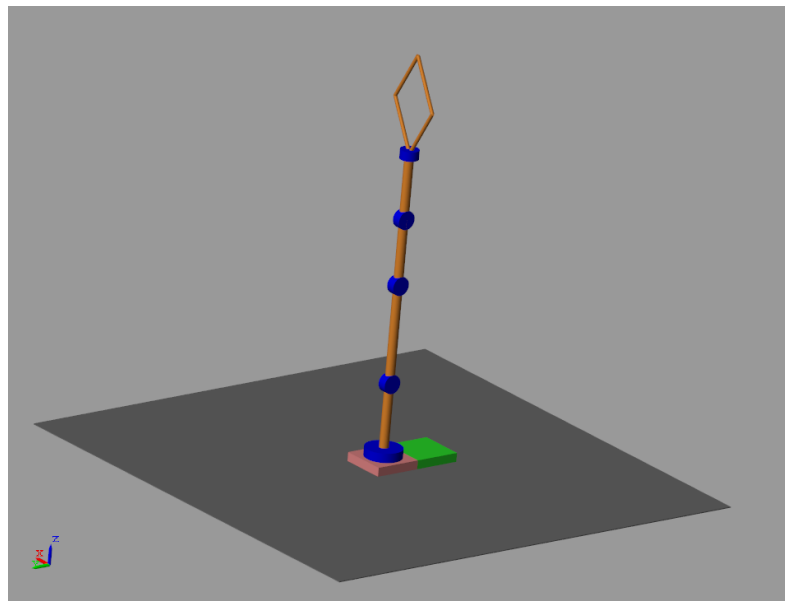


Рисунок 4.6 – Початкове та кінцеве положення РМ

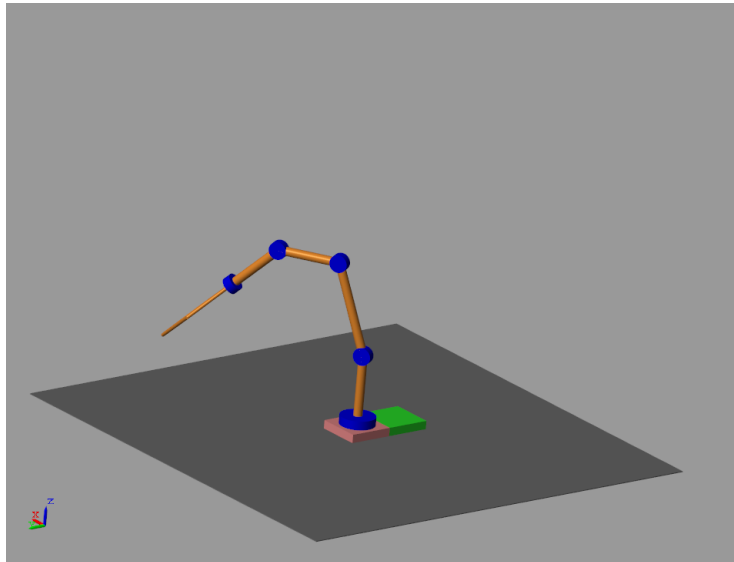


Рисунок 4.7 – Положення РМ у точці №1

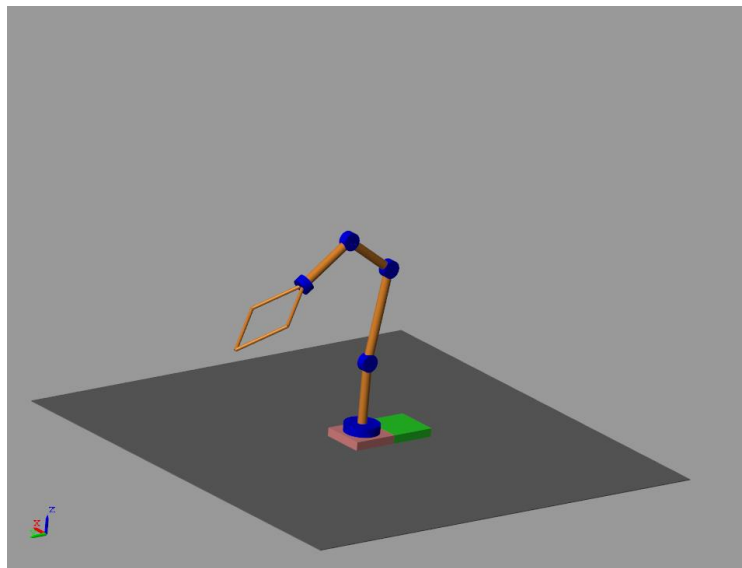


Рисунок 4.8 – Положення РМ у точці №2

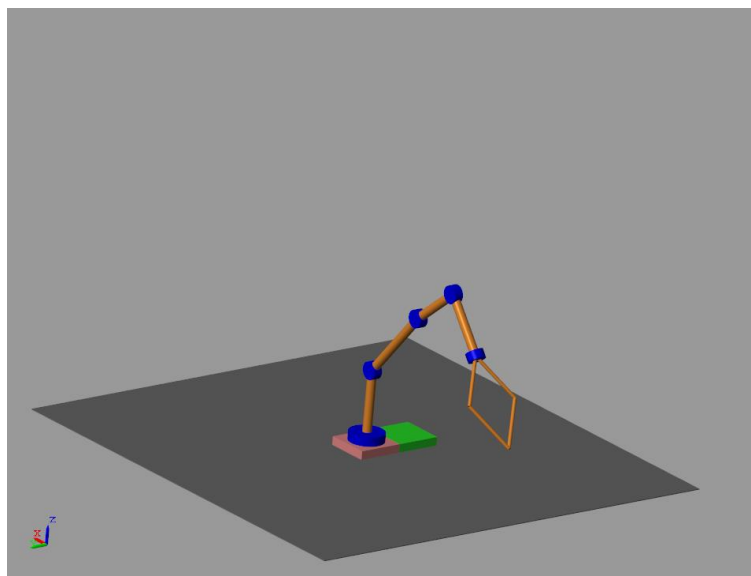


Рисунок 4.9 – Положення РМ у точці №3

Висновки до розділу 4:

В четвертому розділі дипломного проекту, використовуючи правила перетворення систем координат І. Денавіта та Р. Хартенберга, було розташовано вісі кожної з ланок РМ та складено кінематичну таблицю, на основі чого була створена математична модель лабораторного стенду у середовищі Simulink. Додатково навели приклад дослідження роботи робота-маніпулятора з використанням математичної моделі.

ЗАГАЛЬНІ ВИСНОВКИ

У ході виконання дипломного проекту було проведено загальний аналітичний огляд роботів маніпуляторів. Було виокремлено основні етапи розвитку РМ, їх класифікацію та сфери застосування.

Виконавши огляд зарубіжних та вітчизняних лабораторних стендів на основі РМ, що використовуються для освітніх цілей та наукових досліджень, визначили переваги та недоліки кожного з стендів. Як результат, було сформовано завдання до дипломного проекту.

Було виконано аналіз принципу роботи лабораторного стенду, за результатами якого була складена структурна схема стенду робота маніпулятора, до складу якої входить: персональний комп'ютер, блок обчислення траєкторії руху, гальванічна розв'язка, пристрій керування сервоприводами робота та сервоприводи.

За допомогою структурної схеми було розглянуто завдання кожної складової лабораторного стенду та виконано підбір технічного обладнання. У якості блоку обчислення траєкторії руху було обрано одноплатний комп'ютер RaspberryPi Zero, пристроєм керування сервоприводами виступає мікроконтролер STM32, що встановлено на плату розширення ZL-KPZ32, для реалізації гальванічної розв'язки було встановлено модуль TLP281, для переміщення кінематичних пар було обрано сервоприводи типу MG996R та DS3218MG.

Було розроблено програмне забезпечення для кожної складової лабораторного стенду на основі РМ: розроблено застосунок користувача для персонального комп'ютера, що виконує запис, перевірку, відправку даних та подає команду на початок відпрацювання траєкторії руху; створено функції, що виконують зчитування вхідних даних, розрахунку проміжних точок руху РМ, перетворення та передачу результатів розрахунків для одноплатного комп'ютера RaspberryPi; розроблені підпрограми ініціалізації таймерів, USART протоколу та підпрограми запису прийнятих даних для

мікроконтролеру STM32. Додатково було розроблено алгоритм роботи студентів з лабораторним стендом.

Використовуючи правила перетворення систем координат І. Денавіта та Р. Хартенберга, було розташовано вісі кожної з ланок РМ та складено кінематичну таблицю, на основі чого була створена математична модель лабораторного стенду у середовищі Simulink. Додатково навели приклад дослідження роботи робота-маніпулятора з використанням математичної моделі.

В результаті виконання дипломного проекту було створено лабораторний стенд на основі робота-маніпулятора та визначений порядок роботи з ним. Під час роботи з лабораторним стендом студенти мають змогу отримати та закріпити навички вирішення задач кінематики у рамках дисциплін з вивчення робототехніки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Навчальний посібник з дисципліни Маніпулятори та промислові роботи. Для студентів бакалаврів, спеціальності: 131 - Прикладна механіка, 133 – Галузеве машинобудування, / Укладачі.: Михайлов Є. П., Лінгур В.М. – Одеса: ОНПУ, 2019. - 233 с.
2. Brief History of Robots. URL: <https://www.linkedin.com> (дата звернення 11.06.2024).
3. Universal Robot UR5. URL: <https://www.todsystem.com> (дата звернення 11.06.2024).
4. ABB IRB120. URL: <https://new.abb.com/> (дата звернення 11.06.2024).
5. LR Mate 200iD. URL: <https://www.fanuc.eu> (дата звернення 11.06.2024).
6. KUKA KR 6 R700-2. URL: <https://www.kuka.com> (дата звернення 11.06.2024).
7. Yaskawa Motoman MH5: <https://www.robots.com> (дата звернення 11.06.2024).
8. DENSO VS-050/060: <https://www.denso-wave.com> (дата звернення 11.06.2024).
9. Mitsubishi MELFA RV-35. URL: <https://kck.ua/> (дата звернення 11.06.2024).
10. Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни «Основи робототехніки» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Інтернет речей» спеціальності 121 «Інженерія програмного забезпечення» денної форми навчання [Електронне видання] / Реут Д. Т. – Рівне : НУВГП, 2022. – 50 с.
11. Солоня О.В., Купчук І.М., Замрій М.А. Інтелектуальні мехатронні системи керування. Методичні вказівки до виконання практичних та самостійних робіт для студентів денної та заочної форми навчання другого (магістерського) освітнього рівня галузі знань 14 Електрична інженерія

спеціальності 141 Електроенергетика, електротехніка та електромеханіка.
Вінниця: ВНАУ, 2023. 110 с.

12. Національний ТУ «Дніпровська політехніка». Лабораторія мехатроніки та робототехніки Interpipe Mechatronic Lab1. URL: <https://elprivod.nmu.org.ua> (дата звернення 11.06.2024).

13. MG996R Datasheet (PDF). URL: <https://www.alldatasheet.com> (дата звернення 11.06.2024).

14. Kollmorgen. How does a servo Motor Work?. URL: <https://www.kollmorgen.com> (дата звернення 11.06.2024).

15. DS3218MG Datasheet. URL: <https://arduino.ua> (дата звернення 11.06.2024).

16. Mainstream Performance line, Arm Cortex-M3 MCU with 32 Kbytes of Flash memory, 72 MHz CPU, motor control, USB and CAN. URL: <https://www.st.com> (дата звернення 11.06.2024).

17. Digital tutorial ZL-KPZ32. URL: <https://www.bilibili.com> (дата звернення 11.06.2024).

18. Raspberry Pi Cookbook by Simon Monk Copyright © 2016 Simon Monk. -639 p.

19. TLP281 Datasheet. URL: <https://arduino.ua> (дата звернення 11.06.2024).

20. PHPNET. Standard PHP Library (SPL). URL: <https://www.php.net> (дата звернення 11.06.2024).

21. STM32 PWM Example - Timer PWM Mode Tutorial. URL: <https://deepbluembedded.com> (дата звернення 11.06.2024).

22. STM32F103-Keil Examples. URL: <https://github.com> (дата звернення 11.06.2024).

23. STM32F103 PWM (Pulse Width Modulation) signal generation using internal timers, keil MDK-ARMv6 and Stmcubemx IDE. URL: <https://www.engineersgarage.com> (дата звернення 11.06.2024).

24. STM32CubeMX Tutorial Series:PWM. URL: <https://www.waveshare.com> (дата звернення 11.06.2024).

25. StackOverflow. STM32 PWM Generation. URL: <https://stackoverflow.com> (дата звернення 11.06.2024).
26. PWM control and Dead Time Insertion. URL: <https://hackaday.io> (дата звернення 11.06.2024).
27. STM32. Програмування STM32F103. TIMER. PWM. URL: <https://blog.avislab.com> (дата звернення 11.06.2024).
28. IT Master. UART протокол. URL: <https://itmaster.biz.ua> (дата звернення 11.06.2024).
29. Microsoft Learn. Desktop Guide (Windows Forms.NET). URL: <https://learn.microsoft.com> (дата звернення 11.06.2024).
30. Stackoverflow. C# Connect via SSH and changing the file content. URL: <https://stackoverflow.com> (дата звернення 11.06.2024).
31. RaspberryPi Forum. Python On Rasberry Pi Via SSH. URL: <https://forums.raspberrypi.com> (дата звернення 11.06.2024).
32. W3School. Python Read Files. URL: <https://www.w3schools.com> (дата звернення 11.06.2024).